

THE NATIONAL NINETY-NINER

VOL I - NO. 11 - OCTOBER, 1984

MONTHLY NATIONAL NEWSLETTER OF THE

99ERS USERS GROUP ASSOCIATION
3535 SO. H ST., #93
BAKERSFIELD, CALIF. 93304
(805) 397-4361
DON VEITH - EDITOR/PRESIDENT

CREATED FOR TI 99/4A HOME COMPUTER OWNERS AND USERS GROUPS

TIGERCUB SPECIAL OFFER!!!!!!

Individuals and Users Groups cannot purchase a subscription to Tigercub's Tips. The Tips are available only for the exchange of a Users Group newsletter. Jim Peterson has a special offer for anyone interested in obtaining their personal copy of Tigercub's Tips #1 to #14. The Tips are available on disk for a cost of \$15.00 (includes all shipping and handling charges). The Tips disk includes fifty (50) programs, routines, and files. You may obtain the Tips and programs from Jim at the address listed below. While you are ordering the Tips and programs, why not include an extra dollar (\$1.00) and obtain a copy of his Catalog. Credit is graciously provided for this charge by deducting \$1.00 from your first order of programs. Contact Tigercub at:

TIGERCUB SOFTWARE
156 COLLINGWOOD AVENUE
COLUMBUS, OH 42313
(614) 235-3545

MORNING STAR SOFTWARE DISCOUNT OFFER TO USERS GROUPS

Morning Star Software contacted our organization to release news of a discount to Users Groups and their members. A ten (10%) percent discount is being offered off the \$595.00 list price of the CP/M card (\$535.50 discount purchase price) AND software carried by Morning Star. Each Users Group receiving this newsletter is requested to announce the offer to their members. The steps listed below outline how to take advantage of Morning Star Software's discount offer on the CP/M card and software sold by the firm.

1. Each Users Group is requested to collect the funds for each CP/M card being ordered by their members.
2. The funds collected should be forwarded on a single Users Group check to Morning Star Software.
3. Each purchaser's name, address, and home telephone number should be forwarded on a separate list. This information will be used for warranty and shipping purposes.
4. A shipping charge of \$5.00 for each card ordered should be added to the amount of the funds forwarded to Morning Star for each individual order. Each order will be shipped directly to the purchaser by Morning Star Software from the list provided in # 3 above.
5. **IMPORTANT** - Each purchaser has been requested to specify whether the TI or CorComp Disk Controller Card will be used with Morning Star's CP/M Card. The CP/M Card is compatible with the CorComp Disk Controller Card effective November 1, 1984. The CP/M card operates in a single-side, single-density Osborne 1.0 disk format. It cannot utilize the CorComp unit's double density format at this time.

The software available for purchase from Morning Star is listed below. A 10% discount is also available on the software to Users Group members.

<u>SOFTWARE TITLE</u>	<u>LIST PRICE</u>	<u>DISCOUNT PRICE</u>
C-BASIC	\$150.00	\$135.00
BUSINESS MASTER	\$295.00	\$265.50
PERSONAL PEARL	\$295.00	\$265.50
SUPER WRITER	\$295.00	\$265.50
SEEKEASY	\$ 67.00	\$ 78.30

Remember, the discount from Morning Star is only available through a Users Group. Morning Star Software has not placed an expiration date on the discount offer at this time. 99'ERS USA INDIVIDUAL SUBSCRIBERS should contact us at the letterhead address if you wish to take advantage of Morning Star Software's discount offer. User's Groups and their interested members may contact Morning Star at the address below:

MORNING STAR SOFTWARE
4325 SW 109th AVE.
BEAVERTON, OREGON 97005
(206) 646-2412

BULLETIN BOARD TIPS

This new byline, **TOURE DE BOARDS**, will include tips of interest from Bulletin Boards around the United States. The hints and tips will cover all aspects of the 99/4A. Each article printed will include the name of the Bulletin Board where the Tips were located. Ron Albright and Don Veith are combining their efforts to provide these Tips for subscribers and Users Groups. We will endeavor to provide the Tips on a monthly basis as space is available in the newsletter.

NATIONAL X/B LIBRARY

THE 99'ERS USERS GROUP ASSOCIATION is establishing a National Library for Extended Basic programs. Programs submitted to the Library must be forwarded with a Public Domain Statement enclosed (see last page of newsletter for a copy of this form). It will take the Librarian, Curt Purdy, about three (3) months to organize and catalog the Library. Any individual who wishes to donate program(s) to the new X/B National Library should contact Curt at the address listed below:

CURT PURDY
NATIONAL X/B LIBRARIAN
2314 MIDWAY DRIVE
PHOENIX CITY, AL 36867

CORCOMP'S 9900 MICRO-EXPANSION SYSTEM

For any of you who have used or viewed this new product, you must admit it is compact and yet will be capable of replacing a complete TI Expansion System (Disk Controller, 32K Expansion Memory, RS-232) when the upper board becomes available. Our local Users Group TEX-BUG has been using the RS-232 capability for demonstrations of Computer Banking and access to The Source. The finest feature of this unit is its compact size which makes transporting the unit a simple task. In fact, when the upper board becomes available (Disk Controller and 32K Memory), it will make the 99/4A a truly portable computer with full system capabilities. I am not sure this factor has been discussed previously by CorComp or other publications. I plan to transport and use this unit for demonstrations at our Users Group and other functions. As a second system for 99/4A owners, it will be very convenient allowing other family members to utilize a full system without disturbing the primary user who always has to wait in line to use the Expansion System. This will allow the expensive toys the primary household user has installed in TI's Expansion System to be left alone without having work in progress disturbed. I am personally looking forward to the release of CorComp's upper board to take the workload off my primary system. Let's hope it will be available in the near future. I plan to test its portability and report further on this subject. Don Veith, Editor

AN INTRODUCTION TO THE MORNING STAR CP/M CARD

By Don Veith, Editor

A series of articles on the CP/M Card manufactured by Morning Star Software will be appearing in this newsletter. The articles will review my experiences working with the card and CPM. The initial experience working with the CP/M card is quite interesting. It is unique not to load a Disk Manager Cartridge to format or copy disks. The disk drives are identified by a letter of the alphabet rather than a number used by TI.

Drive one (1) is identified as "A" with subsequent drives denoted as "B" and "C". The operator is able to select a menu and arrange the disk drives in an operating configuration of his choice. Thus Drive #1 (inside the Expansion System) does not have to be the location of your system disk. Once the system disk is loaded, what appears on the screen is the current drive letter and the greater than symbol (>) in this format A>. The > (greater than) symbol is used after the current disk drive in use, A>, to indicate where a CP/M command should be entered.

The CP/M board's 8085 Intel Central Processor Unit (CPU) seizes control of the 99/4A system prior to the TI-9900 processor and retains control of the system until the operator tells the 8085 CPU to relinquish control. The initial TI format screen appears with a slight modification when the CP/M card is installed in the Expansion System. The operator is given the choice of (1) the TI Disk Operating System or (2) the Morning Star CP/M Processor. The choice of (1) does not affect system performance in any manner. Choosing (2) powers up disk drive #1 where the CP/M unit seeks its system diskette. From this point, several programs are loaded into memory and a menu appears.

The initial menu that appears on your screen is shown below:

```
CP/M-80 Version 2.2  
Copyright 1981 by Digital Research, Inc.  
ANSI X3.64 Terminal (config 000)  
Copyright 1984 by Morning Star Software
```

A>

From this point the disk routines may be called up by typing commands to perform the desired function. BACKUP is used to make a backup copy of a disk while INIT initializes an unformatted disk. Other built-in commands are DIR, REN, ERA, SAVE, TYPE, and USER. DIR gives a directory of programs on a disk while REN renames a disk file and ERA erases a disk file. SAVE should be self-explanatory. TYPE displays the contents of a file on the screen and USER selects user area (0-9). The CP/M card uses the Osborne 1.0 disk format and the VT-100 or VT-52 terminal emulation modes to operate. The CP/M card has been modified by its developer, Scott Swenson, to operate with the new CorComp Disk Controller Card. The single-side, single-density Osborne disk format must still be used when working with the CP/M card.

I will report further on my experiences with this intriguing addition to the TI-99/4A. Future articles will include more information about CP/M, software sources, working with the unit, and reviews of software offered for the unit.

FOUR NEW PRODUCTS FROM NAVARONE

DATA BASE MANAGEMENT SYSTEM - The database system is written in Assembly Language. You can create databases with up to 35 fields in 32000 records. Record size is limited to 255 bytes. Three modules are used to accomplish all the necessary tasks to create a database. The Entry module sets up screen input and provides file maintenance functions. A Disk Sort module is used to sort files with up to six (6) nested sort keys. The Report module generates custom reports from the database.

HOMework HELPER + - This new educational program for children 8 years or older contains a spelling checker with a 20,500 word dictionary. It was designed to improve study habits and written work quality. The module contains a word processor with a standard format for book reports and class projects.

SPEED READING - This module is provided in two versions, one for children and one for adults. The program is complete in the cartridge. It will operate with a minimum system configuration, and has special features which can be used if a disk drive is available. A complete workbook is included with the package.

CONSOLE WRITER - This new cartridge gives people with a 99/4A and a printer word processor capabilities. The program features true lower case letters and a full screen text editor. Text can be saved to a disk drive if one is available.

Prices for each module and Navarone Industry's address are listed below:

MODULE NAME	PRICE	FIRM'S ADDRESS
DATA BASE MANAGEMENT	\$69.00	NAVARONE INDUSTRIES 510 LAWRENCE EXPWAY # 800
HOMework HELPER +	\$49.95	SUNNYVALE, CA 94086
SPEED READING	\$49.95	(408) 866-8579
CONSOLE WRITER	\$49.95	

JOYSTICK COMMUNICATIONS By George a. Bowman - Midwest Engineering

Communications between printers and computers usually require an RS232 style interface to provide buffering to logic signals which need to travel distances of up to 20 feet. Since a protocol is involved, ordinarily a large degree of software overhead is involved to assure reliable transfers occur at user defined baud rates. TI had erected a massive protocol for its RS232 which was comprehensive, secure but expensive. Other manufacturers chose a less complex method which made their machines more cost effective.

As an engineer I decided to investigate a less expensive method for attaching my printer to the Home Computer. In my after hours time I invented my "Missing Link" which Mr. Veith has granted time to review (THE MISSING LINK WAS REVIEWED ON PAGE FIVE [5] OF ISSUE #7, MAY, 1984). This parallel interface, as opposed to serial, has low software overhead and provides an up-to-1000 cps (recently improved) path from your computer to any "Centronics" style printer using the joystick port.

As an EMI engineer, I have used my computer to design powerful line filters for primary use in computer mainframes which have switcher supplies. Such filters bring computers into compliance with limits on emissions and the computer program I developed is included. Other programs perform wordprocessing, sprite generation and general engineering utilities.

The principles of operation of this interface are interesting. Most of you probably don't know what assembly command activates your joystick port. It's the "LDCR" or "SBO" commands. If you preset the CRU (Reg. 12) to X0024 and send three bits via "LDCR" you can make either joystick perform to your command.

An efficient algorithm was developed which commands the joystick port to send high speed serial data to a hardware circuit which converts it to parallel. A principle called "integrate and dump" has been used to assure data integrity. The principle is called a hybrid technology since it isn't purely digital or analog. Almost all digital multimeters use this idea.

Here is a routine to cause the Joystick port to turn on a light emitting diode connected to Joystick B; pins 2 to 8; and Joystick A; pins 7 and 9. Acquire the proper nine pin connector before hooking it up and try both polarities if it doesn't work exactly right the first time. With both the conventional RS232 and the Link you have a great variety of things you can do. I even include a schematic of a simple controller you can build. In time you will see the Joystick Port to be a much more valuable part of your Home Computer.

```

*****
*      JOYSTICK PROGRAM      *
*                               *
*TO ACTIVATE JOYSTICK 1 PUSH "1"*
*TO ACTIVATE JOYSTICK 2 PUSH "2"*
*TO TURN THEM OFF PUSH "0"   *
*TO LEAVE PROGRAM PUSH "A"  *
*JOYSTICKS CANNOT BE TURNED ON *
*SIMULTANEDUSLY !!!!        *
*****
EXIT  LI  R0,261
      LI  R1,DISLI1
      LI  R2,22
      BLWP @VMSW
      CLR @KEYADR
SCAN1 CLR @STATUS
      BLWP @KSCAN
      LDCR R8,3
      LI  R0,325
      LI  R1,DISLI2
    
```

```

DEF JOYCOM
REF VWTR,VSBW,VMBW,KSCAN
WR BSS >20
STATUS EQU >837C
KEYADR EQU >8374
KEYVAL EQU >8375
PATTERN DATA >FOOF,>FOOF,>FOOF,>FOOF
JOY1ON BYTE >31
JOY2ON BYTE >32
JOYOFF BYTE >30
LEAVE BYTE 65
ATEN DATA >A
BORDER DATA >FFFF,>0000,>0000,>0000
DATA >0000,>0000,>0000,>0000
DATA >0000,>0000,>0000,>0000
DATA >0000,>0000,>0000,>0000
DISLI1 TEXT '** JOYSTICK PROGRAM **'
DISLI2 TEXT '**THE KEY PUSHED WAS **'
DISLI3 TEXT '**PRESS ENTER TO FINISH**'
MSG4 TEXT '**JOYSTICK IS ON NOW!!!**'
MSG5 TEXT '**SORRY JOYSTICK IS OFF**'
RTNVEC BSS 2
JOYCOM MOV R11,@RTNVEC
LWPI WR
LI R12,>0024
LI R5,>0500
LI R6,>0600
LI R7,>0700
LI R0,>0700
BLWP @VWTR
LI R0,>039F
LI R1,>0D00
BLWP @VSBW
LI R0,>380
LI R1,>8400
CLOOP BLWP @VSBW
CI R0,>039E
INC R0
JMP CLOOP
PTERN LI R0,>0800
LI R1,PATTERN
BLWP @VMBW
LI R0,395
MOV @KEYVAL,R4
SRL R4,8
LI R3,404
LI R0,406
BL @FIGUR
LI R0,485
LI R1,DISLI3
LI R2,22
BLWP @VMBW
SCAN2 CB @KEYVAL,@JOY1ON
JNE JY2
MOV R6,R8
LI R0,517
LI R1,MS64
LI R2,22
BLWP @VMBW
JY2 CB @KEYVAL,@JOY2ON
JNE JY0
MOV R7,R8
LI R0,517
LI R1,MS64
LI R2,22
BLWP @VMBW
JY0 CB @KEYVAL,@JOYOFF
JNE NOTIN
MOV R5,R8
LI R0,517
LI R1,MS65
LI R2,22
BLWP @VMBW
NOTIN CB @KEYVAL,@LEAVE
JEQ ESCAP
B @SCAN1
FIGUR MOV R4,R5
CLR R4
AI R5,>30
SLA R5,8
MOV R5,R1
BLWP @VSBW

```

TIPS #15 FROM THE TIGERCUB

TIGERCUB SOFTWARE 156 Collingwood Ave. Columbus OH 43213 (614)235-3545

Copyright 1984 Tigercub Software. May be reprinted by non-profit publications with credit to Tigercub Software. Distributed to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters.

Tigercub Software has over 130 original programs for the TI-99/4A in Basic and Extended Basic, on cassette or disk, at only \$3.00 each. A descriptive catalog is available for \$1.00, which may be deducted from your first order.

I am presently writing Extended Basic versions of many of my programs, enhancing some programs and adding some new ones, and will soon be publishing a new catalog. If your local school has purchased TI-99/4A computers for their classrooms, why not let them know that Tigercub has educational programs at a price they can afford?

No, the Tips are not available by subscription, and I do not have back issues available. However, the entire contents of Tips #1 through Tips #14 are now available on disk, with more added - a full disk of 50 programs, routines and files for only \$15.00 postpaid.

Since no one else ever reviews any of my software, I'll have to do it myself. **WHITWATER RUN** is a raft trip down Whitewater Canyon, in 7 levels of difficulty. You must avoid running aground on the green shores, stay away from the black rocks, and do your best to stay out of the whitewater which may conceal other rocks. Every trip is different and none are impossible. The Greenhorn level is a short leisurely ride just for practice. Raftswan is a longer ride which surprises you with a sudden surge of fast water toward the end. Voyageur is a complete change of pace, as your raft leads through the rapids. The next three levels are much faster, each with an increasing number of hidden rocks in the whitewater. Long Journey takes you through all six levels. Now also available in an Extended Basic version, very fast.

Most of the speed reading programs on the market are worthless, because they flash a phrase from their data file onto the screen and then require you to retype it exactly. Even if the data file of phrases is large, they will soon be recognized from memory rather than from reading. Also, the purpose of speed reading is to quickly grasp the meaning of a text, not the exact wording. **SPEEDER READER** avoids these faults by assembling each sentence at random from files of nouns, verbs, adjectives, adverbs and modifiers. The result is an infinite number of different sentences, always grammatical but usually ridiculous. The sentence is flashed on the screen for whatever interval you select; then you are asked any one of several randomly selected questions about it. Where did the bald ballerina kiss the fat judge? If you can keep from laughing too much, your reading speed can be increased greatly. **JUNIOR SPEEDER READER** is the same except that the sentences describe the activities of various animals, in simpler words.

Several people have sent me enhancements to my Menu Loader, and I greatly appreciate them. The trouble is, if I incorporated them all the program would take up about 25 disk sectors! So, I have borrowed some ideas, added a few of my own, and here is the result. It will list and load up to 99 programs, stopping at the end of every screenfull or stopping whenever any key is pressed and then offering you the choice of loading, deleting or quitting. It will ask you to verify a deletion by name before deleting it, and will display the name of the program it is loading. It also contains a feature to warn you if you are getting a bad count of disk sectors used - which I find happening more often than you might realize.

```

100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
110 CALL CLEAR :: CALL INIT
:: CALL LOAD(8196,63,248)::
CALL LOAD(16376,67,85,82,83,
79,82,48,8)
120 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
130 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
140 CALL CLEAR :: CALL SCREE
N(5):: FOR S=1 TO 14 :: CALL
COLOR(S,7,16):: NEXT S :: C
ALL VCHAR(1,31,1,96):: CALL
COLOR(0,2,16)
150 OPTION BASE 1 :: DIM PG$(
99),T$(5)
160 T$(1)="dis/fix" :: T$(2)
="dis/var" :: T$(3)="int/fix
" :: T$(4)="int/var" :: T$(5
)="program"
170 IMAGE ##
180 DISPLAY AT(1,9):"DISKETT
E MENU"
190 ! IF YOU HAVE MORE THAN
ONE DISK DRIVE, DELETE THE !
IN LINE 200 AND THE FIRST S
TATEMENT IN 210
200 ! DISPLAY AT(12,6):"DISK
?(1-3):" :: ACCEPT AT(12,19
)SIZE(-1)VALIDATE("123"):D$
:: D$="DSK"2D$2".
210 D$="DSK1." :: OPEN #1:D$
,INPUT,RELATIVE,INTERNAL ::
INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2):SEG$(D$,1,4)&" - D
iskname= "&N$;
220 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K:" Prag Fi
lename Size Type:"-----
::
I,VT=0 :: TT=J-K
230 FOR X=1 TO 99 :: IF X/20
<>INT(X/20)THEN 260
240 DISPLAY AT(24,1):"Type c
hoice or 99 for more" :: ACC
EPT AT(24,27)VALIDATE(DIGIT)
:K :: IF K=99 THEN 250 :: IF
K>0 AND K<NN+1 THEN 420 ELS
E 240
250 X=1
260 I=I+1 :: IF I>127 THEN K
=X :: GOTO 360
270 INPUT #1:P$,A,J,B :: NN=
NN+1
280 IF LEN(P$)=0 THEN 320
290 DISPLAY AT(X+4,2):USING
170:NN :: DISPLAY AT(X+4,6):
P$ :: PG$(NN)=P$ :: DISPLAY
AT(X+4,18):USING 170:J :: DI
SPLAY AT(X+4,22):T$(ABS(A)):
: VT=VT+J
300 CALL KEY(0,KK,ST):: IF S
T=0 THEN 310 :: FLAG=1 :: GO
TO 320
310 NEXT X
320 DISPLAY AT(X+4,1):" " ::
DISPLAY AT(X+4,2):USING 170
:NN :: DISPLAY AT(X+4,6):"Te
rminate" :: DISPLAY AT(X+5,2
):STR$(NN+1)&" Delete"
330 IF VT=TT OR FLAG=1 THEN
350 :: DISPLAY AT(2,25)SIZE(
4):VT
340 FOR @=1 TO 10 :: DISPLAY
AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
350 DISPLAY AT(X+6,1):" C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(2)VALIDATE(DIGIT):K ::
IF K<>NN AND K<>NN+1 THEN 41
0
360 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
370 DISPLAY AT(X+5,11)SIZE(1
8):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 370
380 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ":PG
$(KD):"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN
"):Q$ :: IFQ$<"Y" THEN 400
390 DELETE D$&PG$(KD)
400 CLOSE #1 :: CALL VCHAR(1
,3,32,672):: NM=0 :: X=0 ::
GOTO 180
410 IF K<1 OR K>99 OR LEN(PG
$(K))=0 THEN 320
420 CLOSE #1
430 CALL INIT :: CALL PEEK(-
31952,A,B):: CALL PEEK(A*256
+B-65534,A,B):: C=A*256+B-65
534 :: A$=D$PG$(K):: CALL L
OAD(C,LEN(A$))
440 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1)
)):NEXT I :: CALL LOAD(C+I,
0)
450 CALL VCHAR(1,3,32,672)::
CALL SCREEN(8):: FOR S=0 TO
14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ":A$
460 RUN "DSKX.123:4567890"

```

If you don't like my Tigercub cursor, just delete lines 110 (after the CALL CLEAR), 120 and 130. That routine for redefining the cursor has appeared recently in various newsletters without attribution, and I'd like to know who to credit for it. The secret of it is in line 120, where the numbers after 12288 are the decimal equivalents of the hexadecimal numbers (which are the hex equivalent of the binary numbers represented by the off/on pixels) used to redefine a character.

You may have noticed that all programs published in the Tiger Cub's Tips are in 28-column format, just the way they will appear on the screen. And they are printed directly from LISTed actual programs, so that they cannot contain typographical errors - don't you wish the computer magazines did that!? The problem is that when a program listing is merged into the TI-Writer buffer and printed in the formatter mode, the @, &, \$ and the exponent sign are treated as control characters, and strange things happen!

The following program will convert a program, which has been listed to disk with LIST "DSK1.FILENAME", into a file in 28-column format which can be loaded into TI-Writer, and will optionally substitute the left and right braces, ASCII 124 and the tilde for the @, &, \$ and the exponent sign, and transliterate them so that they will print correctly in the formatter mode. However, for that very reason this program will not print correctly! When you come to line 280, type DATA shift 2, fctn F, shift 7, fctn G, shift 6, fctn W, shift 8, fctn A.

```

100 DISPLAY AT(2,4)ERASE ALL
: "28-COLUMN CONVERTER" :: DI
SPLAY AT(5,12): "by Jim Peter
son"
110 DISPLAY AT(7,1): "To con
vert a program, saved": "with
LIST "DSK1.FILENAME", ": "i
nto 28-column format which":
"can be merged into the text
"
120 DISPLAY AT(11,1): "buffer"
of TI-Writer."
130 DISPLAY AT(13,1): "Optio
nally with transliter-": "ate
d @, &, $ and ^ for cor-": "r
ect printing from formatter"
: "mode."
140 DISPLAY AT(18,1): "Do yo
u want to print the": "file f
rom the": " (E)ditor?": " (F)o
rmatter?"
150 ACCEPT AT(23,1)VALIDATE(
"EF")BEEP: Q$
160 DIM A$(1000):: CALL CLEA
R :: INPUT "What is the FILE
NAME? DSK1.": FN$ :: FN
$="DSK1.F&FN$ :: PRINT :
170 INPUT "What is the new F
ILENAME? DSK1.": PN$ :: PN$
="DSK1."&PN$ :: OPEN #1: FN$,
DISPLAY, VARIABLE 80, INPUT :
: OPEN #2: FN$, DISPLAY, VARIA
BLE 80, OUTPUT
180 IF Q$="E" THEN 190 :: PR
INT #2: ".TL 126:94;" :: PRIN
T #2: ".TL 123:54;" :: PRINT
#2: ".TL 125:38;" :: PRINT #2
: ".TL 124:42;"
190 FOR L=1 TO 1000 :: LINPU
T #1: A$(L):: IF LEN(A$(L/1))
=80 OR LEN(A$(L-1))=160 THEN
A$(L-1)=A$(L-1)&A$(L):: L=L
-1
200 IF EOF(1) THEN L=L+1 :: G
OTO 220
210 NEXT L
220 FOR J=1 TO L-1 :: S=1
230 FOR T=1 TO 10 :: B$(T)=S
THEN 240 :: GOSUB 280
240 S=S+28 :: NEXT T
250 FOR N=1 TO 10 :: IF B$(N
)<>" THEN PRINT #2: B$(N)
260 NEXT N
270 NEXT J :: CLOSE #2 :: CL
CSE #1 :: END
280 DATA @, @, &, &, ^, ^, $, $
290 RESTORE 280
300 FOR W=1 TO 4 :: READ CH$
: R$
310 X+POS(B$(T), CH$, 1):: IF
X=0 THEN 330
320 B$(T)=SEG$(B$(T), 1, X-1)&
R$&SEG$(B$(T), X+1, LEN(B$(T)
)): GOTO 310
330 NEXT W :: RETURN

```

Now, if the programs above are what I give away in my Monthly Tips Column, is it not worth a dollar for my Catalog to find out what I'm selling?

Happy hackin'

Jim P.

THE TI PLAYGROUND AND TI IN WONDERLAND - PRICE: \$9.95 EACH

BOOK REVIEW By Darrell Ingold, Staff Analyst

Author - Fred D'Ignazio

Hayden Books, 10 Mulholland Dr., Hasbrouck Heights, New Jersey 07064

Hayden books were reviewed together because they were both written by Fred D'Ignazio and are complimentary to each other. TI IN WONDERLAND contains 21 programs while The TI PLAYGROUND contains 23. The format for both books is identical; and that's good because it is well done.

Before we get into the books themselves, I thought it was particularly interesting that the preface pointed out several programs in the book were actually written by high school students from Patrick Henry High School in Roanoke, Virginia. This fact itself should appeal to many of the kids who might be using the programs in this book as well as serve as an inspiration to other high school students that are learning computers. This idea, according to Fred, was suggested by his publisher, Gary Markman.

Neither of these books require any of the more expensive peripherals; just the basic computer console, TV and tape recorder. In addition all of the programs are rather short (extremely short to an experienced programmer...but then they were not designed for him) and do not take long to type into the computer.

Getting back to the layout of the books, the Table of Contents divides the programs into Faces (right/wrong answer subroutines), Alphabet, Words, Numbers, Colors, Music, Drawing, Knowledge, Hand-Eye and Imagination. The Faces, happy and sad, are in both books and are used as subroutines for the other learning games to indicate right and wrong answers. While the categories in both books are the same, it appears that the programs in TI IN WONDERLAND are somewhat more advanced in skill levels than The TI PLAYGROUND; therefore I would suggest that one read The TI PLAYGROUND first.

Each chapter is set up in a most useful fashion. There is the introductory paragraph for parents and teachers followed by a section for the kids. The kids' section gives them the story for the game while the parents and teachers get a list of the objectives for the kids. It's always nice to know where you are headed!

The program itself follows, printed in clear, easily legible type. There is a "Highlights" section immediately after the program that points out the various programming highlights of the program thereby making it a good learning tool for the programmer (parent).

Next there is a list of variables and exactly what they are representing...for example: W - Whistle counter (6 whistles). It really serves to clarify them in the minds of the less experienced programmer. At the conclusion of each chapter there are several suggestions for do-it-yourself changes and enhancements for the programmer to experiment with, including changes in subject matter, which would really broaden the total number of programs that one could derive from these books.

I would heartily recommend both of these books to the parents who want to give their children a headstart in school (and learn some programming skills at the same time themselves) and also for the teacher who needs specialized instruction in certain areas that the youngsters can do by themselves, such as in a learning center type of application.

INTRODUCTION TO TI BASIC

BOOK REVIEW By Don Veith - Editor

Authors - By Don Inman, Ramon Zamora, And Bob Albrecht
Hayden Books, 10 Mulholland Dr., Hasbrouck Heights, New Jersey 07064

The authors of this book are in a unique position to write about the TI-99/4A Home Computer. The Owner's Manual packaged by TI with the computer was also written by these gentlemen. The introduction states, "This book was designed especially for those people with beginning to intermediate experience with the small home computer."

The book does overlap chapters in the TI manual. The material compliments rather than distracts from information contained in your Users Manual. An example of information presented that is helpful is the construction of a timing delay loop for instructions in games. Normally, a delay loop of the same duration would be created as a subroutine using the GOSUB command. The subroutine program code is shown below:

```
1200 DELAY SUBROUTINE          1220 NEXT DELAY
1210 FOR DE=1 TO 2500         1230 RETURN
```

Now, the code is excellent and will save you coding several Delay Subroutines in your program each time something must be held on the screen for reading. The only problem is that it does not take this much time to read a single line of instructions. Yet, you could not read a full screen of instructions in this same time period.

A unique and very obvious solution was presented in the later chapters of the book. The changes in the Delay Subroutine are shown below:

```
1200 REM DELAY SUBROUTINE      1240 CALL CLEAR
1210 FOR DE=1 TO N             1250 RETURN
1230 NEXT DE
```

Just prior to branching to the Gosub each time, you simply state a value for N that is sufficient in length for the amount of screen text that must be read. This allows the timing variance for main program screen and varying times wherever a delay is needed in the program. This technique of using the delay loop was totally unfamiliar to several programmers I spoke to after reading this book.

The book contains many excellent reference chapters for its intended category of reader. Many programmers as they advance to higher languages do not bother working in BASIC. I prefer to work in it occasionally just to stay familiar with its concepts. Many new users are in need of help to get their programming efforts started. This book is an excellent addition to the Users Manual providing sound programming techniques. Chapter I has an excellent title called "Gateway To Adventure". Programming can be a very rewarding adventure with the satisfaction derived from creating something unique tailored to your specific requirements. This book can aid the beginning and intermediate BASIC programmer in obtaining an excellent start towards this goal. Some experienced programmers can always learn new twists from this excellent publication although we would never admit to the fact that the information presented was previously unknown to us!

CORCOMP DISK CONTROLLER CARD

Version 2.2 of the double-sided, double-density CorComp Controller Card has been released by the firm. The unit is now compatible with all the items that previously required fixes to operate with it. The three problem areas were (1) 99/4's manufactured in 1981 and early 1982, (2) thermal printers, and (3) Foundation's 128 K Memory Card.

Some notes to be passed on from CorComp to Disk Manager owners are:

1. To utilize the 9900 Disk Manager, you need at least 32K Memory in the PES.
2. If the additional memory is not available, the Disk Manager will work with TI's Disk Manager II and fully exercise the unit's full capabilities.
3. Always keep the Alpha Lock key depressed when working with the unit installed. We are advised to enter all file names in capital letters.
4. The Forth Loader for double-sided, double-density operation is available from CorComp.

The 99'ERS UGA writer Jim Vincent provided this software to CorComp in an effort to aid TI Forth users who purchased the 9900 Disk Manager. If you are interested in updating your CorComp 9900 Disk Manager to the 2.2 version, we suggest a telephone call to the firm. CorComp recommends Lorelei Alkire for products in need of attention, Don Scofield for technical problems, and Jackie Sagouspa concerning product marketing or any aspect of company policy. CorComp's telephone number is (714) 630-2903. Tell them you read the information in THE NATIONAL NINETY NINER.

TI FORTH By Jim Vincent

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

BASE->R DECIMAL 16 SYSTEM 0 0 GOTOOX

." FORTH Screens to/from V80 files " CR

." by JWVincent " CR

." These screens will read or write TI

." variable 80 files to or from TI-FORTH

." screens. If DISK_HI equals DISK_SIZE

." one drive will be used. When using one

." drive, begin with FORTH loaded, you

." will be prompted when to load each

." disk. If multiple drives are used place

." FORTH in # 1 and the V80 files disk in

." # 2. V80 files read/written must/will

." be named SCRNXxx where xxx is the

." screen number. When reading a V80 file

." EOF will cause a disk error, after

." -->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

." which the FORTH disk should be loaded

." and the FLUSH command executed.

." The word format is: " CR

." n1 n2 WRITE-V80 (n1=start scrn)

." n1 READ-V80 (n2= end scrn)

68 CLOAD STAT 84 CLOAD message

0 VARIABLE DF 0 VARIABLE PF 0 VARIABLE BUFR 78 ALLOT

PABS @ 10 + BUFR 6144 FILE V80

V80 SET-PAB VRBL 80 REC-LEN F-D" DSKn.SCRNXxx"

: DRIVE-NO DISK HI @ DISK SIZE @ = IF 0 ELSE 1 THEN DF ! ;

: V80-DSK DF @ IF ELSE CR ." LOAD V80 DISK" CR KEY DROP THEN ;

: FTH-DSK DF @ IF ELSE ." LOAD FORTH DISK " CR KEY DROP THEN ;

: FIX-NAME DRIVE-NO PABS @ DF @ 49 + OVER 23 + VSBW

OVER 100 /MOD 48 + ROT 29 + VSBW 10 /MOD 48 +

PABS @ 30 + VSBW 48 + PABS @ 31 + VSBW V80-DSK ;

-->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

: R-LEN DUP 63 + 64 0 DO I OVER OVER - C@ 32 = IF I 63 < IF

DROP THEN ELSE LEAVE THEN LOOP SWAP DROP 64 SWAP - ;

: WRITE-16 16 0 DO DUP BUFR 64 CMOVE R-LEN WRT 64 + LOOP DROP ;

: WRT-V80 CASE 1 OF 1 ENDOF 2 OF SWAP 2 ENDOF

3 OF SWAP ROT 3 ENDOF 4 OF SWAP ROT >R ROT R

SWAP 4 ENDOF ENDCASE

0 DO WRITE-16 LOOP ;

: SCR-RD OVER OVER 4 + > IF DUP 4 + OVER 1 ELSE 0 THEN PF !

0 ROT ROT DO I BLOCK SWAP 1 + LOOP ;

: WRITE-V80 !+ SWAP DUP >R SCR-RD > FIX-NAME DROP OPN

BEGIN WRT-V80 PF @

WHILE FTH-DSK 4 + SCR-RD V80-DSK

REPEAT CLSE ;

-->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

: READ-4 8210 OVER 4 + ROT

DO DUP-1024 32 FILL

16 0 DO RD DUP 64 > IF DROP 64 THEN

I IF ELSE OVER J 32768 OR SWAP 2 - ! THEN

OVER BUFR SWAP ROT CMOVE 64 + LOOP

4 + LOOP DROP FTH-DSK FLUSH ;

: READ-V80 FIX-NAME OPN BEGIN DUP READ-4 4 + V80-DSK AGAIN ;

EMPTY-BUFFERS R->BASE (Can you use =0 -TRAILING FIRST UPDATE and +BUF to shorten this program? Try it)

-->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

: READ-4 8210 OVER 4 + ROT

DO DUP-1024 32 FILL

16 0 DO RD DUP 64 > IF DROP 64 THEN

I IF ELSE OVER J 32768 OR SWAP 2 - ! THEN

OVER BUFR SWAP ROT CMOVE 64 + LOOP

4 + LOOP DROP FTH-DSK FLUSH ;

: READ-V80 FIX-NAME OPN BEGIN DUP READ-4 4 + V80-DSK AGAIN ;

EMPTY-BUFFERS R->BASE (Can you use =0 -TRAILING FIRST UPDATE and +BUF to shorten this program? Try it)

-->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

: READ-4 8210 OVER 4 + ROT

DO DUP-1024 32 FILL

16 0 DO RD DUP 64 > IF DROP 64 THEN

I IF ELSE OVER J 32768 OR SWAP 2 - ! THEN

OVER BUFR SWAP ROT CMOVE 64 + LOOP

4 + LOOP DROP FTH-DSK FLUSH ;

: READ-V80 FIX-NAME OPN BEGIN DUP READ-4 4 + V80-DSK AGAIN ;

EMPTY-BUFFERS R->BASE (Can you use =0 -TRAILING FIRST UPDATE and +BUF to shorten this program? Try it)

-->

(FORTH screens to/from variable 80 files - JWVincent - 8/15/84)

: READ-4 8210 OVER 4 + ROT

DO DUP-1024 32 FILL

16 0 DO RD DUP 64 > IF DROP 64 THEN

I IF ELSE OVER J 32768 OR SWAP 2 - ! THEN

OVER BUFR SWAP ROT CMOVE 64 + LOOP

4 + LOOP DROP FTH-DSK FLUSH ;

: READ-V80 FIX-NAME OPN BEGIN DUP READ-4 4 + V80-DSK AGAIN ;

EMPTY-BUFFERS R->BASE (Can you use =0 -TRAILING FIRST UPDATE and +BUF to shorten this program? Try it)

-->

TI-FORTH's direct sector I/O is a very fast, and efficient method for handling screens within the FORTH environment. However, when you want to trade a couple screens with your friend (along with some other files), or send them via modem, it can be a real problem. These screens solve that problem by formatting screens to TI's standard variable 80 format. They also support reading screens from V80 files.

The screens contain instructions for their use, so, I will use this column to comment on the words defined and other items. First the screens show the instructions so that you can read while they load. Next, they insure that the -FILES and memory resident messages are loaded. I had to use these messages because the disk based ones make the screen buffers flush when READ-V80 encounters EOF. That's disastrous if your using a single drive since your V80 disk is still loaded! Anyway, I found the system very apt to hang with these messages loaded and I don't recommend using them if you have a choice. I also had to clear the buffers myself since the FORTH word CLEAR dumps each time it executes (I wanted to move more than 1 scrn at a time) Due to space and time constraints, the comments will be brief. You must study this to understand how it works. Hint: read FORTH programs from the bottom up.

(Common Words)

(set up the PAB with a dummy filename)

(check for multiple drives and set Disk Flag)

(if single drive display message and wait for key press)

(ditto)

(get PAB addr and use Disk Flag to set disk number)

(divide screen number by 100 then 10 and use the)

(results to fill in the disk name)

(Write Words)

(count trailing blanks in record and adjust record)

(length, minimum record length is one byte)

(write the 16 records from the current scrn to-V80 file)

(reverse the stack order of the screen buffers read)

(and put number of buffers back on top of the stack)

(then loop thru writing each screen that was read)

(if there are more than 4 scrns left to read set PF)

(read up to 4 and leave their buffer addrs and count)

(fix input, read first screens, set file name and open)

(write the screens you read to V80 and check Pass Flag)

(if there's more to read, prompt disk swap and read if)

(if not close the file... your done.)

(Read Words)

(put first scrn bufr addr, scrn num +4, scrn num onstack)

(for each screen bufr up to 4 clear the buffer then)

(read 16 records from the V80 file checking the len)

(after the first read, flag the screen as updated)

(move the actual number of bytes read to the screen)

(loop back for the rest, prompt disk swap and flush)

(set file name, open it read 4 scrns/pass til EOF crash)

(Can you use =0 -TRAILING FIRST UPDATE and +BUF to shorten this program? Try it)

PASCAL NOTES

By EDGAR DOHMANN - JSC USER'S GROUP (JUG)

This month we will take a look at how to set up your TI-99/4A system to use the p-System. The first (and perhaps most difficult) thing to do is obtain a P-Code card to plug into your Peripheral Expansion Box. This card contains a console monitor which replaces the standard /4A monitor when the card is enabled. The card also contains the p-System interpreter which allows you to run programs that have been compiled to p-System pseudocode.

The P-Code card plugs into any slot of the PEB just like any other peripheral controller. A small slide switch is mounted on an extension tab of this card to allow you to select the p-system monitor or the standard console monitor when the computer is powered up. This certainly makes enabling and disabling the card more convenient than removing the card. However, the switch is not particularly rugged and if your PEB is up close to a wall like mine, reaching around back to find and operate the switch is still a bother.

I went to the local Radio Shack and bought a miniature toggle switch, a small plastic hobby box, and a little wire. I drilled two holes in the hobby box, one to mount the switch and the other for wire access. I soldered the wires to my new toggle switch and tack soldered the other end to the terminals of the slide switch on the P-Code card. The wires are about 2 feet long so the little hobby box sits next to my monitor on top of the PEB. Now it is a simple matter to flip the toggle switch one way or the other to turn the P-Code card on or off.

If you decide to do something similar, be sure to use a small wattage soldering iron (15 to 25 watts maximum). If you aren't a qualified technician with proper tools, check around at your local User's Group. Surely most every group will have one or more members capable of doing something like this for you. If you switch your system between the p-System and "normal" use as often as I do, you will enjoy something like this.

If you can get compiled p-code programs on cassettes, you do not need a disk drive to use the p-System. However, about all you will be able to do in this case is run programs others have written. If you want to write and compile your own programs, you will need one or more disk drives. If you can afford it, I highly recommend the CorComp DS/DD controller and two of the half-height DS/DD drives that will fit in the PEB. While you can develop programs with a single SS/SD drive, it is tedious and frustrating and only the heartiest of souls will be able to tolerate such a configuration.

There are 4 disks (SS/SD format) available for p-System support:

PHD 5063 -- Compiler
PHD 5064 -- Assembler/Linker
PHD 5065 -- Editor/Filer (disk 1) and Utilities (disk2)

Each of these 3 items are sold separately. The Assembler/Linker is only necessary if you want to include 9900 Assembly Language routines in your programs. The Editor/Filer and Utilities are essential if you are going to do anything other than merely run programs you get from someone else on disk or cassette. The Compiler disk is necessary if you want to write your own Pascal programs. One nice thing about DS/DD disks is that all 4 of these packages will fit on one DS/DD disk with 210 blocks (420 sectors) still available for other purposes.

An SS/SD disk has 360 sectors while a DS/DD disk has 1440 sectors. TI uses 256 bytes per sector. The p-System treats file storage in terms of 512-byte "blocks". Thus it takes 2 sectors on a TI disk to store a single p-System block. Disks for the p-System can be formatted with the TI Disk Manager Cartridge, the CorComp Disk Manager program, or the DFORMAT utility program on the Utilities Disk.

If you use TI's Disk Manager I cartridge you are limited to SS/SD formats. Disk Manager II will support SS/SD and DS/SD. The CorComp Disk Manager will support SS/SD, DS/SD, SS/DD, AND DS/DD formats. Even though the DFORMAT utility prompts for double density formats, it apparently is not compatible with the CorComp disk controller card and can only be used like the Disk Manager II. If anyone figures out a patch for DFORMAT so it can provide double density formats with the CorComp controller, be sure to let me know.

Regardless of which program you use to format a disk, you will need to use the Zero command of the Filer program to initialize the directory on your disk before you can write to it. When you format a disk for "normal" /4A use, the Disk Manager programs automatically initialize the directory as part of the formatting process. This directory layout is not compatible with the p-System's use of the disk so it must be "Zeroed". Actually, I think this is a misleading term because it actually initializes the directory (which does include zeroing the allocation bit map so any files on the disk are "lost") by writing certain header information in the directory so the p-System can use the disk. This is a necessary step even if you use DFORMAT to format your disks because unlike the Disk Manager programs, DFORMAT does not initialize the directory as part of the formatting process.

Before you start writing your own programs it might be a good idea to obtain a few programs that already work. You can then examine the text files, see how they were written, and perhaps get a few ideas on how to set up a program which may minimize your debugging efforts later. One possible source of programs is the USUS Pascal Users Group. I understand that they have a number of programs available that will run on the /4A. I have written to them to inquire about membership in their organization and will report on them after I receive a response.

Another good source for programs is some of the bulletin board systems run by TI-99/4A owners. I downloaded a REMTALK program (which allows two computers with p-Systems to transfer files to each other via modem) from a bulletin board in Washington, D.C. (301-434-0117). If you know of other bulletin boards that post Pascal programs for downloading, let me know so I can pass it along to others through this article.

I also want to put together a National Pascal Library for the /4A so we can all share in what is available. If you want to contribute, write me at the UGA address on the front of this newsletter. I also would like some input as to the direction you would like to see for future articles.

DISK INSERT ERRORS By George Lambert

Ever tried to read a disk file, and had your program "bomb" out with an error because you forgot to insert the disk or close the drive door? Extended Basic has a statement which will trap the error and send you to a subroutine where you can print a message to give yourself another chance - without terminating the program. The example below will demonstrate.

```
100 ON ERROR 500
110 OPEN #1:"DSK1.FILE"
115 ON ERROR STOP
120 INPUT #1:A$
130 PRINT A$

140 STOP
500 REM ERROR ROUTINE
510 ON ERROR 520 :: CLOSE #1
520 INPUT "INSERT DISK DUMMY":B$
530 RETURN 100
```

Line 100 sets it up to gosub 500 on an error. Line 115 restores the error routine to normal. Line 510 is necessary even though it looks a little strange. An error at 110 leaves FILE #1 neither open nor closed. It can't be closed without an error, nor can it be opened again without one. Line 510 makes an attempt to close and it will fail, but it clears up the confused state of the file status anyway.

One other curious note. If line 110 opens "DSK1", an error will NOT occur on the OPEN statement but on the first input at line 120. In this case, the "ON ERROR STOP" at line 115 must be moved to line 125. NOTE: Article copied from the Birmingham, AL TI UG newsletter BUG NEWS of April/May 1984.

PRODUCT REVIEW BY JOHN PHILLIPS

I thought that this month's article would be a bit different. Although I love to bring assembly language code to you, I have a number of things sitting on my desk that I must attend to. Believe me, it is a pleasure to attend to these items as they all relate to my forte, TRS9900 Assembly Language. With this in mind, I would like to review a wonderful product sent to me by very creative and dedicated /4A owners. I will continue my assembly language articles in the following newsletter, so hang in there!

Incidentally, those of you that have run across a program called **BEYOND PARSEC** on disk for the /4A do me a favor: When the title screen appears, press the SHIFT-8 (*) key. I think you'll be pleasantly surprised at the author of the program. For those of you who have **HOPPER**, press the SHIFT-4 (\$) during the title screen. There is a cheat mode in **HOPPER**. Press the SHIFT-8 (*) on the title screen. You may proceed to any level you wish. I hope you enjoy these tips.

T I R U N N E R

PRODUCT: TI RUNNER
COMPANY: EB SOFTWARE
AUTHORS: JON BURT AND SCOTT EMERY
ADDRESS: 12912 VILLA ROSE DR.
SANTA ANA, CA 92705
DESCRIP: ENTERTAINMENT SOFTWARE
PRICE : \$24.95

When I was at Texas Instruments, Lubbock, I got a first-hand glance at all the "quality" software that was contracted. Some were good, while others weren't worth the time and trouble I put into fixing them. Most everyone knew that these software products were "junk", but none of the management seemed to listen to us.

Those days are past, but I still see a lot of people trying to push a product that really isn't worth a hill of beans. However, a wonderful game jumped into my lap that reinforces the fact, not fiction, that the TI was the greatest game machine ever. This wonderful product is called **TI RUNNER** and was designed by two creative college students somewhere in distant California. I'll say this about **TI RUNNER**: **Buy it!**

Had a product like this come along in Lubbock, I would have sabotaged the other high-priority items to let this fly through. Of all the games I have created or played, **TI RUNNER** is by far the most intriguing, strategic, frustrating, enjoyable game I have ever come across. I must tip my hat to these two whiz-kids. They have outdone themselves!

TI RUNNER is a take-off on a popular program called **LODE RUNNER**, which was released for the Apple, Commodore, IBM, and Atari home computers. TI was looking at this program for possible conversion to the /4A, but never quite made the decision. I remember when we were evaluating the program . . . we just couldn't keep people away from it! It was **ADDICTING!** Unfortunately, **TI RUNNER** has the same effect on we die-hard game junkies.

I brought a friend of mine (another former Lubbock'er) who was one of those addicted to **LODE RUNNER** over for a social gathering last week. Needless to say, I barely got 50 words in all night as he was glued to the screen playing **TI RUNNER**. This program is not a conversation piece, as it becomes difficult to think and talk at the same time. Let me explain.

TI RUNNER has the player controlling a moveable white sprite, shaped like a man, around a maze of ladders and brick walls. Your man has the capability to climb the ladders and bombard the brick walls underneath him. There are bricks, however, which cannot be destroyed. These bricks are distinguishable from the other volatile bricks, though.

There are three other men moving through the maze attempting to catch you. You can escape these enemies by bombing the bricks in front of you or behind you. If the chasers come after you, they will fall in the holes left by the abandoned bricks. You may then run over the top of them (yes, folks, a human bridge!) and proceed as usual. Oh, I forgot to mention, the men escape from their holes within a few seconds and head right for you again.

Located on the maze (or, sometimes, in the maze) are various treasures which you must collect. The chasers can pick up these treasures, as well, but you can retrieve them if you get the chaser to fall in a hole. His treasure will appear above him for your taking. You must get the treasures . . . all of them! Once you do, your escape ladder will appear. You must climb the ladder in order to reach the next level.

Speaking of levels, there are 50! Each maze is stored on diskette as a program file 768 bytes long. They load rather quickly, but the constant loading does get annoying after a while. A nice VDP buffer for storing the mazes while in play would have helped considerably on the disk access time.

The mazes become progressively harder and harder. I was impressed by the careful consideration these two programmers put in on the sequence of events that lead to your escape. Each maze has a solution, but some of these solutions are outlandish. My **TI RUNNER** expert only got to the 11th level after 1 week of play. He'll probably get to level 25 sometime next century.

As with all games, there is a scoring system and you gain extra men as you progress through the levels. This is an absolutely professional product with a full demo mode, great sound effects, and nice special effects. The game is a hum-dinger when played on a 99/8 at full speed! The graphics are good, but not exceptional. However, this does not deter the overall product. You barely have time to notice the graphics anyway!

In my opinion, the product gets 5 stars out of 5 (and then some!). Now, I must get off **TI-WRITER** so I can play a few games of **TI-RUNNER**. If anyone knows of a **TI-RUNNER** rehabilitation center, please give me a call. It's a winner!

RATING: ***** (out of 5)

... John Phillips

TOURE DE BOARDS

By Ron Albright, M.D. and Don Veith

PALM BEACH, FLORIDA TIBBS (tm)

BULLETIN BOARD DOWNLOAD TUTORIAL

IF YOU ARE USING TI-WRITER, A TI COMPUTER, AND ARE SEEING THIS MESSAGE - YOU NEED TO PRESS CONTROL 5 AND GO INTO THE 80 COLUMN MODE! PRESS CONTROL 5 NOW!

This tutorial was originally written to be placed on the Source. Here are some answers to your questions about downloading text from BBS's to your TI-99/4A. Also, I'll pass along a short Extended Basic utility program called CHANGE, which can be used with the TI-WRITER word processor and Terminal Emulator II (we call it TEII for short) to upload messages written with TI-WRITER to a TIBBS (tm) System, as well as Source's POST.

First, TEII. It allows you to copy a "page" of data to another port or device using the Control-2 command. Remember that a page can either be from 34 to 40 character wide, or 80 characters wide, although you can see only a 40-character "window" in that mode. If you are copying to a disk drive, you have to leave TEII with the Control-Zero command, or your file is lost. (More experienced TI users will know there are a couple of ways to retrieve this data, but that's for another story.) TEII files are saved on disk in Display/Fixed 80 or Display Variable 80 format. These files can be directly loaded into TI-WRITER (on which this is being written) and edited from there. TI-WRITER saves files in Display/Variable 80 format. And, TEII's "auto-logon" function accepts files in Display/Variable 80 format. Put these together, and you have the combination that will allow you to upload a text file using the two programs. Page 28 of the TEII manual has an odd explanation about hex bytes and so forth. Forget that. Here's what you need in a TEII logon file: --In front of each line of data, you need to stick the character "1". --Then, if you want to delay sending the next line, put in a line that is made of the character "2", and another character that tells TEII how long the delay should be. (This next part gets a little complicated.) The module figures the delay by dividing by 60 the ASCII number of the character that follows the "2". For example, a small "x" has an ASCII number of 120, so TEII will wait for two seconds before it feeds the next line of text. Now, here's the listing of the CHANGE program, written in Extended Basic, to be used with TI-WRITER:

```
100 CALL CLEAR
120 INPUT "Name of file to be converted: ":F$
140 OPEN #2:"DSK1."&F$&"C"
160 OPEN #1:"DSK1."&F$,INPUT,DISPLAY,VARIABLE 80
180 PRINT "Converting line number ";
200 X=1
220 LINPUT #1:A$
240 IF EOF(1)=1 THEN 380
260 A$="1"&A$&CHR$(13)
280 PRINT X;
300 PRINT #2:A$
320 PRINT #2:"2"&CHR$(120)
340 X=X+1
360 GOTO 220
380 CLOSE #1
400 CLOSE #2
420 PRINT "Conversion complete"
440 END
```

Here's the explanation of the CHANGE listing: LINE 120 prompts the user to type in the name of the file (minus the "DSK1." the computer needs.) The space is needed due to TI's 28-column format:

LINE 140 opens the duplicate file, with the first filename, plus a "C".

LINE 160 opens the original text file.

LINE 180 lets you know the file was found.

LINE 200 sets up a variable for the line count to be shown on the screen.

LINE 220 gets each line of the text file. "LINPUT" exists only in Extended Basic. Standard "INPUT" ends the line when a comma is found.

LINE 240 tells the program when to end.

LINE 260 changes the text line so it can be read by TEII, adding the "1" at the beginning. At the end, "CHR\$(13)" is a carriage return character, needed so the remote system will know the end of the line has been reached.

LINE 280 prints the number of the line being processed on the screen.

LINE 300 stores the converted text line in the duplicate disk file.

LINE 320 places the characters "2x" in the duplicate file. "2" tells TEII to wait, and "x" tells it to wait for two seconds.

LINE 340 increments the line count, and LINE 360 sends the program to the line to check if the end of the original file has been reached.

LINE 380 and LINE 400 close the disk files.

LINE 420 tells the user the duplicate file has been finished.

Here's the way it all goes together: If you see a file from another system you wish to copy, use the Control-2 (OUTPUT) command. Remember to use Control-0 (EXIT) command to close the file properly.

Load the file into TI-WRITER and edit as you wish. IMPORTANT: Set the right margin to 76. If you have a line that's longer than 77 characters, the CHANGE program will make the line too long for TEII to use. Also, keep your file name short, no more than four characters.

When you finish editing, run CHANGE and make the duplicate file. If the original file is named TEST, the duplicate file will be named TESTC. When you're ready to feed the file down to another system, go to the auto-login screen of TEII, and type in the duplicate file name. The file will scroll out, line by line, with two seconds between lines. NOTE: if you use the "enter" key in TI-WRITER to end a line, add this line to CHANGE:

```
250 IF SEG$(A$,LEN(A$),1)=CHR$(13) THEN A$=SEG$(A$,1,LEN(A$)-1)
```

This will remove the c/r off the end of each line. Some systems recognize two consecutive c/r's as a sign you've finished uploading your file.

PHILADELPHIA, PA. TIBBS

** TIPS FOR THE TI-99/4A **

Most of you are probably already aware of most of the information offered here, but for those who are not, here are some tips that will help you with programming your TI.

1...If your one of those people, who, when writing your own programs occasionally enter FCTN +(QUIT) instead of shift + and then ZAP everything you were just working on, try this. In the command mode, type this: CALL INIT :: CALL LOAD(-31305,16) <press enter> This will disable the FCTN QUIT, and you can press it all you want, but it will not quit.

2...Did you ever save a program in Extended Basic using the protect feature, then, at a later date, wanted to list it, make changes, or just make a copy of that program? What you will get when you try to do any of those things is * PROTECTION VIOLATION * Try this. First load the program into your computer the normal way, OLD CS1 or OLD DSK#.filename. Once the program is loaded, remain in the command mode and enter the following... CALL INIT :: CALL LOAD(-31931,0) <press enter> Now you can make copies or list it to make any changes you want. If you resave it, and want to protect it again, you must save it using the protect feature.

3...This one is for programmers who would like the computer to return to the MASTER TITLE SCREEN when the program ends. Instead of the usual END statement that would end your program, use this... CALL INIT :: CALL LOAD(-31962,32) This statement must be in the program with a line number. CAUTION..Do not execute the program with this statement, until you make a copy of your program. When this statement is executed, it will clear everything in memory and bring up the MASTER TITLE SCREEN.

4...To protect a disk from being copied with the disk manager, do the following: When you initialize a disk, prior to saving any files to it, you are prompted for certain characteristics for the disk. When prompted for SINGLE-SIDED, enter FCTN X, ten times. You will then get the symbol <> at the top of your screen. After this, proceed as normal. One other word on this. You can NOT remove this priority protection without the aid of DISK FIXER or by re-initializing the disk.

5...Here is a PEEK address that you might find useful. It is from the SPCHRD equate (speech read) located at Hex >9000. CALL PEEK(-28672,A) IF A=96 THEN.....the Speech Synthesizer is attached to the computer. IF A=0 THEN.....the Speech Synthesizer is not attached.

A FINAL NOTE ON ALL OF THIS. YOU MUST HAVE EXTENDED BASIC WITH THE MEMORY EXPANSION ATTACHED IN ORDER TO UTILIZE CALL INIT, CALL LOAD, or CALL PEEK.

WIRE IT YOURSELF

ITEMS IN THIS SECTION ARE SUBMITTED BY THE CALLERS. THEY ARE MEANT TO BE AN AID IN INTERFACING YOUR T.I. COMPUTER EQUIPMENT. I TAKE NO RESPONSIBILITY FOR THE ACCURACY OR THE COMPATIBILITY OF THE FOLLOWING SUGGESTIONS. THEY ARE IDEAS THAT THE SUBMITTER HAS DISCOVERED OR DEVELOPED, AND HAVE PUT TO THEIR OWN USE, AND WANT TO PASS IT ALONG TO EVERYBODY.

ALTERNATE DRIVE INTERFACES - Submitted by Tom Burke

If you are thinking of adding a second drive to your system, and it is not a T.I. drive, it may or may not work by connecting it to the external drive connector on the back of the Disk Controller Card.

If it does NOT work with the T.I. recommended installation procedure, try this.....

You will need the following:

An adequate length of 34 conductor ribbon cable. (2 to 3 feet)

1 - female 34 pin connector. This will connect to the controller card inside the P.E. box. Part #41-908 6C

```
<----->
<.....>
<.....>
<.....>
<----->
```

It should look something like this. 2 - female 34 contact edgcard connectors. Part #41-946. This makes a total of three connectors. The part #'s are from RESCO Electronics. Use the crimp-on type connectors. Put part #41-908 6C on one end of the cable. Plug this into the controller card inside the PE box. Place the first edge card connector approx. eight to 12 inches from the controller card. Crimp it on the cable.

Now you should loosen the screw on the right, rear, top of the outside of the PE box enough so that you can slide the other end of the cable out through the gap. When this is done, plug the middle connector onto the internal drive, (don't forget to plug the power cable back on the drive), and put the drive back into the PE box, pulling the excess cable through the gap.

DO NOT INSTALL THE DRIVE PERMANENTLY AT THIS TIME. JUST LEAVE IT SIT IN THE FE BOX.

Next, crimp the remaining connector on to the other end of the cable. Bare in mind that when you do put the connectors on the cable, that the same wires must be connected to the same numbered contacts on all three connectors. (e.g. wire one goes to pin one on connectors 1, 2, 3) Plug the last connector onto the external drive. (contact one to contact one, etc). Now test the system. You may or may not have to remove the resistor from drive #1. Information on where the resistor may be found is in the Controller Manual. Any questions, leave me a message and I will try to answer them.

PRINTER CABLES - Submitted by Jon Bohos

The following are the pin connections for for a parallel interface (PIO) from your PS232/C to your parallel port on your printer.

You will need the following:

- An adequate length of 16 conductor ribbon cable.
- One 36 connection male Centronics connector.
- One 16 connection female PIO port connector.

Wire the connectors together as follows:

PRINTER SIDE	<>	PIO CONNECTOR
1-----	<-----	1-----
2-----	<-----	2-----
3-----	<-----	3-----
4-----	<-----	4-----
5-----	<-----	5-----
6-----	<-----	6-----
7-----	<-----	7-----
8-----	<-----	8-----
9-----	<-----	9-----
10-----	<-----	10-----
11-15	<-----	not used
16-----	<-----	16-----

NOTE: For an Epson RX/80 printer, you must wire pin 13 to pin 13.

ATLANTA, GEORGIA TIBBS

AN INPUT ROUTINE IN ASSEMBLY LANGUAGE BY JIM RICE

To be able to use the assembly routine I've listed in this article, you must have the console, the Editor/Assembler cartridge, a disk drive, and the memory expansion card or peripheral. Type the program I've listed at the end of this tutorial through the editor. Save the file as "DSK1.INPUTS", then assemble it. The object code file is "DSK1.INPUT". When you assemble it, "R" is the only option you need to use. This INPUT routine is extremely useful as a subprogram. I have left the listing as simple as possible to make it easily adaptable to any type of application. The routine I've listed here places a nonflashing cursor in the upper left hand corner of the screen. It waits for you to type something. After you do, press enter. It will read what you typed in off of the screen, save it into the CPU RAM scratchpad, and print it back up on the screen in a different location. This saves what you typed in at CPU RAM address(>8300), the starting address of the CPU RAM scratchpad, for easy access for the remainder of the program. This is handy for when you want variables in your assembly programs or for writing to files through the DSR(device service routines) such as disk files, a printer, or a modem.

HERE IS THE LISTING:

```
REF KSCAN,VSBW,VMBW,VMBR * References to the different memory
* resident routines used
DEF RUN * Program name defined
CHAR DATA >007C,>7C7C,>7C7C,>7C7C * Cursor character data.
CURSOR BYTE >A0 * ASCII code for cursor.
SPACE BYTE >A8 * ASCII code for space character.
RUN
LI R0,>D00 *
LI R1,CHAR **** Loads the cursor character data into the cursor
LI R2,8 **** ASCII code.
BLWP @VMBW *
CLR R0
CLR R1
LI R2,>0D00 * Loads register(R2) with ASCII code for "enter" key.
LI R3,>2000 * Loads R3 with byte used to check if key has been pressed
LI R5,>0800 * Loads R5 with ASCII code for backspace(fctn-s) character.
CLR R6
MOVW R0,@>8374 * clears @>8374(tells computer to scan whole keyboard.)
LI R0,1 * loads R0 with 1(tells computer cursor to go in upper left)
More.(A=Abort, any other key to cont.)
J1 LI R1,CURSOR **** Prints cursor to screen at location indicated by R1
```

```

BLWP @VSBW *
CLR R1
BLWP @KSCAN * Scans keyboard
MOVB @>837C,R6 * Loads byte that shows whether a key was pressed into R6
CDC R3,R6 * Checks if key was pressed.
JNE J1 * If not, goes to J1
MOVB @>8375,R1 * Loads key pressed into R1 from address key code stored
CB R1,R2 * Checks if key pressed is "enter"
JED J3 * If so, go to J3
CB R1,R5 * Checks if key pressed is backspace(fctn-s)
JNE J2 * If not, go to J2
MOVB @SPACE,R1 * load space ASCII code to R1
BLWP @VSBW * Put space where cursor was
DEC R0 * Move cursor back 1 space
JMP J1 * Go to J1
J2 BLWP @VSBW * Write key detected to the screen.
INC R0 * Move cursor forward 1 space
JMP J1 * Go to J1
J3 MOVB @SPACE,R1 * Load R1 with ASCII code for space
BLWP @VSBW * Write space to screen where cursor was
CLR R0
CLR R1
CLR R2
LI R0,1 * Load R0 with starting address of variable inputed.
LI R1,>8300 * Load R1 with CPU RAM SCRATCHPAD startin address
LI R2,80 * Move 80 bytes of characters typed on to screen
BLWP @VMBR * Read variable input from screen & write it to CPU(>8300)
CLR R0
CLR R1
CLR R2
LI R0,513 * Load R0 with 513(new address on screen to print variable to.)
LI R1,>8300 * Load R1 with address of data to be read(CPU SCRATCHPAD.)
LI R2,80 * Move 80 bytes from CPU scratchpad to new screen location
BLWP @VMBW * Write variable inputed back to new screen location.
LIMI 2 * Enable quit key
JUMP JMP JUMP * Wait for quit key to be pressed.

```

END

The actual storing of the variable in CPU RAM stops after the BLWP @VMBR statement. After that, the program is just reading the variable that was typed in back to the workspace registers and printing it back on to the screen at screen location 513. You can change where it is reprinted on the screen by changing the 513 to another number. The number must be between 0 and 758. You can compute the number by multiplying the row you want times 32 and adding the column to the product.

You don't have to save your variable to the CPU RAM scratchpad. You could theoretically save it to any CPU memory location. If you don't use the CPU scratchpad, I recommend the memory expansion high RAM(>A000->FFE0). To change location the variable is stored at, change the two times >8300 is used in the program to the new address you want to use. Also, you can save memory by making your variable buffers smaller. Mine, in this case is 80 bytes. That means that 80 characters of what is typed in are saved in the CPU RAM scratchpad. You can make your buffer larger or smaller to fit your needs by changing the two 80's located just before the BLWP @VMBR and the last BLWP @VMBW to however many bytes you want stored in your buffer. To run the routine, the file name is "DSK1.INPUT" and the program name is "RUN"

HAPPY COMPUTING!!!! Jim Rice

DOUBLE-SIDED MASTERDISK MODIFICATION From Larry Wilson

Informatin from Pronto System Comm Link TI 886 in Tampa, Fla. For those of you with double-sided drives and MASTERDISK, the following modifications will allow the program to hold up to 250 disks of information or a listing for 3000 programs.

1. Place MASTERDISK onto a double-sided disk.
2. Access the program named AD.
3. Call up Line 540 and change "IF DT<121" TO "IF DT<251".
4. Call up Line 550 and change "IF PF>1100" TO "IF PF>3000
5. SAVE file and RUN program.

THAT'S ALL FOR THIS MONTH, FOLKS!!!!!!

NATIONAL X/B LIBRARY PROGRAM AGREEMENT

The program(s) included in this package are donated to the 99'ERS USERS GROUP ASSOCIATION Library. The programs may be included in said Library being assembled for distribution by this organization to Users Groups supporting the TI-99/4A Home Computer. I acknowledge that these programs will become "PUBLIC DOMAIN" and may be copied and distributed to the Users Groups and their members. I do not desire or expect any financial restitution for the programs submitted. All rights to these programs and any ownership claims are released.

The 99'ERS USERS GROUP ASSOCIATION accepts these programs as donated and will place them in the X/B Library as Public Domain programs for distribution to TI Users Groups. The 99'ers UGA does not accept any responsibility regarding the donated program(s) origin and must assume the person submitting the programs created them or obtained them from a non-commercial source. Credit will be provided to the program author whenever this information is forwarded to our organization.

NATIONAL X/B LIBRARY
99'ERS USERS GROUP ASSOCIATION
ATTN: CURT PURDY
2314 MIDWAY DR.
PHOENIX CITY, ALA. 36867

NATIONAL X/B LIBRARY DONATION FORM

NAME: _____ DATE: _____

ADDRESS: _____

CITY: _____ STATE _____ ZIP _____

THE X/B PROGRAMS ENCLOSED WITH THIS PACKAGE ARE DONATED TO THE 99'ERS USERS GROUP ASSOCIATION NATIONAL X/B LIBRARY. I UNDERSTAND THAT THE DONATED ITEMS SHALL HEREAFTER BECOME "PUBLIC DOMAIN" AND MAY BE DISTRIBUTED TO TI-99/4A OWNERS WITHOUT RESTRICTION. I DO NOT EXPECT TO RECEIVE ANY COMPENSATION NOW OR IN THE FUTURE FOR THE PROGRAMS DONATED. I HAVE LISTED THE NAMES OF PROGRAMS DONATED TO THE 99'ERS USERS GROUP ASSOCIATION ON THE BACK OF THIS FORM. I WILLINGLY RELEASE ALL INTEREST IN THE ENCLOSED DONATED PROGRAMS.

SIGNED: _____

PRINT NAME: _____

DATE: _____

**SUBSCRIPTION FORM
99'ERS U.G.A.**

NAME: _____

DATE: _____

ADDRESS: _____

CITY: _____ **STATE** _____ **ZIP** _____

MAIL CK./M.O. FOR \$12.00 TO:

**99'ERS USERS GROUP ASSOCIATION
3535 SO. H ST., #93
BAKERSFIELD, CALIF. 93304**

ATTN; DONALD VEITH

3/84

**THE NATIONAL 99'ER
3535 SOUTH H STREET #93
BAKERSFIELD, CA 93304**