

042 8702
T.I.S:HUG

FEB '87 \$2

REGISTERED BY AUSTRALIA POST PUBLICATION NO. NBH5933



NEWS DIGEST

PO BOX 302
CARLINGFORD NSW 2118



TI.99/4A Owners Home
Computer User Group.
TISHUG NEWS DIGEST
November 1986

Correspondence to:

P.O. Box 302
CARLINGFORD
NSW 2118
Australia.
Tel.(02)3191009

Texpac BBS

COMMITTEE MEMBERS

Group Co-Ordinator:

Fred Morris

Secretary:

VACANT-VOLUNTEER PLEASE! Treasurer

Mel Copeland.....Tel.(047)351340

Communications Co-Ordinator:

Shane Anderson....Tel.(02)5690926

Merchandise Co-Ordinator:

Chris Buttner.....Tel.(02)8717753

Technical Co-Ordinator:

Robert Peverill...Tel.(02)6024168

Software Co-Ordinator:

Terry Phillips....Tel.(02)7976313

PUBLICATIONS LIBRARIAN

Brian Graham.....Tel.(02)7743223

TEXPAC BBS SYSOP

Ross Mudie.....Tel.(02)4562122

REGIONAL HOME GROUPS

Country Club:

Brian Graham.....Tel.(02)7743223

Glebe:

Mike Slattery.....Tel.(02)6920559

Penrith:

Mel Copeland.....Tel.(047)351340

Central Coast:

Russell Welham....Tel.(043)924000

Liverpool:

Stan Puckle.....Tel.(046)256157

Bankstown:

Peter Pedersen....Tel.(02)7722396

Illawarra:

Geoff Trott.....Tel.(042)296629

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

Membership fee.....\$8-00

TISHUG News Digest....\$25-00p.a.

Overseas Air Mail...US\$30-00p.a.

Publication Library...\$5-00p.a.

Texpac BBS.....\$5-00p.a.

Members of other

TI.99/4A User Groups..\$10-00p.a.

Public access.....\$25.00p.a.

UP FRONT

IN THIS ISSUE



- Page 2 - General Information and Editorial
- Page 3 - Micro Examination (Review)
- Page 4 - Review of Intern by Ben Von Takach
- 4 - Digitizer Pad for 99/4A by Arto Heino
- 4 - AGM Details
- Page 5 - Customising TI Writer by Bob Pass
- Page 6 - Link It 10 by Ross Mudie
- Page 7 - More Link It
- Page 8 - Link It continues
- 8 - Computer Hackers Wish List by Robert Brown
- Page 9 - Communications - World Time Chart by Shane Andersen
- 9 - 40 Column Text Mode by S Peacock
- Page 10 - Flipping Single Sided Floppies
- 10 - Jenny's Younger Set
- Page 11 - More Jenny's Younger Set
- Page 12 - Programming Music by Robert Montgomery
- Page 13 - Programming Music continues
- Page 14 - Windows for your 99/4A by Arto Heino
- 14 - Midi, Music and the TI99/4A by Arto Heino
- Page 15 - RLE Files by Arto Heino
- 15 - Subroutines
- Page 16 - Software Column by Terry Phillips
- Page 17 - TISHUG Shop by Chris Buttner
- Page 18 - Techo Time - Interfacing the LX86 Printer
- Page 19 - Classified Ads
- 19 - Input Output - TI Mail
- Page 20 - Hardware Gallery - Diagnosing Hardware Malfunctions by Geoff Trott
- Page 21 - Hardware Gallery continues
- 21 - Library TND Index by Brian Graham
- Page 22 - Freeware Version TI99-Opoly Update
- 22 - Regional Reports
- Page 23 - Keys to your Future (Paid Advertisement)
- Page 24 - All About MAX-RLE by Steven Shraibman
- 24 - Location Map for AGM

Program Liftout:

- Page *37 - Automatic Mouse Maze
- *37 - Tigercub Anagrammer
- Page *38 - Cassette Sleeve Maker
- *38 - Disk Sleeve Printer
- Page *39 - At the Beach
- Page *40 - Shaded Velvet

Plus huge AGM Pull Out supplement inside this issue.



Once again, another BUMPER issue of the TISHUG NEWS DIGEST which will hopefully keep you busy for another month, and the final edition for me as Editor. I will be stepping down as both a Committee member and Editor of this publication. After 6 years of involvement, I will be taking a break, and going on to do other things within this group in a non-official capacity. Who ever takes on the task as Editor at the next meeting, will be invited to my home on the paste-up day for the next publication, where I'll be showing him/her how this is put together. I ask that, if you haven't become involved in the running of this, your group, that you consider becoming more involved. I believe that most of us don't wish to see this group fold, but without your help and support, this may be a possibility. I ask that you attend the A.G.M and give the club your enthusiastic support.

Cheers 4 now
SHANE ANDERSEN (Founder)



MICRO EXAMINATION

SHANE K. ANDERSEN



New Products RUNdown

Myarc has been going through some changes recently, and, infact replaced many of their administration. We found this out after one of our members send them a letter complaining about lack of service.

With the letter from Myarc, came the following full details of the new computer which they have been working on, called "GENEVE". It has some good features, but I still wonder (1)If it will really eventuate, and (2)if we will see it here in Australia in any great number if at all. Ever since TI gave up on us, there have been promises from many companies, of support which has not been forthcoming, or news of new consoles and compatables which have fallen by the wayside. Now, Myarc, which has been going through its own bad times promise the Geneve, and all we can do is simply plod on with what we have, with the assistant of our Club, and see what happens.

Lets take a look at the features of the Myarc Model 9640 Family Computer, called the Geneve...

TMS9995 u-Processor runs 3-4 times faster than the 99/4A:

- * Runs same instruction set as 9900 used in 99/4A PLUS 4 new ones.
- * Pipeline processor (i.e u-processor performs several functions SIMULTANEOUSLY).

V9938 Advanced Video Display Processor:

- * Is software compatible with the TMS9918A (used in 99/4A)
- * Uses 46 registers for high speed "HARDWARE" graphics commands
- * Commands include: DRAW SEARCH POINT(status) BLINK FILL MOVE ANIMATION and more.
- * Uses Colour Pallet of 512 colours on the screen at a single time
- * 7 modes of graphics operations; some modes allow 256 colours
- * True BMG (Bit-Mapping-Graphics) operation
- * Both composite (like the 99/4A) and analog RGB outputs (like the Atary ST & Commodore Amiga)
- * Supports up to 256 colours per screen in the 256 by 424 mode or 16 colours in the 512 by 424 mode
- * Comes with 128K bytes of video RAM (8 times the amount of the 99/4A)

Real-Time Clock Chip:

- * Gives you and your program instant access to date and time

Sound Chip:

- * Compatible with 99/4A(i.e 3 simultaneous tones, and 1 noise)

Awesome amount of RAM:

- * 512K of CPU RAM (User configurable between CPU-RAM, RAM-DISK or PRINT-SPOOLER)
- * Expandable to 1 megabyte with Myarc 512K Card
- * In 99/4A mode 64K of the 512K becomes GROM and 16K Cartridge Rom

Built-in Mouse Interface:

- * Installed hardware allows for the MS mouse to be connected directly to the 9640 board
- * Basic language support for the mouse built in. Uses the industry-standard MacIntosh mouse commands

Standard Joystick Interface:

- * Joystick interface is compatible with one used by 99/4A

Hardware & Software Support for most commonly-used peripherals:

- * Floppy Disk Controllers include Myarc, TI, CorComp
- * RS232 cards include Myarc, TI, Corcomp
- * Ram-Disks include Horizon

Software Support Supplied with 9640:

- * Myarc DOS (Similar to MS-DOS 2.1)
- * Myarc Advanced Basic
 - Compatible with TI Extended BASIC & Myarc ExBasic2
 - Supports all models of the Video Processor including 80 column
 - Supports Windows
 - Supports easy to program Mouse Commands
 - Combined Text and Bit-Mapped-Graphics modes
 - Drawing Commands such as Circle, Rectangle etc are built-in
- * Program patches to make TI-Writer 1)more powerful and 2)display 80 columns
- * Program to SAVE your 99/4A cartridges to disk

SOFTWARE SUPPORT FROM OTHER SOFTWARE VENDORS include:

PECAN Systems:

- * UCSD Pascal Runtime (included free with the 9640)
- * UCSD PROGRAMMING LANGUAGES(at additional cost)
 - UCSD BASIC UCSD FORTRAN
 - UCSD COBAL UCSD PASCAL
- * PLUS thousands of other applications ranging from Pig Management to Office Management to Home Education

DATABIOTICS:

- * The Music Shop
- * Super-Super 4th
- * Super Word
- * Lush Brush
- * Professional Business Assistant
- * Macro Assembler
- * Pilot
- * Terminal Connection

PAUL CHARLTON: PIKE CREEK COMPUTER Co:

- * Fast-Term II
- * General Purpose Accounting Software

CLINT PULLEY: BRYGHT-DATA:

- * Big C Compiler
- * Professional Business Acc'ting S'Wr

INSCEBOT: CSI DESIGN GROUP:

- * TI-Artist
- * MacPaint Equivalent

Well, there you have it. Even if this wonderful computer doesn't make it, their heart is in the right place. And besides, we still have a great little computer which is still alive and well, with new Hardware & Software being produced by the Technical team of TISHUG and other similar groups.

1987 holds some very interesting surprises for us, and its going to be fun to continue with your computer as long as you don't loose your imagination.

Regards
SHANE ANDERSEN

Review

TI99/4A INTERN

by Ben von Takach

The book entitled TI99/4A INTERN by Heiner Martin is a most unreadable book. The 207 page A5 format soft cover volume barely contains 5 text pages. The rest is filled with details of the TI99/4A's internal architecture. TI was and still is very secretive about the finer details of its products generally, and this policy also includes the 99/4A although it was abandoned years ago. The author of this book did a splendid detective work to unravel the many unpublished details of the 99/4A. Only a few months have passed since yours truly was attempting to collate the token table of this computer the hard way. Now I found the complete token list in a coherent and orderly manner in this book, together with its associated program steps.

As I said at the start, one does not read this book, one studies certain details of interest. The publication may be compared with the telephone directory. One does not read it, yet it is an essential part of daily life.

Here are the details.

The first part covers the system ROM. A 70 page listing is provided of the console ROM from 0000 through 1FFF. The listing is augmented by extensive remarks to aid understanding.

The GPL (Graphic Programming Language) commands and command formats are analysed in a lengthy chapter. These are preceded by a short description of the language.

The author deals with GROM 0 next.

A listing of GROM 0 -again complete with remarks- is followed by a hex-dump of GROM 0 from 0000 through 17FF. Inclusion of the hex-dump listing is very useful, as TI has produced several versions of GROM 0. This print out will aid comparison.

The BASIC GRAM-s (GRAM1 and GRAM2) are analysed next. A listing of GRAM1 & 2 is reproduced from 2000 through 57FF. The BASIC GRAMs have no known versions, thus inclusion of a hex-dump printout is not necessary.

The volume finishes with a short reference to Extended Basic.



Mr. Martin admits that due to the very limited information available from the manufacturer, the accuracy of the comments can not be guaranteed. The brief and very much condensed explanatory pages are most helpful to understand the internal functions and the logic of the data management used by this computer. Alas, this is not a book for the beginner, albeit is highly recommended for any insomniac TI-user. It will put any novice to sleep in less than 5 minutes! On the other hand it is an essential tool of the serious programmer. The publication is an original work, it does not repeat any related information already published elsewhere, thus prior study of other reference material is mandatory. The in-depth study of the Editor/Assembler manual, as well as the TI Home Computer Technical Data Manual prior to tackling this book is a must. In short, knowledge of the 9900 Assembler is essential.

This long overdue study is a valuable addition to the 99/4A information library.

The book was published by VTH (Verlag fuer Technik und Handwerk GmbH), Baden Baden, West Germany in 1985. ISBN number: ISBN 3-88180-009-3. Translation by Peter Coates. The volume may be purchased through TISHUG, current price (subject to exchange rate fluctuations) \$ 23.00

DIGITIZER PAD FOR YOUR 99/4A

by Arto Heino

Digitizer Pads have been around for your TI for a long time but we say where are they!!!

Well we don't live in the USA so what do we have here Downunder! Micobee AUST have imported a graphic pad from the US they call it CHALK-BOARD.

The price from Micro-bee is approx \$137 cheap. You can use it almost straight away with TI-ARTIST which has a load external device option.

USING IT

Plug it into a spare RS232 socket and move your finger(or back of pen) on the pad surface. To toggle on/off you can either make a foot switch or button somewhere. The pads resolution is 127*127 a bit limiting but you can make it window around.

The software must be written for it to be used with TI-ARTIST but it looks like it will be a cinch!!

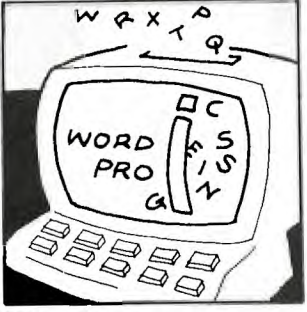
A.G.M

NEW VENUE

SAT: 7th FEBRUARY '87

AT: RSL BURWOOD,
96 SHAFTSBURY RD.
BURWOOD

2 PM SHARP !!!



CUSTOMIZE TI-WRITER.

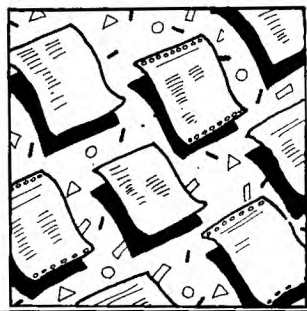
Author : Bob Pass
99'ER ONLINE February 1986

Tired of those screen colors on TI-WRITER ? Poor resolution on your monochrome monitor ? You can change the five default values of screen colors available with CTRL #3 in about 15 minutes. This tip is from the Ozark user group who got it from MICROpendium who got it from Jim MacEachern (Author of Wycove Forth). God bless the grapevine!

- First of all, materials needed :
1. 2 blank initialized disks.
 2. A disk fixer.
 3. TI-WRITER system disk.
 4. Hex color code chart, see below.

HEX COLOR CODES :

- 0 = Transparent.
- 1 = Black.
- 2 = Med. Green.
- 3 = Light Green
- 4 = Dark Blue.
- 5 = Light Blue.
- 6 = Dark Red.
- 7 = Cyan.
- 8 = Med. Red.
- 9 = Light Red.
- A = Dark Yellow.
- B = Light Yellow.
- C = Dark Green.
- D = Magenta.
- E = Gray.
- F = White.



Now for the process :-
Make a backup copy of your TI-WRITER system disk and change the name of the backup disk to "TI-WRITEF1". Put away your original disk and label the backup to indicate that it has altered screen colors. Now, copy file "EDITA1" onto the second BLANK disk (no other files on this disk). I will call this the scratch disk. Set aside the backup copy and perform the next steps on the scratch disk only.

Load your disk fixer and load sector #33 (hex 22) from the scratch disk and display it in hex format. Starting at byte 244, (hex 00F4) there will be a series of five words in the from 87xx . The words are : 87F4, 8713, 87F3, 8717, and 87F1. The "xx" portion is what needs to be changed. The first hex character controls the foreground color while the second controls the background. Chose appropriate color combinations from the chart above and change these five words, keeping the "87" prefix. For example, changing the first word 87F4 to 871E will change the White on Dark Blue screen to Black on Gray. The colors you can toggle with CTRL 3 will start at the second word, proceed to the fifth one word at a time, then to the first, and then repeat.

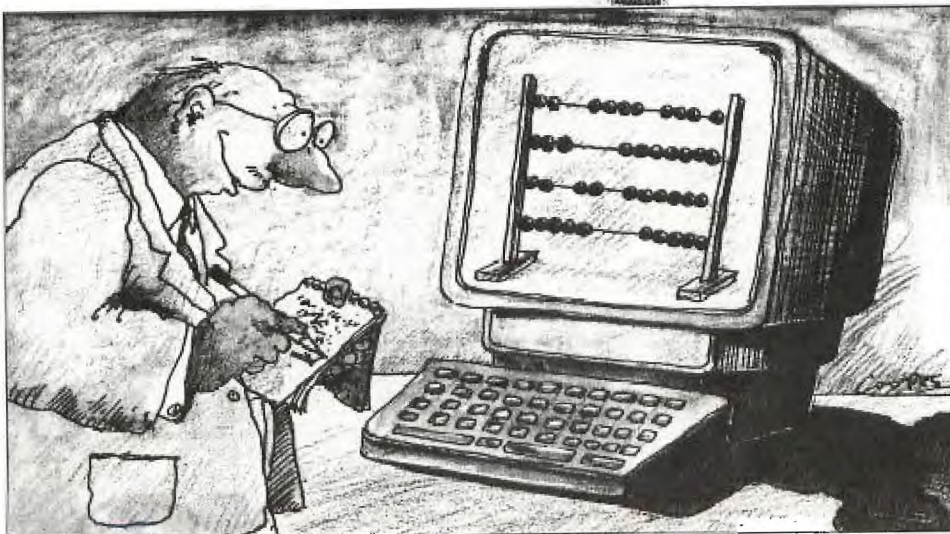
Write the changed sector back to the scratch disk and the worst is over.

Now, using your Disk Manager, copy "EDITA1" from the scratch disk to your backup disk prepared earlier. Insert your TI-WRITER module and load the text editor. Note that the main menu and the formatter screens remain as white letters on dark blue; these screens are resident in the module. Load a word file and test your new color combinations by toggling CTRL 3.

I would suggest that you keep your original version so that you can have ten different screen choices available. Or make a third version and have 15 choices!



* **G.U. at**
TSHUG



"That letter you lost, Miss Betts, never mind. I found it in my briefcase."

LINKING BASIC TO ASSEMBLY



BY ROSS MUDIE.

LINKING TO ASSEMBLY FROM EXTENDED BASIC.

By Ross Mudie of TISHUG. TND February 1987.

SUBJECT.

A disk or cassette loadable PEEKER for DSR ROMs and all of the CPU address space, with the capability of output to printer, disk or RS232.

INTRODUCTION.

This is the tenth article in the series, which I have written on the subject of linking to assembly from basic or extended basic. As readers have no doubt noticed the assembly source files are always heavily documented. The purpose of this is to help those who are trying to learn assembly. This month I have presented the extended basic program with the assembly object file in CALL LOADS so that the program can be typed in and run from cassette for people with 32K memory expansion and no disk drive in addition to providing the source file.

The program will also be included in the downloadable programs on TEXPAC Bulletin Board in February 1987.

For those starting to learn assembly, I believe that linked assembly is a good way to go. You can get a useful program up and running much more easily than a full assembly program by just using assembly for the parts which basic or extended basic can not do. Linked assembly can also be used to speed up the functions which execute too slowly in one of the basics. This program includes assembly because some things could not be done in extended basic & because in the formulation of the print string the extended basic was painfully slow.

The programs presented this month were developed to allow peeking at any Device Service Routines (DSR) ROM which is connected to the computer using assembly and extended basic.

The program can be fully contained in extended basic making it capable of being loaded from cassette. This is extremely handy if you wish to view a disk DSR ROM, when doing diagnostics if the disk controller is faulty or on a cassette system to view the program contents of an Axiom interface or any other peripheral device.

In addition to being able to peek and view the DSRs the program also permits peeking anywhere in the CPU address space from 0000 to FFFF. This allows convenient viewing of how the extended basic program is stored in memory or viewing the system ROM.

PROGRAM OVERVIEW.

In the extended basic program, you are given a list of some of the common CRU addresses and then you can enter the required CRU address in HEX. The extended basic program then converts the HEX value to decimal & passes it to assembly where it is stored.

The start and finish addresses are then entered in HEX and converted to decimal for passing to assembly & for control purposes, (when to finish). The assembly program prints on the screen and returns the print string to extended basic for use with a printer or disk drive. This even allows peeking at the RS232 ROM then printing out to a printer via the same RS232 card.

The required CRU address needs to be passed into the assembly program only once for the device to be read. If no CRU address is given then the default is the TI Disk Controller at CRU >1100. If a CRU address is not required, e.g., peeking outside the range hex 4000 to

hex 5FFF, then the extended basic will still prompt for the CRU address but it should have no harmful effect. If in doubt use a spare CRU address, e.g., hex 1200.

This is the extended basic program containing the assembly object program in CALL LOAD format.

```

100 ! SAVE DSK1.LOAD
110 CALL CLEAR :: CALL SCREEN(6):: CALL KEY(3,K,S)
120 ! This program, is written to allow the inspection
130 ! of the DSR ROM programs.
140 ! It is intended for use in the diagnosis of
150 ! partially or fully dead ROMs or just looking
160 ! at the programs in the DSR ROMs.
170 ! This is version 1.4 of the program, written by
180 ! Ross Mudie, 22nd December 1986.
190 ! The program allows inspection of any DSR ROM.
200 ! It will also allow peeking at any area of CPU
210 ! address space from >0000 to >FFFF. Just enter
220 ! any address for the CRU address, 1200 is safe,
230 ! then enter the start and finish addresses, e.g.,
240 ! start EEFO, finish FFFF will examine part of
250 ! this program.
260 XX(1)=4096 :: XX(2)=256 :: XX(3)=16 :: XX(4)=1 ::
PR$="Y"
270 FOR S=0 TO 14 :: CALL COLOR(S,16,1):: NEXT S
280 H$="0123456789ABCDEF" :: CRU$="1100"
290 DISPLAY AT(1,9)ERASE ALL:"DSR PEEKER" :TAB(5);"for
Cassette load in":TAB(6);"32K extended basic" :: TAB(7)
;"By Ross Mudie."
300 IF F THEN 510
310 CALL INIT
320 CALL LOAD(16368,67,82,85,32,32,32,36,244)
330 CALL LOAD(16376,68,83,82,32,32,32,37,10)
340 CALL LOAD(8194,40,118,63,240)
350 CALL LOAD(9460,200,11,38,52,2,224,38,86,6,160,38,10
,200,32,131,74,38,28,4,96,37,250)
360 CALL LOAD(9482,200,11,38,52,2,224,38,86,152,32,38,3
0,38,32,19,20,2,4,0,16,2,5)
370 CALL LOAD(9504,38,36,2,2,38,118,2,6,0,16,2,7,38,36
,220,149,220,183,6,6,22,252)
380 CALL LOAD(9526,5,133,6,4,22,245,216,32,38,30,38,32
,2,6,0,16,192,96,38,30,2,2)
390 CALL LOAD(9548,38,54,204,129,6,6,22,253,195,32,38,2
8,29,0,6,160,38,10,192,160,131,74)
400 CALL LOAD(9570,216,32,38,33,38,55,210,32,131,74,9,1
36,10,24,200,40,38,118,38,56,210,32)
410 CALL LOAD(9592,131,75,9,136,10,24,200,40,38,118,38
,58,2,10,38,61,2,6,0,6,210,50)
420 CALL LOAD(9614,9,136,10,24,194,104,38,118,222,137,6
,201,222,137,5,138,6,6,22,245,192,160)
430 CALL LOAD(9636,131,74,2,6,0,6,152,18,38,30,17,5,152
,18,38,35,21,2,222,178,16,3)
440 CALL LOAD(9658,222,160,38,34,5,130,6,6,22,243,2,6,0
,29,2,0,2,225,2,2,38,56)
450 CALL LOAD(9680,208,114,2,33,96,0,4,32,32,32,5,128,6
,6,22,248,4,32,32,24,0,38)
460 CALL LOAD(9702,4,192,2,1,0,2,2,2,38,55,4,32,32,16,1
95,32,38,28,30,0,2,224)
470 CALL LOAD(9724,131,224,4,192,216,0,131,124,194,224
,38,52,4,91,4,192,2,1,0,1,4,32)
480 CALL LOAD(9746,32,12,4,32,32,24,18,184,4,91,17,0,32
,32,0,29,46,126,48,49,50,51)
490 CALL LOAD(9768,52,53,54,55,56,57,65,66,67,68,69,70)
500 F=1
510 DISPLAY AT(16,1):"Some CRU Addresses:" : "1100 TI D
isk Controller": "1300 RS 232 (Primary)": "1500 RS232 (Se
condary)": "1800 Thermal Printer"
520 DISPLAY AT(22,1):"IBOO AXIOM Parallax TI": "1D00 Tri
ple Tech Clock": "1F00 P- Code"
530 DISPLAY AT(9,1):"CRU address HEX ";CRU$ :: ACCE
PT AT(9,21)SIZE(-4)VALIDATE(H$)BEEP:HS$ :: CRU$=HS$ ::
GOSUB 750 :: CALL LINK("CRU",DEC)
540 CALL HCHAR(16,1,32,288)
550 DISPLAY AT(11,1):"Start address HEX 4000" :: ACCE
PT AT(11,21)SIZE(-4)VALIDATE(H$)BEEP:HS$ :: GOSUB 750
:: ADDR=DEC

```



TISHUG NEWS DIGEST

LINKING BASIC TO ASSEMBLY



BY ROSS MUDIE.

```

560 DISPLAY AT(13,1):"Finish Address HEX 5FFF" ::
ACCEPT AT(11,21)SIZE(-4) VALIDATE(H$)BEEP:HS$ :: GOSUB
750 :: LASTADDR=DEC
570 DISPLAY AT(15,1):"Print out? Y/N ";PR$ :: ACCEPT
AT(15,21)SIZE(-1) VALIDATE("YN")BEEP:PR$ :: IF PR$="Y"
THEN PRINTER=1 ELSE PRINTER=0
580 IF PRINTER=0 THEN 630
590 CALL HCHAR(17,2,80):: DISPLAY AT(17,1):"Printer? PIO
or DSK1.DUMP etc": "PIO"
600 ACCEPT AT(19,1)SIZE(-28)BEEP:PN$ 610 OPEN #1:PN$
:: IF ADDR>16383 AND ADDR<24577 THEN PRINT #1:"Printing
DSR ROM opened by CRU ">&CRU$
620 PRINT #1:"ADDRESS * HEX * Interpret"
630 PRINT "HOLD...SPACE BAR to pause": :TAB(8);"<FCN>9
to escape": :
640 PRINT "ADDRESS * HEX * Interpret": :
650 CALL LINK("DSR",ADDR,RS$)
660 IF PRINTER THEN PRINT #1:RS$
670 ADDR=ADDR+6
680 CALL KEY(3,K,S):: IF K=15 THEN 720 ELSE IF S=0 THEN
710
690 CALL KEY(3,K,S):: IF K=32 THEN 690
700 CALL KEY(3,K,S):: IF S=0 THEN 700
710 IF ADDR<LASTADDR THEN 650
720 IF PRINTER THEN CLOSE #1
730 PRINT : "Press E to End or any other key to redo" ::
CALL SOUND(130,1400,4)
740 CALL KEY(3,K,S):: IF S=0 THEN 740 ELSE IF K=15 THEN
740 ELSE IF K=69 THEN END ELSE 290
750 ! HEXDEC
760 DEC=0 :: FOR X=1 TO 4 ::
DEC=DEC+(POS(H$,SEG$(HS$,X,1),1)-1)*XX(X):: NEXT X
770 IF DEC>32767 THEN DEC=DEC-65536
780 RETURN

```

This is the source file for the CALL LOADS in lines 320 to 490 of the extended basic PEEKER program.

IDT 'DMPmudie' Ross Mudie 22nd December 1986
DEF DSR,CRU Source=D8 obj=DD8 Vers 1.4.

* CRU Routine. This routine tells the DSR routine which
* Device Servive Routine to read.
* Extended Basic format:
* CALL LINK("CRU",ADDRESS) The address is in decimal.

```

CRU MOV R11,@SAVRTN
LWPI WS
BL @SUBREF Get CRU address
MOV @FAC,@CRUADR Save CRU address
B @END

```

* DSR Routine. This routine opens the DSR page then
* gets the address to start reading from extended basic.
* The HEX start address is placed in the print BUFFER
* followed by the HEX representation for the six bytes
* peeked. The program places the ascii interpretation
* for the bytes in the BUFFER.
* The routine then prints the 29 byte BUFFER on the
* screen with hex 60 added for the extended basic
* environment. The screen is then scrolled up one line
* and then the string is returned to extended basic for
* printing to the printer or disk drive if required.
* Extended basic format: CALL LINK("DSR",RS\$) .

```

DSR MOV R11,@SAVRTN Save return address to x/b
LWPI WS
CB @D2020,@NFULL Has the block N been filled?
JEQ CLRBUF If yes then go to CLRBUF

```

* This routine creates the HEX character look up table
* in the Block with the Symbol Starting (BSS) N.
* This table contains 00, 01, 02 through to FD, FE, FF.

```

LI R4,16 Counter for most significant chars
LI R5,TEXT Where to get the most signif chars
LI R2,N Where to start putting the chars

```

```

LI R0,10 Counter for least signif characters
LI R7,TEXT Where to get the least signif chars
LOOP1 MOVB *R5,*R2+ Put a most signif character in N
MOVB *R7+,*R2+ Put a least signif character in N
DEC R6 Finished this group of characters?
JNE LOOP1 If no, go and do the next
INC R5 Point to the next most signif char
DEC R4 Finished 16 groups of 16 yet?
JNE LOOP2 If no go and do the next group of 16
MOVB @D2020,@NFULL Flag to indicate N is FULL

```

* This routine CLearS the print BUFFER by writing >20s

```

CLRBUF LI R6,16 Number of 2 byte words to write
MOV @D2020,R1 What to write is placed in R1
LI R2,BUFFER Where to start writing in R2
CLOOP MOV R1,*R2+ Write the hex 20s in the BUFFER
DEC R6 Decrement counter, finished yet?
JNE CLOOP If no, go and write the next

```

```

MOV @CRUADR,R12 CRU address from CRU routine
SBO 0 Turn on DSR with Set Bit One
BL @SUBREF Get address in DSR range
MOV @FAC,R2 Save address in R2

```

* BUFFER is 32 bytes in length and the bytes are used
* as follows:

- * Byte 0, unused. Byte 1 is the length byte of 29.
- * Bytes 2 to 5, First address which is peeked e.g 4006.
- * Bytes 6, 9, 12, 15, 18, 21, 24 space, remain hex 20.
- * Bytes 7&8, 10&11, 13&14, 16&17, 19&20, 22&23; these
* words contain the two character hex representation of
* the byte value at a location which has been examined.
- * Bytes 25 to 30, this 6 bytes contain the interpreted
* values for the peeks of printable ascii characters.
- * Unprintable characters are replaced by a dot, i.e., a
* period (.). Byte 31 is unused.
- * Bytes 0 & 31 are unused to allow even word addressing
* in the ADDRESS routine.

```
MOVB @B29,@BUFFER+1 Length byte for BLWP @STRASG
```

```

ADDRESS MOVB @FAC,R8 Peek address first byte in R8
SRL R8,8 Right justify in R8, clear left byte
SLA R8,1 Multiply x 2 for offset in N
MOV @N(R8),@BUFFER+2 Move the WORD into BUFFER

```

```

MOVB @FAC+1,R8 peek address second byte in R8
SRL R8,8
SLA R8,1
MOV @N(R8),@BUFFER+4

```

* This routine places a 2 byte hex representation in
* the BUFFER for each byte peeked, e.g, A5, followed by
* a space (hex 20).

```
LI R10,BUFFER+7 Where to start in buffer
```

```

LI R6,6 Number of peeks to perform
BLOOP MOVB *R2+,R8 Put the peeked byte in R8
SRL R8,8 Swap bytes and clear left byte
SLA R8,1 Multiply x 2 for char lookup in N
MOV @N(R8),R9 Get char from look up table
MOVB R9,*R10+ Move most signif byte into BUFFER
SWPB R9 So that next line can get at LS Byte
MOVB R9,*R10+ Move least signif byte into BUFFER
INC R10 To allow for the space between HEX's
DEC R6 Finished the 6 peeks yet?
JNE BLOOP IF NO, go and do the next

```

* Intepretation for ascii printable characters.

```
MOV @FAC,R2 Where to do the interpretation from
```

```

LI R6,6 How many bytes to do
INLOOP CB *R2,@D2020 Is char below ascii printable #
JLT DOT If yes, go and do a dot (.)
CB *R2,@B126 Is char above ascii printable #
JGT DOT If yes, go and do a dot (.)
MOVB *R2+,*R10+ Put ascii printable in BUFFER

```



COMPUTER HACKERS NEW YEARS RESOLUTIONS.

By Robert Brown

```
JMP DECCTR          JuMP over DOT routine
DOT  MOVWB @B46,*R10+ Put a . in BUF for unprintable
      INC R2          To read next peek (only after .)
DECCTR DEC R6        Finished all the peeks yet?
      JNE INLOOP     If no, go and do the next
```

```
* Routine to print contents of BUFFER on screen.
LI R6,29            Number of bytes to write
LI R0,737          Start at screen row 24, column 2
LI R2,BUFFER+2     Where to get the start of line
PTLOOP MOVWB *R2+,R1 Put a byte value in R1
AI R1,>6000        Add hex 60 offset for extd basic
BLWP @VSBW         Write the byte to the VDP screen RAM
INC R0             For the next VDP screen RAM location
DEC R6             Finished yet?
JNE PTLOOP        If no, go and do the next

BLWP @XMLLNK       The easy way to scroll the screen
DATA >26           up one line.
```

* Send the string in the BUFFER to extended basic for
* the printer or disk drive.

```
SEND CLR RO          Element number in variable
LI R1,2            Argument number in link list
LI R2,BUFFER+1    Point to start of string in BUF
BLWP @STRASG      Transfer string to x/b

MOV @CRUADR,R12   CRU address from CRU routine
SBZ 0             Turn off DSR
```

```
END LWPI GPLWS      Return to x/basic routine
CLR RO
MOVWB RO,@STATUS  Prevent any indication of errors
MOV @SAVRTN,R11   Restore return address
RT
```

```
* Subroutine to get a numeric value from extended basic
SUBREF CLR RO      Element Number
LI R1,1           Argument number
BLWP @NUMREF      Get number from extended basic
BLWP @XMLLNK
DATA >12B8        Convert Floating Point to Integer
RT
```

```
CRUADR DATA >1100 This val is changed by CRU routine
D2020 DATA >2020   Two ascii spaces
NFULL BYTE 0       Flag for character table
B29 BYTE 29       String length byte for BLWP @STRASG
B46 BYTE 46        Dot ie (.)
B126 BYTE 126     Highest printable ascii character
EVEN
```

```
TEXT TEXT '0123456789ABCDEF' Characters for table N
```

```
SAVRTN BSS 2       For storage of return address
BUFFER BSS 32      Buffer for sending strings to x/b
WS BSS 32          For register Work Space
N BSS 512         Block for storage of hex table
```

```
NUMREF EQU >200C
NUMASG EQU >2008
STRASG EQU >2010
XMLLNK EQU >2018
VSBW EQU >2020
FAC EQU >834A
GPLWS EQU >83E0
STATUS EQU >837C
```

END



* I will look under my chair wheels for diskettes before rolling away from my desk.

* I will not turn up my stereo really loud, and then accidentally stick in a computer cassette.

* I will remember to save that long program I just typed in, before I try to run it.

* I won't put my ashtray on top of the disk drive vents.

* I won't keep my tape head demagnetizer in the same room as my diskettes.

* I will remember that using all of my strength on the joystick, doesn't make the game play any better.

* I promise I will never say, "But it will increase my productivity!"

* Before turning up my monitor's brightness control, I'll see how thick the layer of dust is on the screen.

* I will read instructions before I try to run a new program.

* I will make backups of my data disks only!

* I will always turn my computer off before sticking in a cartridge.

* Before complaining about a program at a user's group meeting, I'll make sure that its author is out of earshot.

* Before cussing out the computer or the programmer when an input doesn't work, I'll look and see if my alpha lock is depressed.

* I will somehow try to keep the ratio of sleeves to diskettes somewhat equal.

* I will not use words like "hex", "dump", "registers" in ordinary conversation - unless we're talking about witches, garbage dumps or hot - air heating systems.

* I promise I won't sit a can of soft drink on top of the disk drives.

* For that matter, this year I'll try not to spill anything on the disk drives either.

* I will remember how dumb it's to eat fried chicken while working at the keyboard.

* I will check my paper supply before I begin that really long printout.

* I will untangle my cables.

* I will turn down the volume when playing "TI-Invaders / Parsec" at 3.00am.

* Before I call a new BBS number, I'll call "voice" first to make sure it isn't really somebody's poor Aunt Sue.

* I will keep the dog out of the computer room.

* I will not remove disks while the read / write head is working.

* I will not throw the joystick on the ground, when I lose a game.

COMMUNICATIONS

.....

The Communicators with Shane Andersen

.....

- *For those of you with family & friends Overseas,
- *Wanna communicate with BBS's Internationally?
- *Ordering Hardware & Software around the Globe?

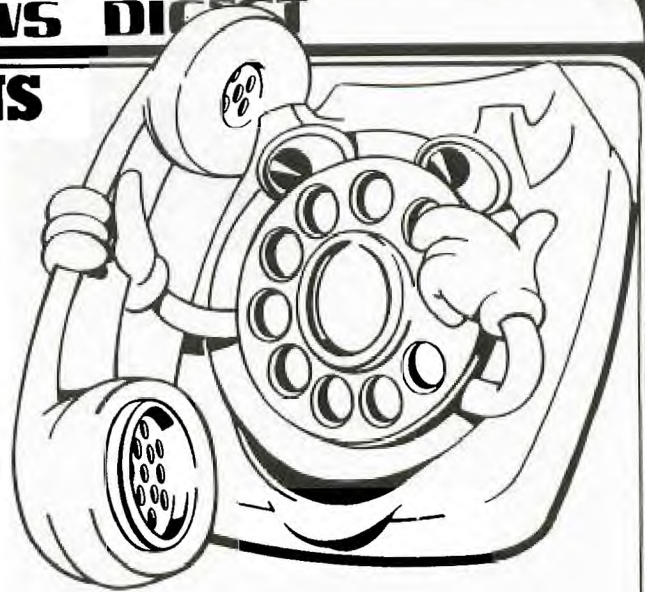
Well, the following will be of help to you, as you check the International Time as supplied by Blackboard Magazine (BBS) (02)5261343

Clock set at 12 noon (ET)...

STANDARD WORLD TIMES

Adelaide (Sth Australia)...	11:30 am
Athens (Greece) ...	4:00 am
Bankok (Thailand) .	9:00 am
Beijing (China) ...	10:00 am
Berlin (Germany)...	3:00 am
Bombay (India) ...	7:30 am
*Buenos Aires (Argentina)..	11:00 pm
Cairo (Egypt) ...	4:00 am
Calcutta (India)...	7:30 am
Cape Town (Sth. Africa) ...	4:00 am
*Chicago (U.S.A)...	8:00 pm
Copenhagen (Denmark)...	3:00 am
Gibraltar (Island of)...	3:00 am
*Hawaii (USA) ...	4:00 pm
*Havana (Cuba) ...	9:00 pm
Hong Kong (Island of)...	10:00 am
Islamabad (Pakistan)...	7:00 am
Istanbul (Turkey)...	4:00 am
Jakarta (Indonesia) ...	9:00 am
Karachi (Pakistan) ...	7:00 am
Kuala Lumpur (Malaysia) ...	10:00 am
Leningrad (Soviet Union)...	5:00 am
Lisbon (Porugal)...	2:00 am
London (England)...	2:00 am
Madras (India) ...	7:30 am
Malta (Island) ...	3:00 am
Madrid (Spain) ...	3:00 am
Manilla (Philippines)...	10:00 am
Mauritius (Island)...	6:00 am
*Montreal (Canada) ...	9:00 pm
Moscow (Soviet Union)...	5:00 am
*New Orleans (USA) ...	8:00 pm
*New York (USA) ...	9:00 pm
Oslo (Norway) ...	3:00 am
Ottawa (Canada) ...	9:00 pm
Paris (France) ...	3:00 am
Perth (West Australia) ...	10:00 am
*Quebec (Canada)...	9:00 pm
Rangoon (Burma) ...	8:30 pm
Reykjavik (Iceland) ...	1:00 am
*Rio de Janeiro (Brazil)...	11:00 pm
Rome (Italy) ...	3:00 am
Rotterdam (Netherlands) ...	3:00 am
*St. John's N.F.(Canada) ...	10:00 pm
*San Fransisco (USA)...	6:00 pm
Singapore ...	10:00 am
Stockholm (Sweden) ...	3:00 am
Suez (Egypt) ...	4:00 am
Suva (Fiji) ...	2:00 pm
Tehran (Iran) ...	5:30 am
Tokyo (Japan) ...	11:00 pm
*Vancouver (Canada) ...	6:00 am
Vienna (Austria)...	3:00 am
*Washington DC (USA)...	9:00 pm
Wellington (New Zealand)...	2:00 pm
Winnipeg (Canada)...	8:00 pm
Yokohama (Japan)...	11:00 pm

* = previous Day (Adjust for local summertime where applicable)



Reprinted from S.N.U.G. Network

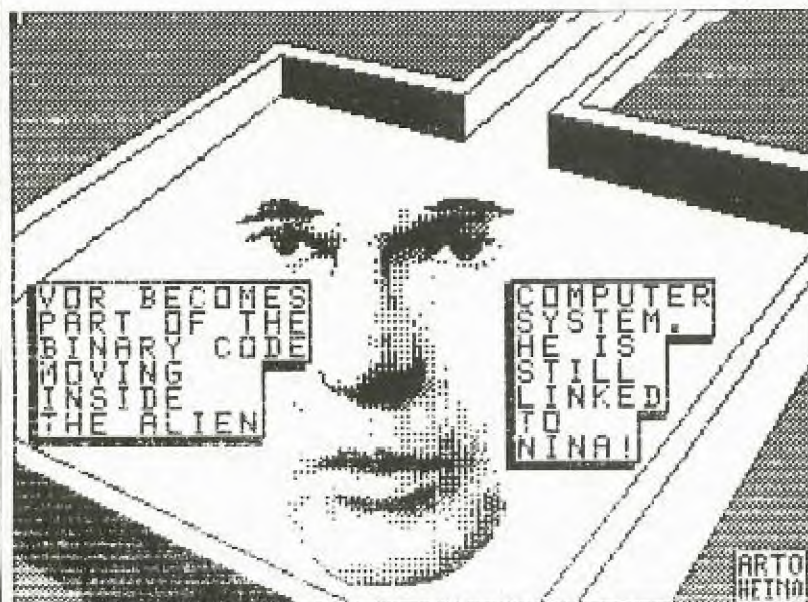
```

*****
* PROGRAM BA10A ==> Basic Assembler # 10 Assembly Vers.*
* DEMONSTRATION OF USING 40 COLUMN TEXT MODE
* (c) 1986 S. PEACOCK
*****

REF VWTR,KSCAN,VSBW * REFERENCES NEEDED IN PROGRAM
DEF START *START OF PROGRAM
START LI R0,>0719 *BLACK TEXT (>1) ON A RED (>9) SCREEN
*****WRITTEN TO REGISTER 7. CHANGE THE 1
*****AND 9 TO ANY COLOR YOU WANT.
BLWP @VWTR *WRITE THE INFORMATION
LI R0,>01F0 *>FO 240d PUT INTO VDP REGISTER 1.
*****SETS TEXT MODE.
BLWP @VWTR *WRITE THE INFORMATION
SWPB R0 *PUT VALUE IN VDP REGISTER 1 INTO
*****THE LEFT BYTE REGISTER 0
MOVB R0,@>83D4 *PUT IT IN >83D4
CLR @>8373 *CLEAR\
CLR R0 *CLEAR =SET UP KEYBOARD SCAN
CLR R1 *CLEAR/
JMP CLSCN *JUMP TO THE CLEAR SCREEN SECTION
LOOP BLWP @KSCAN *BRANCH TO THE KEYBOARD SCAN
MOVB @>837C,R1 *PUT STATUS BYTE IN REGISTER 1
COC @NOKEY,R1 *COMPARE ONE CORRESPONDING=>LEFT BYTE
*****OF REGISTER 1 AND THE VALUE IN NOKEY
LIMI 2 *ENABLE VDP INTERRUPTS
LIMI 0 *DISABLE (FCTN/QUIT WILL WORK)
JNE LOOP *IF NO KEY HAS BEEN PRESSED JUMP TO LOOP
MOV @>8375,R1 *PUT THE ASCII CODE OF THE KEY PRESSED
*****IN REGISTER 1
CI R1,13 *COMPARE IT TO 13 (ENTER KEY)
JEQ CLSCN *IF ENTER PRESSED THEN JUMP TO CLEAR
***** SCREEN
MOVB @>8375,R1 *MOVE THE ASCII CODE OF THE KEY PRESSED
***** INTO THE LEFT BYTE OF REGISTER 1
BLWP @VSBW *PRINT THE LETTER ON THE SCREEN
INC R0 *ADD 1 TO REGISTER 0 (THE PRINTING
***** POSITION)
CI R0,961 *SEE IF THE LAST POSITION HAS BEEN
***** REACHED
JLT LOOP *IF NOT JUMP TO MAIN LOOP
CLR R0 *IF IT HAS RESET PRINT POSITION TO TOP
***** LEFT
JMP LOOP *JUMP TO MAIN LOOP
CLSCN LI R0,0
CLRS LI R1,>2000 *CLEAR SCREEN SECTION
BLWP @VSBW
INC R0 *PRINT A SPACE TO ALL 959 POSITIONS
CI R0,959
JLE CLRS
JMP LOOP
NOKEY DATA >2000 *DATA FOR THE COC
*****IF THE LABEL YOU USE TO START YOUR
*****PROGRAM IS PUT IN THE OPERAND SECTION
*****OF THE END INSTRUCTION, THEN YOUR PROGRAM
*****WILL AUTOSTART
END START

```


THANK YOU PETER, FOR YOUR CONTRIBUTION TO THIS MONTHS YOUNGER SET PAGE, KEEP UP THE GREAT WORK. WE HAVE A LOT OF VERY TALENTED MEMBERS, OF ALL AGES, IN TISHUG, OF WHICH WE ARE VERY PROUD. I ONLY WISH WE HEARD MORE FROM YOU...TODATE, I HAVE NOT RECEIVED ONE ENTRY IN THE ANNUAL CARTOON COMPETITION, WE I'LL HAVE TO DRAW THAT COMPETITION TO A CLOSE. YOUR SUPPORT IN THIS CLUB IS VITAL, IF WE ARE TO REMAIN AS A GROUP, SUPPORTING EACH OTHER, WITHOUT YOUR CONTRIBUTIONS, ALL OF OUR EFFORTS ARE A WASTE OF TIME. AND NOW, LETS SEE HOW 'YOR' IS GOING, IN OUR CONTINUEING SCI-FI TI CARTOON SERIES...



HERE IS A LETTER FROM RICHARD MARTIN (TEXPAC USERNAME: MIAMI)

Dear Jenny,
Here I have tried to give a scattering of hints from 3 Scott Adams adventures. The first bunch are for #2- Pirates Adventure. The second lot are for #3- Mission Impossible. Then, finally, they are for #6- Strange Odyssey. I hope these help any bogged adventurers out there.

1) * The pirate is alcoholic... you can bribe him with booze.

* The pirate loathes the hook... if you have the hook, sailings a no-no!

* Polly likes gold. Polly hates snakes.

2) * A key in the mop- shake it baby!

* Take the saboteurs body and pass to make the visitor room camera mind it's own business. But to break the window, remind your thoughts to what was around at the adventure's start!

* If you can't enter the control room, remember you ballet lessons... 1.. 2.. 3.. KICK!

3) * The ice hound is frigid... it only likes it's own climate!

* Keep treasures on your ship... I mean, who wants to go back to THAT place?

* Appreciate Art! Special art viewing goggles

* Appreciate art! Special art viewing goggles are provided for your enjoyment at our expense in the Hex-Room!

* The hex-room is really an advanced alien time machine (sort of..) pulling and pushing the rod once sets the venue sand touching the plastic makes you travel. The number of times the plastic glove is telling you which location you're at.

Thanks, Richard

Dear Jenny,

Just to add something to Stephen Judd's work in the Christmas issue of TND's Younger Set on Fathom... If you find you always run out of ENERGY, there is a method of getting a good supply. Go to a screen adjacent to the Volcanoe. Now fly to the Volcanoe and collect at least 2 clouds. Go out of the Volcanoe screen briefly and come back in and collect more clouds. As the clouds keep coming back, you can get energy even though it takes a lot of time. If you want to improve your scores, when you have the 3 pieces of TRIDENT, get 950 energies and go to NEPTINA. By the way, If you miss a cloud in the Volcanoe screen, do not try getting it in that screen, as it may take 10 energies to get 5.
Regards, KRESHO SPREM

WELL GANG, THATS IT FOR ME. I HOPE YOU'LL GET ALONG TO THE NEXT MEETING WHICH IS THE ANNUAL GENERAL MEETING (Election time for New Committee etc). This meeting won't be held at Woodstock this month, we're going to be having it at a NEW LOCATION. PLEASE TAKE NOTE OF THIS NEW ADDRESS, AND TELL YOUR PARENTS ... R.S.L. BURWOOD, 96 Shaftsbury Rd, Burwood.

Bye 4 now.
JENNY





TISHUG NEWS DIGEST

Programming Music

Line 370

by Robert Montgomery
Illawarra Regional Group - TISHUG

There are many methods used to program music. Each requires a basic knowledge of music conventions. The style the program has, can either be simple; e.g. a CALL SOUND statement that uses a single frequency that is changed in consecutive statements; to a complex statement that uses a number of programming features.

This article will concentrate on the complex style.

The tune "Peter Gunn", by Henry Mancini, has been programmed using this style and will be critically examined. The full listing is at the end of this paper. Breaking the program into its components reveals the use of strings, loops, SEG\$ statements, arrays, READ...DATA statements and CALL SOUNDS.

The program is written in Extended BASIC.

Program Style

Line 100 - 160

These lines are used to identify the program.

Line 170

An array of 22 members is dimensioned. Each member will have a value corresponding to a certain frequency. The reason for this will be explained when discussing the CALL SOUND statement.

Line 180

A loop used to fill a number of arrays. It is pertinent to note that only the F(I) was dimensioned. The other arrays C(I), T(I), M\$(I), B\$(I) will each have only 10 members. The TI-99/4A automatically allows 11 member arrays without being dimensioned. Because F(I) goes to 22 members, it has to be dimensioned.

The loop is used to READ data into each array. The first five values from the DATA statements:

F(1) becomes 698

C(1) becomes 262

T(1) becomes 200

M\$(1) becomes 1V1V1V1V1V1V1V1V1V1V1V1V1V1V1V1V1V1

B\$(1) becomes DADAEBEAFCEFAHEGDDADAEBEAFCEFAHEGDD

F(I), C(I) is a frequency, T(I) is a time or duration, M\$(I), B\$(I) are strings that are to be broken up for the musical score.

Lines 190 - 195

These lines complete the values in the F(I) array.

Lines 200 - to 350

These DATA statements are used to fill all of the arrays.

Line 360

Two loops are set up. One within the other. The second loop, or nested loop, takes the length of M\$(I) to determine how many times it goes round. The loop is stepped in two's.

This is the line that does all the control of the sound, and is very complex. The CALL SOUND uses a negative time and three frequencies to produce the tune in three part harmony. Because of its complexity it needs to be explained. A minus time signifies to the sound chip to change its frequency immediately another CALL SOUND statement is encountered. The duration is set up as:

-T(ASC(SEG\$(M\$(I),J,1))-48).

To explain this statement, it is necessary to define every thing from the inner-most brackets (I) and move out in pairs.

(I) is the number used for the first loop.

J is the number used in the second or nested loop and shows the starting position of the SEG\$ statement.

M\$(I) is used for the melody line and has the components of duration and melody in it.

SEG\$(M\$(I),J,1) tells the computer to select one character from the character string M\$(I) beginning with the Jth position. SEG\$ stands for "segment" and, of course, stands for a segment of a character string.

If I=1 then M\$(1) is the first character string in the DATA statement. If J=1 then the SEG\$ statement will give the first character of M\$(1) string. That just happens to be 1.

Having sorted that out, what is left is -T(ASC(1)-48).

Now the ASC(1) value is 49; which will give an array of -T(1). T(1) has a value of 200. Because the duration has a minus sign, the value is not significant. Values have been added to this example of programming only to help fully explain the style.

The melody frequency is set up with M\$(2), using every second character. A similar breakdown of the string is employed to create the array member number. The letters of the alphabet are used. An array number is calculated by taking 64 from the ASCII value of the letter. The first frequency encountered is a V. The V has an ASCII value of 86. When 64 is taken from it, it gives a value of 22. The frequency of F(22) is 30000. That is so high it is not normally heard. It, therefore, gives a moment of no sound.

Therefore, the very first M\$(I) will give a period in which no sound will be heard in the melody line.

B\$(I) is used for the base lines. In "Peter Gunn", not only is there a dominating base prevailing through the piece, but there is a secondary counter melody being played at a lower volume.

The counter melody is used in the second frequency using C(I) and a volume of 5. The first character of B\$(1) is D and has an ASCII value of 68; therefore, the counter melody frequency is C(4) or 350.

The base is in the third frequency and uses the F(I) array, with its value divided by 4. This gives the melody and base line a tonal separation of two octaves.

The speed at which the music is played, in this program, is dependent on how long it take the 9900 chip to compute the various statements within the CALL SOUND statement. A trick has been used in each character string M\$() and B\$(). It is that each note played is a factor of the shortest note in the composition. The shortest note played in "Peter Gunn" is a 1/8th note. That is one letter is a single note and will sound for a half-beat; 2 letters will sound for 1 beat etc.

WINDOWS FOR YOUR 99/4A

| by Arto Heino |

When ever you look at other systems (GOD FORBID!) you see they have jumped on the bandwagon and all use WINDOWS!!

Your eyes drool at the MAC when you see those windows dropping and popping up and down and around. The concept is very simple here's some different approaches:

CONCURRENT WINDOWING
MENU WINDOWS
TEXT WINDOWS
GRAPHIC WINDOWS

All these ways of windowing has been done on the 99/4A. Maybe not all together but thats what attracted your attention to the MAC or IBM screen.

The 99/4A has it over those machines because of its SUPERB MINI COMPUTER INSTRUCTION SET!!!

Memory to memory architecture lends itself to windowing quite naturally, just look at TI-WRITER's variation, or GRAPHX even.

Here is a XB version of a MENU window:

```

100 !*****
110 ! XB MENU WINDOW *
120 ! by Arto Heino *
130 ! TISHUG 1987 *
140 !*****
150 CALL CLEAR
160 WOPEN=1 :: WCLOSE=0
170 X1=10 :: X2=21 :: Y1=10 :: Y2=14
180 TEXT$="-----"&"|This is a|"&"|
TI-99/4A |"&"|WINDOW. |"&"-----"
190 !*****
200 !DISPLAY SOME TEXT*
210 !*****
220 RESTORE
230 FOR ROW=1 TO 20
240 READ TXT$

```

```

250 DISPLAY AT(ROW,1):TXT$
260 NEXT ROW
270 !*****
280 !OPEN WINDOW*
290 !*****
300 CALL WINDOW(X1,X2,Y1,Y2,TEXT$,WOPEN)
310 !*****
320 !DELAY LOOP*
330 !*****
340 FOR DELAY=1 TO 1000 :: NEXT DELAY
350 !*****
360 !CLOSE WINDOW*
370 !*****
380 CALL WINDOW(X1,X2,Y1,Y2,TEXT$,WCLOSE)
390 GOTO 390
400 DATA " *****"
410 DATA " * TI-99/4A WINDOWS *"
420 DATA " *****"
430 DATA ""
440 DATA " Study this XB program and"
450 DATA "write a program in assembly"
460 DATA "that does the equavilent"
470 DATA "using a CALL LINK to XB."
480 DATA ""
490 DATA " You could add lots of"
500 DATA "improvements like:"
510 DATA ""
520 DATA " move window"
530 DATA " scroll text in window"
540 DATA " add fancy boarder"
550 DATA " multiple windows"
560 DATA " interrupt driven window"
570 DATA ""
580 DATA " Have a good time with"
590 DATA "your TI-99/4A windows."
600 !*****
601 !WINDOW SUBROUTINE*
602 !*****
610 SUB WINDOW(X1,X2,Y1,Y2,TEXT$,STATUS)
:: Z=0 :: IF USE=1 THEN 630
620 USE=1 :: DIM WIND(768)
630 ON STATUS+1 GOTO 740,640
640 FOR READY=Y1 TO Y2
650 FOR READX=X1 TO X2
660 CALL GCHAR(READY,READX,CHAR)
670 CALL HCHAR(READY,READX,32)
680 Z=Z+1 :: WIND(Z)=CHAR
690 NEXT READX :: NEXT READY
700 FOR Z=Y1 TO Y2
710 DISPLAY AT(Z,X1-2):SEG$(TEXT$,1+
(Z-Y1)*(X2-X1),X2-X1);
720 NEXT Z
730 GOTO 780
740 FOR WRITEY=Y1 TO Y2
750 FOR WRITEX=X1 TO X2
760 Z=Z+1 :: CALL HCHAR(WRITEY,WRITEX,WIND(Z))
770 NEXT WRITEX :: NEXT WRITEY
780 SUBEND

```

MIDI, MUSIC and the TI-99/4A

| by Arto Heino |

Your friends say "WOW you should have heard the live concert last night it was a BLAST!!"

It probably was a blast because of MIDI. I hear you say what the heck is MIDI?? Well to make it short it is a communications protocol and has its own interface(similar to RS232). But instead of interfacing to PRINTERS etc.. it is attached to SYNTHESIZERS, KEYBOARDS, DRUM MACHINES..etc.

The computer becomes the master controller for up to 16 devices and each channel can have 128 notes!!

16*128=2048 now that is a lot of notes, wait thats not all. Each note has its own set of parameters eg. ATTACK, DECAY, PRESSURE, VOLUME, POLY..etc Each of those has 128 selections!!

You can send and receive info from MIDI using a common 2 wire serial data link, transmitted at 31.25 Kbaud rate via a 5 mA current loop.

A MIDI word is composed of 3 MIDI bytes. A MIDI byte is 10 bits wide, eight of these are data bits, one is a start bit and one is a stop bit.

To use MIDI on a 99/4A you must have a terminal program with options for changing musical parameters also a DISK drive for data storage such as compositions or sound structures. Also a hardware interface either to the RS232 or the COMPUTER BUS. If a card was designed it would have its own DSR and it would mean you could control MIDI connected devices from BASIC.

I am presently working on bulding a hardware device for the RS232. The terminal program is at its early stages of development and will be ground tested with a terrific local band "THIN LINE".

Record Length Encoded files

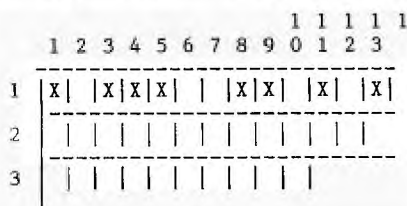
| by Arto Heino |

Nothing is ever wasted if you use your ingenuity. RLE files give you that because of its universal format.

A RLE file consists of ASCII chars from 32 to 127. This means any computer can read them this also means that you can too(with perseverance).

The 99/4A screen is 256*192 in size, RLE looks at it as whats on or whats off and how many times it is on or off regardless of size. This means in short that ALL your graphic files can be used with any computer.(80 column card !!)

eg.. X=on " "=off



This diagram is of the screen area zoomed to pixel size. Row 1 has some pixels on/off, this is what RLE sees it as:

!!#""!!!!

Here's a break down:

- ! = ASC 32 +1 ON
- ! = ASC 32 +1 OFF
- # = ASC 32 +3 ON
- " = ASC 32 +2 OFF

- " = ASC 32 +2 ON
- ! = ASC 32 +1 OFF
- ! = ASC 32 +1 ON
- ! = ASC 32 +1 OFF
- ! = ASC 32 +1 ON

Now your thinking how does work if you have pixels on or off greater than 95 ? Well thats what the ASC 32 does it tells that the next pixel is off or on. Heres 2 examples that are the same:

- 1) ! A 0
- ! = ASC 32 +1 ON
 - = ASC 32 +0 OFF
 - A = ASC 32 +33 ON
 - = ASC 32 +0 OFF
 - 0 = ASC 32 +16 ON
- 2) R
- R = ASC 32 +50 ON

RLE reads the screen from left to right and top to bottom. Say that you have a TI professional and you want to transfere a graphic file to your 99/4A.

Your first job is to define the screen area on the TI/Pro that you want to transfere in 99/4A boundaries(256*192). Run your RLE writer program, which reads the pixels if they are on/off in those boundaries and creates a ASCII file which you can then send to the 99/4A via the RS232.

The best ideas are usually the simplest. The RLE concept will surely put an end to COMPUTER OBSELECNCE!!!

SUBROUTINE

```

100 ! **TWO LINE MAZAMYND**
    **BY DENNIS HODGSON**
    PENRITH REGIONAL GROUP
110 ! SELECT LEVEL UP TO 9
    TRY 4. THE COMPUTER SELECTS
    A NUMBER AT RANDOM OF LENGTH
    TO THE LEVEL CHOSEN.
120 ! ALL DIGITS IN THE
    MYSTERY NUMBER ARE DIFFERENT
    YOU ENTER YOUR GUESSES AT
    THE BEEP.THE COMPUTER SHOWS
130 ! HOW MANY DIGITS ARE
    IN THE RIGHT PLACE(RP) AND
    WRONG PLACE(WP) AND NUMBER
    OF TRIES(#). IF YOU ENTER
    THE SAVE DIGIT MORE THAN
140 !ONCE THE CLEVER
    COMPUTER SCORES IF ANY ARE
    CORRECTLY PLACED AND IGNORES
    THE OTHERS. IF NONE ARE
    CORRECTLY PLACED ONLY 1(1)
150 ! WRONG PLACE WILL
    REGISTER IF APPROPRIATE. YOU
    MAY ENTER LESS THAN THE
    FULL NUMBER OF DIGITS ON ANY
    TURN.
  
```

```

160 DISPLAY ERASE ALL:**MAZD
    AMYND**,"RP WP #" :: INPUT "
    LEV:"L :: RANDOMIZE :: Q=1
    :: FOR T=Q TO L :: C(T)=INT(
    RND*10):: FOR J=Q TO T-Q ::
    T=T+(C(T)=C(J)):: NEXT J ::
    NEXT T :: FOR T=Q TO L :: U$
    (T)=STR$(C(T)):: C$=C$&U$(T)
    :: NEXT T :: FOR
    R=Z TO Z :: M=M+Q :: ACCEPT
    BEEP SIZE(L):G$
170 FOR T=Q TO L :: S$(T)=SE
    G$(G$,T,Q):: A(T)=S$(T)=U$(T)
    ): P=P-A(T):: NEXT T :: FOR
    T=Q TO L :: FOR J=Q TO L ::
    W=W-(S$(T)=U$(J)AND A(J)=Z
    AND POS(A$,S$(T),Q)=Z):: NEX
    T J :: A$=A$&S$(T):: NEXT T
    :: PRINT ,P;W;M :: A$="" ::
    P,W=Z :: R=G$<C
    $ :: NEXT R
  
```

TIPS FROM THE TIGERCUB.

Jim Perterson.
Tigercub Software.

This routine will read a file of 28 character records and scroll them up the lower half of the screen without disturbing the upper half.

```

100 DISPLAY AT(12,1)ERASE AL
    L:"FILENAME? DSK" :: ACCEPT
    AT(12,14)BEEP:F$ :: CALL CLE
    AR
111 OPEN #1:"DSK"&F$,INPUT
112 DIM M$(480)
113 X=X+1 :: LINPUT #1:M$(X)
120 DISPLAY AT(24,1):M$(X)
125 R=24
130 FOR T=X-1 TO 1 STEP -1 :
    : IF R>13 THEN R=R-1 :: DISP
    LAY AT(R,1):M$(T)
140 NEXT T :: IF EOF(1)<>1 T
    HEN 113 ELSE CLOSE #1
  
```

```

10 !ONE-LINE MORTGAGE PAYMEN
    T CALCULATOR BY SAM MORABITO
100 CALL CLEAR :: INPUT "ENT
    ER P,R,N WHERE P=AMOUNT, R=F
    ATE, N=YEARS":P,R,N :: PRINT
    "$";INT((P*R/1200)/(1-1/(1+
    R/1200))^(N*12))*100+.5/100.
    "PER MONTH"
  
```



Welcome back and I trust that you all had a nice Christmas and that 1987 is going as planned so far.

In case you missed out on any of the software released during 1986 here's what was issued. If you are missing any write to the Shop to place your order. A full description is in each issue of the TND.

On Disk:	On Tape:
TI Writer Tutorial	1986/1
Disk Tutorial	1986/2
TI99-Opoly	1986/3
Mudie 86/1	1986/4
Mudie 86/2	1986/5
Funlwriter	Assembly #1
Fast Term	1986/6
Trivia 99'er	1986/7
"C" Language	1986/8
Mudie 86/3	1986/9
Universal Disass.	1986/9A
MS Adventures	1986/10
Funlwriter 3.3	1986/10A
Sorgan & Siegfried	1986/11
Utilities 1	1986/11A
Utilities 2	1986/12
BA Writer	1986/12A
PR Base	1986/12B
Utilities 3	Assembly #2
RLE Pictures #1	Mike Slattery Word Processor
Games/Music #1	Character Definition Utility
Games/Music #2	
November 1986	
Mudie 86/5	
Hymns	
Disk 1986/11	
Happy Holidays	
Snoopy Calendar	
RLE Pictures #2	
RLE Pictures #3 ("R")	
Mark Beck CFS	
Casino Games	
Sideways Print	
December 1986	
Surprise	

OK. So that's what we had last year. Now for this year. First up to be released on disk and available at the February meeting.

1. THE CHECKBOOK AND BUDGET MANAGER - a Fairware offering from the prolific software author John Taylor. Heaps of documentation files on the disk should enable you to get right into it once you have read them.
2. JET DISK 01B - also from John Taylor. The disk contains two of the best education programs I have seen in a long time. Alphanum Delight and Out on a Limb. The first is a letter and number recognition game, written with the pre-schooler in mind, however at advanced levels it can even be a challenging game for adults. Graphics and sound effects are excellent. The second program, Out on a Limb, is a "Hangman" type of game but with a big difference. Again Graphics and sound effects make this a very enjoyable educational treat.
3. A DISK OF CALENDARS - a Fairware offering from Richard Bailey. All types of calendars, including an appointment calendar, a big calendar, small calendar, tiny calendar and a year calendar. Just about a calendar to suit every taste.
4. MORE SURPRISES - following on from the Surprise disk released in December here's some more for you to enjoy.
5. PILOT LANGUAGE DISK - after much testing it appears that we have got a good copy of this exciting language. This will be released on a floppy with the system files on side 1 and the documentation on side 2. Be patient when printing out the docs as there are many pages.

On cassette this month will be the following:-

TAPE 1987/1

The great educational programs mentioned above, Alphanum Delight and Out on a Limb. These are cassette versions and need nothing more than Extended Basic for you to run them. Here's a brief description of what you get:

ALPHANUM DELIGHT DOCUMENTATION - tells you all about the game you are about to play.

ALPHANUM DELIGHT - the game itself.

OUT ON A LIMB DOCUMENTATION - again to tell you all about the game.

OUT ON A LIMB - the game itself.

OUT ON A LIMB DATA FILE - a sample data file containing about 75 words. Easy to load, just follow instructions from the main game.

OUT ON A LIMB DATA BUILDER - this will allow you to customise the game to suit yourself. Once you have typed in the words you want you can save them back to tape (CS1).

GOLF - in Basic. This is a good sports simulation with your choice of using either woods or irons as you hack your way around the fairway. Instructions are included in the game. Simple to play with just key inputs required.

LANDING - in Extended Basic. This is a well written game the object being to land your craft on its base. Easy enough on the first level but it gets harder as the game progresses. Instructions included in the program.

THE DRAGON'S LAIR - in Extended Basic. This is a solo adventure game with your mission being to slay the dragon. A whole host of meanies try to prevent you from doing this. Good Luck! You'll need it to survive.

ATTACKER - again in Extended Basic with 32K expansion. This is a professionally written space game of the invaders type. It is not all that easy to master so it should keep you occupied for some time.

TAPE 1987/2

The same programs as on the Surprise disk also mentioned above. You will need 32K memory Expansion and Extended Basic for this one.

ON TAPE AND DISK - some more data bases for the Tunnels of Doom module. Titles are Tomb of Death, Volcano Fortress and one called FLOOR10 which is actually one of my efforts in playing the game. Those who haven't got this far might like it. Your character is a Hero and is called TERRY (The Hero). PS. If you do play FLOOR10, watch out for the dragons, for they're pretty mean!





As we start another year for the club I would like to particularly thank all of you who supported the shop and made our high turnover possible.

I hope those of you who visited the shop during our monthly meetings will bear with me while I review the past 12 months.

Our expansion began in our joint venture with Computer Resources Company. By special arrangement, we were able to offer supplies (particularly paper and disks) at very attractive prices which were FOB Rail or free delivery in capital city metropolitan areas. I am surprised this service has not been used more widely - particularly by country and interstate members. This service continues to operate.

From that "testing of the waters" we progressed to supplying directly, disk drives, printers, disk storage boxes, consoles and of course our low-priced floppy disks. Our importing activity was also expanded to include a selection of utility/productivity type software including Multiplan, Logo II, SST Expanded Basic Compiler, MiniMemory Modules, and the Auto-Spell Checker for TI Writer files.

One criticism sometimes made of TISHUG is that we are too commercial. As your retiring Merchandise Co-ordinator, I have two comments: (1) if you weren't buying we wouldn't be selling; and (2) the proposition has much wider and deeper implications which need to be fully explored and considered by the incoming executive.

With the AGM and forthcoming election of office bearers (directors) you have the opportunity to determine the course of shop policy for the next 12 months.

The past 12 months has seen some major changes in the TI market world wide. Some changes seem to be the result of increasing polarisation around computer types. In the USA, Emerald Valley Publishing (99er & HCM) broke away from the 99/4A and is now out of business - presumably a failure at attempting to be all things to all computers: Unisource Electronics pulled out of the home computer market completely; Tenex Computer Express launched into IBM compatible machines; Millers Graphics launched its Gram Cracker (now going out of production) while Corcomp and Myarc both persevered with hardware and software development. In Germany, Mechatronics embarked upon hardware and software development, independently of and on a broader scale than that undertaken in the USA. Closer to home (and perhaps more importantly) we saw the 3.3 release of Tony McGovern's Funlwriter disk, Geoff Trott's plug-in console diagnostic card and John Paine's development of a GPL loader.

The significance of all this is a little clouded but here is a very personal assessment. The 99/4A community is still alive and continues to be served by smaller and more specialised (perhaps more dedicated) organisations whose prime objective is to support the 99/4A computer. In Australia, we have software and hardware Guru's who can mix it with the best from overseas and lastly, currency exchange rates mean we will continue to pay a heavy penalty for imported products/technology. If the right incentive "mix" is available, I feel sure we will see bigger and better things from our local TI community in the coming year.

One aspect you must bear in mind when dealing with the shop is that while we don't claim to have the lowest priced product, we do strive to offer the best value for money. Our disk drives for example were Chinon or Epson brand with low power consumption (suitable for mounting 2 in the PE Box without power supply modification) and the printers, Epson units. I am not aware of any dealers who were able to beat our prices on the same product.

There is a very selfish reason for our insistence on quality - we don't want to be involved unnecessarily with "faulty" returns or warranty claims. Also, there's nothing like an Epson for Epson compatibility.

Now the hard sell:-

Multiplan sets complete	\$65.00
Auto-Spell Checker	\$45.00
TI Intern Manuals	\$22.00
Disk Storage Boxes	\$18.60
DS/DD disks (Box 10)	\$19.00
Consoles (Ver. 2.2)	\$65.00
Power Supplies (console)	\$15.00
Micropendiums (December)	\$ 2.90

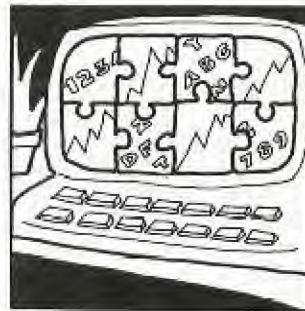
Verbatim and Memorex Head Cleaning Kits and of course monthly tape and disk software releases.

We have sufficient people interested in the 32K in-console memory expansion to justify another "field day". If you are interested in adding the 32K, please send your deposit ASAP to reserve your kit.

HELP: Someone recently returned two tapes: 1986/11 and 1986/12B. The postpak envelope has no sender details and there was no note enclosed. Please let me know who you are and what you require.

Thank you for your past support. Special thanks to Ray, Robert and Cyril who gave their time so willingly.

Chris.



Well that looks a pretty good selection to start the year of with. I hope you like them.

It has been my pleasure to serve your software needs throughout the past 12 months. During this period contact has been made with numerous User Groups in an attempt to build up our software library. Some contacts have been successful while others not so. Of special note has been the exchange of programs with Jim Peterson. The library has grown tremendously in the past 12 months and most of the software received, particularly the Freeware/Shareware has been made available to the membership. There is still plenty of good software around for out TI and every effort is being made to obtain copies and pass them on.

TECHNO TIME



INTERFACING THE LX86 PRINTER

The Epson LX86 printer sold by the club shop can be easily interfaced to the TI RS232 card via the parallel or serial ports.

To make the connection on the parallel (CENTRONICS) port, make your connections as follows:-

PARALLEL PORT RS232 CARD	PARALLEL PORT LX86 PRINTER
PIN #	PIN #
1-----	1
2-----	2
3-----	3
4-----	4
5-----	5
6-----	6
7-----	7
8-----	8
9-----	9
10-----	11
16-----	16

With the above connections, your printer should work perfectly in parallel.

The Serial connection is more complicated. You must first obtain a Serial Interface to plug into your printer. Models are available ranging from a no frills version to an intelligent unit with data buffer.

This article deals only with the basic serial interface (Epson #8143). To make your cable, wire it as follows:-

SERIAL PORT 1 RS232 CARD	SERIAL PORT LX86 PRINTER
PIN #	PIN #
1-----	1
2-----	2
3-----	3
8-----	6
7-----	7
20-----	11

You must now change the DIP Switch Settings to suit the transmission characteristics of your serial port. I have field tested the interface at 9,600 Baud, EVEN Parity, 7 Data Bits and it works faultlessly so if you wish to follow suit, here are the switch positions:-

DIP SWITCH PIN No.	CONDITION OFF/ON
1-1 (JB3)-----	ON
1-2 (J8/7)-----	ON
1-3 (JB1)-----	OFF
1-4 (JB2)-----	OFF
1-5 (JO/E)-----	ON
1-6 (JPDS)-----	ON
1-7 (JB4)-----	OFF
1-8 (P/S)-----	ON

If you want some other characteristics, you will have to refer to pages 5/6 of the interface manual for the switch settings.

The second RS232 port can be used by making the following substitutions:-

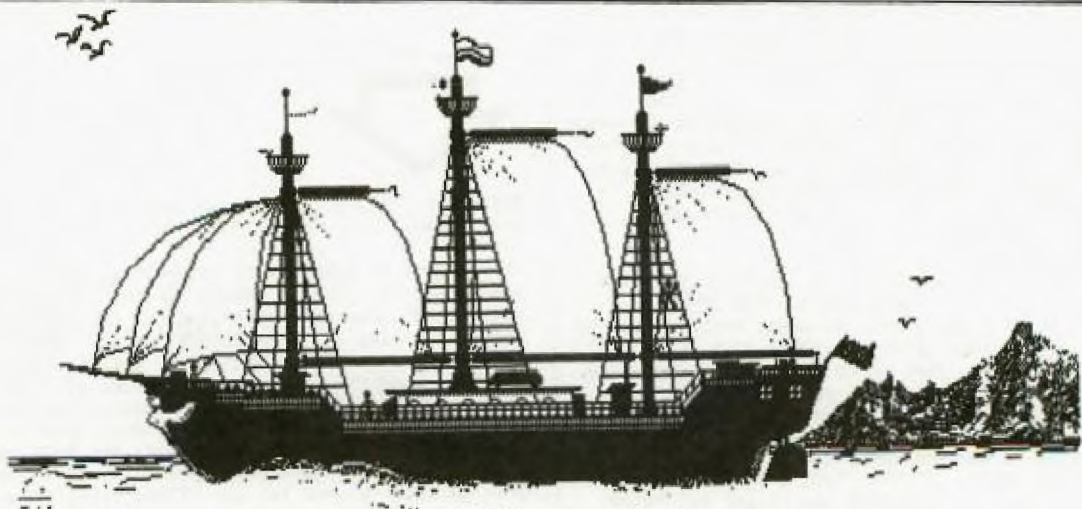
RS232/1 PIN No.	RS232/2 PIN No.
2-----	14
3-----	16
8-----	12
20-----	19

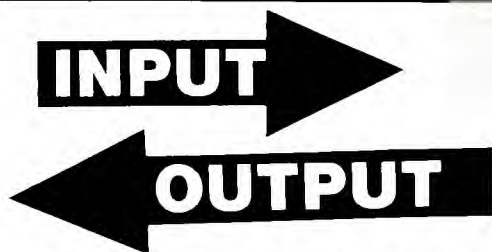
Pins 1 and 7 remain unchanged.

The one remaining task is to correctly set the DIP switches on the rear of the printer. As a starting point, I suggest the following:-

SWITCH 1	SWITCH 2
1-1 OFF	2-1 ON
1-2 OFF	2-2 ON
1-3 OFF	2-3 OFF
1-4 OFF	2-4 OFF
1-5 OFF	
1-6 ON	
1-7 ON	
1-8 ON	

If everything has been done correctly you are now ready for crisp clean printing.





It must be apparent to all that a User's group for a computer which is no longer made has a very uncertain future. This would suggest that it is not a major concern of such a group to make money for a rainy day. Indeed, the question of what happens to the assets of such a group when it ceases to exist should be examined carefully. Let us suggest some ideas about our future which have been talked about in our Regional Group, and which every member may find interesting to think about before the new committee is elected.

There are two things which will keep members in our group: the ability to update their hardware at a reasonable cost, and the knowledge that if something goes wrong it can be fixed; and accessibility to new software of all types.

The first of these is being addressed reasonably well with cheap memory expansion, and mini-peripheral boxes, apart from systems coming on the market second hand. The advent of the Console Tester allows consoles to be fixed in an hour, and could be used to bring new members into the Group. If all the Computer shops, particularly those who used to sell the TI99/4A, were told that consoles could be fixed by TISHUG, then non-members would be brought in touch with the Group. Further, the cost of repair could include a years membership to TISHUG and we may be able to keep some of these members.

The software is more difficult on the surface, as it requires people to write the software, and as the number of users dwindle, it becomes less of a commercial proposition to do so. However, there is a large amount of software which has been written, and which most members have not seen. Most of this software is in the club library already, which in practice is too difficult to get at. It would seem to us that if the library was split up amongst the Regional Groups and rotated periodically, then all members would have a better chance of getting at the software they were interested in. The Regional Groups could be asked to document as many of the programmes as they can in the time they had them, so that eventually the entire collection could be fully documented. The arrival of a fraction of the library would cause great excitement at the Regional Group, and help maintain membership.

What to do about all the money? Firstly, there should be a policy of slowly redistributing the money back to the members. The Dues should be lowered, and there should be no annual charges for use of the library or the BBS. Charges for software should only be the cost of the medium, and this should be at cost or slightly below that. Products for the TI will not be made for too much longer, and some of these should be bought immediately so that some money is converted into goods. For example, the GRAM Kracker is no longer in production, I believe. The group should try to buy as many of these as it can. Similarly, the group should buy useful modules and especially good software like all the Freeware that is around at the moment. It would seem to be far better to have all our assets in goods which can be used by our computer, rather than money, with nothing left on the market to buy. The other possibility is to help with the production costs of useful hardware like RAM disk cards, PIO cards. Even if TISHUG made a loss on these things, if it kept members in the group it would be worth it.

Illawarra Regional Group

EDITORS NOTE: This letter from the Illawarra Regional Group is reproduced in full for the information of members.

Comments like these are always welcome from members. After all this is your magazine where you get to say what you want.

I have now been a member of TISHUG for the past 12 months and decided to write to you to let you know of my thoughts of the Group in general.

First I must say that I have been very impressed with what I have seen and really appreciate the organisation which goes on behind the scenes. This of course includes preparation of the TND, the efforts of those in the Shop, the Software and Publications libraries and the myriad of other planning activities that are necessary to run an organisation such as TISHUG. And all this for such a low membership fee. Value indeed!

I have also met many of the members - those who attend meetings at least - and through these contacts I have learned a lot about my TI. It is friendship, and the caring attitude of members which makes TISHUG such a great group to belong to, and I have been a member of many organisations over the years, but none as well managed, with such friendly members as this one.

Lastly, I believe that as long as there are caring individuals, willing to do that little bit extra, then TISHUG has a bright future. It's all there, talent, helpfulness and that great kindred spirit.

GO FORWARD TISHUG!

Yours sincerely

A Contented Member

EDITORS NOTE: "Contented Members" name and address have been withheld at the writers request.

CLASSIFIED ADS FOR SALE

TI99/4A with unused PE Box, Speech Synthesizer, Joysticks, 8 Cartridges, Extended Basic, lots lots of books ... Phone Tony (02) 888.5937

RS232 Card Phone (02) 727.2008

Hardware Gallery

Diagnosing hardware malfunctions

by Geoff Trott
Illawarra Regional Group - TISHUG

Background

The TI99/4A is quite a sophisticated computer which relies on a number of parts for its correct operation. All microprocessors require a programme in ROM to be accessible upon power up. This is called the monitor programme and for most computers is the only programme which runs the computer until something like BASIC is started. This monitor programme is written in assembler and is run directly by the processor. In the TI99/4A there is a monitor programme in ROM but its main function is to provide an interpreter for another language called GPL (Graphics Programming Language), which is the language in which the operating system of the computer is written. This GPL interpreter expects to find the GPL commands in another sort of read only memory called GROM. The contents of GROM can only be read in such a way that a GROM cannot be used to store assembler language which is to be executed directly by the processor from that GROM. Thus for the correct operation of the computer, both ROM and GROM must be operating correctly.

The 9900 processor requires some RAM for its registers and other system constants. This is called System RAM and 256 bytes are provided in the TI99/4A console. For most of the other storage requirements of the computer VDP RAM is used. This is 16K of dynamic RAM which also is used by the Video Processor chip to contain the information required for the screen display. Once again this memory cannot be used to store assembler language programmes to be executed directly by the processor, whereas the System RAM can be.

If there is a problem with one of the major parts of the console it is quite difficult to determine which one is at fault because of all the interactions between them. If the problem is only in VDP RAM the computer will usually start up and produce a recognisable title screen. This is because the VDP RAM is made up of 8 ICs, one for each bit in the byte. If one IC is faulty it only affects 1 bit of each byte so there are 7 bits correct and a recognisable screen results. If there is a black screen on start up however, the problem could be due to any one of the following being faulty:-

- Video Modulator
- Video Processor
- CPU - TMS9900
- System ROM
- System RAM
- GROM

A large number of other components and ICs

For these reasons it was considered necessary to have a way of determining which parts of the computer were working, and even to pinpoint the actual faulty part. The easiest way to do this is by using the computer itself, but if it was not even giving a title screen this would seem to be impossible.

The LOAD Interrupt

The solution to this dilemma lies in using the LOAD interrupt to start another programme running in hardware external to the console, but using the processor in the console, to check out the System RAM, VDP RAM and hence VDP processor, System ROM and GROM. This must be done without relying on the screen display, but using it if it is working to give more information than would be otherwise be possible. What is this LOAD interrupt?

All microprocessors have a RESET input for power up or panic restarts. The TMS9900 has a RESET which is used for this purpose on power up and whenever a cartridge is pushed into the cartridge port. RESET causes the processor to do an interrupt sequence through addresses 0 to 3 and thus to enter the System ROM and produce the title screen. The TMS9900 has another interrupt input like RESET called LOAD, not normally used in the 99/4A, which causes the processor to do an interrupt sequence through addresses 65532 to 65535, at the top of memory expansion. This LOAD signal is very like the non-maskable interrupt of other processors. LOAD is not very useful normally as one cannot rely on a programme and its vectors to be present in these locations of expansion RAM. However, if a diagnostic programme is put into EPROM with the vectors at these addresses, and some RAM was made available also, then the LOAD signal could be used to start this programme executing regardless of the state of all but the processor in the console. All that would then be necessary would be some indicators to show any errors found, in case the screen display does not work.

The Hardware

The hardware is quite simple, consisting of an EPROM containing the program and the vectors and occupying the last 8K of the expansion memory address space, a RAM chip in the next to last 8K of memory (up to 8K bytes in size), a push button and circuitry for the LOAD signal, address decoding for the EPROM and RAM, and an 8 bit latch which is enabled by a write to an EPROM address. The output of six of the bits of the latch are connected to 3 red LEDs and 3 green LEDs. There is a 44 way edge connector on the printed circuit board and this plugs into the I/O port on the console, and uses the 5 volt supply from the console for power to the board. If a console was in trouble, any internal memory expansion would need to be removed before this device is attached to ensure no address conflicts.

Operational Procedure

The diagnostic board is plugged into the I/O port of a "dead" console and the power turned on. The LOAD button is pressed. This starts the diagnostic programme, and if the processor is working the 3 red LEDs turn on. The first one starts blinking to show that the System RAM is being tested. This is done by writing a pattern into the entire memory and then reading the entire memory checking for any errors. After doing this 100 times, if there are no errors the first green LED turns on, and the blinking LED stays off. If there are errors, the green LED stays off and the formerly blinking red LED remains on.

A similar test is then run on the VDP RAM 21 times while the middle red LED blinks. Since the VDP RAM is attached to the Video processor, some of its functions are also checked. At the end of this test the middle green LED will come on if there are no errors while the red one will remain on if there are errors.

If all is OK so far, the programme sets up the VDP RAM with a character set for 256 characters using the 128 TI-Writer characters repeated once with a red background colour for those codes between 128 and 255. All the characters appear on the bottom of the screen in reverse numerical order, taking up the last 8 lines of the screen. Then the rest of the screen appears with a heading in the first two lines followed by diagnostic information. There is a message about the

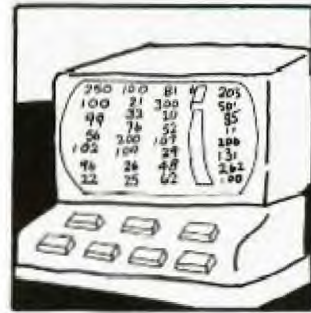




LIBRARY

TND INDEX

Brian Graham
TI-PUBLIB Librarian.



As part of my job as Publication Librarian I have just completed a complete INDEX to ALL articles in ALL editions of both the SND and the TND from APRIL 1981 to DECEMBER 1986.

The Index is compiled on DBASEIII - an IBM program - for ease of operation and is divided into two parts.

Part One covers all articles only as follows :

No. of data records : 1443
Date of last update : 11th. January, 1987.
Size of File : 108484 bytes
No. of Index files : 2
No. of Report files : 2
Total Disk space used : 350Kb.

FIELD	FIELD NAME	TYPE	WIDTH
1	ARTICLE	CHARACTER	20
2	AUTHOR	CHARACTER	13
3	INITIALS	CHARACTER	1
4	DETAILS	CHARACTER	24
5	SND-DATE	DATE	8
6	SND-VOLUME	CHARACTER	6
7	PAGE	NUMERIC	2

Total page width is 75 characters.

The file is also available as an ASCII text file and I am currently seeing if it can be used with a TI Data Base Management System or some other TI program. I will let everyone know how I go.

The second index is of PROGRAMS that have appeared over that same period. Details for this Index as follows :

No. of data records : 349
Date of last update : 11th. January, 1987.
Size of File : 28000 bytes
No. of Index files : 2
No. of Report Files : 2
Total Disk space used : 236Kb.

FIELD	FIELD NAME	TYPE	WIDTH
1	ARTICLE	CHARACTER	20
2	AUTHOR	CHARACTER	13
3	INITIALS	CHARACTER	1
4	DETAILS	CHARACTER	24
5	LANGUAGE	CHARACTER	6
6	SND-DATE	DATE	8
7	SND-VOLUME	CHARACTER	6
8	PAGE	NUMERIC	2

Total Page Width is 78 Characters.

In the meantime I am looking at making it available as hard copy through the Club shop on an order basis. The cost for the report sorted by either author or article will be about \$ 5.00 for Article Index and \$ 1.00 for Program Index to cover cost of photocopying. Both index types will be available on order only basis for delivery at next meeting or postage \$ 1.00 extra.

ORDER YOUR COPY NOW BEFORE IT IS TOO LATE.



System RAM and another about the VDP RAM, which will mirror the state of the first 2 pairs of LEDs. Then the checksum of the System ROMs and the 3 GROMs in the console and 2 GROMs which may be in a cartridge in the cartridge port are calculated and displayed as they are calculated. They are displayed using the full character set and so the actual value can be determined if required. The programme loops around calculating the checksums indefinitely, with the third red LED blinking as it does so. Interrupts are enabled at the end of the loop which means that if the console is working, QUIT will return to the title screen at this time. If the System ROM is faulty the diagnostic may not loop, as the interrupt service routine is in System ROM.

The Checksums

The checksums are calculated by adding the bytes of whichever ROM is being checked. In the case of the System ROMs there are two of them, one for the High or even address byte, and the other for the Low or odd address byte. The checksum is calculated for each of these ROMs and displayed side by side, using 3 characters for each. For the GROMs the bytes of each are added together to produce the checksum. Each checksum requires 3 bytes and is displayed as 3 characters using the character set displayed on the bottom of the screen. The values of the checksums of some ROMs and GROMs are:

System ROM High byte	>048181	
System ROM Low byte	>05FC8C	
System		GROM
0	>075574 (1981)	>0731D7 (V2.2)
	System GROM 1	>091B99 (BASIC)
	System GROM 2	>089ADC (BASIC)

Results

The tester, even in this very simple form, has proved invaluable for fixing consoles. It is very spectacular for consoles with a faulty System ROM, of which there have been many, as at the end of the RAM checking the screen comes to life and the checksums show exactly which chip is faulty. Even in other not so clear cases it has been able to save much time by directing attention to the problem area. There has been one case of an OR gate failing to provide the write select signal for the video processor, and one of an address buffer chip not working. Both of these had similar symptoms to a bad System ROM but behaved differently when executing the diagnostic. Finding these problems took the use of a CRO and a few minutes extra, with the search area narrowed considerably.

If you think you have a need for such a device I am willing to provide them for a cost of around \$100, as long as the demand does not become too great. At the moment, there are two in Sydney and one in Perth to satisfy requirements in those areas, as well as the original one in the "Wonderful City of Wollongong" on the Leisure Coast of Australia.





FREWARE VERSION of TI99-OPOLY. .UPDATE1-6
from Ross Mudie, 29th December 1986.

The freeware version of TI99-OPOLY has been raised from version 1.4 to version 1.5 to overcome a bug which charged rent for an already mortgaged station when a player is advanced to the station by a Chance card card "advance to the nearest railway" and to version 1.6 to overcome rent being charged when advanced to the nearest utility by a Chance card when the utility is already mortgaged.

This update notice includes both the version 1.5 and version 1.6 update information.

To raise a copy of the program disk to V1.6 first make a backup copy of the disk in case of mishap.

Using a Disk Manager, remove write protection from the programs named LOAD and TI99-OPOLY.

Go to extended basic with no disk in drive I. When * READY * and the cursor prompt is visible, place the TI99-OPOLY disk in drive I and type OLD DSK1.LOAD .

Modify the LOAD program, when in memory, in lines 120 and 1090 as follows:

```
120 ! TI99-OPOLY V1.5 LOAD
29th December 1986, Ross M
udie.
```

```
1090 DISPLAY AT(1,5):"LOADIN
G AND RUNNING": "TI99-OPOLY
V1.6 - Ross Mudie"
```

Resave the LOAD program by typing SAVE DSK1.LOAD .

Place the TI99-OPOLY program in memory by typing:
OLD DSK1.TI99-OPOLY .

Modify the TI99-OPOLY program in line 100 and add lines 1965 & 2045 as follows:

```
100 OPTION BASE 1 :: ON WARN
ING NEXT :: ON BREAK NEXT !
TI99-OPOLY V1.6 Ross Mudie
29th December 1986
```

```
1965 IF D(PSN(P),2)=6 THEN C
ALL D(14,B$(123)):: CALL D(1
5,B$(124)):: GOTO 2170
```

```
2045 IF D(PSN(P),2)=6 THEN C
ALL D(14,B$(123)):: CALL D(1
5,B$(124)):: GOTO 2170
```

Do NOT resequence the program.

Resave the TI99-OPOLY program to disk by **typing**:
SAVE DSK1.TI99-OPOLY .

Return to the Disk Manager and re-apply write protection to both programs.

UPDATING AN OLD VERSION OF TI99-OPOLY.

Any person who owns an original copy of the PRE-FREWARE version of TI99-OPOLY may return the original disk and booklet, forward postage paid. The disk and booklet will be updated to the latest freeware version at no further cost and the return postage will be paid by the author.

After enclosing the return disk and booklet in adequate packaging and including YOUR OWN NAME AND ADDRESS, post to: Ross Mudie,
47 Berowra Waters Rd,
Berowra. N.S.W. 2081.
AUSTRALIA.

REGIONAL REPORT.

CARLINGFORD REGIONAL GROUP

The first meeting for 1986 will be on 18 February at 79 JENKINS ROAD, CARLINGFORD (Ph. 871-7753).

Please remember to bring along a print out of your Assembly Language "homework" if you attended the A/L Tutorial with Shane Ferrett at our last meeting for 1986.

All welcome.

The Illawarra Regional Group

This group holds regular meetings on the third Monday of each month (except January) at 7.30 p.m.

PLEASE NOTE: From February 1987 we will be meeting at the Keiraville Public School, Gipps Road, Keiraville, opposite the Keiraville shopping centre.

We also hold occasional hardware and other special interest group meetings at irregular intervals. We are offering memory expansion and other simple hardware expansions upon request, and are working on software for systems without disks but with memory expansion. We also offer basic service facilities for members.

The meetings normally start with a tutorial session on Extended BASIC, followed by a talk and demonstration of some other topic of interest. This leads to some refreshments while members meet each other and chat about problems and interests. We maintain various libraries for the use of members.

REGIONAL GROUP REPORT - BANANA REGIONAL GROUP

The second meeting of the Banana Regional Group was held at the home of Keir Wells in Bellingen on Saturday the 13th of December. In all there were seven TI users in attendance, two of whom are not members of TISHUG but will be sending there applications down to "head office" within the next week or two.

One of the main matters discussed was the appallingly low response rate from members of the TISHUG committee to mail sent down from members of the club in this area. If it be any consolation though, the Software Librarians name was excepted.

We believe that our Regional Group could well expand to have a membership in excess of thirty within the next 4-6 months. There having been more than 200 TI's sold in the Coffs Harbour area the only problem we now face is contacting these people.

Great interest was expressed in the ability to run Memory Image programmes with only the 32K expansion and cassette and most members are hopeful to carry out memory expansion on their systems. On the matter of software, our sincere thanks go to Terry Phillips who supplied Keir Wells with programmes that were requested by members of the group.

If any members of TISHUG wish to attend any of the Banana Regional Group meetings (Coffs Harbour area), then please contact either:

Keir Wells (066)55-1487; or

Kevin Cox (066)53-2649.

THESE COULD BE THE KEYS TO YOUR FUTURE



HARDWARE AND PERIPHERIAL SUPPORT FROM MECHATRONIC OF WEST GERMANY NOW AVAILABLE IN AUSTRALIA.

THE MECHATRONIC EXTENDED BASIC II PLUS solid state cartridge for the TI 99/4A comes with the TI Extended Basic manual and a 96 page supplementary manual covering the additional commands and statements offered by this cartridge. It has 19 additional statements not available from the TI Extended Basic module; off course all the TI EX.BASIC functions are fully supported. Some of the additional statements are:

- *> CALL BHCOPI - will generate a hard copy of the screen on line printers operating in the bit image mode.
- *> CALL VPPEEK - will read VDP RAM memory addresses.
- *> CALL VPPOKE - allows to write bytes directly to addresses in the VDP RAM.
- *> CALL GPPEEK - will read contents of addresses in the computers GROMS.
- *> CALL ALLSET resets all characters, including lower case characters.
- *> CALL WAIT - introduces a specified pause up to 327 s.
- *> CALL MOVE - allows to move contents of memory blocks within the RAM (e.g. from VDP to VDP, from VDP to CPU from CPU to VDP, from CPU to CPU).
- *> CALL MSAVE - will save specified parts of the CPU RAM contents in program format to an external device.
- *> CALL MLOAD - will load program files into CPU RAM being previously saved by CALL MSAVE

The most significant feature of this module is the superb graphic capabilities developed by APESOFT. These are 39 graphic subroutines accessed through CALL LINK statements in the high resolution graphic mode. These enable the user to execute the most complex graphic displays with the greatest of ease and minimum program lines. The Mechatronic users manual gives many examples to aid understanding of the APESOFT concept.

Current price \$ 177.-

TI-MOUSE.

An other high quality MECHATRONIC product. This is the most exiting TI peripheral! It comes complete with power supply, interface, manual and diskette software. It opens up a new field of realistic games, graphics and programs with instant, extremely fast and unrestricted cursor control. One has to see the demonstration game on the supplied diskette to appreciate the potentials of the TI-MOUSE. The 19 page manual explains the essential steps to include the MOUSE calls in a program - the users imagination is the limit. The listable disk files give programming examples of BASIC as well as ASSEMBLY programming methods (source code is included). The MOUSE routine may be easily included in any EXTENDED BASIC program. Disk system and memory expansion are essential.

Current price \$ 266.25

32K RAM & CENTRONIC INTERFACE, 128K RAM & CENTRONIC INTERFACE.

These stand alone peripherals plug into the expansion socket of the console. In addition to the memory extension it also features a centronics (parallel) printer outlet. Due to import pricing (freight,duty and sales tax), the price difference of these two units is negligible. Thus we do not recommend the 32K version. The additional 96K memory of the 128K RAM may be configured to RAM-CARD (3 Cards in banks of 32K each) or may be used as additional memory in bank switching mode, which increases the active console memory to about 142K. Additional stand alone modules or a PE Box may be plugged into the RAM module. The module is supplied with its own plug-pack power supply and a 23 page users manual.

Current price \$ 177.-

MECHATRONIC 128K GRAM CARD.

This card is designed for the PE Box. It is filled with 43 ic-s, Gram and Rom. Naturally it does not come cheap. One can save any TI Module on disk, and subsequently run it from the Gram Card . The card may also be used to produce ROM listings of the modules for subsequent use through the EPROM BURNER. The card was reviewed in the Sept.1986 issue of Micropendium. A 24 page manual explains the options and procedure.

Current price \$ 576.-

MECHATRONIC EPROM BURNER.

The EPROMMER will burn in any of the ROM-s used in the TI system. It can also be used to produce customised EPROMS. The unit is entirely software controlled, it has no switches to manipulate. The diskette will accept the 2716, 2732 or the 27128 EPROMS. It is supplied complete with plug-pack power supply, diskette based software and manual. The Eprommer was extensively tested in NSW. Several dead consoles have been successfully resurrected by newly burned in ROMS. The menu driven software supports the loading of buffer memory direct with contents from ROM, GROM or RAM. Essential hardware: Eprommer, 32K Exp.Mem., Ext.Basic, E/A module or Mini Mem. and disk system.

Current Price \$ 230.-

TI 99/4A-INTERN.

A 207 page book listing and explaining the computer's ROM, GROM, GPL and GRAM contents. An essential and so far not published information of the internal structure of the 99/4A computer. It is an invaluable aid to the advanced programmer.

Price \$ 23.-

All above is now available from:
Ben v. Takach
PO.Box 114
Wahroonga NSW. 2076
Tel: (02) 4894492
VIATEL Mail Box: 248449200
MINERVA Mail Box: TUE001

 * All About... MAX/RLE *
 * by Steven Shraibman...(SUS) *

I am sure that most of you have bought a program by the mystefying name of MAX/RLE from the club shop recently and I am equally sure that most of you have been quite impressed by some of the pictures provided with it and on the additional disks of pictures available.

Well, going back to the beginning RLE stands for Run Length Encoded, something that I am sure is quite meaningless to most people. To put it simply, RLE is a standard protocol that can be used to describe hi-res screens (without colour) on all different computers.

This means that the one file is suitable to describe a hi-res screen on all computers, all you need is the program to translate the file into a picture on the TI. In the TI's case this is MAX/RLE.

This of course means that all those great digitized pictures were most likely not to have been made on a TI. (However all your friends with crummydores don't know that and we won't tell them, will we?). But RLE does not only mean digitized pictures, it can be any hi-res screen, wether from Basic, Assembly Language or a drawing program such as Graphx or TI-Artist.

Most of these pictures are found on bulletin boards across America, literally in there thousands. (There is a file floating around that lists just some of the pictures available on a few of the bulletin boards in America. (Terry, the club librarian might put it on one of the future MAX/RLE disks.)

To get these files of a bulletin board in America all you have to fo is download with XMODEM protocol (and pay an ISD fortune).

The other bonus of this being a standard protocol is that one can transfer pictures between different computers direct. I believe Arty is working on programs to connect our computer with some other computers such as the Apple and the Amiga.

Anyway, getting onto the MAX/RLE program itself, there are some functions that many people will not be aware of.

Firstly, from the main menu, if you typ in "DSKn.", i.e. without a filename, you will get a catalogue of disk drive n.

Obviously if you type in "DSKn.FILENAME" you will load the file FILENAME form disk n. But above that, the program will load in four types of files:

- a) a Graphx file
- b) a TI-Artist file
- c) a Dis/Fix 128 file

d) a Dix/Var 80 file
 (N.B. for TI-Artist, it must be from V2.0 and don't type in the "p" or "c".)

Now that you have the file on the screen what can you do? Well, quite a bit. Firstly if you type the keys 0 to 9 and A to F will change the foreground and background colours depending on if you are in upper or lower case.

Here is a list of the colours:

Keys	Colour	Keys	Colour
0)	transpar	1	black
2 @	Med. green	3 #	Light Green
4 \$	Dark Blue	5 %	Light Blue
6 ^	Dark Red	7 &	Cyan
8 *	Med. Red	9 (Light Red
a A	Darkyellow	b B	Lightyellow
c C	Dark green	d D	Magenta
e E	Gray	f F	White

Pressing "p" will print out the screen to an Epson compatible printer. And pushing "s" will save the file in the four above-mentioned formats. To alter the format just push the space bar.

Not only is this handy in its own merits, its just great for transferring screens between TI-Artist and Graphx and back.

Finally pushing enter will return you to the menu to load another picture.

Quite a handy package indeed. Get yours now from the club library. Only \$5 plus \$5 each for the two extra disks of pictures, and more are on the way.

COME AND JOIN US AT OUR AGM VENUE ON SATURDAY 7TH FEBRUARY (2PM)

We have been able to acquire the use of the R.S.L. HALL in Burwood for the Annual General Meeting. This is a temporary venue because of the expected size of this very special meeting where all financial members are asked to attend.

Please ensure you bring your Membership Card along with you, and fill in the Nomination Form on the back pages of our AGM PULL-OUT SUPPLEMENT.

