

THE GUILFORD 99'ER NEWSLETTER

VOL. 5 NO. 8

AUGUST 1988

Janice Snyder, President , Vice President
Mack Jones, Secretary/Treasurer Herman Geschwind, Program Library
BBS: (919) 274-5760 (DPUS) ROS (919)-621-2623

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 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (George von Seth, Editor)

. OUR NEXT MEETING

**DATE: August 2, 1988. TIME: 7:30 PM PLACE: Glenwood Recreation Center
2010 S. Chapman Street.**

Program for this meeting will be demonstration and discussion of Multiplan by our resident expert Herman Geschwind. Be prepared to ask those questions that have been bothering you and/or any special problems that you are experiencing.

MINUTES

Dvt Mack Jones

The July 5th meeting of Guilford 99ers' User Group met at the Glenwood Recreational Center on Chapman Street in Greensboro, N.C.

Meeting was called to order by outgoing VP Bob Carmany due to the absence of Pres. Janice Snider at 7:45 P.M. There were 20 members and one new member present.

OLD BUSINESS:

Bob gave his reasons for resigning his office of Vice President as not having enough time to spend the proper time on the job. He is writing articles for Micropendium and the Hunter Valley User Group. Bob says this is just not leaving him the time he wishes for the office. Thanks for the good job Bob.

NEW BUSINESS:

There were no volunteers to take Bob's office, so the job of VP will have to be filled at the August or September meeting. All members are asked to think it over and either volunteer, or help pick a good VP.

The disk of the month was offered by Herman which was the new docum program TELCO version 2.0. It was a floppy with the

DDCS on the flip side. There were 3 sold.

Bob told members of a black and silver console for sale for \$20. + mailing which is being offered by Paul Center of the Pit. User Group. Also Dan Post the SYSOP of the RQS has a 1200 baud modem for sale.

PROGRAM:

The program for July was a demo of cMINDEX v/1.0 by Francisco Garcia. The complete TI Home Computer Magazine was contained in Bob's version, and he also had Micropendium up to 1987. He demonstrated how you could find headers and files using the cIndex. Bob gave an example of the versatility of the program by entering "Disk". On pressing enter, the program searched for each "Disk" that was contained in the files. Bob indicated that the program will also search a ram-disk for the same information as disk. All in all, it was a very good demo.

After the regular meeting, the TI auction began. Members were eager to get such bargains as Herman was offering. In all, not counting the sale of the modem to Ben Jones, or the nice donation by his mother, the club took in \$30., one half going to Dan Post at Herman's suggestion, for the manner in which his BBS has served and is serving our members and club. The only items left now are TI-WRITER, and HOME BUDGET MANAGEMENT. The TIW is priced at \$14.00 w/shipping and the HBM is \$4. w/shipping. It was suggested that the secretary enter these items on the TI Echo which has been done.

The meeting was adjourned at 9:05 P.M.

Respectfully submitted.

L.F. "Mac" Jones, Sect/Treas.

Guilford 99ers' User Group

++WELCOME++

We would like to welcome new member Mr. Tony Kleen from Pfafftown, N.C. Tony will be an asset to our club as I hear he knows quite a bit about "C" and that is one of our weak points. Anyhow, be sure to welcome Tony and get to know him. (He sure looks like a VP to me!!)

RAMBLING BYTES

By: Mack Jones

I was very glad to see the good turnout we had for the July meeting. We elected a 4-man search party to see if we can round up our President and see if she is still kicking!! (Just kidding Janice). We do miss Janice and hope to have her back some day.

The auction was a lot of fun and afforded quite a few members some good bargains the SYSOP Dan Post, as Dan has been very generous in letting us fill up his memory with TI goodies for the club members. Nice thought Herman. By the way, Herman, Dan left me a note on his board telling that he had received the check and he appreciated it very much. I think his words were "every little bit helps".

Sure was nice seeing all you folks signing out the different newsletters we are receiving from around the country. I intend to have them there for you at each meeting, so be thinking what you want to check out next. I would ask that you only check out 2 packets at a time so that it will afford each one a chance to read some. Please try to have them back at each monthly meet so they can be kept current.

I would like to thank Mrs. Jerry Jones again for her kind donation to the club. Just seeing her and Ben at the meet was a donation in it's self. Good to see you guys.

I was reading the K-Town newsletter and discovered something that may be of use to you members. Mr. E.M. Smith has come up with a great idea for the new contest that Hershey Chocolate Co. offer of \$100. for certain serial numbers on one dollar bills. E.M. has the idea of using user groups for information for his files he is building on the serial data. Here's the deal... E.M. will handle all the paper work which will consist of seeing who gets what if a match is discovered. In other words...if you have a dollar bill that the serial number matches one of the wrappers he has, you will send him the dollar bill by certified mail return receipt requested. E.M. will send you \$25. and the user group you belong to \$25.

also. The mailing will be refunded also. If, on the other hand, you have a wrapper that contains one of the listed serial numbers, the same deal is true. The remaining amount goes to the K-Town user group and I think E.M. will earn it!! With E.M.'s permission, I will list the serial numbers that he has compiled to date:

00151170-03402510-07245733-10099804-26674663-27179014-33834871-35000090
37191057-43332484-52187988-52492034-52847058-57640624-58424332-62579308
63156288-63165206-63326834-66399090-66879740-73348940-75720657-76056764
77109314-78893143-79151894-80940049-86551830-88385416.

These are the wrappers E.M. has so far. If you have a dollar that matches any of these numbers, call E.M. at: (615) 687-8869. He didn't say so, but I just bet he will pick up the charge for the call!

I really didn't know about this contest until about a month ago when Jo brought me a couple Hershey bars. As I don't eat much candy, I really hadn't had a bar until then. I read the contest rules and tried to match the numbers on the few dollar bills I had, but no match. I had mostly forgotten about them until I read the article in the K-Town today. It sounds like a good way to track down some bucks to me, so if any of you have any wrappers, just send them to E.M. and get you \$25. and the club \$25. After all, \$25. is better than nothing! The address is:

E.M. Smith
3506 Garden Drive
Knoxville, TN 37918

Wrappers must be requested by Nov. 1, 1988, and dollar bills must be sent by no later than Dec. 15, 1988. Any dollar bills sent will not be returnable.

Herman has been persuaded to give another demo on Multi Plan for the August meet. I heard some threats being made if certain members are not present for it, so I sure hope they don't disappoint him. hana.

You know, they say that time really flies as you grow older....I must be close to a hundred, for I have never seen it go so fast! Here it is almost August, and it seems like it should be around June to me. Soon the leaves will be turning and it will be time to hover around the fire and brush up on the files and programs we have stored back 'til we could get to them. We sure have a lot better choice this year than last. So try to remember what you did with all of those disks you stuck back before vacation and get ready to do some "down home" computing soon. See you all at the meeting in August.

TI SECRETS

By: Bob Carman

As time goes by since the day that TI pulled out of the home computer market, new undocumented tidbits of information are almost constantly surfacing. They range in scope from very simple like the undocumented "RUN CSI" that is available in XB to sophisticated bits of information that are only of value to a semi-professional A/L programmer like J. Peter Hoddie or Tony McGovern. Somewhere between the two extremes, are some goodies that can be of use to someone whose programming skill level is somewhat below that of those two luminaries. It is some of these that we are going to look at in this article.

Before we get started, I would like to acknowledge the original author of this material. The May '85 and June '85 issues of MICROpendium carried this as a two-part offering by Laurie Smith of Spartanburg, South Carolina. We are going to try to present this material in a condensed and, hopefully, more palatable form.

Everyone knows that when some cartridges are used, there are some enhanced commands that become available in BASIC. For example, we have CALL LOAD, CALL LINK, CALL VPEEK, etc. in BASIC when we use the Mini-Memory cartridge. These enhanced commands, however, are reasonably well explained in the manual that comes with the cartridge. What you may not know is that there are other cartridges that have equally powerful commands available that are totally UNDOCUMENTED.

For years, there have been articles written with BASIC and A/L routines to try to replace ACCEPT AT, DISPLAY AT, and some of the other commands that you have in the XB cartridge. These articles range from the complicated and ridiculous to simple. They have one thing in common, though. They are all completely unnecessary!!!

How many of you have the PERSONAL RECORDS KEEPING cartridge? If you don't, this might make you think about finding one or at least dusting off the one that you have and trying out some of this material.

With the PRK cartridge installed, you have access to the equivalent of both DISPLAY AT and ACCEPT AT plus several more that you probably never heard of --- I know I never did! Let's take a look at several that might be of particular interest.

1) The first one we are going to look at is the PREP subroutine that is used to reserve space in VDP for a fixed length data file. It takes the form: CALL P(V) where V is the numeric value of the number of bytes to reserve in VDP RAM. For example, you could use CALL P(10000) which would reserve 10K of space in VDP for a data file.

2) Next, we have the LOAD subroutine. It functions similarly in some respects to CALL LOAD in XB and Mini-Memory BAS with the major difference being that there is a return variable specified. It looks like this: CALL L(V\$,V) where V\$ is the name of the external device and filename and V is the return variable. In practice you could use it like this: CALL L("DSK1.DATFILE",V). Then, by using the return variable switch control to another part of the program. For example:

```
100 CALL L("DSK1.DATFILE",V)
110 IF V=0 THEN 120
120 PRINT "FILE LOAD ERROR!!"
```

3) The third enhanced command that we are going to look at is the opposite of LOAD, namely the SAVE command. It is used to save data from the data file area in VDP RAM that was created with PREP. It is similar in form to LOAD and takes the form: CALL S(V\$,V). An example would be: CALL S("DSK1.DATFILE",V). You could, once again, use the return variable to switch control to another program line.

4) The next of the commands that we are going to examine is the ACCEPT command. Here, things get a little more complicated because it can take one of several similar forms. We will look at all four forms and then give an example of the most frequently used of the lot.

```
CALL A(Y,X,C,V,L,H)
CALL A(Y,X,W,C,V)
CALL A(Y,X,W,C,V,F)
CALL A(Y,X,W,C,V$)
```

In this case Y and X are screen position, W=field width, C=return code, V=return variable, V\$=return variable, L=low value, H=high value, and F=field number. Within the return code (C) 1=valid non-empty data was entered, 2=an empty field was entered, 3=Shift-A was pressed, 4=Shift-R was pressed, 5=Shift-Y was pressed, 6= Shift-W was pressed, and 7=Shift-Z was pressed.

Now, let's take an uncomplicated look at how this would work when used in a program. Let's suppose that you wanted to accept input of a "yes" or "no" response at row 12 column 20 (this follows the form of the fourth listing). We will have row, column, field width (analogous to SIZE in XB), the return code, and the actual string itself. The lines of code would look like this:

```
100 CALL A(12,20,1,C,V$)
110 IF C<>1 THEN 110
120 IF V$="Y" THEN 700
130 IF V$="N" THEN 800
```

5) The next subprogram that we are going to look at is DISPLAY. It functions like the equivalent in XB and, like CALL A, has several different forms that it can take.

```
CALL D(Y,X,W,V)
CALL D(Y,X,W,V$)
CALL D(Y1,X1,W1,V3,Y2,X2,W2,V2$,Y3,X3,etc.)
```

Here is the explanation of the variables involved. Y and X are screen position, W=field width (the equivalent of SIZE in XB), V=display value, V\$=display string. In a program, the coding would look like this to display text at row 16 column 4:

100 CALL D(16,4,23,"ENTER YOUR PRINTER NAME")

There are even a couple of subprograms to read and write data to files created with the PREP command but from this point on we are beginning to leave the realm of what could be considered "for the average programmer". If you are interested in examining these undocumented commands in the PRK cartridge further, I would suggest that you read the articles in the May and June '83 issues of MICROpendium. Besides the articles, there are a couple of BASIC programs --- one which takes a PRK file and creates two files that can be accessed in BASIC and a second that does the reverse.

This is just one of the things that TI "forgot" to tell us. Just think what commands are available when you innocently plug one of the other cartridges into the GROM port of your computer --- the mind boggles!!!

FORTH TUTORIAL

By: Lutz Winkler

FORTH TO YOU, TOO! SESSION 6

My original intention had been to write a few tutorials for our local 99ers to get them on their way with Forth. That proved easier than to explain the basics over and over again. And besides, what I had been explaining had already appeared in condensed form in Miller's "The Smart Programmer" anyway. If some questions came up, they were easily resolved at our meetings or by a local phone call. Now that these tutorials have been posted on COMPU-SERVE I receive calls from all over and it looks as if there are a number of details yet to cover.

The question being posed most frequently indicates that some of you are not sure about Forth screens and programs. So let's clear this up:

Think of screens simply as a means to record programs, which are, however, not limited to a single screen but may occupy as many screens as necessary. As an example, let's assume the following: You have made a copy of the TI-Forth disk, booted -EDITOR, -COPY, -PRINT and -BSAVE because these will be the only ones needed for the program you are going to write. You BSAVED your autoboot starting on screen 22. The autoboot occupies screens 22 through 31. Screens 32 to 89 are now available for your program screens. (Remember, if you follow the TASK 22 BSAVE with . (dot) Forth tells you the next available screen after the BSAVE.) The idea is to prepare a working Forth disk with needed load options and then put one or more programs on it. We make the assumption that none of the unbooted load options will ever be needed for the programs on this disk, and therefore we can utilize the screens they occupy. Some people seem to have the idea that one Forth disk can do it all. That simply is impossible. You can have a collection of short routines on a disk but sooner or later you will run out of room if you try to maintain the original load option screens, unless you have two disk drives and put all your routines on a disk in drive 2.

The screens of a program are linked with the --> (load next screen) word which is placed at the very end of the screen. In this manner only one LOAD is necessary to load an entire program (nn = number of the beginning screen). You will not find --> in Brodie's STARTING FORTH, however, it is part of the Controlled Reference Word Set of Forth-83.

The line numbers are for reference only. They are not to be equated with BASIC line numbers. (Line numbers are superfluous in Forth because there is no GOTO.) However, words are compiled sequentially, i.e. starting with the first word of the first program screen and continuing down each screen line by line to the very last word of the last screen. Each word is added to the dictionary provided that any words within its definition can be found there. For example, : INVENTORY IN-STORE IN-WAREHOUSE + ; will not compile unless both IN-STORE and IN-WAREHOUSE have been compiled previously. (For advanced users who should not be reading this: Yes, this is not quite true, but remember that this is for beginners.)

For reasons which I don't understand it seems to be an obsession with some Forth programmers to cram their screens with utter disregard for legibility and clarity. If a program might take 9 screens they use every means to condense it to one less. I can see placing two short words on one line if (and that is a capital IF) one additional line would make it go into the next screen. But otherwise it is not good Forth style and it certainly does not make it easy for a beginning student to understand the program. In general, make it a practice to start words at the beginning of a line, indent the following lines if it takes more than one line for the definition. In long programs I even place the words being defined on each screen within the parentheses on line 0 so that I can find them easily with INDEX.

The normal number base (the one you're out in once Forth is booted) is DECIMAL. Invoke HEX and you can enter your

parameters in hexadecimal numbers, but do not use > to designate them as such. You can also use binary numbers, simply ; your system in that base with 2 BASE ! or better yet, define a nice word like : BIN 2 BASE ! ; I also define : DEC 10 BASE ! ; so I can go from base to base by entering only 3-letter words. In this manner you can use Forth as a handy-dandy conversion calculator. Now HEX will display a decimal number converted to hex. Just don't forget to reset the base with either DECIMAL (or DEC as above). In the same manner you can use any number base (Octal by 8 BASE ! etc), Forth will do the rest. Note: HEX is usually invoked within programs when putting parameters for character or sprite definitions on the stack.

Some of you report encountering problems with the words AT and TOP. They are not standard TI-Forth words, instead John J. Volk, THE elder statesman of TI-Forth whose Data Disks have been distributed nation-wide, originated them to save wear and tear on his typing fingers (just kidding, John). They should be part of everyone's dictionary:

```
: AT GOTOXY ; (AT is a lot shorter than GOTOXY)
: TOP CLS 0 0 AT ; (same as Miller's PAGE)
```

Time for one more question: How do I get out of Forth? Well, that depends. If you are through for the day, just pull out your disk (you might enter PLINK first to make sure there are no loose ends in the buffers) and turn off your system. If you are going to continue, enter HLL. This will return you to the TI color bar screen.

*** END SESSION 6 ***

FORTH MESSAGE CALC

By: Bob Carney

One of the misconceptions about TI-Forth is that it is difficult to have on-screen documentation available for your programs. That's true only if you try to do things the hard way. Messing around with GOTOXY and text strings can make you think that maybe documenting your programs isn't the easiest thing in the world to do. In fact, using that approach is sort of like trying to smother a fly with a baseball bat --- it can be done but it's hardly elegant!

Luckily, there is a much better way to do it! TI Forth has, as part of the resident vocabulary, the Forth word MESSAGE. MESSAGE can be used in a loop to print anything you want from any screen without having to deal with text strings and GOTOXY. There is only one catch --- everything is relative to Screen #4, Line #0. Complicated? Not really!

First of all, you should have the message screens (#4 and #5) installed on every TI-Forth disk that you have in exactly those positions. That way, if you encounter an error, you will get the appropriate message rather than "gobbledygook".

Let's assume that you have finally typed in and debugged that "dynamite" application that you have been working on. Let's say that you started at Screen #10 (on a SCREENS disk) and ran through Screen #15. Now, you are ready to add the documentation on Screen #16. How do you go about it? Quite simply, actually! Just start typing in the documentation on the very first line of Screen #16 --- no quotation marks or any other preparation. The only restriction is to make sure that each line doesn't exceed 160 characters so that it will fit on the screen when it is displayed. Once we are done, the only thing left is to calculate the values for our MESSAGE loop and design a word to access the documentation.

The calculation for the beginning of the loop is really easy to do. Since everything is relative to Line #0 of Screen #4, and there are sixteen lines per screen (0 - 15), all we have to do is subtract 4 from the screen number (16) and multiply by 16. We get 192 as the answer. That is the value for Line #0 on Screen #16. If our documentation stretches for some 10 lines, the other value will be 202 (192 + 10).

Now, we have to go back and design a word to access the documentation. How about HELP? That seems to be the universal word to use.

```
: HELP CLS 0 0 GOTOXY 202 192 DO I MESSAGE CR LOOP ;
```

There we are! Just squeeze this word into Screen #15 somewhere near the end. But let's dissect HELP before we go on. The "CLS 0 0 GOTOXY" starts by clearing the screen and printing our documentation at row 0, column 0 but the value could be easily changed to suit your own preference. "202 192 DO I MESSAGE CR LOOP;" is the remainder of the definition. Since Forth uses a "last in, first out" method of storing values on the stack, the higher of the two line numbers is entered first. All this definition does is to start printing our documentation at coordinates 0,0 and loops through lines 192 to 202 (our docs) with a carriage return at the end of each line. No GOTOXY and text strings for each line and it is much more elegant to use.

The only problem is trying to calculate the values for the starting and ending lines of your documentation. If your math is as rusty as mine is at times, the following program should help you out. Just follow the prompts as they appear on the screen and write down the values to use in your MESSAGE loops. That's all there is to it. Who says that writing on-screen documentation for your Forth programs is difficult?

```

100 DISPLAY AT(2,3)ERASE ALL:"MESSAGE LOOP CALCULATOR"

110 DISPLAY AT(6,4)BEEP:"Screen number " :: ACCEPT AT(6,18)SIZE(3)VALIDATE(DIGIT):SCR :: DISPLAY AT(8,4)BEEP:"Line
number "

120 ACCEPT AT(8,17)SIZE(2)VALIDATE(DIGIT):LNUM

130 LN=((SCR-4)+LNUM

140 DISPLAY AT(12,2)BEEP:"Screen# " :: DISPLAY AT(12,10):SCR :: DISPLAY AT(12,14):"and Line # " :: DISPLAY
AT(12,23)LNUM

150 DISPLAY AT(13,2):"Is Line 0" :: DISPLAY AT(13,11):LN :: DISPLAY AT(13,14):" relative to" :: DISPLAY AT(14,2):"Screen
#4 and Line 0"

160 DISPLAY AT(15,2):"Answer (Y/N)?" :: ACCEPT AT(23,16):CHOICES :: IF CHOICES="Y" OR CHOICES="y" THEN 100 ELSE 170

170 FOR DELAY=1 TO 1000 :: NEXT DELAY :: END

```

TROUBLE SHOOTING

By: Heiner Martin (Translated by Wayne L. Stith)

If It Comes to That...

Translated by Wayne L. Stith from "Im Falle eines Falles", HOME COMPUTER AKTIV (TI-REVUE), NR. 6/87- Juni, p.42f.; published by HCA, Postfach 1161, D-8044, Unterschleißheim, West Germany. Translated and uploaded to COMPUSERVE by permission of the publisher. The values mentioned in this article are for the European version of the computer and may not be valid for the U.S. version. Chip numbers may also be different. Please note that neither the author, the translator, the publisher nor COMPUSERVE accepts liability for any damages which may result from an individual's use of the information in this article.-Tr.

In a worst-case scenario the famous adhesive no longer helps when the TI99/4A refuses to work. Even when we cannot carry out the repair ourselves, it is nevertheless sensible to determine exactly where the error lies. The following report is meant to serve as a guide. We will discuss some details which in the course of time have proved to be weak points of the TI99/4A. For this the possession of the circuit diagrams is naturally of great value. They are even necessary for some comments made here.

The TI99/4A surely belongs among the most reliable home computers, but to be honest about it, no other machine has so completely gone kaputt by itself as this one. In the past that has occurred exclusively due to improper handling (such as removing a PCB card while the electricity was still on) or careless tinkering. However, it can come to this: Just after powering up no title screen appears and the TI99/4A only whistles in an ugly tone, or the crashes during program execution simply pile up. So we attempt to determine the cause. Regardless of the problem, all accessories are first unplugged from the console, and by 'all' I mean all, the cassette recorder included. Only the modulator and the monitor or TV remain connected; we still have to see something if something is there to see. Then we switch on the console by itself. If the problem still exists, then we must distinguish here between two cases: Nothing works, i.e. there is no title screen; or we can at least still see the title and can get to the menu by pressing a key.

(1) Nothing works.

This case is the most difficult. Here we can really only check the power supply of the TI. There are two voltages -- each alternating -- to check on the plug. Depending on the details of the transformer these lie between about 15 to 28 V and 10 to 18 V. However, it would be better if we open the console and measure the voltages directly after the regulator. The voltage regulator is found on the board which is found directly under the cartridge port. From there a four-pole cable connection goes via a plug to the motherboard. We must measure +12V, +5V and -5V on this plug, each time with a tolerance of 0.1V maximum. If you can, you should also measure the voltages under load, i.e. with electricity coming to the main board. If one of the voltages is missing, then we have found the problem. One of the voltage regulators (types 7812 and 7905 are installed here) or the regulator transistor for +5V are the main causes. Incidentally, if the regulator transistor must be

replaced, we need not look for the somewhat unusual description of the original (even TI used different types here). A normal pnp-Si-Watt transistor with a minimal load of 5A will do.

If all voltages are present, then the search becomes more difficult. If possible, we should replace all socketed chips on the TI99/4A (that's the three GROM's, the sound chip and the videoprocessor) with some from an intact console. If that doesn't help, we are left guessing. Only with great care do I mention two chips which, as I have heard, have failed: the TMS9901 (which serves in the TI99/4A as interrupt and keyboard controller) and the clock generator, the 74LS362.

(2) Something comes, but only garbage

We switch the console on, but on the screen appear only crazy characters. Here, too, we must distinguish between two cases. We do indeed see only crazy characters, but when we press a key, the TI functions "normally." A defective TMS9929A (European version?? - Tr.), the videoprocessor, is probably responsible. Since this is socketed, we can install one from an intact console as a test. But if we see crazy characters and the keyboard also does not function, then suspicion lies not only on the TMS9929A but also on the RAM's, the TMS4116. Unfortunately there are eight pieces and all should be replaced. But before we begin the great soldering task, we should likewise replace the socketed GROM's. Since the character set and the title screen are loaded from these GROM's, the problem can be here, too.

(3) The console only crashes sometimes

Well, first of all the GROM port connectors should be cleaned. That has been mentioned often in TI-REVUE but has probably not yet gotten around enough. 99 % of these occasionally occurring crashes can be traced to dirty contacts. Therefore, clean the contacts, but please use an oil-free cleaner like Kontakt WL for example (hard to get in the US, no doubt! - Tr.) If the problem is not with dirty contacts, we should try to determine whether the TI always crashes under certain conditions. For example, if this occurs always in a certain amount of time after powerup, then the problem is probably a heat problem in one of the components. Experienced electrical hobbyists can try to get behind this problem with a lighter and some cold spray. But please, this is really only for those with experience.

(4) Other problems in the console

Two defects of which I have heard more than once should be mentioned here. One is a defective keyboard, i.e. individual keys don't work. Here the keyboard should be investigated for any interruptions. There are keyboard designs by which the key joints break off. That can hardly be seen since it is hidden beneath the solder. Also the cable from the keyboard to the motherboard is often endangered for those who often take their TI apart. The hard cables like to break off or separate from the connection to the plug. Therefore, measure everything exactly and if the keyboard still doesn't function, then it is probably the TMS9901, which converts keypresses so that the processor can understand them.

Due to improper handling from outside the cassette recorder port can go bad. Here it is almost always the optocoupler (TIL118) built into the console which is defective, more rarely, the transistors or the IC-amplifiers in the cassette port. That can be easily checked: If recording doesn't work, then it is the optocoupler; on the other hand, if no sound at all comes from the recorder, it can be the IC-amplifiers.

(5) The console works by itself, but not with the PEB

The wiring of peripheral devices is considerable and so we can only give general guidelines here. Naturally we proceed step by step on the search for the guilty party, i.e. we test individual peripheral devices one after the other. First, the PEB is connected with no cards plugged in.

If the console no longer functions when the PEB alone is connected, the best thing to do is to replace all IC's on the card which is connected to the cable. That's three 74LS244's and one 74LS245. In the plug which is plugged into the console there is another 74LS245. Incidentally, the driver chips in the console also fail, but rarely. They are a 74LS244 and two 74LS367's. Incidentally, if these chips in the console are defective, one will not notice it if the console were only used alone.

Now we may not necessarily notice the defective driver chips in the cable if the console is used with a naked PEB. Frequently these make themselves known only when one of the cards is in place and addressed. Since the 74LS244's are well known for giving up the ghost here and there, according to an American magazine a couple of years ago, we should keep an eye on these chips.

Now we come to the case where individual cards can no longer be addressed: for example, after an OLD DSK1.TEST the succinct error message I/O ERROR appears, and that, of course, even though the disk controller is inserted, with rather great

probability that is due to a defective 74LS259 on the disk controller. These 74LS259's -- since they are installed also on the RS232 card and various third-party peripheral devices -- are especially endangered for users who tinker a lot with their devices. The same goes for the chips with the designation 74LS251, which are also on the disk controller and the RS232 card. If one of these is defective, not even the title screen appears after powerup. That is also really a typical error, if the console works perfectly alone and as soon as the RS232 card or the disk controller is attached, the whole system whistles in ugly fashion and the screen remains empty after powerup. Conditioned by the wiring of the TI99/4A almost all 74LS251's in the system as well as the TMS9902 on the RS232 card are "out of it."

We come to the last peripheral device which sometimes causes confusion: the 32K expansion memory. If a defect occurs here, many people think about the console. That can be easily checked: We take an Extended BASIC program and let it run once only in the console, i.e. without connected expansion memory. If it runs without problems, then the console is in order. With a connected defective expansion memory some time or other a syntax error in line XXXX will usually appear, and when we try to list this line, only junk appears on the screen or the TI hangs up completely. Such a problem definitely lies with a defect in the 32K expansion memory. In order to find the defective chip we need someone who has the debugger of the E/A module (as far as this will still run; if not, then it will get more difficult). By writing to the entire expansion memory once with the values >FF and then with >00, whereby one checks the memory for the correct contents, one quickly finds the defective memory addresses. Then the defective chip can simply be localized by means of the wiring diagram. It is recommended that you always change the eight memory chips of a memory area. If the replacement does not correspond in its data exactly to the remaining IC's, that can lead to disturbances if one changed only one chip.

For an exact search for smaller defects there are also programs from the user clubs, such as those placed at their disposal last year by TI USA. Therefore, we discussed in detail here only the serious disturbances.

Well, whoever cannot discover for himself the cause of a problem need not despair today, the TI firm still repairs the devices. Unfortunately that is not cheap. One can also inquire at the Atronic firm of Hamburg. Regardless, one should get an estimate; it is often cheaper to look for a replacement on the used market.

All suggestions and tips about errors are made here on an as-is basis and derive from six years of TI99/4A experience. Naturally I cannot guarantee that a certain error is due to a certain chip. However, the chips mentioned have proven themselves to be the culprits in the majority of cases. Electronics is such like Lotto, but reversed. The six corresponds to breakdown. Certain statistical accumulations of certain numbers (chips?) can not be excluded?