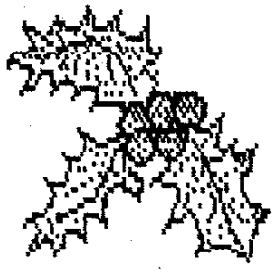
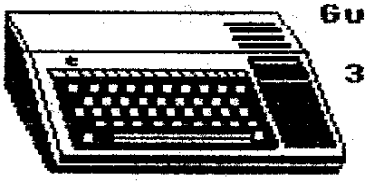
GUILFORD 99'ERS NEWSLETTER

Merry Christmas



SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER



GUILFORD 99'ERS UG
3202 CANTERBURY DR
GREENSBORO NC
27408

TO:

VOLUME 7 NUMBER 12 DECEMBER 1990

Bob Carmany, Pres. (855-1538) Emmett Hughes, Vice Pres. (584-5108)
Mack Jones, Sec/Treas (288-4280) Bill Woodruff, Pgm/Library (228-1893)
BBS: (919)621-2623 --ROS

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to:LF Jones, 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Ed.: 292-2035)

OUR NEXT MEETING

DATE: December 4,1990 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Program for this meeting will be a demonstration Tony Kleen's new TI-Base Publisher adaptation. Tony, as you well know is our resident expert in TI-Base. Stop by for a bit of refreshment and a peek at Tony's latest TI-Base adaptation!!

MINUTES

The monthly meeting of the Guilford 99er Users' Group was held on the second Tuesday, November 13th due to the elections at the Glenwood Recreation Center. There were 5 members present.

The meeting was called to order by President Bob Carmany at 7:45 P.M. The minutes were read and accepted as read. The treasury report was given.

Bob advised members that he has received a disk from OZ (Australia) that will change TIPS pictures to PAGEPRO pictures directly.

Bob then asked members what they would like to do about the Christmas program for next month's meeting. Tony volenteered to bring some soft drinks and each member is asked to bring something to serve. Anything you wish to bring will be ok as long as it is eatable.

Tony has offered to do the program for December. He will present his TI- Base Publisher that he has been working on lately. It should be a good program. Emmett Hughes showed up and resigned his office of Vice President which will be dissolved in December anyhow.

Tony suggested the club buy one of the tapes that are offered after the Chicago Faire. Bob suggested that we wait and find out the ordering info and the price from Micropendium. The suggestion from Tony was shelved for the time being.

Bob then demoed his E-Prommer. He programed the Funnelweb into an eprom and using a TI cart with a blank socket accessable from the top of the module, plugged the eprom in and placed it in the grom slot on the console. Upon powering the console, Funnelweb was instantly one of the menue given at the power up. Bob says that any program 8K or under can be put into the Epromer and kept on hand in case it is needed in the future if a certain chip goes bad. All members were duly impressed with this Epromer and a good demo thanks to Bob.

The meeting was adjourned at 9:00 P.M.

Respectfully submitted,
L.F."Mac" Jones, Sec/Treas
Guilford 99er Users' Group

CALL PEEK(PRES)

The good stuff just keeps rolling in from Australia!! Ron Kleinschafer has come up with yet another modification for his Eprommer package. This involves installing a mini-switch in the Eprom cartridge to allow for having more than one program per eprom. By using a 16K 27128 chip, and flicking the switch when the console is reset, either of two programs can be accessed. It works neat with say ARCHIVER and F'WEB. Ron is almost as prolific as Tony McGovern when it comes to updates and modifications.

As this is being written, I'm patiently waiting for yet another issue of F'WEB Vn 4.31. There were several bugs found in the initial beta test version and I'm sure that Tony will squash them all with his usual efficiency. It will indeed be a sad day when Tony finally winds down his TI involvement in mid-1991. Maybe he won't give it up completely!

This is the last President's column that I will have the opportunity to write. With the New Year, I'll leave the leadership of this lot in George von Seth's hands and confine my activities to getting the newsletter out on time. I would like to encourage each of you to try and write an article or two throughout the year to keep the newsletter rolling. I sure would appreciate more articles than I can use in a single issue of the newsletter.

As yet another year draws to a close let me wish all of you a Merry Christmas and a prosperous and Happy New Year!

RAMBLING BYTES

by "Mac"

I don't know how many of you out there have trouble with files as I do but I found out something that might be of use to you down the road some day. Prehaps you have known all along how to change a DV/80 (display variable) file to a DF/80 (display fixed) file but I didn't. A few months ago, a program showed up either in the library or BBS I don't remember which, but anyhow, it was the VIDEO CATALOG SYSTEM by Bob Pomicter of Hackettstown, NJ. As I have quite a few video tapes that have piled up over the years, I welcomed this program. I had been using a version that I had bashed to do something I wanted it to and I figgured Bob's program would be more to what I needed. Anyhow, after checking it out and seeing just what it would do I took the time (since I have quite a bit now!) and typed in all of the tapes, their contents, type of program and the start and finish of each series on the tape.

Now to the problem! Bob gives you the option of EDIT but the program only pulls up the last tape typed in. The only way you can edit any previous tape is to load it into TI/Writer. Now here's the real catch...Bob has a file that all tape information goes into called VIDEOLOG which is a DF/80 format. To edit your tapes with TIW, you will get a DV/80 file when you load it back to disk and the program will no longer be able to access it!!!

I did this, of course, dumb bunny as I am and then panicked when I realized just what I had done. Thinking all that time and trouble was wasted for nothing, I called one of the members and asked if there was someway I could change a DV/80 file back to a DF/80 file. I was told to use the EA program editor and save it to the disk and it would be in DF format. Not so my friends. It came up another DV/80!! In frustration now, I called an old friend who has dropped the 99A for Big Blue and told him my problem. No sweat, I was told, just use TIW and instead of saving to the disk with SF, to use PF(print file) and when asked for device just use DSK1.VIDEOLOG. Not so my friends, it was the same old DV/80 format!

Just for the heck of it, I reached for the TIW manual and to my delight, on page 77, there it was...Printing in Fixed 80 Format. To save you the job of digging out your reference guide I will spell it out for you and believe me, it works. Select the PrintF command. Type F and space once. Then type the device name. Press ENTER to execute the function. Now my program is working and proves the old adage is true...if it doesn't work, read the instructions!!

I do wish that Bob would add the ability to call up any tape you wish rather than go through all this business with TIW, but he only gives the town in which he lives and no address, so I can't contact him. It would also be nice to have a faster search than what is in the program, but like I said, I have the time so what the heck. All in all it's a pretty good program. It prints out the programs in alphabetical order and gives the number of the tape that your program is on.

As most of you know, Tony Kleen will be taking over my office of the Secretary and Treasurer in December. I would like to take this opportunity to thank each and every member for your support and help over the last two years. I have enjoyed serving you and my club and I look forward to being just a "plain" member for awhile. I know Tony will do an even better job than I so I ask each of you to give him the good support you have given me.

I am anxious to try out the program that Bob has received from Australia that changes TIPS to Page/Pro in one pass. That will sure cut down on the way it has to be done now. The "middle-man" TI Artist will be completely excused from the program now and a time consuming task omitted. I don't know if many of you have used Page/Pro or not but it is a very interesting program. You may insert pictures with your text and print flyers, pages or pictures at will.

Since fall is seemingly here, prehaps we can get down to some serious computing once more. There's one thing for sure, the 'ol TI is slow to die! So until the December meeting, enjoy the good TImes.

COMPUTER PHOTOGRAPHY

There might be times when you would like to take a picture of your monitor or TV screen, be it to capture your stunning graphics creation or the super Parsec score that junior ran up in his twelve hour joystick workout. Whatever the occasion might be, following are some pointers to make your photographs more successful:

1) Use a slow shutter speed (1/10 sec) to avoid video scan lines. Unlike a soap opera there is no fast moving action on your monitor that needs to be stopped thus a slow shutter speed is just fine.

2) Your monitor or TV screen picks up more reflections than you are aware of, but your photo will show. Be sure that all room lights are off and the room is otherwise dark.

3) Turn the brightness control on your TV or monitor way down, since you are working with long exposures anyway, there will still be plenty of light for your film, but the characters on your screen will "bleed" less and thus look sharper in the picture.

4) Any commercially available black and white or color film will do. Film with a higher speed will allow you to stop down more for added depth-of-field which is needed since the surface of the screen is curved and not flat.

5) Move in close so that that which you want to show fills up the frame of your picture. The proportions of the TV screen and 35mm film do not match, thus unless you move in close there will be dark bands at the top and bottom. A single lens reflex camera is best to assure good framing. If you use a view finder camera, read up on parallax correction in the instruction manual that came with your camera (on close distances the axis of your viewfinder and the camera lens are no longer in alignment).

6) The center of the camera lens should be aligned with the center of the screen to avoid distortion. Hint: Use a rubber tipped toy dart as an alignment aid. Attach it to the exact center of the screen and then lign up your camera such that no part of the dart shaft is visible when looking through the camera finder! A sturdy tripod helps.

PRK—THE OTHER DATABASE

While MultiPlan and TI-Writer are two well-known examples of productivity software for the 99/4A, PRK or the Personal Record Keeping command module is less well known and undeservedly so. Unlike MultiPlan and TI-Writer, PRK will work equally well with a cassette based console system or with a fully equipped disk system.

PRK allows you to set up a database with up to 15 fields per record. The maximum field size is 15 characters. Fields can be set up to accept only character, numeric or integer input, the program will validate input once defined. The maximum number of records that PRK will handle are a function of number of fields and length of fields. The absolute limit (15 fields X 15 characters per field is 39 records. Obviously with fewer fields or shorter field lengths the number of records can go into the hundreds.

PRK also has a built-in math capability so that the contents of certain fields can be the result of calculations from other fields in the record. As an added bonus there is even a statistical capability which calculates mean, standard deviation and linear regression on any numeric field specified.

The system provides the usual database manipulation features: add records, delete records or fields, search, select, create sub-set files and print output. PRK supports all both parallel and serial printers. As a bonus there is a "print-screen" feature which gives print output of just about any screen that PRK displays.

The benefit of PRK over "homebrew" data base systems written in Basic: Due to the fact that the program is module-based and evidently written in Assembler, a lot of memory which would be needed for program space is now available for data. Also program execution (sorts, searches, etc) is much faster than could be expected from a Basic program.

The bad news about PRK is that it was originally written for the 99/4 model which had no lower case, also since the program was written way before the PEB was added, the module does not use any PEB memory expansion. While the manual does not say so, I suspect that there is additional memory in the module.

WHERE TO FROM HERE?

That's a really difficult question! There have been a lot of developments in the past several months --and not all of them have been good! First of all, the McGoverns intend to wind down their TI involvement by mid-1991 and move on to other endeavors. I'm sure now that Vn 4.31 of the F'WEB package will be the last full effort from them. I really don't know what

hey could add to the package because the limits of memory have been thoroughly stretched as it is.

DIJIT has left the TI market as well. The 80-column card that they once produced is no longer available. Tom Spillane cites the shrinking TI market as the major consideration in maing the decision to cease production. I don't know how many of the units were sold but it couldn't have been very many.

On the other hand, there is still a bit of life in the ol' TI yet. News of its demise seem to be a bit premature! COMPRODINE has announced a series of new software offerings for the TI which includes ARTIST PRINTSHOP which is a utility to use TI-Artist instances and pictures to create cards, flyers, banners, and letterheads. There are several other new graphic programs in the COMPRODINE offerings that do much the same thing on a lesser scale. They even offer a couple of new games --LIVING TOMB and WAR ZONE. Contact COMPRODINE at the following address: 1949 Evergreen Ave, Fullerton, CA 92635 (714)990-4577.

Interesting!!

The Quest RD200 RAMdisk has made a long awaited appearance. Actually, the Quest has been around for more than a year but it was "lost" in the other hemisphere. The Hunter Valley UG is producing these cards on a limited scale and I would encourage anyone who is thinking of buying a RAMdisk to give this little beauty a long, hard look! I have a couple of them in my P-Box and the thought of a third has crossed my mind more than once.

There are still updates coming out for TI-BASE and some of the other programs despite the fact that the TI market is shrinking. In fact, some of the very best software for the TI has been produced in the past two years.

If you are looking for a TI-oriented magazine, MICROpendium continues to produce a fine effort on a monthly basis. It usually runs about 40 pages or so and there are articles for just about every level of TI user amongsts the pages. There is Regina's (C. Whitlow) column on BASIC which appears every month, a column on c-99, and Jerry Stern writes an XB column on a regular basis. Besides all of that, there are product reviews, news about TI Fairs and gatherings, and a multitude of tips, tricks, and tidbits throughout the magazine. Then there are the advertisements! I have never been a great fan of advertising but MICROpendium has enough contained within its pages to keep you abreast of the latest offerings in the TI market. It is well worth the $25 yearly subscription price!

The real answer to the original question, I suppose, is you and your TI can go wherever you want. Amazingly, it will do just about anything that any of the other "high powered" computers will do. In some cases, it will perform the same function quicker and easier. Remember, the only limit that was ever placed on the TI was by the users ingenuity and imagination --the developments since "Black Friday" have proven that beyond any doubt!!

OBSERVATIONS

A statistic that I saw the other day puts total 99/4A sales at 2,000,000 plus. Of the total by now 90% have disappeared into closets and attics with only 200,000 units still in use. Of the latter group again there are only 10% who are the truly active users. The same survey also indicates that almost all of the "bargain basement purchasers" who spent $50 or less for a console when TI was liquidating by the truckload have given up.

WHAT IS--FORTH

In addition to the two BASICs, Assembler and Pascal, Forth is the fourth programming language which is now available for the 99/4A.

This is another case where TI had a product almost ready to market when Black Friday stopped everything. In this case TI decided to put this language into the Public Domain, allowing the unrestricted copying of both software and manual.

The big advantage of Forth is that it executes with almost Assembler speed but the with ease of learning and interactive program development of BASIC. Unlike Basic, Forth makes very efficient use of hardware resources with the result that Forth variations are available to run on just about everything from the 8K Sinclair Timex to mainframe systems.

Forth does have its quirks and peculiarities one of which is that it uses Reverse Polish Notation which is indeed strange to someone used to the more conventional notation of Basic, etc. (The Basic statement: PRINT 4 + 5 in Forth reads 4 5 + ., to give an example).

Forth was invented by Charles H. Moore, a radio-astronomer and it is not surpising that Forth found its most fervent adherents in the far out scientific community. Over the last several years Forth has been used for many applications ranging

from word-processing systems to games.

Unlike Basic, Forth is not a widely publicized language and program listings in Forth are as good as non-existent. There are, however, some very good textbooks available for Forth.

Again, unlike Basic, Forth is very much standardized since the very nature of Forth is that the language is really only a nucleus which is identical from machine to machine and that the user then creates his own adaptions and extensions to that nucleus, in fact a "do-it-yourself" programming language and operating system.

TI-Forth is much more than just a nucleus, rather it already has a number of extensions to take advantage of the special features of the 99/4A such as sprite graphics, color, floating point math., etc., just to name a few. The only area where TI evidently was not quite finished was sound where TI Forth does not have much beyond a beep command.

TI BASIC TIPS

Unlike TI Extended Basic and Basic's for other computers, TI console Basic offers little beyond the TAB function to format screen output. The only way to force output to certain areas of the screen is to use empty PRINT statements, a PRINT loop, or a series of print delimiters (;;;;;;).

The following subroutine which uses a graphics assembly language call does a good job of emulating the PRINT AT command of other basics:

```
8999 REM PRINT AT ROUTINE

9000 FOR I=1 TO LEN(M$)

9010 CALL HCHAR(YY,XX+I,ASC(SEG$(M$,I,1)))

9020 NEXT I

9030 RETURN
```

In the main body of the program assign whatever line of text should be printed to the string variable M$. Assign the line number where text is to be printed to rowvariable YY and position where printing is to start to columnvariable XX. GOSUB 9000 for subroutine to execute and repeat procedure for other print lines to be executed.

MORE BASIC TIPS

by Tim MacEachern

The program listed below demonstrates how BASIC programs are stored in the 99/4A. The program as listed will work in Extended BASIC with the Memory Expansion card or peripheral attached. A similar program can be run in normal BASIC with the Editor/Assembler or Mini Memory module inserted. To convert this program to normal BASIC simply change the calls to subroutine 'PEEK' in lines 200,240 and 260 into calls to subroutine 'PEEKV'. That is, add a 'V' between the 'PEEK' and the '(' in each line. This program will not work properly in Extended BASIC unless you have the memory expansion.

The techniques used in this program are intended to make it as easy to understand as possible, while still showing how the DEF statement in BASIC can be used to do all the hard work for you. For instance, lines 100 to 130 of the program create a function HEX which will convert a string of hexadecimal (base 16) digits into a decimal number. As can be seen in lines 150 and 170, this allows us to write the actual hexadecimal addresses as used by assembler language programmers.

Line 130 takes the string of hexadecimal digits given to it and pads it with leading zeroes to make sure that there are four hex digits. Then function HEX4 is called to evaluate this four-digit hex number. In line 120, HEX4 splits the number into two two-digit hex numbers and combines them to get the proper decimal result. Similarly, line 110 splits a two-digit hex number into two one-digit numbers. Line 100 then is used to figure out the value of each separate hexadecimal digit.

Using nested DEF statements as in this program can simplify development of a working program, but be warned that DEF statements take considerably longer to run than the exact same code put directly into your lines wherever needed. Still, you may find it convenient to write some programs that consist solely of DEF statements ! After such a program is RUN in normal BASIC (or in Extended BASIC without the memory expansion), the defined functions will be available to use in BASIC's calculator mode. For instance, if your program consisted of lines 100 to 130 only, it would provide a conversion function from hex to decimal that you could use while in calculator or direct command mode.

Let's get back to the program. Line 140 defines a function that is used to convert a 16-bit unsigned number (from 0 to 65535) into a 16-bit signed number (from -32768 to 32767). For some strange reason BASIC insists on signed numbers for addresses passed to PEEK, PEEKV, LOAD and POKEV. So whenever an unsigned address is calculated function MA is used to convert it to a signed number. This function works by comparing its argument to the largest positive value allowed. If the number is too big the comparison yields a value of -1. The rest of the expression then caused 65536 to be subtracted from the argument value, giving the correct result. If the original number is okay (from 0 to 32767) the comparison yields a result of 0 and the value of the function is the same as the value of its parameter. It seems complicated to write functions like this, but try to figure them out - you may find them fascinating.

BASIC stores your program in two sections. In the top of memory it stores each line of the program, not necessarily in the correct order. As a matter of fact, each time you edit a line, it becomes the last line in this area, with all other lines packed together above it. Each statement is made up of three parts. The first byte is the length of the rest of the statement in memory. The last byte is zero, and in between are bytes that represent the particular BASIC statement you have written. BASIC keywords are translated into a single byte each (known as a token) while strings and numeric constants are represented as a leading token (199 or 200) followed by a length byte, followed by the ASCII character values of the string. By running this program you can determine how other elements of a BASIC program are stored.

Underneath the statements (that is, lower in memory) is a list of statement numbers and pointers to the first token in each statement. Each statement in your program has a four-byte entry in this list. The bottom two bytes store the statement number. The top two bytes are a pointer to the first token in the statement (the byte following the length byte). This program goes through this list and prints out each token in the statements of your program.

Pointers to the top byte in the statement pointer list and the bottom byte in the list are stored in the scratchpad RAM and read by lines 150 to 180. The loop that starts in line 190 examines each statement in the program. If you have gotten this far in the article, you will understand how the rest of the lines in the program print out each token of each line.

```
  110 DEF HEX2(X$)=HEX1(SEG$(X$,1,1))*16+HEX1(SEG$(X$,2,1))?
  120 DEF HEX4(X$)=HEX2(SEG$(X$,1,2))*256+HEX2(SEG$(X$,3,2))?
  130 DEF HEX(X$)=HEX4(SEG$("0000"&& X$,LEN(X$)+1,4))?
  140 DEF MA(X)=X+65536*(X>32767)?
  150 CALL PEEK(MA(HEX("8312")),A,B)?
  160 TOSL=MA(A*256+B)?
  170 CALL PEEK(MA(HEX("8310")),A,B)?
  180 BOSL=MA(A*256+B)?
  190 FOR PTR=TOSL-3 TO BOSL STEP -4?
  200 CALL PEEK(PTR,A,B,C,D)?
  210 PRINT "STATEMENT #";A*256+B?
  220 PRINT "TOKENS:"?
  230 SPTR=MA(C*256+D)?
  240 CALL PEEK(SPTR-1,L)?
  250 FOR I=0 TO L-1?
  260 CALL PEEK(SPTR+I,X)?
  270 PRINT X;?
  280 NEXT I?
  290 PRINT : :?
  300 NEXT PTR?
  310 END?
```

JOYSTICK TEST

```
100 CALL TITLE :: CALL CLEAR :: CALL SCREEN(6):: PRINT "RELEASE ALPHA LOCK!!!": : :
110 INPUT "TEST WHICH JOYSTICK?(1 OR 2):":NO :: IF NO<1 OR NO>2 THEN 110 :: NO=INT(NO)
120 CALL CLEAR
130 DISPLAY AT(1,3):"JOYST #";NO :: DISPLAY AT(7,11):"X  Y" :: DISPLAY AT(8,11):"_  _____"
140 CALL CHAR(33,"181818FFFFFF1818"):: CALL PLOT
150 CALL JOYST(NO,X,Y):: DISPLAY AT(10,10):X,Y :: CALL POINT(X,Y):: CALL KEY(NO,K,S) :: IF K=18 THEN CALL FIRE
160 GOTO 150
170 END

180 SUB PLOT
190 DISPLAY AT(10,4):"|" :: DISPLAY AT(12,2):"-" :: DISPLAY AT(12,4):"+" :: DISPLAY AT(14,4):"|"
200 DISPLAY AT(12,6):"-"
210 DISPLAY AT(11,3):"\" :: DISPLAY AT(11,5):"/" :: DISPLAY AT(13,3):"/" :: DISPLAY AT(13,5):"\"
220 CALL SPRITE(#2,89,2,60,41):: CALL SPRITE(#3,88,2,88,14)
230 SUBEND
240 !
250 !
260 !
270 SUB POINT(X,Y)
271 IF X=0 AND Y=0 THEN CALL SPRITE(#1,33,7,88,40):: X=X :: Y2=Y :: SUBEXIT
272 IF X=4 AND Y=4 THEN CALL SPRITE(#1,33,7,80,48):: X=X :: Y2=Y :: SUBEXIT
273 IF X=4 AND Y=-4 THEN CALL SPRITE(#1,33,7,95,48):: X=X :: Y2=Y :: SUBEXIT
274 IF X=-4 AND Y=-4 THEN CALL SPRITE(#1,33,7,96,32):: X=X :: Y2=Y :: SUBEXIT
275 IF X=-4 AND Y=4 THEN CALL SPRITE(#1,33,7,80,32):: X=X :: Y2=Y :: SUBEXIT
280 IF X=X2 AND Y=Y2 THEN SUBEXIT :: CALL DELSPRITE(ALL)
290 IF Y=4 THEN CALL SPRITE(#1,33,7,72,40):: X2=X :: Y2=Y :: SUBEXIT
300 IF X=-4 THEN CALL SPRITE(#1,33,7,88,24):: X2=X :: Y2=Y :: SUBEXIT
310 IF Y=-4 THEN CALL SPRITE(#1,33,7,104,40):: X2=X :: Y2=Y :: SUBEXIT
320 IF X=4 THEN CALL SPRITE(#1,33,7,88,56):: X2=X :: Y2=Y
330 SUBEND
340 !
350 SUB FIRE
360 DISPLAY AT(21,3):"->FIRE BUTTON PRESSED<-"
370 CALL WAIT(4)
380 DISPLAY AT(21,3):"                    "
390 SUBEND
400 !
410 SUB WAIT(T)
420 FOR X=1 TO T :: NEXT X
430 SUBEND
440 !
450 SUB TITLE
460 CALL CLEAR :: DISPLAY AT(8,3):"JOYSTICK TESTER..." :: DISPLAY AT(18,5):"PRESS:" :: DISPLAY AT(20,7):"1-INSTRUCTIONS"
470 DISPLAY AT(22,7):"(FCTN-4)-TO QUIT"
480 DISPLAY AT(24,7):"ANY OTHER KEY TO BEGIN"
490 CALL KEY(0,K,S):: IF S=0 THEN 490
500 IF K=73 OR K=105 THEN CALL INSTRUCT
510 SUBEND
520 !
530 SUB INSTRUCT
540 CALL CLEAR
550 PRINT "PICK JOYSTICK # YOU WANT TO":"TEST.  THE X&Y INPUTS ARE":"SHOWN UNDER X & Y COLUMNS":"THE LAST POSITION OF THE";"JOYST IS"
560 PRINT "SHOWN BY THE LEFT-MOST ":"FIGURE BY A RED CROSS.":"EXIT THE PROGRAM BY PRESSING":"(FCTN-4)."
570 PRINT : : : : : : : :
580 PRINT "PRESS ANY KEY TO CONTINUE."
590 CALL KEY(0,K,S):: IF S=0 THEN 590
600 CALL CLEAR :: SUBEND
```