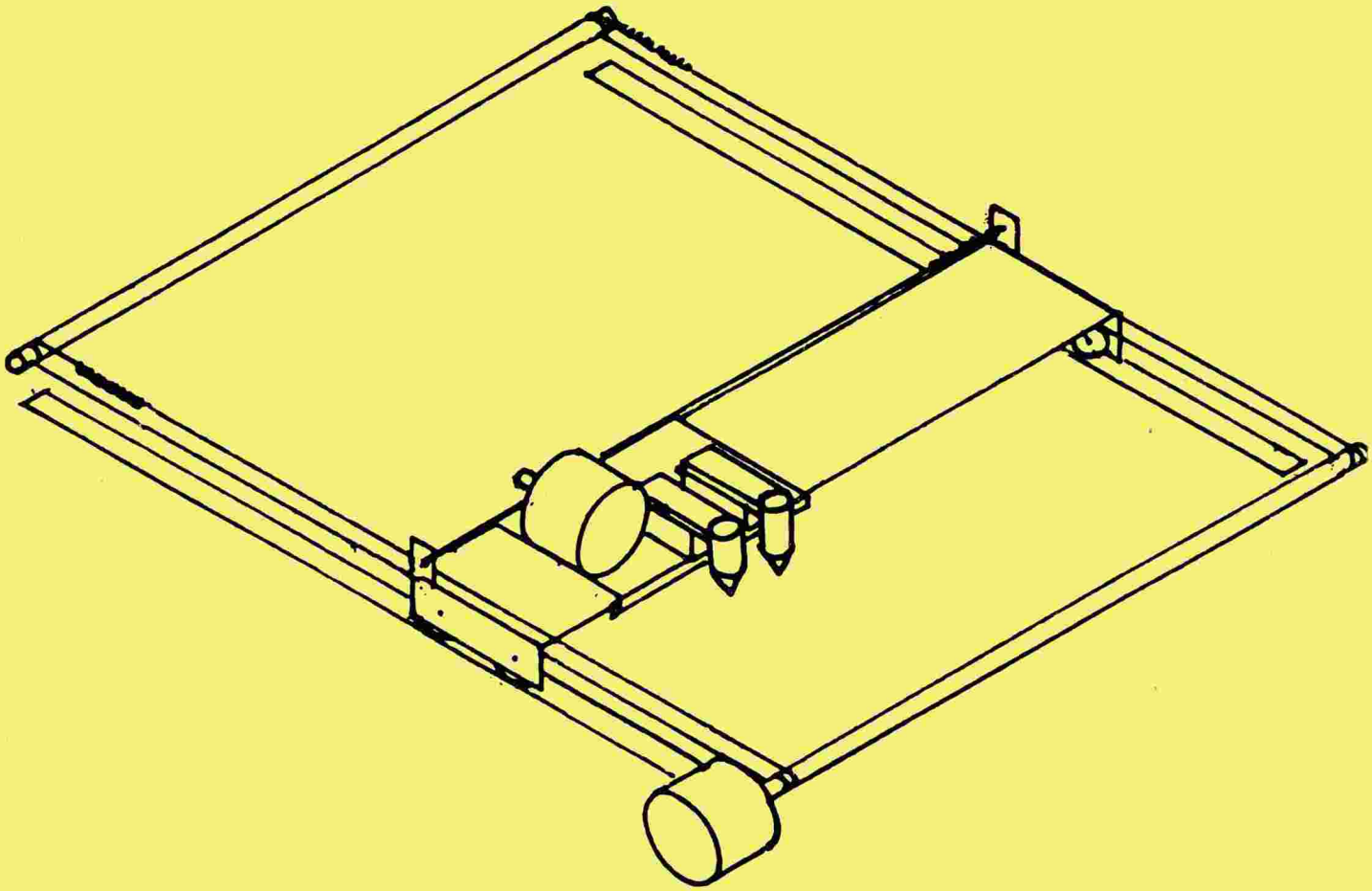
ISSUE Nº18

TI*MES



AUTUMN 1987

IN THIS ISSUE.....

PAGE

1 Editorial
2 Committee News
5 Life After 99? Dave Trevorrow
8 Quick Perpetual Calendar Ex Bas Program G. Ballinger
10 Rambles Stephen Shaw
20 Loadmaker Ex Bas Program
24 Tips From Tigercub Jim Peterson
27 Group Library Catalogue
35 Logo Christmas Logo 2 program TISHUG
37 Basic Programming TISHUG
40 Basic Beginnings-Variables Christina
43 Programming Basic to Assembly Geoffrey Coan
46 Breakout Basic Program Geoffrey coan
48 Realists Computer Glossary Tiny Tim
50 Using Ext Basic II Plus Peter Walker
52 DIY Removing the Alpha Lock Problem Geoffrey Coan
54 Xmas Tree Basic Program Maurice Rymill
58 Graph Plotting John Stocks

COVER PICTURE A Plotter for the TI

TI99/4a USERS GROUP UK and TI*MES magazine 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. The annual subscription runs May to May each year. Details from the Group Secretary on request. Editorial etc is provided by group members other WORLD WIDE TI User Groups and other related sources. Views expressed are those of the writer and not necessarily those of TI99/4a Users Group UK. Whilst efforts are made to ensure accuracy no responsibility can be accepted by TI99/4a Users Group as a result of applying such information found within the pages of TI*MES. You are invited to contribute copy for publication in TI*MES. Please submit copy on A4 ONLY. This must be TYPED AND SUPPORTED WITH A DISK OR TAPE if a program is included. Last date of acceptance of copy for next issue is 20th DECEMBER FOR PUBLICATION IN JANUARY. Unaccepted material will be returned only if accompanied by a S.A.E. No material may be reproduced without credit to the author and TI99/4a USERS GROUP UK. MICRO COMPUTERS OTHER THAN TI99 COMPATIBLES WILL NOT BE ADVERTISED FOR SALE IN TI*MES.

TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES
AUTUMN ISSUE NUMBER EIGHTEEN

4, Thornton Lane, Ulceby, South Humberside DN39 6SR (046 98 404)

T I 9 9 / 4 a U S E R S G R O U P U. K.

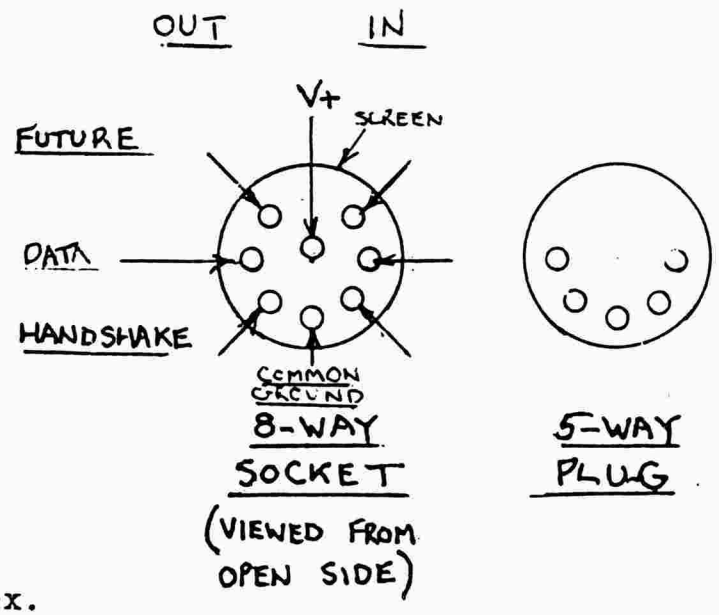
EDITORIAL

by Joint Editor Alan Bailey

Writing as a member of the unexpanded majority, I wonder if there are many others who would welcome a means of escape from the secretive and convoluted world created by the Texas system designers. The high cost of dedicated peripherals and associated software must reinforce this inclination. Advantage can be taken of lower cost memory and mass storage disc systems if suitable interfaces can be provided. The simplest seems to be the serial interface, and I intend to make an RS232, simplified version, to try this approach. I expect it will be distressingly slow, rather like my typing, but since my interests are purely amusement and wholely non-commercial, this does not matter. One of the first applications will be to send output to a teleprinter to give me hard copy for program debugging.

A more distant goal is the use of the generally available disc drives. Connection to the drive itself is more or less standard, but the control is a different kettle of rather smelly fish. Some sort of translator, perhaps a look-up table, is necessary. May be a case for one of the large, inexpensive EPROMS now available.

For serial interfaces I should like to draw your attention to a new standard based on the use of 5-way(180deg.) and 8-way(concentric) DIN connectors, called S5/8, for obvious reasons! Devised by Andrew Hardie for, inter alia, the BSI, and intended for c-mos e.g.HC14, when it takes only ImA at 5V+/-10%, it has a very logical and attractive structure. D-devices have power supplies, and have 8-way socket outlets. They are inter-connected by 5-way cables with 5-way free plugs, i.e. without power supply. S-devices rely on D-devices for their power, at least as far as the interface is concerned, and are connected by captive cables with 8-way plugs. The standard data rate is 9600bit/S, but handshaking pins are provided! This is a local peripheral connection system only. 1.5m length is suggested, with screened plugs and cables. The great advantage is cost and availability owing to the audio market. Although c-mos is the modern termination, bipolar is not excluded. Gemini Computers for example are understood to use S5/8 with bipolar outputs on their Challenger.



Shown here are the pinouts and assignments.

Best wishes,

Alan Bailey
14, Shelley Grove, Loughton, Essex.

COMMITTEE NEWS * COMMITTEE NEWS * COMMITTEE NEWS * COMMITTEE NEWS

HELLO again, you all out there! Hope you've survived the SCHOOL HOLIDAYS and the GETTING-BACK-TO-SCHOOL DAYS. My eldest son had to fly his cosy nest at our quiet village school to join the large comprehensive in the next town. I'll just say that the last four weeks have added greatly to the familys life experiences! In July I was confident that I would prepare the magazine as things came through in the post. But, on the first day of the HOLIDAYS, as I sat happily typing, my boys(and a few others) were up to no good in the back garden. A visit from an angry neighbour caused a re-think. I realised I'd have to have a month off. Strangely enough, the accidents and subsequent hospital visits didn't occur until the return to school....

My contacts with TI people, and, yes, my interest in the TI, have increased since 'doing' the magazine and I have been lucky enough to obtain very cheaply, expansion and a printer. I've always dreamed of having the 'bits'. Thanks to all those who have helped. I'm still going to write Basic for our unexpanded users, though, as I'm having to learn it myself. We have tremendous fun with the printer, but I can't use it for the magazine as it does not have true descenders(the p's and q's etc do not have tails!) Does anyone have a program to put this write, I mean right, for me?

We have been having a recruitment drive via Ceefax, Prestel and by writing to past members of TI-Exchange. Peter Walker, our membership secretary has been working hard and we now have fifty more members since the last publication. He has sent this message for you:

"We are extremely grateful for the comments sent in with your renewals concerning what you would like to see from the new group and in TI*MES magazine. It will take us some time to wade through all your views and discuss them in committee so please bear with us! We hope we will be able to reflect your wants in later issues. In the meantime, we hope you enjoy this issue.

Peter Walker"

The ~~disk~~* librarian, Tim Anderson, has been busy re-cataloging the Group ~~disk~~ library, and is going to add his own collection to it. With this, and also Maurices' collection added to the old library there will be about 800 programs to chose from. The listing published in last TI*MES was of Maurices' own collection, and lots of people complained that it was misleading - the prices were different, the codes were confusing, and essential details (such as joysticks or expansion needed) were not available. So in this issue you'll find re-printed the old library catalogue, with its clear categories and reviews of each program. This is going to be the format from now on; the librarians will update it in

* Tim was not the disk librarian- he was the cassette librarian. This was cleared up in the next issue.

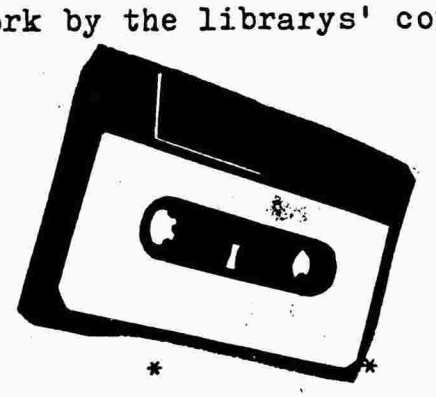
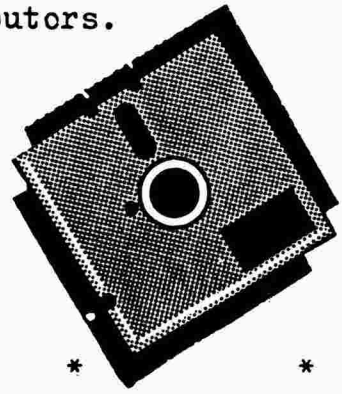
the same style, as time goes on, with their own contributions. The report from Maurice should clear up any other questions:

CASSETTE ~~AND DISK~~ LIBRARIANS' REPORT

The TI99/4a Users Group program library has been used by six different members over the last three months, despite there being some confusion over the pricing difference between the old TI99/4a Exchange library, and that of my personal cassette-based software library. In future the pricing will be the same for all programs, whatever the source:

£4.50 for any three programs
£1.75 each for any additional programs, or
£4.50 for multiples of three.

These fees include the cost of tape, handling, package and posting.
~~For Disk, add 95pence towards postage, packing.~~
I would like to take this opportunity to remind Group members that they can qualify for FREE programs, by submitting programs made up by themselves, which are entirely original.
A list of programs available from the old TI99/4a Exchange library is available in this issue of TI*MES.
Finally, let me ask Group members to please use the library. There are many interesting programs representing thousands of hours of programming work by the librarys' contributors.


Here's a little bit more news from Tim Anderson:
"On the 19th August, three members of the Group, including myself, got together for an evening in Lenzie, just outside Glasgow. We intend to hold more meetings, so if anybody is interested in joining us they can get in contact with me at: 16 Huntly Gardens, Dowanhill, Glasgow G12 9AT"

Geoffrey Coan, who helps on the committee with programming languages, is returning to University at the beginning of October, wishes me to publish his new address:
76 Roundcroft, Romiley, Stockport, Cheshire SK6 4LS

....Christina

DATA PROTECTION ACT AND ALL THAT

A message from your Membership Secretary, Peter Walker

You've probably noticed that your copy of TI*MES is sent to you using a computer printed label. It follows that your name and address is held on a computer file and the thorny question of Data Protection arises and the dreaded "Act". Now it so happens that as an "Unincorporated Members Club" we don't actually need to register under the Act in respect of members' data so long as we give you:

1 The opportunity to object to our holding data on you. (Please shout now if you insist!)

2 An assurance that any disclosure of this data is only made with your consent.

Having got over that formal bit, I should tell you that the only data I intend to hold is your name and address so that I can print off the address labels, and, for those that have informed us, your telephone number and what TI equipment you have. This latter information will help us get the correct balance of articles in TI*MES for both owners of expanded and non-expanded systems. So if you happen to be writing to us or renewing your membership, perhaps you could tell us whether you have:

1. Extended Basic 2. Extension Memory 3. Disk(s) 4. Speech
5. Printer 6. Modem

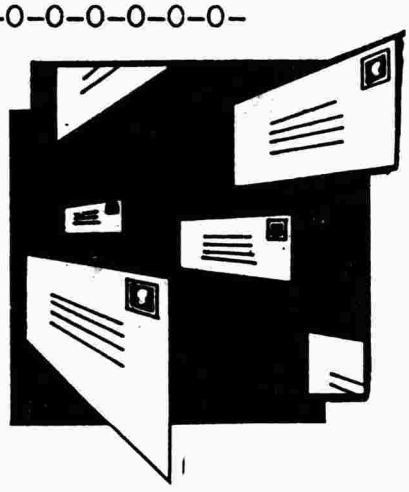
I'm holding the data using the Navarone DBMS system. This is a good deal faster and more flexible than TI's own PRK. This latter module has a number of major restrictions when attempting to store names and addresses, principally that the maximum field length for any data field is 15 characters and there are plenty of names of people, roads, towns and counties which exceed this. (Try Kingston upon Thames for size!)

All this reminds me that I keep meaning to write a review of the Navarone DBMS module... but I guess that will have to wait till another time.

Peter

-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-

YOUR LETTERS



Ian James of Bedworth, Warwickshire, wishes us good luck with TI*MES, and looks forward to many more years with his TI99/4a. He thought the Derby venue was most suitable, and easily accessable from most parts of the country. His suggestion for the 'what is Clive saying to Peter'competition is: "Is your balance sheet in Hex, or does OD stand for overdrawn?" Ian wins a module for that, and for writing such a nice letter. It is nice to hear from someone out there! (I must apologise for the photos in the last issue. I thought it was worth trying to print them, I did want you all to see them.) David Trevorrow wrote an epic letter, also

very nice, and with some helpful advice for us beginners in D.I.Y. repairs. We've re-printed it here in full. Thanks, Dave, for sharing your experiences with us. I hope more of you will do just that - it all helps us along the dark road to enlightenment! Two weeks ago, my son bought an electronics kit, a build-it-yourself burgular alarm. It was specially made for children to use, but as I looked at the diagram I really did feel as if I was lost in that "greenboard jungle" that Dave mentions. Neither of us had done any soldering before, but we built it. It didn't work, but at least we've had some practice at soldering! I can't wait until Christmas because I'm getting a how-to book on the subject. However, books are no substitute for practical experience, so if you've got some I hope you will be generous and share it. Anything that helps us get more out of our orphan computer will be appreciated. Speaking of Christmas, the next TI*MES will get to you in January, after all the festivities. So I'm taking this opportunity to wish you all a Happy Christmas and a peaceful New Year from all the Committee Members, best wishes from us all,

Christina

Christina

Life After 99?

by

Dave Trevorrow

19, Warbreck Road,
Orrell Park,
Liverpool L9 8EF

Wed 22/07/87

Dear Christina,

Is there life after 99? This was the awesome question which confronted me one bleak day last May. Instead of "PRESS ANY KEY TO BEGIN"... my T.V. responded with a deathly hiss, that ominous sound which signifies "DEAD COMPUTER".

Remembering the comforting advice on the cover of the Hitch-hiker's Guide to The Galaxy, I thought "DON'T PANIC". Always look at the simplest and most obvious things first. That's good fault finding advice. It's only the aerial lead come out. No? Then it's just the plug fuse blown. No, not that either. Could it be...? But wait, what about the fuse in the mains transformer box? Aghrr... It was intact. And so the Universe came tumbling all about me. THE T.I. WAS DEAD.

"Now, hold on", says I. "Did you not buy a service manual some time back, for just such a cataclysm as this? And do you not call yourself an electronics technician? "I didn't get where I am today being afraid of the insides of computers", I thought, bravely. Well, cover off and

lets get inside, not an entirely new experience as I had made for myself the "Matchbox" R.A.M. expansion not six months ago.

Wielding my trusty A.V.O. meter and switching on my "Cheapo" oscilloscope for good measure, I ventured to check some voltages. "Aha. The 12 volt supply reads 5 volts, the 5 volts supply is down to 2, and the minus 5 is OK. Something loading the positive lines, then. But what? And where, amongst all those chips, resistors, capacitors that look like resistors, and even coils that look like resistors?". Acute perplexity began to set in.

And then, a flash of insight. Well, a lucky observation really. There near the top right hand side of the main board sat a very charry looking resistor. Too charred to read its value, but obviously the faulty fellow. Obviously? Perhaps too obvious. The manual is not all that clear, especially as it is a somewhat crummy photocopy. You would think T.I. would supply an original for the price. Eventually, however, I tracked it down and changed it. But what is this? This resistor is in the cassette input circuit, and nothing to do with the fault. "Damned red herring", I muttered. "Should have known it anyway. The cassette was reading alright". So much for lucky observations.

And now for a real sorry tale. Jumping to the conclusion that the voltage regulators must be faulty, I decided to change them, even though past experience with such as these told me that it was very unlikely. T.I. are certainly of that opinion. The ruddy things were riveted in place. So, investing time and cash, (luckily they're not that expensive), I changed the 12 volt and 5 volt regs, plus the voltage comparitor for good measure. Guess what? Of course, no difference.

No difference? I should be so lucky. Whilst I contemplated the real cause of the fault, the 5 volt negative supply decided to disappear. "Now where did that go to?", I queried, dimly. "You didn't go and change the minus 5 volt reggy as well, did you?", you might ask. Well of course I did. Any cracks about divvi's at this stage will be found entirely in keeping.

Fool. Dolt. Imbecile. Look at that transformer again. The fuse is in the-5 volt supply. Yes, I know you checked it before. So check it again. There it was. Dead. Like my ego at this point.

Here I have to admit that defeat seemed to mock me from every chip, every line of printed circuit and every dismal waver of the A.V.O. needle as it hovered over the inadequate voltage readings. I contemplated the worst. I must open a distress channel and crave advice and guidance of the GUARDIANS OF 4A LORE. Across the great divide between Liverpool and Bedford came a very pleasant female voice, telling me what I did not want to hear: "We do not supply spare parts but we have a standard repair charge." WHAT? 46 QUID? Why, only in the last issue of TI*MES did I see an advert for a spare console for £48. Here a wave of determination overcame me. I would grasp the nettle and wade through those miles of track on the main board, double sided or no and even tackle mixed metaphores as they arose.

Here beginneth another l-o-n-g saga. But spare a thought for those non-technical readers who know not that of which you write, and those with technical know-how who are laughing their heads off. Let's cut it short a bit. I now started on a long journey, disconnecting coils, flying leads capacitors (busting a few as I went along-things are quite closely packed). At last I arrived at a junction. Literally. After disconnecting one end of a coil (L603, roughly mid-board), the blessed 5 and 12 volt lines

re-asserted themselves. I was nearing the end of the oddyssey. Now to trace the tracks after this junction. Well, one goes to pin 13 of U601, the clock driver chip. And the other.....

Oh no! The ultimate disaster. This track leads to pin 27 of U600. "What be that?", say you. "That", say I, "Is the centre of all things. The mighty TMS 9900. The C.P.U. itself. The very heart of TEXAS."

So had it come to this? There it lay, like some vast centipede with rigour mortice. It is said that even to number its legs is too great a task for most mortals, let alone desolder them all to replace it. But wait. All is not yet lost. Some delicate surgery might yet avert tragedy. With a silent plea that the 4s would be with me, I cut the track leading to pin 27. Then, resoldering L603 I switched on. And yes, the low voltage fault had returned. The faulty chip was the clock driver. Just to confirm it I bridged the cut I had made and desoldered the 12 volt supply pin on U601. Power resumed, and there it was. The voltages were OK with the C.P.U. reconnected. So it was a minnion and not the master that was dead.

I now obtained a replacement TIM9904ANL clock I.C. from Farnell Electronics in Leeds. And so after changing the vassal, the mighty brain (it was a heart before) sprang back to life. A happy ending you might think. Well it should have been. But no, there is more. Apart from leaving L603 disconnected and having the computer resetting itself after a few seconds, (how did it start up properly in the first place?), there was another little episode.

I mentioned earlier that I had made the Matchbox R.A.M. expansion some time ago, (thanks to the Funnelweb guys for this). Now I had connected the P.C.B. to the GROM extender via a D type plug. That was fine. But the connections to the memory select chip (74LS138) and the DBIN BAR lead form the 74LS04 chip went straight on to the board. Well I had clour coded them so as not to confuse them, so it wasn't possible to reconnect them wrongly was it? Oh beloved optimist. When I inserted Extended Basic and typed SIZE, what did I see?: 13928 b tes f s ack free 24488 by es of pr gr m sp ce fre . Those are not typing errors. What are those gaps? Have I now jiggered the memory expansion?

You are free to guess the next bit. Yes, I did buy four more memory chips. Result, same as before. Will this fool never learn? Only after this I changed the simpler, and much cheaper, memory select and data chips. Anybody want four 8K memory chips? Lets say ten quid to help me recover some of my foolish losses. (I'm saving for a disc drive).

And so, after much heartache, much wailing and gnashing of teeth I am finally back in circulation. Or am I? There can't be more to come. There can be more to come. Now some of the keys do not respond. Well it's only some of the connectors come adrift on the keyboard. Easy enough to put back. Well I should think so but after all this, and believe it or not I have not told all, I am beginning to wonder at my fate. I began to speculate upon events, linking fancy unto fancy, (like Edgar Allen Poe in "The Raven").

I have come to the following conclusion: It was not the T.I. that died that fateful day in May, but I. I died, and for my sins on earth I am consigned to wander for all eternity through the labyrinthine wastes of double sided tracks, alternating between hope and despair as I replace this, only to find something wrong with that, causing more havoc as I go. Only the pure undying love of a faithful woman can release me from this terrible destiny.....perhaps the noble lady Christina.....Oh no! She

is already another's. I just though that with a lovely name like that... don't tell the wife).

And here is another irony. I do not have and cannot really justify having TI-Writer or a printer. "So on what are you communicating?" you ask. On an ..strad, using Tasword, in my lunch hour, (Well, in some of the university's time too, for whom I work.

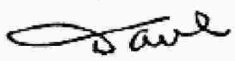
Well I should be able to reconnect those keyboard leads without too much bother shouldn't I? SHOULDN'T I? I might even get to send Stephen a Logo program before too long, this being one of the promises I made during my sojourn in the wilderness.

Maybe, just maybe the contents of this missive will help some other unfortunate traveler in the greenboard jungle tracks Remember, always check the simplest faults first. Follow your first instinct and don't go making complex, and costly diagnoses before CHECKING. (I knew those regulators should have been OK). It is always easy to be wiser, and poorer, after the event.

One day later. Somebody loves me. The curse of the Flying Double Dutchman is lifted. Last night I had a stripper in (wire-stripper, borrowed from work because it is better than mine). So, after removing the ribbon cable from the keyboard and re-dressing the ends, (yes, you use a stripper to re-dress. Funny, isn't it?), I re-soldered the connections and tried it out again. THE RELIEF WAS TANGIBLE. Waves of it pulsed through me as I returned from the dead. Memory is fully restored and I can run TI Logo II again.

And so, some Logo procedures are due to Stephen, and some complimentary words are due to you Christina for your very pleasant editorial in our new TI*MES. Definitely a feminine touch is perceptible in your nice chatty style. A welcome feature, missing from the computer world in general it seems to me. Thanks for taking the time to wade through this sea of troubles. Perhaps I may have something more pleasant to write about another time.

Yours,



David Trevorrow

==========

Program Listing

QUICK PERPETUAL CALENDAR
(by Geoffrey Ballinger)

The TI-99/4A has many good points, enough to make me desert my first love, the Sinclair Spectrum. It has to be conceded however that when a program has been loaded (it is much faster there) it is slow when it comes to the RUN bit.

Or is it? Key in this program. it uses all the ideas I could find to make it faster and more attractive .

If anyone with a printer is interested, a similar program that prints a full calendar for any year on a sheet of listing paper (or A4 sheet) equally fast, is available.

...listing overleaf

```
100 !
110 !~~~~~~~~~~~~~~~~~~~~~~
     ~ CALENDAR CONVERSION ~
     ~ spectrum to ti99/4a ~
120 !~ requires ext. basic ~
     ~ G.Ballinger  may'87 ~
     ~~~~~~~~~~~~~~~~~~~~~~
130 !
140 FOR N=3 TO 4 :: CALL COL
OR(N,7,1):: NEXT N :: FOR C=
5 TO 8 :: CALL COLOR(C,3,1):
: NEXT C :: CALL SCREEN(16)
150 FOR L=9 TO 12 :: CALL CO
LOR(L,6,1):: NEXT L
160 DATA 0,31,28,31,30,31,30
,31,31,30,31,30,31
170 DIM @$(12):: C$="" :: Q$
="" :: G$="~~~~~~~~~~~~~~~~~
~~~~~~~~~~" :: CALL CLEAR
180 GOTO 320 :: CALL KEY ::
CM :: DZ :: F :: G :: H :: J
 :: K :: LY :: MO :: S :: T
:: U :: W :: Y :: !@P-
190 LY=(Y/4-INT(Y/4)):: IF Y
/100=INT(Y/100)THEN IF Y/400
<>INT(Y/400)THEN LY=NOT(LY)
200 T=Y-1 :: S=T+INT(T/4)-IN
T(T/100)+INT(T/400):: RESTOR
E :: FOR F=1 TO MO :: READ C
M :: S=S+CM :: NEXT F
210 READ DZ :: IF MO=2 THEN
IF LY=0 THEN DZ=DZ+1
220 IF MO>2 THEN IF LY=0 THE
N S=S+1
230 H=2+S-(INT(S/7)*7):: IF
H=8 THEN H=1
240 IF W=24 THEN W=0 :: CALL
 CLEAR
250 DISPLAY AT(W+1,2):"Calen
dar",@$(MO);TAB(24);Y;" sun
mon tue wed thu fri sat" ::
G=4*H :: U=4*DZ-3
260 DISPLAY AT(W+4,G):SEG$(C
$,1,(29-G)):: DISPLAY AT(W+5
,2):SEG$(C$,(31-G),27)
270 DISPLAY AT(W+6,2):SEG$(C
$,(59-G),27):: DISPLAY AT(W+
7,2):SEG$(C$,(87-G),27):: IF
 U<143-G THEN 300
280 DISPLAY AT(W+8,2):SEG$(C
$,(115-G),27):: IF U<117 THE
N 310
290 DISPLAY AT(W+9,2)SIZE(27
):SEG$(C$,(143-G),ABS(141-G-
U)):: GOTO 310
300 DISPLAY AT(W+8,2)SIZE(27
):SEG$(C$,(115-G),ABS(113-G-
U))
310 DISPLAY AT(W+11,2):G$ ::
 GOTO 450
320 DISPLAY AT(1,1):"~";G$;"
 calendar program 1753-9999
~";G$
```

JANUARY FEBRUARY MARCH APRIL MAY JUNE JULY AUGUST

```
330 DISPLAY AT(6,1):"THIS PR
OGRAM IS BASED ON THE" :: DI
SPLAY AT(8,1):"CALENDAR IN U
SE SINCE 1752."
340 DISPLAY AT(10,1):"IT CAL
CULATES ALL LEAP YEARS" :: D
ISPLAY AT(12,1):"INCLUDING "
"CENTENNIAL"" ONES;"
350 DISPLAY AT(14,1):"i.e. T
HOSE DIVISIBLE BY BOTH" :: D
ISPLAY AT(16,1):"100 AND 400
.  AS AN EXAMPLE,"
360 DISPLAY AT(18,1):"2000 W
ILL BE A LEAP YEAR YET" :: D
ISPLAY AT(20,1):"BOTH 1800 A
ND 1900 WERE NOT."
370 DISPLAY AT(23,1):"~";G$
380 C$="1   2   3   4   5
6   7   8   9  10  11  12  1
3  14  15  16  17  18  19  2
0  21  22  23  24  25  26 "
390 C$=C$&" 27  28  29  30
31" :: RESTORE 400 :: FOR F=
1 TO 12 :: READ Q$ :: @$(F)=
Q$ :: NEXT F
400 DATA JANUARY,FEBRUARY,MA
RCH,APRIL,MAY,JUNE,JULY,AUGU
ST,SEPTEMBER,OCTOBER,NOVEMBE
R,DECEMBER
410 DISPLAY AT(24,1):"£ pres
s any key to proceed £" :: C
ALL KEY(0,K,J):: IF J=0 THEN
 410 ELSE CALL CLEAR
420 DISPLAY AT(24,2):"Year R
equired:" :: ACCEPT AT(24,16
)VALIDATE(DIGIT)BEEP:Y :: IF
 Y>9999 OR Y<1752 THEN 420
430 DISPLAY AT(24,2):"Month
<numeric> :" :: ACCEPT AT(24
,19)VALIDATE(DIGIT)BEEP:MO :
: IF MO<1 OR MO>12 THEN 430
440 IF Y=1752 AND MO<10 THEN
 420 ELSE 190
450 W=W+12 :: DISPLAY AT(24,
2):"P(+month) L(-month) (Y)y
ear"
460 CALL KEY(2,K,J):: IF J=0
 THEN 460
470 IF K=11 THEN MO=MO+1 ::
GOTO 500
480 IF K=12 THEN MO=MO-1 ::
GOTO 500
490 IF K=18 THEN 420 ELSE RU
N "DSK1.LOAD"
500 IF MO=13 THEN MO=1 :: Y=
Y+1
510 IF MO=0 THEN MO=12 :: Y=
Y-1
520 IF Y>9999 OR Y=1752 THEN
 IF MO<10 THEN 420
530 GOTO 190
```

Rambles

RAMBLES.
OCT 1987. Stephen Shaw.

Hello and welcome to another RAMBLES.

BY SPECIAL REQUEST: Elite font instead of condensed!

Thank you faithful reader for writing to me as requested in the last issue. Another letter this quarter would be just as welcome, and perhaps one more member may read this pot pourri and also write!

My address:
10 Alstone Road, STOCKPORT, Ches, SK4^5AH.

Back to issue 17 first, then some new bits...

COMPETITION... I WONDER, has anyone responded to the challenge on Page 8 of issue 17? So far only ONE entrant to the challenge on page 9... the (original) closing date was November 10th, so there is still time for EVERY member to tidy up an old program or write a nice new original one and send it in. TI BASIC can be very pleasant! ExBas, Logo, Forth, what-have-you (but not Pascal!). Show me you still program out there - or even that you used to program!!!
The closing date may slide a little... or the competition be suspended! There can be no competition without entrants!

THE SMART PROGRAMMER is still a NO NO with deliveries still not being made. The last issue received by me has a cover date of DECEMBER 1986 and was received on 22nd June 1987. This issue has some ads in it, as I was forearned by a little rumour, and MAY confirm another rumour that funds are running out, not just time... the publisher has failed to advise his subscribers of what he is up to, and must therefore accept responsibility for any rumours that appear.

GOOD SUPPLY/ BAD SUPPLY. This is a personal list! The following have given me GOOD and FAST service:
TENEX, TEXAMENTS, NOT POLYOPTICS, MICROpendium.

THE FOLLOWING ARE DOUBTFUL:
Pilgrims Pride, Tex Comp, Asgard, Smart Programmer, Chicago User Group, Boston Computer Society, Ryte Data.

On NOT POLYOPTICS I understand one member sent for SPAD XIII from them, enclosing US$34, and received his disk just ten days after posting his (airmail) order.

On Pilgrims Pride, word is the owner came close to losing his life - and that of his family- in bad flooding, but did lose his business documentation. Latest report is that goods remain undelivered!

On Tex*Comp, various grumbles, but they do have a good stock including some new module titles. I may be able to obtain goods from them "retail" if anyone wants anything (via a US friend).

On ASGARD, quite apart from making a mess of my last two orders, they have now REALLY ripped me off, advertising a new copy of FONT WRITER some four or five months before it was written - and sending me the old version. I understand the program author is also a little unhappy. If you want an Asgard program buy from TENEX or a reputable User Group (or possibly through me).

SOFTWARE LIBRARY... a little checking shows that 22 members have used the Disk library in the last two months. Anyone else out there with a disk drive? Just send two disks and return postage for a free update on what we have in! For example, CFS is now into Vn 7! Keep in touch!

BASIC COMPILER: Ryte Data have not charged my credit card and I have not heard from them. I have heard grumbles elsewhere about unduly long delivery times, and I have two very short reports on the Compliler not being so hot.
Sorry, I am unable to review the Basic Compiler for you, and must warn you off writing to yet another dealer!

REPEAT ANNOUNCEMENT...
Please write in! Questions, for published answers, suggestions (detailed!) for topics to explore together, NEWS, hints or tips to pass on...WAKE UP!!!

Member John Bingham advises me that NEATLIST is a program he uses very often, and suggests that lack of interest in this utility is due to:
a. No-One is programming any more (I'll believe that!)
b. There are few expanded systems (Not that few!!!).
--just to remind you, NEATLIST, the utility YOU aren't using, is the fourth most popular Freeware program in the USA...

THANK YOU to the members who kindly made it possible for me to provide my father with a TI system to help him learn to type without seeing anything. According to the SHELL BOOK OF FIRSTS the first working typewriter was built to enable a blind person to communicate, so it is appropriate in 1987 for a TI to be used to help a blind person use a modern typewriter.

The modern touch typist using ten fingers can place their fingers over the so-called HOME keys, locatable by having two keys marked with raised dots, and then all keys are easily usable.

However, when your hands are a little crippled and you can only use one or two fingers, standard touch typing is NOT so easy!

Using TE2, it is possible to provide INSTANT feedback, so important in any learning situation. I can report that with a little TE2 program my father is very quickly learning to touch type with just one or two fingers. And, an added bonus. The concentrated finger movement is loosening up the problem fingers too.

Amazing what you can do with an old computer...

Jim Ballinger has reminded me of something useful - the memory is very dusty and probably ancient, I have no idea where I first saw this, and certainly forgot all about it....
(The problem was noted by Regena in August 1981, but an incorrect fix was given!).

[post script: talk of the devil... Regena's old tip has just resurfaced, reprinted in the June 1987 issue of the MSP99 Newsletter, extracted from the Tasmanian TI Users Group - but without mention of Regena so possibly a re-discovery and another wrong solution!]

DATA.

Do you ever use DATA items in your programs? Ever LOAD and RUN a TI Basic/ Extended Basic program with DATA lines in it?

Now here's a funny thing.

When the computer has read the last DATA item, it then takes the trouble to look ALL the way through your program so it can make a note of the next item... even if there isn't one.

Consequently, and especially in a long program, reading the last DATA item can introduce a short delay.

Solution on next page>.....

EASY way round it. Add one or more DUMMY DATA items - if the computer wants to take time off looking for something that isn't there, lets GIVE it something! Then it can get on with our program!
... and bearing on this, it seems to be more efficient to have all your DATA items together, rather than spread them around your program.
Whether it is best to put your DATA at the start or end of your program depends on whether you use RESTORE and the ratio of the number of READs to the number of RESTORES- which varies with program length. So try everything!

Hmmm. Odd. I can't find a reference in TI*MES to copying TI FORTH disks - apart from an early erroneous mention.
To keep the record complete...
The disk manager MODULE copies disks as a FILE copier. It is NOT happy copying Forth disks. Important information can be missed!
To copy TI FORTH disks, you CAN use the Forth utility FORTH-COPY, or you can use an TRACK copier, or a SECTOR copier which copies the whole disk. A sector copier which references the bit-map will not function properly.

SPIDER BOP REVISITED
or
YOUR FIRST c99 GAME:

Look back to Page 50 of the last issue of TI*MES and you will find a game written for Basic or Extended Basic. Did you key it in? Try it!
Now lets have another look at it, but this time, using c99! c99 is available on four disks from the Group Disk Library, together with all the other files listed below. c99 Version 2.1 or higher is assumed.

To use this listing you will need the following files:
C99C, C99D, C99E, GRF1RF, GRF1, RANDOM;C, CONC;C, CSUP

First type in the following listing, using the Funlweb Editor set to Ed/As mode or use the Ed/As editor. Save the text as SPIDER;C
Now load C99 (you may use Funlweb loader Option 4, LOAD & RUN).
Just press enter for the first two questions, enter source name as DSK1.SPIDER;C and output name as DSK1.SPIDER/S.

(If you receive any error messages look at both the line reported and the line above it. I missed out two final semi colons at first! You will need to go back to your Editor to change the C source).

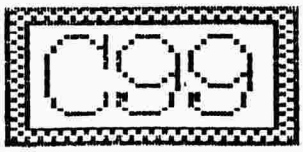
When the c99 program has finished, you must load the TI Assembler, and assemble the file SPIDER/S into SPIDER. Do NOT use any assembly options, not even R!!!

Now you are ready to play your game.

Select Option 4 of Funlweb Loaders (LOAD AND RUN) and load the following files, one after the other:
SPIDER, CSUP, GRF1
Then just press enter, then enter the start label: START and off you go.

You have just entered a C99 game. Not too bad was it! Just a note: speed 9 is slow, speed 1 is suicide!

THE CODE STARTS ON THE NEXT PAGE. Enter as printed!



```
#asm
 REF C$GPLL
HONK BL @C$GPLL
 DATA >36
 B *13
#endasm
#include "dsk1.grf1rf"
#include "dsk1.random;c"
#include "dsk1.conv;c"
int z,temp,a,b,c,e,k,s,j,tt,flg;
int d[32];
char score[8];
main()
{
while (0==0)
{ clear(); grf1(); screen(2);
  chrdef(096,"1818181818181818");
  chrdef(097,"000000FFFF000000");
  chrdef(104,"6699997E5595A5A5");
  chrdef(112,"3C7EFFFFFF7E3C3C");
  chrdef(113,"3C3C7EFFFF7E3C3C");
  e=0;
  b=12;
  tt=0;
  while (e++<9)
  { color(e+4,16,1);
  }
  color(13,16,1);
  color(14,14,1);
  color(15,3,1);
  e=0;
  while (e++<33)
  { d[e]=3;
  }
  locate(4,10);
  puts("SPIDER BOP 2");
  locate(22,5);
  puts("USE <ARROW> KEYS TO MOVE");
  locate(8,8);
  puts("WRITTEN WITH C99");
  locate(9,9);
  puts("BY JOHN BEHNKE");
  locate(24,3);
  puts("PRESS /ENTER/ TO BOP SPIDERS");
  locate(12,11);
  puts("SPEED 0-9?");
  k=0;
  while ((k<48)!(k>57))
  { k=key(0,&s);
  }
  j=k-47;
  j=j*100;
  clear();
  locate(1,10);
  puts("SCORE:");
  hchar(3,1,96,32);
  hchar(4,1,104,32);
  hchar(24,b,112,1);
  randomize();
  a=rnd(32)+1;
  flg=0;
  while (flg==0)
  { a=rnd(32)+1;
    ++d[a];
    k=0;
    while (k<j)
    { ++k;
    }
    vchar(d[a]-1,a,96,1);
    vchar(d[a],a,104,1);
    k=key(0,&s);
    while ((k==68)!(k==83))
    { hchar(24,b,32,1);
      if (k==68) { ++b; }
      if (k==83) { --b; }
      if (b==0) { b=32; }
      if (b==33) { b=1; }
      hchar(24,b,112,1);
      k=0;
   }
   if (k==13)
   { k=24;
     temp=5;
     while (--k>temp)
     { hchar(k,b,112,1);
       z=0;
       while (z<50) { ++z; }
       if (k==d[b])
       { honk();
         z=0;
       while (z<300) { ++z; }
       d[b]=d[b]-3;
       if (d[b]<5) { d[b]==5; }
       vchar(d[b],b,32,23-d[b]);
       vchar(d[b],b,104,1);
       locate(1,15);
       temp=24;
       itod(++tt,score,6);
       puts(score);
     }
   }
   while (k<24)
   { hchar(k++,b,32,1);
     z=0;
     while (z<50) { ++z; }
   }
  }
  if (d[a]==24)
  { hchar(24,a,32,1);
    k=0;
    if (a>b) { k=-1; }
    if (b>a) { k=1; }
    while ((k!=0)&(a!=b))
    { a=a+k;
      hchar(24,a,104,1);
      z=0;
      while (z<j) { ++z; }
      hchar(24,a,32,1);
    }
```

MORE→

```
        hchar(24,a,104,1);
        locate(18,10);
        puts(" GAME OVER ");
        locate(20,5);
        puts("PRESS /ENTER/ TO REPLAY");
        k=0;
        while (k!=13)
        { k=key(0,&s);
        }
        flg=1;
      }
    }
  }
 }
 /* END OF C SOURCE */
```

N.B.

[ ] = [ ]

{ } = { }

( ) = ( )

A FIRST LESSON IN EXTENDED BASIC

PROGRAMMING

by Jim Peterson

Extended Basic is nothing more than BASIC with a lot more words added. If you have learned anything about BASIC programming, it will also apply to Extended Basic.

A PROGRAM is just a numbered series of instructions to the computer, written in more-or-less-plain English, telling the computer to perform a certain task. The computer will follow these instructions in the order they are numbered, except when you tell it to GOTO or GOSUB to another part of the program.

The instructions are numbered by LINE NUMBERS. You can type these in, but it is easier to just start out by typing NUM and Enter. The computer will then automatically give you line numbers starting with 100 and advancing by 10 to 110, 120, etc. This is so that you can later squeeze more instructions in between using 105, etc. If you need to get out of automatic numbering, in order to correct a line or insert a line, just press Enter twice. To start automatic numbering again, just type NUM, space, and the next line number you want, such as NUM 130.

In Extended Basic, you can put several instructions under one line number, by putting a double colon (::) between them. But, while you are still learning, please DON'T! Why not? Well, when you tell the computer to do something it can't do, or can't understand, it will give you an ERROR message, either when you Enter the line or when you run the program, and it will tell you the line number that is causing the problem - but if you have several instructions under that line number, you won't know which one is wrong!

The first instruction we will learn is PRINT. This tells the computer to print something on the bottom line of the screen, and then scroll up one line. Try entering NUM,then -

```
100 PRINT 1
110 PRINT 2
120 PRINT 3
```

- and RUN it.

Now try -

```
100 PRINT A
110 PRINT B
120 PRINT C
```

- and RUN it. It printed a 0 three times, didn't it? Why? When you tell the printer to print anything other than a numeric digit (or a math symbol or decimal combined with a number) it thinks that you are telling it to print the VALUE of a VARIABLE. And if you haven't previously told it otherwise, that value is zero. Try this

```
100 A=10
110 PRINT A
```

So what is a VARIABLE? If you suffered through high school algebra, you may recall equations such as - S x T = D, where S equals speed and T equals time and D equals distance. You could give S and T any values you wanted to, in order to calculate how far something would go at a certain speed in a certain time. T and S and D are VARIABLES. We use them a great deal in programming and you will soon see why.

Now, suppose you really wanted to print the letter A. That's easy, just put it in quotation marks and the computer will know what to do.

```
100 PRINT "A"
```

In either Basic or Extended Basic, the instruction DISPLAY works just like PRINT.

```
100 DISPLAY 999
110 DISPLAY "HELLO"
```

Text scrolling up from the bottom looks rather "cheap", compared to those computers which display text from the top of the screen downward. In Extended Basic we can put the display wherever we want by using DISPLAY AT followed by a row and column number in parentheses. There are 24 rows on the screen and 28 columns when you are using PRINT or DISPLAY.

```
100 CALL CLEAR
110 DISPLAY AT(1,1):1
120 DISPLAY AT(24,1):24
130 GOTO 130
```

We slipped in a couple of new instructions there. CALL CLEAR just erases everything on the screen (actually it fills the screen with the blank space you get by hitting the space bar). GOTO tells the computer to go to another line number. In this case, it goes back to itself over and over and keeps the program running so it will not print READY and scroll that first line off the screen. Use FCTN 4 to get out of it.

Try experimenting with DISPLAY AT to put different numbers, words or phrases wherever you want them on thg screen. You will find that if you specify a row number greater than 24, the computer will just subtract 24 until it gets down to a number within range.

```
100 DISPLAY AT(25,35):"WHERE?"
```

In some programs you may see PRINT folowed by # and a number or variable. This is an instruction to print to a printer, to a disk, a speech synthesizer, or whatever. Actually you can print to the screen by -

```
100 PRINT #0:"SEE?"
```

but there is usually no reason to do so.

Now, a few words about print separators. Try this -

```
100 PRINT 1:2:3
110 PRINT 1,2,3
120 PRINT 1;2;3
130 PRINT "A";"B";"C"
```

See what happens? The colon (:) causes the computer to skip to the beginning of the next line before printing again. The comma (,) causes it to jump half the width of the screen before printing again. And the semi-colon causes it to print one item right after another EXCEPT that numbers are always printed with a blank space before and after them (a negative number has a minus sign (-) instead of a blank before it). Now try -

```
100 PRINT 1:2:3:
110 PRINT 1,2,3,
120 PRINT 4
130 PRINT 1;2;3;
140 PRINT 4
```

The colon after the 3 in line 100 was useless because the computer would advance to the next line anyway. The comma after the 3 in line 110 caused line 120 to print half a screen width after the 3. And the semi-colon after the 3 in line 130 caused line 140 to print immediately after the 3.

Do a lot of experimenting with this, until you know just what the print separators will do. Try -

```
100 PRINT "A","BCDEFGHIJKLMNOPQ"
```

See what happens when the item after the comma is more than half a screen in length? Now try this -

```
100 PRINT 1: : : : :2
```

And try this -

```
100 CALL HCHAR(1,1,42,768)
110 DISPLAY AT(5,1):"TEST"
120 DISPLAY AT(7,1)SIZE(4):"TEST"
130 DISPLAY AT(9,1):"TEST";
```

Line 100 just fills up the screen with something so I could show you that DISPLAY AT erases the remainder of the line unless you specify the length of what you will print with SIZE or, much easier, put a semicolon after the text.

"BASIC-COMPILER" from Ryte Data.

Rambles

No, Ryte Data have not sent me a copy, but your faithful reporter has tracked a copy down in order to report to you on its value.
Firstly, whatever it does, it is NOT a compiler, and anyone selling the program in this country would be a sure loser. My dictionary defines a compiler as a program which produces machine code.
Here is a quote from the documentation: " Principally the Compiler works in a manner similair [not my spelling error!] to the Basic-Interpreter that resides in the Console Rom's"
There is a little difference between a compilation and interpretation!
If you look back to TI*MES No 15, you will see some bench mark programs- not the final measure of a language but at least a rough guide.
TRIGLOG- using floating point arithmatic:
TI Extended Basic= 362 seconds.
Ryte Data "compilation" = 600 seconds!!!

TEXTSCREEN- using floating point arithmatic in TI ExBas but using Integer arithmatic for the compilation (supposed to be faster):
TI Extended Basic = 117 seconds
Ryte Data "compilation" = 168 seconds!
(Remember, Myarc XB = 73 sec!)

Quote "After the Compiler creates the new version of your program, the Loader can then run the new, faster version." Faster???

Quote: "With large programs it can be quite a time saver to use the Basic-Compiler". Possibly.
However, the program to be "compiled" must be somewhat shorter than usual! A program which starts as 19 sectors can become 35 sectors after "compilation" while a program starting at 24 sectors can become 37 sectors. It is not possible to indicate a maximum size the "compiler" can handle, you must use trial and error!
The error message is "PROGRAM TO BIG" (and no that isn't my spelling error). And this error message is not documented either.

You can of course chain programs together using "RUN DSK1.PROGRAM2"- and you know how long loading can take when you do that...
Faster on long programs? I can't say, as I could not find any programs of 20 odd sectors which would run successfully after "compilation", even after deleting the things the documentation says you can't do:
NO: DEFs, SUBs, ON ERRORs, LOAD, LINK, ON BREAK.
NOT DOCUMENTED, but you also cannot use ON WARNING.
Maximum string length is declared and subject to an absolute maximum of 64 characters.
I could not find any reason why the programs which failed did so, they were perfectly in accordance with the documentation!
I did notice that the cursor was not flashing as quickly as it usually does.

The title screen is (c)1984. It is now 1987... and NO major reviews? No reviews full stop... Guess nobody has made it work???
Suppose it did speed things up... it would not help the majority of games programs, as they are dependent on internal timing- it is no good if the alien destroys you before you see it! So rewriting would be necessary to slow some things down!
What does it do? It tags onto the memory image a table of variable addresses and jump addresses. That is all. Thus jumps and references to variables ONLY are liable to be speeded up, possibly. I cannot confirm any such theoretical speed up. In order to function there must also be an interrupt driven routine to bypass the normal system. The slower flashing of the cursor may indicate that the interrupt is taking too long.

----> CONTINUED.....

Ryte Data this time:"Otherwise (apart from the commands listed above) the Basic Compiler fully supports all Basic and Extended Basic statements AND it will compile multi-statement lines in Extended Basic. You do not need to rewrite your programs or use single statement lines". Something wrong somewhere!
This leaves us with a supplier subject to several recent complaints supplying a program carrying a false and misleading title of little use....

Ryte Data is also responsible for a "TI99/4A Compatible" scam, making moves which might make you think they were intending to produce a TI compatible, no they were just doing some market research to see if the Myarc Geneve was worth supporting.

My recommendation: Do not buy the so-called compiler, and do not send any money to Ryte Data.
I shall see if I can get together some product details of quality goods available from reliable suppliers for the next issue!!!

Until then, you may order with confidence from Tenex, Texaments, Not Polyoptics, SSI, Jim Peterson.
Details of other reliable (and prompt) suppliers welcome for next issue please!

The Fortran compiler mentioned in the last issue has been reported on by a couple of US users, who seem to feel that it is a good product for the money, and make the point that it is a SUBSET of ANSI 1977 FORTRAN, and not Fortran IV as advertised.
There are no character variables, which were only introduced in the '77 version, but character data can be handled using integer variables and arrays.
Some of the supported features are interger, floating point, double precision, if/then/else/elseif/endif, dowhile/enddo, as well as just IF and DO! Full multi dimensional array handling, extensions to support screen handling and TI CALLs.
The editor is reported to be about as poor as some main frame Fortan editors, but you can use TI Writer, the compiler detects errors OK but it not so hot on helping you find and correct them- again some main frame compilers are just the same! Preliminary reports are good. Any UK Fortran users out there to give us a full user report?
Supplier is Tenex and the price is US$50.

MACHINE CODE THROUGH EXTENDED BASIC...
Do you program machine code links for ExBas?
Ever suffer odd prangs- screen image table or pattern descriptor table or color table suddenly going whacko while everything else carries on? Two hints:
From anon, reprinted in April 1987 MSP 99 Newsletter:
"WARNING!!! Handling your own DSR PABs and buffer areas appears to be VERY dangrous in the XB environment. You can very easily destroy data in VDP memory currently in use by the XB interpreter. One way of handling this is to OPEN the file in XB and use the value at >833c to find the PAB and buffer area allocated by the file (see page 302 of the E/A manual). OR, save VDP memory you are going to use to RAM and then restore it to the original value after the DSRLNK"
"Also, be advised that the Get String Space routine provided via the GPLLNK appears not to work in the XB environment. EVENTUALLY (usually after the first garbage collection is forced) VDP memory in use by the XB interpreter will get corrupted and.... results may be unpredictable!"

CONSOLE PROBLEMS?

CONSOLE VOLTAGE PROBLEMS

Need a new module connector? Console lock ups? Color gone west?
Reports from Southwest 99ers, Tucson, through MSP 99 Newsletter are that some ageing power supplies (the hot ones in the console!) are allowing their output voltage to drift upwards (somebody can tell me which bit is going- capacitor or resister or diode or something!) such that measured output voltage on the
---continued--->

nominal 5V DC line has been up to 5.4V. Reducing that voltage back to the preferred 5.1V DC appears to solve an awful lot of aging problems. **5V DC**

If anyone has problems, and fixes them with a revitalised (or new) internal power supply please write and tell me, as this looks an interesting area for maintaining console life!

If your internal power supply PCB has a code on it 1053214-2, the 12v line may be unstable, jumping from 9.5 to 11.5 volts! The PCBs marked 1053201 seem to have good 12v lines. **12V**

We have a member with a MYARC DISK CONTROLLER CARD looking for fixes to the various programs which dont like it. MGs EXPLORER has a special version for the Myarc disk controller, but Advanced Diagnostics (and several freeware programs) wont work. Does anyone know WHY and is there a simple fix or is it hopeless? PLEASE WRITE!

==========

INITIATIVE...

Library disk H3+H4 contains a machine code program that copies the data base for the Adventure module from cassette to disk. It seems everyone who sends for it writes to tell me I have missed the copy program off... I haven't, and the program and document files are clearly called COPY! It just is not possible to send out paper documentation on public domain/ freeware items, and quite often documentation and instructions are minimal (sometimes they are overwhelming!).

Look for files in DV80 format- they are either machine code source code- which may include instructions- or they are straight forward instructions. Files in PROGRAM format MAY be in Basic or Extended Basic or Machine Code or merely comprise a memory image of data. Funlweb will tell you if they are machine code or Basic/XB.

IV254 files may be long Extended Basic programs.

DF80 files are usually machine code to be loaded with LOAD AND RUN.

PROGRAM files of 33 sectors, with several files differing only by the last letter, are machine code memory image files- you load only the first file, using RUN PROGRAM FILE or similar.

The Adventure modules use PROGRAM type files but they are neither Basic nor Machine code. They are a data base in memory image format, used by the program in the module.

And if you are interested in adventures on disk for your Adventure module why not send for the Disk Library Catalogue- send two disks and return postage!!!

LET'S ROUND UP THE MAVERICKS! by Jim Peterson

A maverick, for the information of you tenderfeet, is a young Texas critter which has lost its mama. There are over a million of them hiding in the closets of America, and I think it's time for a roundup!

There are perhaps 200, possibly 300, TI user groups in the United States and elsewhere in the world. A few boast of several hundred members, but some have no more than a dozen, and I doubt that the average is more than 50 users actually paying dues and attending meetings. That computes to at most 15,000 members of the "organized" TI world. Of course, there are many others who keep in contact by subscribing to those magazines which support the TI, and still others who are kept up to date on new developments by the catalogs from the big mail order houses. Still, no matter how you compute it, there are certainly well over a million owners of the TI-99/4A who have no way of knowing that our computer is still alive and well.

These people have read that Texas Instruments abandoned the computer. They have seen the supplies of hardware and software disappear from the big retail stores. Many of them bought their computer during the final suicide sales, therefore never got on the mailing list for the Texas Instrument newsletter.

And yet, relatively few of the TI-99/4A are showing up in the classified ads and in the garage sales. A recent national survey found that the TI-99/4A was owned by more people than any computer except the Commodore.

True, many of these owners are only interested in plugging in a module and playing a game. But some have a deeper interest - and even five percent of a million is an awful lot of people!

When I bought my TI, in March of 1982, I searched in vain through the articles and ads of every magazine on the newsstand, for anything relating to my computer. It almost seemed that there was a conspiracy of silence. I had taught myself to program, and written dozens of programs, before I finally made contact with the TI world. I was once a maverick, and I can sympathize with those who are mavericks now.

Is your user group dwindling away, as some of your members move on to bigger but not necessarily better computers, while others become so polarized in their interests that they have little in common with each other? Are your givers tired of giving to your getters, and your doers tired of being used by your users? Do you miss the enthusiasm and excitement of your first meetings, when everyone was learning together? Does your group need a transfusion of fresh blood? The donors are out there and waiting, if you can find them!

Do you want to see new hardware, new software, new publications for your computer? The bigger the market, the more that will be produced to be marketed. And the market is there - it just doesn't know that it's there!

The user groups are the only ones who can round up the mavericks. You can do it by publicizing your meetings, by letting the TI owners in your community know what you can do for them. You can get newspaper publicity and television publicity. Some of you are already offering classes in programming or in computer use to the general public, to the schools, to libraries, to senior citizens, to foster children, to the handicapped. These are very fine endeavors in themselves, and they can also bring the publicity which will attract new members. And here and there among those new members will be an ingenious hardware hacker or programming genius who will make our computer better than ever. JP.

OK GANG- GET OUT THERE...
There are still a lot of UK owners out of contact.
TI may have sold over FIFTY THOUSAND consoles in the UK. TIHCUC managed 4000 members. The two national groups now have under 200 each, and some of those belong to both groups!
There ARE UK Mavericks! I know! Every so often I receive a letter addressed to Stainless Software asking where modules can be bought... or perhaps someone has received a copy of my book with an old console, and writes to me as a result of that. Where possible I pass on details of the TI World! But... there are many more who do not write to me. Can YOU find them? They are out there, waiting to join us, waiting to learn from us, WAITING TO TEACH US. Stephen.

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

LOAD MAKER :

This EXTENDED BASIC program requires a disk drive. It will read the disk menu and create a special LOAD program with its own Menu for your disk.
Please refer to the further notes at the end of the listing!

```
100 ! LOADMAKER
110 ! VERSION XB.1.1
120 ! 29 DEC 84
130 ! by Jim Swedlow
140 ! from SPIRIT OF 99
      July 1987
150 DISPLAY AT(10,10)ERASE A
LL:"LOADMAKER": : : : : :"In
itialising..." :: @=1 :: DIM
 A$(26),A(26):: B$=CHR$(182)
&CHR$(181)&CHR$(199)
160 B=100 :: C$=CHR$(179)
170 OPEN #@:"DSK1.",INPUT ,I
NTERNAL,RELATIVE :: INPUT #@
:A$(C),D,E,F :: DISPLAY AT(1
6,@):"Disk ";A$(C);" * Free"
;F :: A(C)=LEN(A$(C))
180 INPUT #@:D$,D,E,F :: IF
D$="" THEN 220 ELSE IF ABS(D
)<>5 OR D$="LOAD" THEN 180
190 C=C+@ :: IF C<27 THEN A$
(C)=D$ :: A(C)=LEN(D$):: DIS
PLAY AT(17,@):"Reading:";D$
:: GOTO 180
200 DISPLAY AT(16,@)BEEP:"Th
e disk has more than 26   pr
ograms. Do you wish to    pr
oceed with the first 26   pr
ograms? Y or N?"
210 CALL KEY(3,F,D):: IF F=7
8 THEN CLOSE #@ :: STOP ELSE
 IF F<>89 THEN 210
220 IF C=0 THEN DISPLAY AT(1
6,@)BEEP:"No programs were f
ound on   this disk." :: CLO
SE #@ :: STOP
230 CLOSE #@ :: DISPLAY AT(1
6,@):"Reading completed": :
: : : :: OPEN #@:"DSK1.XXX",
VARIABLE 163,DISPLAY ,OUTPUT
240 DISPLAY AT(16,@):"--Maki
ng LOAD--line 100"
250 D$=CHR$(0)&CHR$(100)&CHR
$(131)&CHR$(32)&CHR$(80)&CHR
$(79)&CHR$(71)&CHR$(82)&CHR$
(65)&CHR$(77)
260 D$=D$&CHR$(32)&CHR$(76)&
CHR$(79)&CHR$(65)&CHR$(68)&C
HR$(69)&CHR$(82)
270 G,F=@ :: GOSUB 490 :: D$
=D$&CHR$(182)&CHR$(239)&CHR$
(236)
----to next column...
280 D$=D$&CHR$(181)&CHR$(199
)&CHR$(7+A(0))&CHR$(68)&CHR$
(105)&CHR$(115)&CHR$(107)&CH
R$(32)&CHR$(42)&CHR$(32)&A$(
0)
290 G=@-7*(C<19):: E=2 :: FO
R D=@ TO C :: E=E-(D>C/2)*(G
=@)*INT(C/2):: G=G+(D>C/2)*(
G=@)*14 :: IF D=C AND G=15 A
ND C/2<>INT(C/2)THEN G=8
300 F=E+D :: GOSUB 490 :: D$
=D$&B$&CHR$(A(D)+3)&CHR$(64+
D)&CHR$(32)&CHR$(32)&A$(D)::
 NEXT D
310 F=24 :: G=@ :: GOSUB 490
 :: D$=D$&CHR$(182)&CHR$(238
)&CHR$(181)&CHR$(199)&CHR$(1
7)
320 D$=D$&CHR$(80)&CHR$(114)
&CHR$(101)&CHR$(115)&CHR$(11
5)&CHR$(32)&CHR$(121)&CHR$(1
11)&CHR$(117)
330 D$=D$&CHR$(114)&CHR$(32)
&CHR$(99)&CHR$(104)&CHR$(111
)&CHR$(105)&CHR$(99)&CHR$(10
1)
340 GOSUB 470 :: D$=D$&CHR$(
157)&CHR$(200)&CHR$(3)&CHR$(
75)&CHR$(69)&CHR$(89)&CHR$(1
83)
350 D$=D$&CHR$(200)&CHR$(@)&
CHR$(51)&C$&CHR$(75)&C$&CHR$
(83)&CHR$(182)&CHR$(130)
360 D$=D$&CHR$(132)&CHR$(75)
&CHR$(191)&CHR$(200)&CHR$(2)
&CHR$(54)&CHR$(53)&CHR$(186)
&CHR$(75)&CHR$(192)
370 F=64+C :: GOSUB 500 :: D
$=D$&CHR$(176)&CHR$(201):: G
OSUB 480
380 F=24 :: GOSUB 490 :: D$=
D$&B$&CHR$(7)&CHR$(76)&CHR$(
111)&CHR$(97)&CHR$(100)&CHR$
(105)&CHR$(110)&CHR$(103)
390 D$=D$&CHR$(130)&CHR$(155
)&CHR$(75)&CHR$(194)&CHR$(20
0)&CHR$(2)&CHR$(54)&CHR$(52)
&CHR$(134)
400 D=B :: FOR B=B+10 TO B+1
0*C STEP 10 :: IF B>D+10 THE
N D$=D$&C$
410 D$=D$&CHR$(201):: GOSUB
480 :: NEXT B :: B=D
```

==========continued--->==

Loadmaker continued:

```
420 FOR D=@ TO C :: F=24 ::
G=9 :: GOSUB 490 :: D$=D$&B$
&CHR$(A(D))&A$(D)&CHR$(130)&
CHR$(169)&CHR$(199)
430 D$=D$&CHR$(A(D)+5)&CHR$(
68)&CHR$(83)&CHR$(75)&CHR$(4
9)&CHR$(46)&A$(D):: NEXT D
440 PRINT #@:D$&CHR$(0):CHR$
(255)&CHR$(255):: CLOSE #@
450 DISPLAY AT(16,@)BEEP:"Pr
ogram now on disk":"Now type
 in:":"  NEW":"  MERGE DSK1.
XXX":"  SAVE DSK1.LOAD"
460 DISPLAY AT(22,@):"And th
en you can try it by  typing
 RUN " :: STOP
470 B=B+10 :: PRINT #@:D$&CH
R$(0):: D$="" :: DISPLAY AT(
16,20):B
480 D$=D$&CHR$(INT(B/256))&C
HR$(B-256*INT(B/256)):: RETU
RN
490 GOSUB 470 :: D$=D$&CHR$(
162)&CHR$(240)&CHR$(183):: G
OSUB 500 :: D$=D$&C$ :: F=6
500 IF F<10 THEN D$=D$&CHR$(
200)&CHR$(@)&CHR$(48+F):: RE
TURN ELSE D$=D$&CHR$(200)&CH
R$(2)&CHR$(48+INT(F/10))&CHR
$(48+F-10*INT(F/10))
510 RETURN
520 END
```

Listing produced with Tony McGoverns CO-LIST program, available on disk from the disk library.

The above program reads the disk catalogue and writes a program which you can call LOAD, which allows you to select a program on the disk from a menu, and then the chosen file will load and run.

There are some limitations however!
a. In its present form it can only handle a disk which has 26 or fewer PROGRAM files.
b. It treats all PROGRAM files as though they were XB/TIB - that is, it will include on the menu any memory image machine code programs, graphic images, character sets and so on. This will not trouble you if you keep all your Basic/Extended Basic programs together on disks with no other form of memory image files!
c. However, it will NOT include any long Extended Basic programs which are stored on disk as I-V 254 files. To catch these as well you will need to do a little alteration to LINE 180.

The listing has been produced from a proven working program! If you experience difficulties, check back every line most carefully!
This program is remarkably quick in operation, and is a good example of a "program that writes programs".

USING CHARA1 file

in EXTENDED BASIC

Edward H Shaw has kindly sent me a routine to enable you to load one of the CHARA1 files into your own XB program, giving you access to a true lower case font- or a font of your own design (see last issue!).

Here is the routine as Edward sent it:

```
**************
PAB     DATA >0500,>02FA,>0000,>0800,>000B
        TEXT 'DSK1.CHARA1'
        EVEN
SAVR11  BSS  2
TAB     DATA >0000,>0000,>0000,>0000
        DEF  A
A       MOV  R11,@SAVR11
        LI   R0,>1000
        LI   R1,PAB
        LI   R2,>0015
        BLWP @>2024   VMBW
        LI   R0,>1009
        MOV  R0,@>8356
        BLWP @>2532   DSRLNK
        DATA >0008
        LI   R0,>03F8
        LI   R1,TAB
        LI   R2,8
        BLWP @>2024  vmbw
        CLR  @>837C
        MOV  @SAVR11,R11
        B    *R11
        END
```

You also need a DSRLNK for Extended Basic - there are several around.

Run the following Extended Basic program, which assumes that the object code produced from the above source is called LOWCAS, and that you have DSRLNK and CHARA1 on DSK1 as well:

```
100 CALL INIT
110 CALL LOAD("DSK1.DSRLNK","DSK1.LOWCAS")
120 CALL LINK("A")
130 PRINT "ABCDEabcde"
140 GOTO 140
```

[Ed also inserted CALL DELSPRITE(ALL)]

THIS PROGRAM WILL LOAD THE FILE CHARA1 FROM DSK1. Lower case will retain its loaded definition after the program ends.

****

You dont have a DSRLNK handy? FI! OK here is the same code- adjusted slightly! - with a DSRLNK inserted, and 'cos the DSRLNK needs it, you get a GPLLNK thrown in as well.
The DSRLNK and GPLLNK come from The Smart Programmer magazine.

Same code inclusive of DSRLNK follows----->

```
       DEF  FONT
VMBW   EQU  >2024
*****************
* NEXT BIT IS A GPLLNK ROUTINE
* REQUIRED BY FOLLOWING DSRLNK
* Craig Miller & D C Warren
* from THE SMART PROGRAMMER
*****************
GPLWS  EQU  >83E0
GR4    EQU  GPLWS+8
GR6    EQU  GPLWS+12
STKPNT EQU  >8373
LDGADD EQU  >60
XTAB27 EQU  >200E
GETSTK EQU  >166C
GPLLNK DATA GLNKWS   *R7
       DATA GLINK1   *R8
RTNAD  DATA XMLRTN   *R9
GXMLAD DATA >176C    *R10
       DATA >50      *R11
GLNKWS EQU  $->18
       BSS  >08     *R12>15
GLINK1 MOV  *R11,@GR4
       MOV  *R14+,@GR6
       MOV  @XTAB27,R12
       MOV  R9,@XTAB27
       LWPI GPLWS
       BL   *R4
       MOV  @GXMLAD,@>8302(R4)
       INCT @STKPNT
       B    @LDGADD
XMLRTN MOV  @GETSTK,R4
       BL   *R4
       LWPI GLNKWS
       MOV  R12,@XTAB27
       RTWP
*****************************
* NOW THE DSRLNK BIT
*****************************
PUTSTK EQU  >50
TYPE   EQU  >836D
NAMLEN EQU  >8356
VWA    EQU  >8C02
VRD    EQU  >8800
GR4LB  EQU  >83E9
GSTAT  EQU  >837C
DSRLNK DATA DSRWS,DLINK1
DSRWS  EQU  $
DR3LB  EQU  $+7
DLINK1 MOV  R12,R12
       JNE  DLINK3
************************
* NEXT BIT ONLY USED ON
* FIRST DSRLNK IN OUR
* PROGRAM - OUR PROGRAM ACTUALLY
* ONLY CALLS DSRLNK ONCE ANYWAY!
************************
       LWPI GPLWS
       MOV  @PUTSTK,R4
       BL   *R4
       LI   R4,>11
       MOVB R4,@>402(R13)
       JMP  DLINK2
       DATA 0
       DATA 0,0,0
DLINK2 MOVB @GR4LB,@>402(R13)
       MOV  @GETSTK,R5
       MOVB *R13,@DSRAD1
       INCT @DSRADD
       BL   *R5
       LWPI DSRWS
       LI   R12,>2000
**********************
DLINK3 INC  R14
       MOVB *R14+,@TYPE
       MOV  @NAMLEN,R3
       AI   R3,-8
       BLWP @GPLLNK
DSRADD BYTE >03
DSRAD1 BYTE >00
**********************
* ERROR CHECK
**********************
       MOVB @DR3LB,@VWA
       MOVB R3,@VWA
       SZCB R12,R15
       MOVB @VRD,R3
       SRL  R3,5
       MOVB R3,*R13
       JNE  SETEQ
       COC  @GSTAT,R12
       JNE  DSREND
SETEQ  SOCB R12,R15
DSREND RTWP
*************
* now to load CHARA1
*************
PAB    DATA >0500,>02FA,>0000,>0800,>000B
       TEXT 'DSK1.CHARA1'
       EVEN
SAVR11 BSS  2
TAB    DATA >0000,>0000,>0000,>0000
FONT   MOV  R11,@SAVR11
       LI   R0,>1000
       LI   R1,PAB
       LI   R2,>0015
       BLWP @VMBW
       LI   R0,>1009
       MOV  R0,@>8356
       BLWP @DSRLNK
       DATA >0008
       LI   R0,>03F8
       LI   R1,TAB
       LI   R2,8
       BLWP @VMBW
       CLR  @>837C
       MOV  @SAVR11,R11
       B    *R11
       END
```

To Next Column →

Now assemble. This listing produced directly from source code which has been assembled and which works!

TIPS FROM THE TIGERCUB

#43

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just $2.00 each, plus $1.50 per order for cassette or disk and PP&M. Cassette programs will not be available after my present stock of blanks is exhausted.
Descriptive catalogs, while they last, $1.00 which is deductable from your first order.

Tigercub Full Disk Collections, reduced to $10 postpaid. Each of these contains either 5 or 6 of my regular $2 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!
TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to $15.00 postpaid.
NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also $15 postpaid.

```
*****************************
* NUTS & BOLTS #3 is now    *
* ready, another full disk  *
* of 140 new merge-format   *
* utility subprograms, all  *
* compatible with the pre-  *
* vious. With 11 pages of   *
* documentation, $15 ppd.   *
*****************************
```

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to $10 ppd.
TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $10
TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25 through 32, $10 postpaid.
TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also $10 postpaid.

If you have as much trouble as I do, trying to get the strip labels lined up in the printer, you'll like this one -

```
100 DISPLAY AT(4,7)ERASE ALL
:"TIGERCUB LABELER": : : :"
This label maker will allow"
:"you to specify different":
"printer codes for each line
"
110 DISPLAY AT(11,1):"of a 5
-line label.": : :" You may
stop the program":"while lab
els are printing":"by pressi
ng any key, turn"
120 DISPLAY AT(17,1):"off th
e printer to adjust":"the la
bels, turn it back on,":"and
 press any key to con-":"tin
ue printing."
130 DISPLAY AT(23,1):"Printe
r designation?":"PIO" :: ACC
EPT AT(24,1)SIZE(-28)BEEP:PR
$ :: OPEN #1:PR$ :: P$,E$,DS
$,CEN$="Y" :: DW$,I$,SS$,U$=
"N" :: P=1
140 CALL CHAR(95,"FF")
150 FOR J=1 TO 5 :: CALL KEY
(3,K,S)
160 DISPLAY AT(2,1)ERASE ALL
:"Line #";J;" - PRINT? "&P$
:: CALL QUERY(2,20,P$):: IF
P$="N" THEN L$(J)="" :: GOTO
 360
170 IF J>1 THEN DISPLAY AT(4
,1):"Change codes? N" :: CAL
L QUERY(4,15,Q$):: IF Q$="N"
 THEN 300
180 DISPLAY AT(4,1):"Print p
itch? ";P:" (1)pica":" (2)el
ite":" (3)condensed" :: ACCE
PT AT(4,15)SIZE(-1)VALIDATE(
"123"):P
190 CI=(P=1)*-10+(P=2)*-12+(
P=3)*-17 :: L$(J)=CHR$(27)&"
B"&CHR$(P):: DISPLAY AT(5,1)
:"":"":""
200 DISPLAY AT(6,1):"Double
width? "&DW$ :: CALL QUERY(6
,15,DW$):: IF DW$="Y" THEN C
I=CI/2 :: L$(J)=L$(J)&CHR$(1
4)ELSE L$(J)=L$(J)&CHR$(20)
210 DISPLAY AT(8,1):"Italics
? "&I$ :: CALL QUERY(8,10,I$
):: IF I$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"4" ELSE L$(J)=L
$(J)&CHR$(27)&"5"
220 DISPLAY AT(10,1):"Supers
cript? "&SS$ :: CALL QUERY(1
0,14,SS$):: IF SS$="Y" THEN
L$(J)=L$(J)&CHR$(27)&CHR$(83
)&CHR$(0)ELSE L$(J)=L$(J)&CH
R$(27)&CHR$(84)
230 IF SS$="Y" THEN 250
240 DISPLAY AT(12,1):"Double
-strike? "&DS$ :: CALL QUERY
(12,16,DS$):: IF DS$="Y" THE
N L$(J)=L$(J)&CHR$(27)&"G" E
LSE L$(J)=L$(J)&CHR$(27)&"H"
250 IF P<>1 OR SS$="Y" THEN
270 :: DISPLAY AT(14,1):"Em
hasized? "&E$ :: CALL QUERY(
14,13,E$)
260 IF E$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"E" ELSE L$(J)=L
$(J)&CHR$(27)&"F"
270 DISPLAY AT(16,1):"Underl
ine? "&U$ :: CALL QUERY(16,1
2,U$)
280 IF U$="N" THEN L$(J)=L$(
J)&CHR$(27)&CHR$(45)&CHR$(0)
290 DISPLAY AT(18,1):"Center
 text? Y" :: CALL QUERY(18,1
4,CEN$)
300 DISPLAY AT(18,1):"Type l
ine";J;". Enter each":"scree
n line, enter again":"when d
one." :: DISPLAY AT(22,1):RP
T$("_",INT(CI*3.5)):: R=21 :
: CALL KEY(5,K,S)
310 ACCEPT AT(R,1):M$ :: IF
M$="" THEN 320 :: A$=A$&M$ :
: R=R+1 :: GOTO 310
320 IF LEN(A$)>INT(CI*3.5)TH
EN DISPLAY AT(16,1):"LINE TO
O LONG!" :: CALL SOUND(300,1
10,0,-4,0):: A$="" :: R=21 :
: GOTO 310
330 L=LEN(A$):: IF U$="Y" TH
EN A$=CHR$(27)&CHR$(45)&CHR$
(1)&A$&CHR$(27)&CHR$(45)&CHR
$(0)
340 IF CEN$="Y" THEN A$=RPT$
(" ",(INT(CI*3.5)-L)/2)&A$
350 L$(J)=L$(J)&A$ :: A$=""
360 NEXT J
370 DISPLAY AT(12,1)ERASE AL
L:"Print how many?" :: ACCEP
T AT(12,17):N
380 FOR J=1 TO N :: FOR K=1
TO 6 :: PRINT #1:L$(K):: NEX
T K
390 CALL KEY(0,K,S):: IF S=0
 THEN 410 ELSE CLOSE #1
400 CALL KEY(0,K1,S1):: IF S
1<1 THEN 400 ELSE OPEN #1:PR
$
410 NEXT J
420 DISPLAY AT(12,8)ERASE AL
L:"Another?" :: CALL QUERY(1
2,17,Q$):: IF Q$="N" THEN ST
OP ELSE 150
430 SUB QUERY(R,C,Q$):: ACCE
PT AT(R,C)SIZE(-1)VALIDATE("
YN")BEEP:Q$ :: SUBEND
```

More peculiarities of the TI computer -

```
90 CALL CLEAR :: PRINT TAB(7
);"SPRITE PUZZLE #1":"
  from Tigercub"
100 PRINT "A non-existent sp
rite can be":"created by CAL
L MOTION.": :"It apparently
starts in"
110 PRINT "dot-row 1, dot-co
lumn 1, and":"has color 1, b
ut its pattern":"is not that
 of any ASCII!"
120 !by Jim Peterson
130 FOR CH=0 TO 255 :: PRINT
 CHR$(CH);:: NEXT CH
135 PRINT "CALL MOTION(#1,5,
5):: CALL COLOR(#1,16):: CAL
L MAGNIFY(4)"
140 CALL MOTION(#1,5,5):: CA
LL COLOR(#1,16):: CALL MAGNI
FY(4)
150 GOTO 150
```

And another -

```
100 DISPLAY AT(3,5)ERASE ALL
:"SPRITE PUZZLE #2": :"
 from Tigercub"
110 DISPLAY AT(7,1):"Non-exi
stent sprites can be":"creat
ed by CALL COLOR.": :"Their
existence can be con-"
120 DISPLAY AT(11,1):"firmed
 by CALL COINC, but":"CALL P
OSITION reports that":"they
have no position!"
130 CALL COLOR(#1,16):: CALL
 COLOR(#2,16)
140 CALL COINC(#1,#2,1,X)::
DISPLAY AT(15,1):"COINC #1,#
2=";X :: CALL POSITION(#1,X,
Y)
150 CALL POSITION(#1,X,Y)::
DISPLAY AT(17,1):"POSITION #
1=";X;Y
160 CALL POSITION(#2,X,Y)::
DISPLAY AT(19,1):"POSITION #
2=";X;Y
170 IF FLAG=1 THEN 140 :: FL
AG=1
180 DISPLAY AT(21,1):"PRESS
ANY KEY"
190 CALL KEY(0,K,S):: IF S=0
 THEN DISPLAY AT(21,1):"pres
s any key" :: GOTO 180
200 DISPLAY AT(21,1):"Until
they're set in motion!"
210 CALL MOTION(#1,5,5):: CA
LL MOTION(#2,-5,-5):: GOTO 1
50
```

If you have the Terminal Emulator II, Speech Synthesizer, and a pre-schooler in the house, this will help him to grasp the idea of spelling as well as letter recognition and keyboard familiarization-

```
100 REM  PRE-SPELLER BY JIM
PETERSON
110 REM  TI BASIC WITH TERMI
NAL EMULATOR II AND SPEECH S
YNTHESIZER
120 CALL CLEAR
130 DIM M$(100),S$(100)
140 OPEN #1:"SPEECH",OUTPUT
150 PRINT "        PRE-SPELL
ER"::::::
160 PRINT "TYPE WORDS TO PRA
CTICE"::"TYPE 'END' WHEN FIN
ISHED"
170 X=X+1
180 INPUT M$(X)
190 IF M$(X)="END" THEN 380
200 PRINT #1:M$(X)
210 PRINT "PRONUNCIATION OK?
 (Y/N)"
220 CALL KEY(3,K,S)
230 IF S<1 THEN 220
240 IF K=78 THEN 280
250 IF K<>89 THEN 220
260 S$(X)=M$(X)
270 GOTO 170
280 PRINT "TRY SPELLING PHON
ETICALLY"
290 INPUT S$(X)
300 PRINT #1:S$(X)
310 PRINT "PRONUNCIATION OK?
 (Y/N)"
320 CALL KEY(3,K,S)
330 IF S<1 THEN 320
340 IF K=89 THEN 170
350 IF K<>78 THEN 320
360 PRINT "TRY AGAIN"
370 GOTO 290
380 CALL CLEAR
390 FOR J=1 TO X-1
400 PRINT #1:"CAN YOU SPELL
THIS?"
410 FOR A=1 TO LEN(M$(J))
420 CALL HCHAR(12,8+A,ASC(SE
G$(M$(J),A,1)))
430 NEXT A
440 FOR B=1 TO LEN(M$(J))
450 CALL KEY(3,K,S)
460 IF (S<1)+(K=32)THEN 450
470 IF K=ASC(SEG$(M$(J),B,1)
)THEN 500
480 GOSUB 640
490 GOTO 450
500 C$=C$&CHR$(K)
510 CALL HCHAR(14,8+B,K)
520 NEXT B
530 IF C$<>M$(J)THEN 640
540 PRINT #1:S$(J)
550 FOR D=1 TO 500
560 NEXT D
570 PRINT #1:"VEREE GOOD"
580 FOR D=1 TO 500
590 NEXT D
600 C$=""
610 CALL HCHAR(12,1,32,100)
620 NEXT J
630 GOTO 390
640 PRINT #1:"NO THAT IS NOT
 RIGHT"
650 PRINT #1:"TRY AGAIN"
660 RETURN
```

And, a simple little game that is a bit different than any I've seen -

```
100 !FORMATION by Jim Peters
on - use the S and D keys
110 CALL CLEAR :: CALL CHAR(
100,"381010FEFE383810103838F
EFE10103838"):: CALL SCREEN(
5):: CALL MAGNIFY(2):: RANDO
MIZE
120 V,W,P=0 :: FOR J=1 TO 7
:: CALL SPRITE(#J,100,7,1,25
0*RND+1,10,4):: FOR D=1 TO 1
00 :: NEXT D :: NEXT J :: CA
LL SPRITE(#11,101,16,160,128
)
130 CALL KEY(3,K,S):: W=W+1
:: IF W=150 THEN 170 ELSE IF
 W=300 THEN 180 ELSE IF K=68
 THEN V=V+2+(V>125)*2 ELSE I
F K=83 THEN V=V-2-(V<-125)*2
140 IF P=0 THEN CALL MOTION(
#11,0,V)ELSE IF P=1 THEN CAL
L MOTION(#11,0,V,#12,0,V)ELS
E CALL MOTION(#11,0,V,#12,0,
V,#13,0,V)
150 CALL COINC(ALL,A):: IF A
=0 THEN 130
160 CALL SOUND(1000,-4,0)::
H=MAX(H,W):: DISPLAY AT(23,1
):"SCORE";W:"HIGH SCORE";H :
: CALL DELSPRITE(ALL):: GOTO
 120
170 P=1 :: CALL POSITION(#11
,R,C):: CALL SPRITE(#12,101,
16,160,C-40-(C<40)*256):: GO
TO 140
180 P=2 :: CALL POSITION(#11
,R,C):: CALL SPRITE #13,101
16,160,C+40+(C>216)*256):: G
OTO 140
```

If you can't figure out where all the money goes, this may be an eye-opener -

```
100 DISPLAY ERASE ALL AT(3,5
):"THE COST OF CREDIT" ! by
Jim Peterson
110 S,T,X=0 :: DISPLAY AT(8,
1):"AMOUNT OF PURCHASE?" ::
ACCEPT AT(8,21):A :: B,T=A :
: DISPLAY AT(10,1):"CREDIT C
ARD INTEREST RATE?" :: ACCEP
T AT(11,1):R
120 DISPLAY AT(13,1):"SAVING
S ACCOUNT INT. RATE?" :: ACC
EPT AT(14,1):SR
130 X=X+1 :: I=B*R/100/12 ::
 B=B+I :: T=T+I :: P=B/10 ::
 B=B-P :: S=S+P+S*SR/100/12
:: IF S<A THEN 130
140 D$="$"&STR$(INT((T-A+S-A
+.5)*100)/100)
150 DISPLAY AT(17,1):"If you
 had saved the amount":"of y
our minimum 10% of the":"bal
ance credit card payment":"e
ach month for";X;"months,"
160 DISPLAY AT(21,1):"and us
ed it to pay cash, you":"wou
ld have saved ";D$ :: GOTO 1
10
```

And this is one of the handiest routines I've seen in a long time -

```
10 !TURNS ALL NUMERALS AND P
UNCTUATION WHITE! BY HARRY W
ILHELM IN TWIN TIers UG NEWS
LETTER
20 !TURN IT OFF BY CALL LOAD
(-31804,0)::TURN IT ON BY CA
LL LOAD(-31804,63)
100 CALL INIT
110 CALL LOAD(16128,2,224,38
,0,2,0,8,17,2,1,63,36,2,2,0,
3,4,32,32,36,2,224,131,192,3
,128)
120 CALL LOAD(16164,240,240,
240)
130 CALL LOAD(-31804,63)
```

Memory full

Jim Peterson

2021 clarification: In Issue 17 the report of the first General Meeting of the Group, no longer to be a one man operation, appointed Maurice Rymill as Cassette Librarian and Stephen Shaw as disk librarian.
In Issue 18 (this issue), with no explanation, Tim Anderson was presented as both cassette and disk librarian with a list of available software.
In Issue 19 the roles were clarified as Maurice Rymill, Cassette Librarian and Stephen Shaw as disk librarian as previously reported.
The article and software list from Tim was identified as an editorial slip up.
This group is the first "proper" user group- TIhome TIdings was run by Paul Dicks; TIHCUC was a short lived commercial operation; TI*MES began life operated by Clive Scally. TI Lines was run by Peter Brooks. TI-User and Parco were run by retailers. TI*MES has been the only "true" group run by users and initial problems were to be expected,

RAMBLES — Stop Press!

I have just received direct from BOB BOONE a list of goodies available in Canada from COMPUTER DOWNLOAD UNLIMITED. The list is VERY impressive - including the first ATARISOFT modules I've seen advertised for QUITE some time!
(I doubt if any are held in great quantity though). I have no experience of dealing with this company, but understand that they would LOVE to have an order from the U K.
They list 35 educational modules (Tex Comp list 41).
They list 17 games modules (Tex Comp list 21).
THEY LIST 42 BOOKS!

Goodies include the following (Prices in C$- about C$2 to the pound - and exclude post and packing:
Orphan Survivors Handbook $30, Orphans Chronicles $15, Best of 99er $15,
Smart Programming Guide for Sprites $12, Assembly Language Primer $12,
Beginning Assembly Language on TI $12, TE2 Protocol Manual $12,
Scott Adams Adventure Hints $5, and modules...
Scott Foresman's Number Bowling,Frog Jump,Pyramid Puzzlers etc $15
Centipede, Defender, Pacman, $15 Personal Record Keeping $15
AND MORE...
WICO TRACKBALL (UK power supply required-easily available) $35
Speech Synthesiser $50, or with 32k ram inside (???) $100
Spad XIII on disk $35, Navarone Widget $40

25 Ottawa St Arnprior Ontario CANADA K7S 1W*

REQUEST: If you have purchased goods from the States/Canada in the last 6 months, please write and tell me about the service you received. In the next issue of TI*MES I should like to put our joint experience together with a current list of what is available where into writing for the service of all.
Please write to Stephen Shaw, 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

List number 9785

TI99/4a

U.K. TI USERS SOFTWARE LIBRARY.

EDUCATION.

E1~ALPHABET(SP.J.EX)
Now you will test your speed and knowledge of your A-Z.

E2~HANG THE MAN (EX)
Play against the computer or with a friend. Customise for your own words.

E3~WORLD OF WORDS.(EX)
Makes full use of sprites sounds and colours. Your child will enjoy learning to read the words with pictures.

E4~FACES(B).
Find A to Z on the keyboard of your TI99 your child will be rewarded with happy faces.

E5~TOUCHTYPING.(B)
You can now improve your typing skills. Uses graphics of the QWERTY keyboard.

E6~STELLA.(EX or B).
The main constellations explained.

E7~USA WESTERN STATES. (B or EXT).
A graphic quiz to test your knowledge of USA States.

E8~SHAPE DRAW.(B).
Full colour graphics using pre-defined shapes to make pictures on a grid.

E9~COUNTING(B).
Early learning counting fun.

E10~COMET SUMS(EX)
An educational way of shooting space invaders if you know your x/-/+/-.

E11~EXTENDED BASIC TUTOR(EX)
This program gives on screen examples of some Ext' Basic commands.

E12~TEST YOUR I.Q.(EX)
If you like general knowledge questions then test your IQ with this program.

E13~MATHS CALCULATOR(B)
Calculates averages and triangles in Trig. Useful and simple to use.

E14~MORSE CODE TUTOR(B or EX+SP)
Useful program to help you learn More Morse.

E15~MYSTERY SPELLING(EX)
Tuneful colourful word mystery game suitable ages 5 to 12.

E16~JAWS(B)
Colourful graphic hangman type game. One to get your teeth into.

E17~BOGGLE CHALLENGE(B) Type of Scrabble where you make up words from a large grid.Up to 4 players.

E18~MAGIC CRAYON(DSK+E/A)
S=Save R=Recall C=Colour T=Terminate are the keys when you load and run this program using E/A option 3

E19~TI 99ER CRAYON(B)
Your works of art can be saved onto CS1. excellent program for for all ages.

E20~COMPLETE TYPING COURSE (EX+DSK+32)
Now you CAN really get to grips with typing, this program must be the best tutor around for you to learn and pratice using ALL your fingers. (Counts as THREE programs).

E21~MORSE CODE TUTOR(EX)
This one will teach you morse of course.

E22~MORSE CODE TUTOR EXTRA
(EX+32K+DSK+SP)
This program has more options that will ensure progressive learning of the MORSE CODE for examinations.

E23~ALPHA DELIGHT (EX+32K+DSK+SP)
A program for the young child that will delight. Very clever A TO Z teaching with good use of Graphics. (COUNTS as TWO programs).

E24~SHUTTLE DATA BASE(EX,CS1/DSK)
This program will keep a record (on Disk) of each Flight the US SHUTTLE makes into space, apart from giving you some facts of the early design. Good graphics.

2021 note: Although not stated here, this list if of the group's CASSETTE SOFTWARE LIBRARY The DISK LIBRARY was separate, longer and supplied on disk.

ULTILITIES.

U1~AUTO LOAD(DSK,EX).
Loads EXT BASIC programs automatically see TI*MES review No 8.
U2~GRAPH(B).
BASIC graph plotter.
U3~GRAPH(EX).
For more complex graphs
U4~LIFESPAN(B).
Calculate your future years.
U5~SCREEN MAP(EX,SP).
Find the exact location of a pixal on the screen.
U6~FILER 99ER(EX)
Probably the best filing utility around.
U7~CALENDAR(EX).
Will tell you the exact day of dates from 1750 to 2000.
U7b~CALENDAR(B)...As above but runs in TI BASIC which you can dump on a "TP"Printer if required.
U8~WORDPRO(B,PRINTER/CS1)
Type in and print out letters etc. Uses EDIT,BACKSPACE and SEARCH facilities. This BASIC program will allow you to save and load to CS1 for a friend to print out.
U9~VIDEO CREDITS(EX)
Now give your video recordings titles with choice of scripts. (only useful if you can already use a video with the TI99/4a).
U10~PRINTER SET(PRINTER).
A program to set up a printer buffer (STAR 10X) and slash the zerOs.
U11~LIFE INSURANCE(EX).
Will calculate insurance
U12~CAR INSURANCE(EX).
Will help you on insurance.
U13~HOUSE BUDGET.(EX CS1).
Got a budget problem? This will help you forcast and show your expenses.
U14~CS1 AND CS2 FILE.(B+CS1/2).
Create a large data file using two cassettes.
U15~PIXELDRAW(B).
Use the key board to draw on screen a pixel a time.
U16~CHAR CONSTRUCTOR(EX or B).
Gives you the Hex of characters you construct.

U17~PROGRAM COMPRESSOR (EX+DSK).
Converts your Basic programs into multi-staement extended basic using the merge function, very clever and useful memory saver. (only basic programs that presently run in extended)

U18~APPOINTMENT CALENDAR.
(EX+DISK+32K+PRINTER).
Dumps out on a printer monthly calendar which you record special events etc.
U19~WORDPRO+. (EX+32K+DISK+PRINTER)
A more sophisticated word processing program for business use perhaps.
U20~PRINTOUT TUTORIAL. (EXT+PRINTER).
This one will teach you a trick or two about the printer.
U21~PRICE GENERATOR (EX)
Converts VAT and calculates discounts, clever and useful.
U22~SCREENDUMP(EX,PRINTER,DSK)
A very fast screen dump program for the Star Gemini 10x Printer. If you have the knowledge it can be used for other printers.
U23~MEAL PLANNER (EX,32K,DSK,PRINTER).
The housewife dream, once you have input your favourite meals, you can plan ahead. Prints out an ingredients list too.
U24~HOUSEHOLD FINANCE (EX,PRINTER)
Analysis of your house expense is printed out.
U25~TI-WRITER UTILITY (EX,PRINTER)
Prints out two 28 column format from DIS/VAR 80 Files.
U26~DISASSEMBLER(M/M or E/A or EX+DSK+Printer)
Translates binary machine code in the computers' memory back to assembly code (TMS9900 mnemonics TI99/4a system names) This was donated by Funnelweb Farm Australia.
U27~PERSONNEL REPORT FILE
(DSK+EX+PRINTER)
Almost a complete file can be kept of your personnel, a good program which can be customised for your own use
U28~TRIPLE DISC CATALOGUE
(EX,DSK,PRINTER)
Prints 3 disc catalogues side by side.
U29~ELECTRONIC TYPEWRITER
(EX+PRINTER+32K)
A more than useful utility which will be handy when you just want to type out a letter from screen to printer. Kindly donated by Ed York of CINDAY

2021 Note: Not stated but this list was of software available from the Group's CASSETTE librarian. Software on disk was in a separate library at lower supply cost. There was some confusion in these early days of the "true" user group.

U30~GOTHIC WRITER
(DSK+EXT+PRINTER+32K)
Downloads to your printer Gothic script, full use of Qwerty. Easy to use.
U31~CENTRAL HEATING CALC(EX)
Room by room heating needs are calculated for you. Now you know how much the bill should work out.
U32~DIS/VAR 80 UTILITY
(EX+PRINTER+DSK)
Allows you to examine Dis/var 80 files stored on disk and dumped to your printer.
U33~TEXTWRITER(EX+DSK+PRINTER)
12 functions will assist you in use of this word processing utility. Handy if you do not have TI-Writer.
U34~PERSONAL CALENDAR
(DSK+EX+PRINTER).
Prints out the year using your own message.
U35~BIORHYTHM(B)
Clever little program to check your lifestyle.~U35X~AS ABOVE but for use with PRINTER+EX
U36~EXT CLOCK(EX).
Displays a time clock on screen even when loading other programs.
U37~PRINT A MAZE(EX+PRINTER)
Will create and printout a Maze with choice of size.
U38~PLAYGIRLS (EX+DSK+PRINTER)
Counts as three programs. This is an ADULTS ONLY selection of Calendar girls printed out on hardcopy.
U39~PRINTER EXTRA (DSK+EX+PRINTER)
Counts as six programs a disk chock full of interesting and useful printout programs for you to enjoy.
U40~TELEPHONE COST(EX)
Will give telephone costs per unit.
U41~WILL PRINTOUT(EX+PRINTER)
Your Will and testiment can be printed out as a document. Saves fuss.
U42~DISK SLEEVE(EX+DSK+PRINTER)
Not only design a Sleeve for disks but catalogues them on the cover too!
U43~COMPLETE SPRITE DESIGN
(EX+DSK+PRINTER)
Counts as TWO programs. You can now complete your own library with this program. Facilities to design and use up to twelve sprites. we include a selection which you can use in your own programs. Written by ED YORK.

DEMOS.

D1~SPRITE DEMO (EX).
Includes sprites in 3D.
D2~PATTERN BOX (B).
Generates random patterns and gives you a HEX code.
D3~TE.2 DEMO(TE2 + RS232)
Random demo for speech.
D4~ADVERT AID(B)
Use for clubs etc to display annoucements or adverts.
D5~DOCTOR DOCTOR(B or EX or TE2+PRINTER+SP).
Now you can talk to your computer Doctor about personal problems, a good demo which will run in Basic.
D6~COLOUR MUSIC BOX (B or EX).
As published in TI*MES issue No 7 continuous music random colour patterns.
D7+~EXTENDED BASIC 32K DEMO
(EX+DSK+32K)
Counts as three programs. Bet you did not really know the performance of your computer until you see this. excellent graphics.

F.R.K. MODULE.

P1~BUSINESS GAME.(PRK)
Now see a good example of DISPLAY AT with this game of skill, adults only.
P2~BOAT DESIGN (PRK).
Not a game but a utility for boat design, we think its clever.

MUSIC PROGRAMS

M1. Blowing Bubbles. (EX).
Watch the bubbles and sing the song graphics and music forever.
M2. Dixieland. (B or EX).
A well-known number from the USA, useful as a subprogram.
M3. Carmoon. (EX)
Watch the moon at night and the sun rise as the song plays.
M4. Camptown Races. (B or EX).
You all know this one (music only).
M5. Music Creator. (B or EX).
You can make up your own tunes in Basic or Extended Basic. You should have knowledge of CALL SOUND.
M6. TI Organ. (EX). ***
Enables you to create your own tunes.
M7. Venetian Boat Song No. 1. (EX). ***
Sprites and graphics, lovely music.
M8. Ode to Puppy-Town. (EX). ***
A trio of puppies run in tune past a saloon.
M9. Sound Constructor. (EX).
Helps you to create special sounds or music. This program is well written.
M10, Music, Music. (B or EX).
Delightful selection of six well-known tunes.
M11. Bananas. (EX).
O yes, we do have bananas. Lots of them.
M12. Can't Help. (EX).
Elvis?
M13. Lord's Prayer. (EX).
Impress the vicar.
M14. Wings of a Dove. (EX).
M15. Green, Green. (EX)
Golden oldie from the far side of the hill.
M16~MANDY(EX)
For Mary Banilow fans
M17~SUNFLOWER(EX+DSK)
Scott Joplin rag.
M18~HOUSTON<TEXAS(EX)
Moving graphics.
M19~SUNGLASSES(EX)
Shady tune.
M20~YESTERDAY(EX)***
When did I last hear this Beatles classic.
M21~GHOSTBUSTERS(EX)
Excellent rendering of the No.1 hit for your TI plus G/B motif on screen.

M22~SPACE SHUTTLE ADVENTURE (DSK+32K+EX)
Counts as THREE programs. Load disk sit back and enjoy the Music and excellent moving graphics on a journey into space.
M23~TRAINLOVER(DSK+EX)
Award winning music and graphic program written by Stephen Foster of H.U.G.
M24~MUSIC DISK(DSK+EX)
Counts as Six programs but you will receive a disk with over twenty tunes and lovely hymns well written by Bill Knecht of H.U.G.
M25~MIDNIGHT COWBOY(EX)
Lovely use of the TI99/4a sound.
M26~BEETHOVEN CLASSIC(EX)
Music and excellent graphics make this Variations on a Theme a proud presentation for anyone to show off the TI99/4a. M27~HAYDN'S SONATA 2 (EX)
A classic for your collection. Without gaphics.
M28~BACH (EX)
Lets you invent with this Invention in F. No Graphics.
M29~FAME(EX)
The Famous TV Fame theme.
M30~BOOGIE SPECTACULAR(EX+DSK)***
Excellent selection of programs for you to enjoy, well written with good graphics (Space ships, a TI computer, two dancing robots, and a touch of the wild west. NOTE counts as THREE programs).

*** Indicates Sam Moore compositions, donated by Texas Instruments to the Users-Group Library.
Note All music is NOT for public performance.

GAMES PROGRAMS.

G1~LOONEY QUEST(B).
Silly adventure if you are game for a laugh.
G2~CAVE MAZE(B).
If you could not type in this adventure game feature in TI*MES, here we have done it for you.
G3~FROG(EX).
You may like this splendid version to the old favorite.
G4~PHOTON ATTACK.(B).
Space invader type game.
G5~ALIEN ATTACK(B).
Use the keyboard to stop the Aliens attack from a grid.
G6~COMBAT(B).
A game of Skill for two players, you fight it out on the battle field with Tanks.
G7~WORDSEARCH (B+PRINTER).
Will create and printout your own Wordsearch games.
G8~ISOLA (B or EX/SP).
A board game. An excellent TI Users-group program.
G9~OTIHELLO (B +/or SP).
Another board game for two players. Speech is optional.
G10~YAHTIZEE(B).
This is must be the best game first produced for the TI99. Makes good use of sound and graphics. A dice game for up to four players.
G11~KNIGHTS AND DRAGONS(B).
Can you become a Knight and fight the Dragons. Super use of graphics when you hear the clash of swords they sound very real.
G12~LEAPER(EX,J).
This is a Hunchback type of game which has been adapted for the TI99/4a with six levels of play. This is probably better than the average extended basic games on the market.
G13~FIREFLY SUBMARINE(EX,J) Can you hit the target?.
G14~AIRSTRIKE CHALLENGE (EX,J)
As a fighter pilot you shoot down the enemy.
G15~RUNWAY(EX,J).
You are the Capitain of a modern aircraft can you land safely. Watch out for wind shifts, a steady hand is required.
G16~TILES(EX).
This gives you an idea how smooth the sprites can be used when moving them around on a board either numbers or letters to choose.
G17~CAMEL(B)
A crazy and fun adventure game.
G18~HAMURABI(B).
Use your skill to prevent starvation of a Nation.
G19~CRASH (EX).
A simple game of hit and crash.
G20~MM RACER(M/M).
Now we did not forget the minimem you have to be fast to track this.
G21~DESERT CAR(M/M).
Published in TI*MES No6 Syd Michel has done the hard work for you, see what machine code can do super graphics.
G22~ODDS ON (EX or EX+SP+PRINTER).
Place your bets the race is on today and every day. Printer optionial to record bets and tote.
G23~HI-LO (B).
A graphic card game in which you try your luck of the draw.
G24~MAN TACO (EXT).
Eat your way round like a munchman but watch out for the surprises, good use of graphics and sprites.
G25~SUBMARINE ATTACK(EXT,J)
Captain your own Sub' and sink many ships. Watch out for the depth charges.
G26~TV JACKPOT(EX).
Excellent graphics, proberly the best TI Fruit machine game produced.
G27~BALLOONS(EX+DSK)
Manoeuver a hot air balloon with joysticks. Nice graphics from LA99ers
G27S~CIRCUS BALLOONS(EX)
Help the clown bounce on a trampoline to burst the balloons.
G28~DUCK(EX)
Make it to the top avoiding ducks and frisbees
G29~IN A MAZE(EX)
Find your way out of a 3D maze,complete with map.

G30~PONTOON(B)
Stick or twist?
G31~BINGO NIGHT(EX,OptSP)
Excellent program donated by Texas Instruments Inc. print your bingo cards via printer if wished, call out the numbers if you have a Sp/synth.
G32~GARDEN MAZE(B)
Move round the garden killing off the pests.
G33~AIRTRAFFIC CONTROLLER(EX)
Control the aircraft takeoff and land at a busy airport.
G34~CONNECT FOUR(EX)
For two players, this is a classic game of the same name.
G35~JOY SKETCH(MM+JS)
Not really a game.but a super sketch program published in TI*MES Number 9.

MODEM UTILITIES

B1~MODEM(DSK+MODEM+MM/EA)
Counts as two programs see review in TI*MES No 8.
B2~BULLETIN BOARD (MODEM+DSK+232+EX)
Very useful for BBS fans.
B3~PRESTEL EXCLUSIVE (MODEM +EX+DSK)
This program is exclusive in the U.K. to TI99/4a Exchange members. It is supplied with instructions. (Counts as THREE programs).

BASICODE in M/M

BASICODE~LOAD/READ/WRITE(MM+CS1/DSK)
AT LAST A PROGRAM THAT WILL LOAD THE BBC BROADCASTS INTO THE TI99/4A Exclusive in the U.K. to TI99/4a Exchange members. This program requires the use of MINI MEMORY Module, CS1 and/or DISK. Please write to the BBC for broadcast information. (You must have knowledge of TI BASIC to enable conversion of programs broadcast.)

T.E.2 MODULE

T1~SPEAK NUMBERS(TE2+RS232+ SP).
Enter a number and the computer will tell you in words and print. eg: what is 1000000000000123
T2~UNIVERSE TRANSLATOR(TE2+SP)
Convert spoken English into spoken Alien? It prints out on your printer too!
T3~TE2 TUTOR(TE2+SP)
This will assist in making good use of the module for speech and even make your TI computer sing!
T4~TE2 MUSICMAKER(TE2+SP)
Make music with this useful program on your TI99/4a. Very helpful.

KEY B=BASIC EX=EXTENDED BASIC MODULE
J=JOYSTICK SP=SPEECH MODULE
PRK=Personal Record Keeping MODULE
DSK=DISK DRIVE E/A=EDITOR ASSEMBLER
M/M=MINIMEM RS232=INTERFACE CARD
32K=EXPANSION
~NOTE Programs that indicate PRINTER run on PIO but can be adapted to suit your own requirements.

Cassette

Conditions of the Group Library

To receive a free program on cassette/~~disk~~ we ask members to submit their own programs for the Users-group software library. Programs must be original and NOT copy from commercial software houses or third party copyright programs (unless written consent is obtained to use such programs). Instructions on use should always be included.
When a member submits a program to TI99/4a Group software library it will remain the property of the original author who either coded or translated the program. TI99/4a U.K. TI Users-group does not claim any proprietary rights to any program listed in the software exchange library and cannot be held responsible for their contents.
All programs will be made available exclusively to members. This group will have the right to use such software in exchange of other users programs. IT IS MOST IMPORTANT that members using the software exchange library comply with the COPYRIGHT LAW. Anyone breaking this rule will face legal action being taken. None of the programs can be sold commercially. The programs are yours to keep, for your own use only.
If you have a program to exchange send in your Tape ~~or disk~~ and we will return a free program of your choice. IMPORTANT please enclose Envelope and Stamps for return.
MEMBERS who do not have a program to donate can order for a handling fee of £4.50 any THREE programs. Additional programs are charged at £1.75 each or £4.50 if orders are in multiples of THREE. This fee includes cost of tape handling, postage and packing (~~DISK ADD 75p~~). It should be noted that overseas orders will be sent surface mail.
U.K. FUNDS ONLY CHEQUES crossed made payable to TI Users U.K.
** NOTE ** We cannot accept orders from NON MEMBERS

MEMBERS APPLICATION FOR SOFTWARE.

I _____(PRINT NAME) agree as a member with the rules of the Group library relating to Copyright.

signed_____date_____ | Office use only file _code _date_ _fee___ |

I wish to Exchange/order:- (1)_____(2)_____(3)_____

PROGRAMS ON * CASSETTE __* DISK__ (add 75P)

ADDITIONAL PROGRAMS PLEASE LIST BELOW

This is your address label please print clearly.

NAME_____

ADDRESS_____

_____

_____

_____POSTCODE_____

Please allow 14 days to process. Do not forget to send your program or cheque which **must be with this signed application form** to:-

T.I.99/4a USERS GROUP (U.K.)
GROUP LIBRARY,
231, BOURNVILLE LANE,
BOURNVILLE, BIRMINGHAM B30 1RA

2021 clarification:
This list was printed in error and should have only been the CASSETTE LIBRARY list.
The disk library was operated separately- supply your own disk and cost was just £2 for several (maybe 18) files on a disk.

TI*MES QUESTIONNAIRE

PLEASE complete this Questionnaire so that we know what YOU have. The findings will be published in a future edition of TI*MES, Editor permitting. Please remember that the more Questionnaires returned, including YOURS, the more accurate the findings. PLEASE PUT A TICK TO THE RIGHT OF TI OR OTHER.

COMPUTER; TI 99/4 ____ 99/4A ____

PERIPHERALS;

JOYSTICKS	TI ____	OTHER ____	MAKE? _____		
CASSETTE RECORDER	TI ____	OTHER ____	MAKE? _____		
SPEECH SYNTHESISER	TI ____				
STANDALONE 32K	TI ____	OTHER ____	MAKE? _____		
STANDALONE DISK CONTROLLER	TI ____	OTHER ____	MAKE? _____		
STANDALONE RS232	TI ____	OTHER ____	MAKE? _____		
PERIPHERAL EXPANSION BOX	TI ____				
32K CARD	TI ____	OTHER ____	MAKE? _____		
DISK CONTROLLER CARD	TI ____	OTHER ____	MAKE? _____		
RS232/PARALLEL CARD	TI ____	OTHER ____	MAKE? _____		
P CARD	TI ____				

DISK DRIVES;

40 TRACK SINGLE SIDED	TI ____	OTHER ____	MAKE? _____		
40 TRACK DOUBLE SIDED		OTHER ____	MAKE? _____		
40/80 SWITCHABLE S/SIDED		OTHER ____	MAKE? _____		
40/80 SWITCHABLE D/SIDED		OTHER ____	MAKE? _____		

PRINTERS;

THERMAL	TI ____	OTHER ____	MAKE? _____		
DOT MATRIX SERIAL		OTHER ____	MAKE? _____		
DOT MATRIX PARALLEL		OTHER ____	MAKE? _____		
DOT MATRIX DUAL SER./PAR.	TI ____	OTHER ____	MAKE? _____		
DAISYWHEEL SERIAL		OTHER ____	MAKE? _____		
DAISYWHEEL PARALLEL		OTHER ____	MAKE? _____		
DAISYWHEEL DUAL SER./PAR.		OTHER ____	MAKE? _____		

MODEM; TI ____ OTHER ____ MAKE? _____

TV/MONITOR? _____ B/W/COLOUR? _____ SOCKETS UHF ____ RGB ____ COMP/VIDEO ____

KEY SOFTWARE;

DISK MANAGER I	TI ____				
DISK MANAGER II	TI ____				
EDITOR/ASSEMBLER	TI ____				
EXTENDED BASIC	TI ____	OTHER ____	MAKE? _____		
FORTH	TI ____	OTHER ____	MAKE? _____		
LOGO	TI ____				
LOGO II	TI ____				
MINI-MEMORY	TI ____				
TERMINAL EMULATOR II	TI ____				
TI MULTIPLAN	TI ____				
TI WRITER	TI ____				

MISCELLANEOUS ADD-ONS; (Please give details: e.g. HEX-BUS INTERFACE etc.).

NAME: _____ TELEPHONE NO: _____

ADDRESS: _____

LOGO (Turtle) TUTORIAL

BY

MIKE SLATTERY

Welcome to the CHRISTMAS issue of LOGO CORNER. I thought this month I would do something with a Christmas flavour and the program below is the result.

This program draws SANTA and his sleigh and moves them to a chimney where Santa gets off the sleigh. Santa then goes down the chimney to a room with a Christmas tree with blinking lights. At the same time a simple version of JINGLE BELLS is being played.

This program is a good demonstration of sprites and tiles, and contains a very useful procedure for clearing the screen of both sprites and tiles simply by pressing any key on the keyboard.

By changing STOP to MOVIE in the CLEAR procedure, the program can be made to recycle.

You will notice all the lights on the tree are the same color. Can anyone write a procedure which changes the colors of individual lights only?

I will return to the normal column in FEBRUARY, and will discuss this program more fully later in the year.

Bye for now and a HAPPY CHRISTMAS to everyone.

* REQUIRES LOGO 2 MODULE *

TI Sydney Users' Group

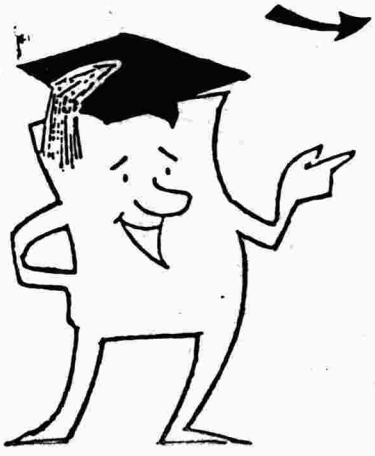
NOTE: LOGO 2 HAS THE FOLLOWING ADDITIONAL WORDS:
BIG SIZE SMALL LENGTH
REVERSE ROTATE FALSE TRUE
and Music words:
CHROMATIC DRUM LEGATO LOOPMUSIC
MAJOR MUSIC NOTE PLAYNOTE
PLAYMUSIC REST SETTEMPO
SETVOICE SETVOLUME STACCATO

LOGO 2 also has additional edit and debug commands and more memory for your p

```
TO MOVIE
CS TELL :ALL SC 0
JINGLE
LOOPMUSIC
CHIMNEY
SLEIGH
WAIT 240
SANTA
TELL :ALL CARRY 17 SC 0 SXY 98 98
COLTREE
LIGHTS
P :ALL
END

TO JINGLE
CM
J1 J2 REST 1
J3 REST 1
J4 REST 1
J1 J6 J7 J8 REST 2
J9 J10 REST 1
J11 J12 REST 1
J9 J10 REST 1
J11 J16 REST 2
END

TO J1
MUSIC [0 9 7 5 0] [2 2 2 2 4]
END

TO J2
MUSIC [0 0 0 9 7 5 2] [1 1 2 2 2
 2 4]
END

TO J3
MUSIC [2 11 9 7 4] [2 2 2 2 4]
END

TO J4
MUSIC [12 12 11 7 9] [2 2 2 2 4]
END

TO J6
MUSIC [0 9 7 5 2] [2 2 2 2 6]
END

TO J7
MUSIC [2 2 11 9 7 12 12 12] [2 2
 2 2 2 2 3]
END

TO J8
MUSIC [12 14 12 10 7 5] [1 2 2 2
 2 4]
END

TO J9
MUSIC [9 9 9 9 9 9] [2 2 4 2 2 4]
END

TO J10
MUSIC [9 12 5 7 9] [2 2 3 1 4]
END

TO J11
MUSIC [11 11 11 11 11 9 9] [2 2 3
 1 2 2 2]
END

TO J12
MUSIC [9 9 9 5 5 9 7] [1 1 2 2 2
 2 3]
END

TO J16
MUSIC [9 9 12 12 10 7 5] [1 1 2 2
 2 2 3]
END

TO CM
SETVOICE 0
SETVOICE 1
END
```

MORE LOGO...

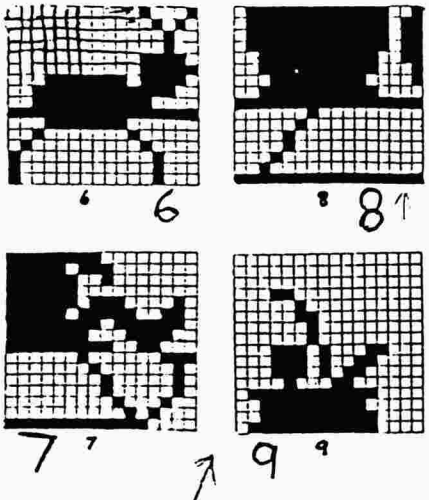
```
TO CAR :C
PT 183 :C 23
IF :C > 31 STOP
CAR 1+ :C
END

TO SANTA
TELL 3 SC 6 CARRY 14
TELL 6 SC 0
TELL 25 CARRY 21 SXY 18 -22
TELL 24 CARRY 22 SXY 18 -6
TELL 23 CARRY 20 SXY  2 -11
TELL [23 24 25] SC 6
WAIT 240
SV 0 -2
TELL 24
POS1
END

TO POS1
IF YCOR < -50 TELL [23 24 25] SC 0
 SV 0 0 STOP
POS1
END

TO COLTREE
TELL TILE 183 SC 8
TELL TILE 231 SC 2
TELL TILE 239 SC 2
TELL TILE 247 SC 2
TELL TILE 255 SC 4
END

TO LIGHTS
TELL [1 2 3 4 5 6 7 8 9] SC 4
 CARRY 25
TELL 1 SXY 8 1
TELL 2 SXY -15 -6
TELL 3 SXY -7 33
TELL 4 SXY -16 8
TELL 5 SXY -8 -23
TELL 6 SXY -31 -31
TELL 7 SXY 16 -31
TELL 8 SXY 9 -46
TELL 9 SXY -16 -46
TELL [10 11 12 13 14 15 16 17 18
 19 20 21] SC 4 CARRY 24
TELL 10 SXY -23 58
TELL 11 SXY 8 58
TELL 12 SXY -32 42
TELL 13 SXY 17 42
TELL 14 SXY -47 18
TELL 15 SXY 32 18
TELL 16 SXY -56 -6
TELL 17 SXY 40 -6
TELL 18 SXY -64 -30
TELL 19 SXY 49 -30
TELL 20 SXY -71 -54
TELL 21 SXY 57 -54
END
```



nb: for LOGO 2 only

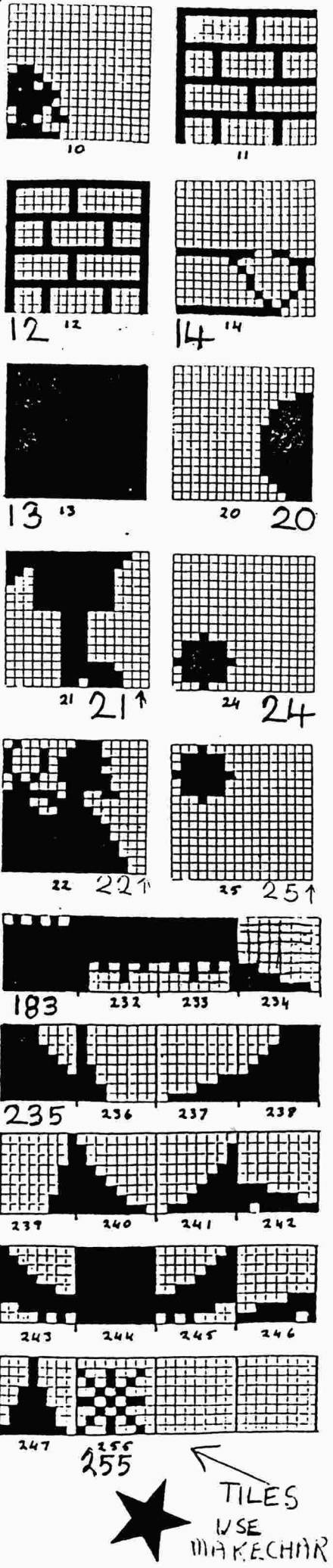
```
TO CHIMNEY
TELL  7 CARRY 11 SXY  6 -38
TELL  8 CARRY 12 SXY 20 -38
TELL  9 CARRY 13 SXY  6 -38
TELL 10 CARRY 13 SXY 20 -38
TELL 11 CARRY 11 SXY  6 -54
TELL 12 CARRY 12 SXY 20 -54
TELL 13 CARRY 13 SXY  6 -54
TELL 14 CARRY 13 SXY 20 -54
TELL 15 CARRY 11 SXY  6 -70
TELL 16 CARRY 12 SXY 20 -70
TELL 17 CARRY 13 SXY  6 -70
TELL 18 CARRY 13 SXY 20 -70
TELL 19 CARRY 11 SXY  6 -86
TELL 20 CARRY 12 SXY 20 -86
TELL 21 CARRY 13 SXY  6 -86
TELL 22 CARRY 13 SXY 20 -86
TELL [7 8 11 12 15 16 19 20] SC 1
TELL [9 10 13 14 17 18 21 22] SC 8
END

TO SLEIGH
TELL 1 CARRY  6 SXY -43 78
TELL 2 CARRY  6 SXY -59 78
TELL 3 CARRY  7 SXY -75 78
TELL 4 CARRY  8 SXY -91 78
TELL 5 CARRY  9 SXY -91 94
TELL 6 CARRY 10 SXY -75 94
TELL [1 2] SC 15
TELL [3 4 5 6] SC 8
TREE
TELL [1 2 3 4 5 6] SV 3 -2
TELL 1
POS
END

TO POS
IF XCOR > 55 TELL [1 2 3 4 5 6]
 SV 0 0 STOP
POS
END

TO TREE
CS
TELL TILE 183 SC 0
TELL TILE 231 SC 0
TELL TILE 239 SC 0
TELL TILE 247 SC 0
TELL TILE 255 SC 0
ADD1 ADD2 ADD3
PT 255 15  3  PT 247 15  4
PT 246 13  5  PT 245 14  5
PT 244 15  5  PT 243 16  5
PT 242 17  5  PT 241 14  6
PT 244 15  6  PT 240 16  6
PT 246 12  7  PT 245 13  7
PT 244 14  7  PT 244 15  7
PT 244 16  7  PT 243 17  7
PT 242 18  7  PT 239 13  8
PT 238 13  9  PT 237 12  9
PT 236 17  8  PT 235 17  9
PT 234 18  9  PT 244 14  8
PT 244 15  8  PT 244 16  8
PT 244 14  9  PT 244 15  9
PT 244 16  9  PT 245 11 10
PT 246 10 10  PT 233 12 10
PT 232 18 10  PT 243 19 10
PT 242 20 10  PT 239 12 11
PT 238 12 12  PT 237 11 12
PT 236 18 11  PT 235 18 12
PT 234 19 12  PT 245 10 13
PT 246  9 13  PT 243 20 13
PT 242 21 13  PT 244 15 20
PT 244 15 21  PT 244 15 22
PT 233 11 13  PT 245 10 13
PT 246  9 13  PT 232 19 13
PT 243 20 13  PT 242 21 13
PT 239 11 14  PT 238 11 15
PT 237 10 15  PT 236 19 14
PT 235 19 15  PT 234 20 15
PT 233 10 16  PT 245  9 16
PT 246  8 16  PT 232 20 16
PT 243 21 16  PT 242 22 16
PT 239 10 17  PT 238 10 18
PT 237  9 18  PT 236 20 17
PT 235 20 18  PT 234 21 18
PT 244 10 19  PT 233  9 19
PT 245  8 19  PT 246  7 19
PT 244 20 19  PT 232 21 19
PT 243 22 19  PT 242 23 19
PT 232 10 19  PT 233 11 19
PT 232 12 19  PT 233 13 19
PT 232 14 19  PT 233 16 19
PT 232 17 19  PT 233 18 19
PT 232 19 19  PT 233 20 19
PT 232 21 19
CARPET
END

TO ADD1
MAKE "P 13 MAKE "T 17 MAKE "C 13
MAKE "R 10 MAKE "Y 12
ADR :C :R :T :Y :P
END

TO ADD2
MAKE "P 11 MAKE "T 19 MAKE "C 11
MAKE "R 13 MAKE "Y 15
ADR :C :R :T :Y :P
END

TO ADD3
MAKE "P 10 MAKE "T 20 MAKE "C 10
MAKE "R 16 MAKE "Y 19
ADR :C :R :T :Y :P
END

TO ADR :C :R :T :Y :P
IF :C > :T MAKE "C :P MAKE "R 1+ :R
 IF :R > :Y STOP
PT 244 :C :R
ADR 1+ :C :R :T :Y :P
END

TO CARPET
MAKE "C 0
CAR :C
END

TO P :ALL
IF RC? CLEAR STOP
TELL :ALL SC 3 + (RANDOM * 11) /9
WAIT 30
P :ALL
END

TO CLEAR
CS SM
TELL :ALL SC 0 CARRY 17 SV 0 0
SXY 96 96 STOP
END

TO HELP
PRINT [ MOVIE RUNS THIS PROGRAM ]
WAIT 180
MOVIE
END
```

USE MAKESHAPE FOR THESE SHAPES:



SHAPES USE MAKESHAPE

USE MAKECHAR FOR THESE TILES



TILES USE MAKECHAR

There are 3 ways...

to find out what this article contains. 1: Read it, start to finish. 2: Skim over it, noting headings. (But beware: they can be misleading.) or 3: Read the next paragraph.

As usual, there will be the TI BASIC Beginners Tips, and the Extended BASIC Beginners Tips. They may seem quite ordinary to be "tips", however, not only can they be useful to beginners, but the Major Tidbits make up for them. The Major Tidbits, of which there is more than one, follow, Making & Saving Room, Keyboard full off Numbers and Computer Codes. The first deals with some potentially powerful memory-saving techniques. The second deals with keyboard diagrams: very useful stuff! The last is to do with "those mysterious tokens"!

TI BASIC Beginners Tips. People who translate programs for their own use, especially when their TI is their second (or third) computer, get "lazy" and don't treat multiple PRINTs properly. Like, 100 PRINT:PRINT:PRINT gets translated to 100 PRINT, 101 PRINT and 102 PRINT, when 100 PRINT ::: does exactly the same thing.

But if it does the same thing, why bother with it? Glad you asked. 100 PRINT ::: does this: takes advantage of something the TI has that many (indeed most) sadly lack; saves quite a few bytes of memory; and this means that it executes (carries it out) faster.

If you were watching, Extended BASIC programmers also benefit, for surprise! surprise! this trick also works in Extended BASIC! (Just remember to insert spaces or semicolons (;) between each colon.)

Extended BASIC Beginners Tips. IF you can get the Extended BASIC module, THEN you can program in Extended BASIC, ELSE you can't.

Yes, the IF statement is certainly powerful. But those graduating to Extended BASIC from TI BASIC can often fail to notice its usefulness, particularly if they've been a long time without Extended BASIC.

Normally, (i.e. in TI BASIC) you can only put line numbers after THEN and ELSE. ZX81 BASIC, which also allows but one statement per line, also allows one statement following THEN, but it does not have an ELSE clause. However, TI may have felt that that is really two statements, and so kept to line numbers.

Of course, you can still put a line number there, but Extended BASIC, what with its multiple-statement-per-line facility, allows multiple statements after both THEN and ELSE. But you can't go past the end of the line.

It's worth noting that many (indeed most) implications of BASIC do not even have an optional ELSE clause. (Commodore BASIC, or at least the early versions, is a good example.)

Sidelong Note: If you use a line number after THEN or ELSE, followed by a double-colon, the rest of the line is treated like a separate line!

Another helpful fact is that you can nest IF statements. But beware of ELSE pairings, especially if the nested IF statements are in the THEN parts. If in doubt, use otherwise unnecessary ELSEs to the next line.

The Extended IF statement, along with :: and (of course) ELSE can help to make Extended BASIC very powerful.

MORE →

Major Tidbit 1: Making & Saving Room.

This Tidbit is divided into general sections. A * in the margin at the start of a paragraph means that this spacing-saving technique also works in TI-BASIC.

1. The Screen. All right, I admit it, I am biased toward DISPLAY AT rather than CALL HCHAR. Actually, you've probably gathered that already, because of my interest with DISPLAY in previous articles.

Anyway, DISPLAY AT can save you memory. How? Well, most programmers use CALL HCHAR to place a character on the screen. Let's look at an example (yes, let's). Say that you've redefined character 126 (that's this shape: ~, it's called a "tilde") to be a one-of shape on the screen, that is, there is only one of them. Okay, so you use CALL HCHAR(12,10,126) to place it on the screen.

Just a moment! How many bytes does that statement take up? If you read the section entitled Computer Codes you will find out that it is... 25 bytes long. Now, let's translate it to DISPLAY AT(12,8):"~";. Note that this does exactly the same thing. The semi-colon? See Tidbits II, but in brief, it prevents DISPLAY AT from erasing the rest of the line.

So how many bytes? Refer to Computer Codes and you should come up with... 17 bytes! That's a saving of 8 bytes!

No! Don't turn to another page, yet. For this trick also works with more than one character. Another example: CALL HCHAR(12,10,126,3) = 29 bytes. Translate it to: DISPLAY AT(12,8):"~~~": which is 19 bytes long.

Some of you may be able to see what's coming: CALL HCHAR(12,10,126,14) (which is 30 bytes long) translates to DISPLAY AT(12,8): "~~~~~~~~~~~~~~"; which is also 30 bytes long!

So the rule here is if the number of repeated characters is 10 or over (just to be safe) then work out the number of bytes each format takes up before you decide which to put in your program.

On a tangent here, if someone wishes to translate your program to another computer, a DISPLAY AT will certainly be self-explanatory as to what it does, whereas what a CALL HCHAR does can be quite subtle or even down right obscure!

Now, both BASICs only allow 28 columns of text. To put letters in the columns 1, 2, 31 & 32, it is necessary to use CALL HCHAR.

* And that leads me to another thing- the wrap-around feature of both CALL HCHAR and VCHAR. In various programs which I've written, you will find CALL VCHAR(1,31,0,96). Now, let's disect it.

I wish 4 columns blank, 2 at the left and 2 at the right. Each column is 24 rows high. 4x24=96. So that explains the 96. But there are only two columns where the statement starts, columns 31 and 32. However, when VCHAR runs out of room, it starts again on the other side. And remember that I want 4 columns. So CALL VCHAR fills the two columns on the right, and "wraps-around" to fill two columns on the left.

* I have another idea. If your display contains many characters, then it may save memory (even time) to use a bank of PRINT statements to setup the screen.

However, I could probably guarantee that many TIsHUG BASIC programmers are somewhat lazy and prefer to just use DISPLAY AT (Extended BASIC only), CALL HCHAR and CALL VCHAR, as they think up the display, rather than sit down for 15 minutes or more and work out just how much memory they can save with PRINT. But believe me, if you really are desparate for more memory, the time taken to work out the least memory-consuming method can be worth it.

AND MORE

2. Variables. Variables on the TI are particulaerly flexible. If you disagree, remember that TI-BASIC was one of the first BASICs to allow reserved words to be part of a variable name.

But there's no need to try and take advantage of the 15 character limit (although I haven't seen anyone doing it!), because as a general rule, the more characters you feed a BASIC interpreter, the longer it takes. So if you can suitably describe the variable, by the name, withih about 6 characters, then by all means do so.

This is a good rule to go by: The more often a variable is used, the shorter its name should be. Remember that numeric variables take up about 8 or 9 bytes plus the length of the name to store.

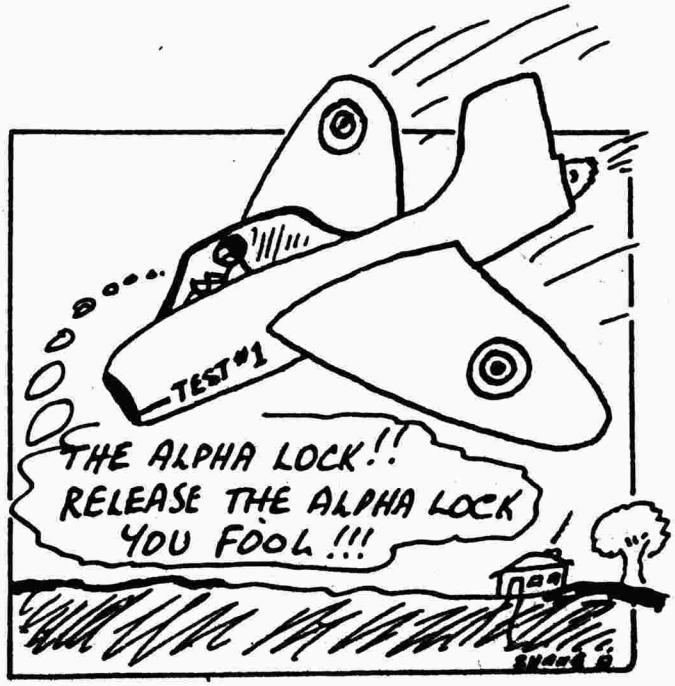
Which reminds me of an unusual programming method. Sometimes, you may have a constant that you use a lot. The common ones are 0 and 1. So then, with memeory conservation in mind, you choose to give the constant to a variable. The best names would be such like _, @, [, \ or]. But remember that you really should need that particular constant a lot (at least 7 or 8 times, preferably much greater) for it to save memory.

Notice that it saves even more memory if the constant is large, like 500.

I will present other sections of Making & Saving Room in later articles.

That's all for this article. Remember, if you want to tell me about anything, just write to:

Wade Bowmer
Lot 11 Yanderra Ave
Bangor 2234 AUSTRALIA

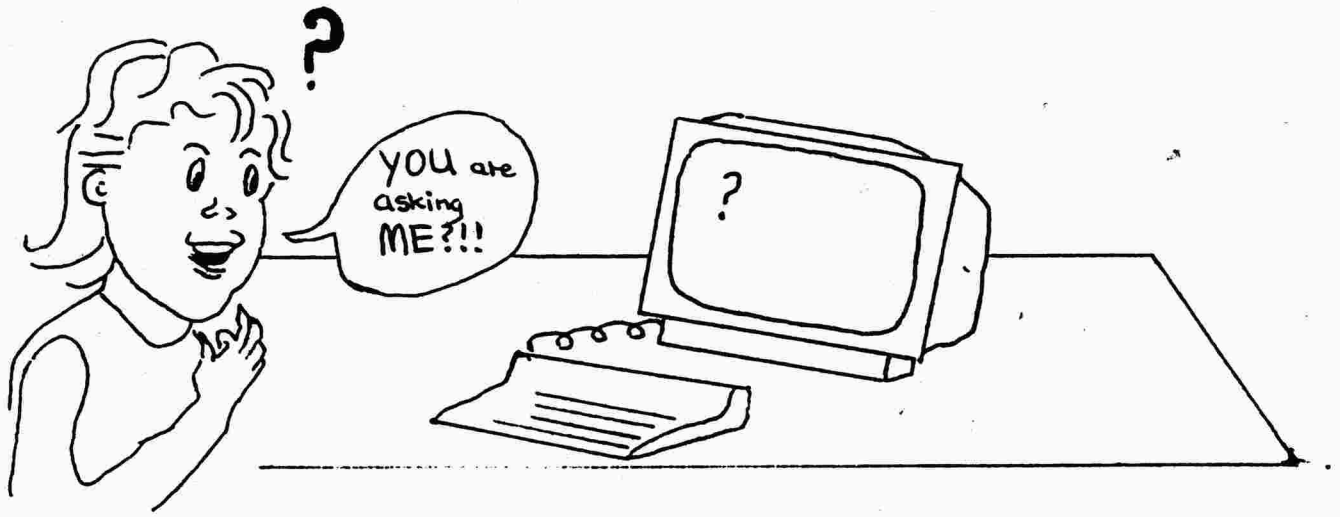


BASIC BEGINNINGS - VARIABLES

Last time I introduced you to numeric and string variables. To summarise, variables are a method of storing data, so that when we call up the variable name we are referencing the data itself - the value of that variable. To show the use of variables in TI Basic we'll run a little program on the computer, but first I have to explain the commands we'll use. PRINT and INPUT are two of the reserved words or commands used to tell the computer how to process the information we'll be giving it. Try PRINT "HELLO" on the computer. When you press ENTER immediately your command will be obeyed, and HELLO appears on the screen directly under your instructions. The quotation marks are not printed-everything you want the computer to PRINT must be enclosed in them, otherwise you will get an error message.The command we've just given made the computer work in its immediate mode. We didn't give the PRINT command a line number, so the computer didn't wait for any more instructions, it acted immediately. If we want to give it several lines to process, each line must be numbered; any sequence will do, but it's best to leave a gap so that extra lines could be added if needed: 10, 20, 30 or 100, 200, 300.

The command INPUT makes the program pause for input from the keyboard, so when the program is RUN it asks for your input with a question mark:type

```
10 INPUT A          (remember to press ENTER
RUN                   after each statement)
?                   is shown on screen. Type in a number, say 12, press
                    ENTER. The screen shows
? 12
**DONE**            as there is no further instruction. A is the variable
                    name, 12 is the value you've given it.
```



We can now have a go at a longer program that will show the use of string variables. The computer will ask for your name, and say 'hello' to you:

```
10 PRINT "I AM TEX, THE COMPUTER"
20 PRINT "FIRST NAME";          (the semi-colon means "display next
30 INPUT FIRST$                  character immediately after this text)
                                (FIRST$ is a string variable name, so is
40 PRINT "SURNAME";              SURNAME$)
50 INPUT SURNAME$
60 PRINT "HELLO, "; FIRST$; " "; SURNAME$; "!"
70 PRINT "I LIKE THE NAME "; FIRST$;
```

RUN

If you forgot the semi-colon, the next part of the text will be displayed at the beginning of the next line on the screen. Type in your name as asked, and make friends!

Let's add a bit more:

```
80 PRINT "WHICH YEAR IS THIS? (2 DIGITS)";
90 INPUT CURRENTYEAR                (asks for a numeric variable)
100 PRINT "WHICH YEAR WERE YOU BORN?(2 DIGITS)";
110 INPUT BIRTHYEAR                 (asks for a numeric variable)
120 PRINT "DEAR "; FIRST$; ", THIS YEAR"
130 PRINT "YOU ARE OR WILL BE"; CURRENTYEAR-BIRTHYEAR
140 END
```

RUN

Line 130 will take the birthyear from the current year and print out your age this year.

Each time we typed an answer to the question mark, we assigned a value to one of the variable names FIRST$, SURNAME$, CURRENTYEAR or BIRTHYEAR. Another way to assign a value to a variable is to do it as part of the program, instead of supplying it from the keyboard. To do this use the assignment statement LET for example:

```
10 LET A=4
20 LET C=1
30 LET B=A+C
```

LET is optional in TI Basic, so we could just say

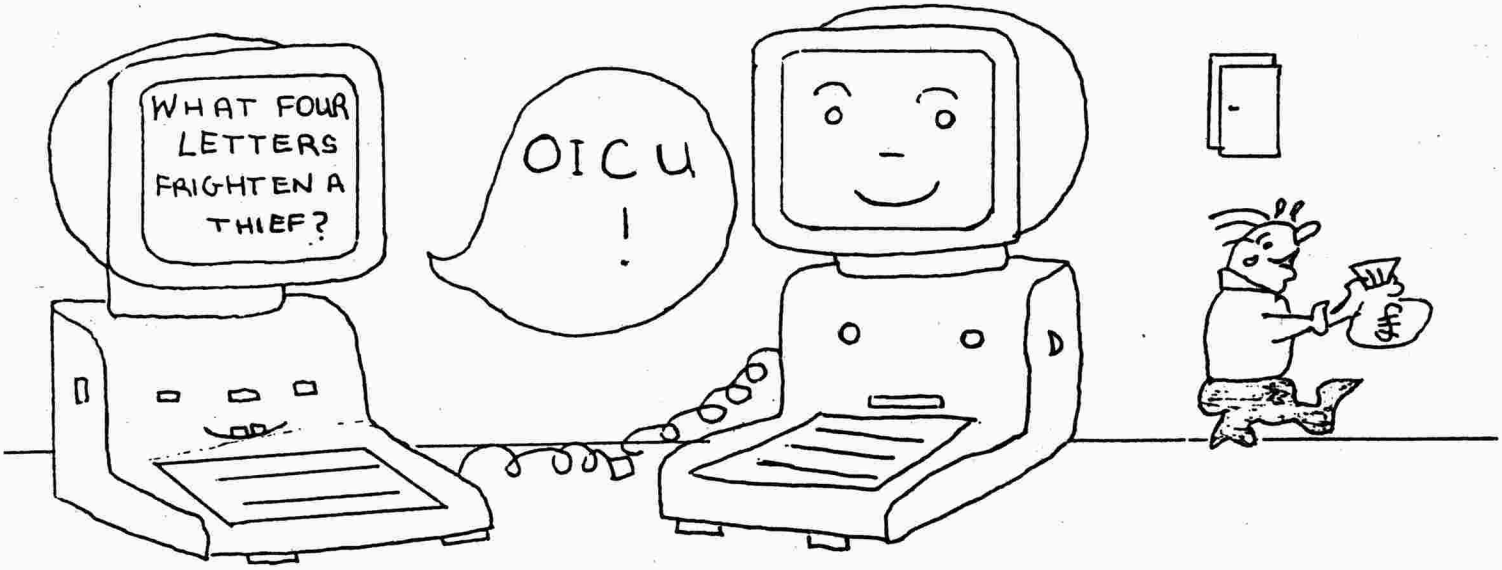
```
10 B=A+C
```

and so on. Now we know two ways to assign values to variables: use INPUT when the value is supplied during the running of the program via the keyboard, use the assignment statement to store a value within the program itself. In line 30 above B is the variable and A+C is an expression. The sign + is an operator, as are -, * and /. There are rules for writing assignment statements, the general form being:

<variable> = <expression> ie. there must always be a variable on the left and an expression on the right. Here are a few sample expressions:

```
3              (a number)
A              (a variable)
2+2            (number, operator, number)
A+B * 3        (variable, operator, expression)
```

In longer expressions brackets may have to be used, just like in maths:
3+(A+2)/2 or B+((C*2)+(D/2))/4.
But unlike maths, the = sign does not mean exactly the same, for example
10 A=1
20 A=A+1 the expression A=A+1 in maths is nonesense, but valid in Basic. Think of it as imagining the LET at the beginning of the statement: the variable on the left of the = sign receives the value of the expression on the right of the = sign.
In an expression each number or variable in the statement must be joined to the next by an operator, so A=B+C(D/3) is invalid, but A=B+C*(D/3) is valid.
Now we know more about the rules for assigning values to our variables. The value we put in that memory location will stay there until we change it, or until the computer is switched off.



In the last issue of TI*MES, we met two string operators: '&' which joins strings together, and SEG$, which cuts them up! Here is one more to play with.
LEN It's sometimes useful to know the number of characters in a particular storage location. LEN will count them for us:
If W$="ROTTEN" then LEN(W$) = 6. Try this on the computer:

```
10 INPUT W$
20 LET N=LEN(W$)
30 PRINT W$; " HAS";N;" LETTERS"
RUN
? CRYING                    (Type in any word you like)
CRYING HAS 6 LETTERS
**DONE**
```

PROGRAMMING - FROM BASIC TO ASSEMBLY LANGUAGE

(A gentle introduction) by Geoffrey Coan

This series of articles aims to show you some of the different programming languages available on theTI-99/4a and how they may best be used. To illustrate the features of the different languages I shall take a simple game and show how it can be written in different programming languages, using the languages' features to improve the game.

The example game that I will use is the game of "Breakout". This fairly common computer game shows a wall of coloured "bricks" near the top of the screen and three normal "walls" around the left, top and right sides of the screen. A ball bounces around this playing area, every time it hits a brick it knocks the brick out of the wall and the score increases by one. On the fourth side of the playing area (the bottom of the screen), there is no wall, instead there is a bat that can be moved left and right by the player. The ball will bounce off the bat if the bat is directly underneath the ball but if the ball goes past the bat and off the bottom of the screen then the game is over. The aim of the game therefore is to keep the ball in play so that it knocks out all the bricks in the wall. When the whole wall of bricks is destroyed then a new wall is drawn and the game repeats.

This is a deliberatly simple game program that can easily be written in TI basic without too many speed problems, however by using advanced facilities of the TI like sprites in Extended basic, we can improve the games playability.

The first version of Breakout is written in TI Basic (but will run in Extended Basic), subsequent versions will be for Extended Basic, C and 9900 Assembly Language with possibly Forth and Pilot-99 versions as well.

The article does assume an elementary knowledge of TI Basic, for introductionary articles on Basic, readers should refer to TI*MES volumes

MAIN SECTIONS OF PROGRAM

Lines 150-230 : Setup all the character patterns to be used by the program for the walls, bricks, bat and ball.
240-270 : Setup the colours of the characters to be used by the program, all characters are white on different coloured backgrounds.
280-300 : Draw left side, top and right side walls of the playing area.
310-380 : New game routine. This prints the top line of the screen (current and high score), sets the current score to 0 and puts the bat at a random position on the bottom line of the screen.
390-490 : New screen routine. This calculates a random start position for the ball on the screen and draws the wall of bricks on the screen.
500-770 : This is the main loop of the program, broken down into -
510-520 : Check to see what is in the new position of the

	ball and plot the ball in its new position.
530-560	Cause the ball to bounce off the top or side walls of the playing area if it is about to hit any of the walls.
570-580	Cause the ball to bounce off the bat if the bat is below the ball.
590-610	Test if the ball has hit a brick. If it has then branch to the hit wall subroutine at Lines 1000-1120
620	Test if the ball has reached the bottom line of the screen. If it has then it must have missed the bat so go to the end of game routine in Lines 580-990.
630-730	Test to see if any keys have been pressed on the keyboard. If the S key has been pressed and the bat is not at the left edge of the screen, then move the bat to the left. If the D key has been pressed and the bat is not at the right edge of the screen then move the bat to the right.
740	Erase the ball from its old position on the screen.
750-760	Calculate the new position of the ball on the screen.
770	Jump back to the start of the main loop at Line 510.
780-990	End of game routine. Update the high score if the current score is greater than the old high score and then print the new high score at the top of the screen. Draw the end of game messages on the screen and then wait for a key to be pressed. When a key has been pressed, erase the end of game messages the bat and the ball from the screen and go to Line 310 to start a new name.
1000-1120	Hit wall subroutine. Erase the brick at the side of the ball off the screen (since bricks are two characters wide and the ball is only one character wide), add 1 to the current score and redisplay the score on the screen. Make the ball bounce off the bricks and then return to the main program
1130-1170	Subroutine to draw the string T$ starting at screen co-ordinates (B,A) on the screen.
1180-1190	Data for the background colors of the characters on the screen, used by Lines 240-270.

MOTION OF THE BALL

The motion of the ball on the screen is maintained by keeping (X,Y) co-ordinates for the ball in numeric variables X and Y, and keeping motion numeric variables I and J. The motion variables only hold -1 or +1 and they determine which way the ball is moving in the X and Y directions respectively (e.g.: if the ball is moving down and to the right then the X direction motion variable, I, will equal +1 and the Y direction motion variable, J, will also equal +1). To make the ball appear to bounce off a wall or a bat, the appropriate motion variable simply has its sign changed (so +1 goes to -1 and -1 goes to +1).

eg: Imagine the ball is at screen position (4,12) which is half way across the screen and near the top wall. If the ball is moving upwards and to the right then the X direction motion variable will be +1 and the Y direction motion variable will be -1. The variables are: X=12, Y=4, I=+1, J=-1

At the end of the main loop of the program (lines 750-760), the motion variables are added to the ball co-ordinates to move the ball in the required direction. The variables now become:

X=13, Y=3, I=+1, J=-1

At the start of the main loop of the program (Line 520), the ball is plotted on the screen at its new position and so it is now next to the top wall of the playing area. Since Y now equals 3, the test in Line 530 of the program fails and so Line 540 changes the sign of the Y motion variable to "bounce" the ball downwards. Hence J=-(-1) or J=+1 and so the variables are:
X=13, Y=3, I=+1, J=+1
Then at the next end of the main program loop, the motion variables are added to the ball co-ordinates and they become:
X=14, Y=4, I=+1, J=+1
when the ball is plotted again on the screen it will be moved down one line on the screen (i.e.: to the player it appears to "bounce" off the top wall).

VARIABLES USED IN THE PROGRAM

Several of the numeric variables are used within the program for simple counts in FOR-NEXT loops but use in the main body of the program are:

A	Y screen co-ordinate to start printing string at, used for subroutine in Lines 1130-1170. Also used for the number of the character on the screen that the ball moves on to.
B	X screen co-ordinate to start printing string at, used for subroutine in Lines 1130-1170.
H	X co-ordinate of the left-hand character of the bat (on the bottom line of the screen).
HI	High score.
I	Motion variable of the ball in the X direction.
J	Motion variable of the ball in the Y direction.
K	ASCII value of the last key pressed on the keyboard (used in Lines 630-740).
S	Status of the keyboard (from CALL KEY routine).
SC	Current score.
T$	String to be printed on the screen, used in subroutine in Lines 1130-1170.
X	X co-ordinate of the ball on the screen.
Y	Y co-ordinate of the ball on the screen.
Z	Loop variable in several places in the program.

COMPLICATED PARTS OF THE PROGRAM

This is the part of the article where I try to explain the more complicated (or dare I say, clever) parts of the program. The BASIC version of Breakout is fairly simple but because of the limitations of TI Basic, some programming tricks have been used to shorten the program size. These are:

Lines 110 and 1010	Since the ball is one character wide on the screen and the bricks are two characters wide, the program must clear the half brick to the side of the ball off the screen whenever the ball hits a brick. If the ball is at an even number X co-ordinate, the half brick will be to the right of the ball or if the ball is on an odd number X co-ordinate then the half brick will be to the left of the ball. This can be verified by checking the screen diagram. An even number is any number that is EXACTLY divisible by 2, or in computer terms, the number

divided by 2 is the same as the integer part of the number divisible by 2. Or, for any number Q, the result (Q / 2) is the same as INT (Q / 2). This comparison is made by the user defined function in line 110, for an even number the comparison "INT(Q/2)=Q/2" will be true and the function EVEN(Q) will return TRUE. For an odd number the comparison will be false and so EVEN(Q) will return FALSE. Line 1010 of the program uses this function to determine if the X co-ordinate of the ball is an even number or not. If it is an even number then the 1/2 brick to the right of the ball is erased, otherwise the 1/2 brick to the left is erased.

Lines 550 & 570 : Since TI basic cannot use the logical operators AND and OR to form complicated conditional statements, we must use the mathematical symbols * and + to duplicate the effect we require. This use of multiplication (*) where an AND would normally be used and addition (+) where an OR would normally be used has been covered in previous articles in TI*MES but basically, the line 550 which is:

```
    IF (X()4)*(X()31) THEN 570
```

can be translated to :

```
    IF (X()4) AND (X()31) THEN 570
```

which means "IF the X co-ordinate of the ball is not equal to 4 AND the X co-ordinate of the ball is not equal to 31 THEN GOTO line 570 of the program". Thus line 560 of the program which bounces the ball off the side walls of the playing area is only executed if the ball is next to the walls.
Similarly, line 570 of the program translates to:

```
  IF (Y=23) AND ( (X=H) OR (X=H+1) ) THEN 580
      ELSE 570
```

which will cause the ball to bounce off the bat if it is above it.

The rest of the program is fairly straight-forward, if you do get stuck then try to run the program through on a piece of paper by writing down what is happening at each stage and the value of the variables involved; or alternatively, drop me a line !

```
1 REM BREAKOUT
2 REM [ BASIC VERSION ]
3 REM BY GEOFFREY COAN
100 CALL CLEAR
110 DEF EVEN(Q)=(INT(Q/2)=Q/
2)
120 CALL SCREEN(2)
130 RANDOMIZE
140 HI=0
150 CALL CHAR(91,"0101010101
010101")
160 CALL CHAR(92,"0000000000
0000FF")
170 CALL CHAR(93,"8080808080
808080")
180 CALL CHAR(94,"FF00000000
000000")
190 CALL CHAR(95,"3C4E8F8FF1
F1723C")
200 FOR Z=96 TO 136 STEP 8
210 CALL CHAR(Z,"FF808080808
080FF")
220 CALL CHAR(Z+1,"FF0101010
10101FF")
230 NEXT Z
240 FOR Z=3 TO 14
250 READ A
260 CALL COLOR(Z,16,A)
270 NEXT Z
280 CALL VCHAR(3,3,91,22)
290 CALL HCHAR(2,4,92,28)
300 CALL VCHAR(3,32,93,22)
310 REM NEW GAME
320 SC=0
```

```
330 T$ = "SC= 0
I= "&STR$(HI)
340 A=1
350 B=3
360 GOSUB 1140
370 H=INT(RND*14)*2+4
380 CALL HCHAR(24,H,94,2)
390 REM NEW SCREEN
400 X=INT(RND*26)+5
410 Y=INT(RND*9)+14
420 I=INT(RND*2)*2-1
430 J=-1
440 FOR A=6 TO 11
450 FOR B=4 TO 30 STEP 2
460 CALL HCHAR(A,B,A*8+48)
470 CALL HCHAR(A,B+1,A*8+49)
480 NEXT B
490 NEXT A
500 REM MAIN LOOP
510 CALL GCHAR(Y,X,A)
520 CALL HCHAR(Y,X,95)
530 IF Y<>3 THEN 550
540 J=-J
550 IF (X<>4)*(X<>31)THEN 57
0
560 I=-I
570 IF (Y=23)*((X=H)+(X=H+1)
)THEN 580 ELSE 590
580 J=-J
590 IF A=32 THEN 620
600 GOSUB 1010
610 IF INT(SC/84)=SC/84 THEN
 400
620 IF Y=24 THEN 790
630 CALL KEY(3,K,S)
640 IF S=0 THEN 740
650 CALL HCHAR(24,H,32,2)
660 IF K<>83 THEN 700
670 IF H=4 THEN 730
680 H=H-2
690 GOTO 730
700 IF K<>68 THEN 730
710 IF H=30 THEN 730
720 H=H+2
730 CALL HCHAR(24,H,94,2)
740 CALL HCHAR(Y,X,32)
750 X=X+I
760 Y=Y+J
770 GOTO 510
780 REM GAME OVER
790 IF SC<=HI THEN 850
800 HI=SC
810 T$=STR$(HI)
820 A=1
830 B=26
840 GOSUB 1140
850 T$="GAME OVER"
860 A=15
870 B=12
880 GOSUB 1140
890 T$="PRESS ANY KEY TO REP
LAY"
900 A=18
910 B=5
920 GOSUB 1140
930 CALL KEY(0,K,S)
940 IF S=0 THEN 930
950 CALL HCHAR(15,13,32,9)
960 CALL HCHAR(18,6,32,23)
970 CALL HCHAR(Y,X,32)
980 CALL HCHAR(24,H,32,2)
990 GOTO 310
1000 REM HIT WALL SUBROUTINE
1010 IF EVEN(X) THEN 1040
1020 CALL HCHAR(Y,X-1,32)
1030 GOTO 1050
1040 CALL HCHAR(Y,X+1,32)
1050 SC=SC+1
1060 T$=STR$(SC)
1070 A=1
1080 B=7
1090 GOSUB 1140
1100 J=-J
1110 CALL SOUND(100,500,5)
1120 RETURN
1130 REM PLOT STRING SUBROUT
INE
1140 FOR Z=1 TO LEN(T$)
1150 CALL HCHAR(A,B+Z,ASC(SE
G$(T$,Z,1)))
1160 NEXT Z
1170 RETURN
1180 REM DATA FOR COLOURS
1190 DATA 2,2,2,2,2,2,3,14,8
,11,6,10
```

SCREEN DIAGRAM

```
 12345678901234567890123456789012
1   SC= 0                HI= 0     1
2   ------------------------------ 2   (- top
3  |                               |3
4  |                               |4  of play area
5  |                               |5
6  |[][][][][][][][][][][][][][][]|6
7  |[][][][][][][][][][][][][][][]|7
8  |[][][][][][][][][][][][][][][]|8
9  |[][][][][][][][][][][][][][][]|9   (- wall
0  |[][][][][][][][][][][][][][][]|0   of bricks
1  |[][][][][][][][][][][][][][][]|1
2  |                               |2
3  |                               |3
4  |                               |4
5  |                               |5
6  |                               |6
7  |                               |7
8  |                               |8
9  |                               |9
0  |              O                |0    (- ball
1  |                               |1
2  |                               |2
3  |                               |3
4  |          ~~                   |4    (- bat
 12345678901234567890123456789012
```

by Geoffrey Coan, 76 Roundcroft, Romiley, Stockport, Cheshire.

A REALIST'S COMPUTER GLOSSARY

BASIC-	Computer language used for generating error messages.
Extended BASIC-	Similar to BASIC but with more error messages.
Assembly language-	Very complicated way to crash your system
Mini-memory-	Yes, except the price.
PRINT-	Program statement that displays information in the wrong screen position.
DISPLAY AT-	Similar to PRINT, but quicker.
Array-	A quick way to lose track of stored tabulated data.
DIM-	How you feel when trying to sort out multi-dimensional arrays.
Expanded system-	Refers mainly to TI's projected profits. (Historical interest only.)
Speech synthesiser-	Small talking box that teaches you to spell in American.
Cassette system-	Slow method of storing programs you'll probably never use again.
Disk system-	Fast method of storing programs you'll probably never use again.
END-	Statement you type when you think your program is complete.
EDIT-	Method of deleting 'END' and inserting five more program lines.
Cursor-	Small blob on the screen that flashes in time with your swear-words.
NEW-	The random keys always found by your young son when you answer the phone.
NUM-	Try a softer chair.
I/O-	Meaningless except when preceded by EIE.
RANDOMIZE-	Shakes up two small foam-rubber dice somewhere inside the console.
Port-	Lets air into the side of the console. Much discussed by hardware buffs.

CALL SOUND-	Laborious method of generating music with wrong notes.
RS 232-	A complicated technical method of failing to get a printer to work.
Modem-	A device that fails to connect two computers by telephone. (Sponsored by BT)
CALL CLEAR-	Blanks the screen ready for the next error message.
QUIT-	The key you gently brush with your sleeve after three hours of typing.
PRK-	A module that failed to put this glossary into alphabetical order.
NEXT-	Increments a counter. Much used in Ludo.
DEF-	Used in combination with other words by programmer's wife.
LET-	Reserves space inside the computer, as in "Room to LET".
INT-	Exasperates the computer by removing all those decimal places it sweated over.
Sprite-	Small graphic that moves across the screen, almost totally out of control.
PI-	To do with chips or something.
RAM-	Male sheep with a good memory.
RUN-	A command that the manual says will get your program going.
OPEN-	A statement that cuts the computer off from the outside world.
CLOSE-	Oh, alright then.

TIny TIm.

USING EXTENDED BASIC II PLUS

by Peter Walker

This does not claim to be a comprehensive review of Mechatronic's Extended Basic II Plus module (XB2+), but gives a personal insight into how I have used its special features.

I bought the module at the Leeds Show and immediately set about modifying some well-loved programs to use the new commands and features available. I noted that the module runs very hot - I hope not too hot! It is supported by a fat manual describing the numerous extra commands but the translation from the German is a little suspect at times ("Uncovered use of CALL MOVE can result in malfunctions of the computer"). The module provides two main areas of enhancement. Firstly a set of new commands (CALLs) which add to the standard Extended Basic and secondly a wide range of graphic commands, courtesy of APESOFT. The latter must be preselected in immediate mode before the program starts. Since the bit mode graphics environment is quite different from the normal 32 column graphics mode, a number of normal commands cannot be used at the same time as the graphics. I have not used the graphics much although this is the key selling point for XB2+. Anyone who has got a copy of the group program D7 (Extended Basic 32K Demo) will have a good idea of the graphic capabilities that the APESOFT software possesses and which are now accessible via XB2+.

My main use of the module has been with the extra CALLs used in normal non-graphic mode. These provide some commands missed out in regular XBas (such as POKEV) but found elsewhere (E/A, Minimem); together with a range of utilities that one could otherwise provide via short Machine Code calls were it not for the lack of time to write all these. (Thus Assembler freaks might find the features of XB2+ of less value). CALLs that I have found useful are:

BHCOPY - copies screen to an Epson compatible printer using bit image mode. The printer must also be set to AUTO FEED XT using pin 14 on the parallel interface. BHCOPY only works in the normal 32 column graphics mode. It is also very slow. Apart from these points, its a very useful utility.

VPEEK - The XB2+ name for PEEKV. Very useful to have this in XBas.

VPOKE - Ditto POKEV.

WAIT - Causes program to halt for a specified time or until a key is pressed. You'll recognise this as a useful feature for start-up screens and is widely employed in TI Modules.

MOVE - Allows chunks of CPU or VDP memory to be moved to another location in either CPU or VDP RAM. Thus you can move the area defining the screen in VDP and store it in CPU and similarly take parts of CPU and load it into VDP. I use this to provide "instant" screen switching for the display of help screens. (And it is INSTANT!). There must be lots of other interesting uses for rapid changes of sound, speech, sprites and character definitions which I have yet to explore. This is an extremely useful command.

MSAVE - Saves part of CPU RAM to a program format file. This enables one to store, for example, machine code programs in the 8K memory to disk or cassette (a boon to cassette owners - please note!). It could also be used in conjunction with MOVE to store screen contents to a file. Lots of uses.

MLOAD - The opposite of MSAVE. This allows loading of machine code programs direct into memory. Not only useful for cassette users but also Disk users who wish to bypass the appallingly slow loader used in normal XBas. For example Oaktree's Display Enhancement Package which regular readers will know I'm a great fan of, now loads in just 6 seconds intead of 90 seconds.

QUITOF/QUITON - Disables/Enables QUIT key. Stays active even after program is stopped. Useful to stop accidental aborting of programs during keying-in or, of course, during execution.

FIND - Searches an array for a specified string and returns the index value of the item. Extremely useful in database programs for finding a key data item quickly and then accessing whole record on disk. Very fast in operation. Another use is for translating one string to another. If you use two arrays, you can FIND the first string in the first array and use the returned index to address the second array where the translated strings are stored. I use this to allow abbreviated field entries to be instantly replaced by their full text. (An example would be entering JAN and having the program display the full text JANUARY). A very useful command to have in XBas.

There are other commands which I have not covered but which may be of use to others. These are GPEEK, ALLSET, BYE, NEW, RESTORE, SPROF/SPRON, and SCREENOF/SCREENON.

I don't intend to review the graphics part of the module. There are 40 commands for doing bit mode graphics which allow very simple drawing of lines, circles, squares, ellipses and addition of shading. Very useful for graphics enthusiasts.

CONCLUSION

I am aware that a number of XBas extensions are now on the market. However this German effort, I feel, is particularly noteworthy and I find I am regularly using its powerful features. I hope this short article has also given some ideas on the applications that can be handled.

DIY HARDWARE PROJECT

Removing the Alpha Lock/Joystick interference problem.

By Geoffrey Coan.
76 Roundcroft, Romiley, Stockport, Cheshire, SK6 4LS.

One of the few hardware problems that exists on the TI-99/4A, as most TI owners probabally know, is the problem that the joysticks will not "up" unless the Alpha Lock key is off (ie: the key is in the up position).

This problem is easily resolved by the addition of one diode to the keyboard circuit and the cutting of one copper track. The components needed for this operation are one low voltage General Purpose Silicon or Germanium diode (eg: Maplin type No. QA95 or QA200); a soldering iron and solder.

Do not attempt this operation if you are unsure of what you are doing !

Turn the computer over and remove the bottom of the case by undoing all the 7 screws and removing the oustide part of the power switch. Locate the bottom right hand corner of the keyboard and compare to the wiring diagram below. Of the two connections to the Alpha Lock key, one of them is connected to copper track running around the bottom edge of the PCB and the other is to a connection going underneath the board. This connection comes up through the board just by the Shift key and then the copper track runs off accross the keyboard PCB towards the cartridge port. It is this copper track that must cut. Scrape back some of the protective covering off this copper track, exposing the copper and then solder the anode (positive) end of the diode (usually marked by a ring or notch on the diode) to the track. Solder the other end of the diode to the copper track where is comes through the board by the Shift key. Now, carefully cut the copper track which the diode now bridges with a sharp knife - make sure that you cut the correct track ! Stick some insulating tape around the diode and its legs to stop the legs shorting out parts of the keyboard and then re-assemble the console case.

Switch on and if all has gone correctly then you should get upper case with the Alpha Lock on and lower case with the Alpha Lock off (as before); but now the joystick will work properly, registering all directions regardless of the state of the Alpha Lock key. If you only get lower case then the diode has been soldered in the wrong way round - disassemble the console and resolder it the other way around.

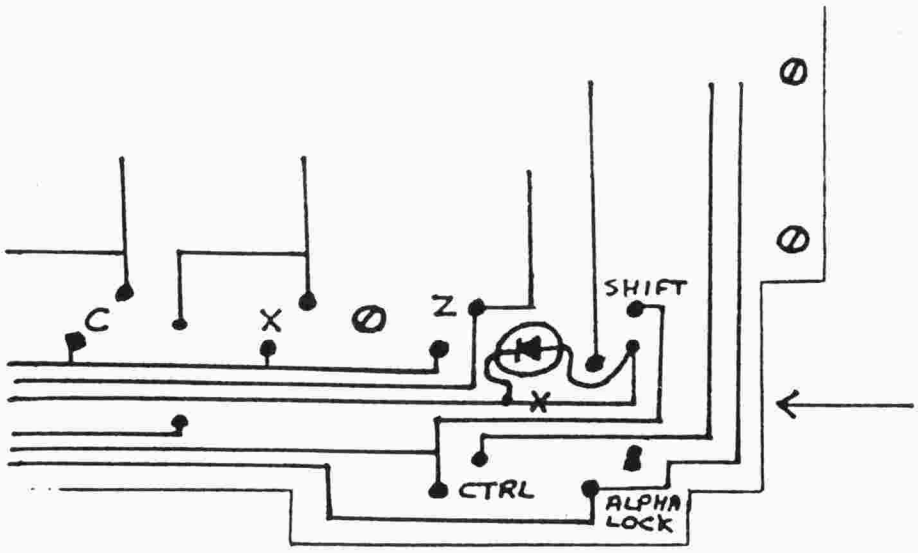
This operation has been performed on two different TI consoles and they both work OK. However, on one console (a 1982 one made in Lubbrock, Texas) after about 15 minutes of use (when the console had warmed up) the Alpha Lock key stopped working. This was cured by connecting a transister in instead of a diode (cutting the spare leg off the transister); this console then worked perfectly.

Both consoles have now been converted for well over three years and have worked faultlessly with the Alpha Lock key ignored all the time.

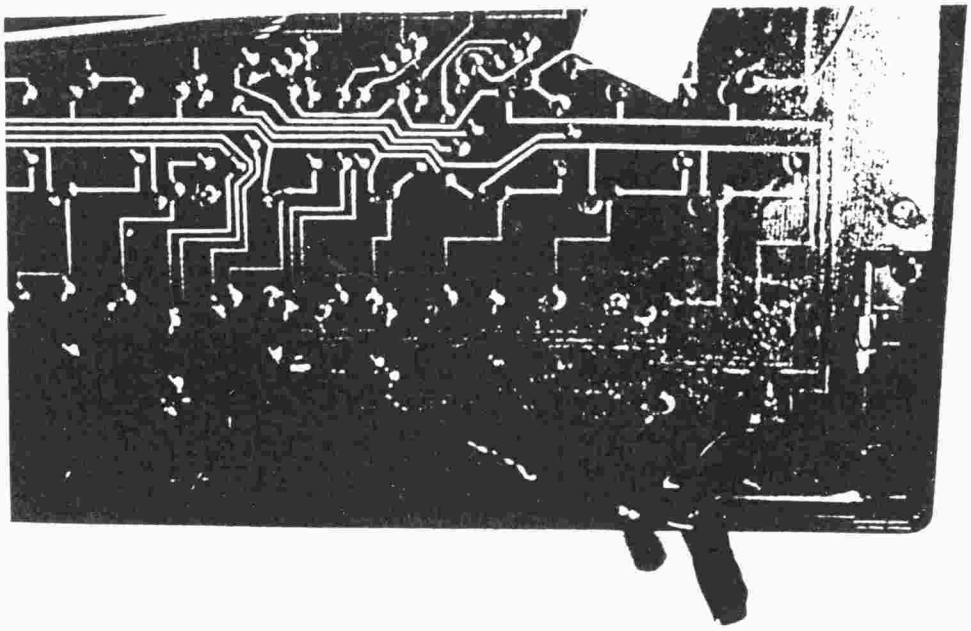
Please write to me at the above address if you have any comments, etc to make on this DIY hardware project.

Geoffrey Coan

The author of this article & TI*MES accept no responsibility for any damage whatsoever made to TI-99/4A consoles when performing this project.



Cut Copper Track at intersection of arrows



PROGRAM LISTING..... XMAS TREE
by MAURICE RYMILL

```
10 REM  ****************
20 REM  *   XMAS TREE   *
30 REM  ****************
40 REM  TI BASIC
50 REM  18/11/85
60 REM    ABOUT 11000 BYTES
70 REM  PRESS ANY KEY TO END SCREEN DISP
LAY
80 CALL CLEAR
90 CALL SCREEN(2)
100 CALL COLOR(1,3,1)
110 CALL COLOR(2,3,1)
120 CALL COLOR(3,7,1)
130 CALL COLOR(4,7,1)
140 CALL COLOR(5,10,1)
150 CALL COLOR(6,10,1)
160 CALL COLOR(7,10,1)
170 CALL COLOR(8,10,1)
180 CALL COLOR(9,7,1)
190 CALL COLOR(10,14,1)
200 CALL COLOR(11,6,4)
210 CALL COLOR(12,8,4)
220 CALL COLOR(13,12,4)
230 CALL COLOR(14,7,4)
240 CALL COLOR(15,15,1)
250 CALL COLOR(16,5,16)
260 REM  PATTERN-IDENTIFIERS
270 REM  FORMAT: IDENTIFICATION, CHARACT
ER NUMBER, HEXADECIMAL STRING...
280 REM   EXAMPLE: TREE TRUNK,142,4E53B6
35C659487A, TREE BODY,143,00000000000000
00...
290 DATA DEC,33,0103070E1C3830E0,DEC,34,
8060383C1E0F0F07
300 DATA M,35,03060C0103060C18,M,36,8099
FADD193366CC,M,37,0080808080386CD8,M,38,
0000000020007699
310 DATA M,39,000000000030C987,M,40,0000
00000088983A,M,41,3163C60000000000,M,42,
9831E0000000000
320 DATA M,43,A3CCF00000000000,M,44,3163
C60000020000,M,45,0306030101030203,M,46,
6CD838D8183060C0
330 DATA M,48,0000000001030600,M,49,0000
0000F0830B0E,M,50,00000000F0800000,M,51,
0000000200000000
340 DATA M,52,0000000000030638,M,53,1030
60E0E0E1633E,M,54,0000003C67CC9831,M,55,
000000CC766CC99F
```

...XMAS TREE BY MAURICE RYMILL

```
350 DATA M,56,0000003E6C99B6DC,M,57,0000
00]E66831EF8
360 DATA LEFTSIDE OUT,96,0107070F0F0F]F7
F,RIGHTSIDE OUT,97,80E0E0F8F0F0F8FE,BORD
ER TOP,98,7F7F3F07070]0]00
370 DATA TREE BOTTOM,99,FFF8F8F0F0E0C000
,B,100,FF3F0F0707030]00,DEC,101,00063632
1E64740E
380 DATA BORDER TOP,102,FFFFFFFFFF000000
,BORDER BOTTOM,103,000000FFFFFFFFFF
390 DATA PLUM,104,10FEFEFEFE7C38]0,BELL,
107,10103838?C?C?CFE
400 DATA PLUM,112,10FEFEFEFE7C38]0,DIAMO
ND,113,1010387CFE7C38]0,BELL,115,1010383
87C7C7CFE
410 DATA PLUM,120,10FEFEFEFE7C38]0,DIA,1
21,1010387CFE7C38]0,BELL,123,10103838?C7
C7CFE
420 DATA PLUM,128,10FEFEFEFE7C38]0,DIAMO
ND,129,1010387CFE7C38]0,BELL,131,10103838
87C7C7CFE
430 DATA LEFT INSIDE,136,FEF8F0F0E0E0C00
0,RIGHT INSIDE,137,7F1F0F0F0707030O,CT,1
38,FFE7E7830101010]
440 DATA BOTTOM IN,139,0]07070F0F0F3FFF,
B,140,80C0E0E0E0F0FCFF
450 DATA TRUNK,142,4E53B635C659487A,TREE
 BODY,143,0000000000000000,POT L,144,3F3
F3F3F3F3F3F3F
460 DATA POT LEFT BOTTOM,146,3F3F3F3F3F3
F0F0F,POT R,147,FCFCFCFCFCFCFCFC
470 DATA POT R B,149,FCFCFCFCFCFCF0F0,P
B,150,FFFFFFFFFFFFFFFF
480 DATA TOP,152,C3C33C]8]83CC3C3,STAR R
ADIAL,153;FFFFFF0000FFFFFF,STAR RADIAL,1
54,E7E7E7E7E7E7E7E7
490 DATA REM DEFINE-LOOP
500 RESTORE 290
510 FOR CODE=33 TO 154
520 READ IDENTIFICATION$,CHARACTERNUMBER
,HEX$
530 IF CHARACTERNUMBER>CODE THEN 550
540 GOTO 560
550 CODE=CHARACTERNUMBER
560 CALL CHAR(CODE,HEX$)
570 NEXT CODE
580 REM  START SCREEN DISPLAY
590 CALL CLEAR
600 REM  ----TREE BODY----
610 CALL HCHAR(24,1,143,32)
620 CALL HCHAR(19,7,143)
630 CALL HCHAR(18,2,143,11)
640 CALL HCHAR(17,3,143,9)
650 CALL HCHAR(16,3,143,8)
660 CALL HCHAR(15,4,143,7)
670 CALL HCHAR(14,4,143,7)
680 CALL HCHAR(13,4,143,6)
690 CALL HCHAR(12,5,143,5)
```

```
700 CALL HCHAR(11,6,143,4)
710 CALL HCHAR(10,6,143,3)
720 CALL HCHAR(9,6,143,3)
725 CALL HCHAR(8,6,143,2)
730 CALL VCHAR(7,7,143,2)
740 REM ----TREE TRUNK----
750 CALL VCHAR(20,7,142,2)
760 REM ----PLANT POT----
770 CALL VCHAR(22,6,144,2)
780 CALL VCHAR(22,7,150,3)
790 CALL VCHAR(22,8,147,2)
800 REM SCREEN LOCATION DATA
810 REM FORMAT: IDENTIFICATI
ON$,ROW,COLUMN,CHARACTERNUMB
ER
820 DATA POT BASE LEFT SIDE,
24,6,146,POT BASE RIGHT SIDE
,24,8,149
830 REM ----FOLIAGE----
840 DATA L0,18,1,96,L1,18,2,
136,L0,17,1,96,L1,17,2,136,L
0,16,1,96,L1,16,2,136,L0,15,
2,96,L1,15,3,136
850 DATA L0,14,2,96,L1,14,3,
136,L0,13,2,96,L1,13,3,136,L
0,12,3,96,L1,12,4,136,L0,11,
4,96,L1,11,5,136
860 DATA L0,10,4,96,L1,10,5,
136,L0,9,4,96,L1,9,5,136
870 DATA L0,5,6,96,CT,5,7,13
8,L0,6,6,96,L1,6,7,136,L0,7,
5,96,L1,7,6,136,L0,8,5,96,L1
,8,6,136
880 DATA R0,5,8,97,R0,6,9,97
,R1,6,8,137,R0,7,9,97,R1,7,8
,137,R0,8,9,97,R1,8,8,137,R0
,9,10,97
890 DATA R1,9,9,137,R0,10,10
,97,R1,10,9,137,R0,11,11,97,
R1,11,10,137,R0,12,11,97,R1,
12,10,137
900 DATA R0,13,11,97,R1,13,1
0,137,R0,14,12,97,R1,14,11,1
37,R0,15,12,97,R1,15,11,137,
L0,16,12,97
910 DATA L1,16,11,137,L0,17,
13,97,L1,17,12,137,L0,18,13,
97,B,18,12,137,B,19,12,99,B1
,19,11,139
920 DATA B1,19,10,140,B0,19,
9,100
930 DATA BOTTOM,19,5,96,B,19
,6,99,B,19,7,139,B,19,8,99,B
,19,4,100,B,19,5,140,B,19,2,
96,B,19,3,99
940 DATA B IN,18,7,139,B OUT
,18,8,136
950 REM ----CROSS----
960 DATA TOP,2,7,152,L RADIA
L,2,6,153,R RADIAL,2,8,153,T
 RADIAL,1,7,154,B RADIAL,3,7
,154,B RAD,4,7,154
970 REM ----ORNAMENTS----
980 DATA OUTSIDE PLUM,7,9,10
4,BELL,20,2,107,PLUM,16,6,11
2,DIAMOND,17,8,121,BELL,16,1
0,115
990 DATA DIAM,11,7,113,PLUM,
12,9,128,DIAMOND,14,4,129,BE
LL,17,3,131,PLUM,12,3,107
1000 DATA PLUM,10,6,120,DIAM
,9,8,129,BEL,13,5,123,BELL,1
5,9,131,PLUM,20,11,104
1010 REM ----DECORATION----
1020 DATA D,6,18,34,D,6,19,3
3,D,6,20,101,D,6,21,34,D,6,2
2,33,D,6,23,101
1030 DATA D,6,24,34,D,6,25,3
3,D,6,26,101,D,6,27,34,D,7,2
7,33,D,8,27,101,D,9,27,34
1040 DATA D,10,27,33,D,11,27
,101,D,12,27,34,D,13,27,33,D
,13,26,101,D,13,25,34,D,13,2
4,33
1050 DATA D,13,23,101,D,13,2
2,34,D,13,21,33,D,13,20,101,
D,13,19,34,D,13,18,33
1060 DATA D,12,18,34,D,11,18
,101,D,10,18,33,D,9,18,34,D,
8,18,101,D,7,18,33
1070 REM ----MESSAGE----
1080 DATA M,8,20,35,M,8,21,3
6,M,8,22,37,M,8,23,38,M,8,24
,39,M,8,25,40
1090 DATA M,9,20,41,M,9,21,4
2,M,9,22,43,M,9,23,44,M,9,24
,45,M,9,25,46
1100 DATA M,10,20,48,M,10,21
,49,M,10,22,50,M,10,23,51,M,
10,24,51,M,10,25,51
1110 DATA M,11,20,52,M,11,21
,53,M,11,22,54,M,11,23,55,M,
11,24,56,M,11,25,57
1120 REM SCREEN LOCATION LOO
P
1130 HOWMANY=148
1140 RESTORE 820
1150 FOR CHARACTER=1 TO HOWM
ANY
1160 READ IDENTIFICATION$,RO
W,COLUMN,CHARACTERNUMBER
1170 CALL HCHAR(ROW,COLUMN,C
HARACTERNUMBER)
1180 NEXT CHARACTER
1190 RESTORE 1330
1200 DIM X(67),Y(67),Z(67)
1210 FOR I=1 TO 67
1220 READ X(I)
1230 NEXT I
1240 FOR I=1 TO 67
1250 READ Y(I)
1260 NEXT I
1270 FOR I=1 TO 67
1280 READ Z(I)
1290 NEXT I
1300 FOR I=1 TO 67
1310 CALL SOUND(X(I),Y(I),2,
Z(I),6)
1320 NEXT I
1330 DATA 250,250,250,250,25
0,250,250,250,250,250,250,25
0,250,250,250,250,250,250,25
0,250,250,250
1340 DATA 250,250,250,250,25
0,250,250,250,250,250,250,25
0,250,250,250,250,250,250,25
0,250,250,250
1350 DATA 500,250,250,250,25
0,250,250,500,125,125,250,50
0,250,250,250,250,250,250,25
0,250,250,250,900
1360 DATA 330,330,494,494,44
0,392,370,330,234,330,370,39
2,440,494,330,330,494,494,44
0,392,370,330
```

```
1370 DATA 294,330,370,392,44
0,494,494,523,440,494,523,58
7,659,494,440,392,330,370,39
2,440,392,440
1380 DATA 494,523,494,494,44
0,392,370,330,392,370,330,44
0,392,440,494,523,587,659,49
4,440,392,370,330
1390 DATA 40000,165,165,165,
165,165,123,131,196,165,165,
165,165,123,40000,165,165,16
5,165,165,123,131,196
1400 DATA 165,165,165,165,12
3,208,220,185,196,196,123,13
1,147,147,165,165,165,165,18
5,40000,40000,196,196
1410 DATA 196,196,185,165,40
000,165,196,185,40000,185,16
5,185,196,196,196,196,147,14
7,156,156,165
1420 A$="FROM"
1430 V=16
1440 W=20
1450 GOSUB 1570
1460 A$="MAURICE R RYMILL"
1470 V=18
1480 W=14
1490 GOSUB 1570
1500 A$="BOURNVILLE"
1510 V=20
1520 W=17
1530 GOSUB 1570
1540 CALL KEY(0,K,S)
1550 IF S<1 THEN 1540
1560 END
1570 FOR I=1 TO LEN(A$)
1580 CALL HCHAR(V,W+I,ASC(SE
G$(A$,I,1)))
1590 NEXT I
1600 RETURN
```

==========

From DARTMOUTH TI USERS
Sprite One-Liner
by Barron Bartlett

Want to frustrate and amaze your Atari, VIC-20, and
Color Computer friends? Just type in the following
in the command mode with Extended Basic.

```
CALL CLEAR :: CALL SCREEN(5) :: CALL MAGNIFY(2) ::
FOR I = 1 TO 28 :: CALL SPRITE(#I,64+I,16,80,3*I,8) ::
NEXT I :: FOR J=1 TO 5000 :: NEXT J
```

Hit ENTER and watch all 28 sprites do their tricks.
If you want to see it again, simply hit FCTN REDO,
then ENTER again as many times as you wish.

Little Bits
This little ditty came from
MICROpendium, by Gary Guibor of Miami.

```
10  FOR C=1 TO 100
20  IF SQR(C)<>INT(SQR(C) THEN 40
30  PRINT C
40  NEXT C
```

The 99/4A produces 10 perfect squares while the TRS-80
model II, Apple, and Pet computers found only 6.
Microsoft Basic, and most CP/Ms discovered 9.
Other than the TI, only Basic Plus on a $100,000
minicomputer found all 10.

==========

FROM L.A. TOPICS JUNE 1987 - - AMENDMENTS TO TI WRITER
Here is a fix to one of TI-Writer's "features" (a.k.a. "bug):
Using a sector editor program (i.e. Advanced Diagnostics),
edit sector 30 (31 if counting from 1) of the FORMA1 file
(or the FORMAT file with My-Word). At byte 58(>3A),
change >A067 to >A002. This will stop the Formatter
automatically putting two spaces after a period.

There are other changes I have made to the Formatter as well.
I have changed the @ and & (bold and underline) to ` and
\ (FCTN C and FCTN Z), and I changed the * (for Mail Merge)
to a | (FCTN A). The reason I changed the last is that, in
a document if you have an asterisk followed by two or more
numbers, such as *11, or *256, the Formatter will drop
the * and the two digits immediately after it.
That's what happened a few places in Tom Freeman's catalog
program. Anyway, the changes are all in the first sector
of the FORMA1 file. For the @ and & fix, change the
@ and & at byte 115 (>73) to ` (FCTN C) and \ (FCTN Z),
or whatever you want them to be.
To change the * to |, change the * at byte 112 (>70) to |
(FCTN A), or anything else.
Rick Cosmano of the Southern California Computer Group
(SCCG) is to be credited for the @ and & fix, the other
two are mine.   MIKE DODD.

==========

GRAPH PLOTTING ON THE TI

BY JOHN STOCKS

When I first bought the TI I was surprised and disappointed to discover how primitive were the graphics facilities - surprised because I had expected the TI to be at least as good as the Spectrum and Commodore, and disappointed because I had hoped to use the TI for demonstration purposes in my work. The rudimentary block graphics were of no use whatsoever for scientific purposes, but this may well have been a blessing in disguise. In facing the challenge of forcing the TI to give me the performance I need, I have learned some invaluable lessons about programming!

The first step was to harness the facility for defining graphic characters as sixteen-digit string functions, the characters being formulated so as to produce the required x/y graph. This was easily extended to produce polar graphs, but an added complication arose here from the possibility of a polar trace crossing over itself. This meant that the string function for each character had to be stored for recall when required so that extra points could be added. Unfortunately, the limited available number of user-defined characters meant that the more complex polar functions could not be completed. Extended Basic conveniently incorporates a facility for character string recall, but the number of characters available is even smaller. I subsequently discovered that a similar x/y plotting program had already been published by Peter Brooks, but the polar version is unique as far as I know.

Both of the above programs were very satisfying to write and quite fascinating to watch as they laboriously plotted their curves block by block. However, they were not really what I needed. If you want to show the effect of varying a particular parameter, you cannot wait ten minutes for each variation to become visible! My next move therefore was to buy a Minimemory, and the impressive "LINES" program immediately showed that I was on the right track. Several sub-programs are possible for plotting each graph point from its x/y coordinates and simple graphs were produced with very satisfactory speed. I soon discovered however that the plots became disappointingly sluggish as soon as any of the GROM routines were called, particularly those involving trig functions. (Some 3D plots requiring three or more trig values per point were almost as slow as my earlier Basic efforts!) Stephen Shaw has pointed out that this is the price we pay for 16-bit accuracy and has suggested the use of abbreviated series and/or look-up tables for faster response. With a vertical resolution of only 192 elements, high accuracy is obviously unnecessary, so I made a table of sine values covering the quadrant 0 to 90 degrees in

steps of 1/200 radian and stored the results at 200 times their correct value, which meant that each could be held in a single byte of memory as an integer. (It occurred to me later that I could have achieved slightly better results by using increments of 1/250 radian and storing each figure as an integer at 250 times its correct value which could still have been held in a single byte, but you cannot think of everything when starting off!)

Having stored a series of sine values in this manner, cosines and the sines and cosines of larger angles can be looked up by making a suitable choice of sign and angle values, the table figures being universally applicable.

The other factor contributing to low running speed is the use of the cumbersome floating-point routines. These were avoided in the table by the multiplication factor of 200, and since the smallest y value required for graph plotting is 1, there is no need to involve them. Remember that when multiplying and dividing in Machine Code, very large numbers of up to 32-bits can be handled, so by operating at some large multiple of the required value, adequate accuracy can be achieved without the use of floating-point. Be careful with negative numbers though. Twos complement negatives are NOT recognised in 32-bit numbers, so all values must be treated as positives and a separate record kept of their signs.

Another hint for extra speed is to set the Workspace Pointer at >8300 instead of the usual >70B8. The region >8300->83FF appears to be unused so long as the GROM routines are not called, so locating the registers here gives a useful improvement. Unlike >70B8 however, the register values are lost when a program terminates and cannot be inspected by Easybug, so it is probably better to revert to >70B8 for program development and trouble shooting.

A GROM routine that you may need is that giving square roots. To avoid calling this, I developed a simple iterative program which automatically terminates when it reaches the necessary 1 in 200 accuracy. I imagine that similar limited-resolution routines can be produced for other mathematical functions, although I have not yet required them.

Results are pleasingly fast, three dimensional plots of considerable complexity being produced in a few seconds. My first attempt at animation, a teaching aid to demonstrate radio-frequency standing-waves, is too slow for smooth movement but it serves its purpose with admirable clarity. Animation requires that each phase of the graph must be erased before the next is plotted. The values of each ordinate must therefore be stored and a modified version of the "plot" sub-routine used to carry out the erasure.

I have not given any subroutine listings here as they occupy an awful lot of space, but I shall be glad to pass them on for an SAE. In fact, I should be happy to compare notes with anybody working in this fascinating field, as practical interest appears to be non-existent!
Address : 11 Stonehill Road, Roxwell, CHELMSFORD, CM1 4PF.

Thanks, John for such an informative article. I've reprinted part of John's letter below, as I know it will be of interest to some of you. Ed.

My motive in submitting is the hope of finding somebody with whom I can compare notes. Ever since I bought the Mini Memory two years ago, I have been trying to find somebody who is actually using it to write programs, but without success. What on earth do people do with their equipment?

Incidentally, it has been prepared with a word processor program in Basic for Mini Memory. Clumsy compared with the "real thing" of course, but it has full editing, correction, and formatting facilities and is a real blessing for a two-fingered typist like myself. Listings gladly supplied to anyone interested.
All good wishes. Sincerely,

John Stokes

==========

The next TI*MES will be printed early in January, so please send any contributions to me by December 20 for inclusion in that issue. My address is the same as for the advertisements on the last page.
Thanks, Christina.

==========

THE OLD ONELINE
by John Witte
This month's "tinygram" comes from the Greater Omaha TI User Group's newsletter. In XB it meets the requirement of being displayed on one screen only a "and not a bad little game to boot."

```
1 CALL CLEAR :: CALL COLOR(10
,16,7,2,11,11):: B,Q=0 :: W=I
NT(24*RND)+1 :: FOR T=1 TO 7
:: CALL VCHAR(5-(T>2)-(T=3),T
+12,42,VAL(SEG$("1353352",T,1
)))):: NEXT T
2 DISPLAY AT(20,2):"GUESS?":"
#1=";R(1):"#2=";R(0)
3 FOR X=1 TO 6 :: FOR Y=15 TO
19 :: CALL SOUND(1,+5,0)
4 IF Q=0 THEN P=1+(P=1):: ACC
EPT AT(22-P,8)BEEP VALIDATE("
123456789"):Q :: B=B+Q
5 CALL HCHAR(6-X,Y,111):: IF
B>W THEN 7 ELSE Q=Q-1
6 NEXT Y :: NEXT X
7 DISPLAY AT(21-(P=1),8):"WIN
S!" :: R((P=1)+1)=R((P=1)+1)+
1 :: FOR J=5 TO 12 :: CALL SO
UND(600,440-11*J,0):: CALL HC
HAR(16-J,13,,111,7):: NEXT J
:: GOTO 1
```

CLASSIFIED ADVERTISEMENTS - CLASSIFIED ADVERTISEMENTS

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.

* * * * * * * * * * * * * * *

WANTED - 32K Exp Card
FOR SALE - Stand-Alone 32K Exp - Offers, or anybody interested in a swap? (Tel. Derby(0332)772612)

* * * * * * * * * * * * * * *

your UNWANTEDS

FOR SALE - MODULES: Music Maker £9; Personal Record Keeping £9; Parsec £5; Alpiner £8.
CASSETTES: Battleships £1; 3D Race £1; Kong £1; Runner on Treiton £1; Othello £1; Ariel(Maze Game) £1; Haunted House/Wumpus £1.25; Treasure Quest/Four-in-a-row £1.25.
CASSETTES(EXT.BASIC): Attack Man £1.25; Super Frogger £1.25; Froggie £1.25.
CASSETTES(ADVENTURE MODULE): Mission Impossible, Voodoo castle, The Count, Pirate, Golden Voyage, Strange Odyessey. ALL £1 each, or £5 for 6.
BOOK: Get More From The TI99/4a by Garry Marshall £2. Please telephone 021-458-4970.

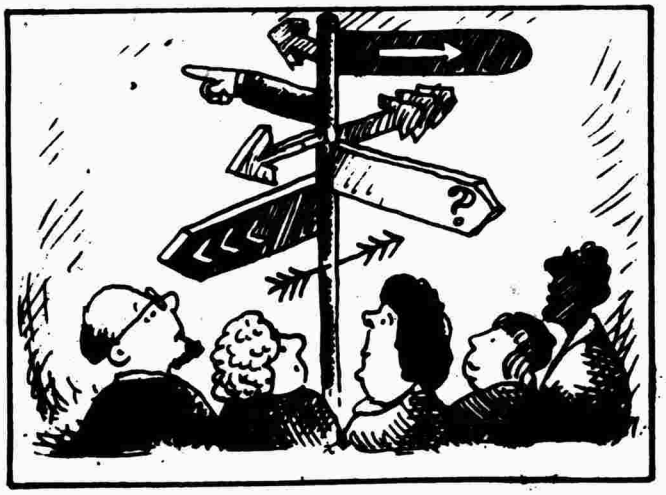
* *

FOR SALE: TI99/4a computer (with psu, modulator and manuals), 2 joysticks (later type), speech unit (with instr), 2 cassette leads, Modules: ExBas (boxed + instr), Parsec, Alpiner, 2 Poker, 1 Adventure. Also book of programs and ~20 games on tape. £100 (plus postage) the lot. Contact: Dave Hewitt, 22 The Meadway, Hoddesdon, Herts. Tel: 0992 463107

* *

FREE ADS FOR CLUB MEMBERS! Advertise on this page. Send ads to: TI99/4a USERS GROUP(UK), 4, Thornton Lane, ULCEBY, South Humberside DN39 6SR for inclusion in next TI*MES.

* *



SIGN OF THE TIMES?