HUNTER VALLEY 99'ERS NEWS

TI 99/4A

HOME COMPUTER NEWSLETTER

OCTOBER 1987

TEXAS INSTRUMENTS
Newcastle
& The Hunter Region
Home Computer
USERS' GROUP

REGISTERED BY AUSTRALIA POST
PUBLICATION No NBG6023

The Secretary - HV'99ERS
6 Arcot Close, TARRO - NSW.
Australia - 2322

YOUR COMMITTEE

PRESIDENT
Paul Mulvaney
26 Marmong St.,
MARMONG POINT 2284
Ph. 583623

VICE PRESIDENT
Alan Lawrence
35 Bayview St.,
WARNERS BAY 2282
Ph. 486509

SECRETARY
Albert Anderson
6 Arcot Close,
TARRO 2322
Ph. 662602
Viatel 496626020

TREASURER
Peter Smith
8 Glebe St.,
EAST MAITLAND 2322
Ph. 336164
Viatel 493361640

SOFTWARE LIBRARIAN
Alan Franks
822 Pacific Hwy.,
MARKS POINT 2280
Ph. 459170

PUBLICATIONS LIBRARIAN
Allen (Joe) Wright
77 Andrew Rd.,
VALENTINE 2280
Ph. 468120

EDITOR
Brian Woods
9 Thirlmere Pde.,
TARRO 2322
Ph. 662307
Viatel 496623070

PURCHASING CO-ORDINATOR
Bob MacClure
75 Deborah St.,
KOTARA SOUTH 2288
Ph. 437431

TECHNICAL CO-ORDINATOR
Gary Jones
53 Janet St.,
JESMOND 2299
Ph. 573744

SIGs CO-ORDINATOR
Brian Rutherford
9 Bombala St.,
DUDLEY 2290
Ph. 498184

CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.

Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX 80 or DIS/VAR 80). A suitable Public Domain word processor program will be supplied if required by the club librarian.

Please include along with your article sufficient information to enable the file to be read by the Editor eg. File Name etc. The preferred format is 35 columns and page length 66 lines, right justified.

All articles printed in HV99 NEWS (unless notified otherwise) are considered to be Public Domain. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.

Articles for publication can be submitted to the Editor, ALL other club related correspondence should be addressed to The Secretary.

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY NINE USER GROUP.

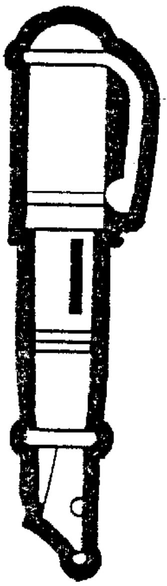
Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result of applying such information.

The views expressed in the articles in this publication are the views of the author/s and are not necessarily the views of the Committee, Editor or members.

TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.

HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS.

PRESIDENT'S



It is heartening to see the membership renewals keeping pace with last year, there is still a reason to keep the User Group going. This lead me to the question of what is the real purpose of the Users Groups where ever they may be. Well, as the name implies they are for users of computers to share their knowledge and allow everyone to gain the maximum benefit from their machine. The type of help varies from hardware to software. The hardware improvements allow the machine to do more and more for its owner, provided the owner is prepared to outlay the money to purchase the additional equipment.

The software improvements are not quite as tangible as hardware but never the less they also allow the owner to do more and more also. Software improvements come in many varieties, from simple routines in UG newsletters such as this, right through to multi-disk commercial programs.

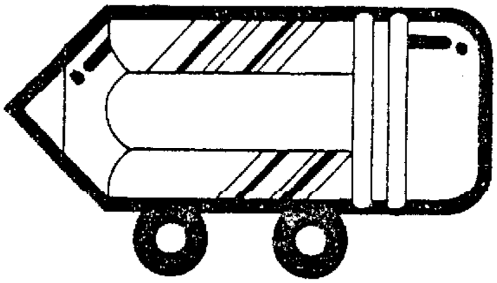
The development of improvements - both hardware and software - takes significant amounts of time and some developers feel they are entitled to some monetary return for their efforts. If we as users want to continue to be able to improve our machines we should encourage the developers by supporting their efforts and actively discouraging unauthorised copying (PIRACY) of their efforts.

The policy of the Hunter Valley 99'ers Users Group is that no commercial or copyright material be circulated. The Librarian has in the past, and will continue, to reject any such material which is sent to him. Our Library contains only material that is freely available. While we cannot control the actions of individuals this has been and will continue to be the policy of HV99'ers Users Group.

If we support developers they will be encouraged to go on to better things, but if we rip them off, then support for our beloved orphan may well disappear.

On a lighter note I received a letter from Bob Carmony the other day. Bob would like to correspond with members, so if anyone would like to write to Bob his address is;

Robert M. Carmony
1504 Larson St.
Greensboro. N.C.
27407



FROM THE EDITOR

I would like to thank the contributors to this month's Newsletter - I think it is possibly one of the best ever!! Special thanks to 3 new contributors; Kevin Cox for his article on the Mini-Memory (a most under rated module), and the other 2 'shy' contributors who prefer to hid their light under a bushel. I hope these prove to be the first of many articles you write for our Newsletter.

On the subject of writing articles, don't forget that Christmas is coming and with it (as well as the expense) comes the Bumper Christmas Issue of the magazine. PLEASE TRY TO CONTRIBUTE SOMETHING for that issue - why not practice with an article for next month? As well contributions are solicited for the February 1988 issue - don't forget there is no January issue, and the first issue after the break is always hard to fill so please help me out.

Enough pleading for now but please try and contribute something SOON.

RANDOM BYTES

WITH

BOB CARMANY

What to do this month? That is always an interesting question! I guess the first thing that I had better do is answer some questions that have come up. Several months ago, my old "trusty" STAR STX-8Ø went to the microchip graveyard. In my search for a replacement, I looked at several printers in to $15Ø - $2ØØ (US) price range. I finally settled on the STAR NP-1Ø. It turned out to be an excellent choice! It was designed to replace both the GEMINI 1ØX and the SG-1Ø -- which it does quite admirably. It has a full range of print styles from superscript, subscript, and condensed to fully enlarged characters. It will print in NLQ mode (yes, Albert, that is standard NLQ Pica) at 25 CPS or draft mode at 1ØØ CPS. Suffice it to say that it is a truly outstanding bargain! Hopefully, there should be a full review of it in one of the upcoming issues of MICROpendium.

One more item of note before we go on to something else. Marty Kroll has come out with an updated version of his outstanding DISK CATALOGUE program. It is updated through June of '87 and has some excellent new features. By the time that you read this, there should be a copy available in the UG library (I've sent it along with this series of articles). The sort is now lightening fast!

Before I get involved in another series of tips, let me tell you a little bit about the system that I work with. Years agao, I elected to go with standalone peripherals rather than the PEB. As a result, my system is composed entirely of standalones. My console is one of the "black and silver" '83 productions. Next to it I have a 32K from Captain's Wheel (ask Tony McG. about them). It has an extra bank of memory at >6ØØØ (cartridge port) selectable with a switch. That allows it to act as a "SuperCart" and accept any program that has been AORGed into that area. Next in line is a speech synthesizer and then a TI PHP 18ØØ standalone disk controller. I am currently running two SSSD disk drives --one TI (Shugart) and one Tandon 1ØØ-1. One the other side of the Controller is a Doryt Systems Paraprint 18A PIO port with my STAR NP-1Ø attached. It isn't as long as it sounds. Oh yes, I have two Prostick II joysticks attached as well!

Now, for a few "goodies". Very few XB programmers make full use of the error handling routines that are available when they write a program. About a year ago, I did a little article for MICROpendium about just that. It might be well to look at what can be done to make your programs more "elegant" and user-friendly. Let's see what we can do with ON ERROR.

Assume that you are writing a program that asks for a filename at a prompt. If you are like most of us, sometimes you either forget to have the correct disk in the drive or just make a typographical error when you enter the filename. What happens then is that the program "burps" up an error message and halts execution --- very UNSATISFACTORY!! Let's see what we can do about it. Here is a portion of a program that addresses just this problem.

```
320 DISPLAY AT(5,3):"Enter Text Fil
ename" :: ACCEPT AT(7,7)SIZE(10
):D$

330 ON ERROR 560
                 q
340 OPEN #1:"DSK1.^"D$.INPUT , DISP
LAY , VARIABLE 80

560 CALL SCREEN(7) :: DISPLAY AT(23
,1)BEEP ERASE ALL:"DRIVE/FILENA
ME ERROR" :: FOR DELAY=1 TO 500
:: NEXT DELAY :: CALL SCREEN(8)
:: RETURN 320
```

Let's analyze what we have done with these few lines of code. Line 32Ø is the input line. Line 33Ø shifts control of the program to the error handling routine should the filename or file parameters not match those of the OPEN statement in line 34Ø. The ON ERROR statement must precede the line likely to generate an error message (in this case, the actual opening of the file).

If an error occurs, line 56Ø handles the error. Instead of the program "crashing" when an error occurs, the program shifts control to line 56Ø. Line 56Ø chages the screen color to red, displays the DISK/FILENAME ERROR" message (with a delay so it can be seen), and then returns to line 32Ø to allow you to once again try to enter the correct input. It may not be the most efficient way to handle the problem but it works.

If you want to disable the <FCTN-4> key in a program, insert the statement ON BREAK NEXT at the begining of your program. To see how all of this works, refer back to the July issue of the HV99 newsletter and my FILE/READ program. Both were used in that program to make everything run more smoothly.

I'm getting another "BUFFER FULL" message and it's time to close for another month. 'Til next time . .

THE VIEW FROM THE WEST

NEWS FROM OUR WESTERN AUSTRALIAN CORRESPONDENT

This is the first of what may, at some future time, become a sort of irregular series of gossip, rumour-mongering and hear-say on the TI scene here in Western Australia. Let me begin by stating just one thing: ANYTHING in this column,no matter how scurrilous,scandalous or outrageous is the OPINION of the author. These ramblings will be based on information at hand. Nothing untrue will be knowingly passed on. These jaundiced views of the local scene MAY be informative, may entertain (and not be deliberately offensive) BUT THEY ARE NOT AN OFFICIAL INPUT FROM ANY USERS' GROUP. I repeat that the opinions expressed are strictly my own.

Well, now that I've gone through the mandatory disclaimer, where do I start a 'sort of a series'? In W.A., as anywhere else in the TI world I guess, there appears to be an enormous amount of 'underground' TI-99/4As. From a high of almost 1ØØ members, TI-UP had, at last announcement,around 3Ø financial members. Since it is rare to see used TI equipment in the For Sale columns of the local newspapers,and even more rare to see used equipment in the local 'VicWest' or '64 Shop', one can only assume that a large proportion of the miriad TI-99/4As sold in the West are hibernating somewhere.As Bernie Elsner once suggested that the ratio of TI owners to TIUP members was anything up to 5:1, it is safe to assume that there are an awful lot of potential members out there somewhere. But how do we reach them? How do we let them know about the wonders of Funnelweb, in-console memory expansion for about $5Ø.ØØ plus a visit to a Users' Group Workshop, RAM disks, the Schubert/Paine expansion system et al? Let's face it, full expansion with greater capabilities than the TI original for around $3ØØ.ØØ plus disk drive is a great deal that I wish were available when I started expanding my system. Next, I must make a clean breast of things...yes, it IS true that at the TIUP AGM held earlier this year, a motion to change the Club's direction to embrace MS-DOS machines was passed. This was seen as inevitable by all but the six dissenters present,and at least one member has flogged his '4A and ordered a MS-DOS machine since that fateful day.We have also had one potential member who came along in anticipation of seeing the MS-DOS magic in action. At the September general meeting there were two MS-DOS and one TI System in attendance which is a BIG improvement on some meetings that had NO machines available.

On the local hardware scene, we have sporadic reports of items appearing in various Big W Stores across the State...a Speech Synthesiser in Belmont,a joystick adaptor in Bunbury etc, but no more than the odd single item. At a recent TIUP meeting mention of the availability of the latest Schubert/Paine progeny was made, but the announcement was neither delivered nor received with anything that resembled enthusiasm. It appears that RAM disks are a non-goer here,as there have been only two confirmed reports of operational RAM disks to my knowledge.

Unfortunately some more of our members have been seduced by other machines(one has admitted to buying an Amiga) as there were two expansion systems offered for sale at the September TI-UP meeting. That same defector also offered his Adventure Game collection (the Zork Trilogy, Hitch Hiker's Guide)for sale. This was immediately snapped up by the Committee for the Club Library. As far as software goes,our very own Phil West has demonstrated his talents with a programme in c (that's Clint Pulley's "Small c") and a VERY FANCY Viatel programme with 3 sizes of screen dumps and the ability to do wonderful things with TI Writer files.This particular product is currently out for 'beta-testing' and should be ready for sale soon. A very clever lad is our Phil,let's hope that he doesn't get lost to the world of MS-DOS. The local availability of F'web V3.4 has also been announced,and the distribution disk courtesy of Albert A has been passed on for redistribution.One suggestion is a chain letter arrangement similar to the HV 99ers great newsletter mail-out.Lets hope that the McGovern clan get some tangible recognition for their efforts.I know that I've sent a contribution...

A demonstration of Arto Heino's "Picasso Publisher" was given at the September meeting,and President Bernie presented some sample print-outs that he had prepared prior to the meeting.It certainly seems to be a good programme for those of us who would like to try their hand at a TI version of Desk Top Publishing.

Enough ramblings this time, maybe some more at a later date (if I'm not run out of town!!).
Must go... bye for now..........

The Western Flyer

READERS CORNER

WITH
JOE WRIGHT

Well another month is here, much warmer than the LAST, perhaps we can now say; "winter has PAST".

There has been much ado in the Group this past month, hence heaps to report.

What has come in this month to the PUBLICATIONS LIBRARY? At the time of writting we have received;
TI-LINES-(JUNE),
OTTAWA-(SEPT),
BITS,BYTES PIXELS-(LIMA,SEPT)
CHUG-A-LUG-(CANBERRA,SEPT),
BUG,BYTES-(BRISBANE,SEPT),
CIN-DAY-(OHIO,AUGUST),
BUG NEWS-(BREA,CALIFORNIA,SEPT)
ROM-(ORANGE,CALIFORNIA,SEPT),
TISHUG-(SYDNEY,OCTOBER)
TID-BITS-:MEMPHIS,SEPTEMBER)

First thing to mention is that our world wanderer Geoff Shipton met up with Ken Hamai from BREA and ROM. They were both attending an annual computer swap in Hawaii at which the ALOHA TI USERS GROUP sponsored a stand. Ken Hamai reports in the BUG NEWS that the ALOHA GROUP put on a marvelous display which attracted much positive comment.

While on the topic of World Wanderers, Richard Terry, our Forth Writer is also away from these shores. In typical Richard fashion 6 post cards turned up at his Surgery within days of one another. They were from Hong Kong, Amsterdam, London, Edinburgh, some where in Wales and the last from somewhere in the Mid Lands region.

One other MISSING person has also been tracked down. Steve Taylor has now settled into his new home down in Tasmania. We had some difficulty with phone numbers and could not contact Steve. All that is sorted out. The trusty TI is about to be taken out of it's hiding place and once again dominate the study. Look forward to a steady flow of letters etc, Steve!! *(& maybe an article or two as well - editor)*

For all the Midnight modem set who have been waiting, Roger Merrit has his second article on printing graphics through the TI Writer Formatter in the september edition of ROM. Once again I have additional copies available on request. He mentions in his article that TRIO+SOFTWARE has a programme called "ARTCONVERT" which converts "TI-ARTIST" instances into TI-WRITER files. It is available through TENEX for US$9.95.

The Publication librarian for one Group is considering ways of getting people to return borrowed materials, publications, etc. He wants to impose some form of penalty for non return. That sounds like a good idea which we could adopt. I have suggested a range of penalties as follows, their imposition would depend on some scale of size of misdemeanour;

1) Play 2Ø games of CAR WARS.
2) Four sessions with RODNEY on how to get through HITCHHIKER.

Now the TOUGHEST!

3) ATTEND 3 COMMITTEE MEETINGS.

Option 3 does not include travel costs to and from places far removed from Newcastle. You never know what the Midnight modem set will try on!!

I expect to be inundated with returned items if the above is adopted.

FUNNELWEB.

Charles Good writing in the September BITS,BYTES AND PIXELS has a very good article on "HOW TO CUSTOMIZE THE FUNNELWEB 'USER LIST'". It is very good - just what I needed. While about instructions for TI Writer/Funnelweb, TISHUG had in it's September News Digest a comprehensive listing of Writer commands.

MORE ON 'C'.

John Susco the CIN-DAY Times Editor has in the August CIN-DAY Times a LABEL programme to type in. He has included instructions on the proceedure for getting the 'C' programme assembled. "The programme is different to the LABEL programme Clint Pulley had on his diskette."

GENEVE- MY WORD.

A review of the word processor for the GENEVE written by Jane Laflamme in the OTTAWA Newsletter is interesting reading. The word processor has many useful features. One of the most useful as I see it is the ability to print the document to screen through the Formatter, (8Ø column screen of course), saves heaps of paper. Switching between Editor and Formatter is easy since they are both resident. You should read this article, it is sure to make you green with envy.

Also in the same Newsletter is an article on speeding up Ext Basic by Lucie Dorias, some very surprisingly fast routines are shown and explained.

RESET SWITCH/NO RESET CARTRIDGE INSERT MODIFICATION.

From the West Penn 99's Club via the September CHUG-A-LUG. Comes an interesting article with accompanying sketch; it shows the proceedure to install a RESET/RESET INHIBIT switch on the console. It has the now well know ability to reset the console. In addition the switch can be used to inhibit the console reset when a module is inserted into the cartridge port. The installation is relatively simple and would be a useful addition to your console.

Also in the Canberra newsletter is an article by Jason Winter. This time he has instructions for the installation and use of 8K of memory for use with Basic. The 8k block lives at >4ØØØ. It can be loaded with routines which can be accessed from Basic by CALL.

ASSEMBLY SUBROUTINES.

Gary Christensen once again has a fine article on assembly language in the September Brisbane BUG. This month Gary discusses and gives examples of the use of Subroutines. I find myself in total agreement with Gary when he states; "I am a firm believer in the use of subroutines in any language. In assembly, they are almost essential".

FINISH.

That wraps it up for this month, except for one final offering. Attended a 21st birthday last week

During a toast to the guest of honour one speaker mention that we all only get to be 21 once. He then offered this anonymous quotation.

I expect to pass through this world but once. Any good therefore that I can do, or any kindness that I can show to any fellow creature, let me do it now. Let me not defer or neglect it, for I shall not pass this way again.

Joe Wright.

FOR SALE

TI PE Box - grey model, push button on/off. 32K Memory Expansion Card
CorComp RS232 Card
TI Disk Controller
1 * TI SSSD Full height drive
+ 2nd Stand alone TI SSSD Full height drive - won't configure correctly, will read/write but won't initialise disks.

COST: $500 with one drive
$550 with both drives
freight the responsibility of buyer

CONTACT:
Geoff Nussbaum
15 Baird Avenue,
NEDLANDS WA
6009

phone 09-3861411

LIBRARIANS CORNER

I must apologize I have not had time to write anything lately so just a short note to let you know disks are still $12 a packet. The only recent addition to the library is an update of Marty Krolls disk catalogue.

All the best till next month.
ALAN FRANKS

A BULLETIN BOARD SERVICE

FOR THE HUNTER REGION

AN OUTLINE
BY
PETER SMITH

Recently a new **BULLETIN BOARD** service opened in the sleepy Valley town of Paterson.

This means that local residents (especially those living in the **HUB OF THE HUNTER**) now have access to a BBS at local call fees.

The most exciting part of this news is the fact that this BBS is part of a network (world-wide at that) which will allow users to send and receive messages using boards all over the world.

We do not need to have any special equipment, only a modem connected to your RS232 port and a suitable soft-ware program and a phone line.

I have enclosed some information which I copied from the BBS the other day. I hope it will give readers some idea of the way BBS's operate, and hopefully develop an interest in this form of computing.

Note.. I have edited the information to give a brief overview of the facilities and functions available...

Enjoy............Peter Smith
****

MATRIX BBS.
This BBS is run by, horror of horrors, a couple of COMMADORES connected to a hard disk and a BIT-BLITZER modem able to opperate at either 300bd or 1200/1200bd.

Operated By: Andrew Pike [Drew]
Written By : Steve Punter
Version : BBS64.5
PunterNet : Node 3

< MATRIX REGISTRATION ACCESS >

There are two types of access for users of Matrix BBS.

PUBLIC MEMBER: 45min on line full access to the board provided that you login once a month this access is FREE.

SUPORTING MEMBER: No time limit access to higher class software in downloads plus a few more extras Cost $1Ø.ØØ once only.

If you choose to become a supporting Member send your payment to.
ANDREW PIKE PO BOX 341 MAITLAND.

Matrix BBS is part of the Punter Network which is a message sending and reciving network which enables you to send and recive messages from other BBS users who are invoved in the NET Sydney/Canberra/Canada etc, so you are not restricted to users on this BBS.

To be able to send network messages , I need to have a small amout of money (min $5) from you to credit your NET account. The cost of sending a NET message is 1cent a line and the minium charge is 15cents a message.

If you would like to use the network send your advance payment to
PO BOX 341 MAITLAND NSW

PLEASE OPEN YOUR CAPTURE BUFFER NOW AS A LIST OF FULL SYSTEM COMMANDS IS ABOUT TO FOLLOW..

+------------------------+

Functions with nothing between the dashes are typed in exactly has seen.

Those with numbers MAY require parameters. Descriptions of each "group number" follow the command list.

```
* Signing Off
G - - Goodbye (Same as QUIT)
QUIT - - Leave System

* Message Reading Functions
F -1- Forward Message Reading
MAIL - - Read Msgs Addressed to YOU
NEW - - Read New Messages
NEXT - - Continue Message Reading
R - - Recall a Specific Message
R -1- Reverse Message Reading
RALL -2- Read Messages Sent to ALL
READ - - Read Msgs Addressed to YOU
SEL -3- Selective Msg Read (All)
SELF -3- Selective Msg Read (From)
SELS -3- Selective Msg Read
(Subject)
SELT -3- Selective Msg Read (To)

* Message Scanning Functions
O -2- Overview of Messages
S -2- Summary of Available Messages
SCN -3- Selective Summary (All)
SCNF -3- Selective Summary (From)
SCNS -3- Selective Summary (Subject)
SCNT -3- Selective Summary (To)

* Message Entry Deletion
DM -1- Delete a Message
E - - Enter a Message

* Uploading Downloading
DP - - Delete a Program
LIST - - List of Available Programs
LOAD - - Load a Program
LONG - - LIST With File Descriptions
SAVE - - Save Program to BBS
STAT - - Load/Save Status Variable

* Mode Toggles
EXP - - Expert Mode (On/Off)
LF - - Line Feeds (On/Off)
NUM - - Line Numbering (On/Off)

* Odd Functions
HELP - - Reprint This List
B - - Bulletin Section
BAL - - PunterNet Account Balance
BAUD - - Change Baud Rate
CAT - - List Message Categories
CONT - - Output Control
DUP - - Change Duplex
LOG - - System Usage Log
TIME - - Show Date Connect Time
U - - Search User List
```

* Multiple Response Parser

Although commands and responses may be entered one at time, as called for, this BBS allows commands to be "pre-entered" at any prompt. For example, if we wanted to recall message 1ØØ, then forward it, we need only type:

R:1ØØ:F

Notice that the responses for upcoming prompts are added by use of the colon.

Also note that responses to "live key" prompts are also accepted.

* Stopping Transmission

At any time, pressing "S" will halt transmission. It can be restarted by pressing "C". Additionally, most functions can be aborted once stopped by pressing "A". To accomplish this press "S" then "A". To abort quickly, send a ᴗTRL-A. To completely abort ANY function, SUCH AS continuous mode message reading, send a CTRL-P.

```
CAT
Ø - GENERAL
1 - FOR SALE
2 - WANTED
3 - MEMBER
4 - PUBLIC
5 - DEBATE 2ØØØ
6 - PROGRAMING
7 - SYSOP SOAPBOX
8 - TECHNICAL QA
9 - NOTICE BOARD
-------------------------------
<  MATRIX  BULLETIN/MAGAZINE   FILES
-------------------------------
Topic:
-----
Opening bulletin........  OB
Newuser bulletin........  NEW
How to use the Net......  HN
Net Nodes...............  NODES
Lost password proccedure.  LP
System Information......  SINFO
Supporting Member Info..  SM
PAMS list sub directory.  PAMS
MATRIX MAGAZINE Directory.M
Who's New on the system.  WHO
Software Review Directory.SR
Hardware Review Directory.HR
Jokes Directory.........  JOKE
Recipe Directory........  REC
Full User Instructions..  USER
Movie Review............  MOVIE
Network News............  NNEWS
Disclaimer..............  DIS
Sysops newsletter.......  SYSN
Technical Info Directory TECH
Telecommunications......  TELE
Ham News................  HAM
Club Notice Board.......  CLUB
```

SOFTWARE CHAIN UPDATE

It is proposed to send to any interested member 5 disks of public domain software for the cost of $12 incl postage. If you are interested, please contact the Editor as soon as possible so we can get it under way.

S.I.G.S.

Don't forget the BIG SIGS DAY to be held at Warners Bay High School on Saturday 24 October, commencing at 1.ØØpm.

Some of the items on the agenda include:-

A demo by Paul Mulvaney of PICASSO, the TI print shop program.

TI Writer hints and tips with Brian Woods.

Games for the kids.

Adventures with Geoff Gray.

Come along and join in the fun.

MULTI-FUNCTION DAY

On Saturday 28th November a social family day with sausage sizzle/bar b que, games swap & sell and a raffle is being organised. A time and venue is still to be decided, but the raffle prizes have been selected, and tickets go on sale immediately.

FIRST PRIZE is One (1) Speed King Joystick with adaptor, valued at $5Ø. There will also be some other prizes for 2nd, 3rd etc. so BE IN IT!!! Tickets are 3 for $1 and are available from Gary Jones. Members outside the immediate area are invited to send to Gary for tickets (see inside front cover for address). Your tickets will be held here (yes, you CAN trust us) and if you win a prize it will be forwarded to you.

DAY
High
ber,

enda

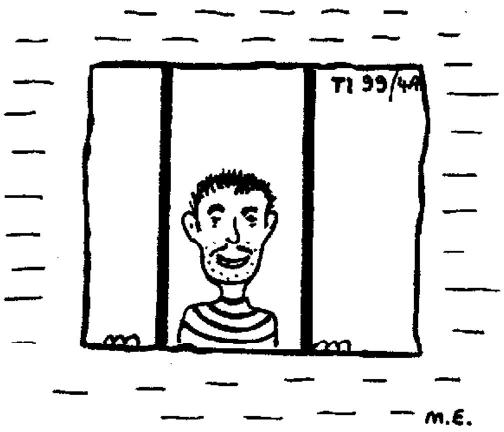
of
am.
with

un.

a
age
ell
A
be
ave
on

eed
ued
ome
so
$1
es.
rea
for
for
eld
if
ded

BREAKOUT



OR HOW I LEARNED TO CURE LOCKUP

BY

LOTHAR NOWAK

Lockup. A word that strikes terror in the heart of TI owners, especially the novices - I include myself in that category!). There you are happily pounding away on the keyboard, feeding those RSI bugs (that is repetitive strain injuries) when suddenly your machine goes WHAMMY. It dies. This does not really mean total disaster, just partial. Your machine has not gone on to the resting ground of old computers. It can be remedied. What has happened is that the program you were working on has passed on but your TI can be saved.

Luckily this has not happened to me and that is probably due more to good luck than good management. Nasty dirt has gotten inside your machine and caused it to foul up the works. Now because this has not happened to me why am I writing this article? I hear you say. There have been some good articles written on this topic but I wanted to write one using what should be a fool proof method.

NOTE: My electronics degree has not yet arrived from Kellogs so getting me to write this article is a masterstroke. If I can do it, so can anyone.

So what do you need?

1. A Phillips head screwdriver.
2. A plastic ruler.
3. Some lint free cloth.
4. A Cotton Bud.
5. Cleaning fluid like alcohol etc.
6. Patience.

Now lets get started.

STEP 1

Clear an area on a table and cover with a blanket or similar material & place your computer face down on it with the keyboard nearest to you. There are 7 screws holding the base; remove these and store safely. Carefully lift off the bottom and note this compares with what is shown in Fig 1.

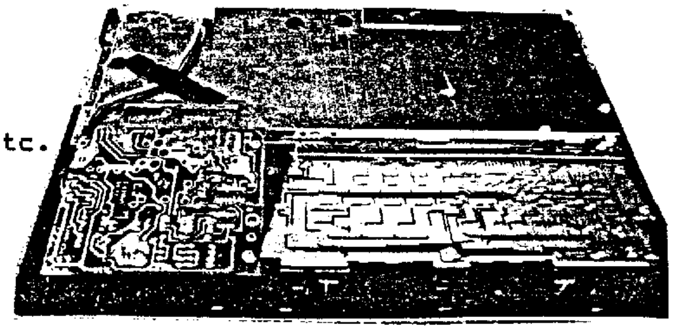


FIG 1

NOTE:

a)key board is on right.
b)power board is on left.
c)main circuit board on top.

STEP 2

Removal of power board (FIG 2).

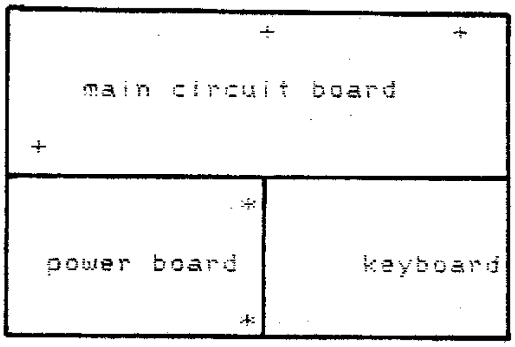
Lift tape from power cable and place power socket on left of computer. Undo the 2 screws(*) from the power board and remove by lifting the board and place on the left hand side of the computer (Fig 3).

Removal of main circuit board.(Fig 2)

Remove the 3 screws(+) from the main circuit board. This is a metallic section and you will note a 15 pin connector when you slightly lift the board up. Prise this connector apart and lay the whole board on the side of the computer. You can now see the villain of the piece. It is the Module Socket which when used can get dirt inside the module, breaking the contact and causing the lockup. (Fig 4)

STEP 3

Looking at Fig 5 shows how to remove the socket module from the board. Care must be taken when you remove this socket so as to not damage it.



Screws to be removed.
+ main circuit board
* power board

FIG 2

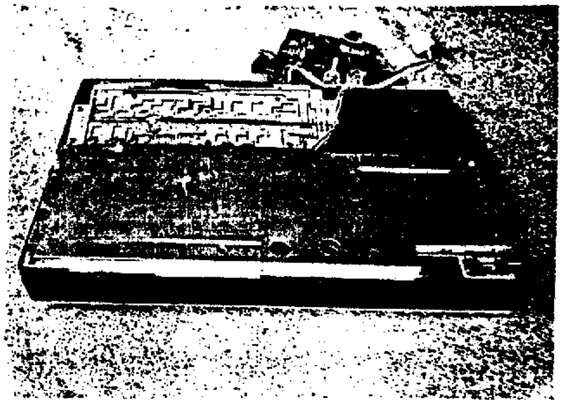


FIG 3

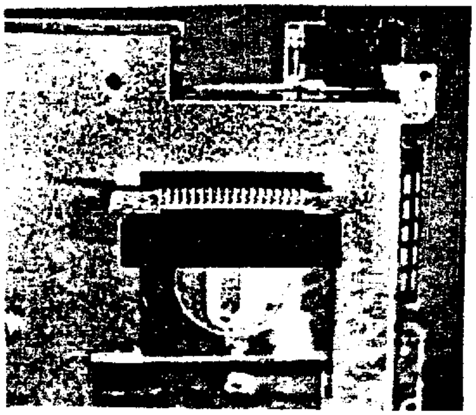


FIG 4

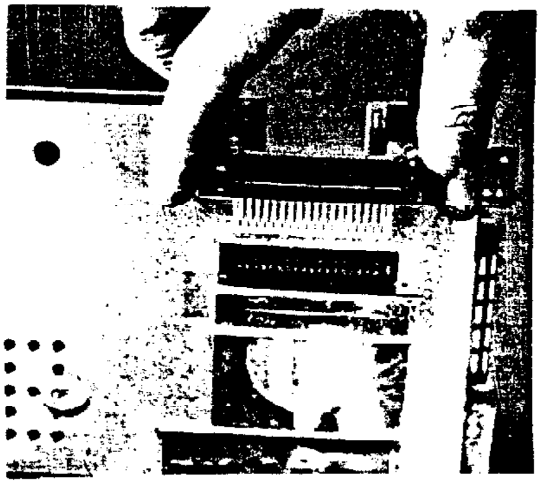


FIG 5

CLEANING

Take the cotton swab dipped in the cleaning solution and clean as illustrated in Fig.6. Then get the ruler, wrap the lint add cleaner and clean the inside as illustrated in Fig 7.

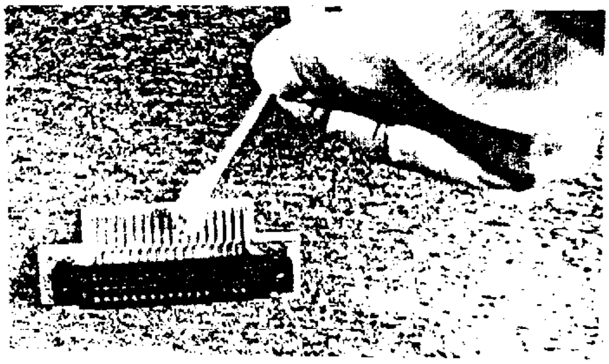


FIG 6

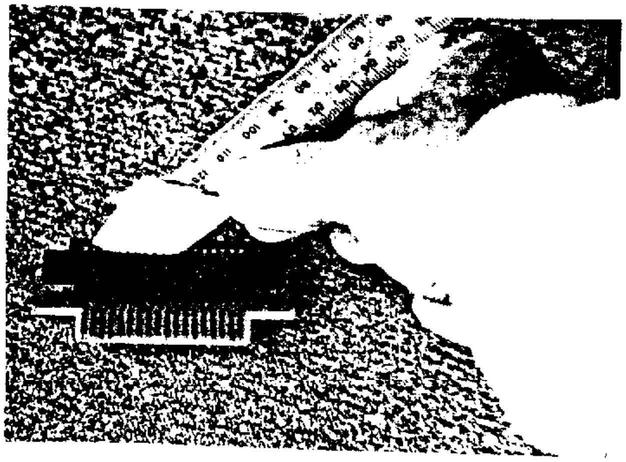


FIG 7

Now all that remains is to reassemble your computer.

REASSEMBLY

1) Reconnect 15 pin socket to main circuit board and relocate board on locating pins.

2) Replace the 3 screws(+) on main circuit board.

3) Replace power board ensuring that the wires to the power plug are not entangled. Redo the 2 screws(*)into the power board.

4) Tape power lead onto main circuit board and reseat power plug.

5) Check position of ON/OFF switch ensuring that it is properly in place.

Check to see that you have no parts left over. When I did the draft of this article I did this exercise on the club computer and when I turned it on IT WORKED, well at least that's my story.

I hope this article shows you that even a rank novice can perform this job at still have a computer that works. As well, a bonus is that you get to have a look at the "guts" of your computer. Best of Luck...

P.S. Thanks Paul for your help...

A PROGRAM FOR MINI MEMORY

TRANSLATED FOR MINI MEM
BY
KEVIN COX

```
BAR GRAPH
VS EQU >6024
VM EQU >6028
   MY BSS 58
   BA LWPI MY

TI TEXT '0   20   40   60   80   100'
***************************************
*DEFINE BAR GRAPH CHARACTERS IN VRAM*
***************************************
GO LI R3,9          *9 CHAR TO DEFINE

    LI R0,>C00      *VRAM LOCATION TO ASCII
    LI R1,>FF       *START WITH BLANK PATT.
    BLWP @VS        *STORE PATT.IN VRAM
    LI R2,7         *7 MORE PATT.BYTES
 ST MOVB R1,@>8C00  *STORE PATT.BYTES
    DEC R2          *NEXT CHAR.PATT.BYTE
    JNE ST          *8 PATT.PER CHAR
    SRC R1,1        *NEXT CHAR.PATT
    LI R2,8         *8 PATT.BYTES PER CHAR.
    DEC R3          *NEXT CHARACTER
    JNE ST          *LAST CHARACTER
    LI R0,>390      *VRAM LOCATION
    LI R1,>5000     *LIGHT BLUE/TRANSP.COLOUR
    BLWP @VS        *COLOUR TO TABLE
    MOVB R1,@>8C00  *IN VRAM
*********************
*TOP OF SCREEN VALUES
*********************
    LI R2,26
    LI R0,3
    LI R1,TI
    BLWP @VM        *SHOW TITLE ON TOP OF SCREEN
    LI R1,MY+32     *STORAGE LOCATION FOR BAR CHARS
    INC R3          *START AT 1
    LI R0,35        *SCREEN LOCATION FOR START OF BAR
    LI R9,>607      *JOY 1/JOY 2 SELECT
    LI R8,3000      *INITIAL DELAY
****************************************
ROUTINE TO SHOW BAR VALUES ON SCREEN *
****************************************
SH MOV R3,R5
   MOV R5,R6
   SRL R6,3         *DIVIDE BY 8
   JEQ EN           *AT LHS SIDE OF BAR
    LI R7,>8800     *SOLID CHARACTER PATTERN
BL MOVB R7,*R1+     *STORE BAR CHARACTER CODE
   DEC R6
   JNE BL
EN CI R1,BA         *AT RHS OF BAR?
   JEQ FI
   MOV R6,R6        *FLAG FOR PART CHARACTER
   JNE SR
   ANDI R5,7        *GET REMANDER OF DIVIDE BY 8
   AI R5,>80        *CALCULATE CHARACTERS
   SWPB R5
SR MOVB R5,*R1+     *STORE BAR CHARACTER CODE
   LI R5,>8000      *BLANK CHARACTERS
   SETO R6         *FLAG FOR BLANK CHARACTERS
   JMP EN
FI S R2,R1
   BLWP @VM        *SHOW BAR ON SCREEN
   MOV R8,R6       *GET DELAY COUNT
DE DEC R6
   JNE DE
*****************************
*SETUP FOR JOYSTICK SAMPLING*
*****************************
JO LI R12,>24      *CRU ADDRESS >0024
   SWPB R9         *ALTERNATIVE JOYSTICK
   LDCR R9,3       *SET FOR JOYST. SAMPLING
   LI R12,6        *CRU ADDRESS >0006
   STCR R5,5       *GET JOYST VALUES
   SLA R5,4        *SET TO CHECK UP/DOWN
   JNC DD          *UP PRESSED?
   JLT TL          *DOWN PRESSED?
DI INC R8          *INCREASE DELAY
   CI R8,30001     *DELAY HIGH LIMIT
   JNE TL
DD DEC R8          *DECREASE DELAY
   JEQ DI          *DELAY LOW LIMIT
TL SLA R5,2        *CHECK LEFT/RIGHT
   JNC IN          *RIGHT PRESSED?
   JGT DC          *LEFT PRESSED?
   SLA R5,1
   JLT JO          *FIRE PRESSED
   BLWP @>0000     *QUIT
DC DEC R3          *LOWER BAR VALUE
   JNE SH
IN INC R3          *INCREASE BAR VALUE
   CI R3,201       *VALUE AT HIGH LIMIT
   JEQ DC
   JMP SH
   END
   AORG >701C      *THESE LINES ARE NECESSARY
   DATA >7FB2      *TO SHOW THE NAME'BGRAPH'
   DATA >7FE0      *AND THE ADDRESS 'GO' SO
   AORG >7FE0      *AS THE COPUTER KNOWS WHERE
   TEXT 'BGRAPH'   *TO FIND THE PROGRAM!.
   DATA GO
   END
```

THE PROGRAM LISTED ABOVE ILLUSTRATES A METHOD OF DIRECTLY CHECKING EITHER JOYSTICK AND DISPLAYING A HORIZONTAL BAR GRAPH ON THE SCREEN.PRESSING EITHER JOYSTICK LEFT OR RIGHT WILL INCREASE OR DECREASE THE BAR ON THE SCREEN.PRESSING THE JOYSTICK UP OR DOWN CHANGES THE SPEED OF THE REACTION ,THE FIRE BUTTON WILL QUIT.

AUGUST ISSUE OF MICROPENDIUM WILL CONTAIN A TUTORIAL ON THE MINI-MEMORY WHICH SHOULD BE VERY INTERESTING!

THE HV 99ers BASIC Group
CLASS NOTES

PREPARED BY

PAUL MULVANEY

The IF-THEN statement allows the programmer to cause the computer to perform operations only under certain conditions. The format is

```
220 IF Relational-Expression THEN Line number
```

For the above example if the relational expression is true then control is transferred to the specified line number. If the expression is not true then control continues to the next line.
Allowable relational operators are; = equal to, < less than, > greater than <> not equal to, <= less than or equal to, >= greater than or equal to

The most usual condition is checking a variable with a constant. Both numeric and string variables can be checked eg,

```
405 IF A=5 THEN 700

200 IF VALUE<20 THEN 160

330 IF E$="NO" THEN 990
```

If the condition is false and you do not want the control to go to the next line in the program the ELSE Line number can be added.

```
150 IF R$="YES" THEN 250 ELSE 800
```

For the example above if R$ is YES then control goes to line 250, If R$ is not YES then control goes to line 800.

IF-THEN-ELSE is a very versatile statement and its full potential can only be discovered by experiment and practice.

The CALL KEY subprogram allows us to insert information via the keyboard without scrolling the information or graphics that are displayed on the screen. While its function is similar to the INPUT statement the ENTER key does not have to be pressed as the information is accepted directly without being displayed on screen. The format for the subprogram is;

```
CALL KEY(Key-unit,Return-variable,Status-variable)
```

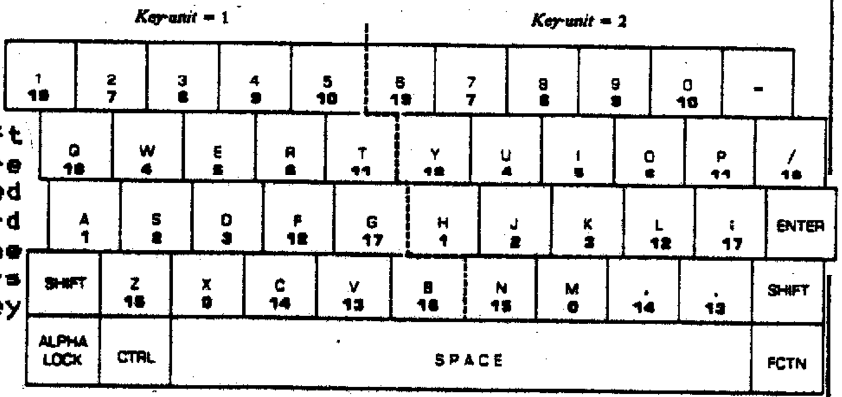
The Key-unit specifies the type of keyboard required.

The Return-variable must be a numeric variable to accept the number corresponding to the key that is pressed. The value is the ASCII number of the key pressed or special values as specified by the Key-unit.

The Status-variable must also be a numeric variable and indicates the status of the keyboard, ie if no key has been pressed since the last scan than the status is Ø. If a key has been pressed the status is +1 and if it has been held down than status is -1. Testing of the status-variable allows the program to loop back to the CALL KEY line while waiting for information.

The best way to become familiar with the subprogram is to run the following little routines and note the results. I found that the information in he Users Reference Guide did not contain all the available codes so I have pencilled them in, maybe the Guide was for the 4 not the 4A.

```
100 CALL KEY(1,K,S)
110 IF S=0 THEN 100
120 PRINT K,S
130 GOTO 100
```



Only the keys on the left half of the keyboard are active. The values returned are shown. No standard ASCII codes are active. The Function and Control keys are not enabled. Hold a key down and S will be -1.

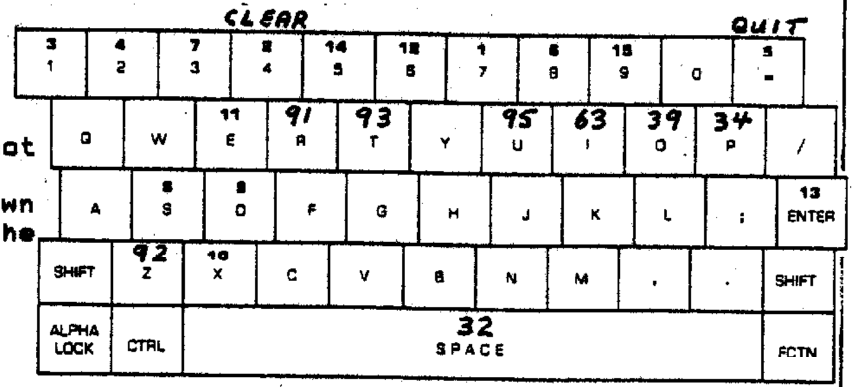
Alter line 100 to;

```
100 CALL KEY(2,K,S)
```

To utilise the Standard keyboad use Key-unit 3, this returns all the alphabet as upper case. This is useful when asking for a Y or N decision as you do not have to test for a lower case answer.

Alter line 100 to;

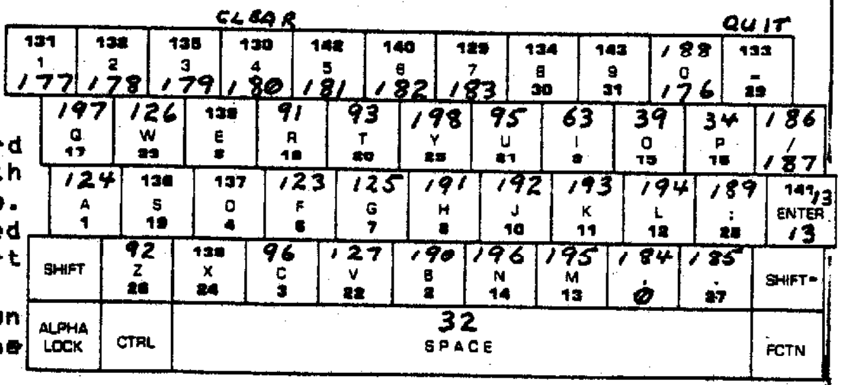
```
100 CALL KEY(3,K,S)
```



The Control key is not enabled.
The Function codes are shown on the upper part of the keys.

Alter lines 100 and 110

```
100 CALL KEY(4,K,S)
110 IF S<1 THEN 100
```

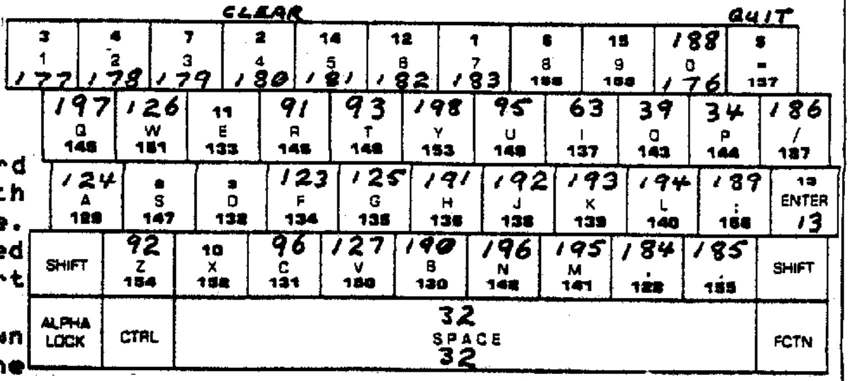


This will place the keyboard in the Pascal mode with both upper and lower case active. The Function codes returned are shown in the upper part of the key.
The Control codes are shown in the lower part of the key.
The change to line 110 will prevent repitition if the key is held down.

Alter line 1ØØ to;

1ØØ CALL KEY(5,K,S)

This will place the keyboard in the BASIC mode with both upper and lower case active. The Function codes returned are shown in the upper part of the key. The Control codes are shown in the lower part of the key.



This is the default keyboard mode if another keyboard is not specified.
If CALL KEY(Ø,K,S) is used the keyboard stays in the last mode specified.

```
*READ DV80 FILE AND DISPLAY
       DEF  BEGIN *NAME OF START OF PROGRAM*
       REF  DSRLNK,VSBW,VMBW,VMBR,VSBR,KSCAN *LIST OF INBUILT ROUTINES TO USE*
STATUS EQU  >837C *ADDRESS OF GPL STATUS BYTE*
POINTR EQU  >8356 *ADDRESS OF DSR POINTER*
BUFADR EQU  >1000 *ADDRESS IN VDP RAM TO ACT AS RECORD BUFFER*
PABADR EQU  >F80 *ADDRESS IN VSP RAM FOR PAB*
READ   BYTE >02 *OP-CODE TO INCLUDE IN PAB TO TELL DSR TO "READ"*
CLOSE  BYTE >01 *OP-CODE TO INCLUDE IN PAB TO TELL DSR TO "CLOSE"*
ONE    DATA >0001 *SETS VALUE OF "ONE" TO 1*
CR     BYTE >0D *HEX VALUE OF CARRIAGE RETURN*
EOF    DATA >0 *END OF FILE FLAG*
PAB    DATA >0014,BUFADR,>5000,>0000,>000A *THIS IS PAB DATA*
       TEXT 'DSK2.FILE1'
ERRMSG TEXT 'I/O ERROR=' *DSR ERROR MESSAGE*
CPUBUF BSS  >50 *ADDRESS IN CPU RAM TO ACT AS A BUFFER FOR RECORDS*
RETURN BSS  2 *ADDRESS TO SAVE RETURN ADDRESS IN*
LEN    BSS  2 *ADDRESS TO SAVE LENGTH OF RECORD IN*
WR     BSS  >20 *ADDRESS TO ACT AS WORKSPACE REGISTER*
SAVE1  BSS  >2 *
HEXFF  BYTE >FF *
BUFFI  EQU  >8375 *
**************************************************
* END OF SET-UP AREA NOW ONTO THE ACTUAL WORKINGS...*
**************************************************
BEGIN  MOV  R11,@RETURN *SAVE THE RETURN ADDRESS *
       LWPI WR *LOAD THE WORKSPACE POINTER*
       LI   R8,>0001 *
       LI   R0,PABADR *VDP RAM ADDRESS FOR PAB*
       LI   R1,PAB  *CPU RAM ADDRESS OF PAB DATA*
       LI   R2,>14 *LENGTH OF DATA*
       BLWP @VMBW *WRITE THE PAB DATA INTO VDP RAM*
       BL   @DSR *OPEN THE FILE-NOTE PAB BYTE 0="00" FOR "OPEN"*
       MOVB @READ,R1 *NOW MOVES OP-CODE TO "READ" INTO R1*
       LI   R0,PABADR *LOAD PAB ADDRESS INTO R0*
       BLWP @VSBW *PUT "READ" INTO PAB BYTE 0*
       CLR  R4 *CLEAR THE RECORD COUNTER*
       CLR  R5
READF  BL   @DSR *PERFORMS DSR ROUTINE*
       MOV  @EOF,@EOF *CHECK FOR END OF FILE*
       JNE  EOJ *IF END OF FILE GOTO END OF JOB*
       INC  R4 *ADD 1 TO RECORD COUNT*
```

OOPS!

As some of you may have noticed, in the last newsletter, the listing of the source code, of my attempt to load and display a DV8Ø file, was somewhat shortened and obviously was not complete.

I hope that this listing, (hopefully in its entirety) of the same program will survive our editing process, and provide a basis for member's comments and feedback to the EA group.

Regards
Peter.

```
       LI   R0,PABADR+5 *ADDRESS OF CHARACTER COUNT*
       BLWP @VSBR *READ THAT COUNT INTO LEFT BYTE OF R1*
       SRL  R1,8 *SHIFT THAT VALUE (BITS) TO THE RIGHT EIGHT SPOTS*
       MOV  R1,R2 *MOVE THE VALUE TO R2*
       MOV  R1,@LEN *SAVE THE VALUE IN "LEN"*
       LI   R0,BUFADR *VDP RAM RECORD BUFFER ADDRESS*
       LI   R1,CPUBUF *CPU RAM ADDRESS FOR RECORD*
       BLWP @VMBR *GET RECORD FROM VDP TO CPU RAM*
PUTR1  CI   R4,>0001        *CHECKS FOR 1ST REC*
       JEQ  PUS1  *IF 1ST RECORD JUMP TO "PUS1"*
       MOV  R5,R0 *IF NOT 1ST RECORD PUT SCREEN POS OF 1ST CHAR OF REC INTO R0*
PUTEND MOV  R0,R7 *PUT COPY OF SCREEN POS INTO R7*
       A    @LEN,R7 *PLACES LOCATION OF LAST CHAR OF REC INTO R7*
       CI   R7,>02FF *COMPARE SCR POS OF LAST CHAR WITH LAST SCR POS*
       JHE  PUKSCN *IF LAST CHAR WOULD BE PAST LAST SCR POS GOTO "ANY KEY"*
PU     BLWP @VMBW *IF NOT THEN PRINT THE JOLLY OLD THING*
************************
PULCHR S    @ONE,R7 *SUBTRACT ONE FROM LAST CHAR POS*
       MOV  R7,R0 *MOVE INTO R0 LOCATION OF THE LAST CHAR OF RECORD*
       BLWP @VSBR *READ THAT CHAR FROM THE SCREEN*
PUTCR  CB   R1,@CR      *TEST LAST CHAR FOR CR*
       JEQ  PULUT   *AND JUMP TO LOOKUP TABLE IF IS CR*
PUNXTP A    @ONE,R0 *IF NOT THEN ADD 1 TO SET UP NEXT POS*
       MOV  R0,R5   *STORE THAT ADDRESS IN R5*
       JMP  READF   *GO BACK AND READ THE NEXT RECORD*
******************
PUS1   LI   R0,0 *LOADS LOCATION OF 1ST SCREEN POS.*
       JMP  PUTEND *GOES TO PUTEND*
*****************
PULUT  LI   R3,>0000 *PUT 0 INTO R3*
       LI   R8,>001F *LAST POS OF LINE NO 1*
       A    R8,R3 *ADD ADDRESS OF END OF LINE INTO R3*
       JMP  CHECK1 *JUMP TO CHECK IF LAST CHAR IS IN THIS LINE*
CHECK  AI   R3,>0020 *ADDS A LINE TO THE POSITION*
CHECK1 C    R0,R3 *COMPARES START OF NEXT CHAR TO THE END OF LINE*
       JGT  CHECK *IF EOL POS IS GREATER GO BACK AND TRY NEXT LINE*
       INC  R3  *SET R3 TO START OF NEXT LINE*
       MOV  R3,R0 *PUT COPY INTO R0*
       MOV  R0,R5 *PUT COPY INTO R5*
       JMP  READF *MOVE BACK AND READ NEXT REC TO BE DISPLAYED ON NEW LINE*
******************
PUKSCN LI   R0,>0000 *START OF "ANY KEY TO DISPLAY NEXT PAGE" ROUTINE*
       MOVB R0,@>8374
       BLWP @KSCAN
       CB   @HEXFF,@BUFF1
       JEQ  PUKSCN   *IF NO KEY PRESSED GO BACK AND CHECK AGAIN*
       LI   R0,>0000 *WHEN PRESSED LOAD R0 WITH 0*
************************
PUCLR  MOV  R1,@SAVE1*THIS SECTION CLEARS THE SCREEN*
       LI   R0,>2FF
       LI   R1,>2020
PUBLA  BLWP @VSBW
       CI   R0,>0000  *CHECKS TO MAKE SURE ALL SCREEN POS ARE CLEAR*
       JEQ  GO        * IF ARE GOTO "GO"
       DEC  R0        * IF NOT CLEAR NEXT LOWER SPACE AND THEN*
       JMP  PUBLA     * GO BACK AND START AGAIN*
GO     MOV  @SAVE1,R1
       LI   R7,0000   *SET UP FOR DISPLAYING NEXT RECORD*
       LI   R5,0000
       JMP  PU        *GOBACK AND DISPLAY ON SCREEN*
**********************
```

```
DSR    LI   R6,PABADR+9 *LOAD R6 WITH DESCRIPTOR LENGTH*
       MOV  R6,@POINTR *MOVE ADDRESS TO POINTER*
       BLWP @DSRLNK *PERFORM DSRLINK*
       DATA 8 *DATA NEEDED BY DSRLINK*
       JEQ  DSRERR *CHECK FOR ERROR*
       RT *RETURN*
DSRERR INC  @EOF *SET EOF INDICATOR*
       LI   R0,PABADR+1 *ADDRESS OF PAB BYTE 1*
       BLWP @VSBR *READ PAB BYTE 1 INTO R1*
       SRL  R1,>D *SHIFT HIGH ORDER 3 BITS TO LOW*
       CI   R1,5 *CHECK FOR END OF  FILE VALUE=5*
       JNE  IOERR *IF NOT END OF FILE THEN OTHER ERROR*
       RT *IF END OF FILE RETURN*
IOERR  AI   R1,>30 *MASK ERROR CODE*
       SLA  R1,8 *SWAP LOW ORDER BYTE TO HIGH ORDER*
       LI   R0,>12B *DISPLAY ERROR CODE*
       BLWP @VSBW *PUT THE ERROR MESSAGE ON THE SCREEN*
       LI   R0,>120 *SET UP LOCATION ON SCREEN*
       LI   R1,ERRMSG *LOAD ACTUAL MESSAGE*
       LI   R2,>A    *LENGTH OF MESSAGE*
       BLWP @VMBW *PUT IT ON THE SCREEN*
EOJ    MOV  @EOF,@EOF *IF END OF JOB REACHED , DSR*
       JNE  NOCLOS *WILL CLOSE FILE*
       MOVB @CLOSE,R1 *MOV CLOSE OP-CODE TO R1*
       LI   R0,PABADR *LOAD PAB ADDRESS*
       BLWP @VSBW *WRITE CLOSE OP-CODE TO PAB BYTE 0*
       BL   @DSR *CLOSE FILE*
NOCLOS DECT @RETURN *ALTER RETURN ADDRESS*
       MOV  @RETURN,R11 *MOVE RETURN ADDRESS INTO R11*
       RT              *RETURN*
       END
```

IN THE NEWS

all the news that is news

edited by

JOE WRIGHT

IN THE NEWS.

From Canberra CHUG-A-LUG. In addition to the modifications to the BASIC Console which have been done by Jason Winter comes news of further work he is doing. "He has been working on a 16 bit bus connection to the matchbox 32K expansion. Preliminary results have shown a massive increase in speed. Imagine , shades of a TURBO TI !!!! Will keep you informed of any successful testing of various TI software. (apparently TI Hopper is unplayable at this speed !!!)"

Also reported in the CHUG-A-LUG is a programme from

QIXOFT
Westralian Instruments' Software Division.
G.P.O. Box S1391
PERTH 6001
Western Australia.

called Minimate, & as it's name implies it is for use with the Mini Memory module. Also required are, 32K expansion, disk drive AND/OR one cassette recorder.

"Minimate allows you to transfer memory image files or file sequences from CASSETTE to DISK, CASSETTE to CASSETTE, CASSETTE to MEMORY, DISK to CASSETTE, DISK to DISK or DISK to MEMORY. You may also load and run standard memory image files and file sequences." "Minimate is a 'TI-BASIC + MACHINE LANGUAGE HYBRID PROGRAMME' which must always be used with MINI MEMORY in the module port. The programme contains machine code which is loaded into the 4K RAM of the Mini Memory module and automatically executed. If you leave the contents of the Mini Memory undisturbed, the MINIMATE programme can be used at any time from basic or from the Mini Memory Menu."

TI WRITER SUPPORT.
Bill Harms writing in ROM has listed some of the programmes he uses to manipulate TI WRITER files.

The list is a result of him discussing a hyphenating programme for TI Writer files. He reports that "it does take some time to go through a big document, until you get used to the programme".

HYPHENATOR.
WAYNE STITH
715 Timken Drive
Richmond
VA 23229.
USA.

He comments that "the programme is very good and includes rather good instructions"

The rest of the list follows;

TI-SORT.
D.R.ROMER
213 Earl St
Walbridge
OH 43465

SPELLCHECKER.
DRAGON SLAYER AMERICAN
SOFTWARE Co.
T.W.KIRK
2606 Ponderosa Drive
Omaha
NE 68123

SORT and SPELL.
Software Specialties Inc
P.O. BOX 3304
Evergreen
CO 80439

STRING MASTER.
R.M.MITCHELL

TI-ARTIST PICTURES CONVERSION.
R.M.MITCHELL
BYTEMASTER Computer Services
171 Mustang St.
Sulphur
LA 70663

TIWRITMAIL.
Mail list programme uses TI-Writer as the file. Has Label/Form letter Value File operations.

FAST-SCREEN.(Screen maker).
Simple block char./text file is created that includes transliterates to load into TI-Writer.

FAST-PRINT.
All purpose printer commands test and save as transliterates to load into TI-Writer.
Bill Harms
6527 Hayes Court
Chino
CA 91710.

SIDEWAYS.
Tom Freeman.
c/o L.A. 99ers
P.O. BOX 3547
Los Angeles
CA 90247

SIDEPRINT.
Jim Swedlow.
7301 Kirby Way
Stanton
CA 90680

PRINTIT.
Roger Merrit.
1949 Evergreen Ave
Fullerton
CA

FONT WRITER.
J.P.Hoddie
via Asgard Software
P.O. BOX 10306
Rockville
MD 20850

Bill Harms suggests that you write to them first for costs etc.

RETURN PAID MAIL.
While on the topic of writing to Overseas organisations, a bit about reply paid coupons.

It is not widely know that the Postal Service makes available an International Reply Paid Coupon. This coupon paid for by you can be enclosed with your letter. The person to whom you are writing presents the coupon at the Post Office with your return letter, the coupon then pays for the return mail. As an example I recently sent an Air Mail letter to Micropendium. The cost was approx $1:40 (if I remember correctly, certainly no more!). I purchased two reply paid coupons which cost a total of $1:64 and enclosed them with the letter. This is an easy way to pay your return mail - remember, slipping money into an envelope and mailing it overseas is NOT legal. I also think it is polite to pay your return mail cost when seeking information.

P.E.BOX SINGLE CARD.
From the October edition of the Sydney News Digest comes news of a new P.E. BOX card. The card will provide 32K/RS232 PORTS 1 & 2, PIO PORT and DSDD DISK CONTROLLER. Peter Schubert writes that this new piece of hardware he is developing will be a single card. No mention of cost at this stage. Peter states "this will make room for more RAMDISKS and also allow you to up grade to DSDD and save heaps of disks."

As news comes to hand will keep you posted. If the price comes out at anything like the savings presented with the MINI P.E. system then he has another winner here!!

FROM GARY COX.
Gary Cox of the MID SOUTH 99/4A USERS GROUP from MEMPHIS in the USA is one gentleman with his ear to the ground. He is a regular contributor to that Group's Newsletter, his articles being on my must read list as they are most informative.

He has masses of information in the September newsletter. I would fill our newsletter to reproduce it all here. I strongly recommend you read this edition of TIDBITS.

The first bit of news from Gary is something that I have been waiting to hear more about myself.

"DIJIT Systems of San Diego, California has introduced it's latest product called ADVANCED VIDEO PROCESSOR CARD. According to the manufacturer, the AVPC fits into the PEB and is compatible with existing TI99/4A software. It features 8Ø-column text and advanced graphics with up to 512 colours, the manufacturer says. The AVPC also supports Mouse and light pen inputs.

The DIJIT systems card contains a Yamaha V-9938 video processor with 192K of video RAM and is designed to work with the "DIJIT-EYEZER" and external Gen-lock and video digitizing accessory. According to their flyer the digitise function can take a TV picture and instantly adapt it to a computer graphics picture! It will also allow titling and graphics overlays on home video as well as computer manipulation of external video images, giving the TI99/4A video processing power comparable with the ATARI ST and the AMIGA.

AVPC was scheduled for release last month for $195. For further information contact.

DIJIT Systems
4345 Hortensia St
San Diego
CA 921Ø3.

MORE FROM Gary Cox.
Databiotics
P.O. BOX 1194
Palos Verdes Estates
CA 9Ø274

has released a number of new products for the TI, including a RAM disk called GRAND RAM with configurations ranging from 64K to 512K of RAM. Prices range from US$129.95 TO US$229.95. According to Databiotics, up to four of the RAM disks may be plugged into the PEB providing more than 2 Megabytes of memory. It is compatable with CORCOMP, TI, MYARC, GENEVE, MORNING STAR and FOUNDATION peripherals and may be used with the RAM disk menu system designed by John Johnson. Accessories for the RAM disk include a plug-in real-time clock and an analog-to-digital device to interface with other devices and an emulator to create cartridges. The clock is priced at US$29.95. The Grand RAM is also available in a kit (build it yourself) ranging in price from US$99.95 for 64K to US$185.98 for 512K.

THAT'S ALL.
That is it for this month!!

See Ya
Joe Wright

and
deo
to
ion
tly
ics
ing
deo
of
the
wer
the

ast
her

new
RAM
ith
to
rom
ing
the
the
es
ith
NG
nd
nu
n.
de
an
to
an
he
he
it
ce
98

BUILD YOUR OWN

ADVENTURE GAMES

an introduction to techniques

by

"bazza" the mad adventurer

The purpose of this article is to help anyone with some interest in playing with their TI in EXB to construct their own Text Adventure.

It is not a game to be typed in as such but all the basics will be there to complete one yourself if enough interest is shown. Console plus XB is all that is required but some storage facility would help to stop going mad!

To ease my mind I will state at the onset that I am a mug programmer and have arrived at these routines just by reading and mucking around, all I can say is that they work for me, obviously my standard is nowhere near INFOCOM.

Initially I will tackle the first article by way of a self contained routine with explanations where I think necessary, the format of any future articles can be adjusted to suit the readers and/or editor- such as more programming less raving etc. etc.
Basically an adventure must -
1 Have clear instructions.
2 Be error free.
3 Restricted/unrestricted movement.
4 Ability to remove/replace objects at different locations.
5 Appropriate responses to valid and invalid input.
6 Logical solutions.
7 Permanent changes to story(I would hate to have to kill a monster every time I entered the same room).

There is probably more you can think of as well but hopefully all will become clear as we progress.

The first routine is mapping and this is only one way of achieving a result.
This is what we end up with -

```
    - LOCATION --change
  !      !        !
  !      !        !
  !   INPUT--VALID
  !      !
  !      !
    - INVALID
```

We will keep our directions to the four points of the compass but with this system more can be added at will. Data statements are read and use is made of SEG$ to arrive at the appropriate digit. The sequence is n,e,s,w and is represented in data statement r$. If we look at each location as a room number then each lot of r$ represents the next room, in the example given the plan becomes-

```
      1-2-3-4
      !
      5-6
      !
      7-8
```

Take room 5 as an example, movement is indicated in each direction except w. r$ is in the format "--------" that is, 8 digits, with "--" being no movement, so the data statement for room 5 would be - n to 1, e to 6, s to 7, no movement w, or "Ø1Ø6Ø7--". The SEG$ command determines the 2 digits involved eg. if we wanted to go s from room 5 then after inputing "s" then SEG$(r$,5,2) selects the 5th 6th digit in the string viz. Ø7 which is changed to 7 with VAL and becomes a valid move to room 7. The data statements must then be restored so that the counting begins at 1.

Here it is in detail typed in lowercase-

```
1ØØ CALL CLEAR
11Ø !map routine
12Ø !
13Ø !
14Ø R=1 !starts game at room no 1
15Ø !
16Ø !
17Ø RESTORE !resets counter back to 1
so that the correct value of r is
arrived at when rooms are changed
18Ø !
19Ø !
2ØØ FOR I=1 TO R :: READ N$,D$,R$ ::
NEXT I ! reads data statements up to
the value of r
21Ø !
22Ø !
23Ø PRINT N$ :: PRINT !prints room number
24Ø !
25Ø !
26Ø PRINT D$ !prints description
27Ø !
28Ø !
29Ø C$="" :: PRINT !null
3ØØ !
31Ø !
32Ø INPUT "instructions? ":A$ :: PRINT
33Ø !
34Ø !
35Ø IF A$="n" THEN C$=SEG$(R$,1,2)!
computes the value of the room from r$
if valid
```

```
36Ø !
37Ø !
38Ø IF A$="e" THEN C$=SEG$(R$,3,2)
39Ø IF A$="s" THEN C$=SEG$(R$,5,2)
4ØØ IF A$="w" THEN C$=SEG$(R$,7,2)
41Ø !
42Ø !
43Ø IF C$="" THEN 27Ø !invalid input
44Ø !
45Ø !
46Ø IF C$<>"--" THEN 52Ø !if C$ is a valid move
47Ø !
48Ø !
49Ø PRINT "can't go there!" :: GOTO 27Ø !no change in room number
5ØØ !
51Ø !
52Ø R=VAL(C$)!the new value of r
53Ø !
54Ø !
55Ø GOTO 17Ø
56Ø !
57Ø !
58Ø DATA room 1, your lounge
room,--2Ø5--
59Ø DATA room 2, bedroom,--Ø3--Ø1
6ØØ DATA room 3,this is a dark room,--Ø4--Ø2
61Ø DATA room 4,you are in a cave,------Ø3
62Ø DATA room 5,outside your house,Ø1Ø6Ø7--
63Ø DATA room 6,local store,------Ø5
64Ø DATA room 7,street footpath,Ø5Ø8----
65Ø DATA room 8,inside the bank,------Ø7
```

The following can be added if needed although not necessary;

```
271 PRINT
272 PRINT"obvious exits are, ";
273 IF SEG$(R$,1,2)<>"--"THEN PRINT"n ";
274 IF SEG$(R$,3,4)<>"--"THEN PRINT"e ";
275 IF SEG$)R$,5,6)<>"--"THEN PRINT"s ";
276 IF SEG$(R$,7,8)<>"--"THEN PRINT"w"
```

Following articles would be routines on taking/dropping objects, construction of the input commands & logic, story logic and construction.

TAKE A NUMBER. . . ANY NUMBER

A FORTH APPLICATION

BY

BRIAN RUTHERFORD

When Richard Terry's Forth group started one of the first tasks he set was to write a word to accept a number, which was expanded to mean any number. I in my perverse way took that to the extreme and wrote a routine to accept numbers even in scientific notation.

Using Basic I first worked out some rules as to what is a valid number to the TI 99/4A. And the rules I came up with are these:-

1. Digit, any
2. Decimal point, only one and not after the "E"
3. Minus sign, must be first or first after the "E"
4. "E", only one, must not be first or last.

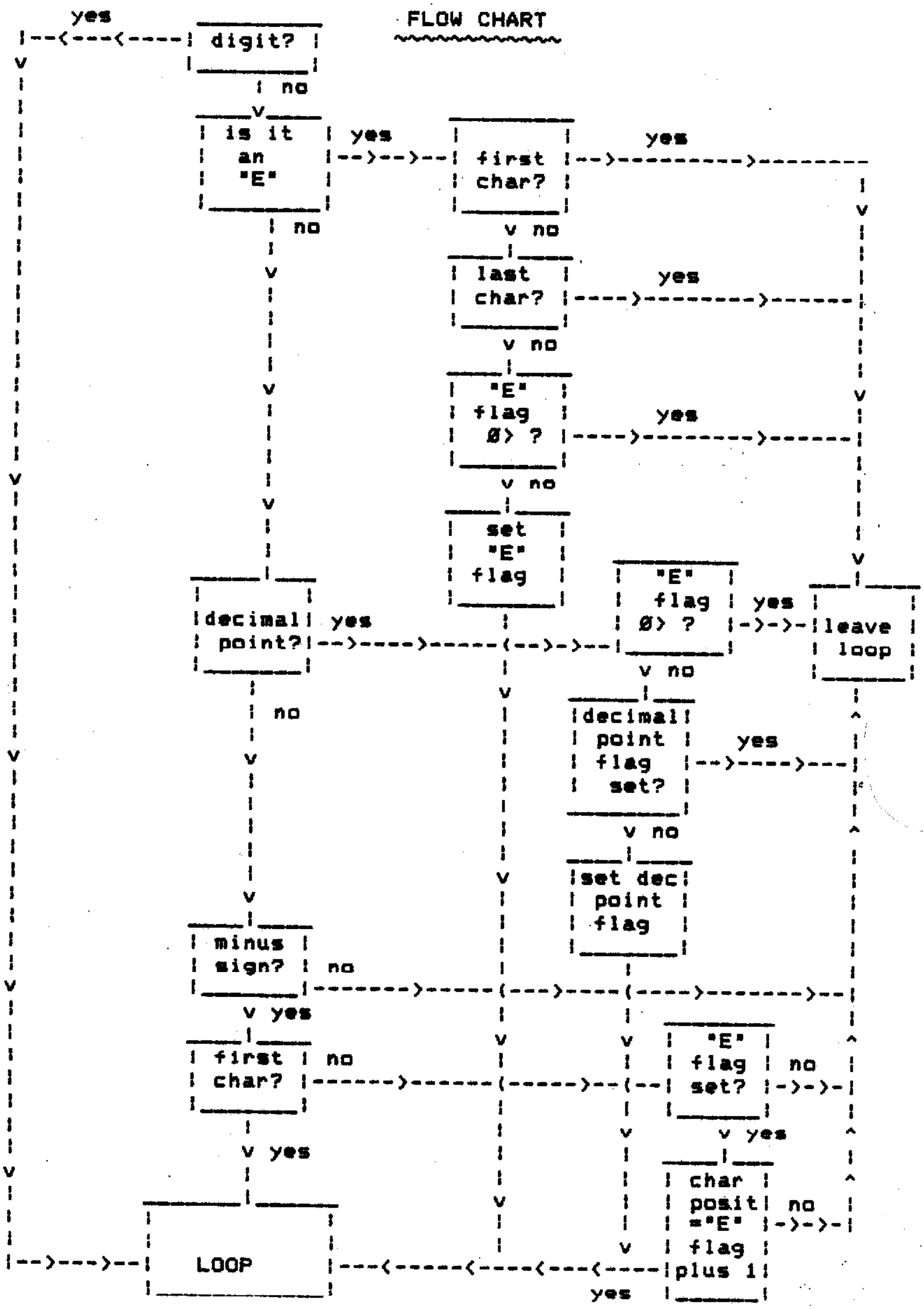
Then after a bit of fooling around with a loop to check the input digit by digit I came to the conclusion that I needed a flow chart for this job. Opposite is the one I came up with - it looks complicated but the routine seems to work.

I wrote my own words to accept an input. They are (ACCEPT) and MOV/BUF which make up the word ACCEPT$. That word is nearly the same as Richard's GET$. The variables are all self-explanatcry, except maybe TEMP, which I use as a pointer to store the address of the buffer. This allows the user to use the routine and store the inputed number in any buffer the wish, without having to alter the WORD(S).

Finally I do not know whether this is good code or not as I do not know enough about Forth. It seems awful long to me, and I thought to shorten it by only accepting decimals, positive or negative, but on looking at the routine I see that would not shorten it very much at all. I think there must be an easier way of doing it, may be somebody out there who programmes in Forth will let me know some day. I will just add a short note to Bob Carmany at this point to let him know that the few of us here doing Forth are not disinterested in Wycove Forth; it's just that most of us are still struggling with TI Forth and it would be like trying to learn TI Basic and jump in to another Basic before we had any real idea of what we were doing in TI Basic.

```
SCR #5Ø
    ( BRIANS INPUT WORDS 2Ø/4/87 )
    Ø VARIABLE CHRCNT                   ( allot chr. count limit     )
    Ø VARIABLE DPOINT                   ( allot  dec. point count    )
    Ø VARIABLE ECNT                     ( allot "E" count            )
    Ø VARIABLE ICNT                     ( allot var. to save loop cnt)
    Ø VARIABLE TEMP                     ( allot temp.point to buffer )
    Ø VARIABLE B$ 8 ALLOT               ( allot 1Ø byte buffer       )
    : (ACCEPT)                          ( the primitive accept       )
               TIB @ SWAP EXPECT        ( put input into TIB         )
               Ø IN ! 13 WORD HERE      ( from k/board WORD puts in  )
               ;                        ( to dict. leave HERE on stck)
    : MOV/BUF                           ( expect adr2 adr1 on stack  )
               COUNT 1+                 ( get adr.+1 with COUNT      )
               SWAP 1-                  ( +1 to cnt. 1- to adr.      )
               ROT ROT                  ( adr2 - top cnt - top CMOVE )
               CMOVE ;  -->             ( to adr2 stack empty        )
```

FLOW CHART


digit?
yes
no
is it an "E"
yes
first char?
yes
no
last char?
yes
no
"E" flag Ø> ?
yes
no
set "E" flag
decimal point?
yes
"E" flag Ø> ?
yes
leave loop
no
decimal point flag set?
yes
no
set dec point flag
minus sign?
no
yes
first char?
no
yes
"E" flag set?
no
yes
char posit ="E" flag plus 1
no
yes
LOOP

```
SCR #51
  ( BRIANS INPUT WORDS 2Ø/4/87 )

  : ADDSPACE                                ( expect nothing on stack    )
              TEMP @ COUNT                  ( get adr+1 and count        )
              OVER OVER                     ( copy adr.+1 & cnt          )
              + 32 SWAP C!                  ( adr+1 + cnt and ! ascii 32 )
              1+ SWAP 1- C!                 ( cnt+1 adr-1 org.adr  ! cnt )
              ;                             ( leave nothing on stack     )

  : ACCEPT$
              (ACCEPT) MOV/BUF              ( expect adr.& cnt on stack  )
              ;                             ( leave nothing on stack     )



  -->
```

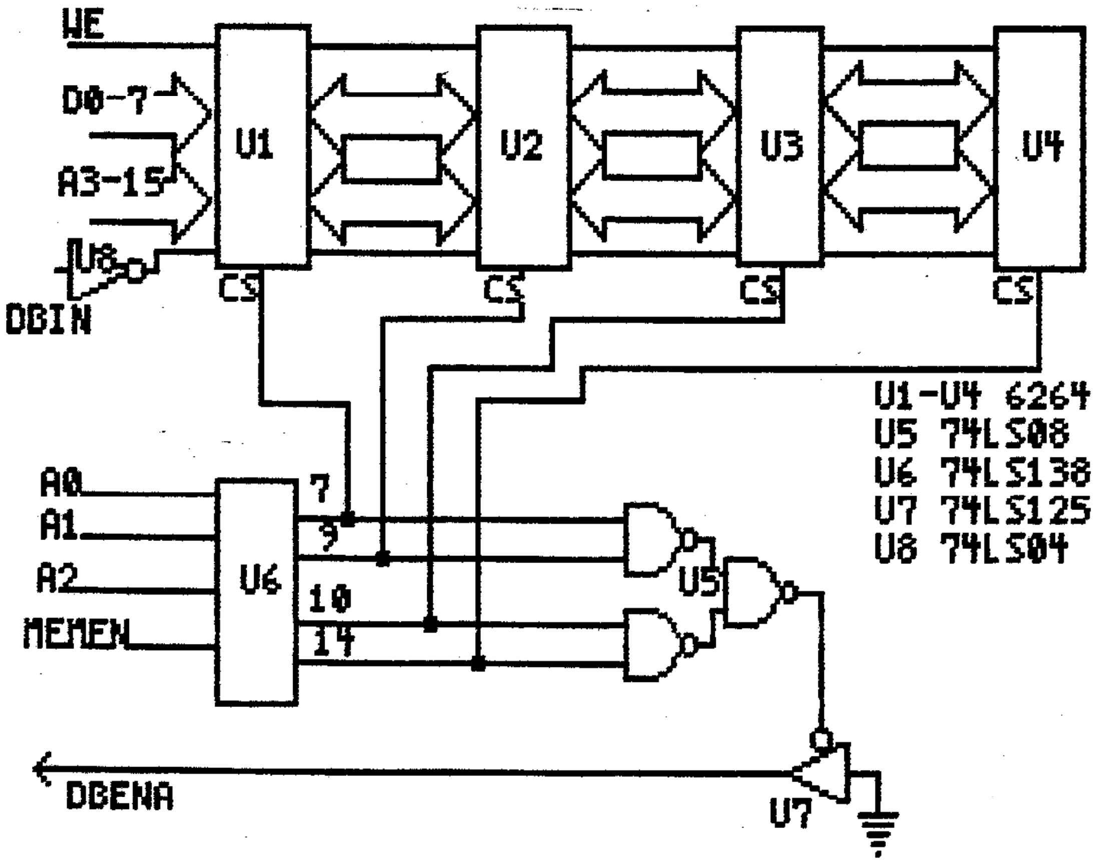
```
SCR #52
  ( BRIANS INPUT WORDS 2Ø/4/87 )

  : E?                                      ( expect nothing on stack    )
              ICNT C@ 1 =                   ( if E is first chr.         )
              CHRCNT C@ ICNT C@ = OR        ( or E is last chr.          )
              ECNT C@ Ø > OR                ( or E flag has been set ?   )
              IF DROP Ø LEAVE ELSE          ( tf. drop chr. Ø on stack   )
              ICNT C@ ECNT C! THEN          ( ff= ok set E flag=I        )
              ;                             ( leave chr. or Ø on stack   )

  : DPOINT?                                 ( expect nothing on stack    )
              ECNT C@ Ø >                   ( E flag set ?               )
              DPOINT C@ Ø > OR              ( decimal point flag set ?   )
              IF DROP Ø LEAVE ELSE          ( tf. drop chr. Ø on stack   )
              1 DPOINT C! THEN              ( set decimal point flag     )
              ; -->                         ( leave chr. or Ø on stack   )
```

```
SCR #53
  ( BRIANS INPUT WORDS 2Ø/4/87 )

  : MINUS?                                  ( expect nothing on stack    )
              ICNT C@ 1 > ECNT C@ Ø=        ( if loop cnt I >1 and       )
              AND                           ( E flag is set              )
              IF DROP Ø LEAVE ELSE          ( tf. drop chr. Ø on top     )
              ICNT C@ ECNT C@ 1+ >          ( check - is next chr after)
              IF DROP Ø LEAVE               ( E tf. drop chr. Ø on top )
              THEN THEN                     ( leaving chr. or Ø on top )
              ;
  : DIGIT?                                  ( expect chr. on top         )
              47 >                          ( compares if less than Ø    )
              OVER                          ( copy chr. to top           )
              58 <                          ( greater than 9 ?           )
              AND                           ( do an and. and leave flag)
              ;  -->
```

```
SCR #54
( BRIANS INPUT WORDS 2Ø/4/87 )
: CHECK-NUM
              TEMP @ COUNT             ( expect nothing on stack    )
              DUP CHRCNT C!            ( get adr+1 & count          )
              1+ 1                     ( dup cnt and ! top copy     )
           DO                          ( for limit compare 1+ for   )
              I ICNT C!                ( loop limit 1 loop start    )
              DROP TEMP @ I + C@       ( ! current loop conter      )
              DUP DIGIT?               ( drop adr.+1 @ byte at adr+I)
              IF NOP ELSE DUP 69 =     ( dup check for digit        )
              IF E? ELSE DUP 46 =      ( tf nop ff dup and check E  )
              IF DPOINT? ELSE DUP 45 = ( else check decimal point   )
              IF MINUS? ELSE DROP Ø LEAVE ( else minus sign else drop)
              THEN THEN THEN THEN      ( Ø on stck for ff leave)
           LOOP
              ;   -->                  ( leaves character or Ø      )

SCR #55
( BRIANS INPUT WORDS 2Ø/4/87 )


: CONVERT
              TEMP @                   ( expect nothing on stack    )
              PAD SWAP  MOV/BUF        ( get adr. of str.num        )
              VAL FAC> ;               ( pad adr. swap mov adr to   )
                                       ( pad convert to fpoint num. )
                                       ( leave fl. point no. on stck)

: STR/NUM
              ADDSPACE CONVERT
              ;


 -->

SCR #56
( BRIANS INPUT WORDS 2Ø/4/87 )

: NUMACCEPT
              Ø DUP DUP                ( expect adr. & chr. cnt.    )
              CHRCNT C!                ( Ø on top make two copies to)
              DPOINT C!                ( reset variables to zero    )
              ECNT C!
              OVER OVER
              19 5 AT ."          "    ( DUP top 2 stack items      )
              19 5 AT                  ( blank scr. for input       )
              ACCEPT$ CHECK-NUM        ( input here                 )
              Ø= IF MYSELF THEN        ( accept num.                )
              ;                        ( not number do again        )
                                       ( leaves adr.& orig. count on)

-->
```

```
SCR #57
( BRIANS INPUT WORDS 20/4/87 )

B$ TEMP !                       ( store buffer adr into pointer)

: PROMPT1 4 5 AT ." Number please:" ;        ( print 1st prompt )
: PROMPT2 3 7 AT ." The number was: " F. ;   ( print 2nd prompt )
                                             ( & print number    )

: TRY      CLS
           PROMPT1
           TEMP @ 8 NUMACCEPT
           DROP DROP STR/NUM
           PROMPT2
           ;


```

RONS RAM CARD



CURIOSITY CORNER

A PROBLEM AND A TIP FROM RON KLEINSCHAFER

One wonders about the intricacies of the internal workings of the TI99/4a and the odd things that crop up from time to time. One story goes like this:

After making up a 32k static memory expansion board for the PE box it was inserted into the PE box (without the RAM chips) for some testing and checking. At this stage the 32k matchbox system was still in the console untouched and working fine, BUT when the 32k board (without the RAM chips) was fitted in the PE box and the system RUN to do some logic tests, the signal DBENA, which enables the data bus buffer on the console end of the PE cable, was initiated. When this occurred the Processor refused to recognise that Memory Expansion existed, producing all sorts of error codes with No Memory Expansion and Syntax error in xxx etc. After some head scratching the RAM chips were fitted and surprise! all of the 32k's little 1's and Ø's got to work straight away. As far as I can see this signal does not return to the console so why does it cause the problem? Why does the 32k in the console no longer work (it is still fully decoded, isn't it ??). Why does the processor refuse to recognise the console fitted memory? Why am I going to take up drinking?? Perhaps someone could shed some light on this oddity - maybe I missed some twist in the systems circuit? A basic circuit diagram as used to make up this RAM card is included for those hardware hackers and others who may wish to ponder over the curiosity.

Another thing to ponder is using that VERY useful utility of Funnelweb, CTRAM. If you don't already know, CTRAM loads up Cartridge Ram so that there is almost instant access to the guts of Funnelweb, bypassing any XB utilities and puts you straight into your User List and E/A loaders etc. To me this was a stroke of genius by Tony and Will. This utility gets you back into Funnelweb very quickly after using some other program that doesn't return automatically to Funnelweb, or destroys Funnelweb when the program is used. Unless my system has some quirk of operation that I am not aware of, I noticed that once the Cartridge Ram was loaded then it was cycled through the operation of first loading Funnelweb from cartridge Ram, selecting user list, then selecting No 9 (which is Cartridge Ram) then again accessing Funnelweb, I found that after cycling through this a few times the cartridge Ram would become corrupt and not load anymore! Anyone know why?? A fix for this was to load up the cartridge Ram, extract the code then program a 2764 Eprom with that code and fit that in place of the RAM chip. It goes without saying that if anyone does this then the battery must be removed from the cartridge and the only other alteration is to connect pin 1 of the Eprom to pin 28. You can still change back to a 6264 ram IC if desired provided you have a socket fitted to the cartridge. Sweet is the word for use of that setup - as one interested observer stated, "I know, you have a RAM DISK!!", when after five quick keystrokes and very little disk activity I had Viatel on screen in a matter of seconds. Of cause my return to that party was "Oh no, its just that these 16 bit jobs are a lot faster that those 8 bit'ers that you use !!" HAA got him, and left him full of curiosity.

R.K.

JOYSTICK ADAPTOR

By Paul Mulvaney

I recently purchased a Speedking joystick for my son, the new joystick was fitted with microswitches and is more comfortable to use than the original TI units. I decided to make an adaptor rather than butcher either the old TI set or the new one. With the aid of a multimeter I worked out the pin connections of the Speedking, these connections should be standard for all Atari compatible joysticks. The first adaptor I made was only for Joyst 1 and it was very quickly pointed out to me that some games work on Joyst 2 so the adaptor had to be able to work as either Joyst 1 or 2. A quick modification and the inclusion of a switch rectified that. Now to change from one to the other simply flick the switch over.

The connections are shown looking at where the wires are attached to the plug or socket. (A plug has the pins sticking out, a socket has the holes for the pins to plug into.)

The parts required are;

A D9 Socket, a D9 Plug, a single pole switch and 5 1n4001 Diodes

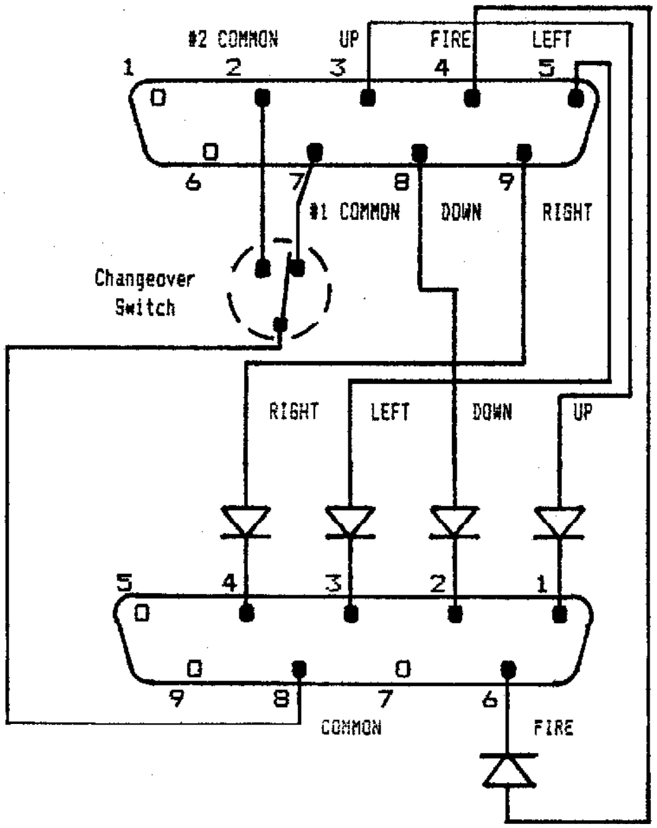
The diodes are to prevent feedback signals when you are going diagonally, if they are not put in the motion will slow and stop when two directions are selected.

Care should be exercised when soldering the diodes, dont use too much heat. Also insulate any connections where there is any likelyhood of them touching together.

To identify the direction of the diodes place the bar on the case towards the joysticks, eg

By working neatly the switch will fit into one headshell and the diodes into the other, no other enclosures are necessary.

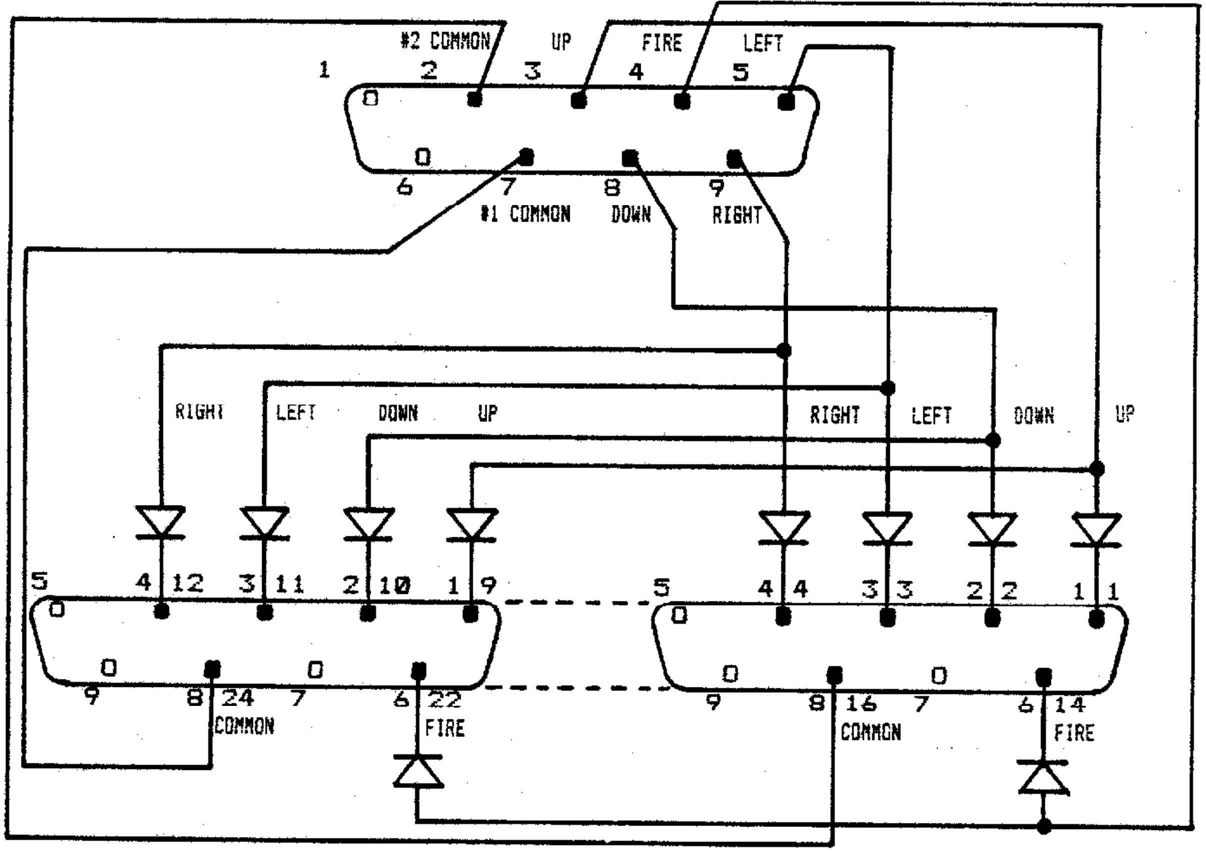
D9 SOCKET (Computer End)



D9 PLUG (Joystick End)

Because of the popularity of the new Joystick I purchased another for my other son (Birthdays come in handy to justify upgrading of equipment) so I now needed an adaptor to allow the use of both joysticks at once. I made a dual adaptor and mounted it in a small box and Gary Jones suggested using a D25 connector and blanking off the centre pins, so two versions exist but both use the same wiring scheme. In my box I mounted the diodes on a piece of strip board, in the D25 the diodes can be mounted directly onto the plug and the wires attached as per the previous adaptor. The numbers on the right are for the D25 plug.

D9 SOCKET (Computer End)



D9 PLUG FOR JOYSTICK 1 D9 PLUG FOR JOYSTICK 2

(Or D25 PLUG)

ASSEMBLY LANGUAGE

FOR THE LAYMAN

WITH ALLAN WRIGHT, HV99ERS

The first duty is to apologise for not having had an article in the Newsletter last month, the second is to THANK Pete Smith for his contribution in my absence. I add my support to his request for articles for the Newsletter, people ARE interested in what you are doing with your computer. Finding black holes, bending bananas or keeping records on Nocturnal Creatures, what ever your use of the TI somebody will be interested.

Also delighted to see the article on Mini-memory by Kevin Cox from Coffs Harbour, thank you.

DSRLNK.
I hope by now that some other readers as well as Pete Smith have managed to get their DV8Ø reader operating, if so feel free to send it to me for a "LOOK AT".

I recently had the need to write a routine to put some DF128 files to disk. I used the pretty much standard method.

```
    PABADR  EQU    >F8Ø
    BUFADR  EQU    >1ØØØ
    PAB     DATA   >ØØØ2,>1ØØØ,>8ØØØ
            DATA   >ØØØØ,>ØØØA
    FNAME   DATA   >DSK2.FILE1
```

I then opened the file after loading the PAB into VDP RAM.

```
            LI     R6,>F89
            MOV    R6,@>8356
            CLR    @STATUS
            BLWP   @DSRLNK
            DATA   8
            JEQ    ERR
```

This section of code was also used to write the file to the disk after the open code >ØØ was changed to write >Ø3.

It was like magic, worked fine, the file was closed using the above code again except with close >Ø1 written to byte zero of the PAB before the DSRLNK was accessed to close the file.

THEN - THEN.
I decided to test the code to ensure that it would pull up on any of the likely errors a user could make with the disks and drives. It passed these tests with flying colours,
It detected;
1) DRIVE DOOR OPEN
2) WRITE PROTECTED DISK
3) DISK UP SIDE DOWN
4) AN ALREADY FULL DISK.

That last test lead me to a problem. I put enough files on a spare disk such that my routine would write approx half it's files then run into a full disk. It should come back and report disk ERROR 4 just like TI Writer Editor, OR so I thought. IN with the disk, enter the filename and way it went saving like a little beauty, THEN - THEN the whole works stopped. The Disk controller light on but no drives running and the TI sitting there waiting for a reply from the controller. What the name of Harry is this? After much thinking and reading of articles on disk access through DSRLNK and DATA 8. I could not see any answer. There had to be but I had hit a brick wall.

Where does a HV 99'er go when he has a disk access problem that he cannot solve. At the last Monthly meeting I cornered WILLS McGOVERN and told him of my problem,"Yes Joe the disk controller light comes on but no disk is running," (I hope I get this correct Wills). "Well your code JOE is only looking for a link error, the disk becoming full during a write is not a link error. This is what you need to do with your code".

```
            LI     R6,>F89
            MOV    R6,@>8356
            CLR    @STATUS
            BLWP   @DSRLNK
            DATA   8
```

```
        JEQ   ERR
        LI    RØ,>F81
        BLWP  @VSBR
        SRL   R1,13
        JNE   ERR
```

This code checks the second byte of the PAB for any error after every access. If the error code is not zero the programme will branch to the error handler.

So with the new code installed in my programme I tried the same trick with an almost full disk. Yes it works fine, ERROR code 4 reported.

Thanks Wills

I have read and reread sections 18.2.2 and 18.2.3.1 of the E/A Manual and still cannot decide if that is telling me what Wills has pointed out or not? What do you think?

SCROLLING.
There are many routines floating around to scroll a section of VPD memory. As I see it there are two sections of VDP RAM that one would be likely to want to scroll. The first is obviously the screen area. The second is some area in which a page to be sent to a printer has been set up. My method of dumping to the printer is to set up the complete page in VDP RAM, starting at the buffer address >1ØØØ for the first line. Then scroll the page one line after each printer access.

I will maybe find more efficient methods as I go but just at this point that approach seems logical. This left me to write a scroll subroutine to be used on either areas.

The first obvious difference is the page length, screen 32 chars printer 8Ø. The code follows and the description of operation follows that.

```
*THIS ROUTINE SCROLLS THE PRINT PAGE IN VDP RAM
PBUFF1 BSS  >5Ø               CPU RAM BUFFER
SPAGE1 MOV  *R11+,R8          PAGE LENGTH
       MOV  *R11+,@SPAGE5     FIRST ADDR VDP RAM
       MOV  *R11+,@SPAGE6     LINE LENGTH
       MOV  R11,@RETUR1       SAVE RETURN ADDR
       LI   R7,1              SET COUNTER
SPAGE4 A    @SPAGE6,@SPAGE5   NEXT LINE
       BL   @SCR9             PUT NEXT LINE INTO BUFF
SPAGE5 DATA >ØØØØ,PBUFF1      VDP ADDR,CPU ADDR
SPAGE6 DATA >ØØØØ             No.BYTES TO READ
       INC  R7                INC LINE COUNTER
       MOV  @SPAGE5,@SPAGE7   MOV DATA FOR WRITE
       MOV  @SPAGE6,@SPAGE8   MOV DATA FOR WRITE
       S    @SPAGE6,@SPAGE7   SUB. LINE LENGTH
       BL   @SCW1             WRITE LINE TO VDP RAM
SPAGE7 DATA >ØØØØ,PBUFF1      VDP ADDR,CPU ADDR
SPAGE8 DATA >ØØØØ             No.BYTES TO WRITE
       C    R7,R8             COMPARE COUNT/PAGE LENGTH
       JLT  SPAGE4            GO AGAIN IF LESS
       MOV  @RETUR1,R11       RESTORE RETURN ADDR
       RT                     RETURN
```

The routine does a complete scroll, that is it rotates the page up one line for each it is called.

OPERATION. One buffer area is set aside at;

```
      PBUFF2 BSS   >5Ø   BUFFER1
```

The subroutine is accessed using BL and three pieces of DATA are carried into the routine;

```
      SPAGE1  MOV   *R11+,R8
              MOV   *R11+,@SPAGE5
              MOV   *R11+,@SPAGE6
```

The first is the page length, for the printer this is 66 lines, for the screen 24. The second is the address of the start of the first line of the page, for the printer this is >1ØØØ and for the screen this is Ø.
The third is the length of line, 8Ø for printer and 32 for the screen.

```
              MOV   R11,@RETUR1
```

Save the return address

```
              LI   R7,1
```

Register 7 is loaded with one, this is the first line of text. The first line is lost with this scroll subroutine. R7 is incremented by one with each line read.

```
      SPAGE4   A      @SPAGE6,@SPAGE5
```

This line of code adds the line length to the first page address, the result is SPAGE5 now contains the start address of the second line.

Two additional subroutines are used inside this routine, this is the first. It uses VMBR to move a line of DATA from VDP to CPU RAM.

```
               BL     @SCR9
      SPAGE5   DATA   >ØØØØ,PBUFF1
      SPAGE6   DATA   >ØØØØ
```

This is an example of moving DATA directly to a second subroutine from outside the original routine by the use of LABELS and the MOV instruction. The address of the start of the page or screen is placed at SPAGE5 and the line length at SPAGE6. This data is then altered as required by the 'A' instruction. It is then passed to the subroutine SCR9. This DATA should look familiar as it is the DATA required for VMBR. That is ADDRESS in VDP, ADDRESS in CPU, number of bytes to move. This subroutine places the second line of the page into PBUFF1.

```
      INC      R7
```

Count first line movement. Which is actually the second line of the screen or page.

```
               MOV    @SPAGE5,@SPAGE7
               MOV    @SPAGE6,@SPAGE8
               S      @SPAGE6,@SPAGE7
               BL     SCW1
      SPAGE7   DATA   >ØØØØ,PBUFF1
      SPAGE8   DATA   >ØØØØ
```

This section of code reads the line just placed in PBUFF1 back to VDP RAM but in the position one line above from where it was read from.
The address from which it was read is MOV'd to SPAGE7. The line length is MOV'd to SPAGE8. The line length is then Subtracted from the address at SPAGE7. This gives the address of the line immediately above that from which it was read.
A subroutine SCW1 is used to write the line back to VDP RAM.

```
               C      R7,R8
               JLT    SPAGE4
               MOV    RETUR1,R11
               RT
```

The number of lines moved is counted in R7. This is compared to the page length placed into R8. While the number of line moved is less than the number of lines in R8 the subroutine will continue to move lines. When these are equal the return address is put back into R11 and the programme flow returns to the calling programme by RT.

To complete this diskussion here is the code for the subroutine SCW1.

```
      SCW1   MOV    *R11+,RØ
             MOV    *R11+,R1
             MOV    *R11+,R2
             BLWP   @VMBW
             RT
```

The operation of this code is self explanitory, SCR1 is similar except that VMBW is replaced with VMBR.

That is all for this month, until next month.

Joe Wright

Hunter Valley '99ERS
USER GROUP
Newcastle

A LETTER WRITING TEMPLATE FOR USE WITH FUNNELWEB

This tip came from an article by Gary Fuquay in TOPICS, the newsletter of the LA 99'ers Users Group, and came to H.V. 99ers via the Melbourne TIMES, August,1986.

If you use TI-WRITER to write letters, you have probably found yourself typing the same things (like your return address) over and over again, each time wondering if the placement on the page was going to look the way you wanted. Well, all of your problems can be remedied by the simple little 'template' below.

```
.FI
.LM 6;RM 72
.IN +40
+
+
John Doe
1234 Main Street
SPRINGFIELD, MO 65804
+
+
+
Date
+
+
+
+
.IN +0
name
address
address
+
+
+
Dear Sirs,
+
.IN +5
.IN +40
+
+
+
Sincerely,
.SP 4
John Doe
```

To use this 'template' simply type it in as it appears above (substituting your name and address for John's, of course) using the Editor and save it to disk (the +'s are marking blank lines - just leave those lines blank except for a carriage return). Then, when you need to write a letter, load the file back into the Editor. Move down to 'date' and type the date OVER what is there. Move to 'name', 'address', 'address' and type in the name and address of the person you are sending the letter to. Then, move to '.IN +5' and press FCTN 2 to 'insert' the body of your letter. That's all there is to it! Save the new file to disk - with a new name of course - and then run it through the Formatter. You'll have a professional looking letter every time.

THE INFORMATION PAGE

IN YOUR NEWSLETTER THIS MONTH

PLUS MUCH MUCH MORE!!!!!

COMING EVENTS

Next Committee Meeting: Tuesday 3rd November
Next General Meeting: Tuesday 10th November

AGENDA FOR NOVEMBER MEETING

Demo of Home Record Keeping program by Brian Rutherford
Disk of the Month Demo
Demo of Viatel

CLASSES AVAILABLE FOR MEMBERS

BASIC group conducted by Paul Mulvaney at the Warners Bay High on Tuesday 20th & 27th October

ASSEMBLY group conducted by Joe Wright. The next meeting will be held on Tuesday 27th October at a venue to be decided - see Joe.

FORTH group conducted by Richard Terry will not meet till further notice - Richard is taking a long overdue holiday.

ANNUAL SUBSCRIPTIONS

Subscriptions to the Group cover the period 1 July to 30 June following year. Membership enquiries are welcome; please address all enquiries to the Secretary.

The annual subscription is:
Australian Residents...$20
Overseas Residents.....$40 (airmail)
$30 (surface)

Back issues of our Newsletter are available for $1 plus postage

C#97 Current

Mr.R. CARMANY
1504 Larson St.
GREENSBORO NC.27407
U.S.A.