042
8705

TIsHUG NEWS DIGEST

TIsHUG

Happy 6th Birthday

COMMITTEE MEMBERS:

Co-Ordinator:
Chris Buttner..Tel.(02)8717753
Secretary:
Terry Phillips.Tel.(02)7976313
Treasurer:
Bert Thomas....Tel.(047)541535
Publications:
Bob Montgomery.Tel.(042)286463
Sysop:
Ross Mudie.....Tel.(02)4562122
Merchandising:
Cyril Bohlsen..Tel.(02)6395847
Technical:
John Paine.....Tel.(02)6256318
Librarian:
Terry Phillips.Tel.(02)7976313

REGIONAL COMMITTEE MEMBERS:

Glebe:
Mike Slattery..Tel.(02)6920559
Penrith:
John Paine.....Tel.(02)6256318
Central Coast:
Russell Welham.Tel.(043)924000
Liverpool:
Arto Heino.....Tel.(02)6038956
Illawarra:
Rolf Schreiber.Tel.(042)842980
Bankstown:
Peter Pederson.Tel.(02)7722396
Carlingford:
Chris Buttner..Tel.(02)8717753
Sutherland:
Peter Young....Tel.(02)5288775
Manly Warringah:
Dennis Norman..Tel.(02)4523920
Coffs Harbour:
Keir Wells.....Tel.(066)551487
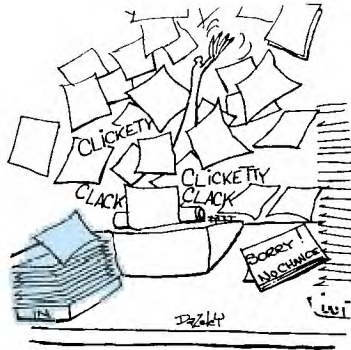
MEMBERSHIP AND SUBSCRIPTIONS:

Joining Fee............$ 8.00
Annual Family Dues......$25.00
Dues O'seas Airmail...US$30.00
Publications Library....$ 5.00
Texpac BBS.............$ 5.00
BBS Membership:
Other TI User Group ·
Members................$10.00
Public Access..........$25.00

GROUP GENERAL MEETING:

First Saturday of each Month at
Woodstock Community Centre,
Church Street Burwood. Starts
 1:00pm

COMMITTEE MEETiNGS:

Before the main monthly
meeting at Woodstock. Starting
it 12:30 pm.

CONTENTS

*   *   *   *   *   *   *   *   *   *   *

This issue of TND centres mainly around the regular
articles. A number of contributions from the
prolific pen of John Paine have been held over till
the next issue. Apart from the regular contributers
there are few local articles coming in. Surely, you
are doing something that may be of interest to others.

As the magazine was being put together, a note was
received from Peter Schubert on a stand-alone disk
contoller board. Those persons who still only have a
cassette based system may find use for one of these.
This is a major breakthrough.

Cheers for now,

Bob Montgomery

CO-ORDINATOR'S

REPORT

with CHRIS BUTTNER

I want to extend a very hearty welcome to all those new members - local and interstate - who have joined the club in the past few months. I hope you will enjoy your time with us and take the time to make some form of contribution to club activity particularly meetings or through the magazine.

All of you who sent in your renewals on time have helped reduce the continual updating process which has been a feature of the secretarial workload from the start. This is the first full year when we have had a common membership date. Although it has been a bit or a rush this month and last, I am confident it will allow us greater scope to get on with the other important work.

One of the problems we have to settle at times like this is the number of copies of the magazine we have printed in order to cater for latecomers. If you are one, this is the last copy of the magazine you will receive UNLESS YOU RENEW YOUR SUBSCRIPTION NOW! Don't put it off. Send your cheque or charge card authority to our Secretary, Terry, at P.O. Box 214, Redfern.

At our last committee meeting it was decided to have the magazine cover reflect the many areas of membership. Our Editor is now ready to accept your artwork. It need not be anything flash - children's line drawings are quite acceptable so we are looking forward to big things from our members on the Central and Northern (Banana) Coast and Western regions. I hope you like the results.

Many of you will also have good reason to celebrate this month when you receive your Ramdisk Card. This project has been quite demanding on John and his willing band of helpers. A lot of hard work has gone into presenting a quality circuit board, software and instructions. If you have Assembly Language skills please join in the spirit of the project by authoring useful utilities. There is every reason to believe this project is no different to the 99/4A - time alone will show the true worth, versatility and usefulness of the product.

Our planning was initially to provide two tutorial days this year: one in June and the other in November. The response to topics YOU want discussed has been quite disappointing. If there is some aspect of programming or application you want covered, send a note to Terry urgently so we can do justice to the topic.

There is now a very good selection of books and other club magazines in the Library. Russell is doing an excellent job of continuing the good work started by Brian Graham. If you have not yet seen the extent of the library, catch-up with Russell at the next meeting.

Our club BBS now has a substantial number of interstate members. If you are interested in "corresponding" with someone (WA, NT, or where-ever) leave a message for SYSOP on the BBS. I am sure Ross will be only too happy to provide you with usernames to get you started.

May has traditionally been our birthday month. To mark the occasion there will be a lunchtime barbecue at Woodstock Community Centre. Food will be provided so come prepared to enjoy yourself. I hope to meet as many of you as possible. If you are a country member please take the time to make yourself known to all committee members.

o

for THIS teacher

by Phil Thomson

(CONVERSATION NO.1)

APPLE, BBC, IBM, MICROBEE. "Well, which do you think?"
"..um"
"How about a Microbee, you get two disk drives and lots of free software?"
"Sounds O.K. but do you think we could get disk drives for our T.I.'s?"
"Says here the offer only applies to these ones on Gov. contract...,sorry"
"Oh well, I guess we may as well try a Microbee.Pity."

* * * * *

It was a case of GOOD NEWS/BAD NEWS.The good news was that our school had been allocated a $1350 grant for the purchase of a computer or computer equipment. The bad news was that I was unable to use it to upgrade the 4 TI-99/4A computers that the school had bought over the last 12 months.
Here is where the story started.
Some 8 or 9 months ago I was posted at a school in Corrimal. The school had no computers at that time and the children and staff alike were completely in the dark as far as computers were concerned.
It was with great interest and excitement then that my boss discovered me using a computer with my 4th class pupils.
This involved me in taking my computer from home to school each day (a practice which didn't please my own three children). I wrote some computing activities into my program which were designed to give pupils some hands-on activity using such programs as :

    MYSTERY SPELL
    READING ON
    MATH MAN
    ALIEN ADDITION
    DIVISION 1   etc.
and before long the school was buzzing with talk of computers. Children from other classes became permanently encamped in my room at lunchtime and lot's of children decided they wanted a computer for Xmas. Children found the TI-44/9A easy to operate and in a matter of weeks some children were able to set-up the computer, load programs, and trouble shoot problems. Our computers were and still are cassette driven. Most of you would not be aware of nature of school cassette recorders.We're talking stone-age here. At least 8yrs old, broken cables, handles falling off, faulty tone controls,heads filthy, repaired at least 3 times.....well, maybe you're starting to get the picture. Anyway...

* * * * *

(CONVERSATION NO.2)
"I've had a few parents up here keen to talk about the computing your class is doing"
"And...."
"Well they seem very keen for the school to develop some kind of computer education."
"Mmmm..." I thought, nows the time to ask. "Look, if I were able to get my hands on a second-hand computer do you think the school could afford to buy one?"
"How much would it cost?"
"$100"
"Could you get three?"   * * * * *

And so it all began...3 computers for less than the cost of one (of the contract jobs) all with new colour monitors, endless supply of free software and servicing from the local user group (not that we've needed any).

* Secretary's Notebook *

Hello again. I hope everyone had a safe and happy Easter break.

At the committee meeting in March it was decided to hold a tutorial workshop as the theme for the June meeting. Further details on this should be appearing elsewhere in this issue. If you can assist in any way in the tutorial please get in touch with a committee member to volunteer your services. The more helpers on the day the better it will be.

A big welcome to the following new members who joined us since last month:

Peter Hogan - Bray Park Qld.
Josephine Jackson - Cattai NSW
Frank Hall - Kendall NSW
Bret Hales - Roseville NSW
Mark Williams - Valley Heights NSW

I hope you all enjoy your association with TIsHUG, and I hope to see you at our meetings some time.

For the information of all members, this was the situation when I closed off to do the mailing for the April TND:

TOTAL ENTRIES IN DATA BASE - 467

Make up of entries:

Complimentaries (Exchanges) - 10
Life members   -   3
Paid up to 30/4/88   - 102
Paid up to 31/3/88   -   1
Paid up to 31/12/87   -   1
Paid up to 30/6/87   -   1

Due to Renew 30/4/87   - 295

Expired on 31/3/87   - 21
Expired on 28/2/87   - 33

TOTAL   - 467

BBS MEMBERSHIP -

TOTAL ENTRIES IN DATA BASE - 122

Make up of entries:

Life members   -   3
Paid up to 30/4/88   - 40

Due to Renew 30/4/87   - 70

Expired on 31/2/87   -   5
Expired on 28/2/87   -   4.

TOTAL   - 121

PUBLICATIONS LIBRARY -

TOTAL ENTRIES IN DATA BASE - 78

Make up of entries:

Life Member   -   1
Paid up to 30/4/88   - 32

Due to renew 30/4/87   - 45

TOTAL   - 78

I guess that isn't too bad up to this point. I hope that the bulk of members due to renew on 30/4/87 stay with us and that I can give you a much better picture next month.

I received a note from the Banana Coast Regional Group advising of their last meeting details. In all 7 members turned up, so that's a good start. Keep up the good work Kevin and Keir. A full reproduction of their letter appears in the regional reports. By the way, if any Regional Group Co-Ordinators want some software to demonstrate at their meetings then please get in touch with me. I'm only too happy to make anything available that you want.

Here's a few tips to finish up this column. These are courtesy of San Diego Computer Society's newsletter. I haven't tried these so I can't guarantee they work.

FORTH through MINI-MEM - Yes, you can load FORTH using MINI-MEM. Use Option 3 "MINIMEM" then Option 1 "LOAD & RUN"

Corcomp says you can't load their manager from within a program. Here's how you can -

100 CALL INIT
110 DELETE "LD-CMDS"
120 CALL LINK("MGR")

Want to design a T-Shirt. Make your design with one of the graphics programs, then output it to the printer, but print it on transfer paper. Then with hot iron, transfer it to your T-Shirt.

I like that last one. Is someone prepared to have a go and bring their results along to a meeting to show us all?

If someone knows the present address for member Ed Zwerus could they please let me know. His newsdigests are being returned to me marked "No longer at this address".   O

BUT NOW FOR THE GOOD NEWS . . .

6TH BIRTHDAY PARTY
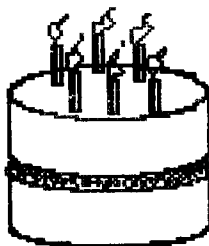
DATE: 2/5/87

TIME: 1:00 PM

WHAT TO BRING:

YOURSELF AND FAMILY

BBQ AND DRINK PROVIDED

TIsHUG SHOP   MAY 1987

Very little has changed in the list of items for sale in the shop from last month.

RAM DISK CARD STATUS REPORT.

Orders have now closed for the RAM DISK PCB .
Over sixty of these boards have been sold.
If there is any over run in the production of these cards we will purchase the complete run and put them on sale, if not and you haven't got your order in, it is too late, YOU have missed out.!!!!

ORDERS ARE STILL BEING TAKEN for the other components:-
          6264LP-15 Ram chips...........$ 5.50 each
          Auxiliary chip and socket set..$25.00 per set
The above items require full payment with order.

I have received about twenty orders for the sets of RAM CHIPS and AUXILIARY components.

Robert Peverill is taking orders for the supply of the three (3) AAA size batteries and battery holders required for the Ram disk card.
The price for these items will be:-
          $10.50 for three standard batteries.
          $ 1.77 for three battery holders.

     OR $12.00 for three batteries as a single unit
          heat shrink wrapped and with solder tags.

All the above items ordered should be available at the SHOP for the MAY general meeting.
PLEASE BRING YOUR RECEIPTS WHEN YOU COME TO PICK UP YOUR ITEMS.

If you cannot attend the meeting and would like your items posted out, please let me know.
Postage and Packing will be an extra, approx $2.05 per unit posted Intrastate or $2.50 Interstate.

SHOP INVENTORY.
This is still similar to last months, with the exception of BLANK DISKS which are on special for the month of MAY only. (why not give your Mother a box on Mother's day) Price......$17.00 per box.!!!!

HARDWARE
     Spike Protectors...................$29.00
     Consoles Ver.2-2...................$65.00
     1 only Double Disk Drive Cable Set..$30.00
     32k Hatchbox Memory Expansion $55.00
          (All these units are fully
          assembled and include a $10.00
          deposit on the GROM PORT.)
          Please note that there are only
          a couple of these left for sale.
     T.I. Joystick handles..............$00.50

SECOND HAND ITEMS
     Keyboards..........................$15.00
     Power Supply Boards................$15.00
     Grom Ports.........................$12.00
     Ivory Console Cases................$ 2.00
     Chips to convert your 1983 console
               to a 1981 version.........$12.00

BOOKS
     Millers Graphics Smart Prog.Guide...$ 7.50
     Technical manual....................$15.00
     Micropendiums.......................$ 2.90
        1986-June to Dec./1987-Jan.to Mar.

SOFTWARE
     Console Writer Modules.............$30.00
     Auto Spell-Check (T.I.Writer).......$30.00
     Club Software Tapes................$ 3.00
     Club Software Disks................$ 5.00
     Boxes of Blank Disks (10)..........$17.00

People with outstanding orders for 32k memory expansion units should contact me with instructions as to how they would like the units dispatched.
     o

ADVENTURE HINTs

MYSTERY FUNHOUSE

Part One

Time for a little spy stuff! A Fun House may seem a strange place for espionage shenanigans, but secret agents have a habit of popping up in the darndest places!

So, here you are, standing, shoes in hand, outside the Fun House. Now, if you just had some money, you could get inside. The first thing to do is drop the watch, because you won't be needing it. Now go East to the parking lot. Ignore the five dollar bill (it's a bill for $5!), and look at the tree.

Get the branch, then look in the grate. Aha, that's where the real money is! Chew the gum (tastes HORRIBLE!) then stick it on the branch. Use the branch to get the coin, then drop both the branch and the gum. Return to the Fun House entrance.

Wear the shoes, then give the dollar for a ticket. Go Fun. You are standing in the Magical Mirror Room, and just up ahead is (gasp!) a maze. Fortunately, it's not a bad one, if you're careful (if you aren't careful, well....).

Go North three times, then West twice. This brings you to a small room. Go West from here to the room with the knobs. Pull the green knob, and you will find yourself in another room. Get the trampoline that's here, then go South to the shooting gallery and pick up the strange spectacles. You have a few minutes, so, if you want to, you can amuse yourself by shooting a few clay pigeons. When you're finished, go back North, then Up, to the knob room.

Now, as you proceed through the adventure, you will from time to time get messages about your shoe heel being loose. Ignore them, and don't touch the heel for now. There will be time for that later. At the moment, you have other things to do. So, head along West to the tank room, and from there, Up to the ledge over the pit. Drop the trampoline here, then go East to the barrel room.

That's all the clues for now! Part 2 of Mystery Funhouse will sneak its way into the magazine next month.

*   *   *

IT'S TIME

Please take the time to look at the mailing label for this month's TND.
If the date shown is 30th April 1987 then your membership has expired and this will be the LAST issue of the TIsHUG NEWS DIGEST mailed to you, unless you renew! Do it NOW, since early renewals will enable us to plan the printing of the appropriate number of copies for the rest of the year.
Also, more importantly, it will avoid the disappointment of missing an issue of the TND.

RENEW NOW

TECHO TIME

with
JOHN PRIME
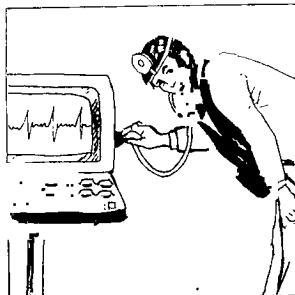
A number of members have asked me how I became involved in hardware repair for the TI Home computer. Actually although I have had formal technical training many years ago, I really had no intention to get involved with being a "TECHO" until I was forced into doing a spot of work on my CorComp Micro Expansion box after Mother Nature decided to scare the hell out of me by dumping a lightning bolt in close proximity to my house.

This lightning bolt creamed the expansion system and left me with no alternative but to either sent my system back to America or do something about it myself.

I figured that seeing that I used to be an expert on something years ago, I should be able to at least make a "simple" expansion system work again. Eventually I did get the system going and full of confidence I made the rash decision to tell the local CorComp Distributer about my "expert ability". Before I knew what had happened, four disk controller cards, a couple of Micro expansion systems and the odd memory card had suddenly piled up and were demanding attention. A crash "bone up" of all the technical manuals, consultations with our local Technical people and the occasional bit of good fortune, and suddenly these devices were fixed. The next stage was to attempt to unravel the mysteries of the consoles and in a short time I had started to have limited success with repairs.

While this was happening I acquired some test equipment and a MILLERS GRAPHICS GRAM KRACKER. You may wonder why the emphasis on the GK, well that device helped me to make quantum leaps in understanding the operating system of the TI. In the meantime I just plodded along trying to fix consoles, with a little success, when suddenly a fellow club member, Geoff Trott came up with his little magic console tester. The average time for diagnosis of a console varied between 10 minutes and 10 months in pre console tester days. Today this average is now between 3 minutes and two hours.

I wish now to put our committee on notice that when the time comes to review our awards for achievements in hardware, Geoff Trott will be my first nomination.

Anyway back to the subject, the next major breakthrough was the release of the Mechatronics Eprom Programmer. With this tool I was then able to routinely make up all the necessary eproms needed to fix about 50% of the dead consoles that fell into my hands. As a matter of interest, generally speaking over 80% of the consoles I have looked at in the last six months were repaired by replacing a faulty rom or VDP choke as mentioned in last months article in FAULT OF THE MONTH. The other 20% of faults varied from faulty VPD ram chips, microprocessors and the occasional logic gate or buffer chip.

As well as the Console Tester and Eprom Burner, I have the mandatory logic probe, multimeter, oscilloscope and a collection of Diagnostic routines that have been tailored to my needs.

Carrying on now to the feature of this column, namely Fault of the Month, I would like to thank those members who responded to my question as to whether this type of information should continue. The response so far has been reasonably positive so here goes with some typical problems encountered.

FAULT OF THE MONTH

Perhaps it may be appropriate to change the name of this feature to FAULTS OF THE MONTH because I have documented so many in the last couple of months, I think it is in all our interests to share this information.

I made mention in the last article of the apparent lack of quality control in the last days of production of the 99/4a. The fault I will describe next is confirmation of this problem.

Recently the club purchased a number of NEW consoles from a major retailer, who had found stock hidden away in his warehouse. Ben Takach bought one of these consoles as a backup unit last year, but had not powered it up until last month. To his horror he was greeted with a black screen but the system seemed ok otherwise. On investigation the computer worked ok but there was no Video screen...just a blank screen.

The fix was quite simple, a couple of transistors that provided the video interface between the modulator and VDP chip needed replacing. The major concern was that these transistors were shot in the first place.

Why were they both short circuited? The actual reason was quite simple, they were inserted on the board back to front and just blew up when first powered up. Now you may ask "How come they were back to front". The only explanation I can offer is that they were inserted that way by TI. Remember that this was a "brand new" console from a major retailer and was still sealed in the original plastic wrapper. This CONSOLE could never have been tested at the factory because it would have failed on the spot. I was under the impression that ALL consoles were burnt in prior to shipping, but now I believe that this could not have been the case with this one. I wonder how many more were shipped without testing? I would like to hear from fellow users who had to return their computers under warranty especially if they have serial numbers or date code information from their original receipts.

The second item for this month concerns a couple of faults found in CorComp disk controller cards which where not really as a result of poor quality control of the manufacturer but more a result of bad luck and possible bad implementation of design rules by certain semiconductor manufacturers.

The first CorComp card seemed to work for an hour or so then completely crash. Switch off for a while and it would then work again for a while and crash again. Use it in the summer months and it would crash in less time. When I questioned the owner he told me that it had been like that from new and he just put up with it until the frequency of failure increased. Obviously this appeared to be a heat related fault so I plugged the card into my PE box and let it run. It would not fail (MURPHY'S LAW). I must mention that my PE box does not have the original TI power supply fitted because of a catastrophic failure some months ago but that is another story. The unregulated dc level in my box is about 9v DC which is fairly close to TI spec. as distinct to the typical 16v DC that can be found in most boxes in Australia. Anyway after applying the good old hairdryer to the card, I found I could crash it in about 2 minutes. The fix was simple, the 2793 controller chip was heat sensitve and a replacement controller has now been running for about 6 months without any trouble. This again appears to be a case of a product that was sold without suitable burn-in but in defence of Corcomp I would like to make the point that the standard TI PE Box power supply as provided for Australian Users was not rated to the specifications laid out in the technical manual. The manual states that the supply voltages out of the power supply should be +8V, +16V and -16V dc. The primary voltage is rated for 220 Volts AC but Australian mains voltages can vary between 230 VAC to 260VAC which in turn makes the unregulated output voltages up around +16V , +24V and -24V.

Continued from P5

As I stated before, the voltages on my PE box are a lot lower than indicated above, and this is why the card in question did not fail because the heat buildup from the regulator on the card was insufficient to affect the controller chip. I would like to think that CorComp when testing the card before shipping had the same problem.

The second problem with a Corcomp card was certainly different in nature but similar in respect of the mortality of one chip. The card was dead, as far as disk functions were concerned but did appear to power up ok. A quick check with the oscilloscope showed that the 741s04 oscillator chip was dead. This chip is configured as a crystal oscillator and provides the necessary clock signal to the disk controller chip (in this case this was a later model card that used a 1773 disk chip). The crystal was an 8 Mhz crystal and when I replaced the 741s04 up it came and appeared to work ok. But.....the controller started to work and then died.On closer examination I found that the output voltage of the oscillator appeared a little low and did not fully reach the required TTL switching level necessary for the controller chip. So in went another 741s04....same result. So in went another....same result... I had then found I had run out of new chips so I scrounged an old chip off another piece of junk. In it went... controller worked ok.

This made me sit up and look at the data sheets on the 741s04 and the design of the circuit. The design of the circuit was straight off the standard TI Controller card and it would appear to sound. Why then did a whole series of new chips not work but an old second hand chip work flawlessly? The secret lay in who made the chips and the respective manufacturers data sheets.

The new chips happened to be Mitsubishi devices and the old second hand chip was a genuine TI device. At this stage I realised that I had never seen anything other than a TI 741s04 chip in a disk controller built by TI or CorComp for that matter. Anyway, after looking at the respective data sheets and seeing that both Mitsubishi and TI sheets said the same things, and then looking at the results on a CRO I came to the conclusion that the TI chip should not work properly either because its rated switching capability did not equate to it being reliable running at 8 Mhz. TI obviously manufactured their chips to a higher specification than stated in their data sheets but a number of other manufacturers built their chips to the standard. CorComp would obviously think that by using an established circuit design that had proved itself over a period of time, should not expect any trouble in that area......WRONG... They in fact were a victim of poor design by TI. It is interesting to note that the CorComp Micro system uses a Cmos 4069 at 8 Mhz and this problem does not occur.

For those that are interested, I checked a number of vendors product in this circuit and found that only a couple will work. They are TI, National, and some Motorola parts. Almost all Japanese devices i.e. Mitsubishi, Hitachi, Fujitsu, and NEC failed to work reliably.
That concludes this months article. Next month I will talk about the repairs on key modules such as Minimemory and Extended Basic.
o

STOP PRESS...STOP PRESS...STOP PRESS...STOP PRESS...

THE PROTOTYPE D/C BOARD IS NOW FULLY FUNCTIONAL AND I CAN NOW DESIGN THE FINAL PRODUCTION STAND-ALONE BOARD. A DESCRIPTION OF ITS CAPABILITIES AND AVAILABILITY WILL BE PUBLISHED IN NEXT MONTH'S TND. REGARDS, PETER SCHUBERT

Power Supply for Disk Drives.

The club shop recently acquired a number of Console Power boards for sale as spare parts. The primary use would be as a backup power board but these boards can fulfill a number of other functions.

The boards are fed with an AC source of between 13 to 18 volts and have a regulated 5 volt output at about 1 amp output, 12 volts at 300-500 mAmp output and a -5 volt output at 100-200 mAmp max.

One function that these boards will fulfill quite nicely is that of a Disk Drive Power supply with little or no modification. Modern slimline drives would be able to be powered directly from this board but a slight upgrade from the existing 78ml2 regulator to 7812 will enable the board to drive a standard TI full height drive. This upgrade will lift the +12 volt supply to be able to supply the necessary 600-800 mAmps needed.

All that is needed then is an AC source, eg 15 volt AC plugpack, disk drive and the power board from the shop and you have an inexpensive power supply for that drive two that you have always wanted.

One note of caution::: This power supply can only power one drive, not two. If you want to add more drives then you will have to use two power boards although if you use a transformer of sufficient rating the two boards can be powered from the one source.

Circuit diagrams of the power boards are available in the technical manual also supplied by the club shop or you may wish to borrow the manual from the club library.

Cheers for now,
John Paine.

* * *

REMEDY FOR ALPHALOCK

John Paine TIsHUG

In a recent edition of Micropendium, Mack McCormick described a simple modification to the keyboard interface that would allow you use your joysticks properly irrespective of the position of the ALPHALOCK key.
The ALPHALOCK and UP position of the joystick share the same input line to the TMS9901 I/O chip and as you are aware, when the ALPHALOCK key is down you cannot use the UP position of the Joystick.
The modification consists of simply wiring a small signal diode in series with ALPHALOCK key and cutting a track on the keyboard. The following sketch will show you where to insert the diode.
Locate pin 6 of keyboard connector, it will run via the printed circuit to one of two pins connected to the ALPHALOCK key. Cut the track to the A/L key that terminates to pin 6 and solder a diode (1N914, 1N4148 or 1N4001-4) in across the cut track making sure the cathode end (Banded End) goes to pin 6 of the connector and anode goes to the ALPHALOCK key.
After you have reassembled the base of the console, load your favorite game and play. You will note that the A/L key no longer disrupts the operation of the joysticks.
I have modified a number of keyboards with no troubles experienced.

```
                   KEYBOARD
        Connector
            1------------o
            2
            3                A/L Key
            4
         <to 5  cut track
  console 6----xxx-----o
               |   |           |
               |  -|<-----
               |   diode
              15
```
o
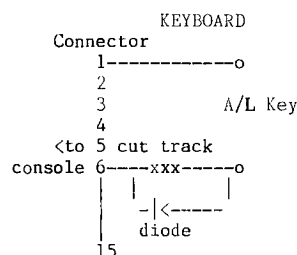
SOFTWARE

column by Terry

I hope you all liked last month's software offerings, particularly the South Pacific Album. I'm sure you will agree, a lot of work went into its programming.

This month, available at the shop, and by mail order, will be the following selection of software. As usual, first up the software to be released on disk.

1. DISK MANAGER 1000 - VERS. 3.5

Several enhancements have been added to the original and later versions by Ralph Romans. A document file on the disk will tell you all about the changes. Notable is the ability to dump a screen from the Utilities menu to either the screen or printer by selecting "T" or "P". I'm sure you will like this one. I have used Bruce Caron's original loader on the disk, so this version will boot up from Extended Basic. Also on this disk is a neat little cataloguer/lister program from our own Robert Peverill. Robert tells me he wrote this program to satisfy a need he had to overcome problems with his growing disk library and not knowing program start names, or whether a program was in XB, Basic or Assembly (Memory Image format). Full instructions are included, and the print-out obtained will tell you at a glance what is on the disk. Robert wrote the program in Basic, so it doesn't have any elaborate error checking routines, but never-the-less I feel it will be a valued addition to any software library.

2. DISKHACKER - from Will McGovern, Hunter Valley 99ers.

This is a great utility that you will enjoy playing around with. The disk contains 116 sectors of documentation, so a thorough reading is required before you get into it. Included on a the same disk is a revised memory image cassette loader and a cassete transfer program.

3. FUNNELWRITER VERSION 3.4 - also from the McGoverns of the HV 99ers is this latest version of the ever popular Funnelwriter. If you want the latest version then you will need to pick up this disk. PLEASE NOTE - at the shop it will only be available in either DOUBLE SIDED or DOUBLE DENSITY FORMATS. If you want it SSSD, then it will need 2 disks. Orders for this format will be taken at the shop for collection at the following meeting or delivery by post.

ON CASSETTE AND DISK.

1. ASSEMBLY GAMES - at least 3 but hopefully 5 games on this disk and cassette. A bit more testing is required yet on some programs to see if they load from cassette. Tape will be numbered 1987/5A. Definites are:

PIZZA - a fast paced action game where the object is to satisfy the ever hungry patrons at the counters who want a pizza. You must also satisfy the boss, for if 3 patrons miss out on a pizza then you're fired and have to start again. Keys 1, 2 and 3 move you from counter to counter, while key 9 serves the pizza. Be warned - you need to be quick with your fingers to survive at all in this game.

ANTEATER - a game that starts out simply but gets harder as you play. The idea is to pick up the food pieces and take them back to your nest, all the time avoiding the hungry anteaters. Once they get on your trail, they sense victory and are hard to shake off.

NIBBLER - move your ever growing snake around the maze picking up apples to score points. This is a very entertaining game and quite difficult to play as the snake is very hard to control, as it grows longer. Make sure your joysticks are in good order.

TAPE 1987/5 - will contain the following Basic and Extended Basic programs:-

BARTENDER - this is an old favourite of mine which appeared some years ago in the old 99'er Magazine. Newer members may not have heard of this magazine, but at one time it was the ONLY magazine worth reading for the TI (apart from the TND of course). Sadly, it has gone, but if you can get hold of any of the early copies they are well worth reading. Anyway, Bartender was originally coded in TI BASIC, but this version has been re-written in Extended Basic. If you like mixing cocktails, then this program will show you the ingredients for about 20 all time favourites, plus you can type in what liquor and mixers you have on hand and the program will tell you what you can make with them. Not only that, but you get a nice graphics presentation of each of the drinks.

BOMBER - this is another program that has been around for some time, but I can't recall it being released as club software. It's a fairly simple, yet absorbing game, where you are flying over a city, dropping bombs, yet all the while your plane is slowly descending. Obviously if you don't blow the tops off the buildings then you are going to crash. In Extended Basic.

BUBBLE - in Extended Basic with lots of sprites. The idea is simply to jump from bubble to bubble to get across the screen. Not as easy as it sounds.

METRIC CONVERTER - if you're like me, and still find it difficult to think in metrics then this program should be very handy. It will just about convert anything, from the most obscure to common measurements. In Extended Basic. Actually this is 2 programs, with part 2 containing even more conversions for you to fool around with.

CRAZYFRAZE - substitute PHRASE for FRAZE and you might get some idea of what this is about. This is a several player game. Player one types in a phrase, then subsequent players are given the option to change a word or two. Some crazy results ensue. The program runs in TI BASIC and has TE2 and printer options.

TERMINAL CHESS - a program from John Behnke which will enable you and a friend to have a game of chess over your modems. The program will run in either Basic or Extended Basic.

BRICKS - this is one of the simplest, yet most graphically pleasing drawing programs I have yet seen. You actually draw with a brick - you can change its colors - and you can get some amazing 3D perspectives. Great fun to doodle away a few hours with on the computer.

JUMPER - this is an Extended Basic version of Q-Bert and is one of the best ones I have yet played. The character moves very easily using just the E, X, I and M keys. You task is to avoid the "creeps" that float across the screen. Good fun to play because you can actually get well into the game with a bit of practice.

Well that's it for this month. I hope you enjoy all this software.

O

'No, you can't go out until the computer's done your work'.'

```
100 !@///////////////@!
110 !@ BOWLING CHAMP @!
120 !@  VERSION 1.2  @!
130 !@ X-BASIC 3.4 K @!
140 !@///////////////@!
150 CALL CLEAR :: CALL CHAR(
129,"000000FFFF0000000000080
CFE0C08"):: CALL CHAR(136,"F
FFFFFFFFFFFFFFFFFFE7BDA5A5BDE
7FF3C7EFFFFFFFF7E3C"):: CALL
COLOR(14,12,2)
160 RANDOMIZE :: CALL CHAR(1
39,"FFFF9F0F9FFFFFFFFFF9F0F
9FFFFFFF"):: DISPLAY AT(8,8)
:"BOWLING CHAMP!"
165 CALL CHAR(141,"E0E0E0E0E
0E0E0E00000FFFFFFFFFFFFFFFFFF
FFFFFF0000")
170 DISPLAY AT(22,1):"USE AR
ROW KEYS TO STEER BALL"
180 DISPLAY AT(13,2):"How ma
ny bowlers? (1-4): 1" :: ACC
EPT AT(13,27)VALIDATE("1234"
)BEEP SIZE(-1):B :: FOR A=1
TO B :: DISPLAY AT(15+A,3):"
Bowler ";STR$(A);"'s name:"
190 ACCEPT AT(15+A,20)BEEP S
IZE(7):A$ :: B$(A)=A$ :: N
EXT A :: CALL CLEAR :: DISPL
AY AT(1,9):"1 2 3 4 5 6 7 8
9 10" :: FOR C=2 TO 10 STEP
2 :: CALL HCHAR(C,11,129,22)
:: NEXT C
200 FOR A=1 TO B :: DISPLAY
AT(2*A+1,1):B$(A-1):: NEXT A
:: FOR A=11 TO 13 STEP 2 ::
DISPLAY AT(A,1):B$(A-11),B$
(A-10):: NEXT A :: CALL HCHA
R(10,1,129,32):: CALL HCHAR(
14,1,129,32)
210 CALL HCHAR(24,1,129,32):
: FOR A=0 TO B-1 :: D(A)=1 :
: NEXT A :: CALL HCHAR(15,1,
136,288)
220 CALL HCHAR(16,6,140):: C
ALL HCHAR(22,6,140):: CALL H
CHAR(17,6,139):: CALL HCHAR(
21,6,139):: CALL HCHAR(18,7,
140):: CALL HCHAR(20,7,140)
230 CALL HCHAR(19,7,139):: F
OR E=1 TO 10 :: FOR F=0 TO B
-1 :: CALL VCHAR(1,2,32,9)::
CALL HCHAR(F*2+3,2,130):: G
OSUB 1000
240 G=0 :: GOSUB 310 :: IF H
<>10 THEN G=1 :: GOSUB 320
250 IF E=10 THEN ON I GOTO 2
60,270,270,260,280
260 NEXT F :: NEXT E :: DISP
LAY AT(12,4):"PLAY AGAIN? (Y
OR N) Y" :: ACCEPT AT(12,25
)VALIDATE("YN")BEEP SIZE(-1)
:C$ :: IF C$="Y" THEN RUN EL
SE CALL CLEAR :: END
270 DISPLAY AT(12,5):"TAKE T
WO MORE BALLS!" :: FOR A=1 T
O 300 :: NEXT A :: DISPLAY A
T(12,1):"" :: D(F)=I-1 :: G=
1 :: GOSUB 310 :: IF C<>10 T
HEN 300 ELSE 290
280 DISPLAY AT(12,6):"TAKE O
NE MORE BALL" :: FOR A=1 TO
300 :: NEXT A :: DISPLAY AT(
12,1):""
290 D(F)=1 :: G=2 :: GOSUB 3
10 :: GOTO 260
300 D(F)=1 :: G=2 :: GOSUB 3
20 :: GOTO 260
```

```
310 FOR J=1 TO 10 :: READ K,
L :: CALL SOUND(20,-3,0):: C
ALL HCHAR(K,L,137):: NEXT J
:: RESTORE :: M=-1 :: H=0 ::
GOTO 330
320 M=0
330 GOSUB 350 :: N=O(F):: I=
D(F):: N=N+C :: ON D(F)GOSUB
520,540,560,580,600
340 O(F)=N :: D(F)=I :: DISP
LAY AT(13+(F<2)*2,26+(F/2=IN
T(F/2))*17)SIZE(3):USING "##
#":O(F):: RETURN
350 P=1 :: Q=23 :: R=-1 :: C
ALL SPRITE(#1,138,2,Q*8-7,8)
360 FOR II=113 TO 185 STEP R
ND*6+4 :: CALL KEY(0,S,T)::
CALL LOCATE(#1,II,8)
370 IF T<>0 THEN 390
380 NEXT II :: GOTO 360
390 HOOK=0
395 CALL SOUND(-4200,150,20,
-7,20)
400 FOR P=8 TO 192 STEP 4 ::
CALL LOCATE(#1,II,P)
410 IF HOOK=0 THEN CALL KEY(
0,KEY,ST):: IF KEY=69 THEN H
OOK=-.25-RND/2 ELSE IF KEY=8
8 THEN HOOK=.25+RND/2
420 IF II>112 AND II<186 THE
N II=II+HOOK
430 NEXT P
440 P=25 :: C=0 :: CALL POSI
TION(#1,ROW,COL):: Q=(ROW+7)
/8
450 CALL GCHAR(Q,P,U):: IF U
=137 THEN CALL HCHAR(Q,P,136
):: C=C+1 ELSE 490
460 CALL SOUND(-40,-6,0):: F
OR R=-1 TO 1 STEP 2 :: V=Q :
: W=P
470 V=V+R :: W=W+1 :: CALL G
CHAR(V,W,U):: IF U=137 AND R
ND<.9 THEN CALL HCHAR(V,W,13
6):: C=C+1 :: CALL SOUND(-40
,-6,0):: GOTO 470
480 NEXT R
490 CALL LOCATE(#1,Q*8-7,P*8
-7):: P=P+1 :: IF P<32 THEN
450
500 H=H+C :: X=C+48 :: IF H=
10 THEN IF M THEN X=88 ELSE
X=47
510 CALL HCHAR(2*F+3,7+2*E+G
+2,X):: FOR J=248 TO 8 STEP
-4 :: CALL LOCATE(#1,185,J):
: NEXT J :: RETURN
520 IF H=10 THEN IF M THEN I
=2 ELSE I=5
530 RETURN
540 N=N+C :: IF C=10 THEN I=
3 ELSE I=4
550 RETURN
560 N=N+C*2 :: IF C<>10 THEN
I=4
570 RETURN
580 N=N+C :: IF H=10 THEN I=
5 ELSE I=1
590 RETURN
600 N=N+C :: IF C=10 THEN I=
2 ELSE I=1
610 RETURN
620 DATA 16,31,18,31,20,31,2
2,31,17,30,19,30,21,30,18,29
,20,29,19,28
1000 CALL VCHAR(15,27,141,9)
1001 CALL HCHAR(15,28,142,5)
1002 CALL HCHAR(23,28,143,5)
1009 FOR A=27 TO 32
1010 CALL VCHAR(15,A-1,136,9
):: CALL VCHAR(15,A,141,9)
```

```
1020 FOR II=1 TO 20 :: NEXT
II
1030 CALL SOUND(40,-7,5)
1040 NEXT A
1050 CALL VCHAR(15,32,136,9)
:: RETURN
```

```
100 CALL CLEAR :: CALL SCREE
N(13):: PRINT "BRIDGE GUARD"
:: PRINT "BY MARK SMETANA"
:: PRINT "1982"
110 REM ANDOVER NJ. 07821
120 FOR D=1 TO 10 :: PRINT :
: NEXT D
130 FOR D=1 TO 500 :: NEXT D
:: CALL CLEAR :: PRINT "USE
THE NUMBERS TO OPEN THE COR
RESPONDING SECTION OF  BRI
DGE"
140 PRINT "'1' IS THE LEFT S
ECTION    '2' IS THE SECOND
SECTION     FROM THE LEFT
AND       '0' IS THE RIGHT
SECTION."
150 PRINT "THE OTHER NUMBERS
CORRESPOND WITH THE SECTION
THAT     NUMBER FROM THE L
EFT SIDE OF THE BRIDGE."
160 PRINT "TRY TO OPEN THE C
ORRECT    SECTION OF THE BR
IDGE TO LET THE METEORS FALL
HARMLESSLY INTO THE RIVER B
ELOW"
170 PRINT "FREE SECTOR AT 50
0 POINTS.  FREE POINTS IF ME
TEOR HITS  PLANE. AS GAME GO
ES ON      METEOR SPEEDS UP.
GOOD LUCK!"
180 INPUT "INPUT THE STARTIN
G SPEED OF THE METEOR
(1-VERY SLOW - 50
-VERY FAST)":SP :: IF SP>50
OR SP<0 THEN PRINT "TRY AGAI
N WITHIN THE LIMITS" :: GOTO
180
190 CALL CLEAR :: A1$="00000
0000060F0F0FCFFFF7F0000000000
000000000F85051FDFFFDFD50301
818" :: CALL CHAR(60,A1$)
200 A2$="0000000808282494945
36B2B17CF3F0F00000040405090A
2A2245850A0CCB0C0" :: CALL C
HAR(36,A2$)
210 A3$="00000A00240412420A0
94505130B0757000050002420484
25090A2A0C8D0E0F5" :: A8$="0
03C7E7E7E3C000000000000000000
0000000000000000000000000000
0000000"
220 CALL CHAR(85,A8$):: CALL
CHAR(76,A3$):: A4$="0000000
00000FFFFFFC2A49898A4C2FF000
000000000FFFFFF432519192543F
F" :: CALL CHAR(128,A4$)
230 A5$="00040C1C3870E0C080C
2A49898A4C2FF002030381C0E070
301432519192543FF" :: CALL C
HAR(132,A5$)
240 A6$="C0C0C0C0C0C0C0C080C
2A49898A4C2FF030303030303030
301432519192543FF" :: CALL C
HAR(136,A6$)
```

Continued from P8

```
250 A7$="0000000004042424202
00C1E3F3F0F00000000202022220
2026260F8FEF8FC00"
260 CALL CHAR(140,A7$)
270 CALL CHAR(120,"FFFFFFFF
FFFFFF")::: CALL CHAR(121,"0
000002072FFFFFF"):: CALL CHA
R(80,"FFFFFFFFFFFFFFFF"):: C
ALL CHAR(81,"FFFFFCFCFCFCFFF
F")
280 CALL CHAR(82,"FFFF3F3F3F
3FFFFF"):: CALL CHAR(83,"010
3070F1F3F7FFF"):: CALL CHAR(
84,"80C0E0F0F8FCFEFF"):: CAL
L CHAR(72,"FFFFFFDB81000000"
)
290 CALL CHAR(96,"03371FFFFF
FFFFFF"):: CALL CHAR(97,"C0E
0F8FFFFFFFFFF"):: CALL CHAR(
112,"0101030303030707"):: CA
LL CHAR(113,"0103070F1F3F7FF
F")
300 CALL CHAR(114,"FFFFFFFF
FFFFFF"):: CALL CHAR(115,"E
0E0F0F0F8F8F8"):: CALL CHA
R(116,"F8FCFCFEFEFFFFFF"):: 
CALL CHAR(111,"FFFFFDB81")
310 CALL CHAR(64,"FFFFFFFFFF
FFFFFF"):: CALL CHAR(65,"FFF
FFFFFF"):: CALL CHAR(66,"FFFF
FFFFFFFFC80"):: CALL CHAR(6
7,"FFFFFFFEFCE"):: CALL CHAR
(68,"F1F7")
320 CALL CHAR(69,"FFFFFFFFFF
FF3F01"):: CALL CHAR(70,"FFF
FFF3F0701"):: CALL CHAR(71,"
7F1F03"):: CALL COLOR(1,9,1)
:: CALL COLOR(13,2,1):: CALL
 COLOR(4,2,1)
330 CALL COLOR(2,12,6):: CAL
L COLOR(14,2,1):: CALL COLOR
(5,6,1):: CALL COLOR(7,2,1):
: CALL COLOR(6,2,5):: CALL C
OLOR(12,5,1)
340 CALL COLOR(9,13,2):: CAL
L COLOR(10,13,5):: CALL COLO
R(11,13,1):: CALL COLOR(8,2,
1):: CALL MAGNIFY(3):: CALL
SCREEN(4):: CALL HCHAR(1,1,6
4,28)
350 CALL HCHAR(1,1,64,128)::
 CALL HCHAR(4,1,64,245):: CA
LL HCHAR(11,26,64,24):: CALL
 HCHAR(13,1,64,6):: CALL HCH
AR(14,1,64,4)
360 CALL HCHAR(12,29,64,4)::
 FOR D=1 TO 32 :: READ B,C :
: CALL HCHAR(B,D,C)
370 NEXT D
380 DATA 15,65,15,67,15,68,1
4,66,14,67,14,68,13,66,13,67
,13,65,13,65,13,70,13,69,13,
66,13,69,13,66
390 DATA 13,67,13,68,12,66,1
2,67,12,68,11,66,11,67,11,65
,11,70,11,69,12,71,12,70,12,
69
400 DATA 13,71,13,70,13,69,1
3,66
410 CALL HCHAR(23,3,121,28):
: CALL HCHAR(24,1,120,32)::
CALL VCHAR(21,1,114,3):: CAL
L VCHAR(20,32,114,3):: FOR D
=1 TO 12 :: READ E,F,G :: CA
LL HCHAR(E,F,G)
420 NEXT D
430 DATA 19,1,97,20,2,115,21
,2,116,23,1,111,23,2,111,19,
32,96,20,31,112,21,31,112,23
,31,111,23,32,111
```

```
440 CALL HCHAR(22,2,114):: C
ALL HCHAR(22,31,114):: CALL
VCHAR(17,5,80,6):: CALL VCHA
R(17,6,80,6)
450 CALL VCHAR(17,27,80,6)::
 CALL VCHAR(17,28,80,6):: CA
LL HCHAR(23,5,72,2):: CALL H
CHAR(23,27,72,2):: FOR D=1 T
O 8 :: READ E,F,G :: CALL HC
HAR(E,F,G)
460 NEXT D
470 DATA 15,5,83,15,6,84,16,
5,81,16,6,82,15,27,83,15,28,
84,16,27,81,16,28,82
480 FOR D=1 TO 16 :: READ Q,
R,S :: CALL HCHAR(Q,R,S)
490 NEXT D
500 DATA 18,1,128,19,1,129,1
8,2,130,19,2,131,18,3,128,19
,3,129,18,4,130,19,4,131,18,
29,128,19
510 DATA 29,129,18,30,130,19
,30,131,18,31,128,19,30,129,
18,31,128,19,31,129,18,32,13
0,19,32,131
520 CALL HCHAR(18,7,128,20):
: FOR U=1 TO 20 STEP 2 :: CA
LL HCHAR(19,6+U,129):: CALL
HCHAR(19,7+U,131)
530 NEXT U :: CALL SPRITE(#1
,136,2,137,49,0,0):: RESTORE
 :: RANDOMIZE :: SCORE=0 ::
OP=1 :: TRIES=3 :: N=0 :: CA
LL SPRITE(#12,60,7,39,1,0,25
):: DISPLAY AT(1,11)SIZE(5):
SCORE
540 DISPLAY AT(1,23)SIZE(2):
TRIES
550 I=(INT((RND*9)+1)*16)+33
 :: CALL SPRITE(#11,140,2,8,
I,SP,0)
560 CALL KEY(0,K,S):: IF S=0
 THEN 620
570 IF S=-1 THEN 620
580 IF K<48 OR K>57 THEN 620
590 K=K-48 :: IF K=0 THEN K=
10
600 CALL PATTERN(#1,132):: C
ALL HCHAR(18,5+(2*K),32,2)::
 CALL HCHAR(18,5+(2*OP),128,
2):: CALL LOCATE(#1,137,33+(
16*K)):: CALL PATTERN(#1,136
)
610 OP=K
620 CALL COINC(#1,#11,15,J):
: IF J=-1 THEN 660
630 CALL COINC(#11,#12,8,J):
: IF J=-1 THEN 770
640 CALL POSITION(#11,X,Y)::
 IF X>122 THEN 800
650 GOTO 560
660 CALL POSITION(#11,X,Y)::
 IF X>160 THEN 670 ELSE 660
670 CALL DELSPRITE(#11)
680 L=SCORE
690 SCORE=SCORE+SP :: IF L<5
00 AND SCORE>=500 THEN TRIES
=TRIES+1 :: GOTO 740
700 IF L<10000 AND SCORE>=10
000 THEN SCORE=SCORE-10000
710 DISPLAY AT(1,11)SIZE(5):
SCORE :: DISPLAY AT(1,23)SIZ
E(2):TRIES :: CALL SOUND(25,
1000,1,750,1,500,1,-6,1):: I
F SP>48 THEN 550
720 N=N+1 :: IF N=10 THEN SP
=SP+2 :: N=0
730 GOTO 550
```

```
740 CALL SOUND(20,131,1):: C
ALL SOUND(20,147,1):: CALL S
OUND(20,165,1):: CALL SOUND(
40,220,1):: CALL SOUND(20,16
5,1):: CALL SOUND(80,220,1):
: GOTO 710
750 IF J=-1 THEN 660
760 CALL MOTION(#11,0,0)
770 CALL MOTION(#11,0,0):: C
ALL MOTION(#12,0,0):: CALL P
ATTERN(#11,36):: CALL PATTER
N(#12,76):: FOR D=1 TO 30 ::
 CALL SOUND(20,110,D,120,D,1
30,D,-5,D)
780 NEXT D :: CALL DELSPRITE
(#11,#12):: CALL SPRITE(#12,
63,7,39,1,0,25)
790 GOTO 680
800 CALL COINC(#11,#1,15,J):
: IF J=-1 THEN 660
810 CALL MOTION(#11,0,0):: C
ALL PATTERN(#11,36):: CALL S
PRITE(#2,76,12,137,I,0,0)::
FOR D=1 TO 30 :: CALL SOUND(
20,110,D,120,D,130,D,-5,D)
820 NEXT D :: TRIES=TRIES-1
:: DISPLAY AT(1,23)SIZE(2):T
RIES :: IF TRIES=0 THEN 850
830 CALL DELSPRITE(#11,#2)
840 FOR D=1 TO 500 :: NEXT D
 :: GOTO 550
850 FOR M=1 TO 10
860 CALL SCREEN(2):: FOR D=1
 TO 5 :: NEXT D
870 CALL SCREEN(7):: FOR D=1
 TO 5 :: NEXT D
880 CALL SCREEN(16):: FOR D=
1 TO 5 :: NEXT D :: CALL SOU
ND(10,110,1,-(INT(RND*7)+1),
1)
890 NEXT M :: FOR D=1 TO 8 :
: CALL SOUND(1000,-D,1)
900 NEXT D :: CALL SCREEN(2)
:: CALL DELSPRITE(#2)
910 FOR D=1 TO 30 :: CALL SO
UND(-250,-5,1):: NEXT D
920 FOR D=1 TO 250 :: NEXT D
 :: GOSUB 940 :: CALL DELSPR
ITE(ALL):: CALL CLEAR :: CAL
L SCREEN(13):: CALL CHARSET
:: PRINT "YOUR SCORE WAS";SC
ORE :: PRINT "DO YOU WANT TO
 PLAY AGAIN   (Y/N)";:: INPU
T AN$ :: IF AN$="Y" THEN 180
930 END
940 CALL MOTION(#12,20,20)::
 FOR W1=200 TO 110 STEP -5 :
: CALL POSITION(#12,X,Y):: C
ALL SOUND(-200,W1,0):: IF X>
140 THEN CALL MOTION(#12,0,0
):: GOTO 960
950 NEXT W1
960 CALL SOUND(900,110,1,-7,
1):: CALL PATTERN(#12,76)
970 FOR D=1 TO 250 :: NEXT D
 :: RETURN
980 END
```

```
1 HS=760
100 RANDOMIZE :: CALL SCREEN
(INT(RND*8)+8)
110 CALL CLEAR
120 ON BREAK NEXT
130 RESTORE :: RANDOMIZE ::
CALL MAGNIFY(3)
140 CALL CHAR(96,"0000000000
61067C7C0661000000000000000
006EEE247F7F24EE6E")
150 CALL CHAR(100,"000000000
7FFFBFBFBFBFF000000000000000
0FEFEFEFEFEFEFEFE")
160 CALL CHAR(104,"000000001
C3F7D7D7D7D3F1C0000000000000
01C7EFFFFEEEEFFFF7E1C")
170 CALL CHAR(108,"000000001
83F7E7D7D7E3F180000000000000
00018FEF7F7F7F7FE18")
180 CALL CHAR(112,"000000000
3056AB6F1B66A050300000000000
00030A05068906B50A030")
190 CALL CHAR(116,"000000000
03F7FFFFFFFFF7F0000000000000
00000FEFDF9F9F9F9FE")
200 GOSUB 1000
210 CALL CLEAR
220 CALL CHAR(120,"000000000
000B37CB0000000000000000000
000000000CC33")
230 CALL CHAR(72,"0000000000
6F6F1C1C6F6F000000000000000
0000COF3FCFCF3C")
240 CALL CHAR(76,"00000000FF
FFFFFFFFFFFFFF00000000000000
00C0BEDFDFDFDFBEC")
250 CALL CHAR(80,"0000000018
1B070C0C071B1800000000000000
000000E07B7BE")
260 CALL CHAR(84,"0000000007
3F7F1F7F3F070000000000000000
00EOBCDEFDEBEE")
270 FRG=0 :: FR=3 :: SC=0 ::
 LE=1 :: CO=0 :: NM=1 :: FOR
I=1 TO 5 :: S(I)=INT(RND*5)
+4 :: LIL(I)=0 :: NEXT I ::
FOR J=6 TO 9 :: S(J)=INT(RND
*5)+4 :: NEXT J :: RESTORE
280 CALL CHAR(33,RPT$("F",16
),40,RPT$("F",16),34,"",41,"
"):: CALL COLOR(13,4,5,1,5,4
,2,2,4,3,2,4,4,2,4,5,2,4,7,1
4,4)
290 CALL SCREEN(INT(RND*8)+8
):: CALL COLOR(8,13,5,12,13,
4)
300 CALL HCHAR(3,1,33,32*10)
:: CALL HCHAR(13,1,34,64)::
CALL HCHAR(15,1,40,32*8):: C
ALL HCHAR(23,1,41,64)
310 CALL CHAR(132,"000F1F3F3
F7F7F7F",133,"7F7F3F3F1F07",
134,"0080C0COCOFCFEFE",135,"
FEFEFCF8F0C")
320 FOR I=3 TO 23 STEP 5 ::
CALL HCHAR(3,I,132):: CALL H
CHAR(3,I+1,134):: CALL HCHAR
(4,I,133):: CALL HCHAR(4,I+1
,135):: NEXT I
330 DATA 00384C546438,001030
101038,00384408107C,00384418
4438,000818287C08,007C403C04
7C,001020784438,007C0810202,
003844384438,0038443C081
```

```
340 DATA 003C40380478,003C44
40403C,007C4444447C,00784478
504C,007C4078407C,007C101010
1,00784444784
350 FOR I=48 TO 64 :: READ A
$ :: CALL CHAR(I,A$):: NEXT
I :: DISPLAY AT(1,1)SIZE(13)
:":;<=> 000" :: DISPLAY AT(1
,19)SIZE(4):"?<@"
360 DISPLAY AT(1,23)SIZE(7):
HS :: CALL CHAR(124,"99BDFF7
E7E6681"):: CALL HCHAR(1,16,
124,3)
370 CALL CHAR(92,"3048B4B547
3F0F1F",93,"1F1F0F071F130878
",94,"0C122DADE2FCF0F8",95,"
F8F8F0E0F8C8101E")
380 CALL SPRITE(#15,116,10,8
1,33,0,S(6),#16,116,10,81,10
5,0,S(6),#17,116,10,81,145,0
,S(6))
390 CALL SPRITE(#21,116,10,4
9,17,0,S(8),#22,116,10,49,12
1,0,S(8),#23,116,10,49,201,0
,S(8))
400 CALL SPRITE(#24,112,16,3
3,41,0,-1*S(9),#25,112,16,33
,97,0,-1*S(9),#26,112,16,33,
185,0,-1*S(9))
410 CALL SPRITE(#18,112,16,6
5,73,0,-1*S(7),#19,112,16,65
,169,0,-1*S(7),#20,112,16,65
,233,0,-1*S(7))
420 CALL SPRITE(#1,96,9,161,
25,0,-1*S(1),#2,104,14,161,7
3,0,-1*S(1),#3,108,15,161,16
1,0,-1*S(1))
430 CALL SPRITE(#7,100,11,12
9,73,0,-1*S(3),#8,108,6,129,
137,0,-1*S(3),#9,98,7,129,22
5,0,-1*S(3))
440 CALL SPRITE(#4,72,8,145,
41,0,S(2),#5,80,7,145,105,0,
S(2),#6,84,6,145,193,0,S(2))
450 CALL SPRITE(#10,76,16,11
3,41,0,S(4),#11,72,8,113,105
,0,S(4),#12,80,15,113,177,0,
S(4))
460 CALL CHAR(136,"000000000
90B0F0703030608000000000000
00090D0F0E0C0C0601")
470 CALL CHAR(140,"00000B0B0
90B07030303060B0B0B000000001
01090D0E0C0C0C06010101"):: G
OTO 490
480 FOR I=1 TO 40 :: NEXT I
490 CALL SPRITE(#13,136,13,1
77,100)
500 CALL POSITION(#13,A,B)::
 IF B<=5 OR B>=251 THEN 580
ELSE CALL JOYST(1,X,Y):: IF
A=97 AND ABS(Y)=4 THEN 510 E
LSE 520
510 SC=SC+(10*LE):: DISPLAY
AT(1,7)SIZE(7):SC :: CALL SO
UND(50,-2,4):: GOTO 670
520 IF Y=0 THEN 560
530 CALL SOUND(50,-2,4)
540 CALL PATTERN(#13,140)
550 CALL LOCATE(#13,A-16,B):
: GOTO 570
560 CALL MOTION(#13,-(ABS(Y)
*4),X*4)
570 CALL COINC(ALL,C):: IF C
=0 THEN CALL PATTERN(#13,136
):: GOTO 500 ELSE 580
```

```
580 CALL DELSPRITE(#13):: CO
=0 :: FR=FR-1 :: CALL HCHAR(
1,16,32,3):: CALL HCHAR(1,16
,124,FR):: IF FR=0 THEN 590
ELSE ON NM GOTO 480,650,660
590 FOR J=1000 TO 300 STEP -
150 :: CALL SOUND(90,J,1)::
NEXT J :: FOR I=1 TO 20 :: N
EXT I :: DISPLAY AT(11,5):"G
AME OVER"
600 IF SC>HS THEN HS=SC :: C
ALL SOUND(300,900,0):: DISPL
AY AT(1,23)SIZE(7):HS
610 CALL DELSPRITE(ALL):: CA
LL CLEAR
620 CALL CHARSET :: DISPLAY
AT(8,10):"YOU SCORED" :: DIS
PLAY AT(10,13):SC :: DISPLAY
 AT(15,8):"HIGHEST SCORE IS"
 :: DISPLAY AT(17,12):HS
630 DISPLAY AT(24,2):"DO YOU
 WANT TO PLAY AGAIN?" :: CAL
L KEY(3,K,E):: IF K=89 THEN
CALL CLEAR :: GOTO 230
640 IF K=78 THEN CALL CLEAR
:: END ELSE 630
650 CALL SPRITE(#24,112,16,3
3,41,0,-S(9),#25,112,16,33,9
7,0,-S(9),#26,112,16,33,185,
0,-S(9)):: GOTO 490
660 CALL SPRITE(#18,112,16,6
5,73,0,-S(7),#19,112,16,65,1
69,0,-S(7),#20,112,16,65,233
,0,-S(7)):: GOTO 490
670 E=6 :: NO=1
680 CALL LOCATE(#13,A-16,B)
690 CALL MOTION(#13,0,NO*(S(
E))):: CALL COINC(#13,#((E*3
)-3),7,C):: CALL COINC(#13,#
((E*3)-2),7,X):: CALL COINC(
#13,#((E*3)-1),7,Z)
700 IF C=0 AND X=0 AND Z=0 T
HEN CALL SOUND(250,-5,6):: G
OTO 580
710 ON NM GOTO 720,790,800
720 CALL POSITION(#13,A,B)::
 IF B<=5 OR B>=251 THEN 580
ELSE CALL JOYST(1,X,Y):: IF
Y=4 AND X=0 THEN E=E+1 :: CA
LL SOUND(50,-2,4):: GOTO 740
730 CO=CO+1 :: IF CO=40 THEN
 760 ELSE 720
740 IF NO=1 THEN NO=-1 ELSE
NO=1
750 NM=1 :: A=A-16 :: IF A=1
7 THEN 810 ELSE CALL LOCATE(
#13,A,B):: GOTO 690
760 ON INT(RND*2)+1 GOTO 770
,780
770 CALL DELSPRITE(#24,#25,#
26):: CO=0 :: NM=2 :: GOTO 6
90
780 CALL DELSPRITE(#18,#19,#
20):: CO=0 :: NM=3 :: GOTO 6
90
790 CALL SPRITE(#24,112,16,3
3,41,0,-S(9),#25,112,16,33,9
7,0,-S(9),#26,112,16,33,185,
0,-S(9)):: GOTO 720
800 CALL SPRITE(#18,112,16,6
5,73,0,-S(7),#19,112,16,65,1
69,0,-S(7),#20,112,16,65,233
,0,-S(7)):: GOTO 720
```

Continued from P10

```
810 CALL POSITION(#13,A,B)::
    IF B>14 AND B<37 AND LIL(1)
    =0 THEN COL=3 :: LIL(1)=1 ::
    GOTO 870
820 IF B>54 AND B<76 AND LIL
    (2)=0 THEN COL=8 :: LIL(2)=1
    :: GOTO 870
830 IF B>94 AND B<116 AND LI
    L(3)=0 THEN COL=13 :: LIL(3)
    =1 :: GOTO 870
840 IF B>134 AND B<156 AND L
    IL(4)=0 THEN COL=18 :: LIL(4
    )=1 :: GOTO 870
850 IF B>174 AND B<196 AND L
    IL(5)=0 THEN COL=23 :: LIL(5
    )=1 :: GOTO 870
860 CALL MOTION(#13,0,0):: C
    ALL LOCATE(#13,17,B):: CALL
    SOUND(250,-5,6):: FOR I=1 TO
     50 :: NEXT I :: GOTO 580
870 CALL DELSPRITE(#13):: CA
    LL HCHAR(3,COL,92):: CALL HC
    HAR(3,COL+1,94):: CALL HCHAR
    (4,COL,93):: CALL HCHAR(4,CO
    L+1,95):: FRG=FRG+1
880 SC=SC+(20*LE):: DISPLAY
    AT(1,7)SIZE(7):SC
890 IF FRG=5 THEN SC=SC+(30*
    LE):: FRG=0 :: DISPLAY AT(1,
    7)SIZE(7):SC :: GOTO 900 ELS
    E 490
900 FOR I=1 TO 5 :: LIL(I)=0
     :: CALL SOUND(I*40,I*110,I+
    1):: NEXT I
910 FOR I=3 TO 23 STEP 5 ::
    CALL HCHAR(3,I,132):: CALL H
    CHAR(3,I+1,134):: CALL HCHAR
    (4,I,133):: CALL HCHAR(4,I+1
    ,135):: NEXT I
920 LE=LE+1 :: IF LE>=3 THEN
    CALL SPRITE(#14,120,14,97,7
    3,0,-S(5))
930 FOR I=1 TO 9 :: S(I)=S(I
    )+INT(RND*3)+1
940 IF S(I)>127 THEN S(I)=12
    7
950 NEXT I
960 CALL MOTION(#1,0,-S(1),#
    2,0,-S(1),#3,0,-S(1)):: CALL
    MOTION(#4,0,S(2),#5,0,S(2),
    #6,0,S(2))
970 CALL MOTION(#7,0,-S(3),#
    8,0,-S(3),#9,0,-S(3)):: CALL
    MOTION(#10,0,S(4),#11,0,S(4
    ),#12,0,S(4))
980 CALL MOTION(#15,0,S(6),#
    16,0,S(6),#17,0,S(6)):: CALL
    MOTION(#24,0,-S(9),#25,0,-S
    (9),#26,0,-S(9))
990 CALL MOTION(#18,0,-S(7),
    #19,0,-S(7),#20,0,-S(7)):: C
    ALL MOTION(#21,0,S(8),#22,0,
    S(8),#23,0,S(8)):: GOTO 490
1000 DISPLAY AT(8,4):"  F R
    O G G E R"
1010 DISPLAY AT(24,3):"DO YO
    U WANT INSTRUCTIONS?"
1020 CALL KEY(3,K,E):: IF K=
    89 THEN 1030 ELSE IF K=78 TH
    EN RETURN ELSE 1020
1030 CALL CLEAR
1040 PRINT "YOU HAVE TO GUID
    E A FROG":
1050 PRINT "ACROSS A BUSY RO
    AD.TO A LILY":
```

```
1060 PRINT "PAD ON THE OTHER
     SIDE OF A ":
1070 PRINT "RIVER.YOU MUST H
    OP ON LOGS":
1080 PRINT "AND TURTLES TO D
    O THIS.":
1090 PRINT "BUT WATCH OUT.TH
    E TURTLES":
1100 PRINT "DIVE FROM TIME T
    O TIME.AND A":
1110 PRINT "SNAKE APPEARS ON
     THE RIVER":
1120 PRINT "BANK AFTER TWO S
    ( WS.":
1130 PRINT "YOU'RE FROG WILL
     DROWN IF HE" :: PRINT "FALL
    S INTO THE RIVER. AND"
1140 PRINT "YOU CANNOT HOP B
    ACKWARDS."
1150 FOR I=1 TO 3500 :: NEXT
     I :: RETURN
1160 END
```



```
100 REM **********
110 REM * BUBBLE *
120 REM **********
130 REM VERSION 1.XB NOV.82
140 REM BY R.S.KEMMETMUELLER
150 CALL CLEAR
160 PRINT "DO YOU WANT DIREC
    TIONS?(Y/N)"
170 CALL KEY(0,K,S)
180 IF K=78 THEN 370
190 IF K<>89 THEN 170
200 CALL CLEAR
210 PRINT "0000000    BUBBLE
    0000000"
220 PRINT
230 PRINT "THE OBJECT OF BUB
    BLE IS TO"
240 PRINT "GET FROM THE LEFT
    SIDE OF "
250 PRINT "THE SCREEN TO THE
    RIGHT SIDE"
260 PRINT "BY JUMPING FROM B
    UBBLE TO"
270 PRINT "BUBBLE WITHOUT MI
    SSING ONE "
280 PRINT "AND FALLING INTO
    THE WATER."
290 PRINT "IF YOU MISS A BUB
    BLE BUT "
300 PRINT "ANOTHER ONE CATCH
    ES YOU"
310 PRINT "BEFORE YOU HIT TH
    E WATER,YOU"
320 PRINT "ARE SAFE. USE JOY
    STICK #1 OR"
330 PRINT "THE S & D KEYS TO
     JUMP LEFT"
340 PRINT "OR RIGHT. PRESS E
    NTER TO GO."
350 INPUT "":ZZZZ$
360 CALL CLEAR
370 CALL MAGNIFY(3)
380 CALL CLEAR :: CALL DELSP
    RITE(ALL)
```

```
390 CALL CHAR(96,"00185A3C18
    18244200000000000000000000000
    00000000000000000000000000000")
400 CALL CHAR(42,"88CCEEFF77
    331100")
410 CALL CHAR(104,"071F3F7F7
    FFFFFFFFFFFFFF7F7F3F1F07E0F8F
    CFEFEFFFFFFFFFFFFFFFEFEFCF8E0"
    )
420 CALL HCHAR(23,1,42,64)::
     CALL COLOR(2,5,6)
430 CALL SPRITE(#2,104,16,10
    0,1,0,0)
440 DIM RV(256)
450 SP=-1 :: SPR=2
460 FOR COL=17 TO 240 STEP 1
    6
470 SPR=SPR+1 :: SP=-SP :: R
    V(COL+8),RV(COL)=INT(RND*4+1
    )*SP :: CALL SPRITE(#SPR,104
    ,16,100,COL,RV(COL),0)
480 NEXT COL
490 CALL SPRITE(#SPR+1,104,1
    6,100,241,0,0)
500 R=100 :: C=9 :: CALL SPR
    ITE(#1,96,2,R,C,0,0)
510 REM
520 CALL JOYST(1,X,Y)
530 CALL KEY(1,K,S):: IF S=0
     AND X=0 THEN 510
540 IF K=2 THEN X=-4
550 IF K=3 THEN X=4
560 CALL POSITION(#1,R,C)
570 R=R+RV(C)/2 :: C=C+X*2
580 IF C>256 THEN 690
590 CALL LOCATE(#1,R,C)
600 CALL MOTION(#1,RV(C),0)
610 CALL COINC(ALL,BAM):: IF
     BAM=-1 THEN CALL SOUND(-100
    ,-2,0):: GOTO 510
620 CALL MOTION(#1,0,0)
630 FOR RR=R TO 192 STEP 7
640 CALL SOUND(-200,-(RR+110
    )+500,0)
650 CALL LOCATE(#1,RR,C)
660 CALL COINC(ALL,BAM):: IF
     BAM=-1 THEN CALL MOTION(#1,
    RV(C),0):: GOTO 510
670 NEXT RR
680 CALL CLEAR :: CALL DELSP
    RITE(ALL)
681 PRINT "TOO BAD"
682 GOTO 160
690 REM YOU MADE IT
700 CALL CLEAR :: CALL DELSP
    RITE(ALL)
710 PRINT "        YOU DID I
    T !!"
720 PRINT ""
730 TT=TT+5 :: IF TT=100 THE
    N 160
740 GOTO 710
```

FORUM

with CHRIS

I have been experiencing problems getting my TI-Writer
to function properly with my printer when using A4 size
paper. I have used the command .PL 70 on the document
without success. Is there something I am missing?
Percy H. ...North Ryde.

Yes Percy, you need to do one more thing. The command
.PL 70 simply tells the FORMATTER the number of lines to
each page of your paper. So far so good but you also
must tell the printer the number of lines per page if
you don't want it to print across margin perforations.
This can be done in one of two ways. The first is with
special character mode of TI-Writer. You will find an
explanation on page 98 of the manual with the characters
listed and defined on page 146. Function "U" is used to
enter and leave the special character mode. The second
method is with the transliterate command. Here is an
example of the transliterate method for Epson printers:
.TL 126:27,67,70(CR)
When the ~(tilde) character is found at the beginning of
the file, it will set the printer to 70 lines per page.
(Don't forget to include the Transliterate command at
the top of the page!) I suggest you get a hands-on demo
of both methods at one of the regional group meetings.

* * * * *

I am writing in regard to your programme in the March
issue of TND named COMPACT. I have finally got the
programme to work using MM and the V2.2 console. Some
modification was necessary to the programme as you will
observe. I usually can get most of the smaller assembly
programmes to work with MM after a bit of fiddling.
Kevin Cox....Bayldon

Here is Kevin's modified programme for the MiniMem and
V2.2 console.

PROGRAMME COMPACT

```
        AORG >7D0C
VM      EQU  >6028
GP      EQU  >6018
TE      DATA >8A80,>A7AF,>A9AE,>A780,>D4CF
        DATA >8080,>A7B0,>ACAC,>AEAB,>80C3
        DATA >CFCD,>D0C1,>C3D4,>808A
SA      BSS  2
LS      DATA >0002
LC      BSS  2
CO      MOV  R11,@SA
        MOV  R11,R10
        LI   R0,>02C1
        LI   R1,TE
        LI   R2,28
        BLWP @VM
        MOV  @LS,@LC
SC      MOVB @LS,@>837C
        BLWP @GP
        DATA >56CD
        DEC  @LC
        JNE  SC
        MOVB @LS,@>837C
        BLWP @GP
        DATA >51A9
        MOV  @SA,R10
        B    *R10
        END
        AORG >701C
        DATA >7FB2
        DATA >7FE0
        AORG >7FE0
        TEXT 'COMPAC'
        DATA CO
        END
```

Life.Be in it.
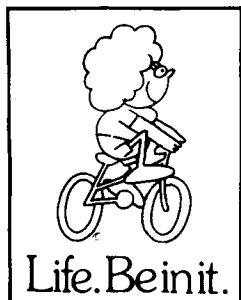
```
 90 CALL CLEAR
100 PRINT "FORCING A GARBAGE COLLECTION"
110 PRINT "by Ross Mudie, January 1987"
120 PRINT :"*BASIC WITH MINI MEMORY"
130 PRINT "This basic program shows how"
140 PRINT "to read memory space free"
150 PRINT "whilst the basic program is"
160 PRINT "running and fills the memory"
170 PRINT "with redundant strings.  You"
180 PRINT "will observe the pause when"
190 PRINT "the garbage collection"
200 PRINT "occurs and then you get the"
210 PRINT "opportunity to cause a"
220 PRINT "Garbage Collection to occur."
230 PRINT "The purpose of this is that"
240 PRINT "using the Assembly Routine"
250 PRINT "you can control when the "
260 PRINT "Garbage is collected and"
270 PRINT "with proper programing you"
280 PRINT "can eliminate the pause at"
290 PRINT "the wrong moments."::
292 REM CALL INIT
294 REM CALL LOAD("DSK1.BICSUP")
296 REM CALL LOAD("DSK1.COMPACT")
300 INPUT "Press enter to continue":E$
310 FOR A=1 TO 45
320 CALL PEEK(-31890,I,M)
330 CALL PEEK(-31974,H,L)
340 PRINT "Free=";STR$((H*256+L)-(I*256+N)),"Test Loop="
    ;STR$(A)
350 A$="AAAAAAAAAAAAAAAAAAAAAAAAAAA"
360 A$=A$&A$&A$&A$&A$
370 NEXT A
380 INPUT "PRESS ENTER for Compact":E$
390 CALL LINK("COMPAC")
400 CALL PEEK(-31890,I,M)
410 CALL PEEK(-31974,H,L)
420 PRINT "After compact FREE="STR$((H*256+L)-(I*256+M))
    ::
430 GOTO 310
```

Thank you Kevin. It is always gratifying to see members
accepting the challenge to modify programmes for their
own particular needs.
Note: The above programme has not been tested. It is
offered for your information and experimentation. If
you find any "bugs" please let us know.

* * * * *

Please publish postage costs on Items from the Shop. It
is hard to authorise a sum on Bankcard if postage is
unknown.
H. Hull....Tamworth.

It is hard to give definitive postage costs because the
charges are arranged in weight bands. The simplest
method is to authorise the cost of items you want PLUS
packing and postage. That way, you are charged only for
actual costs incurred by the club.

* * * * *

Like the Forum and Techo-Time Formats. Wish you all
every success.
D. Gould....Cremorne.

Thank you for your comments and encouragement. It is
nice to get some feedback when changes are made.

* * * * *

```
*****************************
*                           *
*     USING PILOT 99        *
*                           *
*****************************
        |by Arto Heino|
        ----------------
```

To write a new language for our 99/4A is not as awe
aspiring as you think. The FORTH language was written
in such a way that a whole new language can be easily
implemented step by step. PILOT was written in such a
way by THOMAS P. WEITHOFER.

Your PROGRAM is actually a TEXT FILE that you write
using an EDITOR such as E/A, TI-WRITER or even an
EXTENDED BASIC program! Once you have written a
program and saved it, you must now LOAD you PILOT
SYSTEM DISK.

If you have the E/A version and only have XB then LOAD
in your XB FORTH DISK and just type 'COLD' with the
PILOT DISK in your drive.(also done with FORTI) If you
have a RAM DISK you will be very pleased at the new
speed of FORTH or PILOT or any FORTH BASED language.

PILOT is a very powerful language for Programmed
Inquiry Learning Or Teaching which is what PILOT
stands for. Where as LOGO is GRAPHICS based, PILOT is
TEXT based, that is, until THOMAS put a whole lot of
the advanced features of our TI 99/4A into the
language.

The best way to LEARN a new language is to USE IT! I
have written a short program that draws a 12 month bar
graph of the data you input.

```
R: ***********************
R: BAR GRAPH by Arto Heino
R:       T.I.s.H.U.G.
R: ***********************
IT:
CH:
R: ************
R: DEFINE CHARS
R: ************
CP: 128,0000,001F,1010,1010
CP: 129,0000,00FF,0000,0000
CP: 130,0000,00F0,1010,1010
CP: 131,1010,1010,1010,1010
CP: 132,1010,101F,0000,0000
CP: 133,1010,10F0,0000,0000
R: **********************
R: DISPLAY OPENING SCREEN
R: **********************
TC: 12,8
T: PILOT BAR GRAPH
HC: 10,8,128,1
HC: 10,9,129,15
HC: 10,24,130,1
VC: 11,8,131,3
VC: 11,24,131,3
HC: 14,8,132,1
HC: 14,9,129,15
HC: 14,24,133,1
LP: 70
EL:
R: *****************
R: DRAW UP MAIN SCREEN
R: *****************
*GO
IG:
TG: 1,20,PILOT BAR GRAPH by Arto Heino
TG: 16,8,Jan Feb Mar Apl May Jun Jul Aug
TG: 16,40,Sep Oct Nov Dec
TG: 2,3,100
TG: 6,3,75
TG: 9,3,50
TG: 12,3,25
TG: 15,3,0
DL: 16,21,118,21
DL: 118,21,118,208
```

```
R: ****************
R: SET UP VARIABLES
R: ****************
C: #V<-0
C: #N<-8
C: #L<-116
R: ********
R: MAIN LOOP
R: ********
LP: 12
C: #V<-#V+1
C: #N<-#N+16
T:  Values Between 0-100
T:  Input Data for Month :#V:
A:  #A
T:
C: #I<-#L-#A
C: #J<-#N+8
C: #O<-#I-4
C: #P<-#N+4
C: #Q<-#P+8
DR: #I,#N,#L,#J
DL: #I,#N,#O,#P
DL: #O,#P,#O,#Q
DL: #O,#Q,#I,#J
DL: #O,#Q,#L-4,#Q
DL: #L-4,#Q,#L,#J
EL:
R: ***********
R: END OF LOOP
R: ***********
T:  Try Again Y or N ?
A:  $A
M:  Y
JY: *GO
M:  N
JN: @A
E:
```

Commands used in the Program:

```
R:   Remark
IT:  Initialize Text Mode 32*24
CH:  Clear screen and Home cursor 1,1
CP:  Same as CALL CHAR
TC:  Text Cursor Position Y,X
T:   Same as PRINT
HC:  Same as CALL HCHAR
VC:  Same as CALL VCHAR
LP:  Initialize Loop count
EL:  End of Loop
*GO  Lable postion for a Jump
IG:  Initialize Graphics Screen 256*128
TG:  Print At Y,X Graphics Screen 64*16
DL:  Draw Line Y1,X1,Y2,X2 (BITMAP ONLY)
DR:  Draw Rectangle Y1,X1,Y2,X2   "
C:   Compute Numeric Variables
A:   Same as INPUT
M:   Match same as IF A$="Y" THEN TRUE
JY:  Jump with added modifier same as
     IF TRUE THEN JUMP
JN:  Jump with added modifier same as
     IF FALSE THEN JUMP
E:   Same as END
```

'I'm all for computers, they do all the dull,
boring jobs for you.'

THE COMMUNICATORS

The Special Interest Group (SIG) for the TEXPAC
Bulletin Board Service (BBS).
by Ross Mudie, 8th April 1987.

1. BBS USAGE.

The TEXPAC BBS is being regularly used by about 60 of the 130 registered users as a quick and convenient way to get programs, news and information. Each month a different selection of TI Basic and TI Extended Basic programs are placed on the BBS for members to download. In the news and information section there is the latest information on TIsHUG and the regional group meetings, members private advertisements and usually some files of hints or tutorial information.

TIsHUG regional groups are invited to join the BBS, already the usernames GLEBE, ILLAWARRA and NEPEAN reflect participation on a regional group level.

If you want to submit an article for the TIsHUG News Digest magazine, then the easiest way to do this is to place your file on the BBS as mail to the EDITOR. The EDITOR regularly reads his mail off the BBS and this allows the work load of preparing the magazine to be spread out through the month.

The TEXPAC BBS is easy to use since it is MENU driven in addition to the extensive BBS_HELP file in the NEWS section of the BBS. The hardware required by a user is a TI99/4A computer with RS232, modem and Terminal Emulator II cartridge. A 32K expansion memory, disk system and printer are optional (but very handy). If you are equipped with 32K memory expansion and disk system then there are some excellent Terminal programs which overcome the need for the TEII cartridge.

2. USER PROBLEMS.

(a) Program Download.
Users are reminded that whilst you can log on to the BBS and read the NEWS files with any type of computer, you must only use a TI99/4A to download a program via Main Menu option 2. The protocol used for program download in the BBS is the same as SAVE RS232 . This is why the user goes back to TI Basic or TI Extended Basic (without touching the modem) and types OLD RS232 . The program is transferred in memory image format and after being saved is ready to use.

If a user initiates a program download with another computer type they will just get garbage or nothing. The BBS will now wait for the communications of the user's RS232 card and as far as a text type terminal program the BBS will be "hung up". The next occurance is that the caller disconnects from the telephone line and the BBS still waits for the RS232 card to do the download. When the next caller rings in the effect is the same as if the BBS had crashed. Before hanging up in disgust when this happens, just try OLD RS232 from TI Basic. The program will then download and you will be logged on with the user name of the previous user. Use option 9 in the main menu to change to your own user name without the need for another phone call. Overcoming this problem will require a modification to the code in the ROM in RS232 card of the BBS which will be attempted as time permits.

(b) No Action Time Out.
If the BBS is paused in a listing or waits at any prompt for 9 minutes then the BBS will disconnect the modem from the line without warning and the telephone exchange will automatically disconnect the call 90 seconds later.

3. PLACING MAIL or ADVERTISEMENTS on the BBS.

If any user wishes to place an advertisment or public message on the BBS then the item should be addressed to ALL in the option 3 mail routine. Mail posted to ALL will be immediately readable via main menu option 4. When the SYSOP is next on the BBS the item will be transferred to the appropiate file in the NEWS menu and the ALL file will be purged. The adds, which are for members and friends of members are expected to usually be concerned with the TI99/4A computer, software and peripherals. If you are in doubt if an advertisement or announcement may be acceptable then address it to SYSOP. Whilst this will delay the general posting for all to see, it may reduce the complaints from other users who may disagree with the item being placed on the BBS.

Please refrain from placing long files in the ALL mail file unless the item is of immediate importance. Long items should be addressed to SYSOP, please.

When you enter the MAIL routine the BBS reports the number of free sectors on the mail disk. If an attempt is made to exceed the free space the BBS will refuse to accept further mail until another user collects mail and allows deletion.

The valid BBS user names are shown in the file BBS_USERS. Full information on sending mail is found in the NEWS menu file entitled SENDMAIL.

4. The NEW MODEM.

Some problems have been experienced with the smart modem purchased for the BBS. When the modem is placed in service the following changes will be in effect.

(a) The new modem will answer on the first ring, where the old modem answers on the third ring.

(b) You must switch your modem on line as soon as tone is heard after answer to ensure that your modem is correctly identified. The modem goes through the line speed test only once per incoming call.

(c)(i) To PAUSE a listing use <ctrl>S , as is the case at present.

(ii)To UN-PAUSE you must use <ctrl>Q , this is new for TEXPAC BBS.

(d) When using the 300 baud line rate if E is used to to escape a listing it may take up to 500 bytes to escape the listing due to the buffer in the modem.

(e) The pauses in listings at 300 baud, caused by the the disk drive stopping and having to start again, will dissapear due to the effect of the buffer in in the new modem.

(f) When you are downloading a program, (Main Menu #2 on the BBS), you will need to be using 300/300 baud or 1200/1200 baud, unless you have a modified EPROM in your RS232 card. TIsHUG technical members are working on the modifications in an EPROM to replace the ROM in the RS232 card to allow the RS232 card to operate at 1200/75 or 75/1200 baud with OLD RS232. There will be more details of this development on the BBS & in future COMMUNICATORS columns. When using 1200/1200 baud for download the command OLD "RS232.BA=1200" will be required; 300 baud will be unchanged as OLD RS232 .

5. HOW TO JOIN THE BBS.

Financial TISHUG members send $5(PA) to the SECRETARY at PO Box 214, Redfern, 2016, stating preferred username and initial password. The secretary or co-ordinator will advise you of your user number and actual username and password.

Non members of TIsHUG must pay $8 joining fee and $10 annual membership if a member of another TI user group or $25PA if not affiliated with any other TI User group. o
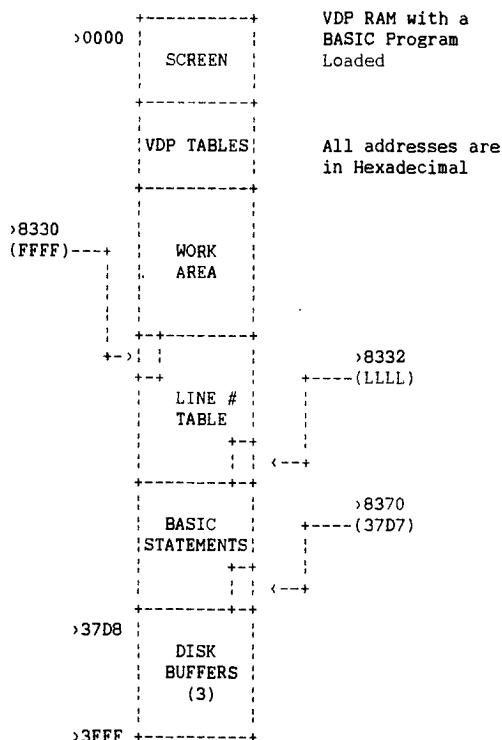
A LOOK AT BASIC PROGRAMS
by R.A.Green.

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

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

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

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

```
                +----------+    VDP RAM with a
     >0000 !          !    BASIC Program
           !  SCREEN  !    Loaded
           !          !
                +----------+
           !          !
           !VDP TABLES!    All addresses are
           !          !    in Hexadecimal
                +----------+
  >8330     !          !
  (FFFF)---+ !   WORK  !
           ! !   AREA  !
           ! !         !
           ! !         !
           ! +-+-------+
      +->! !         !         >8332
           +-+ !         !    +----(LLLL)
           ! LINE # !         !
           ! TABLE  !         !
           !        +-+       !
           !          ! ! <--+
           +---------+-+
           !          !         >8370
           !  BASIC   !    +----(37D7)
           !STATEMENTS!         !
           !        +-+       !
           !          ! ! <--+
           +---------+-+
  >37D8     !          !
           !   DISK   !
           !  BUFFERS !
           !    (3)   !
           !          !
  >3FFF +----------+
```

Each entry in the line number table is 4 bytes. The first
word (2 bytes) is the BASIC line number. The second word
is the VDP RAM address of the start of that statement.

The numbers in the line number table are in reverse order.

For example, if we had a 3 statement BASIC program loaded,
the line number table and statement area might look as
shown below.

```
>8330                 !    VDP RAM      !
(>3600)---+           !>--- 4 bytes --->!
          !              +--------+--------+
      +-- >3600 !  0003  !  360D  !         >8332
                +--------+--------+    +----(>360B)
        >3604 !  0002  !  3641 !      !
                +--------+--------+    !
        >3608 !  0001  !  3701  !>---+
                +-+------+--------+
        >360C ! !                !
                +-+                !
              !  Statement 3   !
              !                !
                +-+--------------+
        >3640 ! !                !
                +-+                !
              !  Statement 2   !
              !                !        >8370
                +-+--------------+    +----(>37D7)
        >3700 ! !                !      !
                +-+                !      !
              !  Statement 1  +-+      !
              !                ! !<---+
                +--------------+-+
        >37D8 !                !
```
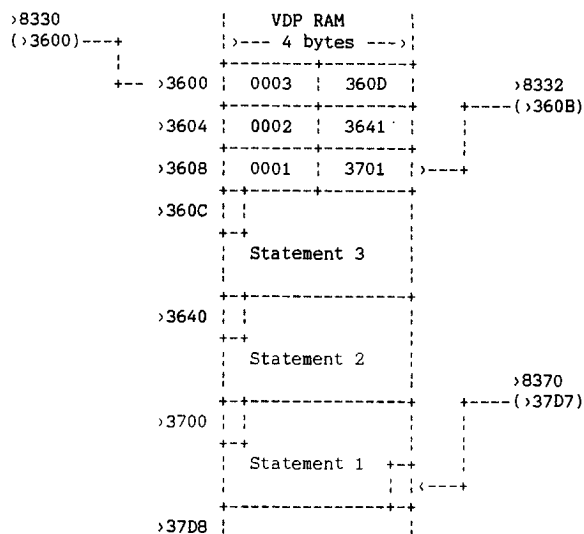
The BASIC statements are not in the form that you typed,
but are "tokenized". In the tokenized form, all BASIC
keywords, operators and punctuation are replaced by
special one byte tokens. Each BASIC statement is
preceded by a one byte length and the last byte in
each statement is always a byte of zeros. The line
or statement numbers for the BASIC statements are not
part of the statement, they exist only in the line
number table.

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

Now if you were to save this program to disk (or cassette)
BASIC adds 4 words in front of the line number table and
writes it out just the way it is in VDP RAM (that's why
you hear PROGRAM type files sometimes called memory image
files). The four words added are :
1. A check word which is the "exclusive or" of
   words 2 and 3.
2. The value from >8332, the VDP RAM address of the
   last byte in the line number table.
3. The value from >8330, the VDP RAM address of the
   first byte of the line number table.
4. The value from >8370, the VDP RAM address of the
   last byte of the statement area.

PROGRAMMERS

* Programming is an inherently difficult and demanding
  task.

* Coding, the expression of the solution in a
  particular programming language, is only one small
  facet of programming and arguably not the most
  important one.

* Programming is a discipline to which certain
  principles may be applied independent of computer
  type, language or application.

* There are objective criteria for good programs; it
  is not a matter of personal aesthetics.

* Despite the appearance of correctness of solution as
  a "syne qua non", the prime criteria in judging a
  program are structure and style rather than
  "cleverness" of the algorithm.

* Given that programs process data, an understanding
  of the nature and structure of the data is an
  essential prerequisite to the construction of any
  program.

* A programmer must be familiar with as many
  techniques as possible, aware of their pros and cons
  and able to recognise situations for their
  application(modified where necessary).   o

Continued from P15

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

```
                         PROGRAM ON DISK
-----           +--------------------------+    ----
        >0000 :    000B  :   360B  :
                +--------+--------------+       Header
        >0004 :    3600  :   37D7  :
                +--------+--------------+    ----
  S     >0008 :    0003  :   360D  :
  E             +--------+--------------+       Line
  C     >000C :    0002  :   3641  :          Number
  T             +--------+--------------+       Table
  O     >0010 :    000?  :   370?  :
  R             +--------+--------------+    ----
  1     >0014 : : :
                +-+
                :       Statement 3      :
                :                        :
                +--+-------------------+
        >0048 : :                      :      Statement
                +-+                     :      Area
                :     Statement 4       :
  ----          :                       :
                +--+-------------------+
        >0108 : :                      :
                +-+                     :
  S             :     Statement 1  +-+
  E             :                  : :
  C             +--------------------+-+    ----
  T     >01DF :                      :
  O             :       Unused         :
  R             :                      :
                :                      :
  2             :                      :
                :                      :
-----           +--------------------+
```

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

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

$$\text{VDP RAM Address} - \text{Word 3} + 8$$

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

$$>3641 - >3600 + 8 = >0049$$

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

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

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

Of course I've done it !

o

'Sometimes I feel people don't care a jot about me...'

I am slowly recovering from micro-surgery in Wollongong hospital. My flagging spirits have been somewhat boosted by a letter from Adam Sutherton of Oak Flats. Thanks Adam. It is certainly an interesting rocket.
I am looking forward to getting more letters by next month.

Jenny.

Dear Jenny,
    Hope you are feeling better. I've drawn a picture on Graphx to cheer you up. It's the first drawing I've done on the computer. I'm 13 years old and I like to play action and Adventure games. I like Dig Dug, Pole Position, Big Foot, and Driving Demon. I wish I had a Speech Synthesizer so I can hear the computer talk.

AND LAST BUT NOT LEAST PLEASE MEET POLICE CONSTABLE DOS WHO (GOODNESS! COULD THIS BE A PERFECT MATCH?!) HAS AN IQ OF JUST 2.1!...

FROM BASIC TO EXTENDED BASIC

by Ben Takach

The convenience of auto-boot from DSK1 at power up, the immediate access to exp.mem and the very much improved language structure of XB makes it superior to the very basic console BASIC.

Running a BASIC program in XB mode generally involves little, if any, program modifications. One may get a bad value error due to a CALL CHAR or CALL COLOR subprogram. Neither of these are hard to correct, since both are well documented in the XB manual.

On the other hand, conversion of an XB program generally means a complete rewrite. This is not only due to the multiple statement option of XB, but also the numerous added choices of like statements (e.g. compare IF THEN ELSE, DISPLAY options and additional statements such as ACCEPT, LINPUT etc.).

These differences are by and large transparent to the user with the majority of the TI issued programs. These were well written and will run in either of the two languages. The compromise of course increases the running time. This did not matter much. Since TI did not issue any competitive commercial class programs. One which comes close is the disk based Lease/Purchase Decisions (model PHD5038). This was a valiant effort by Ronald Zussman to create a commercial class program in console BASIC. He used every trick of the trade to make the inadequate language do the near impossible. Well, apart from a few little bruises he nearly succeeded.

Let us look at his constraints:

—the package consist of 3 independent but very similar programs, these would have been rolled into one, given adequate memory space;

—even 1 of the three program has barely enough RAM space (need to start with a CALL FILES(1) command before running it);

—the programs are excruciatingly sssllooowww ... running due to the frequent garbage collecting excursions and the BASIC screen dump. Again forced upon him by the memory restriction;

—the prescan delay is very long (at first I was convinced my system had locked up).

Considering all this, one understands why this release never made the headlines. Any one of the three programs will take about 2 hours to run from start to finish —if one makes no incorrect entries. The program has no oops provision, it is a case if you make a mistake then start again.

To sum it up, you would never chose this software to run your business if you made a living out of leasing. On the other hand the program is far too complex to aid one in the monumental decision to lease or buy when the old faithful family sedan makes those expensive noises which signals the beginning of the end (unless you are really desperate to impress your uncooperative bank manager with some non comprehensible expressions like internal rate of return). The most impeccable analysis will not lead you to the pot of gold at the end of the rainbow, but at least it will ensure a smooth' ride towards bankruptcy .

From my point of view, it is a good example to highlight some of the less well known differences between TI-BASIC and EXT.BASIC. The programs are well worth converting to XB, to minimise the initial frustration and to postpone the agony.

The CALL FILES enigma.

The user's manual of DM2 clearly states that CALL FILES may not be used as a statement in BASIC. How about in XBasic? The XB manual makes no reference to this subprogram at all! If CALL FILES are included in an XB program you will be greeted by the error message: Subprogram not found in nnn. However the LOAD program of PHD5029, Cash Management, a more recent TI disk program written in XB contains these lines:

```
340 ON ERROR 360
360 CALL FILES(1)
365 RE   370
370 RUN "DSK1.CM100"
```

The above lines raise some questions.

a. What kind of error was anticipated in a load program, which neither asks for nor accepts any entries from the keyboard?

b. What would be the likely response of the 99/4A if the lines 360–365 were encountered?

No doubt you would like to know the answer. So would I. So far I have been unable to find a way to successfully execute a CALL FILES statement from an XB program.

The Bracket puzzle.

Decisions program should run OK from XB. Indeed it did until line 1510 was encountered. I was face to face with an error message never seen before on my screen: Stack overflow in 290 called from line 290, 320, 1510.

These are the offending lines:

```
280 DEF
B$(I)=STR$(INT(I*100-.5*((I>0)-(I<0)))/100+.001)
290 DEF C$(I)=SEG$(B$(I),1,LEN(B$(I))-1)
300 DEF D$(I)="$"&C$(I)
310 DEF C(I)=25-LEN(STR$(I))
320 DEF E(I)=28-LEN(C$(I))
330 DEF H(I)=1+E(I)

1510 PRINT #PF:"A.ASSET COST";TAB(E(PV));D$(PV):
"B.SALES TAX(%)";TAB(H(COM));C$(COM):"C.LEASE
TERM";TAB(C(N));N
```

It was found after line 1510 was temporarily REM–ed that the subsequent 5 lines would also invoke the same error message. These are:

```
1520 PRINT #PF:"D.PREPAYMENT" ;TAB(E(PRE));D$(PRE):
"E.LEA SE PAYMENTS":"F.ADDED OPERATING":" COSTS IF
BOUGHT"

1530 PRINT #PF:"G.LOAN";TAB(E(LOAN));D$(LOAN):"
RATE(%) ";TAB(H(R));C$(R):" MONTH AFTER
PURCHASE";TAB(C(DY));DY

1540 PRINT #PF:"H.COST OF
CAPITAL(%)";TAB(H(K));C$(K):"I .TOTAL TAX
RATE(%)";TAB(H(TAX));C$(TAX):"J.ITC(%)";

1550 PRINT #PF:TAB(H(ITC));C$(ITC):"K.DEPRECIATION
METHO D";TAB(C(DEP));DEP:"L.DEPRECIATION
LIFE";TAB(C(L));L

1560 PRINT #PF:"M.BOOK VALUE(%)";TAB(H(BK));C$(BK)
:"N.SA LVAGE VALUE(%)";TAB(H(SALV));C$(SALV)
```

Finally, yet one more PRINT line was causing a premature program termination:

```
4300 PRINT TAB(4-LEN(STR$(A)));A;TAB(16-LEN(C$(T*B)));
C$ (T*B);TAB(E(-X));C$(-X)
```

Looking at these lines, it is obvious that the 99/4A ran out of fingers and toes; it was suffering from parentheses indigestion.

Continued from P17

This led me to the conclusion that TI-BASIC allows a higher level of parentheses than XB. Searching through the manuals is futile. There is no mention of the work space limitation of BASIC or XB. Simply, if you exceed it, you will get an error message. Further, there is no specific error message for stack overflow in BASIC. This error will be put under the umbrella of the all encompassing memory full message, which is not very helpful,considering BASIC does not understand the SIZE command. So it is up to you to fathom out the real roots of the evil.

Solving the problem. Complex DEF statements which cause stack overflow error or memory full error in BASIC should be simplified or converted to variables.

As an example we shall insert a subroutine of variable conversions for the benefit of the affected lines 1510–1560 and 4300, and of course rewrite these lines. The original DEF lines will be left standing for the benefit of the other program segments which use them without any problem. At the same time the value of DEF-ining a function will also be evident. Programming is more flexible and far more compact.

These are the additional lines (the new subroutine):

```
331 GOTO 340
332 BB$(W)=STR$(INT(P*100-.5*((P>0)-(P<0)))/100+.001)
333 CC$(W)=SEG$(BB$(W),1,LEN(BB$(W))-1)
334 DD$(W)="$"&CC$(W)
335 EE$(W)=28-LEN(CC$(W))
336 HH(W)=1+EE(W)
337 RETURN
```

We have duplicated the DEF-ined functions as a variable (of course changed the names to make it legal). The offending PRINT lines have been rewritten:

```
1510 P=PV :: GOSUB 332 :: PRINT #PF:"A.ASSET
COST";TAB(EE(W));DD$(W):: P=COM :: GOSUB 332 :: PRINT
#PF:"B.SALES TAX(%)";TAB(HH(W));CC$(W):"C.LEASE
TERM";TAB(C(N ));N
```

```
1520 P=PRE :: GOSUB 332 :: PRINT
#PF:"D.PREPAYMENT";TAB(EE(W));DD$(W):"E.LEASE P
AYMENTS":"F.ADDED OPERATING":" COSTS IF BOUGHT"
```

```
1530 P=LOAN :: GOSUB 332 :: PRINT
#PF:"G.LOAN";TAB(EE(W));DD$(W):: P=R :: GOSUB 332 ::
PRINT #PF:" RATE(%)";TAB(HH(W));CC$(W):" MONTH AFTER
PURCHASE";TAB(C(DY ));DY
```

```
1540 P=K :: GOSUB 332 :: PRINT #PF:"H.COST OF
CAPITAL(%)";TAB(HH(W));CC$(W):: P= TAX :: GOSUB 332 ::
PRINT #PF:"I.TOTAL TAX
RATE(%)";TAB(HH(W));CC$(W):"J.ITC(%)" ;
```

```
1550 P=ITC :: GOSUB 332 :: PRINT
#PF:TAB(HH(W));CC$(W):"K.DEPRECIATION METHOD";T
AB(C(DEP));DEP:"L.DEPRECIATION LIFE";TAB(C(L));L
```

```
1560 P=BK :: GOSUB 332 :: PRINT #PF:"M.BOOK
VALUE(%)";TAB(HH(W));CC$(W):: P=SALV :: GOSUB 332 ::
PRINT #PF:"N.SALVAGE VALUE(%)";TAB(HH(W));CC$(W)
```

```
4300 P=T*B :: GOSUB 332 :: PRINT
TAB(4-LEN(STR$(A)));A;TAB(16-LEN(CC$(W)));CC$(W );::
P=-X :: GOSUB 332 :: PRINT TAB(EE(W));CC$(W)
```

There are not many programs where these lesser known differences between BASIC and XB may be demonstrated out of context. On the other hand, reproducing the entire program would take up far too much valuable magazine space. In this respect R. Zussman's program offered a perfect object for our tutorial.

This program is called LEASE-BUY, the other two on the disk are named LESSOR and INVESTOR. These may be converted using the same technique and almost identical program lines, however the line numbers differ slightly.

FROM MY OWN RED FACE DEPARTMENT.

by Ben Takach

Errare humanum est –as the Romans said it. Which proves the point that human errors were with us for a very long time.
Well I have to admit that my keyboard matrix drawing published in the April issue of the TND does not show the semi-colon key. TI was not very generous with its keyboard design, and to top it off I managed to misappropriate one.
Below is the corrected keyboard matrix.
Please accept my apology!
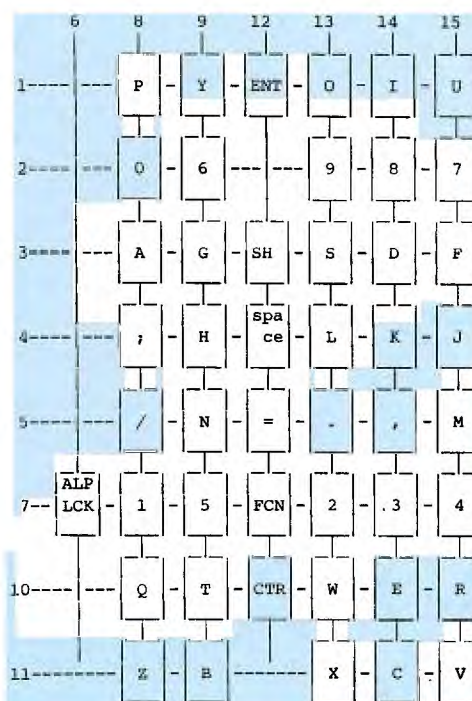
The key-board matrix of the TI-99/4A.
The joystick port connections are not shown.

• • • • •

'I'm sorry, Mr Finsbury, but according to our computer you're dead.'

REGIONAL GROUP NEWS

SUTHERLAND GROUP REGIONAL REPORT

The March meeting saw a small but enthusiastic group
in attendance and a special welcome to a new member in
Ron Stewart.

Some time was spent in reviewing a collection of
cassette based assembly language games, which were
kindly supplied by the Illawarra Regional Group.

Our group is currently endeavouring to build up a
peripheral expansion system from available parts and
is interested in purchasing an RS232 card to fit in the
expansion box. Please phone Peter Young on 5288775 if
you can assist.

Monthly meetings are held on the third Friday of each
month starting at 7.30pm, all welcome.

Regards
Peter Young

ILLAWARRA Regional Group.

Regular meetings are held on the third Monday of each
month starting at 7.30pm at Keiraville Public School,
Gipps Rd, Keiraville. Opposite Keiraville Shopping
centre.

CARLINGFORD Regional Group.

Regular meetings are third wednesday of
each month.

LIVERPOOL REGIONAL GROUP –
Contact Arto Heino 603-8956 for Meeting info.

BANKSTOWN Regional Group.

For information contact Peter Pedersen, (02) 772 2396.
Meetings are held at 11 Bastille Close, Padstow
Heights.

GLEBE Regional Group.
7th May 1987, 8pm, 43 Boyce St, Glebe Contact Mike
Slattery, 692 0559.

Regular meetings on the Thursday evening following the
first Saturday of the month.

CENTRAL COAST Regional Group.

Meetings are normally held on Second Saturday of each
month in Bugewoi area.

Contact Russell Welham (043 92 4000)

CALL PLOT   by Arto Heino

If you still have only a bare console and an XB
cartridge, you will be delighted with this plotting
routine. The routine is slow but quite adequate for
plotting maths functions or statistical data or your
favourite cartoons.

```
1 !*********************
2 !  CALL PLOT in XB   *
3 ! revised by A.Heino *
4 ! T.I.s.H.U.G.  1987 *
5 !*********************
6 !CALL PLOT(X,Y)       *
7 !     X=1 TO 256      *
8 !     Y=1 TO 192      *
9 !*********************
10000 SUB PLOT(V,Q):: IF I T
HEN 10010
10005 I=1 :: H$="0123456789A
BCDEF" :: N$="00000001001000
11010001010110011110001001 10
10101111001101111011111"
10010 X=INT(Q):: Y=INT(V)::
IF Y<0 OR Y>256 OR X<0 OR X>
192 THEN SUBEXIT
10050 C=C-(C=0)*143 :: W=INT
((X-1)/8)+1 :: Z=INT((Y-1)/8
)+1 :: E=X-((W-1)*8):: F=Y-(
(Z-1)*8):: CALL GCHAR(W,Z,A)
:: A,D=MAX(32,A)
10060 CALL CHARPAT(A,A$):: P
=E*8-8+F :: @=INT(P/4.06)+1
:: F=F+(F>4)*4
10070 IF A<>32 THEN C=C+1 EL
SE A=C :: IF C<34 THEN SUBEX
IT
10090 B$=SEG$(A$,@,1):: IF S
EG$(SEG$(N$,POS(H$,B$,1)*4-3
,4),F,1)="1" THEN IF D=32 TH
EN SUBEXIT ELSE 10200
10100 CALL CHAR(A,SEG$(A$,1,
@-1)&SEG$(H$,POS(H$,B$,1)+2^
(ABS(F-4)),1)&SEG$(A$,@+1,16
-@)):: CALL HCHAR(W,Z,A)
10200 C=C-1 :: C=C-(C=0):: S
UBEND
```

Add these routines for more plotting power:

```
10500 SUB CIRCLE(X1,Y1,R,Q):
: L=INT(Q):: G=(Q-L)*10 :: G
1,L1=1 :: IF L>G THEN G1=G/L
ELSE IF G>L THEN L1=L/G
10510 FOR N=0 TO 2*PI STEP P
I/(R*2.2):: CALL PLOT(X1+(R*
COS(N))*G1,Y1+(R*SIN(N))*L1)
:: SUBEND
```

```
X1,Y1 = CENTER OF CIRCLE
   R = RADIUS
   Q = RATIO 1.1=CIRCLE
              1.2=OVAL
              ^ ^
   1 to 9 ___| |___ 1 to 9
```

```
10600 SUB LINE(X1,Y1,X2,Y2):
: X1=INT(X1):: X2=INT(X2)::
Y1=INT(Y1):: Y2=INT(Y2):: X3
,X4,X5,Y3,Y4,Y5=0
10610 X4=(X2-X1)*-(X1<X2)+(X
1-X2)*-(X1>X2):: Y4=(Y2-Y1)*
-(Y1<Y2)+(Y1-Y2)*-(Y1>Y2)::
IF Y4=0 THEN X5=1 :: Z1=ABS(
X1-X2)
10620 IF X4=0 THEN Y5=1 :: Z
1=ABS(Y1-Y2)ELSE IF Y4>X4 TH
EN X5=X4/Y4 :: Y5=1 :: Z1=Y4
10630 IF X4>Y4 THEN Y5=Y4/X4
:: X5=1 :: Z1=X4 ELSE IF X4
=Y4 THEN Y5,X5=1 :: Z1=X4
10640 FOR Z=1 TO Z1 :: CALL
PLOT(X1,Y1):: X1=X1+X5*X3 ::
Y1=Y1+Y5*Y3 :: NEXT Z :: SU
BEND
```

```
X1,Y1 = START OF LINE
X2,Y2 = END   OF LINE
```

**   *   *