FOR THE RECORD

by Ed Bittner
Recording Secretary

The March meeting of the West Penn 99er,s came roaring in on the twenty-first of the month with President Mickey Schmitt opening the session at 7:22 PM. The late start, I presume, was the result of the waiting for the late arrival of the recording secretary (me) but maybe ego is getting the better part of me. (Rec. Sec. note-probably not). The minutes of the Feb. meeting were accepted with accolade and praise (Rec. Secs. recollection) and the treasurers report was flourishing (so were the goodies).

Rob Ekl, our librarian, announced that our MacFlix collection has grown to 70 diskettes. WOW ! Also the bare bones TI Tacks is available at ?? prices. Gene Kelly (the B B S er, not the dancer), re-announced the club contest and that the deadline for submission would be the upcoming April 18 meeting. Joe Ekl, the membership chairperson introduced a new member Ray Broderick. Welcome Ray. By the way, old members, DUES are DUE. for the 1989 year.

To old business, Mickey announced that a complete inventory of all club equipment has been accomplished; also that we as a club still sell data cases, books, micropendiums, and diskettes, all at a great savings to club members at every meeting.

Upcoming shows have dominated the new business as of late and stories of fun had by all who participate in them are entertaining. Mickey suggested for the Harrisburg show that the club might want to get a table in addition to the adventure table which she will be running. If your interested in the Harrisburg show contact G. Taylor. The Boston show (April 1) has passed and we look forward to another report of the misadventures of Mickey and the "show people" at the April meeting. Upcoming shows are in Ottawa (April 29) and Lima (May 20) ,thats Ohio, not Peru, Columbus.

Editor in chief, or mischief, John Willforth collected himself for a few moments and entertained us with information lists on Rave 99, Ramcharge Computer Prices, Genial Computer Catalog, Asguard Catalog, as well as some E-proms and other good stuff.

Scheduled demonstrations included Typewriter 99 (Jim Reiss) shown by Mike Sealy; Recipe Writer by Paul Brock, and The Mind as well as a Speech Editor by J. Willforth. The raffle was well supported again by all, Prizes included data cases, diskette pens, Typewriter 99 and and Atari Joystick adapter.

Humbly submitted,
Scoops Bittner

WEST PENN 99'ERS CLUB INFORMATION

NEXT MEETING DATE: APRIL 18 1989

MEETING LOCATION: UNITED PRESBYTERIAN
                            CHURCH OF THE
                            COVENANT

                        CORNER OF 4TH AND
                        OAK STREETS, IRWIN

TIME OF MEETING: 7:00 P.M.

LIST OF WEST PENN OFFICERS FOR 1989

PRESIDENT:	MICKEY	335-0163
VICE PRESIDENT:	SCOTT	523-3754
TREASURER:	JAN	863-1575
RECORDING SEC:	ED	864-4924
CORRESPONDING SEC:	GENE	829-0469
LIBRARIAN:	ROB	864-1233
NEWSLETTER EDITOR:	JOHN	527-6656

GENERAL ITINERARY OF THE CLUB'S MEETING

6:45 P.M.	DOORS OPEN
7:00 P.M.	GENERAL MEETING
7:45 P.M.	DEMOS AND NEW INFO
8:45 P.M.	HARDWARE CLASS
8:45 P.M.	INTRO TO FORTH
8:45 P.M.	TIPS FOR BEGINNERS
8:45 P.M.	USING YOUR CASSETTE
11:00 P.M.	DOORS CLOSE

MEETING HIGHLIGHTS FOR THIS MONTH

NEW T.I. PRODUCTS, FROM THE BOSTON SHOW

GENIAL'S CHAINLINK, DEMO BY MIKE SEALY

CASSETTE EA#5 LOADER, DEMO BY BILL VORP

LIBRARY "DEMO OF THE MONTH" BY ROB EKL

LATEST SOFTWARE DEMOS BY JOHN WILLFORTH

RENEW YOUR MEMBERSHIP DUES!

$15.00 PER YEAR FOR INDIVIDUAL / FAMILY
$10.00 PER YEAR FOR JUST THE NEWSLETTER

TREASURER'S REPORT FOR MARCH 1989

FROM JAN TRAYERS

3/21	CASH ON HAND	$ 56.50	
"	LIBRARY SALES	36.00	
"	MICROPENDIUMS	28.50	
	TI BOOKS	22.00	
	DISKS & CASES	84.00	
	DUES	50.00	
	TI BASE	21.00	
	TOTAL	$331.75	
4/6	DEPOSIT	- 281.75	
4/6	CASH ON HAND	50.00	

3/21	BANK BALANCE	$1420.10	
3/21	RAFFLE PRIZES	-100.00	
		1320.10	
"	JOHN (POSTAGE)	- 77.30	
		1242.80	
4/1	MICROFENDIUM	- 30.00	
		1212.80	
4/6	DEPOSIT	+ 281.75	
	TOTAL	1494.55	

TOTAL CASH BALANCE $1544.55

OFFICIAL WP99 COVER BANNER CONTEST DEADLINE NEARS........

The CONTEST announced in the Feb. issue of the West Penn 99'er will be officially over on April 18,1989 at the meeting. You have until then to get your entry in. Read the rule book in Feb. issue, and let's see your entry both in hard-copy and on disk in a TI-ARTIST format as a full picture. There will be multiple prizes. And you must be a PAID UP MEMBER! Bring it but don't show i to anyone because the membership is doing the judging by secret ballot. Contact any of the officers above.

Notes:

This section is about UNIT's. A UNIT is a separately compiled procedure or function that is called by a program and that program is also compiled without the procedures or functions that the program needs.

What this allows us to do is to make a lot of separate procedures and functions (sort of a library) independent of a program and then use what ever procedure we want with a program without having to recompile the procedure (so long as we do not want any changes in the separately compiled procedures).

Let us look at a simple system. Below is a program and a procedure which will determine the factorial of a number.

```
Program Main;
Uses {$U Mydisk:Math.Code} Math;
Var Number:Integer;
Begin
 Repeat
  Write('Input a number: ');
  Readln(Number);
  Writeln(Number, ' Factorial is ',Factorial(Number));
 Until Number=0;
End.
```

Notice that this program does not contain the function called Factorial. If more that one procedure or function is used in a program of this type each one would be referenced as shown above using the rules of Pascal. That function is on disk as a separate file as show below;

```
Unit Math;
Interface
Function Factorial(N:Integer):Integer;
Implementation

Function Factorial
Begin
 If N=0
Then
 Factorial:=1
Else
 Factorial:=N*Factorial(N-1)
End;
Begin
End.
```

Compile the "Unit Math" function first naming it "Math" on a disk called "Mydisk". (You can substitute your current disk name for the "Mydisk" name in the "Uses {$U Mydisk:Math.Code} Math" line. You could also use the drive number). Then compile the main program saving it as any filename that you wish.

A short explanation of the "Unit Math" function. There are several things different about this piece of code. The term "Unit"; this is a piece of syntax that tells the compiler that this is a "Unit" and not a complete program. The term "Interface"; If we had a more complicated piece of code that had several procedures and functions in it, the only procedures or functions that could be accessed by the main program would be those that were declared in the "Interface" section. Those procedures or functions that were not declared would be hidden from the main program and could not be called by anything but the Compiled Unit itself. The "Implementation" section has the actual procedures or functions that are to be compiled and used. This piece of code ends with a "Begin..End." pair that has a period at the end in order to make it look like a "program" to the compiler. The functions or procedures declared in the interface must have the values that it accepts and/or passes and these values are not repeated again when the function or procedure is completely written. It is like a "Forward" called procedure or function. If a procedure or function is not in the Interface and it is passing values then the procedure or function will be written normally in the Implementation section. There are some restrictions dealing with UNITS but these are clearly spelled out in the Compiler documentation.

Enough this time, next time we will discuss how to run the program. It took a bit of digging in order for me to find out how to do it. I feel that the instructions in the Pascal documentation were confusing.

DISK TO CASSETTE TRANSFER
BY BILL VORP (WP99)
CASSETTE LOADER FOR 32K
MEMORY EXPANSION

WEST PENN 99'ER

APRIL 1989

These two programs are written by Will McGovern of Funnelweb fame. As you would expect from him, they are offered as Fairware, so a contribution is in order if you use them. The cassette loader program requires 32K and Extended Basic. The cassette transfer program requires a disk.

The cassette transfer program allows you to put E/A program image (option 5) files on cassette. The proceedure is very simple. When the program has been loaded the Fairware reminder is displayed and below that are two lines that ask for input filename and output filename. The input filename is whatever the file is called on the disk, such as **DSK1.OHMUMMY**. The output filename is for the cassette file. Not knowing exactly how this program worked, I used **CS1.OHMUMMY**, and it worked fine. After you enter the filenames a message is displayed:

        Current filename: **DSK1.OHMUMMY**
        Press **ENTER** to load file
After the file is loaded, another message is displayed:
        Current filename: **CS1.OHMUMMY**
        Press **ENTER** to save file
When you press **ENTER** this time, the familiar cassette **SAVE** routine appears. Follow them as usual. If there is a second file to the program, the transfer program will auto-increment the filename and display:
        Current filename: **DSK1.OHMUMMZ**
From here the proceedure is the same as saving the first file in the program, except you should not leave any space between the two program segments on the cassette. The reason becomes apparent when you use the cassette loader program.

Before I go on to the cassette loader program, I'll explain a mistake I made, so others won't. I tried to transfer the loader program to cassette using the transfer program. It won't work that way since the cassette loader program is an Extended Basic program, not E/A program image. To put it on cassette simply load it from disk using **OLD DSK1.CASSLOAD**, then **SAVE** it to cassette using **SAVE CS1**. Don't type **RUN** after you have loaded it, since **FCTN 4** will not "break" the program.

The cassette loader program loads normaly from cassette, using **OLD CS1**. When you type **RUN**, a Fairware reminder screen comes up with the message **PRESS ENTER** near the bottom of the screen. When it is pressed, the familiar cassette loading routines appear. Follow the instructions to load the first segment of the program. After it is loaded a message is displayed:

IGNORE REWIND TAPE MESSAGE
If the second segment appears immediately after the first, pressing **ENTER** twice will start the second segment loading. When all segments have loaded and you press **ENTER** following the prompts, the program will start.

I just used a lot of words to explain a very easy to use program. I think it will be welcomed by all users that have 32K memory expansion and a cassette system only!

ED. NOTE: This will be demonstrated at the APRIL meeting.

One day, I got a HFDC card and a Miniscribe 20 meg harddrive. When I got it installed, I loaded up the disk manager program, and formatted the harddrive with all the defaults. I messed around with it for a little while, copying disks to it and everything worked great.

After a little while, I read the manual some more, and it says to reformat the harddrive using several different Interlace and Step Rate setting. So I loaded up the disk manager program again, and used a different, faster interlace setting. When the program came to the point to start formatting the drive, the access light stayed on, and the screen showed the disk was not formatting. The only way to abort was to reset. So I did.

When I went to reload the disk manager program, I found the disk controller was accessing the harddrive first to check for a floppy emulation file, and because of the incomplete format, the first track of the harddrive was corrupt, and I received an error when I tried to load the disk mananger.

The next thing I did was disconnect the cables to the harddrive from the disk controller and then I could load the disk manager. I went through, and when the format routine was ready to access the harddrive, I connected it back up. Then it said "Drive not present" for the harddrive, which means it didn't find the harddrive because I disconnected the cables and the disk controller determined that I had no harddrive in the system and wouldn't change it's mind.

I tried changing the harddrive number from Harddisk #1 to #2 and #3, thinking something might change. All stayed the same.

I put the harddrive back to #1, and tried changing the floppy device to DSK2 and tried loading the disk manager from there. No luck, because the loader for the main disk manager files always looked to DSK1. I stopped playing with it for that night, and I talked to John Wilforth. He suggested changing the disk manager to drive 2, and I told him that I already tried that, and he asked me if I had used a sector editor to change all the DSK1's to DSK2's. Hmmmm.

I backed up the disk manager disk and searched and changed all references from DSK1 to DSK2. Then I tried to load it from with Extended Basic loader, which has an assembly command to directly access drive one, which I could not easily change, it didn't work. So I tried it from the Editor Assembler loader, and the disk manager loaded. I went all through the formatting routine, using the defaults again, and it formatted fine. I haven't tried to reformat it again.

There are two places you must change DSK1 to DSK2, in the MDM5 file, sector >69, byte >60, and in the MDM287 file, sector >69, byte >6C. Make these changes on a backup ONLY, and to whatever floppy drive you want MDM5 to load from.

* *

Thank you Rob! I think that with more of you getting into the HRDC from MYARC there will be more and more achievements and failures. Let's try to emphasize the achievements, and turn the failures into successes. J.F.W.

This is the first of what I hope will be a series of non technical articles aimed at users who don't want to program but who would benefit from learning more about some of the enhancement utilities that are available for extended basic. These utilities are often no more difficult to use than DM-1000.

For our first example let's use Barry Boone's SYSTEX to improve the loading speed of the fantasy game CARFAX ABBEY. CARFAX ABBEY is a fairware game by David Vincent of Kent UK. This game uses 7 assembly language graphic routines that take over 1 minute to load. After these programs are in memory, the main program begins to load. From start to finish it takes about 3 minutes to get the program booted. Now this is a short time in geological terms, but it is forever and a day to a computer user soooo let's improve it.

First you'll need one disk drive , a back-up copy of CARFAX ABBEY and a copy of SYSTEX. Now turn on the TI. At the READY prompt type:

        CALL INIT::CALL LOAD("DSK1.GRAPH1","DSK1.GRAPH2","DSK1.G
        RAPH3","DSK1.GRAPH4","DSK1.GRAPH5","DSK1.GRAPH6","DSK1.G
        RAPH7")

Place the disk containing CARFAX ABBEY in drive 1 and press enter. Now read the paper until the load is done. Place the SYSTEX disk in drive 1 and type:

        RUN "DSK1.SYSTEX"

You will soon be asked if the assembly language routine has been loaded. Press "Y" then "ENTER". Now type:

        11 RUN "DSK1.CARFAX"

Press "ENTER". Now type

        SAVE DSK1.GRAPHICS

Place a formatted disk in drive 1 and press "ENTER". Believe it or not we have just performed a very delicate operation, 7 assembly language routines have just been disguised as 1 fast loading EXTENDED BASIC program. There will also be a saving of about 47 sectors of disk space. Place the copy of CARFAX ABBEY in drive 1 and type:

        OLD DSK1.CARF/LOAD

Press enter. At the READY prompt type 440 and press the down arrow (Fnct x) you should see:

        440 CALL INIT :: CALL LOAD("
        DSK1.GRAPH1","DSK1.GRAPH2","
        DSK1.GRAPH3","DSK1.GRAPH4","
        DSK1.GRAPH5","DSK1.GRAPH6","
        DSK1.GRAPH7")

As you can see this is a step that we want to eliminate so press FCTN 3 followed by FCTN x. The display should now say

        450 RUN "DSK1.CARFAX"

Rewrite the line to read

        450 RUN "DSK1.GRAPHICS"

Now press ENTER, place the disk containing the the file we just created into drive 1, type

        SAVE DSK1.LOAD

Press enter. The final step consists of transferring the needed files from the old CARFAX ABBEY disk to the new one the files that must be moved are:

            CARFAX
            CARFAX/DOCS
            HELP (if your disk has a file called CHEAT/NOTES,
                   copy it and change its name to HELP)

I hope you have tried the steps listed above and don't forget to pay the authors if you make these programs a part of your library. SYSTEX can be applied to other programs like Steve McWatty's GRAPHIC LABELER or many others that load in assembly programs. Try it out, you'll get an error message if it doesn't work and you'll lose about 3 to 4 minutes of your time.

P.S. Let me know if the instructions are too detailed. I am aiming at the tyro and devout non programmer feedback from any source will accepted and acted upon as time and ability allow.

These changes are recommended for ALL HORIZON RAMdisks and are compatible for use with the TI99/4a or Geneve.

1. RESET on power-up

This change allows the computer to reset the HORIZON during the CPU power up cycle. The reset feature, as TI designed it, does provide a reliable method to hold the HORIZON in the shut-off state until the PE-Box voltage has been on long enough to stabilize.

The modification consists of the removal of one diode, one resistor and one capacitor. These parts are replaced by one wire from pin 6 of the card-edge connector (bottom edge of ramdisk card) to the positive side of the capacitor location.

HORIZON serial numbers below 100:
Remove C8, CR2 and R2. Connect wire to front (or left) hole of C8 location.

HORIZON serial numbers above 100:
Remove C1, CR3 and R5. Connect wire to + (positive) side of C1 location.
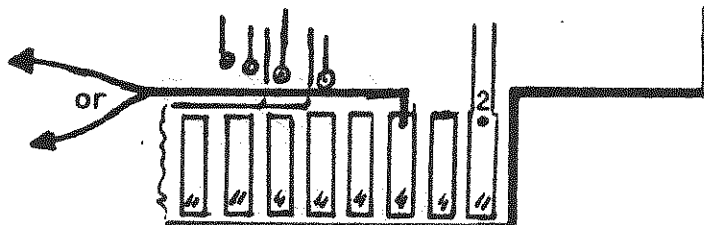
Connect other end of wire to pin 6 of card-edge, i.e., the 3rd lead from the right on the COMPONENT side of the PC board.

2. DISABLE SWITCH

This modification provides a method to turn off (or hide) the HORIZON from the rest of the system. This switch allows you to turn off the ramdisk in the event of a system crash when the computer locks up. With the card turned off, you can power up the console and PE-Box, turn the card back on and proceed to re-load the operating system. No need to remove the batteries to erase the contents and in most cases the files may be recoverable. Other reasons for "hiding" the card could be a conflict between the ramdisk and a program you want to run - or you may wish to keep the kids out of it.

The mod is simple: We remove the voltage from pin 6 of U20 (serial 1999 and below) or U20A (HRD+,2000 and up) and reconnect it via a resistor (1K-10K will do) thru a SPST switch to ground. Closing the switch pulls the pin low and shuts off the CRU access at U20.

Bend pin 6 of the chip out, attach enough wire to reach the switch and connect the resistor from this pin to pin 16 of the same chip. Run the other end of the wire to the switch.

NOTE: The HRD+ circuit board on cards with a serial number below 1999 required stacking of U20. Attach the wire and resistor to the top chip's pin 6 and cut off the bottom end.

Mount a miniature SPST at the top back edge. Run a lead from one pole to a nearby ground.
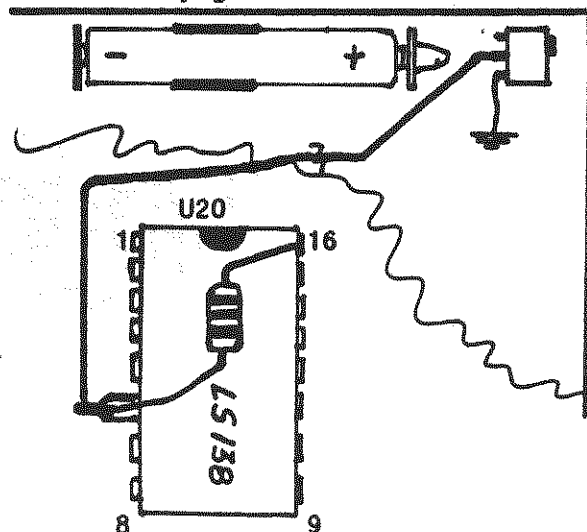
OPTIONAL HORIZON 32K MOD TO REPLACE PE-BOX 32K CARD
by John Guion and Bud Mills

```
Parts List:   2 ea. 1N34A diodes            1 ea. 14-pin socket (optional)
              1 ea. 74LS08                  1 ea. 28-pin socket (optional)
              1 ea. HM62256-LP12 or 43256-12L   Hook-up wire
```

Use of the optional sockets will allow you to replace a defective chip should a
failure occur. Also, this additional memory could be disabled by unplugging the
chips. The memory MUST be installed on top of the U11 chip, but its control pins
and pin 28 (Vcc) must be isolated from the HORIZON card. Note that the address
and data lines are shared and that the separation of the control lines assures
proper data handling.

To install this modification, use the sketch below for reference and:

1. Place the 74LS08 chip (or the optional 14-pin socket) on top of U18. Bend all
   but pins 7 and 14 out for connection of wires later. Solder pins 7 and 14 of
   the chip or socket to pins 8 and 16 of U18. Pin 7 needs to be spread slightly
   to reach pin 8 of U18. Note: HRD+2000 cards can use U24 pins 7 and 14 only.
   The HORIZON 3000 will use U25 pins 7 and 14. (U24 and U25 were set up for the
   Phoenix mod.)

2. Place the 32x8 memory chip (or the optional 28-pin socket) on top of U11.
   Bend pins 1, 20, 26 and 28 out for connection of wires later and solder all
   other pins to U11.

3. Install the two diodes as shown below. Make sure the cathode band is oriented
   correctly, i.e., toward the new chip/socket on top of U18. Connect one anode
   to the anode of CR5, the other to anode of CR7. Connect both cathodes to pin
   8 of the new chip/socket.

4. Connect wires as follows:
   ```
   From 74LS08  pin # 1 to pin  9 of U20 *)     pin 2 & 4 to pin  7 of U20 *)
     socket)          5 to pin 10 of U20 *)     pin       9 to pin 14 of U20 *)
              3 & 13 to pin  1 of U11           pin 6 & 12 to pin 26 of U11
                   8 to pin 20 of U11 and the diodes
                  14 to pin 28 of U11
   ```
   Connect pin 10 of the 74LS08 to pin 11 of the same chip/socket

   *) If U20 is stacked, make connections to the chip closest to the board.
      Wires may be run on the back side of the card.

5. Double check your wiring, plug in the chips (if sockets were used). Remove
   old 32K memory card from PE-Box. The easiest way to test if your new memory
   expansion works properly is to use Extended Basic. The SIZE command should
   result in a display of 11840 BYTES OF STACK FREE, 24488 BYTES OF PROGRAM
   SPACE FREE. A major deviation from these values indicates a problem. As a
   final test run an assembly language program (like DM1000). If a problem does
   exist, the program will not run.
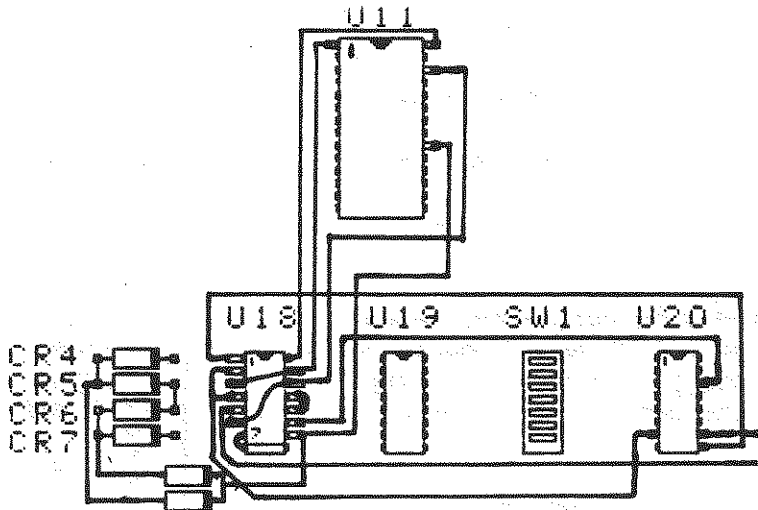
TIPS FOR THE MONTH

by Bob August

Saving and retrieving your data to a disk drive or cassette is easy. We have two samples below for the disk drive and the cassette recorder. The first one is for one bit of information and the second one is for multiple bits of information. Remember that if you are entering or saving more than 10 bits of information you must dimension statement "DIM N$(15)". You can use any number between 1 and 255 to open your file. You also must close you file after saving or loading your data. Play with the samples and you will feel at home with you disk drive or cassette recorder. The disk drive defaults to display variable 80 and can be read with TI-WRITER. Here are the samples in BASIC, so if you are using Extended Basic, use : : where you see ::.

```
100 REM CASSETTE SAMPLE 1
110 CALL CLEAR
120 PRINT "PLEASE ENTER YOU
NAME"::
130 INPUT "":NAME$
140 PRINT ::"SAVING DATA"
150 OPEN #25:"CS1",OUTPUT,FI
XED,INTERNAL
160 PRINT #25:NAME$
170 CLOSE #25
180 PRINT ::"ENTER: RUN 190"
190 STOP
200 CALL CLEAR
210 PRINT "LOADING DATA"
220 OPEN #10:"CS1",INPUT ,FI
XED,INTERNAL
230 INPUT #10:N$
240 CLOSE #10
250 PRINT ::"HERE IS YOUR NA
ME"::N$
260 END
```

```
100 REM DISK DRIVE SAMPLE 1
110 CALL CLEAR
120 PRINT "PLEASE ENTER YOU
NAME"::
130 INPUT "":NAME$
140 PRINT ::"SAVING DATA"
150 OPEN #25:"DSK1.DATA",OUT
PUT
```

```
160 PRINT #25:NAME$
170 CLOSE #25
180 PRINT ::"ENTER: RUN 190"
190 STOP
200 CALL CLEAR
210 PRINT "LOADING DATA"
220 OPEN #10:"DSK1.DATA",INP
UT
230 INPUT #10:N$
240 CLOSE #10
250 PRINT ::"HERE IS YOUR NA
ME"::N$
260 END
```

```
100 REM CASSETTE SAMPLE 2
110 CALL CLEAR
120 PRINT "PLEASE ENTER 3 NA
MES"::
130 FOR N=1 TO 3
140 PRINT :"NAME: #";N::
150 INPUT "":NAME$(N)
160 NEXT N
170 PRINT ::"SAVING DATA"
180 OPEN #7:"CS1",OUTPUT,INT
ERNAL,FIXED
190 PRINT #7:N
200 FOR NAME=1 TO N
210 PRINT #7:NAME$(NAME)
220 NEXT NAME
230 CLOSE #7
240 PRINT ::"ENTER: RUN 260"
250 STOP
260 CALL CLEAR
270 PRINT "LOADING DATA"
280 OPEN #5:"CS1",INPUT ,INT
ERNAL,FIXED
290 INPUT #5:HOWMANY
300 FOR N=1 TO HOWMANY
310 INPUT #5:N$(N)
320 NEXT N
330 CLOSE #5
340 PRINT ::"HERE ARE YOUR N
AMES"::
350 FOR NAME=1 TO N
360 PRINT :N$(NAME)
370 NEXT NAME
380 END
```

```
100 REM DISK DRIVE SAMPLE 2
110 CALL CLEAR
120 PRINT "PLEASE ENTER 3 NA
MES"::
130 FOR N=1 TO 3
140 PRINT :"NAME: #";N::
150 INPUT "":NAME$(N)
160 NEXT N
```

```
170 PRINT ::"SAVING DATA"
180 OPEN #7:"DSK1.DATA",OUTP
UT
190 PRINT #7:N
200 FOR NAME=1 TO N
210 PRINT #7:NAME$(NAME)
220 NEXT NAME
230 CLOSE #7
240 PRINT ::"ENTER: RUN 260"
250 STOP
260 CALL CLEAR
270 PRINT "LOADING DATA"
280 OPEN #5:"DSK1.DATA",INPU
T
290 INPUT #5:HOWMANY
300 FOR N=1 TO HOWMANY
310 INPUT #5:N$(N)
320 IF EOF(5)THEN 340
330 NEXT N
340 CLOSE #5
350 PRINT ::"HERE ARE YOUR N
AMES"::
360 FOR NAME=1 TO N
370 PRINT :N$(NAME)
380 NEXT NAME
390 END
```

----

by Bob August

The program this month gives you a nine by eleven multiplication table. This would be good for your children that are just learning their multiplication tables or as a review. The first version is in Basic and the second version is in Extended Basic.

BASIC VERSION

```
100 REM ELEVEN BY NINE
110 REM MULTIPLICATION TABLE
120 REM BY R.W. AUGUST
130 DIM M(11,9)
140 CALL CLEAR
150 CALL SCREEN(8)
160 CALL CHAR(96,"FF")
170 CALL CHAR(97,"0808080808
080808")
180 CALL CHAR(98,"FF08080808
080808")
190 PRINT TAB(6);"MULTIPLICA
TION TABLE"::
200 FOR A=1 TO 11
210 FOR B=1 TO 9
220 M(A,B)=A*B
230 IF M(A,B)<10 THEN 330
240 PRINT " ";STR$(M(A,B));
250 IF M(A,B)=99 THEN 350
```

```
260 NEXT B
270 PRINT ::
280 IF A>1 THEN 320
290 PRINT
300 CALL HCHAR(21,5,32)
310 CALL HCHAR(23,4,96,27)
320 NEXT A
330 PRINT "  ";STR$(M(A,B));
340 GOTO 260
350 CALL VCHAR(6,6,97,19)
360 CALL VCHAR(5,6,98)
370 CALL VCHAR(3,6,97,2)
380 CALL KEY(0,K,S)
390 IF S=0 THEN 380
400 END
```

EXTENDED BASIC VERSION

```
100 ! ELEVEN BY NINE
110 ! MULTIPLICATION TABLE
120 ! BY R.W. AUGUST
130 DIM M(11,9)
140 CALL CHAR(96,"000000FF",
97,"0808080808080808",98,"08
0808FF08080808")
150 DISPLAY AT(2,5)ERASE ALL
:"MULTIPLICATION TABLE" :: C
ALL HCHAR(5,3,96,27)
160 FOR A=1 TO 11 :: FOR B=1
 TO 9 :: M(A,B)=A*B :: IF M(
A,B)<10 THEN N=1 ELSE N=2
170 DISPLAY AT(A*2+2,B*3-N):
STR$(M(A,B)):: CALL HCHAR(A*
2+1,5,97):: NEXT B
180 CALL HCHAR(4,4,32):: CAL
L HCHAR(A*2+2,5,97):: NEXT A
 :: CALL VCHAR(5,5,98)
190 CALL KEY(0,K,S):: IF S=0
 THEN 190 :: END
```

THANKS TO "BUG NEWS" APRIL '89

I'm starting to worry that maybe nobody is building the hardware projects that I'm putting in the West Penn 99'ers and the PUG newsletters. Back in February 1987 (2.2 years ago), I published a circuit of a 32K memory board with an 8K SuperCart Ram space. It wasn't until January of this year I had someone call me telling me that he couldn't get the 8K Ram space to work. Then at the end of February, Russ Norman of Redmond, WA called to tell me that everything looked OK except for the decoding of the >6000 - >7FFF address space, which is the SuperCart space. Joe Spiegel designed this circuit, but I had to get my hands into it (never letting anything alone), so I redrew the schematic. This is where I believe the **GREMLIN** enters.

If you got the 32K working, but got frustrated with the 8K memory, take it back off the shelf, and move the wire you soldered on pin 11 of the 74LS138 that goes to pins 1,2 of the 74LS00, to pin 12 of the 74LS138. Pin 11 on the 74LS138 will not be used.

Thanks to Russ Norman for this correction to what is otherwise, (I hope) an accurate circuit.

NO DISK AUTOLOAD WHEN XBASIC SELECTED ....

This first appeared in TI SIG. If you have a Grand Ram or Horizon Ram Disk and use BOOT or MENU program, then this is for you. Type in the following, and save it to the Ram Disk. Put in the menu as an option, and be sure Xbasic is the cartridge active at the time. When the option is selected, you will immediately see the Xbasic prompt. This means that the **LONG DELAY** waiting for the **DSK1.LOAD** to time out is gone!

```
100 CALL INIT
110 CALL LOAD(-31952,255,255,0)
```

A PROGRAM TO PRINT A COVER PAGE ....

I wrote the following program to print the outside cover page where the mailing label goes. It asks for the **ISSUE NUMBER**, the **MONTH**, as well as the **YEAR**, and any **SPECIAL MESSAGE**. It of course prints the **WEST PENN 99'ERS** and my return address and a place is set off for the label to be inserted. You would usually position the paper half way up, so that the final print will be at the bottom, but this is of course optional. I'm including this so that some of you who have not gotten into programing your new printer may look at a relatively easy print program which may easily be altered to meet your needs.
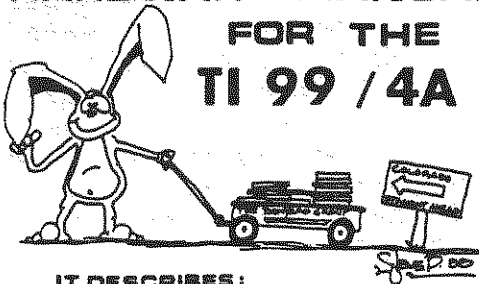
```
100 CALL CLEAR
110 INPUT "ENTER ISSUE NUMBE
R. ":NO$
120 INPUT "ENTER MONTH    ":
MNTH$
130 INPUT "ENTER YEAR     ":Y
R$
140 INPUT "SPECIAL MESSAGE
":MSG$
150 OPEN #1:"PIO"
160 PRINT :"SET-UP PAPER"
170 INPUT "PRESS ANY KEY":A$
180 PRINT #1:CHR$(27)&CHR$(1
20)&CHR$(49)
190 PRINT #1:CHR$(27)&CHR$(6
5)&CHR$(0)
200 FOR LOOP=1 TO 3
210 PRINT #1:CHR$(14)&"WEST
PENN 99'ERS"
220 NEXT LOOP
230 PRINT #1:CHR$(27)&CHR$(5
0)
240 PRINT #1:"":"% JOHN F. W
ILLFORTH":"R.D. #1    BOX 7
3A":"JEANNETTE,       PA":"
            15644"
250 PRINT #1:CHR$(27)&CHR$(6
9)
260 PRINT #1:"":"ISSUE NUMBE
R ";NO$:MNTH$;"-";YR$
270 PRINT #1:"":"":"":"":TAB
(30);"_____
    "
280 FOR X=1 TO 9
290 PRINT #1:TAB(29);"!";TAB
(67);"!"
300 NEXT X
310 PRINT #1:TAB(30);"_____
_____
___"
320 PRINT #1:CHR$(27)&CHR$(6
5)&CHR$(0)
330 PRINT #1:MSG$
```

THE BUNYARD GROUP
P.O. BOX 62323
COLORADO SPRINGS, CO 80962-2323
(719) 488-2576
January 31, 1989

WHOLESALE PRICE LIST

0-14 $21.95 **THE BUNYARD GROUP** pays postage
15-25 $15.00 Purchaser pays postage (UPS)
26-UP $12.00 Purchaser pays postage (UPS)

TERMS: Certified Check or Money Order, or Special Arrangement.

Manuals shipped no later than two weeks after order recieved.

The Bunyard HARDWARE MANUAL is about the best manual you can get to help you understand the T.I. 99/4A Home Computer as well as it's expansion peripherals. They have moved, and if you were thinking of ordering a manual, you should check the AD on the left, as well as the quantity pricing listed above!   J.F.W.

WEST PENN 99'ERS

% JOHN F. WILLFORTH
R.D. #1   BOX 73A
JEANNETTE,   PA
                15644

ISSUE NUMBER 4
APRIL-1989
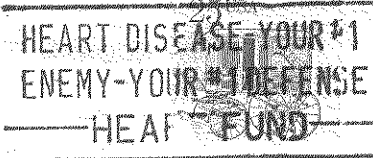
MICKEY SCHMITT   DUES PAID!
196 BROADWAY AVE.
LOWER BURRELL PA 15068

SEE CONTEST PAGE 2!