




THE FLUG TI ROUNDUP

THE OFFICIAL NEWSLETTER OF
THE FOREST LANE T.I.
USERS GROUP - DALLAS, TX

USERS OF THE TI 99 AND
COMPATIBLES

AUGUST 1987
VOLUME 1, ISSUE 11
EDITOR: Richard A. Fleetwood

OFFICERS FOR 1987:
 PRESIDENT Richard Fleetwood
 VICE PRESIDENT Oscar Smith
 SECRETARY Keith Joyner
 TREASURER Ron Kuhlman
 CORRESPONDENCE SECT. Wilson Taylor
 MEETING PROGRAM CHAIRMAN James Carson

Next meeting of the Forest Lane Users Group is:

SEPTEMBER 6TH, 1987-- PROGRAM : Auction of TI Hardware and software

IN THIS ISSUE:

- Upcoming AUGUST meeting information - By =RAF=
- PRESIDENTIAL ROUNDUP - A message from your chief executive officer - By =RAF=
- MINUTES OF JUNE MEETING - - BY Keith Joyner
- DALLAS TI FAIRE - Latest news and more volunteers- By Richard Fleetwood
- A "C" Voyage - Part three - By Marc Jensen
- Assembly Lines-Column 11 - More on ASSY Language cataloger - By Richard Denamer
- TI 994A PRODUCTS - latest news and rumors - By Richard Fleetwood
- TE11 AND SPEECH - Some TERNIFIC hints - By Barbara Berg
- 99FORTRAN - A review of a new language -
- LEGENDS - Some History behind the 9974a world - By Richard Fleetwood
- BOOKS ON ASSEMBLY - A bibliography and review - Richard Rosen
- FCC REGULATION - Use Your modem often? READ THIS!!
- TIWRITER VALUE FILES - Some hints and techniques -
- PUTTING YOUR XD CARTRIDGE IN YOUR CONSOLE - John Willforth

ASSORTED OTHER GOODIES....

AUGUST MEETING INFORMATION

By Richard A. Fleetwood

Below are the planned activities for the AUGUST 2,1987 meeting of the FOREST LANE USERS GROUP.

- 2:00 - BUSINESS MEETING
- Officers reports
- committee reports-TI99S, NEWSLETTER
- Library report
- TI99S -NEW NUMBER
- TI FAIRE news
- WASS BUYS-hardware and software
- ANY NEW BUSINESS?

- 2:45 - END OF BUSINESS MEETING
- 2:45 - 3:15 -- QUESTION AND ANSWER SESSION-ANY AND ALL ARE INVITED
- 3:15 - 3:25 -- COFFEE BREAK- 10 MINUTES ONLY
- 3:25 - 3:45 -- AUCTION OF HARDWARE AND SOFTWARE FOR FUNDRAISING FOR TI FAIRE
- 3:45 - 4:45 -- TI 99/4A CONSOLE CLEANING
 - TAKING THE CONSOLE APART
 - CLEANING OR REPLACING THE GRON PORT
 - MAJOR COMPONENTS OF THE TI99/4A
 - REPAIRS AND TROUBLESHOOTING PROBLEMS
- 4:45 - 5:15 -- SWAP SHOP AND NEWSLETTER LIBRARY BROWSING

Please help this and all future meetings run smoothly by asking questions at convenient times and not interrupting presentations.

PRESIDENTIAL ROUNDUP

By - Richard A. Fleetwood

Hi there, sports fans! Glad to see that most of you have made it thru most of summer now, and I hope you guys are ready to get into some heavy computer usage in the coming months. The upcoming meeting is going to offer something for everyone: great prices on items in our auction, and how to disassemble, clean, and maybe even repair your 99/4a consoles.

Last meeting we covered telecommunications, or using your computer and telephone line to call other computers. We discussed software to use, hardware that is required to get online, and different systems that are available for calling up whenever you feel like telecommunicating. We had a little bit of fun trying to get my 1200 baud Avatex modem to work, not knowing that the phone line we're using wasn't working, and we finally after about 10 minutes of confusion, moving of desks and chairs, and people tripping over the AS232 card, got the system online and working for everybody to see. To the people who stayed around, we were able to answer a lot of questions and show off some things that many users don't get to see everyday.

In the last newsletter, I had a copy of the questionnaire that I asked EVERYBODY to fill out, regardless of if you had done it months ago. As of this writing, ABSOLUTELY NO ONE has returned a single filled out sheet back to me.

COME ON PEOPLE.....WE WOULDN'T BE ASKING FOR THIS STUFF IF WE DIDN'T NEED IT!

PLEASE FILL OUT THE ENCLOSED QUESTIONNAIRE AND RETURN IT PROMPTO.

We have several members in the club who we do not have phone numbers to, and we HAVE to have this information from you to better serve you. PLEASE take the next ten minutes and fill it out before you do anything else. It is in the back of this newsletter, inside the back cover. Please be complete, and give us some ideas of what YOU want to get from the club, and what you would like to see.

If you'll notice, we have two events planned in the next five weeks to raise money to hold the TI Faire. At this meeting, and the whole september meeting, we are going to hold an auction and sell donated items from as many members as we can get. All funds raised will go towards supporting the faire. We need YOU to reach into the bottom drawers of your computer desks and find all those old cartridges, Acoustic modems, cables, consoles, peripheral expansion devices, extra memory cards, old disk drives, boxes of labels or paper, used disks, ribbons, flip files, or ANYTHING you might have to spare. We will start the bidding on all items at VERY low prices so that everyone will have a chance to get some fantastic buys on things they can use. If you can donate anything, please call me at 328-9257, and I will gladly come and pick it up and place it in the holding pen for september.

Also the FLUG is planning a Spaghetti dinner for the date of Sunday, August 16th, at TI Industries, on the southbound side of Interstate 35, just north of I-635. Take the Valwood exit from northbound 35, and make a u-turn to the left to come under the highway and come down the southbound service road. Look for

TD industries on the right about half a mile south of Watwood. Its hard to miss.....its the only building with a FLAGPOLE in the front of it. We plan to start around 2 pm. We will have a few systems there, and if Ron Schwab is ready, we will have the FLUG library there, and you are also invited to bring you software down to swap with us. We will have blank disks for sale, as well as some special utility disks with the latest and greatest from the the freeware world. The utility disks will be sold for \$4 each, each containing multiple programs. The dinner will \$5 a plate, and \$2 for kids under 10, with the whole family invited out to visit and have an afternoon of fun. Please contact me IMMEDIATELY to get your tickets to the dinner, so we will know how many to plan for.

As many of you should know by now, the FLUG TIBBS has been moved to its new (old?) home. The new number is (214)328-4880. It is still available 24 hours a day, and is almost ready for the completely new, latest version of the TIBBS program I have been working on. This new version will feature new additions such RLE pictures online, this complete newsletter in its own special menu, catalogs from my business as well as a few others, and HARDWARE projects online. Its really going to be a one of a kind system by the time we're thru with it. Also, with the help of the co-sysops mentioned in the minutes of the last meeting, updates will be done with much more regularity. Please pass this number around to EVERYONE. The TIBBS hasn't been very busy as of late, due to the notice not being out until this newsletter.

As for the latest on the TI Faire, please read the column inside for the latest info.

Hope to see all of you at the next meeting!!

=raf=

MEETING MINUTES

By - Keith Joyner

Meetings: June 7th, 1987

Quorum: 10
Present: 11

The idea of a member call up was presented so that everyone would know about the meeting time.

\$ 515.02 was transferred from the TI Lewisville Users Group into the FLUG treasury, bringing the account up to \$1179.02

The club SysOp is resigning. Rick Morgan had other projects and can no longer operate the system. Richard Fleetwood, club president, offered to take the system back over to keep it going, with a committee being set up to divide up the duties so that one person does not have to handle the whole load.

Committee members are: Richard Fleetwood
Annie Fleetwood
Richard Danzner
Marc Jenson
Roy Willis

Terry Morgan, the newsletter mailout chief, resigned her position, and the post was filled by Annie Fleetwood.

There was a proposal to poll members as to the preferred meeting time and date. Motion to have the August meeting at 2:00 pm Sunday. Motion passed.

Several sites were proposed for the TI Faire: Holiday Inn in Richardson, Richland Community College, ad some possible locations in the Richardson Independent School District.

Motion to replace the disk drives on the TIBBS with half power drives. Motion passed. The drives that come out are to be sold to the highest bidder at August or September meeting.

Motion to have the TIBBS moved back to Richards house. Motion passed.

A call was made for systems to use in the TI Faire. 8 systems were offered. Aproximate need may be 20 to 30 full systems, and 5 to 10 console/tv systems.

TI FAIRE NEWS

By - Richard A. Fleetwood

Alls going well except for a couple of minor details. One problem that we have to take a close look at is that the NET99ers user group, our sister group in Fort Worth, is having financial problems, and doesn't really have ay way to help us finance the up front money we need for the Faire. I have talked to a couple of other Texas Users groups, and so far the Brazos Valley TI Users group has pledged to help us with a \$50 donation to the cause. The other users groups are taking it under consideration, and will let me know something within the next week or so. I think that the club should vote on whether or not to handle the financial aspects ourselves and therefore gain the most, since the other clubs aren't sure yet what to expect. I have been talking to the people at the Holiday Inn on Central and Campbell for the past couple of weeks, and so far so good. I had also looked into using the TI SC cafeteria that we use to hold our meetings and spent a week contacting the proper individuals within TI, only to find out that TI policy did not allow for VENDORS of any kind to come on to TI property and sell goods and services to either TI employees or family and friends of same. Since that is one of the main purposes of a faire, that got rid of the idea of using that site.

I had hoped to have some numbers as to costs we would be looking at for deposits with Holiday Inn, for reservations and food, but unfortunately, I am not seeing them until this friday, right before the meeting. I will bring all the information to the meeting.

A few weeks ago, Roy Willis, Murty McCain, and myself went to the NET99ers meeting to talk with them about deciding on the place to hold the faire, and we actually chose to use the Holiday Inn, due to their generous offers that have already been mentioned, which is that they will give us free use of the meeting rooms in exchange for guaranteeing 30 reservations and having the dinner there saturday night. These two items will be paid for by the attendees who come to them, and all FLUG and supporting groups have to do is cover the minimum numbers if we have some no shows. I think we will have no problem making the minimums, and expect to see more than twice the above number of reservations mentioned.

We are signing up volunteers for both manning booths, providing systems for demos, and providing security for the faire, as well as directing traffic. These sign up sheets will be mailed out to all members we have on our mailing lists, and duplication is preferred to get the maximum number of volunteers. If you can help, either directly or indirectly, please call me at 214-328-9257. The mailout mentioned about will be an information packet that will be sent to all known users groups, vendors of hardware and software, and members of FLUG. The packet will contain reservation forms, requirements of systems, vendor reservations, information and timetables, and a few other small details. It should follow this newsletter in about two weeks. Please be sure to help spread the word.

For the latest information, please contact me.

=raf=

ASSEMBLY LINES

(Column 11) 07/87

By - Richard Danzner

Last month we left off after defining the file we must open as INPUT, RELATIVE, INTERNAL, FIXED 38. I said that this month we would write the PAB for file, and so we will. All of the information required can be found in Sec. 18 (FILE MANAGEMENT) of your EDITOR/ASSEMBLER manual. Once again, initialize a new disk. Copy onto it ASSM1, ASSM2 and EDIT1 from the Editor/Assembler Part A disk. This will be the last disk I have you create and will eventually contain the assembly language disk cataloger. Ok, create the following SOURCE file.

An explanation will follow.

```

*
* DATA FOR PAB SETUP (INPUT,REL,INT,FIXED 3B)
*
PABBUF EQU >1000
PAB EQU >0F80
PNTR EQU >8356
PDATA BYTE >00 I/O OP CODE
        BYTE >00 FILE SPEED
        DATA PABBUF VOP DATA BUFFER ADDRESS
        BYTE >26 LOGICAL RECORD LENGTH
COUNT BYTE >26 CHARACTER COUNT
REC DATA >0000 RECORD NUMBER
        BYTE >00 SCREEN OFFSET (CASSETTE)
LEN BYTE >05 DEVICE NAME LENGTH
FILE TEXT 'DSK1.'
READ BYTE >02
CLOSE BYTE >01
*
```

Save this to disk as "DSK1.PAB"

EXPLANATION:

The first 3 lines are comments describing the function of this program segment. I have not mentioned this previously, but any line with a * in column 1 is a comment statement. These lines will show up in the LISTING file, but have no effect what so ever on the object code that is created.

The next 3 lines are EQUATES to memory locations in VOP ram. PABBUF is equated to address >1000 in VOP ram. It will be the starting address of a buffer area in VOP ram through which all data will be passed. PAB is equated to address >0F80 in VOP ram. It will be the starting address for the PAB data. (All the data from this program segment will be moved to there.)

PNTR is equated to address >8356 in VOP ram. This address is a POINTER, and must contain the LENGTH of the file name to be accessed when a DSR LINK is performed. NOTE: Although I will show you HOW to use this in future columns, I probably won't explain WHY. So, take the time now to READ THE BOOK so you won't be lost later.

The next line PDATA, is the beginning of the actual PAB data. Refer to page 293 of the E/A manual (PERIPHERAL ACCESS BLOCK DEFINITION) for a detailed explanation. I will not explain all the code here. I leave that to you and page 293. I will however explain the first 2 bytes and the last 2 bytes.

The first byte is easy. It is the I/O op code (00) to open a file. All op codes are defined and can be found on pages 295-298.

The second byte must be defined a few bits at a time. Bits 0,1 and 2 are used for error reporting by the DSR link utility. So, just set them to 0. That gives us 000.

Bit 3 must be 0 in our case to specify a FIXED length record. We now have 0000.

Bit 4 must be 1 to specify INTERNAL. We now have 00001.

Bits 5 6 must be 10 to specify INPUT. We now have 0000110.

Bit 7 must be 1 to specify RELATIVE. We now have 00001101. Converting this to HEX, our byte is defined as >09.

The last 2 bytes of the program segment, define the I/O op code for READ and CLOSE respectively.

Good news for some, bad for others. I was planning to stop here this month. But, I have changed my mind for a couple of reasons. First, I was planning to take you all through the entire process I went through to develop this program. However, a quick look back showed me that I went through 7 iterations of the source code plus several subroutines. Also, I have been working on presenting this since May. At this rate it would be next May before I finish. Too long for both all of us I'm sure! Second, I just received the June newsletter and found there won't be one in July. Therefore, this column will be much longer than usual and, I will need to take some shortcuts in the

future. So, here's a listing of the first iteration of the main program:

```

*
* DISK CATALOGER - VERSION 1
* (DISPLAYS DATA ON DISK ONE RECORD AT A TIME)
*
DEF START
REF VBNB,VSNB,KSCAN
REF VBNR,VSNR,DSRLNK
STATUS EQU >837C
KEYADR EQU >8374
KEYVAL EQU >8375
START MOV R11,05A0RTH SAVE RETURN ADDRESS TO E/A MODULE
      CLR 0REC
      LI R0,PAB
      LI R1,PDATA
      LI R2,16
      BLMP 04VBN
      LI R6,PAB+9 LOAD PAB DATA INTO VOP RAM
      MOV R6,0PNTR
      BLMP 00SRLNK OPEN THE FILE
      DATA 8
REREAD MOV 0READ,R0
      LI R0,PAB
      BLMP 04VBN CHANGE I/O CODE TO READ
AGAIN LI R1,>2020
      LI R2,57
      LI R3,BUFF1
ASH MOV R1,R3+ CLEAR BUFFERS
      DEC R2
      JNE ASH
      LI R0,PAB+6 POINT R0 TO RECORD # IN PAB
      LI R1,REC POINT R1 TO RECORD # IN PAB DATA
      LI R2,2
      BLMP 04VBN WRITE THE RECORD # TO THE PAB
      LI R6,PAB+9
      MOV R6,0PNTR
      BLMP 00SRLNK READ ONE RECORD
      DATA 8
      IMC 0REC REC. # IN PAB DATA +1 FOR NEXT PASS
      LI R0,PAB+5 POINT R0 TO READ CHAR COUNT
      CLR R1
      BLMP 04VBR COUNT TO MSB OF R1
      MOV R1,0COUNT
      SHRP R1
      MOV R1,R2 BYTE COUNT TO R2
      LI R0,PABBUF
      LI R1,BUFF1
      BLMP 04VBR MOVE CHARACTERS TO BUFFER
      LI R0,1
      LI R1,BUFF1
      LI R2,11
      BLMP 04VBN WRITE 1st 11 BYTES TO SCREEN
      LI R0,53
      LI R1,BUFF1+11
      LI R2,9
      BLMP 04VBN WRITE NEXT 9 BYTES TO SCREEN
      LI R0,65
      LI R1,BUFF1+20
      LI R2,9
      BLMP 04VBN WRITE NEXT 9 BYTES TO SCREEN
      LI R0,97
      LI R1,BUFF1+29
      LI R2,9
      BLMP 04VBN WRITE LAST 9 BYTES TO SCREEN
      LI R7,19
      LI R1,BUFF2
      LI R4,BUFF1
      HTOA MOV #04,R3
      LI R2,4
      LI R5,>0FF
      HTOA MOV R3,R6
      SRC R3,12
      SZC R5,R6
      SRC R6,4
      CI R6,>0A00
```

```

JLT HTDA2
AI R6,>3700
JMP HTDA3
HTDA2 AI R6,>3000
HTDA3 MOVB R6,R1+
DEC R2
JNE HTDA1
DEC R7
JNE HTDA
LI R0,161
LI R1,BUFF2
LI R2,22
BLMP @R0WB WRITE 1st 22 BYTES TO SCREEN
LI R0,193
LI R1,BUFF2+22
LI R2,18
BLMP @R0WB WRITE NEXT 18 BYTES TO SCREEN
LI R0,225
LI R1,BUFF2+40
LI R2,18
BLMP @R0WB WRITE NEXT 18 BYTES TO SCREEN
LI R0,257
LI R1,BUFF2+58
LI R2,18
BLMP @R0WB WRITE LAST 18 BYTES TO SCREEN
CLR @KEYADR
SCAN CLR @STATUS
BLMP @SCAN
CB @ANYKEY,@STATUS
JNE SCAN
CLR R1
MOV @KEYVAL,R1 WAS A "Q" INPUT ?
CHECK CI R1,>5100 WAS A "Q" INPUT ?
JEQ CF
CF MOVB @CLOSE,R1
LI R0,PAB
BLMP @VSHB CHANGE I/O CODE TO CLOSE
LI R6,PAB+9
MOV R6,@PTR
BLMP @BSLBC CLOSE FILE
DATA 8
MOV @SAWRTH,R11
LIMI 2
CLR @STATUS
RT RETURN TO E/A MODULE
#
# COPY OTHER SOURCE CODE
#
# COPY "DSK1.PAB"
#
# DATA AND RAM BUFFER AREA
#
SAWRTH DATA >0000
ANYKEY DATA >2000
BUFF1 BSS 38
BUFF2 BSS 76
END

```

Save this source code as DSK1.SOR to the same disk you save PAB. NOTE the "COPY" instruction (9th line from bottom) specifies DSK1.PAB. If your system has multiple drives, be sure to place the source code in drive #1 or, change the copy instruction to reflect the drive you assemble from. Ok, assemble DSK1.SOR. Use the LOAD-RUN option to load it. The program name is START.

IMPORTANT !!!!: REMEMBER ! You are now dealing with direct disk access! If you make an error in your source code but it somehow manages to assemble correctly anyway, you could WIPE OUT YOUR DISK !! Always test new code on a WRITE PROTECTED disk or one that you don't mind losing. Don't test your code on the disk you use to create the code, or you may lose all your hard work !

Ok, so run the program. Here's what SHOULD happen:

- 1) the program reads the first record. The data EXACTLY as read will be displayed to the first 4 lines of the screen. To see the exact format, check the source code. This data is then converted from hexadecimal to

- ascii and re-displayed on the next 4 lines. Again, check the source code for the format.
- 2) now press any key other than Q and the process will repeat for the next record. You should be able to recognize the disk name and file names at the top of the screen. Study of the last 4 lines will reveal that the file names are in internal format and the numeric data is in RADIX-100 format. (RADIX-100 is covered in the E/A manual.)
- 3) when no more file names appear, press Q to return to the E/A module. Since the program is already loaded, you may repeat the process simply by using RUN option with the program name START to look at any disk you wish.

That's it this month. Good luck, and have fun.

A "C" VOYAGE, Part 3

By - Marc S. Jensen

This is the third of my articles on C, and I hope you are ready, for this time we're going to learn about such exciting things as keyboard input, loops, and decision-making statements.

If you have read the previous installments of this series, you should at this time know how to use the following commands: puts, gets, putchar, getchar, int, char, #include, #asm, #endasm, and printf. Knowledge of these commands (or most of them, at least) is necessary to understand the programs presented in this article.

The last program in last issue's article was one to take two numbers, manipulate them in several ways, and then print the results. While this is more exciting than finding the 17th character in a string, it lacks one thing: KEYBOARD INPUT! It is possible (you guessed it!) to input the two numbers from the keyboard instead of just giving them a fixed value at the start of the program. Try the following:

```

#include DSK1.STUID
extern PRINTF(), SCANF();

main ()
{
    int number1,number2,result;

    printf("\nPlease enter a number: ");
    scanf("%d",&number1);
    printf("\nPlease enter another number: ");
    scanf("%d",&number2);
    puts("\n\n");

    result=number1+number2;
    printf("%d plus %d equals %d.\n",number1,number2,result);
    result=number1-number2;
    printf("%d minus %d equals %d.\n",number1,number2,result);
    result=number1*number2;
    printf("%d times %d equals %d.\n",number1,number2,result);
    result=number1/number2;
    printf("%d divided by %d equals %d.\n\n",number1,number2,result);
}

```

You should note two new things in this program. First of all, we have replaced the lines "#asm ... #endasm" with the statement extern command. This does the exact same thing, but it's more like standard C. "extern," of course, stands for "external," which makes sense since it allows you to use external additions to the language (like printf, for instance). Secondly, and more important, is the use of the scanf command. This is the input equivalent of printf, and uses the same specifiers ("%d" and "%c", for example). Scanf basically works exactly opposite printf. The variable(s) after the comma are where the input data is placed. Thus, scanf("%d",&var1); means "Scan the keyboard and place the first decimal number entered into the location occupied by var1." Note that an ampersand "&" was placed in front of the variable. The ampersand literally

means "the address of." Whatever you type in will be stored in the memory location used by the variable. Also, it is possible, like in the printf command, to use several variables in one statement. Thus, the statement

```
scanf("%d%d",&number1,&number2);
```

would accomplish the same as

```
scanf("%d",&number1);
scanf("%d",&number2);
```

The following, however, would NOT be a legal statement in C:

```
scanf("Please enter a number: %d", &number);
```

A statement such as this would have to be broken into two parts, one of which would printf the prompt, while the other would scanf the number.

And now... to something COMPLETELY different! We're now going to make decisions using the C language. This is done using the "if" command, which works in almost the same way as its BASIC and Pascal counterparts. The basic format for "if" is

```
if (condition) statement;
```

For example,

```
if (x==2) printf("This is printed because x is 2");
```

This would be the same as the BASIC statement IF X=2 THEN PRINT "THIS IS..." Now, if you want to execute a GROUP of statements if a condition is true, you must use the braces once again. (Those squiggly lines are used quite often in C, as you will see) For example,

```
if (x==2)
{
printf("This is done because x is 2");
printf("So is this");
printf("And this");
}
printf("This is outside the braces and will ALWAYS be printed!");
```

The "if" statement can be made even more powerful with the addition of "else." This, too, is much the same as in BASIC, as far as logic is concerned. As an example,

```
if (x==2)
{
printf("This is done because x is 2");
printf("So is this");
}
else
{
printf("X is not 2");
printf("That's why this is printed...");
}
}
```

To illustrate the if...else command, try this small program:

```
#include DSX1.STDIO;
extern SCANF();
main()
{
int num;
puts("This program will print one of two");
puts("different phrases, depending upon");
puts("keyboard input. Press 1 or 2.");
scanf("%d",&num);
if (num==1) puts("Programming is fun!");
else puts("Programming in C is more fun!");
}
```

If you type a 1, the program prints "Programming is fun!". If you type in

anything but a 1, it prints "Programming in C is more fun!"

Those of you who have read carefully will have noticed that TWO equal signs have been used in all these examples. That's just the way it is in the C language. "=" is an assignment operator; it is used to make something equal to something else. "==" is used to TEST if two variables/numbers are equal. Don't ask me why it's like that, it just is. Accept it and learn to live with it!

Having covered input, output, and conditionals, guess what the next important step is going to be... That's right, we need LOOPS! The C language has several kinds of loops. There's the FOR loop, the WHILE loop, and the DO...WHILE loop. Some versions of C also support a loop structure using UNTIL, but we won't talk about that until someone makes it for the 99/4A. For now, let's figure out how to use the simplest of the bunch, the WHILE loop. The general format is

```
while (condition) statement;
```

This will continually execute the statement(s) until the condition is false. Again, you will need the braces to execute groups of statements instead of just a single command. For example,

```
x=1;
while (x<=10)
{
printf("x is %d",x);
x++;
}
```

will count from 1 to 10. (Remember, x++ means x=x+1).

How's that for speed? In just 3 sittings, you've (hopefully) learned the most important aspects of C: input, output, conditionals, and loops. Next week, we'll take a look at creating other functions (besides MAIN) and how to use them. Also, we'll make a game program using everything that has been covered so far...

In case you only have one disk drive (like me) and are finding yourself having to switch disks frequently, I suggest making a disk with the following on it: The c99 compiler, E/A editor, Assembler, PRINTF, SCANF, CSUP, and STDIO. This should leave you with plenty of room for your programs and some additional support files. For ease of keeping track of your files, the following method of naming programs is commonly used: PROG/C is the C source code (what you type in). PRG/S is the code produced by the compiler (Assembly source), and PRG/O is the object code produced by the assembler. Hope this helps. If not, I guess that's just too bad... Also, if any one of you have problems/questions with C, feel free to call me or leave mail on the BBS. If I can't help, I might at least be able to direct you to someone more knowledgeable...

I AM COMPUTER, HEAR ME ROAR!
By Barb Berg

If you have the Speech Synthesizer and a speech-accessible module, such as Extended BASIC, Terminal Emulator II, or a game such as Parsec or Alpinar, you have probably heard your computer talk to you at some time or another. Writing a program that includes speech is really not that hard, as you may have discovered. The TE II module allows text-to-speech, which makes it quite easy to have words or phrases "spoken" by the computer, and XBASIC allows allophones and uses the resident vocabulary of 373 words and phrases in the speech synthesizer.

But if you HAVE used the TELL for text-to-speech, you must have discovered that some words are pronounced incorrectly. Some words in our English language do not sound like they are spelled. Can you imagine the trouble a foreigner must have, trying to learn our language with words like tough and through? Both have the "ough" at the end of the word, but definitely have different sounds.

This is similar to what happens with the speech synthesizer. While a lot of the common sounds (and many of the uncommon ones) are translated correctly, a few words manage to sound like they belong in some other language. I noticed this most when I wrote a simple spelling program for my kids to practice their weekly spelling words on. Where they go to school, they have to learn how to spell 75 words every week. My 6th grader just brought home a list with words I'd never even heard of, and most of the rest were words I didn't know until high school!

They have learned through the years that it helps to sound out a difficult

word in order to pronounce it properly (or at least closely!). In the spelling program, I made allowances for words that didn't sound like they were spelled. I used two different variable arrays, one for the correct spelling of the word, and one for the correct pronunciation. If the pronunciation of the word was all right when entered as correctly spelled, it automatically went into the pronouncing array. If not, one could enter the word phonetically, or the way the word sounded. This spelling would go into the pronunciation array only.

However, if you write a program which uses a lot of speech or uses words like these, you may decide that even those words that come out sounding ok are still not quite what you had in mind for that particular program. The TELL module has a very nice system for that purpose, in that the text is translated into allophones according to previously programmed speech rules, and these allophone strings are then changed into what the manual calls Linear Predictive Coding (LPC) strings. It is the LPC string that the speech synthesizer "speaks".

Now, I don't know about anyone else, but the default "voice" resident in the synthesizer sounds an awful lot like Marvin the Paranoid Android from the "Hitchhiker's Guide to the Galaxy" series shown on IPT a few years ago. This robot was a terribly depressed individual, and his voice sounded like he was ready to "pull the plug" on himself any minute. The speech rules in the TELL module include sentence intonations, but like Marvin, they make the voice sound like your computer just died and it's in mourning. You can change the pitch and slope, but even then it can sound like either a soprano Marvin (you can do your own jokes here, folks) or like him with a bass voice.

So how can you make the voice more interesting? Well, you can include inflection symbols, which tell the computer that you want a stress on this word or that syllable, but that doesn't always help as much as I'd like. Or, you can vary the pitch and slope slightly within the sentence or phrase itself. I don't recommend the latter, as it makes the speech sound very choppy, like Mel Tillis when he tries to say something and can't get it out right away.

We have a program from the UB library that has a little robot character singing "Old MacDonald". In this program, the pitch and slope are altered according to the notes that the words are supposed to be sung on, and if you've seen and heard it you probably have noticed how choppy it sounds. Good effort, but frustrating to listen to, especially if the kids decide they want to sing along.

Well, where does that leave you? It doesn't sound like the TELL and speech can be programmed to sound very realistically, does it? Ah, but wait a minute! There are also allophones. But what are they, and how do you use them?

Allophones are simply the individual sounds of speech. Remember when you had to learn "long" and "short" vowel sounds? Well, they don't call them that anymore, I guess, but at any rate, each letter in the alphabet has one or more sounds, depending on what letters they are used in combination with.

For example, notice the difference in the combination "in" in the following words: in, insane, think, thing. In the word "insane", the i is spoken for a shorter length of time than the i in "in". The n is pronounced differently in the last two words than in the first two. We speak these words without giving thought to the length of time we hold the i, and "ing" and "ink" are second nature. And we may use a different inflection in our voice to set the mood of what we say. But the speech synthesizer can't do all of that by itself. If you want to say IHSane instead of inSANE, you just do it! But you have to tell the speech synthesizer where to put the stress point in the word so it can say it the way you want.

But it looks so time consuming to program a sentence with allophones. Right? The following program is on page 40 of the TELL manual:

```
100 OPEN #1:"SPEECH",OUTPUT
110 OPEN #2:"ALPHON",INTERNAL
120 INPUT "PHRASE ":A$
130 IF A$="" THEN 120
140 PRINT #1:A$
150 INPUT #2:B$
160 Z=LEN(B$)
170 FOR R=4 TO Z
180 PRINT ASC(SEQ$(B$,R,1))
190 NEXT R
200 GOTO 120
```

What this program does is a great time saver. File #1 opens a channel to the speech file, which speaks the words you enter for text-to-speech. File #2 opens the alphon file, which breaks the words down into allophones. When you run the program, you are asked for a phrase. Whatever you enter here is first spoken by the speech file and then input into the alphon file. The standard inflections and stresses make up the first three CHR\$(of the phrase. These include the sentence break code 250 and the number of secondary stress points before and after the primary stress point. Since all we want are the

allophones for the words in the phrase, we start at the fourth CHR\$ in the phrase and the computer displays the allophones for the words used. Now you can see what allophones are used in each word by comparing the screen display with the chart on page 41.

You can then take your word or phrase as broken down and do a variety of things to it. Add pitch and slope, change the vowel sound or the length of time it is held, alter the consonant sounds, whatever you want to do. When you make up the speech string from allophones to add to your program, the allophone numbers become CHR\$(numbers. For example, take the word "the". The allophones used by the TELL for "the" are 95 and 18. CHR\$(95) is the hard "th" sound to be used at the beginning of a word and CHR\$(18) is a very short "uh" sound. Change CHR\$(18) to CHR\$(69) to have the "uh" sound held longer and for a more distinct sound. Now you can put "the" into a string variable:

```
A$=CHR$(95)CHR$(69)
```

Want to add pitch and slope? CHR\$(251) is the slope indicator and CHR\$(252) sets up the pitch. The default values for pitch and slope are 43 and 128, respectively. We can add them to our string like this:

```
100 THE$=CHR$(95)CHR$(69)
110 PITCH=CHR$(252)CHR$(43)
120 SLOPE=CHR$(251)CHR$(128)
130 A$=PITCH$SLOPE$THE$
```

If you don't understand the difference changing slope makes, or the formulae on pages 34-35 of the manual, then you should experiment with it a little. I have found that the lower the number the smoother the slope. If the slope is a high number, you get a rougher sounding voice. That isn't the right word to express it, but you really have to hear it to understand.

As for using the formulae to determine slope, first use the 32 X 102 of the pitch. If the pitch is 35, then 32X3=96, and that is what the manual says should be the best sounding. Now check this in the other two formulae: first, $yyy<[xx-1]X16$. With our values replacing the variables, the result is this: $96<[35-1]X16$ or $96<34X16$ or $96<544$. Definitely true. In the other, $yyy<[63-xx]X16$ becomes $96<[63-35]X16$, or $96<28X16$ or $96<448$. You can see, this gives you a lot of room to work with when you use a low pitch. (The lower the number, the higher the pitch.) Just keep the slope under 255. The format in text-to-speech format is: PRINT #1:"//xx yyy" where file #1 is opened to the speech file, xx=pitch and yyy=slope.

Need stress points in your phrase? Use CHR\$(253) for a primary stress in which the sound must rise and CHR\$(254) for one in which the sound lowers slightly. CHR\$(249) indicates a secondary stress point, which puts less emphasis on a syllable than the primary stress point. CHR\$(250) should be used at the beginning of a sentence if stress points are used as this tells the computer how many secondary stresses there are both before and after the primary stress point. Can't understand how? Start your string with CHR\$(250). Count the number of stress 2's come before your stress 1 (A). Now count the number of 2's after the 1 (B). Then use this form:

```
CHR$(250)CHR$(A)CHR$(B)
```

Continue to build your string from there. If the entire string is supposed to rise or fall, use either CHR\$(254) or CHR\$(253) after the CHR\$(250). Then add a CHR\$(A) where A = the number of vowels in the phrase. As an example, enter the following:

```
100 OPEN #1:"ALPHON",INTERNAL
110 A$=CHR$(250)CHR$(2)CHR$(1)CHR$(252)CHR$(46)CHR$(249)
120 C$=CHR$(117)CHR$(34)CHR$(249)CHR$(74)CHR$(67)CHR$(127)CHR$(254)
CHR$(86)CHR$(50)CHR$(249)CHR$(87)
130 C$=C$CHR$(82)CHR$(69)CHR$(127)CHR$(127)
140 B$=CHR$(250)CHR$(3)CHR$(0)CHR$(252)CHR$(46)CHR$(249)
150 D$=CHR$(126)CHR$(17)CHR$(5)CHR$(249)CHR$(50)CHR$(126)CHR$(
249)CHR$(85)CHR$(46)CHR$(127)
160 D$=D$CHR$(254)CHR$(73)CHR$(69)CHR$(98)
170 E$=B$A$C$
180 F$=B$A$D$
190 PRINT #1:E$A$F$
200 GOTO 110
```

Believe it or not, all this does is say "Hello, Barbara. How are you, love?" You can substitute the allophones for your name where the ones for mine are and change the pitch of the voice as desired. A\$ sets up the sentence break, number of secondary stress points and the pitch, then indicates that the phrase will start with a secondary stress point. C\$ is the first sentence. B\$ does

the same as A\$ but it does it for the second sentence, D\$. The main thing I want you to notice is how the number of secondary stresses are added after the CHR\$(250) in A\$ and B\$. CHR\$(249) indicates a secondary stress, and you will see that there are 2 of these before the primary stress in C\$ and 1 after it. In lines 120 and 160, CHR\$(254) is the primary stress point in each sentence and signifies a falling contour. CHR\$(252) in A\$ and B\$ is setting up for the pitch parameter, CHR\$(46). This is not the same as the allophone CHR\$(46) in line 150. All other CHR's are allophones, also.

Now, how does the computer know that the CHR\$(46) in A\$ and B\$ is not an allophone? Very simply, it is because of the CHR\$(252). When the ALPHON file sees a 252 coming through, it knows that the next allophone is going to be the value for the pitch. Similarly, when it sees a 250 coming, it knows that the next TWO CHR's are going to tell it how many secondary stresses to expect. If a 254 or a 255 is used as the first parameter after a 250, then the ALPHON file assumes no secondary stresses are coming, the primary stress will be on the first syllable, and the CHR\$ following the 254 or 255 will tell it how many vowels (or syllables) there are in the sentence. Here, 254 indicates the sentence will have a rising pitch and the 255 means the pitch will fall.

Completely confused? I was at first. It was a little difficult to understand how CHR\$(254) could be both a rising and a falling contour until I realized that, like the allophones, what they indicate depends on the CHR\$ they follow. Think about it a little; sooner or later it does sink in!

Now, if a CHR\$(255) is used within the phrase itself (not in the sentence break parameters), it indicates a temporary pitch change. The CHR\$ that follows it will tell the pitch for the next syllable ONLY.

If you are beginning to get the idea but are still having a little difficulty grasping what I have said here, it may help if you try out the little "programettes" on pages 38-40 in the TEII manual, like the one I included earlier in this article. That's what I had to do to understand better how allophones are used! I don't understand why there aren't more programs with speech, especially when you really CAN do quite a bit with them.

Just for the heck of it, and partly because there are a lot of music programs around that show how the SOUND subprogram can be used, I decided awhile back to see if it was possible to make the computer actually sing. And I wanted a smoother result than good ol' Old McDonald. So I sat at the computer for hours on end, playing with the allophones, finding pitches to correspond with the CALL SOUND tones, and experimenting with slopes until I understood them to my own satisfaction. (When it comes to computers, I'm insatiable!) My first result was a program I called BEATSING, in which the computer sang "I Saw Her Standing There" and displayed an exceptional graphic of Paul McCartney. (No brag, just fact!) The time it took to write this program led me to write a driver program called TI SINGS, which allows anyone to enter a song for the computer to sing. It also allows disk storage of completed songs. This article is also on the disk in TI-WRITER form, and with the completion of this article came the completion of the TI SINGS disk. It is now available through Triot Software, and if this article piques your interest, it can be had "for a song". (Ba-a-ad pun—sorry!)

I hope this article gives you a start with programming speech with the TEII. My only disappointment with the speech system is that in order to write a program that uses the speech capabilities of the TEII module, one is limited to BASIC. Maybe someone out there will begin to see the possibilities with the speech system and design an extended BASIC with the speech capabilities of the TEII. I would have loved to be able to use some of the X-BASIC features when I wrote the TI SINGS program, but had to settle for BASIC. And it would have been nice to be able to access the 32K memory for longer songs. How about it, hardware developers?

99 FORTRAN from LGMA Products

A Review of First Impressions

by Ralph Landru, HUG member

I recently bought the LGMA 99 FORTRAN package that is advertised in the new TEXEX catalog. So far I've studied the manual and compiled and linked the example programs that come with the package. It is well planned for the user. The manual is well written. It will be clear to anyone the least bit familiar with FORTRAN at any level. It is clearly meant for people who use the TI99 in XBASIC, but who want compiled versions of their programs. Assembly language programmers can also use internal TI99 subroutines and their own assembled code within the structure.

WHY FORTRAN?

FORTRAN has a conversational syntax like BASIC, and is therefore easier to use for me than A/L or C. In fact, the LGMA package is actually a combination of BASIC and FORTRAN II, being a subset of FORTRAN 77, rather than FORTRAN IV as advertised. I am familiar with (though not a trained programmer in) several forms of BASIC, FORTRAN II, and IV.

FORTRAN uses true subroutines, which I need in what I want to do with a computer. XBASIC uses true subroutines also.

FORTRAN is a compileable language. I want to be able to compile to machine language for speed. BASIC is compileable in some versions (for example IBM PC), but none has brought out a good compiler, using true subroutines, for the TI99. So, FORTRAN could let me have a more familiar language, using true subroutines, but compiled for operating speed.

THE LGMA 99 FORTRAN Package

LGMA Products, Box 210, RD4, Apple-Butter Hill Road, Coopersburg, PA, 18036, is a company unknown to me. Alan L. Beard signs letters for them. Their 99 FORTRAN package was advertised in the latest TEXEX catalog for \$49.95. The package comprises two disks of ver. 2.1.3, and an excellent manual.

One disk has the boot (in E/A, N/Y, BASIC, or T/W); the Full-screen Editor, Optimized Compiler, Linker, Debug, and example programs. The second disk has an excellent object module library with 78 functions and subroutines, including math functions (both single and double precision), and all the graphics and sound functions of TI BASIC. Included are CHAR, CHARPA, COLOR, DELAY, DELETE, DELSPRITE, FILES, GCHAR, HCHAR, JOYST, KEY, MAGN, MOTION, POSIT, SCREEN, SET32, SET40, SOUND, VCHAR, WAIT.

I find the manual to be VERY well written and organized. It explains things very simply for average programmers like me, but it also goes into detail for those excellent systems programmers who will want to use internal subroutines of the TI99 roms, or want to add their own assembled routines to the library. Of course, you can write FORTRAN functions and subroutines, compile them, and add them to the library. Whoever did the manual must be an expert programmer AND user.

Your system requires 32K, at least one SSSD disk drive, and E/A, T/W, XBASIC, or MW. Remember that this FORTRAN is a SUBSET of FORTRAN 77, with a few extra features. For example, it does not support the ENTRY statement of FORTRAN 77, but it does support the DOWNLINE statement from PASCAL—NOT FORTRAN 77. It is a subset in other ways, of course, being shoehorned into a small computer. Its program limit is 2 segments of 8K each. Integer constants take 2 bytes as do logical constants. Single-precision constants occupy 4 bytes, while Double-precision ones occupy 8 bytes. The author includes a section of the manual explaining various tricks of the system to save space.

IS THE PROGRAM WORTH THE MONEY?

If you are comparing the too cheap cost of the programs from Clint Pulley, and the FREE and from the heart contributions of Warran Agee, Ron Albright, and many others who gave and taught us our C99 language, then you will look at \$50 as a lot. However, because of the quality of work, the completeness, and comparison with the cost of other commercial programs, I find it reasonable.

I have not tried to program and run benchmarks against other programs, nor have I yet tested the optimizer by comparing routines like double-nested DO LOOPS compiled from source and written in assembler, but my elation in finding the system to be 77 instead of IV, the first programs I've compiled, the obvious effort of the author to make the system comparable to the XBASIC system we know with graphics and sound, and the excellent manual make me vote overwhelmingly YES, the program is more than I expected, and worth the money. In the little time I've had to exercise the program, I find only two things I hope can be improved in future. One is to have a scale on the screen to tell me where I am on the eighty-column line. The second is to allow the LINKER program to automatically scan through more than one library disk just as it automatically iterates to let you load more than one OBJECT file. Those are not big objections ... they could just be made more convenient.

DISCLAIMER

(NOTICE: The following article was written specifically for use of users of the DELPHI database system, and especially for a conference that was held on the subject matter. Permission is given to reproduce this article for other club newsletters, ONLY if the author's name and the source of that article

(DELPHI) is kept intact. All matter contained within was verified to the best of my knowledge, and use of any information within is done with the knowledge that the author may not be held responsible for any problems caused by use or knowledge of information contained within the article.)

LEGENDS...

TI 99/4A Family hardware and software that "never was"

By - Richard A. Fleetwood

President-Forest Lane TI Users Group
Dallas, Texas

There have been many stories about things that TI was working on for their home computer before they shut down the home computer division due to heavy financial losses. There were several items that TI pre-advertised before actual availability, and many others that never even made it to that point. This article is going to cover many of these hardware items, and even some software projects that were never officially released thru proper channels. The purpose of all this information is for your personal enlightenment, as well as some historical significance, since very little mention has ever been made in other reading material. All this information also comes from first hand knowledge, and actual use of the mentioned items--These projects are not figments of my or anyone elses imagination, they were and are real. Calling TI to verify this information will put you up against a blank wall. The people you will get in the phone weren't even around when some of this stuff was going on, and no one will admit that this stuff was ever made, much less left the premises.

So with that out of the way, let us start this feature by giving you a list of some of the things I have, have had, have used, or have seen. This short list is by no means complete, but it is full of things not seen by a vast majority of 99ers. If you know of something TI worked on but do not find it listed, please contact me so I can add to my files for future reference. I have a decent sized library of tech info and can probably return the favor.

SOME OF TI'S LEGENDS

- T.I. 99/4 The "Original"
- T.I. 99/4A The Computer most of us have come to love
- T.I. 99/4B The "missing" link between the 99/4A and the 99/8
- T.I. 99/8 The ULTIMATE TI home computer
- T.I. 99/2 TI's entry into the "Timex" arena
- T.I. CC40 TI's entry into the "portable" arena

Peripherals!

- THE GRID BOX
- DS/DD CONTROLLER
- HARDDRIVE CONTROLLER
- IEEE-488 INTERFACE CARD
- VIDEO CONTROLLER CARD
- 128K MEMORY SUPERRAM CARD
- 374K MEMORY ULTRARAM CARD
- FORTI 4 CHANNEL MUSIC CARD
- 4 PORT RS232 CARD
- ARMADILLO INTERFACE CARD

HEXBUS peripherals

- MODEM
- RS232
- PRINTER/PLOTTER
- PRINTER 80
- 5 1/4 " DISK DRIVE
- SMART MODEM
- VIDEO CONTROLLER
- HEXBUS INTERFACE

The 99/4 is what started everything off for TI in the home computer market. Originally designed in the late 1970's, it finally hit the market in 1980, and for over \$1100 you got a 16K, 16 bit computer with builtin BASIC, and also a 19" color monitor. The worst thing about the 99/4 was the keyboard. It used the

same chiclet style keys found on TI's entire line of calculators. It also was very limited when it came to expanding--in fact at first, there was no way to expand at all! That came months later.

TI listened to all the feedback from customers and finally did the right thing. They redesigned the 99/4, adding a REAL keyboard (although downsized) making touchtyping MUCH easier. They rewrote some portions of the operating system to make some things easier. They changed the video processor to the 9918A (hence the 'A' in '99/4A') and in the process added gobs of power to the overall system. They also around this time came out with the PERIPHERAL EXPANSION BOX, and several cards to make use of it. In between the 99/4 and the 99/4A, TI developed the choo-choo style method of upgrading--making the 32k memory, the RS232, the Disk Controller, and the PCODE unit in standalone units with trim that matched the color and profile of the black and silver consoles. If you had a fully expanded system, your computer desk was at least four feet wide, with all your peripherals extending from the right side of the console to the far end of your desk. When it came to software, the 99/4A is what really unleashed the real power and graphics of the machine, and many new things began to flow from Lubbock, including some of the kinds of programs that used to require the "other" brands of computers in order to run them. TI late in 1982 and into 1983 started letting 3rd party companies have access to the inner workings of the 99/4A so they could produce (under license) their own software. AtariSoft, Milton Bradley, Parker Brothers, and many others finally got their arcade programs on the market. Unfortunately, by the time things were rolling for them, TI was already headed downhill. More on software later...

Also about the time the good things were happening with software, TI was looking towards the future and other possibilities of expansion. The engineers were playing with ways to make things simpler, yet more complex. They threw together a half dozen "all new" 99/4X computers, based upon some market research and other projects. These new computers were little more than proposals and brainstorming sessions, as well as prototypes for the new wave of home computers that would renew the infamous computer wars of the early 1980's. These few consoles were known as the 99/4B's. They had a brand new memory mapper and used the big brother of the TI 99/4A's microprocessor. They were equipped with the new 9995 u-processor that enabled much more memory to be accessed, and a much better throughput of machine code. These new chips were much more efficient than the 9900, and soon proved very capable and powerful in a small home based system. The 99/4B was equipped with 32k of memory, a fast-system clock, and a semi-new keyboard. All six of these were pretty much handbuilt, and weren't meant to be marketed. They were just the testing ground for the 99/4A's "big brother".

The 99/8 was TI's final attempt at making it to the top of the home computer market. At the time it was going to be released, it would have been more powerful than any other home computer in its price range. It came equipped with 64k of memory, upgradeable to 2 megabytes. It had built-in speech, unlike all other computers. It had EXTENDED BASIC II included as the standard language at powerup. BII included over a dozen new commands, including new graphics commands, new string handling code, and several new routines that made use of hexadecimal/decimal number handling. Also available at the powerup screen was the PASCAL Prnde system. The 99/8 was in a class by itself with all these STANDARD features. Ports included the cassette, video, ac power, and the all new HEXBUS port. The expansion port on the side of the console had 50 pins, versus 44 on the 99/4A. The extra pins and some juggling of signals gave the 99/8 true 16 bit handling on its I/O bus. The Cartridge port was mounted on TOP of the unit, and installing a cartridge consisted of inserting straight down, instead of pushing the cart into the front of the console. The keyboard was totally redesigned, with several new keys added to lessen the use of the FUNCTION key for characters such as "?", "!", ":", ";", and so forth. The FUNCTION key was also moved to the left side of the keyboard, so you could have your full cursor control with one hand instead of both. Overall the keyboard was almost 4 inches wider, and touchtyping was much easier because it felt like a full size SELECTRIC typewriter keyboard. I almost forgot to add that the powerup menu of the 99/8 offered another option. This fourth selection was for setting the SPEED of the system. You could choose between SLOW mode, 99/4A mode, or FAST MODE. This speed control made it possible to change the speed at which your program ran. It was quite interesting to try to play RUNCHMAN at FULL speed, because everything ran much faster, as did most other modules.

Some specifics about the 99/8 project...

According to discussions with over a dozen different individuals, I have put together the following facts. The 99/8 project almost died in the prototype stage because of the complexity of the memory mapper. A BIG breakthrough by one individual engineer kept the project going. I have yet to get the name of this guy, but one day will run across him and shake his hand. There were

approximately 1000 etched PC boards made. Only 250 of these were actually assembled into working units. Out of this 250 unit, only about 150 were the FINAL pre-production versions. These early units, if they had the PASCAL system installed, had them on ROMS instead of GR0MS. This was for ease of debugging, until they had the FINAL version. Speaking of PASCAL, here is a interesting TIOBIT-I talked with the fellow who had the responsibility of taking the actual silicon wafers from the SC building-after etching, to Singapore, where the final GR0M chips were to be manufactured. He made it as far as Los Angeles before he got a call on BLACK FRIDAY to come back home, that the whole project was being scrapped. I'll bet that probably less than two dozen 99/8's have the Pascal system intact. My very own 99/8 does not, but then again, I still don't use my PCODE card in my 99/4A system! I have personally seen close to two dozen different 99/8's here in Texas, and all while cosmetically the same, seem a little different in the feel and some of the operating characteristics. In fact, in the FLUG group, three of our members(including me) have these consoles. The codename for the entire 99/8 project was "ARMADILLO", which for some of you non-Texans, is a fiesty little armored animal that roams the plains of the great state of Texas. More than once I have had my 99/8 on display somewhere, and along would come a former or present TI employe who would say, "How 'An Armadillo!" Most of these guys had heard about the project but had never seen one. The ones that had are some of the guys I have gotten such of this information from. Nevertheless, it does make a great conversation piece.

One thing I do need to mention about the 99/8 was it COMPATIBILITY with 99/4A software. Anything that is on cartridge will work perfectly, EXCEPT Extended Basic, for obvious reasons. Most Basic and X0 programs will work also. The kinds of programs that WON'T work are Machine code programs that are ADDED into a specific memory location. Anything that is relocatable in memory will work. The kinds of X0 programs that WON'T work are the ones that contain hidden machine code in the loaders. To tell the truth, I'm not sure if these loaders don't work with the 99/8 or the HEXBUG disk drive controller, which uses a 99/5 microprocessor as a controlling chip. I'll find out someday as let you know.

The ONLY reason that the TI 99/8 didn't make it to the market is due to the CEO-J. Fred Bucy. He was the man who decided that TI had to do the quickest thing it could come up with to heal the wound thru which hundreds of millions of dollars was flowing out. He called a board meeting and because of that meeting, the whole home computer division was canceled. So ended the hope for the future—the 99/8, as well as all the support and everything else that owners of 99/4A's needed. We still survive as users of the 52/4A, but in spite of TI.

Another project that appeared around the time of the 99/8, if not before, was the 99/2. This computer was the "littie" brother of the 99/4A, and had some very limited features. However, it did have one thing that the 99/4A could have used—it had a VERY fast system clock that enabled some kinds of programs that had repetitive code to run FAST...almost 3 times faster than the 99/4A. The 99/2 was the brainchild of a couple of engineers who thought it would help to cut costs down, and improve marketing coverage. TI's home computer marketing guy at that time decided to use the 99/2 to compete with the very popular Sinclair ZX81(a.k.a. TIMEX SINCLAIR 1000). The only problem was that by the time TI had the 99/2 ready to market, the price of the TIMEX 1000 had dropped WAY BELOW the cost it took to just build the 99/2. This little jewel never really had a chance, and most of the reason can be blamed on the overzealousness to control the ENTIRE home computer market—a market TOO big for any one manufacturer to handle. TI lost several million dollars in time and money working on the 99/2 project, and should have applied it ALL to the big brother(the 99/8), so that it would have had a better chance.

The features of the 99/2 that are worth noting are its small size, its black and white output, instead of color, its lack of a front cartridge port, instead using a new cartridge port in the very back of the console, and its HEXBUS port. There was no side I/O port, since this computer was designed for small budgets and projects. It was limited to 16k of memory, which compared directly to the Timex 1000 with its memory expansion bringing it up to 16k total. I have seen several of these consoles on the used market in the Dallas area for anything from \$50 to \$100. The 99/8 consoles have been sold a little as well, with prices going from \$600 to \$900, depending on peripherals.

I had intended to try to cover nearly everything I had mentioned at the start of this article, but the time is late, and I still have so much to cover, it may take several more hours. I will finish the history of some of the TI legends in the next installment, and will start with the TI GR0M BOX, the ancestor of the GRANCRACKER.

If you have specific questions about any of the above items, or have some

interesting tidbits to share, please leave me mail on DEPHI or GENIE, or contact me at the following address:

F.L.U.G., BOX 743005, DALLAS, TEXAS 75247
ATTENTION: RICHARD FLEETWOOD.

I will be more than happy to help in any way I can.

Til next time,

~*~*~

TI ASSEMBLY LANGUAGE REFERENCES

By: Richard Rosem
Access Engineering Inc.

The following are mini-reviews of some of the more popular assembly books available today for anyone interested in learning 9900 assembly.

If you do not know it by now the Editor/Assembler or Mini-Memory Manuals were meant to support, and not teach assembly language on the TI. However there are at least three books on the market that do cover the fundamentals.

Introduction to Assembly Language for the TI Home Computer

By Ralph Moleworth, Steve Davis Publishing.
(Available from Disk Only Software, Box 4170, Rockville, MD 20852 for \$7.95)

This book is a good for the excellent and feature-filled examples of assembly language programming. The author has a nasty habit of referring you to the mistake ridden E/A manual for information on the instruction set of the TRS9900.

Despite this, however, this reference is filled with information on how the 9900 works, how to use the Editor/Assembler and Mini-Memory modules for creating assembly code, how to do graphics, file handling, and array handling all in assembly. There is also an interesting section on using assembly through BASIC and Extended BASIC, as well as memory maps useful to the programmer.

Learning TI99/4A Home Computer Assembly Language Programming

By Ira McComic, Prentice Hall Publishers

This is a good book for learning TI assembly language. It's explanations of numbering systems, boolean algebra, how instructions work, ETC. are better than Moleworth's book. This book also has a section on how to use the Debugger or Easy Bug (MiniMemory), which is very helpful to the beginner. The book has few assembly language programming examples in comparison to the book by Moleworth. However, the book has a complete listing of the instruction set (machine code, mnemonic, what it does, examples). Hopefully this listing has fewer errors than the E/A manual.

Fundamentals of TI-99/4A Assembly Language

by M. S. Morley, Tab Publishers

This is the best book of the three. The coverage of topics like boolean algebra, numbering systems, addressing modes, what instructions do, ETC. is excellent. The author goes out of his way to explain how the 99/4A and computers in general work. Assembly language allows the programmer direct access to all of the computers hardware. This book is excellent for learning how to access the 99/4A's hardware. The book is excellent for showing the

programmer how to use the utility routines. The book has lengthy examples of how to program your own floating point routines as well as how to use the console monitor floating point routines. This book also presents the complete instruction set in the back of the book. If I were to write such a book it would be very similar.

Highest recommendation.

There is another book worth mentioning as reading for beginners or intermediates it is:

From Chips to Systems: An Introduction to Microprocessors

By Rodney Zaks, Sybex publishers

Since there is no better way to learn and use assembly (especially machine) language than to learn something of the hardware in a microcomputer, this book is a great companion to the assembly language books above. In it you can learn a little of the electronics in the chips and devices in your computer to take some the mystery out of the insides. This is realized as it goes into detail of how microprocessors in general work on the inside (8080 is the example yet with some detail on the 9900). In it you will also better understand how different memories and I/O devices work (such as DRAM, SRAM, floppy disk controllers, RS232). The book also discusses the history of the microprocessor and other chips found in your computer. It also gives a detailed comparison between popular microprocessors which gives you an inside understanding on their speed and features. With this book you will be able to discuss the esoteric details of hardware with the pro's, worthwhile experience to have in the 9900 world.

PRODUCT NEWS - LATEST RUMORS AND NEWS

By - Richard A. Fleetwood

NEWS FLASH!!!

The MYARC 9640 Computer is still not complete. The H-DOS operating system is now up to version 0.8, with many bugs and several commands still not complete. An Epson has been released that now lets owners of non-Mycrc cards to work with the system, but according to some reports, there are still lots of areas about the machine that need further work. The HYDRON processor is almost complete and finished, but those users who have gotten it have not received documentation on all the new additions to it. Also, a few very minor bugs still need to be squashed. Level 3.0 basic is nowhere to be seen, and little has been heard of its impending release. The GPL interpreter does seem to work with little to no problems. The MULTIPLAN upgrade does seem to work ok. After seeing two prototype units this past year, talking to several new owners of 9640's who can't use their machines for several different reasons, and just loads of both complaints and soothing messages, I have a couple of things to say, most of which has already been said by others.

1) the 9640, while being a viable upgrade to the 99/4a, is STILL not a finished product. Until it is, and until Myarc can make statements they can stick behind, I refuse to personally buy one. I have other products available to me that do what they are supposed to, that offer features found on the 9640.

2) After HELL over a year of hearing that the DOS will "be ready next week", and next week bringing nothing but another promise of "next week", I have come to believe that Myarc needs to rework its priorities, stating FACTS instead of HOPES. They have a good product, but they need to spend time FINISHING it instead of delivering promises they know they can't keep, such as delivery dates of hardware already promised.

3) Lou Phillips is one HELL of a guy, in the good sense of the phrase. He has done more to prolong the life of our orphan computer than anyone else in this country I can think of, and deserves more praise than he has ever gotten. I know he really truly wants his delivery dates to be met, and I'm sure that he truly believes they can, but when running a company such as Myarc, he HAS GOT to find a way to MEET those dates, and STOP delivering half finished software as he has to appease the masses and quiet them down for a little while longer. Such workmanship has brought down larger companies forever, and I'd hate to

see Myarc go that route. PLEASE, LOU, DO SOMETHING BEFORE ITS TOO LATE.

Now that I've got THAT out of my system, lets go on to a few other things. DIJIT Systems, the San Diego based company that brought professional quality RGB display to the TI 99/4a, introduced its latest product at the 99/FEST-WEST/87 in Los Angeles: The Advanced Video Processor Card. The AVPC fits into the Peripheral Expansion Box and is compatible with the existing TI99/4A software. It features 80 column text and advanced graphics with up to 512 colors. The AVPC also supports Mouse and Light Pen inputs. The DIJIT Systems card contains 192k of video RAM and is designed to work with the "DIJIT-EYEzer", an external Gen-lock ad video digitizing accessory. It will allow titling and graphic overlays on home videos as well as computer manipulation of external video images. The DIJIT Systems ADAP gives the TI-99/4A video processing power comparable with the Atari ST and the Amiga. The product is scheduled for release in August for \$195.00.

DIJIT Systems
4345 Hortensia Street
San Diego, CA
92103
(619) 295-3301

Switching to software, one of the most powerful programs to come along for the TI 99 yet is a graphics package called THE PRINTERS APPRENTICE. It allows Character Editing, Picture Editing, Formatting of text, and Scheduling all parts of a page to produce a full page, one pass thru the printer, output. You can produce ALL sorts of newsletters, advertisements, announcements, and more thru this package, and the finished product is something that is USEFUL. There are very few programs on the market today that actually make something useful. TPA is written in Forth and is powerful and fast. It is written for use exclusively with Epson and Gemini printers, and is unprotected. The TPA program comes with six different fonts for different uses. TPA FONTS DISK ONE adds 10 more new and exciting fonts. Mike McCann has also JUST released a new set of programs call the TPA TOOLBOX, with a series of programs that allow PAGE MANAGEMENT, a SIGN TOOL, FONT CONVERSION of TI Artist and CS6D fonts to the TPA font style, a BORDER BUILDER that provides over 20 different types of borders for use in your printout, a FONTS tool for making mass numbers of boxes much like columnar pad, or for making score cards or accounting worksheets. The disk also contains two new fonts for use with the TPA package. Altogether, these programs make for a TRUE Print Shop type system for us poor updated Computer owners.

Texaments just announced the availability of two new sets of USER DISKS for the owners of any of the three CS6D programs. Users Disks # 5 and 6 contain all new assortments of fonts, pictures and small graphics from that wizard of the graphics world, Dave Rose. Some of the pictures on these are the best yet. Rose has also released a CS6D CATALOGER, that allows you to catalog and sort out ALL your CS6D disks. It is a much needed tool for those of us who have WAST quantities of disks with CS6D stuff on them. The user disks will retail for \$10.95 each, and the CS6D Cataloger will retail for \$6.95. Steve Lambert, owner of Texaments, also said that the long awaited release of TURBOPASC99 is just around the corner. He finally got some word back from the authors in Germany, and has had to make some additions to the original contract. The authors in Germany are finally sending the source code, so that Steve's beta testers can make some patches to interface more cleanly with hardware more commonly used over here, as well as streamlining some of the areas that could be a little cleaner. The only real holdup now is getting the manual printed up as it is several hundred pages of both technical information and first time programmer help. Hopefully this won't take too long. The price is still set at \$79.95, but Steve is looking at a few ways to bring the price down, such as a few creative ways of making the large manual available in different ways. Stay tuned for further details.

While on the subject of THE PRINTERS APPRENTICE, CS6D, and TI ARTIST, I talked to Steve for almost an hour this evening about all sorts of things, and brought up to him the subject of TPA. He had heard a little bit about it, but had never seen it in action. I told him about the things TPA could do, and especially about the features that the TOOLBOX offered, concerning converting fonts and pictures for use with TPA. Steve had never talked to Mike, so after getting off the phone with him, I called Mike up and told him about Steve being interested in getting together with him to possibly come up with some mutually beneficial ideas and maybe even products, and Mike was very interested in doing so. Hopefully, we 99ers may see something even more awesome than we've seen before come out of this union if these two guys can get together and make something happen.

On the public domain front, Travis Watford has released his new terminal emulator program OREGA. This program, besides being a good terminal program, also features programmable keys, xmodem transfers, ascii downloads, and (here's the kicker) ONLINE viewing of RLE TYPE pictures. I am working on adding this function to the FLUG TIBBS, and should have it online shortly. Travis called the TIBBS tonight, and I chatted with him for a little while, and then went voice. He says that OREGA is far from being finished, and has much more to come, probably by this fall. The version he has in release now was to soothe those people who wanted to view RLE pics online. He said they have been BEGGING him for months to release it, and he got it to a reasonable point. A few things that will be added in the near future are ASCII UPLOADS (some of the code is already there, but now usable as is), and another method of viewing graphics. He'll have more information available later. He also is going to be forwarding me some information that will let owners of FOUNDATION 128k cards to upgrade to 512k rather cheaply. Stay tuned...

That about wraps it up for this issue. If you have some information you'd like to share with us, please contact me either at home or via the TIBBS thru a message or feedback.

'til next time...

=raf=

FCC PLANNING ACCESS SURCHARGE

The following letter was uploaded to the FLUG TIBBS last weekend, and I know for a fact that it has been going around the country for the last several weeks, as the deadline comes closer to the FCC making a decision about what to do. This surcharge is just another way for businesses such as Southwestern Bell to get their grubby paws into another profit making venture, affecting millions of home computer users who use their systems and phonelines for telecommunications. This surcharge could tack on an additional \$5.00 per hour to the usage fee of all database services, such as COMPUSERVE, THE SOURCE, DELPHI, and GENIE, and packet services such as TELENET and TYMNET. If we, the average user, do not do something about this REGULATION of public utilities, then we will all wind up being losers. Please reproduce this letter below, and forward it (preferably in multiple quantities) to the Chairman of the FCC, whose address is below. AND DO IT NOW!!!

JUNE 11TH,1987

Mr. Dennis Patrick
Chairman
Federal Communications Commission
Washington, D.C. 20534

Dear Mr. Patrick:

I am writing in regards to the recent FCC proposal to institute an access surcharge for commercial and public use of the public switched network on calls involving modems and data communications.

It is my opinion that the movement in the entire field of telecommunications is toward multiple use over common facilities. Thus any given call might contain voice, data, text, graphics or video information. As the nationwide networks become digital and as all services are digitized from end to end, what difference does it make to the type of information those lines carry?

With the advent of personal computers, and low priced modems, Americans are just beginning to realize the many benefits of communicating via computers. Shopping anywhere in the country, sending personal mail, researching databases, and many more applications only scratch the surface of what will be available in the coming years. It is my opinion that slapping an additional surcharge, over and above the already escalating end user charges, would severely retard the growth of this new power in communications. It is almost like saying that a call home to Mom on Mother's Day should carry a different charge than a call to that same number to discuss business. The distance is the same, the

connections are the same, only the content of the call is different. I believe that a voice call is inherently the same as a data call. Tones are sent over the lines from one end to another. What business is it of the FCC the telephone companies or the long distance companies as to the content of those calls?

I therefore call for a re-thinking of this proposal and hope that it will be nipped in the bud and not become effective. If greater access charges are needed, put it on all calls, equally, and let the user determine the type of information that travels down that line. Make a movement toward greater use of the networks, not less. Give the power of communications to the greatest mass of people, those who can least afford yet another surcharge on their telephone service. And let de-regulation, powered by the forces of the marketplace determine the types of services offered.

Sincerely yours,

TI-WRITER MAIL PROGRAM

By Bill Harris - April 6, 1986 - Poona Valley Users Group

I created this program to assist TI-Writer users when they are working with the form letter option of the software.

TI-Writer makes it easy to put values into a memo, letter, article, essay, form paper or any other correspondence. Just enclose a code # for the text that is variable within the asterisks. The text for the value can be up to 77 characters long. There are 99 codes (1 thru 99) possible in one document.

One needs to set-up a file of the codes and text for each (values), which is used by the Formatter when using the Mail Merge option. If no value file exists, or the value for the # within the asterisks is not in a file, you'll be prompted to enter the text as the document is printing. This is nice for text that changes often, ie. "your current status is: whoops"

The value file for this note would look like this:

- 1 Mrs. Nancy D.
- 2 Anderson
- 3 8824 S. Milltown Street
- 4 Chino, CA 91710
- #
- 1 Andy
- 2 Jones
- 3 Apt 87 - West Hillsborough Avenue
- 4 Poona, CA 91807
- #

Thus, you can print an original letter to each person in the value file. You could have as a 5th value for each record - a special personal/individualized message or comment. It's also nice for long words or phrases that are used several times in a memo, since you just type in the # # # referring to the long (77 char. max.) value in the value file. One can print all or selected values -DATA SETS- (separated by asterisks.) See your TI-Writer Manual on pages 111 and 163.

However, when sending these "form" notes to people, we usually want labels also. Since I couldn't figure out a quick way to directly use the value file for Name Address labels, I wrote a program to do it. (having written this article and program I've figured out and included an option in the prog. to just print labels from a TI-Writer created

Value File.) I got tired of maintaining both a value file and a label file. The program is in our user group library. It's a very short, simple, fast prog. and could run in BASIC or EXTENDED BASIC.

The clincher was the availability of a true Mail Listing. A listing can have lots of status codes also and one can sort it on different criteria. One of the best sorters of TI-Writer files is TI-SORT by R. Rorer and J. Clulow. It's a super fast assembly program that one just chooses from the TI-Writer menu as #3 UTIL. It's a RAM sort and is limited to 300 records, so I limited my program to 300 also. TI-SORT allows you to sort on two different fields. A major and minor sort criteria. Our club library has a copy of this fine freeware prog.

Anyway, back to my program to create Value files and Labels and get a printout of a Listing —. It works off a file you create, usually in TI-Writer with a line for each DATA SET or each Home Address record. The first record/line is a Label of the file (whatever you want. ie. date, diskfile name, content desc.) To create the file is EASY. In TI-Writer just set TABS at the indicated positions and type away. You can easily customize the file and program as it's fully explained and really simple.

This is part of the program (the routine for reading the listing file and creating a file of values).

```

10 OPEN #1:DSK1.LISTFILE,INP
UT,DISPLAY, VARIABLE 80
15 OPEN #2:"TIO" (or DSK1.VA
LUEFILE, OUTPUT, DISPLAY VAR
TABLE 80)
20 INPUT #1:LABEL$
25 DISPLAY LABEL$
27 INPUT:FROM
28 INPUT:TOO
30 FOR A=1 TO FROM-1
35 INPUT #1:ITEM$
40 NEXT A
45 FOR A=FROM TO TOO
50 INPUT #1:ITEM$
55 PRINT #2:"1 ";SEGB(ITEM$,
1,13)
60 PRINT #2:"2 ";SEGB(ITEM$,
14,27)
65 PRINT #2:"3 ";SEGB(ITEM$,
31,35)
70 PRINT #2:"4 ";SEGB(ITEM$,
60,20)
75 PRINT #2:"*":
80 CALL SOUND(-450,110,0,111
0,112,0)
85 NEXT A
90 CLOSE #1
95 CLOSE #2
    
```

The program I wrote has one feature not found in most mail listers — you select the block of records (Name Addresses) to be printed. You enter the # of the first one and the # of the last one (or just 300, if you want them all). Now we have a full data base system. TI-Writer for the great file data entry and editing and block move, copy, delete, save, combine, merge capabilities; plus my program to save a little time and TI-SORT. Again, my laziness has forced me to a better solution to a problem.

—EXPLORE— in Herms' Way

SPAGHETTI DINNER

WHO: any Dallas area TI99/4A user and their entire family
 WHEN: Sunday, August 16, 1987 at 3:00 pm till

WHERE: TI Industries, I-35 on the southbound service road, just south of GALLAGHER PARKWAY
 LOOK FOR THE FLAGPOLE!!

WHAT: Dinner, library swap, and program demos and help
 COST: \$5 per adult, \$2 under 18
 CALL KICKER AT (214)328-7257 FOR RESERVATIONS AND INFORMATION

NOTICE

THERE IS GOING TO BE AN

AUCTION

OF TI 99/4A COMPUTER HARDWARE AND SOFTWARE AT THE

AUGUST AND SEPTEMBER MEETINGS

OF THE FOREST LANE TI USERS GROUP

PLEASE DONATE YOUR UNUSED or SPARE equipment to help raise money for the TI FARE being donated for October 18.

PLEASE SEE INSIDE FOR LOCATION/TIME

I'VE GOT IT!..

...HAVE YOU?

HARDWARE MANUAL FOR THE TI 99/4A

IT DESCRIBES:

- CONSOLE DESIGN
- CUSTOM CHIP OPERATION
- TMS 9900 H/W ORGANIZATION
- TMS 9900 INSTRUCTION SET
- INTERFACING PITFALLS
- CONSOLE SCHEMATICS
- PEB CARD DESCRIPTION
- GROM SIMULATOR DESIGN
- EXTENDED BASIC MODULE DESCRIPTION & SCHEMATICS

WEEKEND HARDWARE SEMINAR
 (INFORMATION AVAILABLE ON REQUEST)

Send \$19.95 Check or Money Order To:
 (CANADA & FOREIGN SEND \$21.50 U.S. FUNDS)
 (VOLUME RATES AVAILABLE... ON REQUEST)

THE BUNYARD GROUP
PO BOX 53171, LUSBOCK, TX 79453

FROM:

~~STANIS SS~~ TEXAS INSTRUMENTS GROM MODULE PARTS, PAGE 1

~~MOBILE AL~~

THIS IS A REARRANGED LIST OF THE GROM CHIPS AND MODULE PARTS LISTING SUPPLIED BY TEXAS INSTRUMENTS IN MAY 87. (THEIR LIST IS DATED SEP 85.)

1501701-0122	CAP, .1UF 50V	0.14	1015960-1073	GROM, E/REA	3.80
1015960-0326	IC, SN6/ELF	3.60	1015960-1001	GROM, EARLR	3.60
1500773-0033	RES, 100 OHM	0.02	1015960-1002	GROM, EARLR	3.60
1015921-0004	PCB	3.60	1015960-1204	GROM, ED/AS	3.60
1037200-0004	PCB	1.96	1015960-0115	GROM, EXT.B	4.00
1056412-0101	T/C BEIGE SNAP	0.30	1015960-1113	GROM, EXT.B	3.60
1056411-0101	B/C BEIGE SNAP	0.34	1015960-1114	GROM, EXT.B	3.60
1053555-0107	T/C BEIGE SCREW	0.36	1015960-1122	GROM, EXT.B	3.60
1053554-0107	B/C BEIGE SCREW	0.30	1015960-3115	GROM, EXT.B	3.60
1015923-0007	MODULE DOOR BEIGE	0.06	1015960-0445	GROM, FACEM	3.60
1015928-0001	SPRING	0.05	1015960-0446	GROM, FACEM	3.60
1010685-0034	SCREW	0.02	1015960-2009	GROM, FOOTB	3.80
1015960-0156	GROM, 4A 2	3.60	1015960-2010	GROM, FOOTB	3.80
1015960 0157	GROM, 4A, 2	3.60	1015960-0214	GROM, HANGM	3.80
1015960-0313	GROM, 4A AR	4.20	1015960-0015	GROM, H BUD	3.80
1015960-0134	GROM, A&S 1	3.80	1015960-0211	GROM, INVAD	3.80
1015960-0135	GROM, A&S 1	3.80	1015960-0255	GROM, LOGO1	3.60
1015960-0136	GROM, A&S 1	3.80	1015960-0256	GROM, LOGO1	3.60
1015960-0137	GROM, A&S 2	3.80	1015960-0257	GROM, LOGO1	3.60
1015960-0138	GROM, A&S 2	3.80	1015960-0280	GROM, M/ADD	3.80
1015960-0139	GROM, A&S 2	3.80	1015960-0278	GROM, M/COM	3.60
1015960-0333	GROM, ALEAD	3.80	1015960-0288	GROM, M/DEC	3.60
1015960-0298	GROM, ALLMI	3.80	1015960-0283	GROM, M/DIV	3.60
1015960-0299	GROM, ALLMI	3.80	1015960-0286	GROM, M/FRA	3.80
1015960-0341	GROM, ALPIN	4.00	1015960-0287	GROM, M/FRA	3.80
1015960-0342	GROM, ALPIN	4.00	1015960-0284	GROM, M/LAW	3.80
1015960-0343	GROM, ALPIN	4.00	1015960-0290	GROM, M/PER	3.80
1015960-0344	GROM, ALPIN	3.80	1015960-0349	GROM, M/PLA	3.60
1015960-0121	GROM, AMAZI	3.60	1015960-0350	GROM, M/PLA	3.60
1015960-0013	GROM, BEGRA	4.00	1015960-0351	GROM, M/PLA	3.60
1015960-0014	GROM, BEGRA	4.00	1015960-0352	GROM, M/PLA	3.60
1015960-1012	GROM, BEGRA	3.60	1015960-0353	GROM, M/PLA	3.60
1015960-1013	GROM, BEGRA	3.60	1015960-0496	GROM, MASH	3.60
1015960-1014	GROM, BEGRA	3.60	1015960-0497	GROM, MASH	3.60
1015960-0128	GROM, BLKJA	3.80	1015960-0498	GROM, MASH	3.60
1015960-0022	GROM, CHESS	3.80	1015960-0252	GROM, MATH	3.80
1015960-0023	GROM, CHESS	3.80	1015960-0253	GROM, MATH	3.80
1015960-1020	GROM, CHESS	3.80	1015960-0254	GROM, MATH	3.80
1015960-1021	GROM, CHESS	3.80	1015960-5253	GROM, MCH.M	3.60
1015960-0261	GROM, CHISH	3.80	1015960-5254	GROM, MCH.M	3.60
1015960-5296	GROM, CON60	3.80	1015960-0347	GROM, MET M	3.80
1015960-5297	GROM, CON60	3.80	1015960-0348	GROM, MET M	3.80
1015960-0368	GROM, DEC 1	3.80	1015960-0101	GROM, MIND	4.00
1015960-0024	GROM, DEMO	4.00	1015960-0215	GROM, MINME	3.80
1015960-1026	GROM, DEMO	3.60	1015960-0303	GROM, MINUS	3.80
1015960-2006	GROM, DIA6	3.40	1015960-0304	GROM, MINUS	3.80
1015960-0236	GROM, DIV 1	3.80	1015960-0466	GROM, MOONM	3.60
1015960-0237	GROM, DIV 1	3.80	1015960-0467	GROM, MOONM	3.60
1015960-0238	GROM, DIV 1	3.80	1015960-0468	GROM, MOONM	3.60
1015960-0239	GROM, DIV 1	3.80	1015960-0469	GROM, MOONM	3.60
1015960-0345	GROM, DRAGM	3.80	1015960-0470	GROM, MOONM	3.60
1015960-0346	GROM, DRAGM	3.80	1015960-5227	GROM, MOONS	3.60
1015960-0234	GROM, DSKM6	3.60	1015960-0262	GROM, MTH-6	3.80
1015960-0235	GROM, DSKM6	3.60	1015960-0263	GROM, MTH-6	3.80
1015960-0069	GROM, E/REA	3.80	1015960-0140	GROM, MULT	3.80
1015960-0070	GROM, E/REA	3.80	1015960-0141	GROM, MULT	3.80
1015960-0072	GROM, E/REA	3.80	1015960-0142	GROM, MULT	3.80
1015960-1071	GROM, E/REA	3.80	1015960-0221	GROM, MUNCH	4.00
			1015960-0106	GROM, MUSIC	3.80
			1015960-0107	GROM, MUSIC	3.80
			1015960-0108	GROM, MUSIC	3.80
			1015960-0011	GROM, NUM.M	3.80
			1015960-0319	GROM, NUM 1	3.80

SMAUG/88 TEXAS INSTRUMENTS GROM MODULE PARTS, PAGE 2 MOBILE, AL

1015960-0320	GROM, NUM 1	3.80	1015960-0217	GROM, RDRDU	3.80
1015960-0321	GROM, NUM 1	3.80	1015960-0218	GROM, RDRDU	3.80
1015960-0322	GROM, NUM 1	3.80	1015960-0219	GROM, RDRDU	3.80
1015960-0323	GROM, NUM 1	3.80	1015960-0220	GROM, RDRDU	3.80
1015960-0324	GROM, NUM 1	3.80	1015960-0314	GROM, READ	3.60
1015960-0325	GROM, NUM 1	3.80	1015960-0315	GROM, READ	3.60
1015960-0116	GROM, NUTRI	3.80	1015960-0316	GROM, READ	3.60
1015960-0117	GROM, NUTRI	3.80	1015960-0317	GROM, READ	3.60
1015960-0118	GROM, NUTRI	3.80	1015960-0318	GROM, READ	3.80
1015960-0119	GROM, NUTRI	3.80	1015960-0074	GROM, SECUR	3.80
1015960-1120	GROM, NUTRI	3.80	1015960-0075	GROM, SECUR	3.80
1015960-0180	GROM, P-COD	3.60	1015960-0076	GROM, SECUR	3.80
1015960-0181	GROM, P-COD	3.60	1015960-0077	GROM, SECUR	3.80
1015960-0182	GROM, P-COD	3.60	1015960-5237	GROM, SEWER	3.60
1015960-0183	GROM, P-COD	3.60	1015960-0447	GROM, SLYMO	3.60
1015960-0184	GROM, P-COD	3.60	1015960-0097	GROM, SOCCE	3.60
1015960-0185	GROM, P-COD	3.60	1015960-0098	GROM, SOCCE	3.60
1015960-0186	GROM, P-COD	3.60	1015960-0131	GROM, SP/TE	3.80
1015960-0187	GROM, P-COD	3.60	1015960-0132	GROM, SP/TE	3.80
1015960-0102	GROM, P. R. E	3.80	1015960-2032	GROM, SPEEC	3.80
1015960-0103	GROM, P. R. E	3.80	1015960-0222	GROM, SPELL	3.80
1015960-0104	GROM, P. R. E	3.80	1015960-0223	GROM, SPELL	3.80
1015960-0105	GROM, P. R. E	3.80	1015960-0224	GROM, SPELL	3.80
1015960-0295	GROM, PARSE	3.80	1015960-0226	GROM, SPELL	3.80
1015960-0296	GROM, PARSE	3.80	1015960-0065	GROM, ST/SM	3.60
1015960-0297	GROM, PARSE	3.80	1015960-0490	GROM, STAR	3.60
1015960-0060	GROM, PER. R	3.80	1015960-0491	GROM, STAR	3.60
1015960-0061	GROM, PER. R	4.00	1015960-0064	GROM, STAT	3.60
1015960-0062	GROM, PER. R	4.00	1015960-0066	GROM, STAT	3.60
1015960-0063	GROM, PER. R	4.00	1015960-0067	GROM, STAT	3.60
1015960-1060	GROM, PER. R	3.80	1015960-0068	GROM, STAT	3.80
1015960-1061	GROM, PER. R	3.80	1015960-5224	GROM, STY M	3.60
1015960-1062	GROM, PER. R	3.80	1015960-5225	GROM, STY M	3.60
1015960-1063	GROM, PER. R	3.80	1015960-0229	GROM, T. DOO	3.60
1015960-0111	GROM, PERRP	3.80	1015960-0230	GROM, T. DOO	3.60
1015960-0112	GROM, PERRP	3.80	1015960-0231	GROM, T. DOO	3.60
1015960-0008	GROM, PHY F	3.80	1015960-0232	GROM, T. DOO	3.60
1015960-1007	GROM, PHY F	3.80	1015960-0233	GROM, T. DOO	3.60
1015960-5286	GROM, PIRAT	3.60	1015960-0130	GROM, T. EMU	3.80
1015960-5287	GROM, PIRAT	3.60	1015960-1129	GROM, T. EMU	4.00
1015960-5288	GROM, PIRAT	3.60	1015960-2129	GROM, T. EMU	3.80
1015960-5289	GROM, PIRAT	3.60	1015960-1092	GROM, T. I. R	3.60
1015960-5290	GROM, PIRAT	3.60	1015960-1093	GROM, T. I. R	3.60
1015960-0405	GROM, PLATO	3.60	1015960-1094	GROM, T. I. R	3.60
1015960-0406	GROM, PLATO	3.60	1015960-1095	GROM, T. I. R	3.60
1015960-0407	GROM, PLATO	3.60	1015960-0258	GROM, TCHTY	3.80
1015960-0408	GROM, PLATO	3.60	1015960-0259	GROM, TCHTY	3.80
1015960-0193	GROM, RD6FU	3.80	1015960-0260	GROM, TCHTY	3.80
1015960-0247	GROM, RD6FL	3.80	1015960-0168	GROM, TILO6	4.20
1015960-0248	GROM, RD6FL	3.80	1015960-0169	GROM, TILO6	4.20
1015960-0249	GROM, RD6FL	3.80	1015960-0170	GROM, TILO6	4.20
1015960-0250	GROM, RD6FL	3.80	1015960-0171	GROM, TILO6	4.20
1015960-0251	GROM, RD6FL	3.80	1015960-1155	GROM, TMC	3.80
1015960-0189	GROM, RD6FU	3.80	1015960-0188	GROM, TOMB	3.80
1015960-0190	GROM, RD6FU	3.80	1015960-5281	GROM, TREAS	3.60
1015960-0191	GROM, RD6FU	3.80	1015960-5282	GROM, TREAS	3.60
1015960-0192	GROM, RD6FU	3.80	1015960-5283	GROM, TREAS	3.60
1015960-0305	GROM, RDRAL	3.80	1015960-1035	GROM, V. 6AM	3.80
1015960-0306	GROM, RDRAL	3.80	1015960-1036	GROM, V. 6AM	3.80
1015960-0307	GROM, RDRAL	3.80	1015960-1019	GROM, VGRAP	3.80
1015960-0308	GROM, RDRAL	3.80	1015960-0071	GROM, WUMPU	3.80
1015960-0309	GROM, RDRAL	3.80	1015960-0153	GROM, YAHTZ	3.80
1015960-0216	GROM, RDRDU	3.80	1015960-0212	GROM, ZEROZ	3.80

TI-99/4A OWNERS SURVEY

DATE: _____

PLEASE USE A DARK COLORED FELT PEN, AND SELECT ONLY ONE ANSWER.

- A1 WHAT IS YOUR SEX? M ___ F ___
- A2 WHAT IS YOUR AGE GROUP? 17 OR UNDER ___ 18-24 ___ 25-34 ___ 35-44 ___ 45 AND OVER ___
- A3 HOW MANY YEARS OF EDUCATION? 11 OR LESS ___ 12 ___ 13-15 ___ 16 ___ 17 AND OVER ___
- A4 YOUR OCCUPATION? STUDENT ___ BLUE COLLAR ___ WHITE COLLAR ___ PROFESSIONAL ___ RETIRED ___
- A5 DO YOU USE A COMPUTER AT WORK? Y ___ N ___
- A6 WHICH BRAND? N/A ___ IBM ___ APPLE ___ DEC ___ SPERRY ___ NEC ___ DONT KNOW ___ OTHER ___
- A7 HOW MANY TI-994 OR 4As DO YOU OWN? 1 ___ 2 ___ 3 OR MORE ___
- A8 HOW MANY PEsystems DO YOU OWN? 0 ___ 1 ___ 2 ___ 3 OR MORE ___
- A9 WHICH MEMORY EXPANSION DO YOU HAVE? NONE ___ TI ___ CORCOMP ___ FOUNDATION ___ MYARC ___ MECHATRONIC ___ OTHER ___
- A10 WHAT SIZE? N/A ___ 32K ___ 128K ___ 256K ___ 512K ___ 1024 ___
- A11 WHICH DISK CONTROLLER CARD DO YOU OWN? NONE ___ TI ___ FOUNDATION ___ MYARC ___ CORCOMP ___ OTHER ___
- A12 HOW MANY 5-1/4" DISK DRIVES DO YOU OWN? NONE ___ 1 ___ 2 ___ 3 OR MORE ___
- A13 CONFIGURATION? N/A ___ SS/SD ___ DS/SD ___ DS/DD ___ DS/OD ___
- A14 DO YOU OWN A HARD DISK SYSTEM? Y ___ N ___
- A15 WHATS THE SIZE? N/A ___ 10 OR LESS MEG ___ 20 MEG ___ 30 OR MORE MEG ___
- A16 DO YOU OWN A MODEM? Y ___ N ___
- A17 INDICATE HIGHEST BAUD RATE? N/A ___ 300 ___ 1200 ___ 2400 ___
- A18 WHICH PRINTER DO YOU OWN? NONE ___ TI ___ EPSON ___ PROWRITER/NEC ___ OKIDATA/C ITOH ___
STAR MICRONICS ___ COMREX/TOSHIBA ___ BROTHER/CANNON ___ JUKI/CITIZEN ___ OTHER ___
- A19 DO YOU OWN A MONITOR? Y ___ N ___
- A20 ARE YOU USING A TV? Y ___ N ___
- A21 WHICH MONITOR BRAND? N/A ___ TI ___ AMDEK ___ SANYO/TAXAN ___ NEC ___ TATUNG/SAMSUNG ___
MAGNAVOX/SONY ___ EPSON/PRINCETON ___ HITACHI ___ OTHER ___
- A22 IS YOUR MONITOR? N/A ___ RGB ___ RGB/COMPOSITE ___ MONOCHROME ___
- A23 DO YOU HAVE THE GRAM KRACKER? Y ___ N ___
- A24 HOW MANY SOFTWARE CARTRIDGES (MODULES) DO YOU OWN (INCLUDE CASSETTE) ? 9 OR LESS ___ 10-24 ___ 25 OR MORE ___
- A25 HOW MANY SOFTWARE DISKS DO YOU OWN? 9 OR LESS ___ 10-24 ___ 25 OR MORE ___
- A26 HOW DO YOU USE YOUR TI? ENTERTAINMENT ___ BUSINESS ___ EDUCATION ___ PROGRAMMING ___
- A27 PROGRAMMERS, WHAT DO YOU PROGRAM IN? N/A ___ BASIC ___ XBASIC ___ ASSEMBLY ___ FORTH ___ C ___ OTHER ___
- A28 EVER WRITE A COMMERCIAL PROGRAM FOR TI (INCLUDE FREEWARE)? Y ___ N ___
- A29 IN WHAT LANGUAGE? N/A ___ BASIC ___ XBASIC ___ ASSEMBLY ___ FORTH ___ C ___ OTHER ___
- A30 DO OTHERS USE YOUR TI? Y ___ N ___
- A31 HOW MANY OTHERS? N/A ___ 1 ___ 2 ___ 3 OR MORE ___
- A32 DO YOU OWN ANY OTHER BRAND OF COMPUTER? Y ___ N ___
- A33 IF SO WHICH BRAND? N/A ___ IBM ___ APPLE ___ COMMODORE ___ ATARI ___ OTHER ___
- A34 APPROXIMATELY HOW MANY HOURS PER WEEK DO YOU USE YOUR TI? 4 OR LESS ___ 5-9 ___ 10-14 ___ 15 OR MORE ___
- A35 ARE YOU A MEMBER OF A USERS GROUP? Y ___ N ___
- A36 HOW MANY MEMBERS? N/A ___ UNDER 33 ___ 34-49 ___ 50-74 ___ 75-99 ___ 100-150 ___ 151-199 ___ OVER 200 ___
- A37 ARE YOU MEMBER OF A COMMERCIAL NETWORK? Y ___ N ___
- A38 WHICH ONE? N/A ___ COMPUSERVE ___ SOURCE ___ GENIE ___ OTHER ___
- A39 HOW MUCH IN DOLLARS DO YOU USE IT PER MONTH? N/A ___ \$29 OR LESS ___ 30-49 ___ 50-74 ___ 75-99 ___ 100 OR MORE ___
- A40 DO YOU PLAN TO PURCHASE (OR HAVE) THE MYARC 9640? Y ___ N ___
- A41 WHAT SOFTWARE AREA YOU WOULD LIKE FOR THE 9640? N/A ___ CAD/GRAPHICS ___ DESK-TOP PUBL ___ DATABASE ___
WORD/PROC/MULTIPLAN ___ TRUE BASIC ___ ASSEMBLY ___ C-LANG ___ OTHER LANG ___
- A42 WHAT ABOUT HARDWARE FOR THE 9640 ? N/A ___ APPLE COMPATIBILITY ___ IBM COMPATIBILITY ___ RGB MONITOR ___ 3.5" MICRO
DISKETTE ___ CD ROM ___ SPEECH RECOGNITION ___ INTELLIGENT MODEM ___
- A43 DO YOU PLAN TO PURCHASE (OR HAVE) TRITONS TURBO XT? Y ___ N ___
- A44 HOW MANY EXTRAS DID YOU BUY WITH THE TURBO-XT? N/A ___ 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___
- A45 WOULD YOU LIKE AN 'AT' UPDATE FOR THE TURBO-XT? Y ___ N ___
- A46 HAVE YOU PLANS TO PURCHASE CARDS BY OTHER VENDORS? Y ___ N ___
- A47 DO YOU PLAN TO PURCHASE (OR HAVE) RAVES KEYBOARD? Y ___ N ___
- A48 DO YOU PLAN TO PURCHASE (OR HAVE) A MOUSE? Y ___ N ___
- A49 DO YOU SUBSCRIBE TO: MICROPENDIUM Y ___ N ___
- A50 DO YOU SUBSCRIBE TO: SMART PROGRAMMER Y ___ N ___
- A51 DO YOU SUBSCRIBE TO: COMPUTER SHOPPER Y ___ N ___
- A52 DO YOU SUBSCRIBE TO: GENIAL TRAVELER Y ___ N ___
- A53 DO YOU SUBSCRIBE TO: UG PUBLICATIONS Y ___ N ___
- A54 WHAT IS YOUR OPINION OF THIS SURVEY? VERY POOR ___ POOR ___ OK ___ GOOD ___ VERY GOOD ___
- A05 THE NEXT 2 QUESTIONS ARE DEMOGRAPHIC. IF YOU ARE IN USA OR CANADA WHAT IS YOUR TELEPHONE AREA CODE: _____
- Z06 IF YOU ARE IN USA OR CANADA PLEASE ENTER YOUR ZIP CODE. ALL OTHERS ENTER COUNTRY: _____

OPTION: SEND YOUR NAME & ADDRESS & COMMENTS ON THE OTHER SIDE OF THIS FORM.

MAIL TO: ALI ULGEN ATTN: SURVEY 752 E PARKHAVEN DR SEVEN HILLS OH 44131-3918 (Dn; SURVEY F0; S/T12).

U L G E N

952 E. PARKHAVEN DR. - SEVEN HILLS, OHIO 44131-3918

(216) 524-6491

TI-99/4A USERS GROUP SURVEY

DATE: _____.

PLEASE USE A DIFFERENT COLORED FELT PEN, THANK-YOU.

- 01) WHAT IS THE NAME OF YOUR USERS GROUP; PLEASE REFRAIN FROM USING COMMAS, APOSTROPHES, AND UNLESS YOUR NAME IS AN ACRONYM, PERIODS:
- 02) ADDRESS:
- 03) CITY:
- 04) STATE/PROVINCE:
- 05) COUNTRY/ZIP CODE:
- 06) TEL. NO (VOICE):
- 07) HOW OFTEN DO YOU HAVE MEETINGS? WEEKLY ___ BIMONTHLY ___ MONTHLY ___ UNSCHEDULED ___
- 08) HOW MANY ACTIVE MEMBERS?
- 09) DOES YOUR UG PRINT A NEWSLETTER? Y ___ N ___
- 10) WHAT IS THE TITLE OF YOUR NEWSLETTER?
- 11) WHAT IS THE ANNUAL SUBSCRIPTION COST?
- 12) WHAT IS YOUR UG PRESIDENTS/OFFICERS NAME?
- 13) DOES YOUR UG OPERATE A BBS? Y ___ N ___
- 14) WHAT IS YOUR BBS NO?
- 15) WHAT BAUD RATE IS YOUR BBS? 300 ___ 1200 ___ 300/1200 ___ 3/12/24 ___
- 16) WHAT HOURS IS YOUR BBS OPEN?
- 17) DOES YOUR BBS HAVE A HARD DISK DRIVE HOOKED UP? Y ___ N ___
- 18) DOES YOUR BBS HAVE A LIBRARY? Y ___ N ___
- 19) HOW LARGE IS YOUR BBS LIBRARY?
- 20) DO YOU HAVE A REPRESENTATIVE ON SOURCE, COMPUSERVE, OR GENIE? Y ___ N ___
- 21) YOUR SOURCE ID?
- 22) NAME ON SOURCE?
- 23) COMPUSERVE ID?
- 24) COMPUSERVE NAME?
- 25) YOUR GENIE ID?
- 26) NAME ON GENIE?
- 27) OPINION OF THIS SURVEY? VERY POOR ___ POOR ___ OK $\frac{f}{}$ GOOD ___ EXCELLENT ___
- 28) CARE TO MAKE ANY SUGGESTIONS TO IMPROVE THIS SURVEY?

On; SURVEY Fn; S/UG.

INSTALLING EXTENDED BASIC INSIDE

YOUR CONSOLE..... John F. Willfooth
of West Point 99

For many of us there has been much frustration over the last several years about the "HANG-UPS" that occur to the TI-99/4A using extended basic, just as the most critical part of a program or game is reached. There are those who would lead you to believe that the power supply has been the culprit in the majority of the console locking in their club. This may have been the problem experienced in the microcosm they are in. I have experienced the problems with inconsistent and noisy D.C. voltages issued from the TI supplies also. A few months ago I ran through 5 straight VPP memory problems in a row, and could have made the statement that most if not all TI console problems will be found to have defective 4116 dynamic ram chips. This would have been absurd! I'm making this statement only to try to reassure you that of all the possible causes for console hangs, the gram connector/cartridge connection is far and away the most common, and in particular the mating (or lack of) between the Extended Basic and Gram Connector, is the greatest culprit. The purpose of this article is to assist those of you who would like to move the Extended Basic on-board.

----- DO THE FOLLOWING AT YOUR OWN RISK !

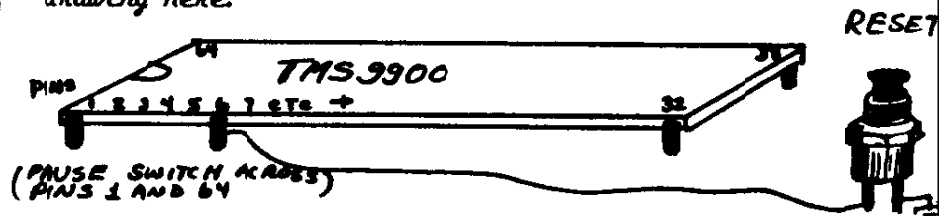
- PARTS:**
- * 1' ribbon cable (16 lead) or 2 lengths of 25 lead cable
 - * 1 Extended Basic Cartridge (shell removed)
 - * 1 Double-pole, single throw slide switch (for enabling/disabling ext.-basic)
 - * Solder, iron, wire cutters, etc

I'm not going to get into the details for I feel if by now that you can't remove your CPU from your console, you shouldn't be attempting this. Get someone who can. Remove the Gram Connector from the unit, and attach the ribbon cable to the pins of the rear of the circuit card that the Gram Connector is attached. Attach all but pins 4 and 6 to ribbon cable. 4 and 6 are unused here. Now, keeping the length of the wire to less than 8" attach the other ends of the corresponding wire to the Extended Basic card edge connector lands, remembering the relationship between the two. i.e.; pin 1 must go to pin 1, 2 to pin 2, and so on. (not 4 on 6) Before you attempt any further modification to the machine, reassemble and see if Extended Basic comes up on the menu, and still functions, SIZE, ACCEPT AT, etc. If you are still functional continue.

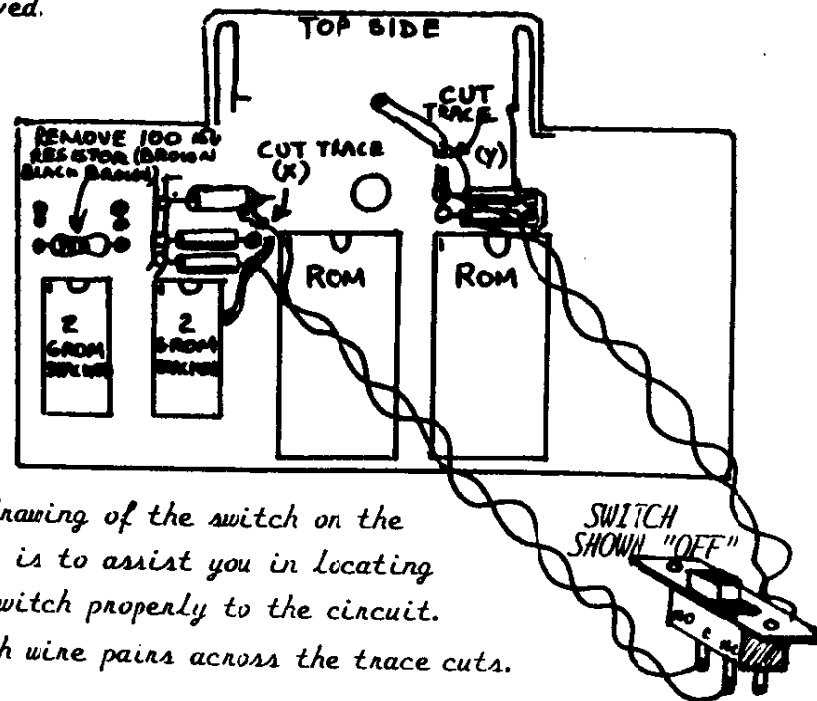
Remove the 100 ohm resistor indicated, and cut the two traces where shown. Now you may solder 4 equal lengths of wire to the switch (not longer than 10" in length). Attach two of the wires across the cut in the trace (x) and the other two across (y), making sure that the pairs are on the same switch pole set. Now, test the results again.

If the extended basic works when the switch is closed, and the cartridge slot will accept cartridges (meaning that a variety of GRAM/ROM cartridges will function), when the switch is open, then you're almost home. Mount the switch in a convenient location, and insulate the bottom of the Extended Basic Cartridge and locate on TOP RF SHIELD to left of gram conn. Reassemble. It would also be of benefit to you at this time to install a reset button across pin 6 of the CPU chip and ground. Trust me.

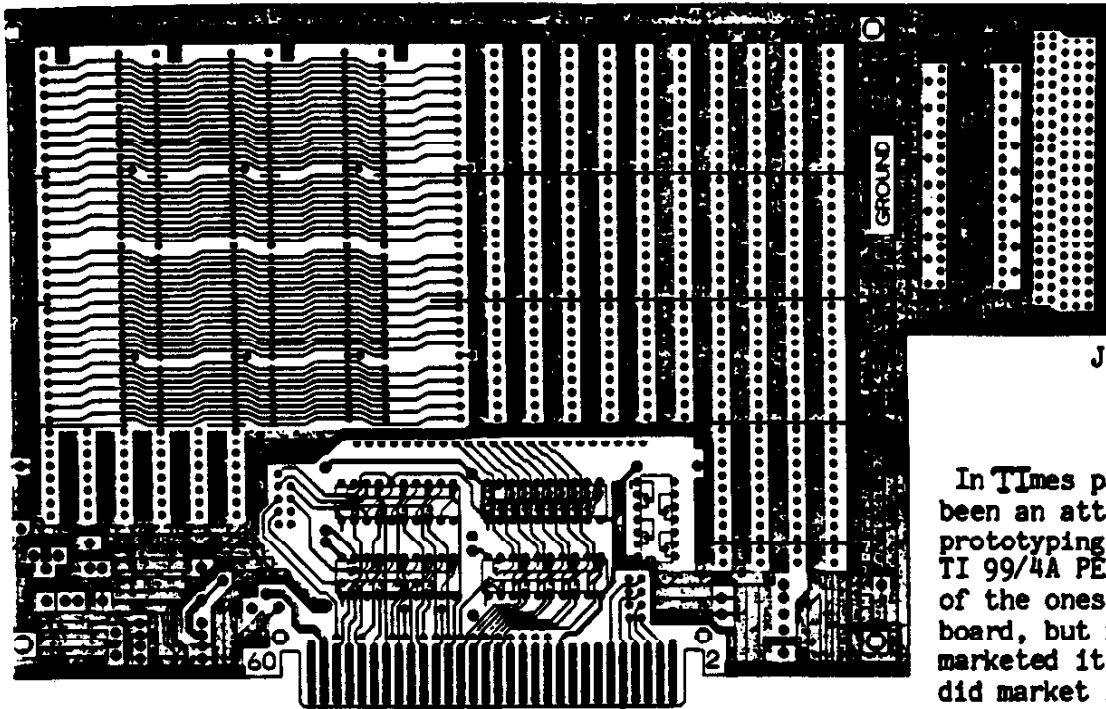
The reset switch will be particularly useful, in that you will now not be able to reset the machine by inserting the Extended Basic cartridge. You may find that without the RESET switch, you will have to turn the console power switch off and then on to begin operation after switching from basic to xbasic or other cartridges. This is an easy option to install with a momentary contact switch across pin 6 on the CPU chip (TMS9900) and ground. Reference drawing here.



The view below is here to help you find the two traces that must be cut, and the 100 ohm resistor that must be removed.



The drawing of the switch on the right is to assist you in locating the switch properly to the circuit. Attach wire pairs across the trace cuts.



PROTO BOARD
FOR PEB

by
John Willforth

In Times past, there has been an attempt to produce prototyping boards for the TI 99/4A PEB. TI was one of the ones who made the board, but never really marketed it. Others who did market it, were not as successful as they would

like to have been. The primary cause of their failure was the methods TI used in marketing the 99/4A, software, and peripherals. No one knew enough about the TI 99 system and internals to write software, let alone design hardware (which would have needed software in order to function). We believe that "Times have changed!"

Above you see a very reduced picture of a new product that Scott Coleman and I are attempting to produce NOW for those who want to do their own prototyping of PEB peripherals, or would like to take advantage of the new projects that have come out recently for the PEB, such as speech on a card in the PEB, 32K memory and Super Cart Memory in the PEB, John Clulo's DSR card for the PEB and others that are just waiting for a place to reside.

This card will not just serve the TI99/4A community, but when the "fire-hose" is removed and a 9640 (GENEVE) is in that old PEB, we have taken the special requirements of the 9640 and it's additional use of some of the sixty pin buss in the bottom of the PEB into consideration, and made it easier for some one developing for that buss to use the same card in that environment.

Scott and I are nearing completion of the design of this card as of this date (6-25-87), and within 10 days we hope to have definite pricing on the manufacture of this board. We will have everything in place to have the units made, and then take orders. So if you are interested, let us know, but don't order until we are ready. I'm not going to say more about this board until it is ready. Wish us luck.

ELECTRONIC PARTS.....

Originally I gathered the parts for the HORIZON RAM disk together myself to build the unit. feeling that there was no way that BUD MILLS could provide these parts for less than I could acquire them, and still make any profit. "I WAS WRONG!" Bud can not only provide these parts down to the battery holder, but he can provide you with many parts for other projects TI related and otherwise. The next time I order parts, it will be from BUD. At least give him a call. 419 385-5946 or write:

BUD MILLS
166 DARTMOUTH DR.
TOLEDO, OH 43614

The kit cost for the RAM DISK (less the board/docs and schematics) is \$110.00. You will have to order the board FROM:
HORIZON COMPUTER LTD. BOX 554 WALBRIDGE, OH 43465
\$53. s/h qty 1