ISHUG NEWS DIGEST

Focusing on the TI-99/4A Home Computer

Volume 6, Number 6 **July 1987**

Registered by Australia Post - Publication No. NBH5933

M19
PRINTER

P.O. Box 214, Redfern, New South Wales, Australia, 2016 **$2**

TISHUG

TISHUG NEWS DIGEST

TI99/4A Owners Home Computer
User Group
TIsHUG NEWS DIGEST

JULY 1987

Correspondence to:

PO Box 214
REDFERN NSW 2016

Texpac BBS: Tel.: (02)319.1009

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.(046)603956
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:

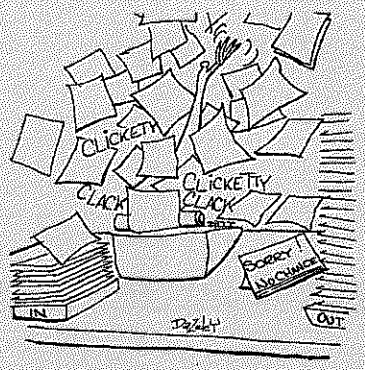
Joining Fee.....$ 8.00
Annual Family Dues.....$25.00
Dues O'seas Airmail...US$30.00
Publications Library....$ 5.00
Texpac BBS.....$ 5.00
BBS Membership:
Other TI User Group
Members.....$10.00
Public Access.....$25.00

GROUP GENERAL MEETING:

First Saturday of each Month at Woodstock Community Centre, Church Street Burwood. Starts 2pm

COMMITTEE MEETINGS:

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



TIsHUG NEWS DIGEST ISSN 0819-1984

CONTENTS PAGE

General Information and Editorial.....1
Coordinator's Report By Chris Buttner.....2
Piracy by Jim Peterson.....2
Secretary's Notebook by Terry Phillips.....3
Special Meeting of Members.....3
TIsHUG Shop by Cyril Bohlsen.....4
Letter to Jenny.....4
Techotime by John Paine.....5
Software Column by Terry Phillips.....6
Link It by Ross Mudie.....7
The Communicators by Ross Mudie.....9
Type In Programs
Grumplin.....10
Multiplication Invaders.....11
Phantom5.....12
Sea Diver.....13
The 4A Eprom Experiment by Arto Heino.....14
Disk Contollers - How They Work by Jerry Coffey.....15
AT-99 Disk Control System by John Paine and Peter Schubert.....17
Regional Reports.....18
Save Tutorial by Scott Darling.....19
Forum.....20
Adventure Hints.....20
Modules by D.N.Harris.....21
D/D Disk Control Board by Peter Schubert.....22
Comments by Eric Whelan.....23

Please note that on P3 there is the official nomination form for the Special General meeting in July. A photocopy or Facsimile is needed before nominations can be accepted.

In Most orginisations there are always a few people who can be relied upon. In preparing the Digest, the regular contributions come in like clockworkand makes the job of being Editor easier. To those people I am extremely grateful. This month, I would like to thank D.N.Harris and Eric Whelan for their efforts in describing how they are using their computer.

Bob Montgomery

CO-ORDINATOR'S REPORT

.... Chris Buttner

Last month many of you took advantage of the full day tutorials conducted by Ross Mudie and Russell Welham. I was heartened by the response to these sessions and on behalf of all club members sincerely thank both Ross and Russell. Incidentally, if you are a country member who was unable to make the meeting and wish to "keep up" on assembly language, send in a note asking for the "Linking to Assembly Language" handout prepared by Ross.

One fact emerged from the questions asked during the day and that is a number of you do not have a good grasp of Basic/Extended Basic. If you are in this category please make an effort to attend future meetings as we will be holding classes especially for your benefit.

During May, I was able to combine a working trip to Queensland with a meeting with Brisbane User Group members. Just for good measure, I took along a working Ram-Disk card for demonstration. The interest was very positive to the extent that we now have orders for 18 cards from Brisbane members. Charles Bagley brought along his Phillips 415 colour television and demoed its outstanding colour purity and definition when used in monitor mode. The on-screen picture is as good as any I have seen on a dedicated monitor. I expect a future issue of the Brisbane magazine will carry an article on constructing the circuit board which allows the console to interface with the TV.

This month we have the important task of electing our directors so that we can incorporate. All nominations must be accompanied by a written "Consent to act as Director" form signed by the person nominated. There are a few things which I feel must be stressed at this time. In the past we have worked entirely with a committee style of management. That committee has tried to hold two sets of reins - the long term development and policy of the club and secondly, the bread and butter day to day running. We are now looking to the directors being responsible for club policy and our statutory obligations, while a committee (for example, your present committee) is responsible for the day to day running of the club. It is most important those people nominated as Directors can, and will, attend regular meetings: after all, they are the ones who are "carrying the can". From past experience I believe five is a reasonable number of Directors.

Arto Heino has now started production of his "PICASSO" program and I was very pleased to see many of you rewarded his efforts by purchasing a copy at the last club meeting. The program allows you to combine text and graphics in any order/position on a page and to also change type styles. If I had to give a concise description of the program it would be a desk-top publishing package for the TI. All manner of graphics can be handled: GraphX, TI-Artist, even Basic/XBasic.

By the time you read this, our treasurer Bert Thomas hopefully be up to his ears in figures collating all the information for our tax return. If all goes well, I hope to be able to publish in the next magazine, (with the help of Microsoft Multiplan) information on just where the money goes.

A number of you are still sending mail to our old address. The correct address for all correspondence is:

The Secretary,
TISHUG,
P.O. Box 214,
REDFERN, N.S.W. 2016.

In coming months I hope the magazine will carry articles which show how to interface your computer to the real world. Many get the impression it is really only for running programs which display something on the screen. Although small, the TI is capable of much more than that and in its "interface mode" doesn't always need the monitor - a perfect excuse to put that spare console to work. Make sure we have your current address so you don't miss a single issue of the magazine.

TIsHUG Meeting 1/8/87.

Woodstock Community Centre,
Church St, Burwood. 2pm to 4.30pm.

BASIC & EXTENDED BASIC PROGRAMMING PROBLEM CLINIC.

This meeting will allow members who are having problems with programming in both basic and extended basic to bring their problems along to be resolved.

The meeting will handle both beginners' problems and advanced programmers' problems. There will be 2 groups in operation, nominally for Basic and Extd Basic, bring along problem programs on cassette tape, disk or printout.

If you are having problems unstanding how to use one or more basic or extended basic commands or statements then bring along your questions and let some of TIsHUG's experienced programmers help you.

PIRACY - by Jim Peterson

From the legal standpoint - It's a violation of Federal law. It's a criminal act. It's a crime.

From the religious standpoint- It's a violation of the command- ment "Thou shalt not steal". It's a sin.

From the moral standpont - You're not stealing from a face- less corporation. The software producers still supporting the TI are one-man operations, or one-boy operations - most of the best current software writers are between 14 and 21 years old and some of them are hoping to finance a college education. From the practical standpoint- Did you buy a computer because you wanted to catalog your butterfly collection? And you've been searching ever since for a butterfly-cataloging program?

Now, suppose a friend called you up and said he had just bought a really great butterfly-catalog program at a very reasonable price. Answer honestly - would you ask for the address and so you could order it? Or would you ask how soon you could copy it?

Think about that for awhile, and see if you begin to under- stand why there has been no economic incentive to write butterfly cataloging programs. From the pragmatic standpoint- Piracy will not stop. There will be no worthwhile market for software. If you want a butter- fly catalog program, you'll have to write it yourself. If you're mad about that, be mad at the pirates - and if you find your finger pointing back at yourself.....!

* Secretary's Notebook *

Terry Phillips

First a big warm welcome to the following 6 new members who joined us since my last report:

G Bellamy - Cartwright
P Luck - Hobartville
J Krupski - Blakehurst
J Ryan - Mullaway
L Saunders - Yagoona West
H Rafferty - West St Clair

We are now fast approaching 300 financial members (That target will have been achieved by the time you read this) which shows that our computer is still alive and well. Many of you now have RAM cards installed and having gone to this considerable investment - time and money - you will no doubt want to stay. Another good portion of the membership has installed 32K expansion inside their consoles. For you, and all others with 32K there will be some exciting software released over the next few months.

At the next meeting we have an important election to conduct. This will see the election of the required 5 people to take on and guide this group to incorporation. We have talked about this for some time now, but finally it looks very close to fruition.

At the August meeting the activities will be centred on problem solving in Basic and Extended Basic. There will be 2 groups set-up and the idea is for you - the member with a programming problem - to bring along a copy of that program on tape or disk - and a printout if available, and resident experts in programming will iron out your problems for you. This sounds a great way for group participation and I hope the day goes well.

Advice was received from Rex Shepherd, Tasmania User Group, to the effect that his group will cease operations as from 1 July. I will be writing to Rex to see if his members would like to join our group. It is of course sad to see a TI Group fold, and I gather from reading an earlier editorial by Rex in the Tassy User Group newsletter that the main reason was the lack of support his members were putting into the group. I suppose there is a valuable lesson to be learned here and that is that a group can only survive as long as the membership supports it. By the way can you remember when you last done something for this group, e.g. submitted an article, volunteered for some duty or other?

A big thanks to Eric Whelan from Healesville. Eric submitted a good article for the TND in program REM statements, which I converted to a disk file. Eric's article will be appearing in a future edition. If you want to see your work in print then why not submit an article as Eric did. It matters not to me how the articles come in as long as they are coming in.

Philip Heskett, a member from Bowral wrote concerning an article in the form of a chart showing current and future peripheral development with prices. He suggested this to awaken the "sleeping membership" as to what is available for their computer. Philip, this idea is a good one and you will be hearing from me shortly on how I think we can go about it.

Well, that's it for another month. Remember if you have an enquiry or problem you need an answer to then write to us for inclusion in the Forum column.

SPECIAL MEETING OF MEMBERS

Members are advised that a special meeting is called for July, 4th, 1987 to consider the following:

"To receive nominations for, and elect five (5) directors of TIsHUG"

It should be noted that in accordance with company's code, persons should not nominate if they are unable to attend meetings on a regular basis. Acceptance of nominees will only be by written consent of the nominee on the pro-forma with this notice. Nominators nor nominees should not show a particular position they are interested in filling.

Why is it necessary to call this special meeting? The answer to this is that because after the last AGM there is an infinite number of committee members which makes it impossible to proceed with Incorporation. Most of you will recall that the membership voted overwhelmingly at the 1986 AGM to proceed with Incorporation, so now it is necessary to have a working, duly elected board to guide this resolution.

The special meeting will be held at our usual venue - Woodstock Community Centre, Church Street, Burwood, commencing at 2pm, on July 4th, 1987. Be there if you can to cast your vote on this important issue.

NOMINATION FORM

I,_____,

a fully paid up member of TIsHUG, hereby nominate,

_____,

whom I also understand is a fully paid up member to TIsHUG, for a position of Director of TIsHUG.

Signature _____

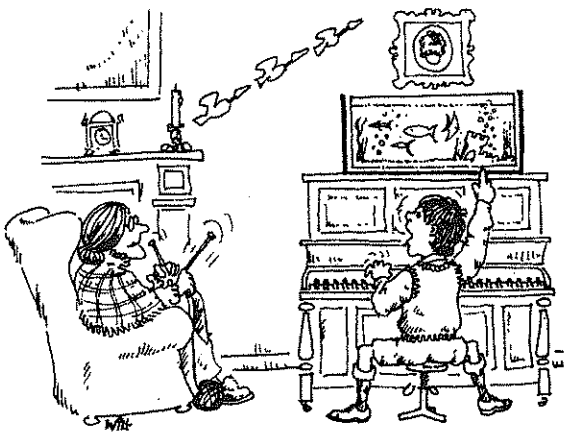
Membership No. _____

I, _____

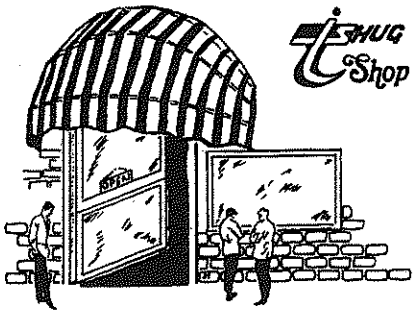
the person nominated above, give my consent to my nomination.

Signature _____

Membership No. _____



'How do you change the program on your machine, Granny?'



TIsHUG SHOP JULY 1987

RAM DISK CARD STATUS REPORT:-
As far as I know there have not been any total failures (but some have gone close) in the assembly of the 64 RAM CARDS distributed so far.

We are now on the second run of RAM Cards and parts, at the end of the JUNE meeting we have received orders for
38 Ram card PCBs
11 Auxillary component kits
8 Sets of sockets
353 6264L-15 RAM chips

At the time of starting the second list for these items the SHOP was only taking names of interested people and no deposit on the items as we did not know if there would be enough interest.

BUT THE TIME HAS COME FOR FULL PAYMENT for these items. The prices are:-

RAM CARDS.....$35.00
AUX.COMPONENTS.....$20.00
SOCKETS.....$ 5.00
6264L-15 RAM CHIPS (13)..$71.50 (SINGLE SIDE)
" " " (24).$132.00 (DOUBLE SIDE)

PLEASE SEND YOUR MONEY IN URGENTLY OR YOU MAY MISS OUT.

Robert Peverill may still be taking orders for the supply of the three (3) AAA size batteries and battery holders required for the Ram disk card. For the price and availability please contact Robert (02)6024168 Please note that the battery holders are very hard to obtain so you may be forced to use the 3 in one heat shrink variety.

PRINTER BUFFER (see March Aust.Electronics monthly.) There has been interest shown by some of the members to build the "Don McKenzie" 256K Centronics Printer Buffer in either PIO or SERIAL form. If you are interested please let me, and orders can be placed. The larger the order the cheaper the price will be.

SHOP INVENTORY.

(a) HFi DS/DD 5 1/4" Disks.....$19.00 per box
(b) Spike Protectors.....$29.00
(c) Consoles Ver.2-2.....$65.00
(d) 32k Matchbox Memory Expansion $55.00
(e) T.I. Joystick handles.....$00.50
(f) Peter Schubert's mini-expansion unit
DS/DD Disk controller card.....$180.00
RS232 board..... POA
RS232 board with 32K memory..... POA
All 3 above items together..approx.$300.00
(g) Stand alone RS232.....$99.00

SECOND HAND ITEMS :-

(a) Keyboards.....$15.00
(b) Grom Ports.....$12.00
(c) Ivory Console Cases.....$ 2.00
(d) Chips to convert your 1983 console to a 1981 version.....$12.00

BOOKS :-

(a) Back issues of SND.....$ 1.00
(b) Fun & Games with your TI-99/4A.....$ 4.00
(c) Technical manual.....$15.00
(d) Micropendiums.....$ 2.90
1986-June to Dec./1987-Jan.to May

SOFTWARE :-

(a) Club Software Tapes.....$ 3.00
(b) Club Software Disks.....$ 5.00
(c) Picasso Publisher V1.0 (Arto Heino).$20.00

POSTAGE
Please NOTE that with all mail orders YOU have to pay postage and packaging.
See JUNE ISSUE of SND for approx rates.

If you are phoning the SHOP please note that I am NOT normally available before 7pm week days. (02)639 5847

Dear Jenny,

I was sorry to hear that you were sick. I hope you are feeling better. Here is a short program for a short introduction program for a guessing game.

```
100 REM
110 CALL CLEAR
120 PRINT "HELLO, HOW ARE YOU?"
130 INPUT A$
140 INPUT "WHAT IS YOUR NAME?":A$
150 INPUT "THAT'S A GOOD NAME. WANT TO PLAY A
GAME?":C$
160 IF C$="Y" THEN 190
170 PRINT "TOO BAD....BYE!"
180 END
190 INPUT "A NUMBER BETWEEN 1 - 10?":D
200 RANDOMIZE
210 E=INT(RND*10)
220 IF E=D THEN 260
230 INPUT "NO, SORRY, PLAY AGAIN?":F$
240 IF F$<>"N" THEN 190
250 PRINT "TOO BAD, BYE"
255 END
260 PRINT "RIGHT, THANKS FOR PLAYING. BYE"
270 END
```

Here are some hints for Adventure #9: "GhostTown"

* Horse won't move? SHOE HIM

* SNAKE? Find something to get rid of him (clue bang)

* MOUNTAINS IN DISTANCE TOO FAR OUT? Now be honest, how would you go about getting there?

* A PLACE TO SLEEP THE NIGHT? Read the sign.

* JAIL A PROBLEM? Magnet might help.

That's all for this month.

yours faithfully,
Vincent Maker.

Thanks for your best wishes about my health. I always feel so much better when I receive a letter. Its a great little program you have written. Adventure game are something of a mystery to me. Taking the wrong step is what I am best at and never being able to complete the puzzle. Those hints are sure to help.

Thanks again, Jenny

TECHO TIME

.... John Paine

DISK DRIVES, and Should They be Cleaned.

The above question is one that has been raised many times in the past and I daresay, will be raised many times in the future.

My only real comment is to say that if all is working well, leave it alone. If all is not well check out every possibility of an other type of failure first.

This advice is given on the understanding that I have not ever given any disk drive that I have used with this or any other computer system a single head clean.

If you are having troubles with data corruption check that:

1) The floppy is ok.
2) All the connectors to drives are intact and clean.
3) The file you are working on has been closed properly.
4) Run the destructive disk test from Disk Manager Two.

Then if all else fails, maybe then you may wish to use a disk cleaning agent.

I am against these agents because they are ABRASIVE and will accelerate wear on the heads. If you do decide to use such an agent then follow the instructions very carefully. If they say to use only 2 to 3 drops then DONT use 20. If the instructions say to only run the drive for 20 seconds, DONT run them for a minute.

If you do wish to clean your heads, here is a little XB program that will make life easier and takes advantage of an error message to keep the drive running long enough to actually do some good.

```
100 CALL CLEAR
110 PRINT "***DRIVE RUNNING ***"
120 PRINT "HOLD FUNCTION 4 TO STOP"
130 ON ERROR 150
140 RUN "DSK1.XXX"
150 GOTO 130
```

When you run this routine, make sure that you have a stopwatch or wristwatch handy so you do not allow the drives to run any longer than necessary.

RAMDISK CONSTRUCTION TIP

Make sure that you only use germanium diodes with a low forward voltage drop, such as the OA47 Gold Bonded diodes, otherwise your RAMDISK may behave erratically.

NEW MINI-PE BOX ON SHOW AT JULY MEETING

The new peripheral for the TI-99 will be demonstrated by Peter Schubert and all will be welcome to try it out. This little box will be fitted with 32K, RS232 and DSDD Disk Controller. It has no power cord and plugs into side of console.

FAULT of the MONTH

This month's discussion will focus on one apparent difference between the earlier models of the 99/4A and the later grey case models.

I recently was given a black case console to test and repair, and quite routinely discovered that the 4 phase clock driver chip (TIM9904) was not working and therefore was reasonably confident that replacing this chip would solve all my troubles. I assumed that to repair this console would only take a few minutes and routinely removed the offending chip.

About 10 minutes later, after carefully removing the offending device, and replacing it with a known good one, I fully expected to see a some nice clock signals running up to the 9900 processor chip.

WRONG.....

The clock was just as dead as before. The next step was to check the crystal circuit and to my amazement, I found what I thought was the wrong crystal in circuit. Every console that I had seen in the past had a 12 MHz crystal but this board had a 48 MHz crystal in place.

A quick look at the technical data book assured me that the correct crystal should be 12 Mhz so a closer look at the older console configuration was necessary.

The chip removed was definately marked as TIM9904 but closer inspection of other grey consoles laying around, showed that TI had changed the designation of the chip to TIM9904A. All TIM9904A chips ran from a 12 Mhz oscillator circuit consisting of a crystal, 3.3 uHenry choke and a small 47pf capacitor.

The TIM9904 oscillator circuit uses a 48 Mhz crystal, a 0.33 uHenry choke and a .22pf capacitor. All that was needed was to change the crystal,capacitor and inductor and low behold, the console fired up.

What initially seemed to be a five minute job turned into a five hour job because of the undocumented differences between two similar versions of the same chip used in production of the consoles.

Subsequent checking of the relevant DATA books showed that the Semiconductor Division of TI actually changed the mask and operation of the 9904 chip in the early 80's and upgraded to the 9904A specification. This upgrade still used all the same pin connections but allowed use of cheaper components to get the clock up and running. I have also found other simple chip changes in the consoles over a period of time, and if one were to read the Orphan's Cronicles published by Millers Graphics, one can understand that during the great price war of 83, any cost savings in production would be highly regarded by the management of Texas Instruments.

A little bit of trivia now for all those people who wish to know how to identify by sight, whether the Grom 0 chip in their spare motherboard is a version 2.2 or 1981 version.

Grom 0 version 2.2 will be marked as follows.

CD2155	same for 1981 version
8327DCHY	date code and batch.
Phillipines	Place of manufacture.

Version 1 or 1981 title versions where also manufactured with the same part number, datecode and location but only version 2.2 groms have the batch designation of DCHY.

Welcome to this months software column. First up I should thank those who responded to my plea in the last issue to let me know what you wanted on tape. There have been quite a number of responses, all yet to be anaylised fully, but on the surface it appears that the younger members want games and educational programs while the more mature members are interested in routines and utilities. Hopefully over the next few months we can fill each and every members needs.

Some good news is that we recently wrote to several Fairware authors enclosing a contribution from TIsHUG. Two of the authors have responded so far with updated versions of their fine programs.

The first was Mark Beck who forwarded version 6 of his Creative Filing System. This version runs to 3 disks and will be available at the shop at the July meeting. Having spoken with Cyril, he has agreed that if you return your original version, which was on 2 disks, he will swap it over for you with this version for the cost of only one disk. This version contains heaps of documentation files. In fact one disk is full of them while the another has 3 more files on it. To print out the documentation, load the Formatter of TI Writer or Funelwriter, insert the disk designated 2 into a drive and enter DSKn.CFS_DOCP00 as the filename. This will print out the documentation files CFS_DOCP01 to CFS_DOCP10. After this is done reload the Formatter and insert the disk marked 3 into a drive. Enter DSKn.CFS_DOCP00 again as the filename. Files CFS_DOCP11 and CFS_DOCP12 are then printed. Please when you have printed out all the documentation read it carefully before using the main program.

The second response came from Rene' Leblanc, author of Universal Disassembler. Rene' also sent his source screens, and his modified Forth Compiler which he used in creating Universal. Also enclosed in his package was an Extended Forth package from Bil Wedmore. Any one interested in these packages can see me at meetings to arrange a copy as I figure there would not be enough general interest in them to release them through the shop.

Now to this months software:

ON DISK:

1. The previously mentioned Creative Filing System update. Don't forget, this is on 3 disks, so bring your old version back to trade up to the new one.

2. Assembly Utilities and routines from Brisbane. This is a full double sided disk of handy routines and utilities complete with heaps of documentation on the disk. To release it, knowing that a lot of you don't have double sided drives, it will come on a "flippy" and as double sided. So make sure you get the right version for your system.

3. Disk 1987/7 - see under Tape programs for a description of the contents.

ON TAPE:

One tape only this month, number in the series 1987/7. Here's what will be on it.

COLORVISIONS - in Extended Basic, this is a demonstration program with everchanging patterns and colors. Very fascinating to watch.

DIGITAL CLOCK - runs in either basic. A large on screen digital clock that can be set to either 12 or 24 hour mode. Keeps good time. I like fooling around with this one adding chimes and alarms.

DOGFIGHT - an old Extended Basic game where the object is to shoot down your opponent. This is a well written game with various options available to the players.

FUNNYFACE - is in Extended Basic and if you have seen the module Facemaker then you have some idea what this is about. Personally, I feel it is better than Facemaker and rate it excellent for the younger children.

GLENDY BURK - the Glendy Burk was an old riverboat and this is the song about that boat. Good colorful graphics, nice music and the words on the screen to sing along to. In Extended Basic.

MARTY MARTIAN - this was to be the first in a series of educational programs with an astronomy theme. The idea is to help Marty learn facts about the planets. Needs Extended Basic.

PROVERBS - in Extended Basic. Selected proverbs and quotations with the "thinker".

SKYSCAPE - again requires Extended Basic. Quite a well done program that charts the planets against the zodiacal background. Shows what planets are visible at any hour of the day/night. Could be a big help to those with an astronomical interest.

LAST MONTHS TAPE (1987/6) - time beat me in putting a description of the contents in the TND. So here is a brief description of what was on the tape.

BASIC PRIMER - a tutorial that will run in either basic.

CATAPULT CAPERS - In Basic. This is quite complex and good fun to play. In addition you will learn a bit of mathematics as you go.

FLIGHT SIMULATOR - In XB. Also see the instructional program preceding this on the tape. A very good flight simulator that has been around from some time. This version only recently updated to XB with the instruction file.

JOES CASINO - In Basic and optional TE2. A simple program but there appears more to it than what you see.

MAKING TRACKS - In Basic. A very hard to play game where you have to keep the ball in motion and not run out of track. Similar to the program called Diablo that won awards here and overseas.

MARYLAND SONG - good music and graphics written in Basic.

PLANETARIUM - again in Basic and written for the Northern Hemisphere. Someone like to try their hand at turning it upside down?

SPATIAL RELATIONS - in XB and for the younger children. Teach them all about circles, squares etc.

INCOME TAX - in XB. It was right for last years tax returns but probably now out of date with changes. Shouldn't take too much fiddling with to get right for this year.

On a closing note, one thing that was requested by several of my correspondents was the re-release of the "Best Of" earlier tapes. Watch for more news on this next month in this column.

Hope you enjoy this months selections, and remember if you have any software queries, drop me a line.

LINKING EXT'D BASIC-ASSEMBLY

WITH ROSS MUDIE.

Scanning a joystick & Half the Keyboard, quickly.

by Ross Mudie of TIsHUG, 13th June 1987.

Readers have probably been wondering what has happened to my regular LINK-IT articles. Quite simply there are only a limited number of hours available each month for home computing and most of these hours are used each month improving and maintaining the TEXPAC BBS. The BBS has required a lot of assembly programming and this has given me valuable programming experience in a quite complex environment for the TI99/4A.

This month's program was written for my younger son, Peter aged 10, who was writing an ext'd basic program which scanned the left side of the keyboard & Joystick number 1. The program used the results of both scans to move a sprite around a 10 by 10 board which is located at the bottom centre of the screen. His x/basic program then tests for the FIRE button (or Q key) and if operated displays a normal character at the Row and Column currently occupied by the sprite. In extended basic the program ran too slowly, hence the request for the functions to be performed in assembly.

1. THE SPECIFICATION.

a) The assembly program is to scan the left side of the keyboard and Joystick 1.
b) Keys E,S,D,X are to be used for Up, Left, Right and down. Keys W,Z,R,C must provide diagonal values. The Joystick must provide Vertical, Horizontal and Diagonal values. The Joystick Fire Button must also be active.
c) The assembly does all maths to return the pixel row and pixel column for control of the sprite. The pixel column value is adjusted to line up with the columns in extended basic. (Screen columns start 2 columns to the left of extended basic DISPLAY AT or PRINT columns).
The value of any key pressed on the left side of the keyboard is to be returned.
The row and column must also be returned for other control functions.

2. EXTENDED BASIC DEMO PROGRAM.

The small extended basic program shows how to load and link to the assembly routine. In line 210 the DISPLAY AT statement shows the values returned by the assembly. The program will move the sprite in extended basic on the 10 by 10 board drawn in extended basic. If you delete or remark out line 210 you will be able to see how much faster the sprite can be moved on the board.

The link to assembly executes considerably faster than an equivalent extended basic program.

```
100 ! SAVE DSK1.DEMO_JS/KB
110 CALL CLEAR :: CALL SCREEN(6)
120 FOR S=0 TO 12 :: CALL COLOR(S,16,1):: NEXT S
130 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
140 CALL INIT
150 DISPLAY AT(10,1):"Loading Assembly"
160 CALL LOAD("DSK1.JO"):: CALL CLEAR
170 FOR R=15 TO 24 :: DISPLAY AT(R,10):STR$(R):: NEXT R
180 FOR C=12 TO 21 :: DISPLAY AT(14,C):CHR$(C+64);
190 NEXT C
200 CALL LINK("SCAN",Y,X,F,R,C)
210 DISPLAY AT(1,1):"Y=";Y,"X=";X:"F=";F:"R=";R,"C=";C
220 CALL SPRITE(#1,128,12,Y,X)
230 GOTO 200
```

3. THE ASSEMBLY SOURCE FILE.

The first time that the program runs it initialises the RMIN and CMIN values, then sets the FLAG to prevent reinitialisation.

The keyboard is scanned in a mode of 1, left side and Joystick 1. If a key is detected it is passed back to extended basic in the third argument in the CALL LINK which is the variable F. If no key is detected the value is 255; CALL KEY if no key is pressed returns -1.

The program then tests for one of the keys E,X,S,D,W, R,C,Z being operated. If one of these keys is found the program either increments or decrements Register 3, which stores the value of the current row or Register 4 which stores the value of the current column. Each time the value in R3 or R4 is changed it is checked against the MINimum or MAXimum value and if it is out of limits then it is placed back within limits. If an operated key is not found then the Joystick is tested and the values of Registers 3 and 4 adjusted as required.

The program takes the row and column values in R3 & R4 which are multiplied by 8 using binary multiply and a value is added or subtracted as required to give the sprite row or column value. The sprite row & column, character row & character column are passed back to the appropiate variable in the CALL LINK argument list.

```
       IDT  'JS/KBmud'              source=J  object=JO
*                                Ross Mudie 12th June 1987.

       DEF  SCAN              CALL LINK("SCAN",Y,X,F,R,C)
*                            Y=Pixel Row, X=Pixel Column,
NUMASG EQU  >2008                F=Left keyboard value,
XMLLNK EQU  >2018           R=Screen Row, C=Screen Column.
KSCAN  EQU  >201C
VSBW   EQU  >2020

FAC    EQU  >834A              Floating Point Accumulator
STATUS EQU  >837C                                STATUS
GPLWS  EQU  >83E0          Address of workspace used by X/B


ZERO   DATA 0                                 A word of Zero
ONE    DATA 1                                  A word of one

* Byte constants used for comparisons.
MINUS4 BYTE >FC          Joystick value if down or left
B2     BYTE 2                          Keybd 1  S  left
B3     BYTE 3                        Keybd 1  D  right
B4     BYTE 4 Keybd 1  W  up/left  or Joyst up or right
B5     BYTE 5                          Keybd 1  E  up
B6     BYTE 6                      Keybd 1  R  up/right
B14    BYTE 14                   Keybd 1  C  down/right
B15    BYTE 15                   Keybd 1  Z  down/left
       EVEN

FLAG   DATA 0     Used to store when program first runs

RMIN   DATA 15                                MINimum Row
RMAX   DATA 24                                MAXimum Row
CMIN   DATA 12                             MINimum Column
CMAX   DATA 21                             MAXimum Column


WS     BSS  32  Space for the Registers used by program
SAVRTN BSS  2       Space to save return address to x/b

*Register Usage: 0,1,2-Utilities, 3-Up/down row counter
*                              4-Left/right column counter
```

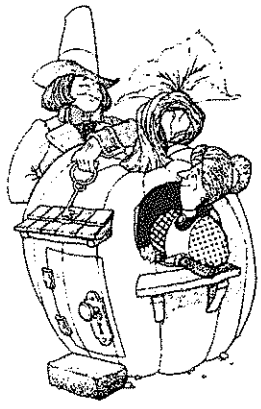
```
SCAN   MOV  R11,@SAVRTN  Save the return address to X/B
       LWPI WS           Load the register Work Space

       C    @ONE,@FLAG   Has the program initialised?
       JEQ  TEST1                              Yes
       MOV  @RMIN,R3    Place the starting Row in Reg 3
       MOV  @CMIN,R4    Place the starting Col in Reg 4
       MOV  @ONE,@FLAG  Set the flag to one to indicate
*                              program is initialised.

TEST1  MOVB @ONE+1,@>8374 Tells KSCAN left keybd & JS1
       BLWP @KSCAN       Scan left keyboard & JoyStick 1

       MOVB @>8375,R1 Put result of KSCAN in Register 1
       SRL  R1,8    Moves left byte to right byte in R1
       MOV  R1,@FAC      Put kscan result as word in FAC
       BLWP @XMLLNK  ¶      Convert Integer in FAC into
       DATA >20      ¶   Floating point number in FAC.
       CLR  R0             Element in Variable in X/B
       LI   R1,3          Argument 3 in the LINK list
       BLWP @NUMASG     Pass the key press value to x/b

       MOVB @>8375,R1      Get the same Key press again
       CB   R1,@B5               Is it a  E  (up) key?
       JEQ  UP                             If YES then Jump
       CB   R1,@ZERO           Is it a  X  (down) key?
       JEQ  DOWN                           If YES then Jump
       CB   R1,@B2              Is it a S  (left) key?
       JEQ  LEFT                           If YES then Jump
       CB   R1,@B3            Is it a  D  (right) key?
       JEQ  RIGHT                          If YES then Jump
       CB   R1,@B4          Is it a  W  (up/left) key?
       JEQ  UP                             If YES then Jump
       CB   R1,@B6          Is it a  R  (up/right) key?
       JEQ  UP                             If YES then Jump
       CB   R1,@B14       Is it a  C  (down/right) key?
       JEQ  DOWN                           If YES then Jump
       CB   R1,@B15         Is it a  Z  (down/left) key?
       JEQ  DOWN                           If YES then Jump

TEST2  MOVB @>8376,R1 Put result of Joyst up/down in R1
       CB   R1,@B4                      Is Joystick UP?
       JNE  YN4                If no then jump to Y Not 4
UP     DEC  R3   Take one off value in R3 (row counter)
       C    R3,@RMIN  Compare R3 value with Minimum Row
       JHE  SENDY                 Jump if still in range
       INC  R3     Out of range, add one to value in R3
       JMP  SENDY                  Jump to SEND Y value

YN4    CB   R1,@MINUS4                  Is Joystick DOWN?
       JNE  SENDY                    No, Jump to SEND Y
DOWN   INC  R3                     Add 1 to Row counter
       C    R3,@RMAX        Is Row counter out of range?
       JLE  SENDY                    No, Jump to SEND Y
       DEC  R3   Out of range, subtract one from R3 val

SENDY  MOV  R3,R1     Put the value of row in Register 1
       SLA  R1,3             Multiply the row value X 8
       AI   R1,>FFF9  Subtract 7 to adjust for pixel row
       MOV  R1,@FAC     Put integer pixel row val in FAC

       BLWP @XMLLNK  ¶ Convert the Integer in FAC into
       DATA >20      ¶    Floating point number in FAC
       CLR  R0             Element in variable in x/b
       LI   R1,1              Argument 1 in link list
       BLWP @NUMASG     Pass the Pixel row value to x/b

       MOV  R3,@FAC    Put integer Row value in the FAC
       BLWP @XMLLNK  ¶ Convert the Integer in FAC into
       DATA >20      ¶    Floating point number in FAC
       LI   R1,4          Argument 4 in the link list
       BLWP @NUMASG         Pass the ROW value to x/b

       MOVB @>8375,R1      Get the key press value again
       CB   R1,@B4           Is it a  W  (up/left) key?
       JEQ  LEFT                           If YES then Jump
       CB   R1,@B15         Is it a  Z  (down/left) key?
       JEQ  LEFT                           If YES then Jump
       CB   R1,@B6           Is it a  R  (up/right) key?
       JEQ  RIGHT                          If YES then Jump
       CB   R1,@B14       Is it a  C  (down/right) key?
       JEQ  RIGHT                          If YES then Jump

TEST3  MOVB @>8377,R1   Put Joyst left/right scan in R1
       CB   R1,@B4                              Is it RIGHT?
       JNE  XN4                If no, then Jump to X Not 4
RIGHT  INC  R4                   Add one to column counter
       C    R4,@CMAX      Is column above maximum limit?
       JLE  SENDX                                       No
       DEC  R4    Subtract 1 to put inside column limit
       JMP  SENDX                      JuMP  to SEND X value

XN4    CB   R1,@MINUS4        Is Joystick operated left?
       JNE  SENDX                                       No
LEFT   DEC  R4            Subtract 1 from col value in R4
       C    R4,@CMIN         Is column value below limit?
       JHE  SENDX      If in limits Jump to SEND X value
       INC  R4       Add 1 to R4 val to put col in range

SENDX  MOV  R4,R1                    Put column value in R1
       SLA  R1,3              Multiply column value in R1 X 8
       AI   R1,9             Add 9 to adjust for pixel column
*                                  to match up with x/b columns
       MOV  R1,@FAC      Put integer column value in FAC

       BLWP @XMLLNK  ¶ Convert Integer column value in
       DATA >20      ¶ FAC  into floating point in FAC
       CLR  R0             Element in variable in x/b
       LI   R1,2              Argument 2 in link list
       BLWP @NUMASG  Pass the Pixel column value to x/b

       MOV  R4,@FAC      Put Integer column value in FAC
       BLWP @XMLLNK      ¶  Convert Integer in FAC into
       DATA >20          ¶  Floating point value in FAC
       LI   R1,5              Argument 5 in link list
       BLWP @NUMASG         Pass the column value to x/b

END    MOVB @ZERO,@STATUS  Prevent false err indication
       LWPI GPLWS      Reload register workspace for x/b
       MOV  @SAVRTN,R11     Get saved x/b return address
       RT                   Branch to the address in R11

       END
```

* DISK DRIVE FOR SALE *

I HAVE A SPARE MITSUBISHI SLIM-LINE DSDD DRIVE FOR SALE. IT IS IN EXCELLENT CONDITION AND IM ASKING $180 FOR IT. ALSO HAVE A POWER SUPPLY UNIT FOR IT, ORIGINAL TI UNIT, LOOKS GOOD, VERY COMPACT, FOR $40. PHONE (02) 358 5602 PETER SCHUBERT.

* NEW MODEM FOR SALE *

I STILL HAVE ONE AUTO-THUNDERER MODEM FOR SALE FOR $260. IT HAS AUTO-ANSWER 1200/75/VIATEL AND CAN BE USED WITH ANY COMPUTER WITH RS232 PORT. P.SCHUBERT 358 5602, PO BOX 28, KINGS CROSS 2011.



THE COMMUNICATORS

Special Interest Group for Users
of the TEXPAC Bulletin Board Service.
by Ross Mudie, SYSOP, 10th June 1987.

1. GENERAL OPERATIONAL INFORMATION.

The TEXPAC BBS is now capable of recovering from the user hanging up at any point in its operation without ill effect, including the program download area. Any caller who is using STD to access the BBS should now disconnect from the phone line as soon as you have finished getting the required programs or information, there is no need to go through the log off procedure.

As a result of a suggestion by Steven Shraibman, username SUS, a new prompt is received after reading or [E]scaping from any file:

[M]ain, [N]ews menu or News # >

M <enter> goes to the Main Menu, N <enter> goes to the News Menu, whilst a valid number goes directly to that News menu item. Numbers entered out of the range of the News Menu receive the prompt "COMMAND username? >". This allows users to read and printout the menu once, then to go from one News menu file to the next without having to go back through the menu. The menu is however always available.

If you start to read any file, menu or message which you wish to escape from then press E once only. To pause any file, menu or message use <ctrl> S to PAUSE & <ctrl> Q to UNPAUSE.

At the end of reading mail or after [E]scaping from from reading mail with E, the prompt is now:

[S]ave or [D]elete your mail >

You must press S <enter> to Save your mail or D <enter> to Delete your mail before the program will proceed. If you decide to save your mail, (because you want to check that it has been successfully saved on your log file or you forgot to turn on the logging option first) then please delete your mail at the first available opportunity. Main menu option 9 allows you to log on again to re-read the mail without having to call again. If you just need to delete the mail then press E as it starts to list then D <enter> when the prompt is received to free up file space on the BBS mail disk.

When sending mail from the keyboard, check each line before proceeding to the next, the mail editor in use at present will not allow you to go up a line. You may use <ctrl> H to perform a destructive back space in the current screen line.

When sending mail from a disk file use SENDMAIL4. To upgrade SENDMAIL3 to SENDMAIL4 change the following lines:

```
100 ! SAVE DSK1.SENDMAIL4
110 ! Version 1.4 by Ross Mudie, 10th June 1987, for
use with TEXPAC BBS.
380 IF A$="" OR A$=CHR$(13)THEN A$=" "
```

This modification should remove the need to pass TI WRITER files through the E/A Editor to modify lines which contain only a carriage return.

The program SENDMAIL4 is available on the downloadable software on the BBS. This program requires a 32K disk system to operate.

2. AUTOMATIC RESET OF PROGRAM DOWNLOAD.

The TEXPAC BBS has a different selection of basic & extended basic programs for downloading each month. A long term problem with this feature has been that if a user failed to complete the download properly and then hung up from the phone call, then follow on callers to the BBS would not be able to log on normally. This problem has now been overcome & I feel that the method used will be of interest to both users & programmers.

When the download routine is entered the user types OLD RS232 whilst the BBS invokes the equivalent of SAVE RS232. Once in the SAVE routine in the RS232 ROM the ROM program looks only at the RS232 port and the CLEAR key on the BBS keyboard. The Carrier Detect line from the modem enters the BBS via the DTRin line of RS232/2, whilst I/O communications is via RS232/1. The method of providing the required escape from the SAVE program is to monitor the DTR of RS232/2 whilst download is occuring.

I used a dissassembly of the RS232 DSR ROM prepared by Geoff Trott of the Illawarra Regional Group to find the CHECK KEY routine. The relevant part of the source file was typed in to a file using the E/A Editor and the AORG directive to simulate the locations of the code in the ROM. At the entry point to CHKEY I created a JuMP to my new code which was to fit in the ROM after the end of the standard code. The new code, which uses 22 bytes, compares a flag in the general memory space & a new constant in the RS232 ROM. If the download flag flag is set then the DTR of RS232/2 is tested for loss of carrier, which if detected, causes the SAVE routine to branch to the error routine and terminate the download. The error routine closes the open file in the correct manner and then exits to the calling program.

When I assembled the small section of modified source code for the RS232 ROM, I used the List option which gives absolute address, (because of AORG), & the object code bytes in addition to the source listing. I then loaded the memory contents of a standard RS232 ROM into the RAM of an EPROM programmer. Using the alter bytes option of the EPROM programmer, the bytes were changed to match the assembler List file. The code was then saved into a 2532 EPROM which was then placed in a socket in the RS232 card of the BBS in place of the DSR masked ROM.

3. BBS HELPERS.

Each month a number of members provide contribution of material for use on the BBS. This material comes from both local and overseas sources, the criteria for BBS material is that it is topical, interesting or new. The BBS provides a good source of programs and ideas.

I acknowledge with thanks the recent contributions of the following people: Shane Ferrett, Shane Andersen, Peter Schubert, Terry Phillips (TIsHUG Program Librarian), John Paine (TIsHUG Technical Co-ordinator), Gavan Element, George Meldrum and Rolf Schreiber.

Contributions of suitable programs and files are constantly needed to make the TEXPAC BBS a dynamic data base for TI99/4A users, can you help?

4. BBS ACCESS.

Access to the BBS is available to registered users only, each user is issued with a user number, user name and password. BBS membership is handled by the secretary of TIsHUG. If you wish to join the BBS then write to the Secretary TIsHUG, PO Box 214, Redfern. 2016. You should advise of your preferred user name, (usually a nickname), and your initial password. The Secretary will advise you of your user number, actual user name (in case the requested user name is already used), and password. The BBS membership fee for existing TIsHUG members is $5 for the first part year and $5 PA. TIsHUG membership is required for this rate of BBS membership.

To use the BBS you will need the following equipment:

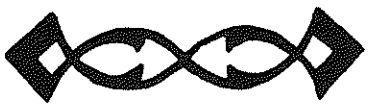
Minimal system: TI99/4A, RS232 interface, TE2 module, Cassette Recorder, Modem.

Expanded system: In addition to minimal system: Disk Drive, 32K Expansion Memory, Extended Basic, Printer, Disk Based Terminal Emulator (Fast Term or 4A Talk).

To access the BBS you need a telephone connection for your modem and a suitable budget for the cost of STD phone calls if calling from outside the Sydney telephone zone.

```
100 REM ************
110 REM * GRUMPLIN *
120 REM ************
130 CALL CHAR(112,"003C7E7E7
E7E3C")
140 CALL COLOR(11,12,2)
150 INC=1
160 CALL CHAR(136,"0000003C7
E7EA5A5")
170 CALL CHAR(128,"493094692
6D92548")
180 CALL CHAR(104,"A2FFFFFFF
FFFFF2A")
190 CALL CHAR(105,"F7FF7FFFF
EFFF7FF")
200 CALL CHAR(106,"55FFFFFFF
FFFFFAA")
210 CALL CHAR(107,"FFFFFFFFF
FFFFFFF")
220 CALL CHAR(108,"7EFFFFFFF
FFFFFE7")
230 CALL CHAR(109,"FF7EFFE7F
F7EFFFF")
240 CALL CHAR(129,"")
250 CALL CHAR(42,"7E81A58181
BD817E")
260 CALL CLEAR
270 PRINT "WELCOME TO THE GR
UMPLIN'S   CAVE. ESCAPE TO T
HE TOP AND YOU SURVIVE. OTHE
RWISE, YOU WILL BURN!": :
280 PRINT "YOU ARE IN A SUBT
ERRANEAN   CAVE OF FIVE LEVE
LS. YOUR   GOAL IS TO ESCAPE
 FROM THIS DARK PLACE.": :
290 PRINT "YOU BEGIN IN LEVE
L 3. IF YOUCAN REACH THE OPE
NING";CHR$(112);"YOU ADVANCE
 TO THE NEXT"
300 PRINT "LEVEL. IF GRUMPLI
N GETS YOU, YOU FALL TO THE
NEXT LOWER LEVEL.": :
310 INPUT "WHAT IS YOUR NAME
 PLEASE?   ":NM$
320 IF LEN(NM$)<14 THEN 350
330 PRINT "NAME TOO LONG, TR
Y AGAIN"
340 GOTO 310
350 GOTO 1640
360 PRINT : :
370 PRINT
380 CALL COLOR(14,2,1)
390 CALL COLOR(10,7,1)
400 CALL COLOR(2,2,1)
410 CALL COLOR(9,2,2)
420 CALL CLEAR
430 DL=3
440 CALL SCREEN(DL*2+2)
450 X=110
460 MONROW=2
470 MONCOL=16
480 CALL CHAR(129,"")
490 IF (DL=2)+(DL=4)=0 THEN
510
500 CALL CHAR(129,"001818242
48181FF")
510 CALL CLEAR
520 FOR Y=1 TO 23 STEP 2
530 CALL HCHAR(Y,2,103+DL,30
)
540 CALL SOUND(-100,INT(RND*
1000)+110,1)
550 RANDOMIZE
560 J=INT(RND*14)+9
570 IF J=16 THEN 560
580 FOR I=1 TO 2
590 K=INT(RND*29)+2
600 IF (K=L)+(K=M)+(K=J)<0 T
HEN 590
610 L=K
620 CALL HCHAR(Y,K,32)
630 IF Y>20 THEN 650
640 CALL HCHAR(Y+1,INT(RND*2
8)+3,129)
650 IF Y=23 THEN 670
660 CALL HCHAR(Y+1,J,104+DL)
670 NEXT I
680 M=J
690 NEXT Y
700 LV$="LEVEL "&STR$(DL)
710 FOR I=1 TO 7
720 CALL HCHAR(23,10+I,ASC(S
EG$(LV$,I,1)))
730 NEXT I
740 CALL HCHAR(1,2,103+DL,30
)
750 CALL HCHAR(1,16,112)
760 COL1=INT(28*RND)+3
770 ROW1=24
780 CALL HCHAR(ROW1,COL1,42)
790 CALL HCHAR(MONROW,MONCOL
,136)
800 CALL JOYST(1,X1,Y1)
810 CALL KEY(1,K1,S1)
820 IF (X1=0)+(Y1=0)+(DL<3)=
-3 THEN 1690
830 IF (K1<>18)+(X1=0)+(Y1=0
)=-3 THEN 800
840 OLDCOL1=COL1
850 OLDROW1=ROW1
860 IF Y1=-4 THEN 930
870 IF Y1=4 THEN 970
880 IF X1=-4 THEN 1130
890 IF X1=4 THEN 1180
900 GOTO 1690
910 CALL HCHAR(ROW1,COL1,42)
920 GOTO 1690
930 ROW1=ROW1+1
940 IF ROW1=25 THEN 1430
950 GOSUB 1380
960 GOTO 1360
970 ROW1=ROW1-1
980 GOSUB 1380
990 IF ROW1>1 THEN 1360
1000 DL=DL-1
1010 CALL SOUND(500,1000,0)
1020 CALL HCHAR(2,16,32)
1030 CALL HCHAR(1,16,42)
1040 IF DL<>0 THEN 440
1050 CALL HCHAR(ROW1,COL1,42
)
1060 CALL CHAR(42,"7E81A518A
599817E")
1070 CALL SOUND(500,262,3)
1080 CALL SOUND(250,330,2,26
2,2)
1090 CALL SOUND(500,330,1,26
2,1,392,1)
1100 CALL SOUND(250,330,2,26
2,2)
1110 CALL SOUND(750,262,3)
1120 GOTO 1230
1130 COL1=COL1-1
1140 GOSUB 1380
1150 IF COL1>1 THEN 1360
1160 COL1=31
1170 GOTO 1360
1180 COL1=COL1+1
1190 GOSUB 1380
1200 IF COL1<32 THEN 1360
1210 COL1=2
1220 GOTO 1360
1230 CALL HCHAR(OLDROW1,OLDC
OL1,32)
1240 PRINT "YOU MADE IT!"
1250 MONROW=2
1260 MONCOL=16
1270 FOR DELAY=1 TO 600
1280 NEXT DELAY
1290 PRINT "DO YOU WISH ANOT
HER GAME?"
1300 CALL KEY(0,KEY,STA)
1310 IF STA=0 THEN 1300
1320 CALL CHAR(129,"")
1330 IF KEY=78 THEN 1350
1340 IF KEY=89 THEN 420 ELSE
 1300
1350 END
1360 CALL HCHAR(OLDROW1,OLDC
OL1,32)
1370 GOTO 910
1380 CALL GCHAR(ROW1,COL1,TE
ST1)
1390 IF TEST1=112 THEN 1000
1400 IF (TEST1=129)+(DL<>5)=
-2 THEN 1470
1410 IF (TEST1<>32)+(TEST1<>
129)=-2 THEN 1430
1420 RETURN
1430 CALL SOUND(500,110,1,-7
,1)
1440 ROW1=OLDROW1
1450 COL1=OLDCOL1
1460 GOTO 1690
1470 GOTO 1610
1480 FOR TONE=110 TO 130 STE
P 2
1490 CALL SOUND(-100,TONE,1,
-6,1)
1500 NEXT TONE
1510 FOR TONE=130 TO 110 STE
P -2
1520 CALL SOUND(-100,TONE,1,
-6,1)
1530 NEXT TONE
1540 CALL HCHAR(ROW1,COL1,32
)
1550 ROW1=ROW1+4
1560 CALL GCHAR(ROW1,COL1,TE
)
1570 IF TE<>104+DL THEN 1590
1580 COL1=COL1+1
1590 GOSUB 1380
1600 GOTO 1360
1610 CALL HCHAR(OLDROW1,OLDC
OL1,32)
1620 CALL HCHAR(ROW1,COL1,12
8)
1630 GOTO 1480
1640 PRINT
1650 PRINT "USE JOYSTICK #1
TO STEER    YOURSELF AROUND
THE CAVES."
1660 INPUT "PRESS ENTER":ENT
ER$
1670 PRINT : : : : : : : : :
1680 GOTO 360
1690 CALL HCHAR(MONROW,MONCO
L,32)
1700 CALL SOUND(50,X,4)
1710 IF TEST<>129 THEN 1730
1720 CALL HCHAR(MONROW,MONCO
L,129)
1730 OLDMONROW=MONROW
1740 X=X+1
1750 OLDMONCOL=MONCOL
1760 IF MONROW>ROW1 THEN 194
0
1770 IF MONROW<ROW1 THEN 205
0
1780 IF TOG=1 THEN 1810
1790 IF MONCOL>COL1 THEN 210
0
1800 IF MONCOL<COL1 THEN 217
0
1810 MONCOL=MONCOL+INC+TOG*I
NC
1820 IF MONCOL<31 THEN 1850
1830 MONCOL=2
1840 TOG=0
1850 IF MONCOL>1 THEN 1880
1860 MONCOL=31
1870 TOG=0
1880 GOSUB 2240
1890 IF (TEST=32)+(TEST=129)
=-1 THEN 790
```

```
1900 TOG=1
1910 MONCOL=OLDMONCOL
1920 INC=-INC
1930 GOTO 790
1940 MONROW=MONROW-1
1950 IF TOG=0 THEN 2010
1960 MCOL=MONCOL-INC
1970 CALL GCHAR(MONROW,MCOL,
TEST)
1980 IF TEST=103+DL THEN 201
0
1990 MONCOL=MCOL
2000 OLDMONCOL=MCOL
2010 GOSUB 2240
2020 IF (TEST=32)+(TEST=129)
=-1 THEN 790
2030 MONROW=OLDMONROW
2040 GOTO 1810
2050 MONROW=MONROW+1
2060 GOSUB 2240
2070 IF (TEST=129)+(TEST=32)
=-1 THEN 790
2080 MONROW=OLDMONROW
2090 GOTO 1810
2100 MONCOL=MONCOL-2
2110 GOSUB 2240
2120 INC=-1
2130 IF (TEST=129)+(TEST=32)
=-1 THEN 790
2140 TOG=1
2150 MONCOL=OLDMONCOL
2160 GOTO 1800
2170 MONCOL=MONCOL+2
2180 GOSUB 2240
2190 INC=1
2200 IF (TEST=129)+(TEST=32)
=-1 THEN 790
2210 TOG=1
2220 MONCOL=OLDMONCOL
2230 GOTO 1810
2240 CALL GCHAR(MONROW,MONCO
L,TEST)
2250 IF TEST=42 THEN 2270
2260 RETURN
2270 CALL HCHAR(ROW1,COL1,12
8)
2280 CALL SOUND(1000,-6,0,11
0,7)
2290 PRINT "          GOTCHA!
"
2300 DL=DL+1
2310 IF DL<>6 THEN 440
2320 CALL CLEAR
2330 CALL CHAR(46,"FFFFFFFFF
FFFFFFF")
2340 CALL CHAR(62,"FFFEFCF8F
0E0C080")
2350 CALL CHAR(95,"FF7F3F1F0
F070301")
2360 CALL CHAR(60,"80C0E0F0F
8FCFEFF")
2370 CALL CHAR(94,"0103070F1
F3F7FFF")
2380 CALL SCREEN(10)
2390 CALL CHAR(129,"4062666E
7CFEFFFF")
2400 PRINT "    ²..........
.......<      ²..........
.......<     ²..........
.........<"
2410 PRINT "  ...
       ...    ...    ²>
 _<    ...    ...    ²>
  <    ..."
2420 PRINT "  ...    ²..
 ..<   ...    ...    ...
 ...   ...    ...    ...
 ...   ..."
2430 PRINT "  ...         ²<
       ...    ...         ..
       ...    ...         ..
       ..."
2440 PRINT "  ...        ....
       ...    ...
       ...    ...
       ..."
2450 PRINT "  ...    ².......
..<    ...    ...    .........
....   ...    ...    ..>
_...   ..."
2460 PRINT "  ...
       ...    ...
       ...    ...
       ..."
2470 PRINT "  _..............
.........>     _.............
........>"
2480 CALL HCHAR(24,8,46,18)
2490 M$=NM$&" IN LEVEL 6"
2500 FOR X=1 TO LEN(NM$)+11
2510 CALL HCHAR(23,11-.5*LEN
(NM$)+X,ASC(SEG$(M$,X,1)))
2520 NEXT X
2530 CALL COLOR(13,7,1)
2540 RANDOMIZE
2550 ROW=INT(RND*24)+1
2560 COL=INT(RND*24)+5
2570 CALL HCHAR(ROW,COL,129)
2580 GOTO 2550
2590 END
```



```
100 REM XXXXXXXXXXXXXXXXX
110 REM X                 X
120 REM X MULTIPLICATION X
130 REM X    INVADERS     X
140 REM X                 X
150 REM X BY D. PERKOVIC X
160 REM X      1985       X
170 REM X                 X
180 REM XXXXXXXXXXXXXXXXX
190 SCORE=0
200 QU=0
210 RANDOMIZE
220 CALL CLEAR
230 CALL SCREEN(2)
240 CALL COLOR(3,16,2)
250 CALL COLOR(4,16,2)
260 CALL COLOR(2,16,2)
270 CALL COLOR(5,16,2)
280 CALL COLOR(6,16,2)
290 CALL COLOR(7,16,2)
300 CALL COLOR(8,16,2)
310 CALL COLOR(13,16,2)
320 DISPLAY AT(12,3):"  MULT
IPLICATION INVADERS  ": : :
330 PRINT "     BY DAVID PER
KOVIC   ": : : : :
340 CALL CHAR(128,"3C7EBDBDF
F3C4281")
350 CALL SPRITE(#1,128,16,10
,1)
360 CALL SPRITE(#2,128,16,35
,1)
370 CALL SPRITE(#3,128,16,75
,1)
380 CALL MOTION(#1,0,40)
390 CALL MOTION(#2,0,-50)
400 CALL MOTION(#3,0,60)
410 DISPLAY AT(21,1):" < PRE
SS ANY KEY TO PLAY? >"
420 CALL KEY(0,K,S):: IF S=0
 THEN 420
430 CALL DELSPRITE(ALL)
440 CALL CLEAR
450 CALL COLOR(13,16,2)
460 R=12
470 CALL HCHAR(R,14,128,4)
480 QU=QU+1
490 M=INT(12*RND)+1
500 N=INT(12*RND)+1
510 L=M*N
520 CALL HCHAR(R,14,128,4)
530 PRINT M;" * ";N;"="
540 INPUT A
550 PRINT : :
560 IF A<>L THEN 780
570 PRINT "CORRECT!!  ALIENS
 DESTROYED"
580 PRINT "-------": :
590 TONE=110
600 FOR COUNT=1 TO 10
610 CALL SOUND(-500,TONE,1)
620 TONE=TONE+110
630 NEXT COUNT
640 CALL COLOR(2,2,2)
650 CALL COLOR(13,2,2)
660 FOR X=1 TO 10
670 CALL COLOR(2,16,2)
680 FOR DELAY=1 TO 10
690 NEXT DELAY
700 CALL COLOR(2,2,2)
710 FOR DEL=1 TO 30
720 NEXT DEL
730 NEXT X
740 CALL COLOR(2,16,2)
750 SCORE=SCORE+1
760 IF QU=10 THEN 970
770 GOTO 440
780 PRINT "WRONG!!"
790 CALL SOUND(200,440,0)
800 R=R+1
810 CALL HCHAR(R-6,14,32,4)
820 CALL HCHAR(R-1,14,32,4)
830 CALL HCHAR(R,14,128,4)
840 IF R=19 THEN 880
850 IF R=24 THEN 880
860 IF R>24 THEN 880
870 GOTO 800
880 CALL SOUND(400,120,0)
890 IF R=24 THEN 910
900 GOTO 520
910 PRINT "   THE ALIENS HAV
E INVADED     EARTH": :
920 PRINT "   THE CORRECT AN
SWER IS ";L
930 FOR DELAY=1 TO 1000
940 NEXT DELAY
950 IF QU=10 THEN 970
960 GOTO 440
970 CALL CLEAR
980 PRINT "YOU ANSWERED ";SC
ORE;" CORRECT OUT OF TEN"
990 P=SCORE/10
1000 IF P<.9 THEN 1030
1010 PRINT "EXCELLENT"
1020 GOTO 1120
1030 IF P<.8 THEN 1060
1040 PRINT "VERY GOOD"
1050 GOTO 1120
1060 IF P<.7 THEN 1090
1070 PRINT "ABOUT AVERAGE"
1080 GOTO 1200
1090 PRINT "PRETTY POOR!!! Y
OU BETTER   NOTIFY YOUR TEAC
HER"
1100 CALL SOUND(500,110,0)
1110 GOTO 1200
1120 TONE=110
1130 FOR COUNT=1 TO 10
1140 CALL SCREEN(16)
1150 CALL SCREEN(12)
1160 CALL SCREEN(1)
1170 CALL SOUND(+500,TONE,0)
1180 TONE=TONE+100
1190 NEXT COUNT
1200 PRINT
1210 PRINT "PLAY AGAIN(Y/N)"
1220 INPUT AG$
1230 IF AG$="Y" THEN 190
1240 END
```

```
100 REM ************
110 REM * PHANTOM5 *
120 REM ************
130 CALL CLEAR
140 DISPLAY AT(6,1):"THIS GA
ME USES THE JOYSTICKSTHE GOA
L OF THE GAME IS TO  DESTROY
 THE MAXIMUM TARGETS WHITOUT
 BEEN TOUCHED BY THE D.C.A"
150 DISPLAY AT(11,1):"YOU ON
LY HAVE 20 BOMBS AND  IF YOU
 TOUCH THE RED CROSS  IT WIL
L COST YOU POINTS."
160 DISPLAY AT(18,1):"
 GOOD LUCK...!"
170 DISPLAY AT(23,1):"TO STA
RT PRESS FIRE BUTTON"
180 CALL KEY(1,K,S)
190 IF K=18 THEN 210
200 IF S=0 THEN 180
210 CALL CLEAR
220 VERT=214
230 CALL CHAR(96,"0000104444
440000")
240 CALL CHAR(97,"1010387CFE
541028")
250 CALL CHAR(98,"2020205202
020500")
260 CALL CHAR(99,"0000202004
040000")
270 CALL CHAR(100,"084828125
C284404")
280 CALL CHAR(101,"000010381
C280000")
290 CALL CHAR(102,"4C3EFC7E3
FFC7C68")
300 CALL CHAR(103,"FF9999FFF
F9999FF")
310 CALL CHAR(104,"FFC3A5999
9A5C3FF")
320 CALL CHAR(105,"DB5A3C180
8080808")
330 CALL CHAR(106,"3C3C183C3
C3C3C3C")
340 CALL CHAR(107,"3C3C3C3C3
C183C3C")
350 CALL CHAR(108,"08183C7FF
E3C5890")
360 CALL CHAR(109,"00FE7C7F7
CFE0000")
370 CALL CHAR(110,"FFFFFFFFF
FFFFFF")
380 CALL CHAR(111,"FFFFFF000
6666060")
390 CALL CHAR(128,"FFFFFFFFF
FFFFFF")
400 CALL CHAR(115,"3C78F0F07
83C1E3C")
410 CALL CHAR(119,"3C3E1F3F7
E7F7E3C")
420 CALL CHAR(136,"7E7E7E7E7
E7E7E7E")
430 CALL CHAR(112,"181818181
8181818")
440 CALL CHAR(32,"FFFFFFFFFF
FFFFFF")
450 CALL CHAR(120,"44FE44FE4
4FE44FE")
460 CALL COLOR(13,15,4,14,5,
4,1,4,4,12,2,4)
470 SPEED=10 :: PERTE=0
480 CALL SCREEN(4)
490 CALL MAGNIFY(2)
500 CALL VCHAR(1,5,128,24)::
 CALL VCHAR(1,6,120,24):: CA
LL VCHAR(1,28,136,24):: CALL
 VCHAR(1,30,128,24)
510 VIV=0 :: VIH=0 :: POSV=1
28 :: COUNT=21 :: BO=20 :: P
P=0
520 CALL SPRITE(#8,103,7,200
,175,SPEED,0)
530 CALL SPRITE(#10,105,2,22
5,175,SPEED,0)
540 CALL SPRITE(#11,105,2,19
0,75,SPEED,0)
550 CALL SPRITE(#12,106,13,1
30,30,SPEED-10,0)
560 CALL SPRITE(#13,107,13,3
0,230,SPEED+10,0)
570 CALL SPRITE(#14,108,13,6
0,200,SPEED+5,-5)
580 CALL SPRITE(#15,109,13,1
45,200,SPEED,5)
590 CALL SPRITE(#16,110,12,1
75,200,SPEED,0)
600 CALL SPRITE(#17,110,12,1
35,200,SPEED,0)
610 CALL SPRITE(#18,110,12,1
95,15,SPEED,0)
620 CALL SPRITE(#19,110,12,1
,48,SPEED,0)
630 CALL SPRITE(#20,111,14,4
8,48,SPEED,0)
640 CALL SPRITE(#21,111,14,1
08,248,SPEED,0)
650 CALL SPRITE(#23,111,11,1
50,128,SPEED,0)
660 CALL SPRITE(#25,112,5,14
8,36,SPEED+5,0)
670 CALL SPRITE(#26,112,5,17
6,36,SPEED+5,0)
680 CALL SPRITE(#27,112,5,14
3,36,SPEED+5,0)
690 GOSUB 880
700 CALL JOYST(1,X,Y)
710 CALL KEY(1,K,S):: IF K=1
8 THEN 1030
720 IF X=-4 THEN CALL MOTION
(#2,0,-20)
730 IF X=4 THEN CALL MOTION(
#2,0,20)
740 IF X=0 THEN CALL MOTION(
#2,0,0)
750 DISPLAY AT(1,1):"LOSSES"
;PERTE;"BOMBS ";BO;"POINTS";
PP
760 GOSUB 1440
770 LET COUNT=COUNT+1
780 IF COUNT<20 THEN 840 ELS
E 790
790 CALL POSITION(#9,AAA,BBB
):: IF AAA>195 THEN LET HQ=0
800 IF HQ=1 THEN 840
810 COUNT=0
820 IF X=0 THEN CALL MOTION(
#2,0,0)
830 RANDOMIZE :: CALL SPRITE
(#9,104,15,1,INT(RND*140)+57
,SPEED,0):: LET HQ=1
840 GOSUB 910
850 GOSUB 990
860 GOSUB 990
870 GOTO 700
880 REM
890 CALL SPRITE(#2,97,16,84,
POSV,VIH,VIV)
900 RETURN
910 CALL POSITION(#21,XXX,YY
Y):: IF XXX>200 THEN CALL DE
LSPRITE(#21):: CALL DELSPRIT
E(#22):: LET TR=0
920 IF TR=1 THEN RETURN
930 RANDOMIZE :: AAA=INT(RND
*3)+1
940 ON AAA GOTO 950,970,960
950 CALL SPRITE(#21,115,5,1,
VERT,SPEED,0):: LET TR=1 ::
RETURN
960 CALL SPRITE(#21,115,5,1,
VERT,SPEED,0):: CALL SPRITE(
#22,119,5,250,VERT,SPEED,0):
: TR=1 :: RETURN
970 CALL SPRITE(#21,119,5,1,
VERT,SPEED,0):: LET TR=1 ::
RETURN
980 RETURN
990 IF X=4 THEN CALL MOTION(
#2,0,20)
1000 IF X=-4 THEN CALL MOTIO
N(#2,0,-20)
1010 IF X=0 THEN CALL MOTION
(#2,0,0)
1020 RETURN
1030 IF BOMB=1 THEN 1130
1040 BOMB=1
1050 CALL POSITION(#2,AA,POS
V)
1060 CALL SPRITE(#3,98,2,84,
POSV,SPEED,0):: FOR I=1 TO 5
0 :: NEXT I
1070 CALL JOYST(1,X,Y)
1080 IF X=0 THEN CALL MOTION
(#2,0,0)
1090 IF X=-4 THEN CALL MOTIO
N(#2,0,-20)
1100 IF X=4 THEN CALL MOTION
(#2,0,20)
1110 CALL PATTERN(#3,99):: F
OR I=1 TO 50 :: NEXT I
1120 GOSUB 1140
1130 GOTO 700
1140 REM
1150 BOMB=0
1160 CALL COLOR(#3,10):: CAL
L PATTERN(#3,100):: CALL SOU
ND(500,-7,1)
1170 CALL COINC(#3,#8,10,HIT
):: IF HIT=-1 THEN PP=PP-5
1180 CALL COINC(#3,#9,10,HIT
):: IF HIT=-1 THEN PP=PP+1
1190 CALL COINC(#3,#10,10,HI
T):: IF HIT=-1 THEN PP=PP+2
1200 CALL COINC(#3,#11,10,HI
T):: IF HIT=-1 THEN PP=PP+2
1210 CALL COINC(#3,#12,10,HI
T):: IF HIT=-1 THEN PP=PP+10
1220 CALL COINC(#3,#13,10,HI
T):: IF HIT=-1 THEN PP=PP+10
1230 CALL COINC(#3,#14,10,HI
T):: IF HIT=-1 THEN PP=PP+10
1240 CALL COINC(#3,#15,10,HI
T):: IF HIT=-1 THEN PP=PP+10
1250 CALL COINC(#3,#16,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1260 CALL COINC(#3,#17,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1270 CALL COINC(#3,#18,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1280 CALL COINC(#3,#19,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1290 CALL COINC(#3,#19,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1300 CALL COINC(#3,#20,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1310 CALL COINC(#3,#21,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1320 CALL COINC(#3,#23,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1330 CALL COINC(#3,#24,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1340 CALL COINC(#3,#25,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1350 CALL COINC(#3,#26,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1360 CALL COINC(#3,#27,10,HI
T):: IF HIT=-1 THEN PP=PP+1
1370 CALL JOYST(1,X,Y)
1380 IF X=-4 THEN CALL MOTIO
N(#2,0,-20)
1390 IF X=4 THEN CALL MOTION
(#2,0,20)
1400 IF X=0 THEN CALL MOTION
(#2,0,0)
```

```
1410 CALL DELSPRITE(#3):: CA
LL DELSPRITE(#5):: DCA=0
1420 GOSUB 1640
1430 RETURN
1440 IF DCA=1 THEN RETURN
1450 LET DCA=1
1460 RANDOMIZE
1470 CALL SOUND(10,-7,5):: C
ALL SOUND(100,40000,30):: CA
LL SOUND(100,-7,1)
1480 CALL POSITION(#2,DS,PL)
1490 CALL SPRITE(#5,101,2,54
,PL,SPEED,0)
1500 CALL JOYST(1,X,Y)
1510 CALL KEY(1,K,S):: IF K=
18 THEN 1030
1520 IF X=-4 THEN CALL MOTIO
N(#2,0,-20)
1530 IF X=4 THEN CALL MOTION
(#2,0,20)
1540 IF X=0 THEN CALL MOTION
(#2,0,0)
1550 CALL PATTERN(#5,102)::
FOR I=1 TO 8 :: CALL COINC(#
2,#5,10,KZ):: IF KZ=-1 THEN
LET PERTE=PERTE+1 :: CALL DE
LSPRITE(#5):: DCA=0 :: RETUR
N
1560 CALL JOYST(1,X,Y)
1570 CALL KEY(1,K,S):: IF K=
18 THEN 1030
1580 IF X=-4 THEN CALL MOTIO
N(#2,0,-20)
1590 IF X=4 THEN CALL MOTION
(#2,0,20)
1600 IF X=0 THEN CALL MOTION
(#2,0,0)
1610 NEXT I :: CALL DELSPRIT
E(#5)
1620 LET DCA=0
1630 RETURN
1640 REM
1650 LET BO=BO-1 :: IF BO<1
THEN 1670
1660 RETURN
1670 CALL DELSPRITE(ALL):: C
ALL CLEAR
1680 DISPLAY AT(16,1):"YOU H
AVE LOST ";PERTE;"BOMBERS."
1690 DISPLAY AT(18,1):"YOU H
AVE DESTROYED";PP;"TARGETS."
1700 PERTE=PERTE+1
1710 DISPLAY AT(20,1):"YOUR
AVERAGE IS ";INT(PP/PERTE)
1720 DISPLAY AT(24,1):"AGAIN
 PRESS FIRE BUTTON"
1730 CALL KEY(1,AAA,BBB)
1740 IF AAA=18 THEN 1760
1750 IF BBB=0 THEN 1730
1760 GOTO 32767
1770 END
```

```
100 FOR Z=1 TO 12
110 CALL COLOR(Z,16,1)
120 NEXT Z
130 CALL SCREEN(5)
140 CALL CLEAR
150 DISPLAY AT(5,5):"SEA DIV
ER"
160 !
170 FOR T=1 TO 300
180 NEXT T
190 PRINT "  YOU ARE A DEEP
SEA DIVER
            DIVING FOR GOLD
.
            YOU HAVE GOT TO
"
200 PRINT
210 PRINT "GET TO THE BOTTOM
 OF THE SEA"
220 PRINT
230 PRINT "AND BACK TO THE S
URFACE.
            DODGING THE SHARK
S,CRABS,"
240 PRINT "JELLY FISHES AND
THE SHIP.  USING THE JOYSTIC
K."
250 PRINT
260 PRINT " YOU HAVE TO GET
20 BAGS OF
            GOLD TO WIN."
270 PRINT "PRESS FIRE TO DIV
E DOWN.
            PRESS 'S' TO STAR
T."
280 CALL KEY(1,S,D)
290 IF S=18 THEN 300 ELSE 28
0
300 CALL CHAR(60,"081C2A0814
14"&RPT$("00",24))
310 CALL MAGNIFY(3)
320 RANDOMIZE
330 CALL CLEAR
340 CALL SCREEN(5)
350 CALL COLOR(1,5,5)
360 CALL CHAR(32,"00C0201008
040300")
370 CALL CHAR(33,"0003040810
20C000")
380 FOR T=1 TO 32 STEP 2
390 CALL VCHAR(1,T,33,24)
400 NEXT T
410 CALL COLOR(1,6,5)
420 CALL CHAR(34,"")
430 CALL HCHAR(1,1,34,64)
440 CALL CHAR(40,"0000000014
1830509419305010FF7F3F000000
0000405060C244485060FFFCF0")
450 CALL SPRITE(#2,40,15,8,1
25,0,10)
460 CALL CHAR(44,"0000282810
543810"&RPT$("00",24))
470 CALL SPRITE(#1,44,16,18,
140)
480 CALL CHAR(48,"0000000000
C36F3F67C3000000000000000000
C060F8DFF8F7FC")
490 FOR T=5 TO 8
500 CALL SPRITE(#T,48,10,T*1
2,INT(240*RND)+10,0,INT(20*R
ND)+10)
510 NEXT T
520 CALL CHAR(52,"0000000000
000000000000D63276A1102000000
00000000000000F0F8FC482448")
530 FOR T=9 TO 11
540 CALL SPRITE(#T,52,13,174
,INT(250*RND)+1,0,-10):: CAL
L COINC(ALL,D):: IF D=-1 THE
N CALL DELSPRITE(#T):: GOTO
540
550 NEXT T
560 CALL CHAR(56,"0000000307
030519224C519226281324000000
E0F8FCFEFF1F37CA30C08")
570 CALL SPRITE(#13,56,8,45,
INT(240*RND)+10,0,INT(10*RND
)+10)
580 CALL SPRITE(#12,56,8,30,
INT(240*RND)+10,0,INT(10*RND
)+10)
590 CALL SPRITE(#14,56,8,120
,INT(240*RND)+10,0,INT(10*RN
D)+10)
600 CALL SPRITE(#15,56,8,145
,INT(240*RND)+10,0,INT(10*RN
D)+10)
610 CALL COLOR(9,12,12)
620 CALL HCHAR(23,1,99,64)
630 CALL KEY(1,K,S):: IF S=0
 THEN 630
640 CALL JOYST(1,K,S)
650 CALL MOTION(#1,10,K*2)
660 CALL POSITION(#1,R,C)::
IF R>190 THEN 880
670 CALL COINC(ALL,GT)
680 IF GT=0 THEN 640
690 CALL SOUND(1000,-5,0)
700 LI=LI+1 :: IF LI=5 THEN
710 ELSE 640
710 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL CHARSET ::
CALL SCREEN(5)
720 FOR Z=1 TO 12 :: CALL CO
LOR(Z,16,1):: NEXT Z
730 PRINT "   YOU NOW LIVE W
ITH"
740 PRINT
750 PRINT " THE LITTLE FISHE
S "
760 PRINT
770 PRINT "AT THE BOTTOM OF
THE SEA"
780 PRINT
790 PRINT "YOU HAVE ":POI:"B
AGS OF GOLD"
800 FOR T=1 TO 5 :: PRINT ::
 NEXT T
810 FOR BMX=1 TO 24
820 READ B,A
830 IF B=0 THEN 860
840 CALL SOUND(A,B,0)
850 NEXT BMX
860 DATA 294,400,294,300,294
,240,294,300,349,400,330,300
,330,350,294,350,294,300,277
,350,294,500,0,0
870 FOR T=1 TO 900 :: NEXT T
 :: RUN
880 CALL MOTION(#1,0,0):: FO
R T=1 TO 10
890 CALL SOUND(-100,T*110,0)
900 NEXT T
910 CALL PATTERN(#1,60)
920 CALL JOYST(1,K,S)
930 CALL MOTION(#1,-10,K)
940 CALL POSITION(#1,R,C)::
IF R<18 THEN 970
950 CALL COINC(ALL,DF):: IF
DF=-1 THEN 690
960 GOTO 920
970 CALL SOUND(-100,-7,0)
980 POI=POI+1
990 IF POI=20 THEN 1020 ELSE
 1000
1000 CALL MOTION(#2,0,10+POI
)
1010 CALL PATTERN(#1,44):: G
OTO 640
1020 CALL CLEAR
1030 CALL DELSPRITE(ALL):: C
ALL CHARSET
1040 CALL CLEAR
1050 CALL SCREEN(16)
1060 FOR Z=1 TO 12
1070 CALL COLOR(Z,5,1)
1080 NEXT Z
1090 PRINT "     **********
**
                 * WELL DONE
 *               *
 *"
1100 PRINT "     **********
**"
1110 PRINT
1120 PRINT
1130 PRINT
1140 PRINT " YOU HAVE GOT AW
AY WITH 20"
1150 PRINT
1160 PRINT "     BAGS OF GOL
D"
1170 PRINT
1180 PRINT
1190 PRINT
1200 PRINT "   PRESS 'S' TO
START"
1210 CALL KEY(1,A,S)
1220 IF A=2 THEN RUN ELSE 12
10
1230 END
```

```
**************************
* THE 4A EPROM EXPERIMENT *
**************************
```

by Arto Heino

The Drummer of the band THINLINE expressed his concern to me about the cost of DIGITAL SAMPLES that are available on EPROMS for his SIMMONS drum synthesizer. Well, I said lets have a look at whats involved in making a different sample from the info I can find inside the original eprom.

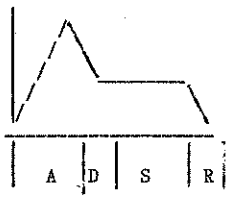
I asked JOHN PAINE if he could read the original EPROM with his MECHATRONICS EPROM PROGRAMMER and send the file up the MODEM so I could investigate.

The file was a 32 sector Program image file with no header, mmm I thought, I'll read it with a CALL LINK that reads the sectors directly and dumps each sector to VDP so I can manipulate the DATA in BASIC.

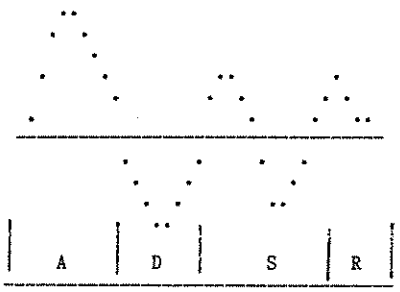
The next step was to dump the data in a suitable structure so my eyes could understand the total form. A small printer routine was in order.

To my suprise I hit it on the button with how the contents of the EPROM was structured. The original sample was a TOM-TOM sound and the ADSR was obvious. (Attack,Decay,Sustain,Release)

Here is a diagramtic sample of what a sound is composed of:



Here's what it looks like with the sound wave added:



This would be a very short and very low frequency sound.

I proceeded from this into creating a small program that would duplicate the sound formula. The first step was to program a SINE wave from basic plotting the wave on the screen and adjusting until it matched the wave shape of the EPROM. From there I wrote this program, not a the best example of progamming but it worked. Mind you this routine took about an HOUR to run.

```
100 REM ***************
110 REM SINESOUND MM+32K
120 REM SOUND PROGRAMMER
130 REM ***************
140 REM by arto heino 1987
150 CALL CLEAR
160 INPUT "RAM SIZE  1-16384          16384:":RAM
170 INPUT "START NUM 0-255            127  :":START
180 INPUT "ATTACK  LENGTH(%)          1    :":ATTK
190 INPUT "DECAY   LENGTH(%)          30   :":DCAY
200 INPUT "SUSTAIN LENGTH(%)          40   :":SUS
210 INPUT "RELEASE LENGTH(%)          29   :":REL
220 INPUT "WAVE FREQUENCY             4000 :":FREQ
230 FREQ=(100/FREQ)*256
240 INPUT "MAX AMPLITUDE              255  :":AMP
250 INPUT "FREQUENCY SHIFT(%)         10   :":SHFT
260 A=(ATTK/100)*RAM
270 RISE=AMP/A
280 GAIN=AMP/A
290 D=(DCAY/100)*RAM
300 LOSE=AMP
310 S=(SUS/100)*RAM
320 R=(REL/100)*RAM
330 FALL=AMP/(D+R)
340 H=(SHFT/100)*RAM
350 SH=FREQ/H
360 FOR X=1 TO A
370 Y=SIN((X-1)/FREQ)*GAIN+START
380 GAIN=GAIN+RISE
390 FREQ=FREQ+SH
400 GOSUB 800
410 NEXT X
420 FOR X=A+1 TO A+D
430 Y=SIN((X-1)/FREQ)*LOSE+START
440 LOSE=LOSE-FALL
450 FREQ=FREQ+SH
460 GOSUB 800
470 NEXT X
480 FOR X=A+D+1 TO A+D+S
490 Y=SIN((X-1)/FREQ)*LOSE+START
500 FREQ=FREQ+SH
510 GOSUB 800
520 NEXT X
530 FOR X=A+D+S+1 TO A+D+S+R
540 Y=SIN((X-1)/FREQ)*LOSE+START
550 LOSE=LOSE-FALL
560 FREQ=FREQ+SH
570 GOSUB 800
580 NEXT X
590 INPUT "PRINT OUT Y/N ? ":YN$
600 IF YN$="N" THEN 150
610 PRINT "DEVICE NAME ?":"RS232.BA=4800.DA=8.CR"
620 INPUT DV$
630 OPEN #1:DV$
640 PRINT " ":"       WORKING....":""
650 C$=CHR$(27)&"K"
660 CR$=CHR$(10)&CHR$(13)
670 PRINT #1:CHR$(27);"A";CHR$(1);CHR$(27);CHR$(50);
680 FOR LP=-24576 TO -24577+RAM
690 CALL PEEK(LP,CH)
700 AM=CH-1-(CH<1)
710 PRINT #1:C$;CHR$(CH-(AM=0));CHR$(0);
720 FOR ZZ=1 TO AM
730 PRINT #1:CHR$(0);
740 NEXT ZZ
750 PRINT #1:CHR$(1);CR$
760 NEXT LP
770 CLOSE #1
780 CALL KEY(0,KY,ST)
790 IF ST<1 THEN 780 ELSE 150
800 IF X>16384 THEN 830
810 Y=Y+(Y>255)*(Y-255)-(Y<0)*(-Y)
820 CALL LOAD(-24577+X,Y)
830 RETURN
```

Once the program finished running I run MINI-MATE and saved the memory area (>A000->BFFF) that contained all the new DATA.

Now going to a Sector editor I took out the header and shorten the file by 6 bytes and VOILA !!!!! It was ready to be BURNT in. Sending the file to JOHN , he Burnt the EPROM. I delivered the EPROM to the Drummer and he PLUGGED it in and to his amazement it WORKED!!!

This was an exersize for the trusty 99/4A and it got an A+ for being spot on with its MATHS, HARDWARE, SOFTWARE and VERSATILITY as a HOME COMPUTER.

DISK CONTROLLERS - HOW THEY WORK

The following article written by Jerry Coffey appeared in the NorthWest Ohio 99er News. While it deals largely with the Myarc Disc Controller Card, many of you will find it gives a concise and clear explanation of controllers in general.

The disk capacity of the TI99 has increased in just a few years from less than 80K (a single sided 35 track drive) to almost 2.9 megabytes (four double sided double density 80 track drives). The early standalone was replaced by the PEBox system which would support three double sided 40 track drives (540K). Corcomp introduced their 4 drive double density system (1440K) followed by Myarc's similar system with two double density formats (1280K and 1440K). Then in 1986, Myarc offered its 80 track upgrade which doubled capacity again. Even as capacity was increasing rapidly, the TI and Corcomp controllers differed only modestly in I/O speed. When Myarc introduced its fast DSDD controller, few reviewers did justice to its speed advantage. Early comparisons were done at the standard TI or Corcomp interlace, but the big speed gains required taking advantage of the much tighter sector interlace possible with the high-speed Myarc card. To understand how this works we need to take a look at the way a disk drive performs.

Disk Drive Fundamentals

A floppy disk drive writes information in concentric rings called "tracks" on a thin plastic disk coated with a film of magnetic particles. Each track in turn is divided into blocks of information called sectors. A blank disk has one (or more) index holes used to synchronize the process of writing to and reading from the disk. The type with many holes are called "hard sectored" since each sector has its position fixed by an index hole. The type of disks used by most computers have only one hole and are called "soft sectored". In this system, the computer must write magnetic signposts on the disk to mark out each sector in a process called "formatting" or "initializing" a disk. These signposts take up a substantial part of the space on a track since they include not only sector numbers but buffers (filler bytes) that allow the computer synchronization to read or write sectors of data and to prevent a sector identifier from being overwritten by a drive operating at a slightly different speed from the drive that formatted the disk.

The typical 5.25 inch disk drive has a "stepper motor" capable of moving the drive's read/write head(s) in or out along a radius of the disk in steps of 1/48 of an inch (thus the terminology "48 tpi" = 48 tracks per inch). Since the inner tracks have a smaller circumference, they crowd the bits of information together. Magnetic coatings on a floppy disk are rated by their capacity in bits per inch at standard magnetic flux for the write head. This figure is usually over 5000 bpi for modern floppies, but was somewhat lower a few years ago. The circumference of the inner track of a 40 or 80 track disk is about 10 inches — which allows about 6250 bytes to be written on the track without exceeding 5000 bpi. For comparison, the Corcomp double density format requires over 6400 bytes per track. Media limitations were the reason that some early drives only used the outer 35 tracks. The 16 sector (by 256 bytes/sector) format recommended by most drive makers requires only 6250 bytes per track and includes several hundred additional "buffer" bytes to compensate for differences in drive timing.

Timing is Everything

With soft-sectored disks, the integrity of the read/write process requires critical timing. The disk rotates at 300 rpm within a small margin. This means there are about 250 thousand magnetic pulses (bits) passing beneath the head each second. In single density format, the majority of these pulses are timing or filler bits -- in double density, many of the timing bits are suppressed in order to double the rate of data bits. In a typical sector read the drive must bring the disk up to speed, recognize the index hole, step out to track zero (to get its bearings), determine single or double density, verify its psition, step in to the target track, verify the track number (written in the format operation), detect the sector identifier as it flies past, then immediately read the 256 data bytes into memory. Five of these operations require accurate reading of the magnetic pulses whizzing by at over 250K bits per second.

If you do some quick arithmetic (256 bytes/sector = 2048 bits/sector into 250K bits/second) ... hmmm ... Why can't the drive read a 125 sector file in 1 second? Well, firstly, many of the bits are not data bits, they are overhead to keep things synchronized and allow for timing variation between drives. Secondly, some time is used moving the head from one track to the next when more than one track must be read. Thirdly, 250K is the instantaneous read rate and the computer must take time to do other things like move the last sector out of its buffer to make room for the next one. In the standard TI protocol for reading a disk, the data is moved into VDP ram (so the drive could be used without the memory expansion) before it goes to the expansion memory. All this thrashing eats great chunks of the time available for reading data. By the time one sector is safely tucked away in the 32K card, several sectors have already passed by the drive's read head. If the sectors were written consecutively on the disk, we would have to wait a full revolution (0.2 seconds) before the next sector would pass under the head. To avoid this inefficiency, the consecutively numbered sectors are spaced out around the disk so that they are separated by just enough time to take care of other business. The actual pattern in which the sectors are scattered is called the "interlace". The idea of the interlace is to spread the sectors out to match the timing needs of the hardware -- both the time needed to stash each sector and the time needed to step from one track to the next and get the head settled down for some serious (250K bps) reading.

Interlace and Head Step Times

Life was simple with the TI controller. Both the interlace and the head step time were locked into the controller's PROM (that's the programmable chip that contains the control programs for the card). The head step time is the built-in delay between step signals to allow the stepper motor to move the head one "click" in or out. The TI settings are very conservative (read "slow") to allow for slow drives. The step time is 20ms -- if you step from track zero to track 39, it takes 20X39=780ms, almost four revolutions of the drive. The TI interlace lays the sectors down on a track in the order 075318642. This allows all sectors to be read in four revolutions of the disk though the slow head step lets another revolution go by between tracks. Thus the maximum read rate is about 9 secotrs per 5 revolutions (=one second) or 2304 bytes per second.

When Corcomp designed its double density controller, allowances were made for the increased speed of later drives by permitting the step rate to be set with DIP switches for each drive. The step rates available are 30, 20, 12, and 6ms (the faster values quoted in the CC manual are referenced to the wring clock speed). They also provided a choice of interlace options, though only a couple of them are practical. The default interlaces are labeled "7" for single density and "10" for double density. The single density interlace is the same as TI's, but with a faster step setting the head can be moved without losing a revolution and thus reads 20% faster than the TI controller. The double density interlace allows 18 sectors to be read in 5 revolutions but it doesn't leave enough margin to stash the last sector and step the head in time to catch the zero sector of the next track (that's why the sector number "hangs" for 0.2 seconds each 18 sectors while verifying a formatted disk -- you are seeing an extra revolution needed to acquire the first sector of the next track). Thus the maximum read rate is 18/1.2 or 15 sectors per second, about 67% faster than the TI controller. Users

of the CC controller have probably noticed that it loads its own MANAGER program faster than this. In this case a special loader bypasses VDP and loads directly to CPU RAM -- this faster handling of the data allows the stepper motor to be activated sooner and saves one revolution per track (so the 98 sector file can be read in about 5.5 seconds). This provided a foretaste of the speed that Myarc would achieve with its double density controller.

The Myarc Controller bypasses VDP RAM to load directly to CPU RAM. This technique, coupled wiwth a buffer RAM chip on the controller card provided a quantum jump in disk I/O speed. The Myarc card reads the TI single density interlace at 11.25 sectors/second (the same as Corcomp) and reads the CC 18 sector/track interlace at 18 sectors/second (the same speed Corcomp reads its MANAGER program), but this is only the beginning. Since the hardware empties its sector buffer faster, consecutive sectors can be placed closer together allowing a track to be read in fewer revolutions, i.e., it supports a faster interlace. With fast drives, the 9 sectors/track single density format can be read at interlace "2". (NOTE: In Myarc terminology, the interlace number represents the number of disk revolutions required to read a track.) This works out to 22.5 sectors/second compared to 9 for the TI and 11.25 for the CC controller. The Myarc 16 sector format can be read at interlace "3", 26.67 sectors/second -- 3 times as fast at the TI controller and almost twice as fast as Corcomp double density. The Corcomp 18 sector format can be read at interlace "3" or "4" but the data rate is the same in either case, 22.5 sectors/second. Interlace "4" is smooth but requires a very quick head step; interlace "3" reads the track in 3 revolutions but forces an extra revolution for the step from track to track because sectors 17 and 0 are adjacent on the disk. Though both interlaces have the same data rate, interlace "3" is safer if you are uncertain about the speed of your stepper motor.

In order to read and write both double density formats, the Myarc system must insert an additional step in some I/O operations -- sector zero must be read to determine whether a double density disk has 16 or 18 sectors per track. This datum is needed to convert the logical sector numbers used by the TI operating system into track and sector-within-track addresses for the floppy disk controller chip. The TI and Corcomp controllers do not need this step because they do not use the full potential of the TI disk I/O protocol. Once this step, accessing sector sero, is added to the various disk operations, it opens the system up for using more than two formats -- including 80 track formats.

Beyond Double Density

A two format system can be managed usisng only the floppy disk controller's inherent ability to sense single and double density recording patterns. To get beyond this limitation, the additional data stored in sector zero must be read, stored, and used to modify the special binary commands sent to the FDC (floppy disk controller) chip. Fortunately the TI99/4A system design already provides for such innovations through the Device Service Routine concept and standard "GPL" calls. The system doesn't care what hardware is attached as long as it plays by the rules -- an interface program stored in a memory chip (PROM) on the peripheral device does the trick. This program handles calls for I/O operations from other programs such as TI Writer or the Basic INterpreters. Another set of rules controls the way disk and file information are saved on a disk. Disk parameters are stored in sector 0, while sector 1 must have a two byte "pointer" (a hexadecimal sector address) for each block (one sector) containing the bookkeeping data for a file. It is these blocks that are scanned in order to display the disk directory.

Since the Myarc controller must read sector zero to determine the number of sectors per track, the other parameters in that sector are available to control other variables such as number of tracks. But there wer other limitations to overcome. The number of files on a disk is limited by the space available for pointers. 256 bytes at 2 bytes per pointer would give 128 files -- except the pointer list must end with a null word (>0000) so directory routines know where to stop -- so we get 127 files per disk. The pointer itself can address sector numbers as high as 65535, so this is no problem. The real limitation is the sector 0 bit map. It begins at byte 56 leaving only 200 bytes or 1600 bits available to map the disk. Since a bit must be turned on for each sector used, the 1440 sector DSDD 40 track disk is already near the limit. The answer devised for the 80 track DSDD system is to map two consecutive sectors with each bit. It wastes some space but no more than systems that use the standard 512 byte sector.

Making the Quad System Work

So now let's say we have new code in the disk controller EPROM (an "erasable" version of the PROM chip TI use) that does all the proper tricks with the bit map and has the FDC commands to control the new 80 track drives we have added to the system. We still have to tell the controller which drives are 80 track and find a disk manager program that can use the new commands. The selection problem can be taken care of using the DIP switches on the card (but in the process you lose their original function - setting step speed). Since the EPROM responds to standard GPL calls,most functions can be handled by the TI Disk Manager 2 cartridge. The exception is the disk formatting process - formatting works OK, but the initial data written into sector zero is for the standard bit map. (This can be fixed by changing byte 56 from >03 to >01 with a sector editor.) Read/Write operations from XB or TI Writer work fine since they use the GPL protocols. Myarc has an excellent disk manager program that works beautifully with 40 track drives, but it has suffered from a number of subtle bugs in 80 track mode. This program, like many others designed for high speed I/O, uses assembly language code to handle the FDC - bypassing some of the routines in the EPROM. Differences in bit map handling, even slight differences in execution times can affect the performance of 80 track drives. The code in the 80 track EPROM has had a lot of attention to proper timing - the price you pay for higher performance.

Fine Tuning the Myarc Disk System

Before you start using the Myarc system routinely, there are some experiments that can get maximum performance from your drives. Use the Myarc disk manager to try different interlace settings - first with your 40 track drives, then with the 80 track drives. Watch for hesitations as each formatted disk is verified, then use the Test option to read the sectors you have layed down. Look and listen for "retries" - when the sector number pauses with a head seek noise. Use the best disks you have and note the combinations that test smoothly. With fast drives in good condition, you should be able to run 9 sector (single density) format at interlace 2 and 16 or 18 sector double density format at interlace 3. Don't worry if 18/3 pauses at the end of each track - this is just the extra revolution forced by having sectors 17 and 0 adjacent on the disk.

When you try this with 80 track dirves, don't be surprised if the results are different. The time needed for the head to settle into a wide standard track may not be adequate to get it reading properly from the narrow tracks on the quad drive. Such subtleties as erase delays and disk quality are also more critical on the skinny, low power tracks. My Mitsubishi 4853s (96 tpi) will support both 16/3 and 18/3 but are unreliable at 18/4, while my TEAC 55Bs support all three at 48 tpi. Don't take chances with any setup that is marginal. The error rate may be low, but it always seems to happen to a file that isn't backed up.

For those of you who wish to extend the practical limits of Myarc systems, I suggest you read the full article in the April issue of the NorthWest Ohio 99er News. There is information specific to the Myarc 80 track card and in particular discusses the topic of "hotrodding", "Interlace patterns", FDC controller chip replacement and other card modification.

AT-99 DISK CONTROL SYSTEM (revision)

by John Paine
and Peter Schubert
7th June 1987

The AT-99 Disk control board is a very advanced and powerful package of software and hardware integrated with the MINI-PE SYSTEM, a small stand-alone expansion system for the basic TI-99 Console, which uses the latest advances in technology to provide a low cost means of upgrading for those with just a basic console or cassette based system.

The Disk Control Board consists of the following components:

99 Disk Control Board which can control up to 4 disk drives in single or double density format. Maximum storage capacity per diskette is 1440 Sectors or 368000 characters of information.

Software in ROM (read-only memory) which enhances the TI-99 Console Operating System (console Basic). This is available by the use of CALL statements in Basic or XB.

Diskette of utilities and Manager software.

Because of the software enhancements in the Disk Controller ROM, you do not need a Command Module to perform Disk functions or to load most software from disk.

POWER REQUIREMENTS

a) None. (ie. Although the Disk Control Board uses approx. 0.3Amp average current, this can be supplied from the console +5 volt supply if the console is in good order, and the original TI power transformer is used) Note; Heat generated within the console is largely due to the +12 volt supply and this is not used by the Mini-PE, so there should not be any noticeable increase in heat build-up within console.

b) External power. Optional 9 volt DC plugpak can be provided to power Disk Control Board.

NEW CALLS AVAILABLE

1) CALL ILR

Loads the standard E/A utilities into low memory. This will clear out Low Memory Expansion and sets it up for loading DIS/FIX 80 Assembly files. (This is done automatically by CALL LR).

Example:

Basic or Xbasic Command Mode:
```
CALL ILR
```
Running Xbasic Program:
```
CALL LINK("ILR")
```

2) CALL LR("device.filename")

Loads a DIS/FIX 80, autostart or non-autostart Assembly Program. (This is exactly the same as option 3, load and run on E/A menu.). Includes automatic loading of the E/A utilities (CALL ILR) if they have not already been loaded or if an error occurred during previous load.

Examples:

Basic or Xbasic Command Mode:
```
CALL LR("DSK1.FRED")
```
Running Xbasic Program:
```
CALL LINK("LR")(DSK1.FRED)
or
CALL LINK("LR")(A$) where A$="DSK1.FRED"
```

3) CALL LLR("startname")

This starts a non-auto start program. This is the same as option 4-RUN on E/A menu.

Examples:

Basic or Xbasic Command Mode:
```
CALL LLR("START")
```
Running Xbasic Program:
```
CALL LINK("LLR")("START")
or
CALL LINK("LLR")("A$") where A$="START"
```

4) CALL RUN("Device.Filename")

This loads Assembly PROGRAM IMAGE files like option 5 RUN PROGRAM of E/A menu. This call also sets up E/A environment in VDP Memory.

Examples: Basic or Xbasic Command Mode:
```
CALL RUN("DSK1.PROGRAM")
```
Running Xbasic Program:
```
CALL LINK("RUN")("DSK1.PROGRAM")
or
CALL LINK("RUN")("A$") where A$="DSK1.PROGRAM"
```

PROGRAM must be a Program Image Assembly File that can be loaded and executed from E/A option 5 RUN PROGRAM FILE.

5) CALL RUN

This CALL without brackets or a filename automatically loads DSK1.UTIL1

Examples:

Basic or Xbasic Command Mode:
```
CALL RUN
```
Running Xbasic Program:
```
CALL LINK("RUN")
```

6) DELETE "XILR"
or
OPEN #1:"XILR"

Sets up the E/A utilities into low memory from a running XB Program. It also sets up the Link names for all the above calls and utilities so that they may be acessed from a running program.

NOTE: This last CALL must be executed before the above CALL LINK examples can be executed from a running XB program!!!!

7) CALL WRTRG

This utility loads the VDP Write Only Registers. The eight VDP registeres are numbered 0 through 7 and are used to control the following features:

Register 0 - Used with register 1 to select the Graphics mode, also controls whether an external Video picture can be shown.
Register 1 - As above, also screen blank control, sprite size and interrupt enable/disable.
Register 2 - Defines location of screen image table.
Register 3 - Defines location of colour description table.
Register 4 - Defines location of Pattern (character) table.
Register 5 - Defines location of sprite characteristics table.
Register 6 - Defines location of sprite pattern table.
Register 7 - Sets the Foreground and Background colour.

```
BASIC Syntax:
CALL WRTRG(register num,value[;register#,value;....])

EXTENDED BASIC Syntax:
CALL LINK("WRTRG")(register num,value[;register#,
value;.....])
Examples: BASIC:
               100 CALL WRTRG(7,1)
               Writes 1 into VDP register 7 and
               changes the screen colour to black.
          EXTENDED BASIC:
               100 CALL INIT::DELETE "XILR"
               110 CALL LINK("WRTRG")(7,1)
```

8) CALL MOVEM

This utility allows you to move blocks of memory from one location to another.
BASIC Syntax:

```
CALL MOVEM(type,from,to,num of bytes to move[;...])

XB Syntax:
CALL LINK("MOVEM")(type,from,to,num[;.....])
     Type of move:  1= Move from VDP to VDP
                    2= Move from VDP to CPU
                    3= Move from CPU RAM or ROM to VDP
                    4= Move CPU RAM or ROM to CPU RAM
Examples: BASIC;

CALL MOVEM(1,0,704,64)
     Copies lines 1 and 2 (64 bytes) on the screen to
     lines 23 and 24 on the screen.
CALL MOVEM(2,0,40960,768)
     Copies the entire screen(768 bytes) to address
     40960(Hex >A000) in expansion memory to the screen
CALL MOVEM(3,40960,0,768)
     Copies 768 bytes from address >A000 to the screen
CALL MOVEM(4,0,40960,8192)
     Copies entire Console ROM, 8K, to Expansion Memory
     address >A000

EXTENDED BASIC;

100 CALL INIT::DELETE "XILR"
110 CALL LINK("MOVEM")(3,0,0,128)
     Initializes memory expansion and sets up E/A
     utilities into low memory expansion, then in line
     110 copies first 128 bytes of console ROM to top
     of the screen.
```

9) CALL EXEC

This utility will execute an assembly language routine. It may be in RAM or ROM. With this you can execute your own assembly routines in expansion memory by calling its starting address. The assembly routine must end with a RETURN or you get a lock-up condition.

```
BASIC Syntax:
CALL EXEC(addr[;addr;addr;....])

XB Syntax:
CALL LINK("EXEC")(addr[;addr;addr;...])

Examples: BASIC;
100 CALL MPEEK(3,0,A)
110 CALL EXEC(A)
     Reads the start address of the power-up routine.
     Executes a warm power-up and displays the TI title
     screen. Turn console off and then on again to
     display the AT title screen.
EXTENDED BASIC;
100 CALL INIT::DELETE "XILR"
110 CALL LINK("MPEEK")(3,0,A)::CALL LINK("EXEC")(A)
     As in Basic example but first initializes Memory
     Expansion and loads E/A utilities into low memory.
```

Further information on Disc Control Board or MINI-PE SYSTEM may be obtained from: Peter Schubert
P.O. Box 28
Kings Cross 2011
(02)358 5602

SUTHERLAND REGIONAL USERS GROUP

The past two months have seen a sharp increase in the number of local members attending the Regional meetings. Although we are not quite to the stage of putting up the house full sign, it now appears necessary to have two systems available on meeting nights.

A special welcome is extended to Dick and Adrian Baker of Lugarno who recently joined the group and also to Jack Krupski of Blakehurst who is our first new recruit to T.I.S.H.U.G.

Derek Wilkinson demonstrated his new ramdisk at the last meeting which proved to be quite interesting as this was the first time that the card had been tested. Further interest has been shown by other members in this product, after seeing it in action, which will lead to several orders being placed with the shop for a re-run of this kit.

I am still on the lookout for an RS232 card for the P.E. box if anyone can assist (Ph.5288775).

The group has also taken up Terry Phillip°s offer to copy some of the Library software, which will add to the collection of goodies available for review at meetings.

LIVERPOOL REGIONAL GROUP

No agenda was planned for the meeting at STANs place so some demos were shown.

Arto showed how the MINI RS232 can be used to make interfacing a sinch. It was configured as a MIDI interface with IN and OUT ports. The DSR source code for the EPROM was also shown and the new enhancements available: RTTY,MIDI,VIATEL as names with default baud rates. Baud rates from 50 to 31250.

PICASSO PUBLISHER was also demonstrated to show some of its features.

JOHN PAINE showed an EX/BASIC program 'EASY BUG' (same as Minimem) which was created using the Gram Kracker.

Some drives were looked at by JOHN, showing how he goes through the diagnostic procedure.

```
NEXT MEETING - HANS ZECEVIC
               33 Malinya Cr.
               Moorebank,
       PHONE - (02) 600 8716
        TIME - 7.30 pm
       THEME - Assembly Language
             - Using Data bases
```

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.

SAVE TUTORIAL By Scott Darling

After reading some messages concerning using the Save utility in the Editor Assembler package I awaited one individuals tutorial on the subject. But none appeared. So I decided to sit down and write one for it. Now some of you may know how to do this, but evidently not everyone does; so here goes.

Obviously to use the Save utility, you need the E/A package. But a couple of programs are helpfull. A sector editor such as DISK-AID, DISKO, and the like. The Mini-Memory module is a nice shortcut for finding addresses.

USING THE SECTOR EDITOR

First off any D/F 80 file will work, compresed or uncompressed. You should copy the program to a clean disk to make it easier to find using the sector editor. Load the editor and search for the end of the program. Disk-Aid will tell you where the last sector of a program lies. Toggle to Ascii and look at the sector info. There should be some names and references to VSBW, VMBW, START and so forth. Compressed code is easier to read by the way. Look carefully to see if the words SFIRST, SLAST, or SLOAD are there. IF they are skip the rest of this and go directly to "USING THE SAVE PROGRAM"!!

If those words are not there. Look to see if there are any lines of code that look like the actual program. The below will serve as an example. It is non-relocatable code. The "9" stands for that. An "A" stands for relocatable. (much easier to work with by the way) (this code was loaded into E/A editor, it will look different in sector editor)

```
0023 9E1D8B03807FD91F
0024 1E0467FEB9F
0025 6E046START 7FD06F
0026 :       99/4 AS
```

You will notice that at the line 24 there is a "1" to designate an auto start program. Change the "1" to a "F". This will defeat auto-start. There may be a "2" there, it also means auto-start.

Directly after that 1 or 2 is an address. This is the "START"ing address of the program. SFIRST and SLOAD will equal this address. The starting address also appears in front of the word "START". As is the case with most "DEFS" in this location.

To determine the SLAST is a little trickier. Just before the 1 or 2 auto-start was an address >E1D8. Now count over 1 byte for each 2 letter number combos. Excluding the B and 7. In other words: 9E1D8_0380_FD91_.

This will make SLAST = E1DC

Another example is for RELOCATABLE code:

```
0023 A01D8B0380FD91F
0024 100007FEB9F
0025 60000START 7FD06F
0026 :       99/4 AS
```

A01D8_0380-FD91_

In the above example I have changed a few numbers to show what relocatable code might look like. You will notice the "1" for auto-start is still in place, but there now are "0000" instead of and address. But 0000 is an address. It means that no matter what the lead address is, that the start of the program is "x000" So SFIRST equals "A000" and SLOAD is the same. Now slast will work the same as above except this time it equals A0DC.
Confused?? I hope not. If so reread this before continuing!!

ADDING AN ASSEMBLY DEF PROGRAM

I hope that you realize that there was a reason for all of the above jumble! Be- cause now you are going to use those address to build a short ASSEMBLY program!

Type in the following:

```
       DEF SFIRST,SLOAD,SLAST
SFIRST EQU >A000
SLOAD  EQU >A000
SLAST  EQU >A0DC
       END
```

Save this program using whatever name strikes your fancy. Load the assembler and assemble using another name. Bypass list, and bypass options. It will take a whole 2-3 seconds to assemble and you are ready for the next step. This source code can be used indefinitely. Just plug in the appropiatte addresses.

USING THE SAVE PROGRAM

1) Bring up the "LOAD & RUN" option of the E/A module.

2) Load your non auto-start program.

3) Load the "DEF" program that you wrote. (unless the program already has the DEFs in the program)

4) Load the E/A "SAVE" utility

5) Hit enter to "PROGRAM NAME" and hit enter

6) The next prompt will ask for a name. Remember that it will increment the name the next ASCII character if the resulting is going to be larger than 8K (33 sectors), plan accordingly.

7) Now hit enter, and try running your NEW program in "RUN PROGRAM FILE" option.

If it runs you are home free. If not you may want to go back and check to see if used the correct addresses. Most of what I wrote above was trial and error on my part, so experiment and play around with the programs.

The above was a rather short description. There is a shortcuts to find addresses that uses the Mini-Memory.

Initiate the M-M, then load the program. If it starts running you know that you have to defeat the auto-start feature. Escape the M-M and bring up Easybug. Do "M7020" and look at page 74 of the M-M manual. 7020 is default entry address. Or SFIRST and SLOAD. Step down to 7024 and this is the last free address in high memory or the end of the program. IF 7024 is "0000" then you know that the program is probably absolute code.

My only dissapointment is determining how to save low and high memory code. If a program loads at >2000 to >4000 then continues at >A000. The above steps are useless, because it assumes a continuous count between two addresses. I have not figured out or taken the time to attempt this format.

If anyone can add to this file, Please feel free to do so. I am by no means an expert when it comes to assembly. Have fun and I will answer any questions within my capability!

020 09 21 22 #FF A0DC

FORUM

Q. I have problems with my Chess module, the screen fills with random characters and sometimes gives a MEMORY FULL error.

A. The most likely cause of problems of this type is dirty contacts in the module and the GROM port socket, which is where the module plugs in. This problem affects all modules, especially those with double sided contacts, such as Extended Basic & Terminal Emulator II.

a) Carefully clean the contacts of the Command Cartridge using a clean cotton cloth over the end of a small screw driver. Ensure sufficient layers of cloth to prevent the screwdriver from touching the contacts or scratching the board. Observe static precautions.

b) If lockups are persistant then dismantle the console (observing static precautions) and remove the front of the cartridge port connector. Remove the oiling pad, (which will probably be loaded with dirt), and discard the pad. Clean the socket with a clean cotton cloth carefully pushed into the contact area with a blunt knife blade or small screwdriver. Flush out contacts with a FREON contact cleaner if available. Replace the front part of the connector which acts as a guide for the Solid State Cartridges.

Ensure that the contact leaves no oil or other residue and that the contacts are not scored by any form of abrasive cleaning method.

Q. How do I redefine the colour of the cursor in extended basic.

A. The cursor is ascii character 30, which is in colour set 0 in extd basic. The statement CALL COLOR(0,16,1) will redefine the cursor to white with a transparent background. If redefinition of both upper and lower case is required then you could use the following:

```
FOR S=0 TO 12 :: CALL COLOR(S,16,1):: NEXT S
```

Micropendium published the following letter:-

> "If your computer locks up while saving an Editor/Assembler file then type CALL LOAD(-31860,96,41) in BASIC to return to E/A without reinitializing the memory expansion. You can also use this to rerun a program that you loaded in E/A."

By experimenting, I have discovered that other GPL routines can be executed from Mini-Memory BASIC, Editor/Assembler BASIC, and Extended BASIC usisng a similar method. To do this, increment the subroutine stack pointer (@>8373) and place the wanted subroutine GROM address on the next location of the stack. Further, these values need to be followed by 152,0 (>9800=GRMRD).

Hence the GPL link is thus:-
CALL LOAD(-31885,144,"",-31858,A,B,152,0)
where A is the high byte, and
B is the low byte of the address of the GROM GPL subroutine.

Some of the routines tried are:-

with Extended BASIC after a CALL INIT command or BASIC with an E/A or MM module:

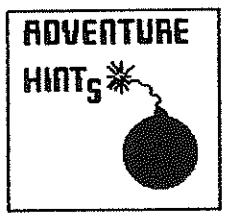
```
CALL LOAD(-31885,144,"",-31858,0,52,152,0) ACCEPT TONE
CALL LOAD(-31885,144,"",-31858,0,54,152,0) BAD TONE
CALL LOAD(-31885,144,"",-31858,0,32,152,0) POWER UP
CALL LOAD(-31885,144,"",-31858,8I,169,152,0) GARBAGE
                                              COLLECTION
```

and with TI BASIC only with an E/A or MM module:

```
CALL LOAD(-31885,144,"",-31858,34,23,152,0) RUN
CALL LOAD(-31885,144,"",-31858,34,25,152,0) NEW
CALL LOAD(-31885,144,"",-31858,34,29,152,0) LIST
```

All these routines were executed from a running programme. So far I have not encountered any problems but my question is: Is this a valid and safe way to perform a GPL Link? Are there situations which will cause the stack to be larger or smaller than usual? George Meldrum....Bulli.

Dear George,
The question you pose is quite a meaty one. The general concensus is the calls are valid. Depending on how you propose implementing them, you may need to consider if you are going to cause any interrupt routine conflicts which may cause your system to lock up.



MYSTERY FUNHOUSE

Part Three

Once in the merry-go-round room, push the blue button to stop the ride. Go merry, then go horse. Climb the pole in the horse's back, which brings you to the top of the ride. Look up, and GameSig Archives Page GSA-1407 you will see a rope. Jump! Now look, and you will see you are on a catwalk. Go East, and unlock the door. Drop the key, and look at the shelves. Grab the flashlight and the wrench, then head back West, and climb all the way back down again, then return to the knob room. As you pass the fortune-telling machine, pick up the "out of order" sign.

Pull the green knob, then go South to the shooting gallery. Drop the sign there, then return to the knob room. Get the match, then pull the yellow knob. This takes you to a small room with strange music. Go North into the maze (again!), then go East, South, East, South, East, and you're in the Mirror Room. From there, go South out of the Fun House.

Now it's back to the parking lot. Open the grate. You will only be able to open one bolt, but you can slide the grate to make room for yourself to go down. Drop the wrench and get the gum. Turn on the flashlight, then go down the manhole, and East to the room with the second grate.

Close the door, and drop the ticket. Now, the big moment has arrived! Remove the heel from your shoe. A couple of things will fall out. One is a note expaining what this is all about, the other is a fuse. Drop the heel and the note, and get the fuse.

Chew the (yuck!) gum again, then stick it to the fuse, then stick it to the grate. Light the fuse, and POP! the grate blows open. Get the ticket, then go through the hole. Now up through the shooting gallery (good thing you put that sign there!) and South to the hidden lab. There they are...the secret plans! Grab them and...congrats! You have successfully completed your assignment! After all this, you could use some peace and quiet. How about a trip to some faraway place, like Egypt? In fact, you might even want to visit the pyramids.....

SOME ARTICLES ON THE MODULES FOR THE TEXAS INSTRUMENTS COMPUTER

BY D.N.HARRIS

I have already commented on the use of Terminal Emulator for a Word Processor.

Those of you with a Console Writer sometimes complain there is no way to get rid of blank lines, whereas in fact there is if you would just sit down and read the Instructions!

The FUNCTION 7 INSTRUCTION DELETES LINES AND THE FUNCTION 8 INSTRUCTION CREATES BLANK LINES, THE FUNCTION 1 INSTRUCTION GETS RID OF EITHER BLANKS OR CHARACTERS IN A LINE AND THE FUNCTION 2 INSTRUCTION CREATES BLANKS INTO WHICH CHARACTERS CAN BE TYPED AND SPACES THE WHOLE LINE ACCORDINGLY.

Now for the bugs. The well known bug is Selection 7...hang up! Another one is fields of blank lines above or below headings.. centred headings appear left justified and fields get flung about.

Remedy;

use fields of Asterisks or some other decorative character above and below your Signature on a document, and above and below your Adress on the top. AFTER COMPOSING THE DOCUMENT BUT BEFORE PRINTING THESE FIELDS BE DELETED USING FUNCTION 3 OR FUNCTION 7 EVIDENTLY THIS PUT BLANKS IN TO BALANCE BUT HITTING ENTER AT THE END OF A LINE OF TEXT DOES NOT RELIABLY PUT ENOUGH BLANKS IN TO STOP THE CONCATENATION LOOP GOING OUT OF KILTER. This nearly always works, and can save a document from being mangled even if the printer has a slightly wonky buffer.

The use of Function 7 just before Cassette Save will enable you to trim all those blank lines out of the buffer from below your text. Just hold Function 7 until the "bong-bongs" turn into a rhythmic throbbing and get no faster.

When you reach this point all you save is the print, not all the unused blank lines in the buffer.

When you load from Cassette first select EDIT then quit back to the main menu as this initializes the buffer for printing otherwise a lot of garbage will come out including the title of the programme and possibly some Assembler as well followed by the printing at some unpredictable point.

A good practice is to go to the EDIT and clean up any blanks before printing or before saving so as to ensure the buffer is ready for printing and also to get rid of the excess blank lines that will otherwise use up the medium either paper or tape.

The other thing that may be of avail about Modules is the techniques of Games firstly starting with a semi-serious matter; VIDEO GRAPHS can be saved to Cassette yet one almost never sees members in the act of swapping patterns saved from VIDEO GRAPHS nevertheless suppose you get a Cassette full of stamp patterns flag patterns or portraits in colour of human beings done in "MOSAIC" out of VIDEO GRAPHS. To view the whole Cassette quickly, start the Cassette playing at any convenient point by hitting semi-colon and then choosing L or alternatively go in through the front door from the initial menu.

As soon as each pattern is loaded hit ENTER but do not switch off the Cassette Recorder, just look at the pattern and then engage L again so that the patterns are loaded continuously and you can control the action with one hand until you come to the pattern you especially want to look at . THEN adjust the Cassette machine. No use being "smart" and trying to use a Disk Drive as there is no such option with Video Graphs!

Getting 1,300,000 + in Buck Rogers..... The first screen has 7 of everything, and the rule is fly the ship fairly quickly through the Electron Posts and then watch hoppers VERY CAREFULLY as they are deceptively agile in this first screen, and there is time to let discretion be the better part of valour!

The second screen does things by 9s, and the need is to be a little more careful with the electron posts and take the stick back when shooting hoppers although pushing it smarly forward now and then is useful to dodge the hoppers.

I am assuming the knack of blasting spacecraft comes naturally, but when you get to the mothership putting the joystick forwards and going after it is often the safest manoeuvre

The way continues from 9 to 15 and then at last the green and red screen turns up and now although you may crash a few, do not slow down in the electron posts and do not ease up on the firing at the hoppers and finally try not to miss any spacecraft as this is a bottleneck frame demanding speed in everything.

The blue and red frame that follows is just as tough.

The green frame that follows you can ease back a little going through the electron posts, and begin to try to keep the hopper-hunt a little to the right of the screen.

In following screens the electron posts should be taken along a narrower band about 1/6 of the screen just to right of centre. The hoppers should be hunted only about the centre of the screen.

Getting the spacecraft becomes a kind of magic reflex action like spraying horts without getting stung. The rate of fire goes up so you get about 10 spacecraft in every burst so it is not so important to save fuel and more time can be spent on avoiding electron posts and watching hoppers.

At 900,000 the display spells out a message in Japanese in the score line, and after 999999 the Japanese characters are replaced by a string of zeroes. Here we go, now for 2,000,000 you think! At about 2,300,000 the fuel keeps burning up but your ship goes into slow motion and blows up, but sometimes by pushing the joystick as far forward as possible the end can be delayed a little. I suppose the Japanese thought a million is enough and now for a little homework or whatever!

I recently got hold of PARSEC and found the technique to get things is;

1. Three positions of the joystick will get all the swoopers and arrowships. Three different heights and wait a bit and then fire.
2. The ships that shoot back need to be sprayed like hornets three or four quick sweeps across their field of fire while firing will do the trick.
3. The Dramite ships can sometimes be better beaten in Lift 2.
4. The explosive little flying saucers need a good joystick to chase down the screen.

DOUBLE DENSITY DISK CONTROL BOARD
GOES INTO PRODUCTION (at last!).

The production version of the board has been tested and all looks good. A quantity of 25 or 50 boards will be ordered soon. Orders will be taken on the following basis:

... a) Payment of a deposit of $80 to the SHOP or Peter Schubert.

b) A receipt will be issued to purchaser with the date of payment.

c) As boards are completed and tested they will be supplied in order of date of payment of deposit. So first to order gets first delivery.

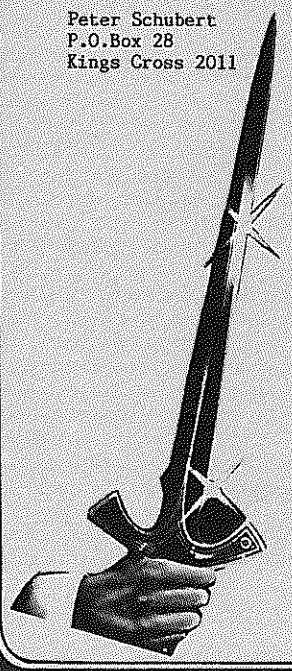
d) The balance of purchase price must be paid when you pick up the board.

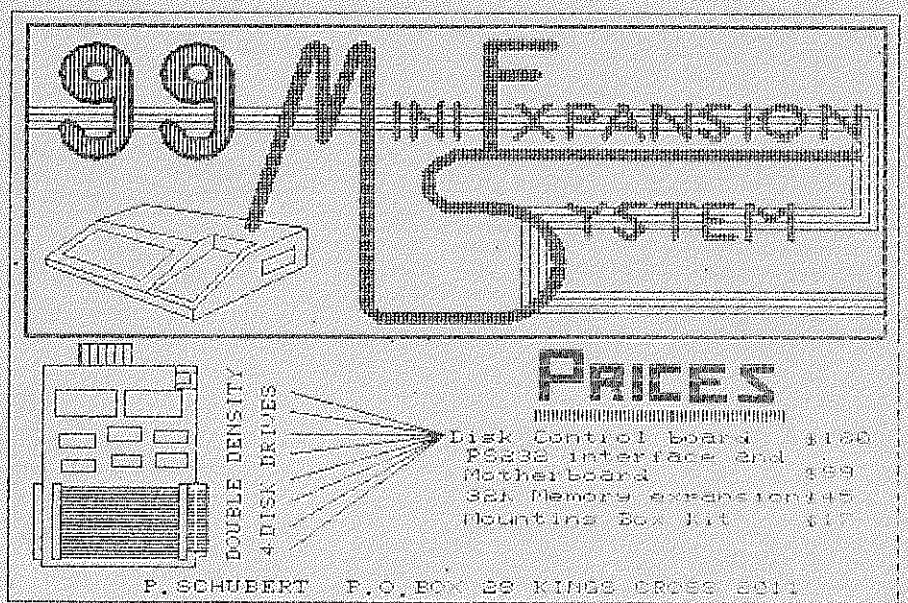
This Disk Control board is designed to fit into the MINI-PE BOX, and it plugs onto the RS232 board which some of you already have. Even if you do not have the RS232 board (which also has provision for 32K Memory) you can still order a D/C board as it may take 3 or 4 weeks to get it. The RS232 board is available now and I also have a BOX KIT and drill and cut details if you wish to fit your own. The box size is 120mm wide x 55mm high x 170-190mm long and diecast aluminium is prefered for shielding.

If you have 32K matchbox memory in console and wish to add D/C board on its own, it is possible to fit 44 way connector to it but the board does not have any provision for mounting into box (and no space to drill a hole!!!), however it can be just plugged into the console as a tempory measure until one of the other expansion boards can be purchased although I do not recommend this. Besides the RS232 board other boards suitable for the D/C to mount onto are being designed. One is a PIO output port and also there is a 90-220K RAMDISK + 32K all battery backed.

Price of the Disk Control Board is $180 and this version is powered from Console. If you wish to power it externally provision has been made on the board for the regulator and connecting a 9 volt DC plugpak. If you need more information you can phone me on (02) 358 5602. Payment by BANKCARD must be to the club SHOP. Mail orders can also be sent to Shop or to me on this address;

Peter Schubert
P.O.Box 28
Kings Cross 2011






```
**************************
*                        *
* MORE PLOTS by Arto Heino *
*                        *
**************************
```

If you typed in the CALL PLOT from last month here is a threedimensional math function to try.

To put in your own maths change line 200. Try these:

```
Z2=SQR(X2*X2+Y2*Y2)/4

Z2=SQR(-X2*X2-Y2*Y2+3200)

Z2=SIN(X2/4)*X2*Y2/30
```

```
100 !THREEDIMENSIONAL
110 !BY ARTO HEINO 1987
130 !*****************
140 CALL CLEAR
150 N,M=8 :: XO,YO=-35 :: X1
,Y1=20 :: NO=(X1-XO)/N :: MO
=(Y1-YO)/M
160 A=112 :: B=112 :: A1=0.5
2359878+.5 :: S1=SIN(A1):: C
1=COS(A1)
170 FOR Y2=YO TO Y1 STEP MO
:: FOR X2=XO TO X1 :: GOSUB
200 :: GOSUB 210 :: CALL PLO
T(B-V/2,A+U):: NEXT X2 :: NE
XT Y2
180 FOR X2=XO TO X1 STEP NO
:: FOR Y2=YO TO Y1 :: GOSUB
200 :: GOSUB 210 :: CALL PLO
T(B-V/2,A+U):: NEXT Y2 :: NE
XT X2
190 GOTO 220
200 Z2=150/((X2*X2/300)+(Y2*
Y2/500)+1):: RETURN
210 U1=X2-Y2*C1 :: V1=Z2+Y2*
S1 :: U=INT(U1):: V=INT(V1):
: RETURN
220 CALL SOUND(200,110,2)::
CALL SOUND(200,660,2):: CALL
 SCREEN(16)
230 GOTO 230
```

```
**************************
*   JOYSTICK SAMPLING    *
*                        *
*   BY ARTO HEINO 1986   *
**************************
       IDT  'JOYTEST'
       DEF  JOYSTK
XMLLNK EQU  >2018
NUMASG EQU  >2008
FAC    EQU  >834A
GPLWS  EQU  >83E0
STATUS EQU  >837C
SAVRTN DATA >0000
MYWS   BSS  32
*
************************
*      MAIN ROUTINE    *
************************
*
JOYSTK MOV  R11,@SAVRTN
       LWPI MYWS
       CLR  R5
       LI   R12,>0024
       LI   R9,>0606
       LDCR R9,3
       LI   R12,6
       STCR R5,5
       AI   R5,-7936
       ABS  R5
       SRL  R5,8
       MOV  R5,@FAC
       BLWP @XMLLNK
       DATA >20
       CLR  R0
       LI   R1,1
       BLWP @NUMASG
       LWPI GPLWS
       MOV  @SAVRTN,R11
       CLR  @STATUS
       B    *R11
       END
```

COMMENTS ON T.I.99/4a
BY ERIC WHELAN
P.O.BOX 20
HEALESVILLE 3777.

At 73 years old on a part pension and part super you know either you can do it or you can't; so I decided to try.

I bought a T.I.99/4a a few books and set to work. Three years later with a consol, T.V., four recorders, XBASIC & SECURITIES modules, six club tapes and hours of learning to type it is a very useful piece of equipment. At worst I talk to it and it talks to me and I have learned a lot.

You want input from members. Here is a start. maybe it is a poor one, but it might help those who are better equipped, to join in.

I bought my TI99/4a to use financial information and I had to learn the lot alone. There has been progress. However, until the last issue of T.N.D. I had no idea I could contribute to the mag. via cassette and I did know the problems caused by hand written papers. Now I want to know if I can PRINT this without having disks, and any further hints on writing copy. I have enjoyed every issue of S.N.D. though most of it was way over my head because I came into the club too late to benefit from the early issues. However I have typed for hours to enter programs and spent more time debugging and learning to change them. My four tape recorders talk to each other, the console and the T.V.

It took weeks to achieve that, but I can run CS1 and CS2 together and load from all four recorders. I have about twenty books; including Flynn's "HOME APPLICATIONS" and "33 PROGRAMS" which have been taped for almost constant use. Not having a printer I have hand copied many results for ten year comparisons. They have been most useful.

Anyone interested in financial advancement could find these books very useful. I have graphed complicated results over twelve years from "BAR CHARTS" in "HOME APPLICATIONS". I am sure I should have advanced much faster had I been able to attend the club meetings and at last I've joined the MELB. Group hoping to be a regular attender. I could do with an extra 32k, a few more tapes and a couple of modules together with VINCE APPS book of T.I.99/4a PROGRAMS which would help extend some of the work I am doing and help tie up some loose ends in filing practice.

The next move is to learn as much as possible about word processing. This it will look like in print. Writing this is so different from my usual input I just cant visualize the result, or believe it will fit. What kind of information do members want? I dont know, but I should like many short ones on simple error finding. For instance I have learned to use "PRINT (R,C)" when a program crashes. However, what next when you get 0,0? I know it means the variable in (R,C) is 0, but what is the next move? The error is probably twenty lines away and I still have a couple to find. Probably the answer is in one of my books, but I haven't been able to find it. Another help would be to have a series on the headings in T.I. EXTENDED BASIC. This book is very good, but articles in the mag.by members who have solved the problems would be much easier to understand and use.

For instance an explanation of "IMAGE" and "PRINT USING" with a short program from someone who has used it would probably be much easier than sorting it out from 220 pages in the book.

Another subject could be on files. I am learning very slowly and with lots of mistakes. Files are very interesting and very useful, and there is much I could learn from one column practical input from experience.

In November 1985 Jane LaFlame had a very good full page on files, and there useful input. Perhaps someone could outline practical applications in future issues. They could start with the very simple uses and in five or six issues report on discovered ways to use files and to benefit from them. An introduction to programming would be a help. At this time the most interesting input would be on word processing. The more simple it is the better as I'm at the very beginning of the road and apart from your effort in T.N.D. I have no knowledge at all. And please, will someone tell me if it is possible to print this without disks because I shall'nt be adding them to the outfit and I should like to print short stories and poetry I have written over the last 20 years and check any future efforts for T.N.D. before I submit them to the editor.

I'm amazed at the progress I have made since I started this article. The first 250 lines took days; the rest to here have taken only hours. I can see much more than this being completed in minutes. There is no doubt the way to learn is todo, but help from others who also are learning can be very useful. Perhaps at a later date I may be able to contribute something worthwhile. In the meantime I'll be glad to exchange information per letter or cassette or through T.N.D. with anyone interested in the growing pains of learning to use our TI99/4a.

I'm using National RQ 2106 and RX 5100 with 2 Sanyo DR 201 Datarecorders. It should be possible to get them talking to other recorders with a small amount of effort. It would be worth a try. Is anyone interested in swapping cassettes and experience of word processing or maybe anythihg else which may be interesting. I have an open mind, but of course my main interest is financial followed by learning anything about T.I.99/4a. Perhaps discussion on various pages of T.I.EXTENDED BASIC could produce some interesting information.

Possibly this could lead to production for T.N.D. which would help other members. At worst it could improve typing.

Your's Eric.