

---

---

**ISHUG**

---

---

**NEWS  
DIGEST**

---

---

Focusing on the TI99/4A Home Computer

---

---

Volume 12, Number 9

October, 1993

---

---

Registered by Australia Post - Publication No. NBH5933

---

---



---

Sydney, New South Wales, Australia

---

\$3

---

TiSHUG (Australia) Ltd.  
A.C.N. 003 374 383

## TiSHUG News Digest

All correspondence to:  
C/o 3 Storey St.  
Ryde 2112 Australia

TiSHUG News Digest

ISSN 0819-1984

## T I I N D E X

### The Board

#### Co-ordinator

Dick Warburton (02) 918 8132

#### Secretary

Russell Welham (043) 92 4000

#### Treasurer

Cyril Bohlsen (02) 639 5847

#### Directors

Percy Harrison (02) 808 3181

Peter Schubert (02) 318 1191

### Sub-committees

#### News Digest Editor

Bob Relyea (046) 57 1253

#### BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4608

#### Merchandising

Percy Harrison (02) 808 3181

#### Publications Library

Russell Welham (043) 92 4000

#### Software Library

Larry Saunders (02) 644 7377

#### Technical Co-ordinator

Geoff Trott (042) 29 6629

### Regional Group Contacts

#### Central Coast

Russell Welham (043) 92 4000

#### Coffs Harbour

Kevin Cox (066) 53 2649

#### Glebe

Mike Slattery (02) 692 8182

#### Hunter Valley

Geoff Phillips (049) 42 8176

#### Illawarra

Geoff Trott (042) 29 6629

#### Liverpool

Larry Saunders (02) 644 7377

#### Northern Suburbs

Dennis Norman (02) 452 3920

#### Sutherland

Peter Young (02) 528 8775

### Membership and Subscriptions

Annual Family Dues \$35.00

Associate membership \$10.00

Overseas Airmail Dues A\$85.00

Overseas Surface Dues A\$50.00

### TiSHUG Sydney Meeting

The October Meeting will start at  
2.00 pm on the 2nd October 1993  
at Ryde Infants School,  
Tucker Street, Ryde.

Printed by

Kwik Kopy Parramatta

Title	Description	Author	Page No.
BITS AND BITES	HINTS	LARRY SAUNDERS	7
BURGLAR ALARM FLIP FLOP	PROJECT	KEN GLADYSZEWSKI	4
BURGLAR ALARM WHITE ELEPHANT CONSOLE	PROJECT	BERNARD ZCKERMAN	2
DEFINE YOUR OWN PRINTER CHARACTERS	HINTS	STEPHEN SHOUSE	16
EXTENDED BASIC TIPS	HINTS	ROSS MUDIE	9
LEARNING TO KNOW YOUR TI No.9	PROGRAMMING	PERCY HARRISON	6
LETTER FROM AUSTRALLIA No.2	GENERAL INT.	TONY McGOVERN	10
NOSTALGIA TIME	GEN. INT.	GEOFF TROTT	8
REGIONAL GROUP REPORTS	GEN. INT.		23
TiSHUG SHOP	CLUB NEWS	PERCY HARRISON	2
TiSHUG SOFTWARE	CLUB NEWS	LARRY SAUNDERS	4
TI-WRITER	HINTS	DICK ALTMAN	13
TREASURER'S REPORT	CLUB NEWS	CYRIL BOHLSSEN	10
USING A RAMDISK IN A SINGLE FLOPPY DISK SYSTEM	HINT		9
VINCENT'S CORNER	HINTS	VINCENT MAKER	5

## I B M I N D E X

ARCHIVING PROGRAMS	18
HOW MUCH MEMORY DO YOU NEED	22
MOTHER BOARD MEMORY	18
PC MEMORY BASICS	20
THE DOS CONNECTION	21
THE IMPORTANCE OF MEMORY	19



## TISHUG SHOP

with Percy Harrison

I have become somewhat concerned about the way that we have been conducting our monthly meetings through most of this year and feel that if we do not get them organised a little better than we have been doing of late then we will hasten the demise of our club. For example, we rarely seem to have our formal meetings which served the purpose of letting our members know what their Directors were planning for forthcoming meetings. They also served to hear from our members just what they would like in the way of tutorials, demonstrations and technical advice as well as give them their opportunity to air their complaints and to offer constructive criticism. This enabled the directors to take the necessary action to satisfy your needs.

To this end I did raise the subject at our August meeting with the result that at future meetings we will start proceedings by holding a formal session right at the start of the meeting, ie 2.00 pm for the half day meetings and 10.00 am for the full day meetings. It is hoped that the meetings will be kept brief but will serve to give members an idea of what is planned for the day and also for the next month or so and at the same time give you a chance to communicate with the Directors. No, this is not something new but somehow, over a period of time, it seems to have become very spasmodic and somewhat out of hand. I feel that such meetings are essential to the proper operation and running of our club especially at a time when we are expanding our membership to cater for IBM compatible user's. So, let us all try and make a go of it by coming to the meetings and actively participating.

By the way, please remember that the October meeting will be the last one that we hold at the Top Ryde Infants School venue. From November on our meetings will be held at the Meadowbank Primary School, Thistle Street Meadowbank (see September TND for details). Entry to the parking area is from Gale Street on the western side of the school (refer to your street directory).

The club will now be releasing IBM compatible fairware disks but initially they will only be supplied against firm orders to ensure that we do not waste time copying disks that are not sold. The first disks to be released are listed below and are priced at \$3.00 for 5.25 format and \$3.50 for 3.5 format. Wherever possible we prefer you to order 5.25 format but offer 3.5 for those members who do not have a 5.25 drive on their system. At each meeting I will have a form on which you can indicate your requirements and they will be available at the following meeting, alternatively you can order them by mail or phone but please allow for postage if you want them mailed to you.

### PRICE LIST.

5.25 in. DSDD Disks (Box of ten)	.....\$6
5.25 in. HD Disks (Box of ten)	.....\$10
3.5 in. DSDD Disks (Box of ten)	.....\$10
3.5 in. HD Disks Formatted (Box of 10)	.....\$14
5.25 in. DSDD Half Height Drive (New)	.....\$35
12 Volt AC Transformer	.....\$4
13 Volt Arlec Transformer	.....\$12
8.5, 17 Volt Transformer	.....\$25
60 VA Transformer	.....\$20
MFC Printed Circuit Board	.....\$30
MFC Kit (Disk Controller)	.....\$103
Music Kit with PCB	.....\$65
32K Memory PC Board	.....\$7
Horizon Ram PC Board	.....\$45
Horizon Ramdisk Basic Kit	.....\$35
Funnelweb Eprom Set (3 Eproms)	.....\$36
32K Static Ram IC (62256)	.....\$10
8K Static Ram IC (62641P)	.....\$5
Exchange Console	.....\$30
ROS Version 8.14	.....\$12
Mini Master 99	.....\$70
Mini PE RS232/32k Card	.....\$80
Mini PE RS232/PIO PC Board	.....\$30
Modulator UHF or VHF	.....\$15
TI Power Supply	.....\$25
TI 32K Memory Card	.....\$40
RS232/PIO Card	.....\$80
Modem PE Card (300 Bd)	.....\$30
PE Expansion Box with I/F	.....\$150
PE Ramdisk (184k Eprom)	.....\$140
PE Ramdisk (248k Eprom)	.....\$180
PE Ramdisk (320k Horizon)	.....\$210
Printer (Serial)	.....\$120
Thunderer Modem with Viatel Cart	.....\$130
Standalone Disk Drive	.....\$40

### IBM SOFTWARE

As Easy As (Spreadsheet)  
Flash (Horse Racing Data Sheet)  
Powermenu (Menu)  
PC Files (Data Base)  
VDE (Word Processor)

Above Disks on 5.25 Format	.....\$3.00
Above Disks on 3.5 Format	.....\$3.50

Note: Where possible please order 5.25 format.

Packaging and postage extra on all items.

**END OF ARTICLE**

## "WHITE ELEPHANT" CONSOLE

Turns Into Home Security System  
by Bernard Zuckerman

The following item appeared in the newsletter of the Cleveland Area TI99/4A User group. We are publishing it here because it demonstrates what a little ingenuity and a TI console can do - Ed.

It all started way back in 1990 when I ordered a disk from the user group library. I do not remember the program I wanted but on the disk was a program written by R.E. Lunsden of Winnipeg, Ontario, called BURGLARM. It used the joystick port wired to the perimeter contacts and keyswitch of a house burglar alarm system. The monitor gave all the instructions and the resulting audio (beeps, noise, siren) were appropriate for a burglar alarm. It was fascinating to hook up and even more interesting to work out the logic. After a while, the disk was stored away and almost forgotten.



Last spring my wife and I went to a white elephant sale. She usually looks around at the kitchen gadgets and books, while I dig into the camera, electronics, etc shelves. Lo and behold, there on a back shelf, covered with years of dust, was a TI99/4A console - no cables, no power supply, no modules, no nothing - just a bare console! How can a true Tier resist? For \$1.00 it was mine.

"What in heaven's name are you going to do with another computer?" Was my dear wife's comment.

"Oh, it probably does not work so I will give it to Ken for parts, or I will make it into a burglar alarm." Wow! That idea came out of the blue. And the console worked fine!

### THE TI IS THE BRAIN

But saying "make it into a burglar alarm" and making it work was another story. In its elementary form, a burglar alarm system consists of perimeter wiring with contacts - NO or NC (Normally Open or Normally Closed) - at each door, window, etc. And a "keyswitch" to arm or disarm the system. The "brains" must be able to distinguish between a legitimate opening and a "break in" and send or give an alarm when appropriate. In my case, the brain is to be the computer.

Lunsden wrote a program that makes the console into a burglar alarm brain. It checks the perimeter, the keyswitch and provides delays for coming and going. The CALL SOUND command made all the noise, beeps and sirens via the monitor, and the instructions were all displayed on-screen.

Well and good. But I do not need the monitor if I am not at home, and I do not want to keep my PEB on while I take my Hawaiian vacation for three weeks. And I do not want to activate my automatic dialer to call the police. So I set up the parameters of my project:

- \* A stand alone console - no PEB, no monitor, no modules.
- \* It must be able to sound a siren or loud horn, as well as communicate to me, the operator.
- \* It must provide a NO contact which will close and start my automatic phone dialer.
- \* It must signal me if the window or door is open or if it is not properly armed.
- \* No joysticks allowed.

### A LITTLE HELP FROM A FRIEND

Well, I can eliminate the monitor by connecting the audio output to a free-standing audio amplifier and speaker and use the CALL SOUND to send beeps, notes, songs and other noises as signals as to what is going on. A cassette will load the program and then be removed. But how do I get an output that will close a contact (energise a relay)?

That is where Ken Gladyszewski came into the picture. At one of our meetings I discussed the project with him and he picked up on it. Inside a month he designed and built an electronic circuit which worked on a CALL JOYST(2,X,Y) command. See the companion article for the circuit and how it works - Ed.

An audio amplifier integrated circuit from Radio Shack worked fine and the program was modified (see below) to provide the brains.

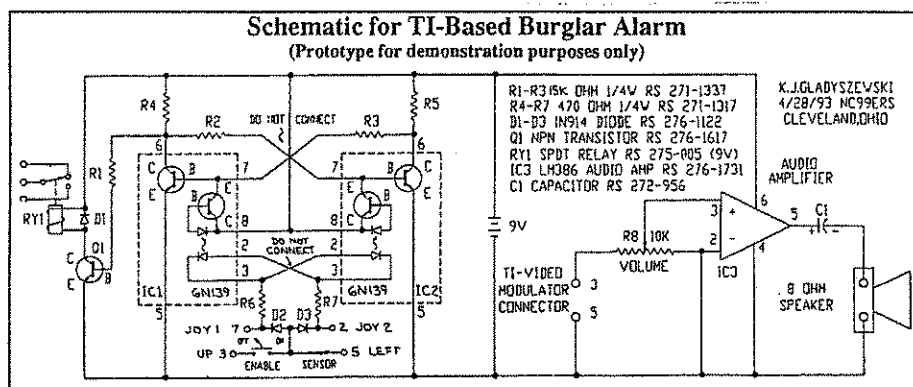
I broke one part of my original goal. Since I had a spare Speech Synthesizer, I revised the program so that the computer tells me what to do. And it worked!

But what if the power goes off?

```

100 REM BURGLAR ALARM PROGRAM, ELECTRONICS BY KEN G.,
    PROGRAM BY LUNSDEN, REV. BY B.Z.
101 REM FILE BURGLARM4A-BASIC, NO VIDEO, SPEAKING AND
    SIREN
110 REM Y IS KEYSWITCH; X IS PERIMETER.Y=+4 K.S.
    CLOSED; X=-4 PERIMETER CLOSED.
120 REM START WITH KEY SWITCH A NO PERIMETER SWITCH OPEN
130 ENDEL=3000
140 EXDEL=3000
150 SKIPD=1
160 SW=1
170 CALL JOYST(1,X,Y)
180 GOTO 440
190 PRINT #1:"OK THANKS OK THANKS"
200 CALL JOYST(1,X,Y)
210 IF Y=0 THEN 600
220 IF SKIPD>1 THEN 250
230 GOSUB 410
240 SKIPD=SKIPD+1
250 CALL JOYST(1,X,Y)
260 IF X=-4 THEN 200
270 REM TIS SITUATION CAN OCCUR EITHER (1)LEGITIMATE
    OPENING, OR (2)BREAK IN. SO WE GIVE A FEW SECONDS
    BEFORE ALARM IS SOUNDED.
280 FOR ENTRDEL=1 TO ENDEL
290 NEXT ENTRDEL
300 CALL JOYST(1,X,Y)
310 IF Y=- THEN 390
311 PRINT #1:"ALARM TRIPPED BE ALERT BE ALERT BE ALERT!"
320 CALL JOYST((Y/4+1),X,Y)
329 REM ALARM BROKEN SIREN
330 FOR LOOP=1 TO 5
340 FOR SIREN=700 TO 900 STEP 10
350 CALL SOUND(-99,SIREN,0)
360 NEXT SIREN
370 NEXT LOOP
380 GOTO 391
390 PRINT #1:"WELCOME HOME"
391 END
400 REM EXIT DELAY INITIATED FOLLOWED BY ARMING
    PROCEDURE
410 FOR DELAY=1 TO EXDEL
420 NEXT DELAY
430 RETURN
440 REM CHECK PERIMETER. IF O.K. CHECK KEY. IF NOT,
    RECHECK PERIMETER.
450 CALL JOYST(1,X,Y)
460 IF X=-4 THEN 190
461 IF SW>1 THEN 481
470 IF SW>1 THEN 481
479 REM PERIMETER OPEN WARNING SOUND
480 OPEN #1:"SPEECH",OUTPUT
481 PRINT #1:"//40 500"
482 PRINT #1:"DE PERIMETER IS OPEN SHUT THE WINDOW"
490 SW=SW+1
510 GOTO 450
599 REM KEY SWITCH OPEN WARNING
600 PRINT #1:"DE KEYSWITCH IS OPEN THE SYSTEM IS NOT
    ARMED"
650 GOTO 200

```



# FLIP FLOP

## FLIP-FLOP SOLVES HARDWARE PROBLEMS FOR BURGLAR ALARM by Ken Gladyszewski

When Bernie and I first talked about his desire to control an automatic telephone dialer using a bare console, used as a burglar alarm, he suggested using an electronic circuit to energize a relay when it "hear" the console make a siren sound. I told him I had seen circuits for this somewhere, but though I knew of a better and simpler way.

I had already used the pulses generated by the joystick port, when a CALL JOYST command in basic is executed, to control an analog-to-digital converter chip. And thought about using the same pulses to control a flip flop.

A flip flop is an elementary building block in digital electronics and is the basis of all memory and counting circuits. It has two stable states and in its simplest form consists of two transistors cross connected so only one transistor is "on" at a time. The circuit is said to "flip" when the on transistor turns "off," and vice-versa. The method normally used to cause the flipping action is to apply a pulse to one of the two transistors.

In review, the joystick port consists of two outputs (pins 2 and 7) which are normally both at +5 volts with respect to the common of the console power supply. When a CALL JOYST command is executed, one of the outputs goes to -5 volts for 150 microseconds. If a connected joystick is pushed in any direction, a switch in the joystick unit is closed and the output is connected to an input. The inputs are scanned and the program returns a value indicating joystick position.

In my early projects I used the 10 volt differential between the two outputs to fire a transistor, which then produced pulses compatible with the integrated circuits I was trying to control. A transistor was used because only 1ma of current is available, which is not enough to light a light emitting diode (LED) in an Optocoupler - or so I thought.

### OPTOCOUPLER FORMS FLIP-FLOP

Recently I discovered an Optocoupler (6n139) that requires only 1/2ma, and it just so happens that two of these devices may be interconnected to form a simple flip-flop! By executing a CALL JOYST(2,X,Y) command it will flip in one direction energizing a relay, and a CALL JOYST(1,X,Y) command will make it flip back, de-energizing the relay. At the same time, the status of both the key and perimeter switches are returned in the X and Y values. The circuit shown in the diagram was used to develop the concept and was used in Bernie's demonstration. to be used as an actual burglar alarm, because of the likely distance between the computer and the switches, Optocouplers would need to be used as buffers.

The circuit as shown is for only one zone, but it may easily be expanded to two zones by hooking each zone to a direction input and the keyswitch to the fire input and modifying the program appropriately.

I am working on an expanded version for three zones using two more intergrated circuits (eight or more zones are possible using a different scheme) for use in my own home. Three zones would be nice because my house has three levels, with an entrance on each. The computer could detect and announce the level on which the problem occurred.

I plan to remove the motherboard from a spare console, along with the switching power supply, and house them in a wall-mounted metal box, along with some gel cell batteries for program backup. The peripheral circuit shown would be powered from the 12 volts available at the modulator port, thereby eliminating the need for a 9 volt battery.

As further reading, I suggest the following:

- \* Home-security cookbook, parts 1 and 2, by Ray Marston
- \* Radio-Electronics Magazine, May and July, 1988
- \* Working with Flip-Flops, by Ray Marston, Radio-Electronics, June, 1987
- \* Optocoupler Devices, by Ray Marston, Electronics Now Magazine, Aug, 1992

**END OF ARTICLE**

## TISHUG Software

by Larry Saunders

Disk GO40

Used= 354 Free= 4

Gridword: A word game.

Honey Hunt: A very good assembly language program that has as far as I can tell never been seen by most members before. You are a bee and have to collect the necture from the flowers without getting caught by the spider etc. The game is by Milton and Bradley.

Munch Mobile: A car driving game, for younger children.

Documentation for the ORB game

Game program written by Jon Dyer and Joe Delektio. Graphics designed by Jon Dyer and documentation written by Jon Dyer. Loading the game:

\* The program is a standard E/A 5 program file and should load with any E/A 5 program loader. The filename (unless changed by someone else) is ORB.

\* NOTE: I have set the program up to run from the ROOT loader and you do not have to use Editor Assembler option 5.

```
ORB_DOC  -- This file
ORB      \
ORC      - The program files
ORD      /
```

Playing the game:

Game play is pretty simple. You can play either one or two players, or you can let the computer play itself in demo mode. To play one or two players, just press '1' or '2' at the title screen. For demo mode, just let the counter reach zero. Demo mode will start automatically. You can return to the title screen at any time by pressing 'BACK'.

With one player mode, only joystick 1 is used. Try to shoot the other player as many times as you can without getting hit yourself. Each time you hit the other player, your power level increases by one unit and your opponents power level decreases by one unit. Whichever players power level is empty first loses.

When the game is over, you may then choose to return to the title screen (of the game) to play again or quit the game.

Have fun playing ORB!!

GRIDWORD	17 Prog	HONEYHUNT1	33 Prog
HONEYHUNT2	33 Prog	HONEYHUNT3	33 Prog
HONEYHUNT4	29 Prog	LOAD	5 Prog
MUNC	33 Prog	MUND	33 Prog
MUNE	27 Prog	ORB	33 Prog
ORB_DOC	10 d	80 ORC	33 Prog
ORD	7 Prog	ROOT	28 Prog

Disk P041

This disk is full of converted pictures from assorted formats to Page Pro by me. Most are very good quality pictures.

Used= 356 Free= 2

APPLE	3 I 13	BANDAID	2 I 13
BASKET	18 I 13	BELL	5 I 13
BIRDBATH	3 I 13	BOOK	4 I 13
BOOK2	2 I 13	BOW	2 I 13
BROOM	4 I 13	CAMERA	5 I 13
CANDLE	17 I 13	CANDLE2	2 I 13
CANDLE3	2 I 13	CANDLE4	7 I 13
CANDLE5	2 I 13	CASTLE	2 I 13
CHEESE	8 I 13	CHISEL	2 I 13
CHURCH	3 I 13	CIGARTTE	3 I 13
COMB	3 I 13	CROSS	2 I 13
CUPID	6 I 13	CUTTERS	3 I 13
DIAMOND	3 I 13	DIPLOMA	4 I 13
DONOT	4 I 13	FLAG	3 I 13
GASPUMP	3 I 13	GLASSES	3 I 13
HAMMER	2 I 13	HEART	2 I 13
KEY	3 I 13	KEY2	2 I 13
KNIFE	2 I 13	LGHTBULB	3 I 13
LIBRTBL1	3 I 13	LOOK	2 I 13
MATCHES	3 I 13	MNKYWRNC	3 I 13
MORSECDE	2 I 13	MRRYXMAS	6 I 13
MRYXMAS2	6 I 13	MSHRMS	3 I 13
NEWYEAR	7 I 13	PAGODA	2 I 13
POINTER	2 I 13	RADIO	3 I 13
ROSE	23 I 13	RTANGLE	4 I 13
SAW	3 I 13	SCISSORS	2 I 13
SCREW1	6 I 13	SCREW2	6 I 13
SCRWDRVR	2 I 13	SOCRCBLL	2 I 13
SPADE	2 I 13	STAR	3 I 13
THUMSUP	4 I 13	TORCH	2 I 13
TOUCAN	10 I 13	TREE	16 I 13
TREE2	2 I 13	TRKYDAY	19 I 13
WHTCAT	6 I 13	WINEGLSS	3 I 13
WNEGRPS	27 I 13	WREATH	6 I 13
XMASTRE1	2 I 13	XMASTRE2	3 I 13
XMASTRE3	12 I 13		

Disk ADO42

This disk contains three adventure games. The MS-Adventure Series consists of an MS-Adventure program and three MS-Adventure databases. The MS-Adventure databases describe the adventure. The MS-Adventure program reads the databases and controls the adventure which the databases describe.

The following is a list of the MS-Adventure databases which are included in this series.

SEARCH FOR MURGEN'S KEEP:

This adventure is Part I of the Quest for the Key. The Princess Dianna has been imprisoned by the magician Murgan, and the key to release her is in his castle. In Part I you must search a magical land for entrance into the magician Murgan's castle. Difficulty Level: Advanced

THE ENCHANTED KEEP:

This adventure is Part II of the Quest for the Key. The search for the key to free the imprisoned Princess Dianna continues through Murgan's enchanted castle. Difficulty Level: Moderate

THE NEW KING:

The Quest for the Key is complete! The princess has been freed! Now you must journey to Dianna's kingdom and vanquish the evil knight who has taken over the kingdom. Difficulty Level: Advanced

Used= 352 Free= 6

ADV/C	25*Prog	ADV/LOADER	14*D 80
ADV/P	25*Prog	ADVDATASMK	32*I192
ADVDATEK	24*I192	ADVDATEATNK	33*I192
DOC/FAIR	20*d 80	DOC/JETCAT	48*d 80
DOC/MAIN	25*d 80	DOC/QFTK	13*d 80
DOC/TNK	8*d 80	LOAD	19*Prog
PDTTEXT	9*Prog	SOL/SMK	9*d 80
SOL/TEK	9*d 80	SOL/TNK	9*d 80
UTIL1	30*Prog		

Disk P043

Another disk full of quality Page Pro pictures converted by me.

Used= 341 Free= 17

ANGLFISH	9 I 13	BIRDS	14 I 13
CAT	17 I 13	COUGAR	25 I 13
DWARFS	17 I 13	EAGLE	14 I 13
GIRL	12 I 13	GODZILLA	5 I 13
IMPALA	11 I 13	KNIGHT	23 I 13
KNIGHT2	12 I 13	MCKY&MNE	26 I 13
OTTER	18 I 13	PEASANTS	18 I 13
PYRAMID	20 I 13	RACECAR	13 I 13
RX	2 I 13	SEAL	14 I 13
SMLDISK	3 I 13	SNOWHITE	20 I 13
STAMP	3 I 13	STAPLER	2 I 13
STOPSIGN	2 I 13	SWAN	17 I 13
TI#1	3 I 13	TI#2	3 I 13
TRFCLGHT	2 I 13	TURTLE	13 I 13
WHLCHAIR	3 I 13		

**END OF ARTICLE** 

## Vincent's Corner

### ADVENTURE REVIEW

This is a quick adventure rundown. It covers all the Adventures, the 13 Scott Adams Adventures and the one other one.

ADVENTURE #1: ADVENTURELAND: Supposedly the easiest of the lot of them ... I still have not solved it! You are in a magic "land" and have to find thirteen (13) treasures. I have found twelve. I rate it 4 out of 10.

ADVENTURE #2:PIRATE'S ADVENTURE: This comes with the module. This is by far the best adventure of the lot of them, with possibly the exception of #8, Pyramid of Doom. You have to make your way from a flat in London to Treasure Island and collect two (2) treasures. It is not as easy as it sounds. I rate it 7.5 out of 10.

MISSION IMPOSSIBLE (#3): Well ... Mr. Phelps ... yes, this is the monumental Mission Impossible man himself trying to disarm a nuclear bomb. I rate it 6.5 out of 10.

VOODOO CASTLE (#4): You are trying to bring Count Christo out of the curse that has been placed on him. Good if you like magic. Seven out of ten!

THE COUNT(#5): Not the best of adventures! You are trying to find out who you are, why the Postman delivered a bottle of blood and trying to kill Count Dracula in the process. Nothing spectacular! Rating- 2 out of 10.

STRANGE ODESSEY(#6): Your power crystal does not work on your spaceship so you have to get out of the ship, find your treasures and get another power crystal. Good if you like Dr. Who and all that. Rating- 5 out of 10.

MYSTERY FUN HOUSE(#7): pretty good. You are James Bond(#007) and you have to search the equivalent of Luna Park to find the top secret plans before the park closes at midnight. Cute! Seven out of ten.

PYRAMID OF DOOM(#8): This equals Pirate's Adventure(#2). You are in ancient Egypt to search a (you guessed it!) pyramid. You have to find 13 treasures. I published a review on this one in 1987, October issue. Quite a solid adventure. Rating- 7.5 out of 10.

GHOST TOWN(#9): I have already published a review of this in my column so there is no need to go over it again. Rating- 6.75 out of 10.

SAVAGE ISLAND SERIES(#10&#11): You are on this island and you have to do something- do not ask me what? One out of 10!

GOLDEN VOYAGE(#12): You start off in a Persian city. It is soon revealed to you that you must restore the king's youth. So you set forth to "the four corners of the globe" to do exactly that. A very broad-minded adventure. Quite interesting. Not very easy! Rating- 7.25 out of 10.

SORCERER OF CLAYMORGUE(#13): You are facing a castle- cast SEED will get you in. What happens from there is beyond me. This is the only one NOT by Scott Adams. Zero out of 10!

RETURN TO PIRATE'S ISLE(#14): You are on a ship and an alarm clock rings. Outside there is the back of .. you guessed it! ... PIRATE'S ADVENTURE (#2). Find the thirteen treasures and where to store them and you win. I know where to store them. Not as good as No. 2 which it is based on. Rating- 5 out of 10.

I would suggest the beginner start with numbers #2 and #3 and perhaps #4.

Cheers 4 now,  
CROCODILE JONES

**END OF ARTICLE**

## LEARN TO KNOW YOUR TI LESSON 9

with Percy Harrison

This month we will look at TAB, function arguments and delay loops.

Delay loops slow the program down so that its operation can be more easily observed. They are also used for portions of the program which must run at certain speeds, in which case they should rightly be called "timing loops".

The TAB command adds flexibility to the screen display. TAB is used in a PRINT command and is like the tab on a typewriter, it allows interesting displays of written information. If you are not very familiar with a typewriter you may need extra help in seeing what a TAB is good for. Enough of this pre-amble let's get on with the lesson.

### LESSON 9 TAB AND DELAY LOOPS

#### THE TAB COMMAND

TAB in a PRINT command moves the cursor to a new spot to the right and the next thing to be printed goes where the cursor is.

Try this:           10 PRINT "123456789ABCDEF"  
                      20 PRINT "Y";TAB(5);"Z"

Rule: After TAB(N), the next character will be printed in column N ie. in the above example the "Y" will be printed in the first column and the "Z" will be printed in column 5 both, on the second line.

CAREFUL!

run this:           10 TAB(5)

You see             INCORRECT STATEMENT  
                      IN 10

TAB() has to be in a PRINT command. Ypu cannot use TAB() by itself.

#### YOU CANNOT TAB BACKWARDS

Try this:           10 PRINT "123456789ABCDEF"  
                      20 PRINT "1";TAB(9);"9";TAB(3);"3"

The TAB() command can move the printing to the right but if you try to move back to the left, the computer will move down one line first as was the case in the above example.

#### YOUR NAME IS FALLING

```
10 CALL CLEAR
20 LET N=1
30 PRINT"YOUR FIRST NAME"
40 INPUT W$
50 PRINT TAB(N);W$
60 LET N=N+1
70 GOTO 50
```

Now run this program.

Press FCTN CLEAR to stop the program from running.

This program prints your name in a diagonal down the screen, top left to bottom right.

Try other values of N.

Now try changing lines 20 and 60.

```
20 LET N=25
60 LET N=N-1
```

#### FUNCTIONS DON'T FIGHT BUT THEY HAVE ARGUMENTS

TAB() is a command which is like a "function".

In later lessons we will study other functions like RND(), INT(), SEG\$(), etc.

The number inside the () is called "the argument of the function". TAB() says "move the cursor over" and the argument tells "where to move it".

### Assignment 9A:

1. Write a program which asks for the last names and nicknames. Then print the last name starting at column 5 and the nickname at column 15. Use a GOTO statement so that the program is ready for another name-nickname pair.

2. Write a "praise" program. It asks your name, then it beeps and writes your name and then it TABs over in the line and prints a praise.

### DELAY LOOPS

Here is a way to slow down a program. It is a "delay loop".

```
10 REM HIDE AND SEEK
20 CALL CLEAR
30 PRINT "YOU ARE IT"
40 FOR T=1 TO 1000
41 NEXT T
50 PRINT "COMING READY OR NOT"
```

Lines 40 and 41 are the delay loop. The computer counts from 1 to 1000 before going on to the next line. It is like counting when you are "it" in a game of hide and seek.

Try changing the number 1000 in line 40 to some other number.

Remember, each 350 in the delay loop is equivalent to about a pause of 1 second.

Try this program:

```
10 REM--- TICK TOCK ---
20 CALL CLEAR
30 PRINT "WAIT HOW LONG ";
31 INPUT S
36 T=350*S
40 FOR I=1 TO T
41 NEXT I
45 PRINT
46 CALL SOUND(500,300,10)
50 PRINT S;" SECONDS ARE UP"
```

Line 36 has a multiplication in it. We will study this in the next lesson.

### Assignment 9B:

1. Write a "slow poke" program, making it print "STEP BY STEP" with several seconds between each word. Have the computer beep before each word.

### ANSWERS TO LESSON 8

### Assignment Question 8-2

```
10 REM TEEN TIMES
11 REM
20 PRINT "TEEN POWER"
21 PRINT
22 PRINT
23 PRINT
30 GOTO 20
```

### Assignment Question 8-4

```
10 REM FRIENDS
15 CALL CLEAR
16 PRINT
20 PRINT "MIRANDA"
22 CALL SOUND(200,200,10)
25 PRINT
30 PRINT "NELLIE"
40 CALL SOUND(200,400,10)
99 GOTO 16
```

Bye for now.

 **END OF ARTICLE**



# **PUT PEN TO PAPER AND WRITE A TND ARTICLE**

## **Bits and Bites**

October TISHUG BBS  
by Larry Saunders

\*\*\*\*\* Rock Runner \*\*\*\*\*

### Requirements:

- TI-99/4A
- Editor/Assembler module
- 32K
- Disk drive
- Joystick

### Loading and running the program:

Select 2- Editor/Assembler  
then 5- Load and Run  
and load the program.

It will start automatically. If you quit the program you can restart it by selecting option 4. Run and pressing <enter>.

At the title screen you can select a screen by moving joystick #1 or #2 up or down. There are 15 screens available (lettered A through O). Press the <fire>-button to start playing the selected screen.

The object of the game is to collect as many diamonds as possible, digging tunnels, setting up traps, and all this without being hit by a falling rock or by one of the moving creatures. An important part of the game consists in exploring the world and its various inhabitants. The character is controlled using the joystick.



# Nostalgia Time

by Geoff Trott

At the top of the screen some information is displayed:

- BOMBS: The number of bombs you are currently carrying. If you find one, you can pick it up for use later one. You can set a bomb by pressing <fire> and running away. It will explode shortly afterwards, leaving a 3 by 3 hole. Bombs can be useful to blow up unfriendly creatures or to create passages through solid rock.
- DIAMONDS: The number of diamonds you've collected so far vs. the number of diamonds you need to finish this screen. You may collect more diamonds than strictly necessary to earn some extra points.
- TIME: The time you have left to complete this screen. This time is added as a bonus if you finish the screen.
- SCORE: The current score. You receive points for every diamond you collect and for ending a screen successfully.
- Some hearts: These indicate the number of spare lives you have left.  
During the game you can use the following keys:
  - <enter>: Pause. Press <enter> again to continue.
  - <space>: Advance to the next screen if this one has been completed, or give up otherwise. The latter may be useful if you've manoeuvred yourself into a hopeless situation.
  - <redo> or 8: Restart the game on the same screen.
  - <back> or 9: Return to the Rock Runner title screen.
  - <quit> or =: Return to the master title screen.

## Conditions:

This program may be distributed freely. It is provided "as is" and comes with absolutely no warranty.

## More information:

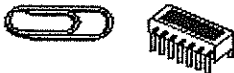
As you may have noticed the game is inspired on the excellent arcade game Boulder Dash, with some slightly different features. Most notably the use of bombs is a nice touch not present in the original game. The program has been written using the Mini Memory module, the line-by-line assembler, a 32K Expansion and a cassette recorder. The game code fitted entirely inside the 4K module RAM, while the program data were stored in the 32K. The game uses an undocumented graphical mode in order to get some more colorful graphics than previously possible.

In the past Rock Runner was marketed by Asgard Software. That contract has expired now and the game has been put into the public domain.

If you have any comments, please contact the author at the address below. Have fun!

Eric Lafortune  
E-mail: ericl@cs.kuleuven.ac.be  
Snail-mail:  
Rijweg 120  
3020 Herent  
Belgium

 END OF ARTICLE



This series of articles consists of my observations on the contents of the early TNDs. Assuming that you find it interesting, I am continuing with the series this month. Please stop me if you do not want me to continue. I will repeat my general disclaimer in case anyone reading this article gets the wrong idea. I am attempting to describe the look, layout and content of the newsletters without any critical intent. I will try to avoid using any adjectives which could cause offense and if anyone takes offense, that is purely their interpretation of the words and not my intention. I hope that makes my position clear and that no one will be offended.

The August 1984 issue had a green cover which felt thinner than the white pages. The picture on the front was of the Opera House with yachts on the harbour in the middle of a fine square grid. It contained 24 pages, with the inside front cover page having information and the back two pages for cutting up. John Robinson, in the Secretary's Notebook, wrote that new memberships are still coming in at a rate of more than one a day. The membership numbers were over 1000 at this time. He mentioned the arrival of Super-Bugger and also gave connections for a modem into the telephone, particularly if having problems. The BBS was up and running with a promise of full speech for those with a Speech Synthesizer.

There was an article on international news which contained news from CorComp. The MiniPE system without disk controller was available in Australia. The International User Group suffered a loss and Triton was chosen to dispose of all TI's home computer products. Peter Day returned with his next article on learning to program in BASIC and there was an Assembler language tutorial from David Ramsey which showed how to use assembler in Extended BASIC programs. There was a review of the Multi-disk Informer program for cataloguing a number of disks at once and providing reports of various types. George Steffen (from USA) had an article on converting a BASIC program written with TI-Writer into a merge file for Extended BASIC enabling the editing of the program to be done where things like global changes can be made. D. Maclean from TIBUG wrote about addresses in system RAM which can be used for various functions, for example to disable the quit key. Howard Sharp had a review of a program called Tower. Jan Todd wrote a simple program to help determine the spelling which makes speech more recognisable for the TEII module. There was a program called Limericks by Tony Falco for typing in. Regional news came from Bankstown, Newcastle, Illawarra, Liverpool, Blaxland, Nepean and Marrickville.

The cover of the September issue was thicker white card again, with a picture of a design team in blue. Along the bottom of the cover was a black silhouette of the city of Sydney with the Centerpoint tower up the left of the page. Inside the black were the date, title and logo of the club. This latter part of the cover was used for the remaining issues of 1984. There were 28 pages with the rest of the cover pages used for the same purposes as for those issues at that time. Secretary's Notebook mentioned that the number of copies of the Newsdigest were up to 1200, there were enhancements for TI-Writer and Multiplan and the source code for TI-Forth was available along with the documentation. At the 30th Co-ordinating meeting there were two resignations; Paul Mansell and Peter Lyden. Greg Hope took over Advertising, Ian Docherty joined the committee but someone was needed to help Shane with the magazine. They decided to purchase a system for the BBS so that it could be on-line more regularly; to buy a 4 tape copier (for \$1600); to buy an amber monitor and an overhead projector. John gave his solution for his password problem with the best solution submitted by Jim Peterson. John suggested that there was an urgent need for a decent database program and an eighty column display.



When IMAGE is used the image string needs no quotes, (although they may be used). In the following examples which specify the image string in the PRINT USING statement the image must be enclosed in quotes, the IMAGE line is not required in the following examples.

```
100 ! SAVE DSK1.USING
120 INPUT A
130 PRINT USING "$####.##":A
140 GOTO 120
```

To use these statements with a program which outputs to your printer change the program as follows.

```
100 ! SAVE DSK1.USINGPIO
110 OPEN #1:"PIO"
120 INPUT A
130 PRINT #1,USING "$####.##":A
140 GOTO 120
```

The use of IMAGE is to allow one specification of the way the output is formatted then this way is referenced from lines throughout the program. The advantage of this is that in a complex program which is employing IMAGE, it is possible to change the handling of output format by changing just one line.

PRINT USING and PRINT USING IMAGE are a convenient way to line up columns of figures. The use of the \$ sign is also optional, any character or string could be used in this position or the \$ sign can be simply left out, e.g., \$US or \$A could be used.

Tabulating with PRINT USING must be set up as a pending print prior to the PRINT USING statement. By adding the following line to any of the above programs TAB may be used: 125 PRINT TAB(10);

PRINT USING can also be used with strings. In this case the reference point is left justified, rather than from the decimal point which was used in the numeric examples. For this type in a name of 8 characters or less; greater than 8 characters gives \*\*\*\*\*. The maximum length of the field is the number of # characters in the IMAGE clause or the PRINT USING string. Try an input string with leading spaces, you will find that the leading spaces are ignored.

```
100 ! SAVE DSK1.USINGSTR
120 INPUT A$
125 PRINT TAB(10);
130 PRINT USING "HELLO #######":A$
140 GOTO 120
```

The following example shows the use of IMAGE with strings:

```
100 ! SAVE DSK1.IMAGESTR
110 IMAGE HELLO #######
120 INPUT A$
125 PRINT TAB(10);
130 PRINT USING 110:A$
140 GOTO 120
```

USING can also be used with DISPLAY and DISPLAY AT; the references are found in the Extended Basic manual as follows:

DISPLAY USING	page 79
IMAGE	pages 97-100
PRINT USING	page 150

#### 4. PROGRAM FORMAT WHICH WILL NOT RUN IN EXTENDED BASIC

A number of people have experienced problems with programs which show up as PROGRAM format when cataloged using the disk manager but give an error if loaded into extended basic with OLD.

It is most likely that these programs are in fact for loading under option 5 of Editor/Assembler which is the RUN PROGRAM FILE option. These programs will probably also be loadable from BEAXs option 5.

The programs which load under e/a opt 5 are assembly in memory image format and the disk catalog does not differentiate between this format and a Basic or Extended basic program which is also stored in memory image format.

**END OF ARTICLE**

## TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month .....	\$ 357.50
Expenditure for previous month ..	\$ 1108.26
Loss for previous month .....	\$ 750.76
Membership accounted for \$ 35.00 of Income.	
Shop sales .....	\$322.50 of Income.
The expenditure was made up of the following	
Purchase of club assets .....	\$ 548.00
Printing & Postage of TND .....	\$ 326.80
Payment of insurance .....	\$ 233.46

## LETTER FROM AUSTRALIA No.2

from Tony McGovern, Jun/30/92

The other night I was sitting downstairs with one of the 1992 lima M-U-G tapes on the VCR (we now have one that plays NTSC tapes) and there was scratch, scratch, bang, bang on the back door beside the TV. Sure enough, it was a visiting possum looking for its feed of bread. Possums are one of the few native marsupials that manage to coexist with suburbia. Around here there are two kinds, the little ring-tail possums which you do not see very much and the larger and cheekier brush-tails, usually with a black brush tail and grey body, but sometimes a more gingery colour. Normally they eat leaves and flowers and so on, but we have seen one sitting on a post on the neighbours' back veranda in the summer cicada season just grabbing them out of the air, holding them by the wings and eating the bodies before dropping the wings in a pile.

Possums have one baby at a time, which rides around clutching on to mother's back when large enough to come out of the pouch. They do this until they are almost as big as mother and it is clear that being a mother possum was never meant to be easy. My visitor had been left to fend for itself at an early age, mother probably having been killed by a large dog, of which several roam the area uncontrolled by their owners. When very young, it would hold onto a finger for balance while you held the bread for it. I think this one lives out in the rafters of the garage.

Why am I talking about possums in a computer newsletter? Well, they can be very territorial, especially the females and it is really something to hear several of them in a snarling, grunting, roiling, travelling brawl over the roof at 3 a.m. in the morning. This has given rise to an old Australian phrase "stirring the possums" and I must confess to doing a little of this in the last letter, with my remark about the Myarc HFDC. Thanks for the network/BBS transcripts Charlie! Sure did stir some HFDC possums! Yes, if you want a hard disk on your TI99/4A that is for now what you have to live with and maybe for some that equates to viability, but I do stand firmly by my comments on the state of engineering development of the device. Actually, all Myarc products seem to have unnecessary flaws that more and better quality development could and should have cured, and the more recent the product the more flawed it is. I use several Myarc devices here, a Myarc 512K RAMdisk in the reserve machine at Hawks Nest and a Myarc floppy disk controller in both machines. Despite its DSR failings and minimal hardware design, the Myarc floppy disk controller is the disk controller of choice for use with 80 column systems, as after power-up it never bothers VDP RAM again unnecessarily. That can be a trap when writing software for general use.

On viewing the tape, I was interested to see that a SCSI controller is in the wings. I was not quite convinced by the "all we have to do is write a DSR" bit though. That sounds even further away from finality than "97% done" (or should that be Vn 0.97?). The SCSI specification is a long and complex document (William FTPed the whole damn thing recently) and device manufacturers have not always come to the same conclusions in their interpretations of it. For instance, I have seen magazine reviews of Floptical drives on PCs which refused to work with other than the SCSI controller supplied with them. I have seen them working on other PC SCSI controllers though. Some models sold for use with Amigas require an obscure SCSI command to remove write protection. You can also get into a lot more difficulty setting up terminations on the SCSI chain than a simple reading of instructions would have you believe. Will is an experienced writer of SCSI device drivers for Amiga third party hardware, so we do have a little background knowledge here in this area.

I am still not sure, even if a good SCSI controller becomes available on the age, TI99/4A, that I would go that way. Part of it is a matter of economics - how much more \$\$ is it worth putting into the TI99/4A rather than into the PCs or Amigas around here. More seriously there is a matter of aesthetics - what makes an elegant TI99/4A system without unbalancing overboard? Just to mention a couple of examples, clearly the Horizon RAMdisk has been a marvellous invention in the true spirit of TI's design. Enhanced video with 80 column adapters are a little less easy in fit to the system but still essential for the serious user. The latest and best of these would appear to be the OPA TIM/SoB. I know they exist because I have seen Al Lawrence's here, but when I was talking to Geoff Trott in Wollongong a couple of days ago, the Sydney order for a bunch of them from last year still had not been delivered. Floppy disks at 720Kb seem just about right, but hard disks seem to me to be out of scale with the TI99/4A though. Not that we have anything against them except their cost; between the A3000UX and the 486 PC there is about 900 Mbytes total normally on line here. A SCSI system that works flawlessly, with its inherent other possibilities as well, would be a far more attractive proposition than the HFDC though. I expect the DSR/software model for the SCSI card will almost of necessity follow the pattern of the IEEE-488 (GP-IB) card of which TI only ever made a small batch. SCSI is a higher performance development of the GP-IB idea which allows a bunch of individually addressable devices on a common bus. I gather, without ever having seen it, that the even rarer Hex-Bus peripheral for the TI99/4A followed this pattern too, as it also drives a bus (low performance in this case) which can have a variety of

What has been driving me crazy though, is the behaviour of the HRD3000 in my AVPC system. I gather Bud Mills says the 3000 with 8.14 ROS is incompatible with the AVPC, but I still do not know whether the problem is in the hardware of the HRD3000 or in the 8.14 ROS or both (or the AVPC - see postscript). The card passes memory tests with flying colours and I have bought and substituted every other IC on the board and checked every diode in the circuit. If anyone has a patch that works for the ROS, I would be delighted to have it. The malfunctioning is very strange. I can in fact use it (and am doing so right now) as a repository for programs or text files - one 400K drive is set up as my Funnelweb and utility program drive and I can store files on the other. But I cannot write D/80 files to it, record by record, even though they can be read that way. Maybe some of it has to do with the 8.14 ROS version, because I am sure it has worked better. A check showed the ROS version being used is dated 05/01/90. I also have a 05/21/90 version, and if that is used the machine locks up entirely, the way it did when we tried installing the RAMBO, and the ONLY way to restore it is to pull the 8K RAM chip on the Horizon RAMdisk right out of its socket (this card has a wired-in lithium battery). Maybe I was using yet another issue at other times, but I may have updated it out of existence while reorganizing disks. Geoff says it sounds like an interrupt problem. What we do need in any event from Bud Mills, as the Horizon RAMdisk supplier, is a plain ROS which is compatible with the AVPC and other devices, works faster, supports 800K drives and is without any of that RAMBO complication. The RAMBO version, particularly if inherently hostile to other devices, should be left as a fully proprietary item to be supplied with the RAMBO device. Even better would be a release of the basic ROS source code, in the spirit of an open system, so that people could make fixes or enhancements as necessary for things that were never dreamed of by the writers. It certainly was a great plus to have the Miami Vn 7.3 source even narrowly available.

This console now has a GRAM device in residence for the first time after all these years. It is a SuperModule 2 from Germany and in effect, is a simplified version of the Mechatronics GramKarte. It fits very neatly into a standard cartridge shell and does not run particularly warm even. There is no battery backup and on initial power-up it comes up as Editor Assembler. It has a 40K GRAM bank (the 5 GRAMS 3-7) and 32K in 4 x 8K RAM banks at >6000. The second GROM bank reached from Review Module library is always Editor Assembler. It looks like a few more little utilities to handle RAM bank switching and write un/protection are in order.

Time to close now on a miserable and rainy winter evening here at Funnelweb Farm. Maybe I will look at putting the wildcard function recently done for the 80-column editor in string searches into the 40-column version before sending it all off to lima.

Postscript: Since writing the above, I have watched Bud's presentation on the lima video. Maybe the solution is after all a revised EPROM for the AVPC, but I will believe it when I see it. I have not had a chance recently to try it in the Mechatronics 80-Z machine, but I know the Myarc RAMdisk print spooler does not work in this AVPC machine either. On the other hand, Bud seemed to say in the lima tape that a HRD4000 is coming to fix problems in the 3000. Maybe that is one of them. Also two cold, wet and hungry possums arrived at the back door at the same time, which is a guarantee of conflict.

END OF ARTICLE



# Techo Time

by Geoff Trott

## Power supplies and RAMdisks

I have been looking at two problems relating to power supplies which you may find interesting. The first was a power supply for a two way interface, you know, that series of connectors that allows up to two PEBox cards to plug into the side of the console. This is one of Lou Amadio's babies, and it is a terrific idea if you have a fully expanded AT card (disk controller, 32 Kbyte memory, PIO and RS232) and a RAMdisk as you then have a very powerful and compact system with almost everything you could want except for an 80 column card! Chris McCarthy has been trying to set up such a system for a while now and for one reason and another, he has been having problems with the AT card. We decided it would be best if I got the AT card working in the two way interface to make sure that all was working.

Chris is planning to use a console power supply board to power the two way interface. He was going to use the 12 volt supply for the low voltage supply for the PEBox boards. He has not the RS232 interface on the board which is the only use for the other voltages. The problem with this is that the 12 volt supply comes through a series regulator from a 23 volt unregulated supply. This means that the power dissipated in that regulator is proportional to the current flowing through the regulator, which may be quite large. It is also supplying a large voltage to the 5 volt regulator on the board, which is also going to dissipate a large power. The AT card draws about 0.5 amperes which means that 9 watts would have to be dissipated  $((23 - 5) * 0.5)$  in the regulators leading to a lot of heat.

I had a bit of a think and decided that it should not be too hard to make the 5 volt regulator on the power supply board produce a larger voltage. Since the voltage would be regulated, only about 7.5 volts would be needed for input to the 5 volt regulator on the card. This power supply regulator is a switching regulator which makes it very efficient as it converts the 23 volts to 7.5 volts (or 5 volts in the original power supply). The switching regulator consists of a uA723 voltage regulator IC driving two transistors which do the power switching. The reference input to the uA723 comes from a zener diode and resistor divider and comparator (TL331) which allows the voltage to be adjusted by removing links across a chain of resistors. There was not enough adjustment to get to 7.5 volts so I modified the circuit a bit. Firstly I opened circuited the zener diode (snip or unsolder one end of CR8) and shorted out the resistor in series with the zener ( $R1 = 1.2 k$ ). Then I soldered a 5.6 k resistor in parallel to  $R15$  (3.3 k). This gave a voltage out of the regulator of just greater than 7.5 volts. With this voltage as input to the 5 volt regulators on the AT card, they should only have to dissipate 1.3 watts and so run much cooler.

I then thought, why not use the other supplies on the power supply for the supplies for the +12 volt and -12 volt supply inputs in case Chris wants to put in an RS232 sometime. The supplies on the power supply are normally +12 volts and -5 volts. These are both 3 terminal regulators so they could easily be replaced by +15 volt and -15 volt regulators and then the power supply would be providing three regulated outputs of +7.5 volts, +15 volts and -15 volts which are just what the PEBox cards require. In the event, I only changed the -5 volt regulator to a -12 volt regulator so that the voltages are +7.5, +12 and -12. These would still be adequate as the supplies getting to the chips on the boards would be about +10 and -10 which is fine for RS232. Quite a neat solution to the power supply problem I thought.

The other problem related to the loss of ROS and data on a Horizon 3000 RAMdisk. RAMdisks are a potential source of problems as when the power comes on, the 5 volt supply must take over from the battery without extra pulses and the battery must take over from the 5 volt supply when the power is turned off. Problems abound with both cases and it is not easy to examine what is happening to find out how to fix it. I have looked at this problem with a number of different RAMdisks and come up with a few different solutions. Bud Mills has also had people spending time looking at the problem and has solutions also. Let me explain what some of the problems are.

When the power is turned on, the rectifier diodes allow pulses of current into the capacitors. The capacitors start with zero volts across them and the pulses of current only flow as long as the rectified voltage is greater than the voltage across the capacitor. The rectified voltage looks like the top of a picket fence and the maximum current that can flow depends on the impedances of the transformer and diodes as well as the capacitor. It is not likely that the capacitor will charge up fully with just one pulse. If the first pulse is a short one (because of when the switch is turned on), it may take three pulses of current before the capacitor is up to its operating voltage. After each charging pulse, the capacitor must supply the current requirement of the circuit on its own so it starts discharging. This means that the voltage into the regulator attached to the voltage across the capacitor may become large enough to allow the regulator to operate correctly and then drop below this voltage causing the regulator output to behave very erratically. This may happen once or twice, depending on the particular circuit, each time the power is turned on.

When the power is turned off, the voltage across the capacitors in the power supply discharge at a rate controlled by the size of the capacitors and the current drawn from them. This may take some time and so is quite slow compared with the normal operating speed of the logic. When power is first applied to logic or when power is removed from logic circuits their behaviour cannot be predicted. The faster this is done, the less time there is for unpredictable behaviour to occur but as I have explained, both on turn on and turn off, the power does not appear quickly nor disappear quickly. It is while the logic is in this in-between state that corruption of the contents of a RAMdisk can occur unless special action is taken.

The RAM chips themselves have inputs called Chip Enable (CE) lines, which are used to enable changes to be made to the contents of the chip and to read the contents of the chip. If these CE lines are not asserted, the contents of the memory chips will remain as long as the battery supplies enough voltage. The 8 Kbyte chip which holds the ROS has two CE lines, one asserted low and the other asserted high. In the RAMdisk, the asserted low CE line is used for selecting the chip by logic when the RAMdisk is enabled and the address range of >4000 to >57FF is accessed. The asserted high input is used to disable the chip when power is turned off. The 32 Kbyte or 128 Kbyte chips have only one CE line and it must be used for both purposes. In the earlier RAMdisks using the 32 Kbyte chips, a resistor was put between the battery supply and the CE line to hold these lines high when the power was removed. This relied on the logic attached to these CE lines not drawing current into their outputs when they had no power (sometimes not true). In the latest RAMdisks, the logic chips connected to the CE lines are CMOS types (HC) and they are also powered from the battery when the supply goes off. These CMOS chips also have a CE line which is asserted low and so must be held high when the power disappears. It is only one chip (or two) rather than all the memory chips, but the problem is the same.



# INSTRUCTION & HINTS FOR TI-WRITER WORD PROCESSOR

by Dick Altman, USA

In the light of the discussion on how power is applied and removed, it would seem that what is required is a circuit that waits for about 20 milliseconds after the power first appears and then enables the memory and also recognises immediately that the power has been removed and disables the memory immediately. I have designed a circuit that does that and applied it to the write enable line so that nothing can be written to the memory even if it is enabled. This has been used successfully on the MiniPE RAMdisk, which has its own set of problems. With the PEBox power supply, this has a very fast turn on as it is a much larger voltage than is required by the 5 volt regulators and the RAMdisks survive quite well.

The Horizon 3000 seemed to pose more difficult problems, judging from the changes to the circuit that have been made. All the RAMdisks use a clever means of disabling U11, the ROS chip, using the second asserted high CE line. A LED is connected from the 5 volt power line to a resistor to ground. The connection between the LED and the resistor is taken to the CE line of U11. An observant reader will recognise that as the way that Peter Mudie suggested connecting the clock chip to stop it losing its settings. What this does is make the voltage at the CE input to the RAM, 2.3 volts less than the voltage out of the 5 volt regulator. This means that the RAM chip is disabled until the output voltage of the regulator is at least 4.1 volts, by which time the logic should all be working correctly. The output of the regulator only needs to drop to 4.1 volts or so and the chip will be disabled, well before the other logic starts to misbehave. This is a good idea except for two cases. There are RAM chips which do not work properly with such a low voltage on the CE line and some LEDs have less than 2.3 volts drop across them.

The other CE line to the CMOS decoder chip requires the opposite polarity of voltage. It needs a low voltage when the power is on and a high voltage when the power is off, without drawing any current from the battery in that case. A resistor is connected from the battery supply to the CE pin and then an npn transistor with collector to the CE pin and emitter to ground. When the transistor is off, no current will be drawn through the resistor and the voltage will be high. When the transistor is on the voltage on the CE line will be close to zero. The transistor needs to be turned on after the power comes on and off as soon as the power is removed. If a resistor is connected from the base to the CE line going to U11 this should happen. In order to make the turn off as fast as possible and to delay the turn on, a resistor should be put between base and emitter as well. I added a 1 k resistor there and could not get the card to misbehave at my place. I hope it turns out to be the same when it gets back to its owner.

As a final comment, I would like to talk about batteries. It would seem that re-chargeable NiCad batteries would be the best to use for devices like RAMdisks. I am not so sure. NiCad batteries last best if they are discharged to almost flat and then charged up again. This is not the way you want the batteries on a RAMdisk to behave. I have also seen RAMdisks where something has come out of the NiCad batteries and damaged the tracks of the printed circuit. The current drain of a device like a RAMdisk, when it is on battery supply, is very small and all that is needed is a battery with a long shelf life. These batteries should last several years even if not re-chargeable. As the operation of a re-chargeable battery in a RAMdisk is one of constant charge and very little discharge, it will probably last less time than a normal battery. I have used three normal AAA size batteries for years and only have problems every two years or so. Lou was the cheekiest with batteries. He used to use the AA batteries his kids had flattened until they would not work their toys properly. They still had enough charge in them to last for more than a year as RAMdisk batteries. This just proves my point that it is the shelf life of the battery that is important, rather than the capacity of the battery.

**END OF ARTICLE**

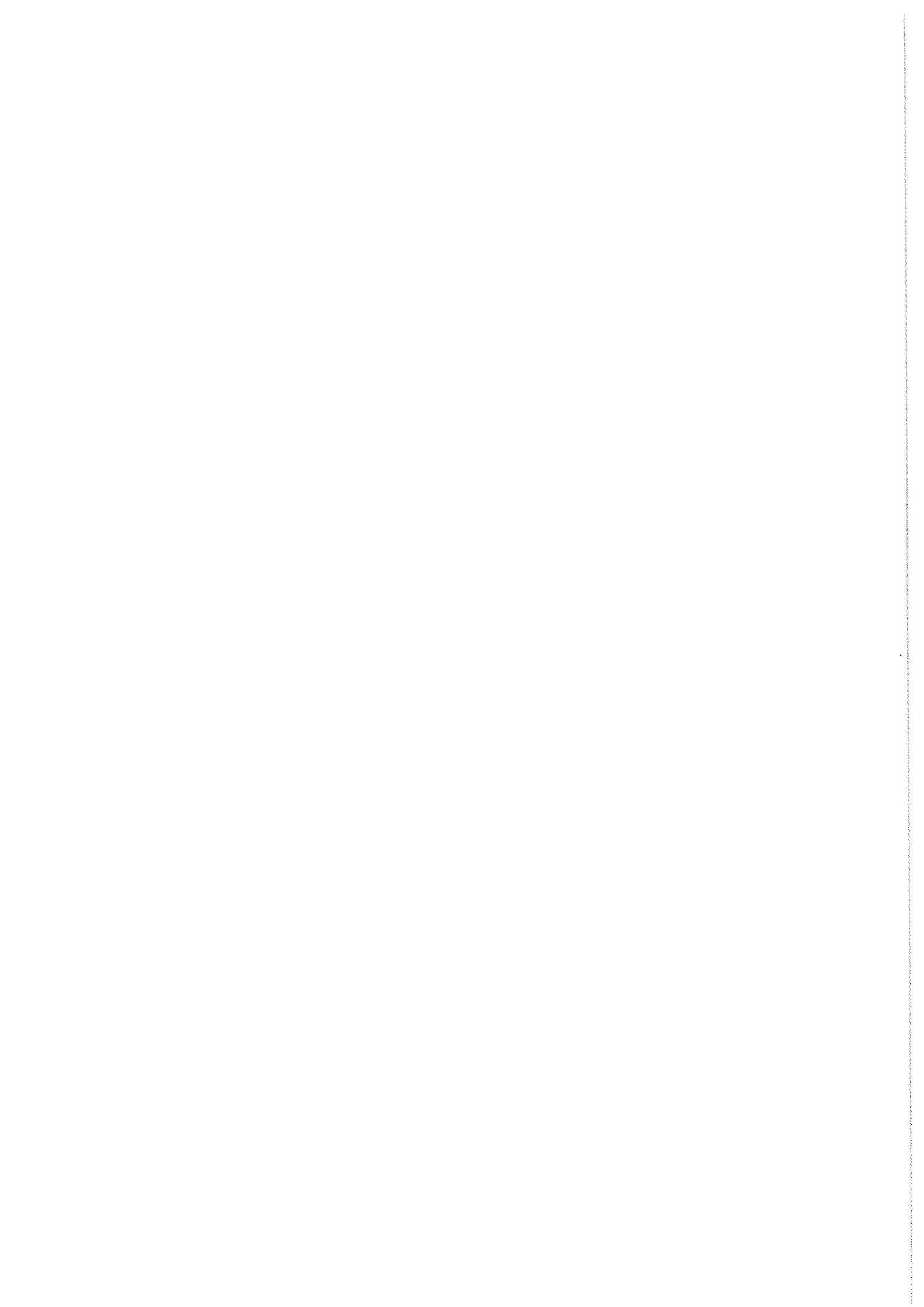
It CAN be mastered! It just takes perseverance and determination and a desire to do it. I have been using TI-Writer since January 1985 and I do not have it all yet, but I can use it to my immense satisfaction. This came from months of sitting with the large manual in my lap flipping pages back and forth until I had practically memorized the E&W thing! I was at the point where, when I had a problem I could say "Oh that is on page 146" or whatever. For instance: this article was done using TI-Writer, and I now do ALL of my correspondence with it also.

If you received the disk with this article, load it up in TI-Writer and call it up on the screen so that you can see which commands were used, and where they were used, to cause the different effects shown in this article. If you received the disk only, then you are not reading this unless you have already booted it up. It is suggested that you run off a printed copy, then reload this back up so that you can see the commands in use as you read the article. There are comments in the program just below or above the commands that do not show in the printout! This is another 'FREEMWARE' item. There is no price set for it. Feel free to pass a copy on to whomever wants it. If it will help only one or two people that are struggling to learn TI-Writer, I will be pleased. If you learn anything from it, and are inclined to fairness, send a few bucks when you can afford it to Dick Altman, 1053 Shrader St., San Francisco, CA 94117. There is no big deal if you do not; only your conscience will know. At least drop me a note and let me know it helped someone.

This is going to be loo-o-ng, but still much shorter than the 175 page instruction manual!

**FIRST RULE:** Read the TI-Writer Quick Reference card and reread it. Of course this means after you read this article. Do all of the operations shown on the card, at least once, even though you might think you will never need that particular one. You will find you have to open up the big manual probably, to accomplish some of the operations. After you have almost 'memorized' the card (literally!), then you will find yourself using TI-Writer almost exclusively and very seldom having to refer to the cumbersome manual. Personally I think the manual is poorly written.

You will find 3 'windows' (from left to right) to obtain the 80 columns (80 normal characters) width. Each window is 40 columns wide. The first one is from 0 to 40, second one is from 20 to 60, and the third is from 40 to 80. The first thing I do upon booting up TI-Writer is to set my limits to 37 characters wide. ~~If I take a whole window of 40~~ characters, it seems to crowd my screen, and I do not like to window back and forth to read my work. I do this by pressing "T" (for TABS), then press ENTER, then placing an "L" on the second dot, and an "R" on the 39th dot, then pressing ENTER again. (If your TV or Monitor is adjusted well, then the L can be left at the zero character position for 40 characters. ED). Now I find my cursor blinking at me from line E0001. Here is where I tell the printer what margins I want it to print my work within. It is also at this point that I select condensed type because I like it better than the normal size type, and I can get 132 characters per line if I wish. It just looks better in my opinion. I normally do this on line 0002 because I used 0001 to set up the formatting (margins, etc.) commands to the printer.



I find once in a while, some one command (never the same one twice) seems to falter. Just redo it. Sometimes I think some command must be there that is invisible (this is possible!) so when you run into an unexplainable problem, go back to your formatting command line(s), which are usually lines 0001 and 0002, put the cursor at the end of each of your commands then press FCTN and 1 and hold them for a couple of seconds to delete any possible typing errors that placed some sort of 'hidden' command in that line. (One possible problem is the presence of a space after a number in a dot command. ED)

Another good command to learn is the 'OOPS' command. Merely Control and the number one. This eliminates only your last change just now typed in, and returns your work to its former self (hopefully!).

Another good habit to get yourself into, is 'SAVING' your work every few minutes (or every few pages). Power glitches do occur from any power company. Either surges, or stumbles. Sometimes just an electric motor in your home (refrigerator, etc.) kicking in will cause a momentary change in the power supplied to your computer (you have seen your lights flicker). If you save your work every once in awhile, you someday will be glad you were in the habit. Especially if you have just put into the word processor—a—20,000—word—story.—The—power—glitch could cause you to lose it all! If you have been saving it on a disk, when that glitch occurs you will have all but a small part of it saved. When you save something to a disk, then come back to that same disk and save something else with the same name, it replaces the first item with the second. It does not become two separate items on the disk. Of course, if you are really a worry-wart, you will do the saving on two disks, alternating back and forth, just in case that glitch comes while you are in the act of saving your work.

When you wish to reload a file from a disk back into the word processor, it is easy! When you first bring up the word processor in the Editor mode, you are automatically in the command line. Just type LF (for Load File) and hit ENTER, then type in DSK1.(and the name you gave it) then hit ENTER again and wait a few seconds for the work to be loaded into your computer from the disk.

If you want a rough draft of your work on paper (I find it easier to proof than on the screen) just remove your commands for double strike or emphasizing to conserve your printer ribbon. It will not be so easy to read, unless your ribbon is new, but it will be done faster, as well as not using up ribbon ink unnecessarily.

In the book you will find two methods of going to the disk, then to your printer. Printing should be done from the disk, not from the computer. You will find a command of 'Print File'. That is not the one I use! The one I have become accustomed to using may take a few seconds longer, but it is the one I learned first, and I have just stuck with it. It is as follows. After I have finished typing my letter or whatever, return to the command line with FCTN 9, there type a Q (for Quit) hit ENTER, then S (for Save) and ENTER, then DSK1.TERRY or whatever name I want to give the file instead of TERRY, then ENTER. I usually use a short two or three character name. I have even been known to use £1, or £2, or something like that (the file name cannot be more than 10 characters long, and you can not have any spaces in a file name). Then, after the work goes from the computer to the disk, you can either print it now or sometime next week. The command to go to the printer at this

point is like this: Q (for Quit) ENTER, then E (for Exit) and ENTER again. This takes you back to the master menu. This time, you select £2, or the FORMATTER. After it comes up, you have to type in DSK1.(filename) and hit ENTER. Then you have to type in the command telling it to go from the disk to the printer, instead of to the screen. (With the use of DISKO or some such assembly language disk repair program, you can insert the command to your printer so that it is a default just like all the other selections on the screen. It is in 'FORMA1' of your TI-Writer disk.) Without knowing what kind of printer you have, I can not give ~~exactly the correct command here, but it will~~ be something like this: PIO.LF or RS232.BA=4800.LF, then you will have five more choices, mostly for which you will just press ENTER for each of them. Perhaps you might wish more than one copy, so on the correct one you would punch in that number. Be sure your printer is turned 'on' before hitting the last ENTER,(the one that says "PAUSE AT END OF THE PAGE?) because you will be printing immediately afterwards.

### Printing Documentation Files

Here is a short file that you can save as HEADER on your TI-Writer (or TK-Writer) workdisk which will make your documentation files neater and easier to keep. Here are steps to use:

1. In Edit mode LF DSK1.HEADER
2. LF 10 DSKn.docname (will put the documents after the HEADER.)
3. SF DSKn.newname
4. Use Formatter to print 'newname'

The resulting printout will be spaced to allow three ring punching, have wider margins, and be numbered consecutively from Page 2 on.

HEADER

```
.CO file to print DV80 instructions in
.CO Elite with room to 3 hole punch
.CO codes are for Gemini 10X
.CO next line changes to Elite print
.TL 35:27,66,2
E
.CO next line sets Left Margin to 12
.TL 36:27,77,12
$
.HE Page %
.PL 62
```

(Tom Rhodes—Bluegrass Area 7/85)

### Mail Merge

Several people have asked why they cannot get the mail merge option of TI-Writer to work. I have an idea that will probably explain 99.9 percent of the problems. When you create the value file for your text file, be sure that the left margin is set at the extreme left position. If any spaces are placed in the file before the variable number, mail merge will not work. You can use either of two methods of accessing a value file. The first is to answer Y to the use 'mailing list' Option from the prompt screen. Then enter the value file name. The other is to answer Y to the prompt, then just press ENTER when prompted for the file name. In this case, you must use the ".ML filename" file management command in your text file. ~~If you answer yes to the use mailing-~~ list prompt and then enter a value file filename, the formatter ignores any .ML command in the text file. Try it, it should work.

by Arthur Author (Airport Area CC)

Many times I have wondered about what TI-Writer would look like if there was only a single window to look at. If you have too, then this article is for YOU!

Let us begin by asking ourselves about how we use TI-Writer. If composing our documents on the screen is typical, then it is a distinct advantage to not have to either scroll between three screens or to have to print the document to see what we just said in the last sentence or paragraph.

Scrolling makes a 40 column screen one third the size of an 80 column screen, which we all know is not true, but it does make reading difficult. It also adds extra steps and sure does chop down a few trees in the process.

Well, there is another alternative. The secret is to set the left margin at 0 and the right margin to 39. The next requirement is that the line numbers are turned off (FCTN 0). This gives you a full 40 column screen and with the word wrap capabilities that we appreciate so much. To test the space requirements on the diskette, the following test was run.

Three files were created with the same data (all 'x's). The first had the margin set at 21 and 60, with 10 paragraphs of 6 lines (240 characters of data). The second file had the margins at 1 and 80 with 10 paragraphs of 3 lines (240 characters of data). Comparing the sectors used, the first file required 19 sectors, while the second only used 12.

This represents a 58% loss of capacity on each floppy and would raise a question about the end justifying the means. Then the light bulb came on. The only reason the margins were set to 21 and 60 was to get a full 40 column screen of data. In truth the only purpose was to eliminate the line numbers on the left of the screen.

Without considering the data storage requirements it would appear an optimal choice. Hence the third file was created using margins at 0 and 39 and with the line numbers turned off. With 10 paragraphs of 6 (240 characters), I was ready to use SD (ShowDirectory) to tell me the answer. Surprise! The sectors were not 19 or even 13. It required 12 sectors. The same as with a full line length at 80 columns. If this fact surprises you as much as it did me, you may wish to set up a different test and verify my results. In the mean time, I plan to do all of my composing on a 40 column screen, and then use the Formatter to adjust the printed page back to 80 columns.

Who said you cannot get anything for free. YOU now have a choice of 40 or 80 column mode without feeling guilty about requiring more disk space.

(George Brandt, Southwest 99ers Nov/Dec85)

10X/zero Program

This is an improvement on the slash zero for the Gemini 10X printer. It can be run in XB or BASIC. You turn on your printer then run the program and do not turn off your printer. You can shut off your computer, as it will stay in your printer until you turn it off.

```
50 REM WRITTEN BY DANE R. HEATHERINGTON
60 REM FOR SYSTEM WITHOUT 32K MEMORY
70 REM USE 110 THRU 140
100 ON BREAK NEXT :: CALL CLEAR :: CALL INIT
110 OPEN E1:"PIO"
120 PRINT E1: CHR$(27)&CHR$(42)&CHR$(48)&
CHR$(0)
130 PRINT E1:CHR$(27)&CHR$(42)&CHR$(1)&
CHR$(48)&CHR$(0)&CHR$(92)&CHR$(34)&
CHR$(0)&CHR$(81)&CHR$(8)&CHR$(69)&
CHR$(0)&CHR$(34)&CHR$(29)
140 PRINT E1:CHR$(27)&CHR$(36)&CHR$(1)
150 CLOSE E1 :: CALL PEEK(2,A,B) :: CALL
LOAD(-31804,A,B)
```

When you are using the "C" command, prior to typing in your full text, then it is easy to just

1,10,10

several times to get the lines in. But when your text is all done and you want to add a separator line (for example), then you should start your copying with the last instance. Reason being for this is, you will have written down all the line numbers where you want to put the separator, but after the first copy, the line numbers have all been advanced. If you start at the last instance, then the line numbers are still the same earlier in your text.

(Paul Sparks, LA 99ers)

END OF ARTICLE

DEFINE YOUR OWN PRINTER CHARACTERS

by Steven Shouse, (TI-RUG 1.8) adapted from PUG Peripheral by Norman Rakke

Have you ever wished that your printer had some special character or symbol? Well, if your printer has dot graphics capabilities and allows you to mix text and graphics on the same line, then you can use TI-Writer to create characters which you define, then print in documents that you create. Before going into detail, you should note that the information in this article relates specifically to the TI 99/4 printer; however, the control codes used are standard Epson codes and the technique should work with most Epson compatible printers. An attempt was made to verify this technique on the Star Micronics Gemini 10, but the attempt failed because the Gemini 10 will apparently not allow graphics and text to be mixed on a line.\*\* If you have some other printer which has the above mentioned capabilities, you can still do what is described below; however, you should read your printer manual to determine how your printer does each of the things mentioned. If anything is done in a different manner, you will have to take that into account.

First of all, you must make sure that your printer is ready to receive graphics data. You may have to remove the cover from your printer and change the position of a DIP switch so that the printer can receive 8 data bits. Check your printer manual to make sure that the graphics mode works properly. Finally, if your printer is connected to the serial port of the RS232 interface, you will need to include DA=8 in your file description EVERY time you use the printer.

Printer graphics consist of one or more columns of dots. There are a total of 480 such columns across a page. Each column is 8 positions high and a dot can appear in any of the positions. Each position has a data value associated with it as shown in the figure below. The data representing a particular column is simply the sum of the data values for all the positions where a dot is to be printed.

```

+----+
|128|
+----+
| 64|
+----+
| 32|
+----+
| 16|
+----+
|  8|
+----+
|  4|
+----+
|  2|
+----+
|  1|
+----+

```

For example, to print a column where only the top dot is printed would require a data value of 128. A column in which both the top and bottom dots were would require a data value of 128+1, or 129. A column which had all 8 dots printed would require a data value of 128+64+32+16+8+4+2+1, or 255.

Now let us create our own graphics character and see how we can incorporate it in a document prepared with TI-Writer. Let us make an arrow pointing up as our special character. The normal characters built into the printer are as wide as 6 columns of graphics (480 columns/80 chars), so let us make our character the same size. It is helpful to draw the character on graph paper, so let us do that.

```

+---+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+---+
| | | * | | | |
+---+---+---+---+---+---+
| | * | * | * | | |
+---+---+---+---+---+---+
| * | | * | | * | |
+---+---+---+---+---+---+
| | | * | | | |
+---+---+---+---+---+---+
| | | * | | | |
+---+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+---+

```

The data for the columns from left to right is 16, 32, 126=(64+32+16+8+2), 32 and 16. (You might find it useful to know that the normal text characters of the printer do not use the column on the right (to prevent characters from running into each other) or the bottom row (except in lowercase characters with descenders). To send the graphics data to the printer, we first need to send a control code. For normal graphics mode this is in the form of ASCII codes 27;75;n1;n2. The codes n1 and n2 define the number of bytes of graphic data to be transmitted. This represents the number of columns of dots which will be printed. If N is the number of columns of graphics to be printed, then n2 is the integer result of N/256, and n1 is N MOD 256, or the remainder of N/256. For our special character, n2 is 0 and

n1 is 6. The graphics data follows immediately after n2. The complete string of ASCII values needed to print our special character is therefore 27, 65, 6, 0, 16, 32, 126, 32, 16, 0. We will create the special character by using TI-Writer TL command. We will use some character we will not be using, such as ~, and TL it to the string of data we define above.

We can now use our special character in a document. If we were writing directions for using a program where pressing the E key moved something up on the screen, we could write the following:

To move up ~ Press E

If we now print this short file using TI-Writer's Formatter, we will get the arrow in the tilde's location.

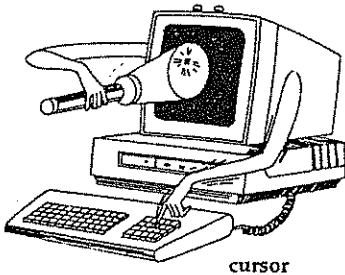
Using special characters which are six graphics columns wide allows you to still be able to use the .AD and .CE comands even if special characters are present in the text. If you are not going to use either of these commands, you can make your characters of different widths than 6.

**CAUTION!** Using the method described, you can design almost any character you might desire. Unfortunately, the values 8, 12, and 13 cause problems which disallow their use for graphics data using this method. You may have to modify your character to avoid these problems.

\*\*Star Micronics printers can be used to perform similar tasks by accessing the block graphics and special character set by means of the TL command or the downloadable character set (available on some machines including the X series) may be used to design your own characters. The procedure is similar to that explained above, but there are some significant differences. See Chapter 8 in the Gemini 10X/15X Users Manual for details.

As a matter of fact, this version of the original article was printed using the Gemini 10X. ~~The up arrow was achieved by using the TL command to print character 164 from the Gemini's block graphics and special character set. Your printer may have similar capabilities. Check your manual. One caution, the downloadable character set and special character sets cannot be used simultaneously.~~

**END OF ARTICLE** 



cursor



## SECTION FOR IBM COMPATIBLES

THE FOLLOWING ARTICLES ARE REPRINTED  
FROM OTHER MAGAZINES WITH THEIR PERMISSION.

### Archiving programs

*What is the best file compression/archiving software to use? I have seen ARC, ZIP and LZH file extensions, and I understand that they all use different techniques to compress data? Are all of them incompatible with one another?*

*C. Williams  
Jeeralang, Vic.*

As is the case with most decisions in life, there is no simple answer to this, owing to the rather checkered history of PC archiving programs in general. The first PC archiving package to appear on the scene was Systems Enhancement Associates' ARC. This not only provided a means of compressing files to make them smaller on BBS's hard disks and faster to transmit, but also allowed several files to be combined into a single 'archive', so that a program file, its overlays and documentation could be distributed as a single unit without the danger of a vital part of the program going astray.

Since the appearance of SEA's ARC, archiving files on bulletin boards became almost universal. PKWare, an American company, thought that it could do a better job and released PKARC, which was faster than ARC in compression and decompression. However, in an effort to standardise on the ARC file format, PKARC used the same file compression techniques (with one addition) and file formats, so that files created by the two programs were compatible.

Since PKARC was faster than ARC, and had an additional compression algorithm built-in, it became the preferred compression program. SEA didn't like this, and in typical US style, took PKWare to court, although the two ultimately settled out of court. The settlement basically meant that PKWare could

no longer market products which used SEA's compression techniques, and it could no longer use the term ARC. Yet another standard bites the dust!

PKWare changed PKARC and PKXARC to PKPAK and PKUNPAK, using the PAK file extension, and later changed to PKZIP and PKUNZIP. The latter programs were even faster than any of their predecessors — for a full report, have a look at John Hepworth's 'IBM Underground' column in the May 1989 issue.

A relative newcomer is LHARC, which creates files with the extension LZH. This is a favourite among BBS Sysops because it creates the smallest files of any compression program, thus maximising valuable disk space and reducing file transfer times to an absolute minimum. The penalty is speed — it's much slower than ZIP or PKARC.

So the real answer to your question is 'it depends'. If you are compressing files for transmission over expensive phone links, or to cram the absolute maximum amount of data on a disk, then LHARC is probably for you. However, if you are looking for a quick way to backup your work directory at the end of each day before going home, then speed is probably more important, so PKZIP would be a better choice.

By the way, if you are fortunate enough to have a modem with MNP-5 data compression, don't expect it to transmit compressed files any faster than normal — MNP-5 is just another file compression system, and files which are already compressed don't compress well a second time, so the transmission speed will not improve much, if at all. Note that the slight increase experienced with MNP-4 will still occur with compressed files, since this is achieved by removing the start and stop bits before transmission, not by compressing the file.

### Motherboard memory

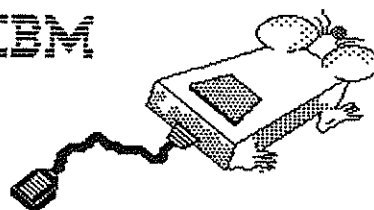
*I have a rather elderly AT (about 3 years old), which I put together myself from boards acquired from a variety of sources. The motherboard's RAM uses 41256 chips, giving me 1Mb of storage, but I would like to increase that to 4Mb, without using an extra memory board, if possible. Can this be done? Presumably I will need 411000 chips (if that's the right number), but the motherboard manual only mentions 4164 and 41256 chips.*

*E. Catling*

The simple answer is 'No', it can't be done. Not easily, at least. The motherboard was obviously designed before 1Mb chips became widely available, and so no provision was made for them. So, unless you feel like cutting tracks and soldering wires to the motherboard, forget upgrading it, and get yourself a memory board.

There is another advantage of using a memory board, especially on a '286-based system. Most boards can be configured as either extended or expanded memory, so that you can select the appropriate type of memory for the applications which you are running. While EMS emulators are available, which make extended memory on a '286 look like expanded, they are slow, since the processor needs to be running in protected mode to access the extended memory, and switching back to real mode necessitates resetting the processor, which wastes a considerable amount of time.

IBM



# The Importance of PC Memory

Memory is a crucial factor in determining PC usability and performance. Without enough of the right kinds of memory, your PC simply can't run some of today's most popular software - from graphical environments such as Windows 3.0 to sophisticated applications like Lotus\*1-2-3\* and pop-up utilities like Borland's sidekick\* Plus. A PC's ability to do multitasking also depends on having plenty of memory - without it, you're limited in the number of applications you can run simultaneously.

Without enough of the right kind of memory, you may find that you can run certain programs but can't take advantage of their advanced

features. The amount of data each application can handle is limited as well, so you may have to scale back the size of your database or the level of detail in your CAD drawings to avoid getting "Out of Memory messages".

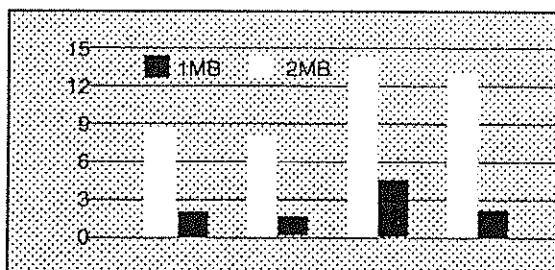
Even system performance suffers. Instead of quickly accessing data from memory, the CPU must go repeatedly to disk, slowing the processing of application information.

Fortunately, solutions are readily at hand. Hardware products make it easy to add memory to the PC - by filling all the memory sockets on the system-board (motherboard) or by installing one or more

memory boards in the PC's expansion slots. Solutions also include a range of software products that help make the most of PC memory.

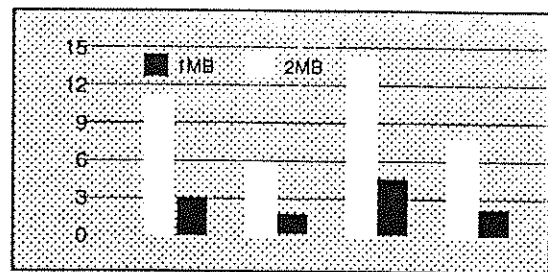
To decide which memory solution is right for you, it helps to understand the essentials of PC memory. Why is memory important? How does DOS manage memory? How much memory do you need? What are the advantages of various memory products? This guide answers those questions and offers suggestions on what to look for when you shop for memory.

## A Megabyte Makes a Difference Above™ Board Memory



Memory has a direct effect on performance. In this test, adding a single megabyte of memory cut the time to switch from one Windows 3.0 application to another by 72 to 83 percent. The additional memory also brought up to a fourfold speedup in operations within a given application. The benchmark used an IBM PS/2 Model 50, which comes with 1MB of memory, and an Above Board MC memory board with 1MB of memory

## Matched™ Memory



Adding a megabyte of system-board memory shortened the switching time between applications and improved the performance within individual applications. We added a 1MB Intel Matched Memory Module to the system-board of an IBM PS/2 Model 55SX, which comes with 2MB. The results: recalculation of an Excel spreadsheet is 4X faster, scrolling from top to bottom on a Word® for Windows document is 5X faster, and switching from one application to another is 3 to 4X faster

# PC Memory Basics

Memory - or random access memory (RAM) - is the circuitry that stores a software program while it is active, as well as the data you enter into an application. If you're a Windows 3.0 user, for example your PC memory might hold the DOS operating system, Windows 3.0, several multitasking applications and a data file for each active application. Also resident in RAM would be pop-up programs such as clocks, calendars and notebooks; these terminate-and-stay-resident (TSR) utilities remain in memory whether they're active or not.

access 16MB of memory. Today's 80386 and 80486 processors have a 32-bit architecture and can directly address four gigabytes (four billion bytes) of memory. When they address their full memory range, these processors run in protected mode-which enables software to access this memory as one contiguous block. To provide compatibility with software developed for previous processors, the 286,i386™ and i486™ microprocessor can also operate like an 8088 or 8086, using the

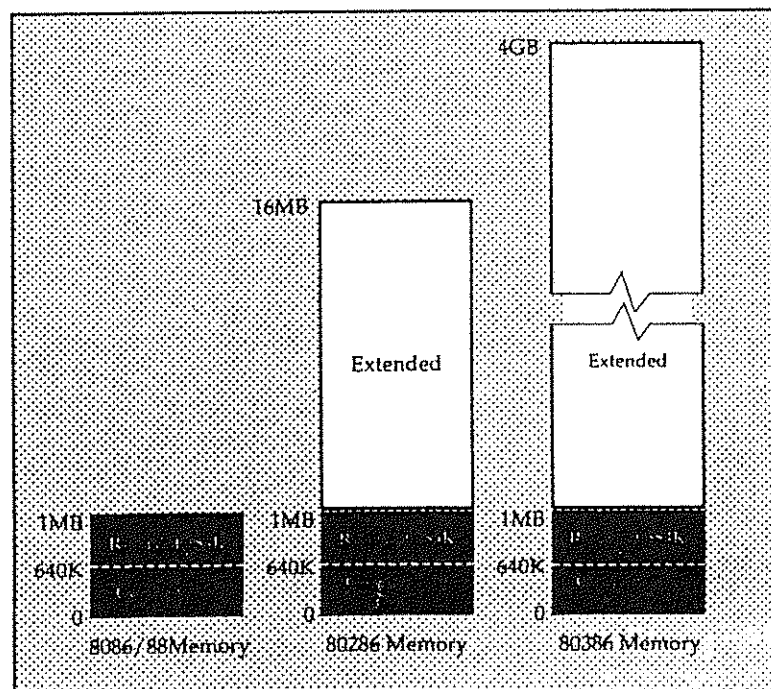
earlier processors smaller address range.

Out of the original one megabyte of memory, PC architecture reserves the top 384K for system functions, leaving 640K available for the operating system, applications, data and device drivers (which control a printer, mouse or other pieces of equipment). This 640K is now called conventional memory, and the upper 384K is called high memory or reserved memory. The additional 15MB and 4GB that the 80286 and 80386 processors can address is

Each block of PC memory has an address that allows the PC's central processor (CPU) to access the information it contains. The Intel 8086 and 8088 microprocessor that powered the early PCs had a 20bit address bus, from which the processor could access (2<sup>20</sup>) unique addresses. These systems could directly access up to a megabyte (1,048,576 bytes) of memory.

Microprocessor technology has advanced dramatically since the dawn of the PC. The Intel 80286 processor, used in AT\*-class machines, had a 24-bit address bus and could directly

➔ The microprocessor at the heart of a personal computer affects how much memory the PC can directly address.



# The DOS Connection

Memory management also depends on the PCs operating system. While advanced operating systems like OS/2\* and UNIX\* take advantage of the 386s memory features, most PCs run DOS. DOS was designed in an era when PCs were single-user, single-application machines. The PCs of this era had 64K of memory, and a megabyte seemed like all anyone would need.

Today, multitasking is common, systems are frequently networked, and CPUs can address vast amounts of memory. But, except for a few specially-written programs such as VDISK, DOS still recognises only the original 1MB.

Fortunately, there are several

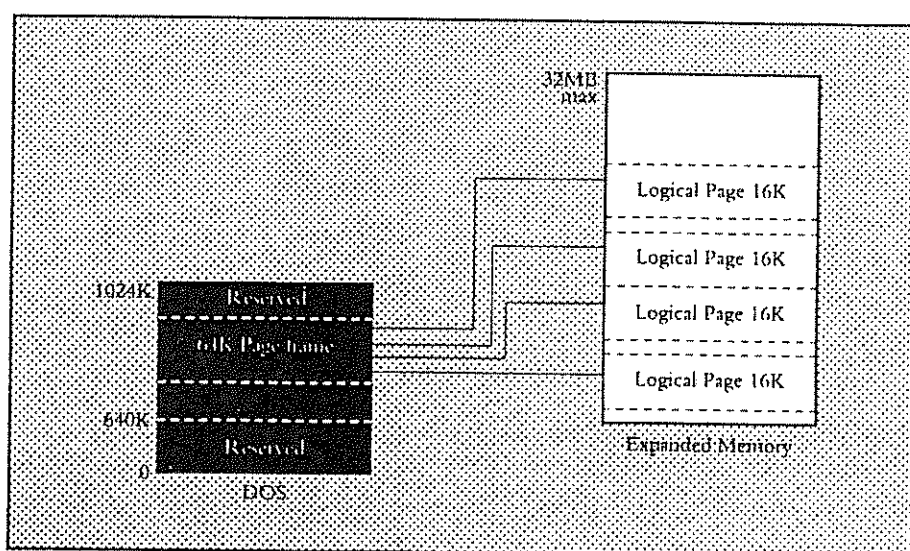
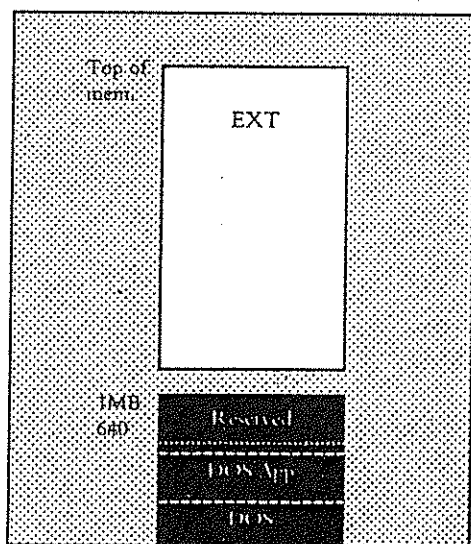
ways around the DOS' 640K barrier. One is to use a software environment such as Windows 3.0, which creates its own memory management system and addresses the full range of available extended memory. Another approach is the Expanded Memory Specification.

## Expanded Memory

First developed in 1984 by Lotus, Intel and Microsoft, the Expanded Memory Specification (EMS or LIM EMS) is now in its fourth revision (EMS 4.0). It is compatible with 8086/8088, 80286 and 80386 PCs, and hundreds of applications have been written to take advantage of it.

EMS defines a way to access up to 32MB of memory that lies "outside" the ordinary memory map. EMS accesses this additional or "expanded" memory by specifying a special, 64K area called a page frame in the reserved memory area above 640K. EMS then uses a technique called paging or bank switching to access 16K-size chunks or pages of expanded memory.

For 8086/8088- and 80286-based systems, special software called an expanded memory manager controls how memory is accessed. Expanded memory managers are provided with most EMS memory boards.



# How much Memory Do You Need?

Most PC users find they need more than the minimum amount of memory. Graphical operating environments like Windows and DESQview require memory both for themselves and for the applications running within them. Many applications require large amounts of memory:

- ▶ Windows 3.09 requires a minimum of 640K conventional and 256K extended. Efficient multitasking calls for having 2-4MB in your system.
- ▶ Lotus 1-2-3 Release 2.2 requires a minimum of 640K of conventional memory and uses as much expanded memory as is available.
- ▶ WordPerfect\*5.1 needs at

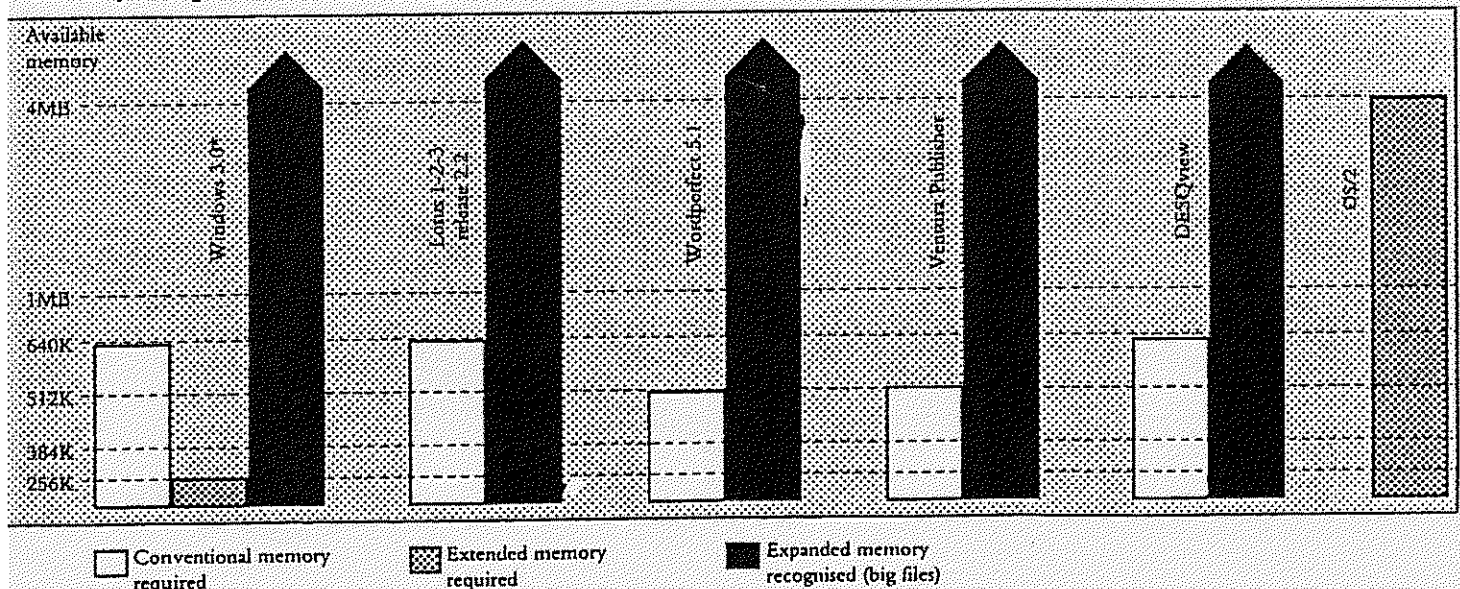
least 384K of conventional memory, and 512K is recommended for better performance. Version 5.1 can use as much expanded memory as is in the system.

- ▶ Ventura Publisher\* calls for 512K of conventional memory and recommends 640K if you create documents longer than 20 pages. VP can use all the expanded memory that's available.
- ▶ DESQview recommends 640K of conventional memory and takes advantage of expanded memory for multitasking performance.
- ▶ dBASE\* IV 1.1 requires 640K of conventional memory. When extended

and expanded memory are available, dBASE uses them for caching, to speed database computations.

Applications aren't the only consideration. Devices such as scanners and fax boards typically need 100K of memory for driver support. If you're linked to a local area network, you may need as much as 120K for network device drivers. Add that to the 60K that DOS requires and you're left with less than 500K of conventional memory - not enough to run a large graphics program, for example, and print your file over a networked printer. And if you decide to move to OS/2, you'll face additional memory requirements: OS/2 needs 3MB just to load, and 4MB is recommended.

Memory Requirements of Popular Programs





Regional Group Reports

Meeting Summary For OCTOBER

Banana Coast	10/10/93	Sawtell
Central Coast	09/10/93	Saratoga
Glebe	07/10/93	Glebe
Hunter Valley	09/10/93	
Illawarra	12/10/93	Keiraville
Liverpool	08/10/93	Yagoona West
Northern Suburbs	28/10/93	
Sutherland	15/10/93	Jannali

BANANA COAST Regional Group  
(Coffs Harbour Environs)

We never miss meeting at Kerry Harrison's residence 15 Scarba St. Coffs Harbour, 2 pm second Sunday of the month. Visitors are most welcome. Contact Kerry 52 3736, Kevin 53 2649, Rex 51 2485 or John 54 1451.

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The meetings are usually held on the second Saturday of each month at members homes starting at 3:15 pm. Check the location with Geoff Phillips on (049) 428 176. Note that after 9:00 pm this number is used for the ZZAP BBS which includes TI-99 information. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Tuesday of each month after the TISHUG Sydney meeting (except January) at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Geoff Trott on (042) 29 6629 for more information.

\*\*\*\*\*  
\* LIVERPOOL Regional Group \*  
\*\*\*\*\*

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home) After 9.30 PM or at work (02) 708-1987 Liquorland Yagoona for more information.

\*\*\* ALL WELCOME \*\*\*

8th October 1993	*****
34 Colechin St	* Pre Scanning *
Yagoona West 2199	* Programs *
	*****

12th November 1993	*****
34 Colechin St	* Pre View of *
Yagoona West 2199	* some NEW games *
	*****

Bye for now Larry.  
Liverpool Regional Co-Ordinator

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02) 918 8132. Come and join in our fun.  
Dick Warburton.

SUTHERLAND Regional Group

The August meeting was fairly low key, with only three members in attendance, Derek, Herbert and myself. Topics included Gofer, following a quick tutorial from Larry Saunders at the last main meeting. We also brushed the cobwebs off Multiplan and became much more proficient in its use by the end of the meeting. All this practice will be put to good use when we start to summarise the News Digest indices for 1993. Another popular subject was a review of the latest TI Artist fonts.

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young.

TISHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. The October meeting will be the last one at this venue. Check the September issue for more information. Regular items in the TND include news from the directors, the publications library, the shop and demonstrations of monthly software.

OCTOBER MEETING - 2nd OCTOBER

\*\*\*\*\*

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

November	10th October
December	14th November

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

