THE GUILFORD 99'ER NEWSLETTER

VOL.5 NO.7 JULY 1988

Janice Snyder, President — Bob Carmany, Vice President
Mack Jones, Secretary/Treasurer — Herman Geschwind, Program Library
BBS: (919)274-5760(OPUS) — 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: July 5, 1988. TIME: 7:30 PM PLACE: Glenwood Recreation Center 2010 S. Chapman Street.

Program for this meeting will be demonstration and discussion of "CMINDEX". It is a data base program for storing and indexing magazine articles, audio, video tapes and many other items.

AUCTION-AUCTION"

PHM 3008 Video Chess Cartridge and documentation
PHM 3058 MiniMemory Cartridge, Manual, Cassette (needs new battery)
PHM 3016 Tax/Investment Record Keeping Cartridge and manual
PHM 3035 TEII Cartridge and manual PHM 3044 Personal Report Generator -goes with PHM-cartridge and manual
PHM 3066 Home Financial Decisions Cartridge and manual
PHM 3111 TI Writer - Catridge, disks and manual
PHM 3040 TI LOGO cartridge, disks and manual
PHM 3013 Personal Record Keeping cart and manual
PHM 3007 Household Budget Management cartridge and manual
MODEM VOLKSMODEM-300 baud with cables and manual and introductory subscription to Compuserve

MINUTES

The June 7th meeting of the Guilford 99er Users' Group was held at the Glenwood Recreational Center on Chapman Street in Greensboro, N.C.

The meeting was called to order at 7:45 by VP Bob Carmany due to the absence of Pres. Janice Snider. The May minutes were read and corrected to show the power supply for the donated disk drive was by Bill Woodruff. Also, I made a mistake by stating the modem that will be auctioned was a smart/1200 modem. I was looking at the box instead of the modem which in truth is a 300 baud dumb one.

There was no old or new business for discussion.

The meeting was then turned over to Bob for the demo of his QED 32K module that was sent him by our friends in Australia. Bob passed the module around for members to look at. It is contained in a TI cart. and except for a slot in the module showing the battery, could be just any TI cart by appearance. Bob says the QED acts just like a mini-ramdisk. Apparently, the QED did not recognize the 32K memory that we have installed in the console, for it would not work. John Goller had a TI 32K at the meeting that he had been having problems with and to try it, I had to run out to my Daughter's house near by, and borrow her console without the built in 32K. Then we had some trouble with John's 32K but we finally got a short demo which consisted of loading some super-cart games and a disk called G-Language which contained graphics demos.

The disk of the month was offered and 2 were sold. There were 8 members and one visitor present. The meeting was ended at 9:06 PM.

Respectfully submitted,
L.F. "Mac" Jones, Sect./Tres.

MIDNITE MODEM

By Bob Carmany

One of the most exciting aspects of computing is being able to access the vast world of bulletin boards. It gives you unrestricted access to just about any program that you could ever want. The fact is, our own ROS board has well in excess of 5 megabytes of programs available on just about every subject imagineable. They are all ARCed and compressed and, expanded, would occupy about 20 megs or more of disk space. Of course, you will need a modem (there is one to be auctioned off this month) and a suitable terminal program. That is what this article is all about!

One of the early, good terminal programs in FAST-TERM written by Paul Charlton. Paul is one of the telecommunications pioneers in the TI world. He is largely responsible for the TI Xmodem protocol as it exists today. FAST-TERM has been through a number of revisions and is still one of the most highly used terminal programs currently available. It supports a variety of features including, quite naturally, Xmodem. While it is excellent, it does have some major drawbacks. The most devastating of which is the necessity of multiple key-presses to accomplish a task. Pressing <SHIFT><CTRL> and "N" at one time is definitely NOT the easiest way to do something. In fact, one "wag" came out with a "FAST-TERM utility" ---a two-inch square pice of plywood on a short stick to assist in pressing as many as three keys at once! FAST-TERM remains, however, a fine terminal program.

MASSTRANSFER was the next to make its appearance. It, too, has been through several revisions. It has all of the features of FAST-TERM but eliminated the multiple key-presses and some of the other inconveniences that are inherent in FAST-TERM. It supports both Xmodem and Ymodem file transfers (Ymodem is a 1K block transfer). The latest version has a 20-sector buffer that, while it increses the number of disk accesses, reduces the total duration of the accesses. The result is that your total file transfer time is reduced. MASSTRANSFER is still a bit cumbersome to use. The choices for uploading and downloading and not particularly logical but it is easier for the novice to use than anything that came before.

The last entry in the field is TELCO. Like the others, it has been through several revisions with the latest version being 2.1. It has Xmodem and Ymodem as well as CompuServe "Quick B" file transfer. It also adds the ability to emulate a variety of terminals. It is more logically put together and the instructions for configuring the system are easy to follow. Enough to get you interested?

It supports auto-dial, logon macros, and much more. The program uses the unique procedure of moving pieces of code in and out and presenting them in "windows". All in all, it is probably the easiest of the lot to use. The latest version has everything speeded up and the file transfers are almost effortless.

I guess the bottom line, though, is use what you are used to. There are "die-hards" out there that swear by FAST-TERM with its multiple keypresses and other minor shortcomings. Some will use no other terminal program but MASSTRANSFER despite the fact that the disk accesses are frequent and the menu isn't the most logical or easiest to use. But a growing number of the "midnight modem" set is switching over to TELCO.

So, if you have an RS232 and a modem (again, one to be auctioned at this meeting), pick one of these fine terminal programs and you are all set. Incidently, all of these programs are either fairware or user-supported software and readily available for a nominal price. It would be a shame to miss the myriad of programs and applications that are available just by

/aling a local telephone number ---it's amazing what you can get for a dime these days!!!

RANDOM BYTES

By Mac Jones

The good news this month is the auction of the modem that Herman donated to the club a couple months ago. We thought it best to make sure that every one got the news so we waited for the July meet to have the auction. Even if you have a modem, this one would be an excellent back-up in case some gremlins happened to land on the one you are using.

There are some good articles this month in the newsletters. In case any of the adventure freaks have ADVENTUREMANIA ADAMITIS and haven't been able to figgure it out, the EAR 99'ers June issue has excellent hints as well as a map for anyone who wants to check it out. Also Mini-Memory part IX. You might also want to check out the fine article by John Wilforth on the clock/calander analog-to-digital circuit for the "Proto Board" and also "A REAL TIME CLOCK FOR THE TI-99 4/A. Both of these articles are from the West Penn 99'ers Club newsletter. These newsletters that we get from our friends in other cities are full of good information and I hope you will check them out at the next meeting. I have finally gotten them together in each group, so you must check out a certain group instead of just one issue.

I really think it is time that we as a club took a good look at where we are going and where we want to go. At the last three meetings, we have only had seven or eight members present. I wanted to blame it on the summer vacations but there couldn't be that many weeks of vacation allowed workers. I guess what it boils down to is do we want to continue as a club or do we want to give up what we have and just a few of us that are still interested in the TI just meet in each others homes when ever? I really hate to even think of the group breaking up, but there is really no need for a hand full of people trying to maintain meetings, newsletters, and programs if no members care enough to show up once a month.

There have been very good demos and informative programs given with very little attendance. It isn't easy to work hard preparing a program and then not having but a few there to enjoy it. Think on what you want to do as far as continuing as a group and we will talk about it in the coming meetings. We have heard that the Winston Salem group is about to go under and from some of the newsletters we get, other states have the same problems we are having of poor attendance. Come on to the July meeting and let's talk it over.

Bill Woodruff is going to exchange the transformer for us so we can build the power supply and enclose it in the case with the second disk drive. As soon as we get the transformer, George and I can start building the case and hopefully, have it at the August meeting.

Here's hoping you all have a happy 4th of July and see you at the meeting on the 5th.

DISK CONTROLLERS

Disk Controllers - from TI to MYARC

The views expressed in this article reflect the author's personal experience with TI, Corcomp, and Myarc disk controllers. Technical data has been verified wherever possible, but is not publicly documented in some instances. Please bring any errors to the attention of the author.

The disk capacity of the TI99 has increased in just a few years from less than 80K (a single one-sided 35 track drive) to almost 2.9 megabytes (four double-sided, double-density, 80 track drives). The early standalone was replaced by the PEBox system which would support three double-sided 40 track drives (540K). Corcomp introduced their four drive double-density system (1440K), followed by Myarc's similar system with two double-density formats (1280K and 1440K). Then in 1986, Myarc offered its 80 track upgrade which doubled capacity again. Even as capacity was increasing rapidly, the TI and Corcomp controllers differed only modestly in I/O speed. When MYARC introduced its fast DSDD controller, few reviewers did justice to its speed advantage. Early comparisons were done at the standard TI or Corcomp interlace, but the big speed gains required taking advantage of the much tighter sector interlace possible with the high-speed MYARC card. To understand how this works

we need to take a look at the way a disk drive performs.

Disk Drive Fundamentals

A floppy disk drive writes information in concentric rings called "tracks" on a thin plastic disk coated with a film of magnetic particles. Each track in turn is divided into blocks of information called sectors. A blank disk has one (or more) index holes used to synchronize the process of writing to and reading from the disk. The type with many holes are called "hard sectored" since each sector has its position fixed by an index hole. The type of disks used by most computers have only one hole and are called "soft sectored". In this system the computer must write magnetic signposts on the disk to mark out each sector in a process called "formatting" or "initializing" a disk. These signposts take up a subtantial fraction of the space on a track since they include not only sector numbers but buffers (filler bytes) that allow the computer to get into synchronization to read or write sectors of data and to prevent the sector identifier from being overwritten by a drive operating at a slightly different speed from the drive that formatted the disk.

The typical 5.25 inch disk drive has a "stepper motor" capable of moving the drive's read/write head(s) in or out along a radius of the disk in steps of 1/48 of an inch (thus the terminology "48 tpi" = 48 tracks per inch). Since the inner tracks have a smaller circumference, they crowd the bits of information together. Magnetic coatings on a floppy disk are rated by their capacity in bits per inch at standard magnetic flux for the write head. This figure is usually over 5000 bpi for modern floppies, but was somewhat lower a few years ago. The circumference of the inner track of a 40 or 80 track disk is about 10 inches -- which allows about 6250 bytes to be written on the track without exceeding 5000 bpi. For comparison, the Corcomp double density format requires over 6400 bytes per track. Media limitations were the reason that some early 5.25 disk drives only used the outer 35 tracks. The 16 sector (by 256 bytes/sector) format recommended by most drive makers requires only 6250 bytes per track and includes several hundred additional "buffer" bytes to compensate for differences in drive timing.

Timing is EVERYTHING

With soft-sectored disks, the integrity of the read/write processes require critical timing. The disk rotates at 300 rpm within a small margin. This means there are about 250 thousand magnetic pulses (bits) passing beneath the head each second. In single density format, the majority of these pulses are timing or filler bits -- in double density, many of the timing bits are suppressed in order to double the rate of data bits. In a typical sector read the drive must bring the disk up to speed, recognize the index hole, step out to track zero (to get its bearings), determine single or double density, verify its position, step in to the target track, verify the track number (written in the format operation), detect the sector identifier as it flies past, then immediately read the 256 data bytes into memory. Five of these operations require accurate reading of the magnetic pulses whizzing by at over 250K bits per second.

If you do some quick arithmetic (256 bytes/sector = 2048 bits/sector into 250k bits/second)... hmmm... Why can't the drive read a 125 sector file in one second? Well first many of those bits are not data bits, they are overhead to keep things synchronized and allow for timing variation between drives. Second, some time is used moving the head from one track to the next when more than one track must be read. Third, 250K is the instantaneous read rate and the computer must take time to do other things like move the last sector out of its buffer to make room for the next one. In the standard TI protocol for reading a disk, the data is moved into VDP ram (so the drive could be used without the memory expansion) before it goes to the expansion memory. All this thrashing eats great chunks of the time available for reading data. By the time one sector is safely tucked away in the 32K card, several sectors have already passed by the drive's read head. If the sectors were written consecutively on the disk, we would have to wait a full revolution (0.2 seconds) before the next sector would pass under the head. To avoid this inefficiency, the consecutively numbered sectors are spaced out around the disk so that they are separated by just enough time to take care of other business. The actual pattern in which the sectors are scattered is called the "interlace". The idea of the interlace is to spread the sectors out to match the timing needs of the hardware -- both the time needed to stash each sector and the time needed to step from one track to the next and get the the head settled down for some serious (250K bps) reading.

Interlace and Head Step Times

Life was simple with the TI disk controller. Both the interlace and the head step time were locked into the controller's PROM (that's the programmable chip that contains the control programs for the card). The head step time is the built-in delay between step signals to allow the stepper motor to move the head one "click" in or out. The TI settings are very conservative

(read "slow") to allow for slow drives. The step time is 20ms -- if you step from track zero to track 39, it takes 20x39=780ms, almost four revolutions of the drive. The TI interlace lays the sectors down on a track in the order 0753106942. This allows all sectors to be read in four revolutions of the disk though the slow head step lets another revolution go by between tracks. Thus the maximum read rate is about 9 sectors per five revolutions (= one second) or 2304 bytes per second.

When Corcomp designed its double density disk controller, allowances were made for the increased speed of later drives by permitting the step rate to be set with DIP switches for each drive. The step rates available are 30, 20, 12, and 6ms (the faster values quoted in the CC manual are referenced to the wrong clock speed). They also provided a choice of interlace options, though only a couple of them are practical. The default interlaces are labeled "7" for single density and "10" for double density. The single density interlace is the same as TI's, but with a faster step setting the head be can moved without losing a revolution and thus reads 20% faster than the TI controller. The double density interlace allows 18 sectors to be read in five revolutions, but it doesn't leave enough margin to stash the last sector and step the head in time to catch the zero sector of the next track (that's why the sector number "hangs" for 0.2 seconds each 18 sectors while verifying a formatted disk -- you are seeing the extra revolution needed to acquire the first sector of the next track). Thus the maximum read rate is 18/1.2 or 15 sectors per second, about 67% faster than the TI controller. Users of the CC controller have probably noticed that it loads its own MANAGER program faster than this. In this case a special loader bypasses VDP and loads directly to CPU RAM -- this faster handling of the data allows the stepper motor to be activated sooner and saves one revolution per track (so the 98 sector file can be read in about 5.5 seconds). This provided a foretaste of the speed that MYARC would achieve with its double density controller.

The MYARC controller bypasses VDP RAM to load directly to CPU RAM. This technique coupled with a buffer RAM chip on the controller card provided a quantum jump in disk I/O speed. The MYARC card reads the TI single density interlace at 11.25 sectors/second (the same as Corcomp) and reads the CC 18 sector/track interlace at 18 sectors/second (the same speed Corcomp reads its MANAGER program), but this is only the beginning. Since the hardware empties its sector buffer faster, consecutive sectors can be placed closer together allowing a track to be read in fewer revolutions, i.e., it supports a faster interlace. With fast drives, the 9 sector/track single density format can be read at interlace "2". (NOTE: In the MYARC terminology, the interlace number represents the number of disk revolutions required to read a track.) This works out to 22.5 sectors/second compared to 9 for the TI and 11.25 for the CC controller. The MYARC 16 sector format can be read at interlace "3", 26.67 sectors/second -- 3 times as fast as the TI controller and almost twice as fast as Corcomp double density. The Corcomp 18 sector format can be read at interlace "3" or "4", but the data rate is the same in either case, 22.5 sectors/second. Interlace "4" is smooth but requires a very quick head step, interlace "3" reads the track in 3 revolutions but forces an extra revolution for the step from track to track because sectors 17 and 0 are adjacent on the disk. Though both interlaces have the same data rate, interlace "3" is safer if you are uncertain about the speed of your stepper motor.

In order to read and write both double density formats, the MYARC system must insert an additional step in some I/O operations -- sector zero must be read to determine whether a double density disk has 16 or 18 sectors per track. This datum is needed to convert the the logical sector numbers used by the TI operating system into track and sector-within-track addresses for the floppy disk controller chip. The TI and Corcomp controllers do not need this step because they do not use the full potential of the TI disk I/O protocol. Once this step, accessing sector zero, is added to the various disk operations, it opens the system up for using more than two formats -- including 80 track formats.

Beyond Double Density

A two formal system can be managed using only the floppy disk controller's inherent ability to sense single and double density recording patterns. To get beyond this limitation, the additional data stored in sector zero must be read, stored, and used to modify the special binary commands sent to the FDC (floppy disk controller) chip. Fortunately the TI99/4A system design already provides for such innovations through the Device Service Routine concept and standard "GPL" calls. The system doesn't care what hardware is attached as long as it plays by the rules -- an interface program stored in a memory chip (PROM) on the peripheral device does the trick. This program handles calls for I/O operations from other programs such as TI Writer or the Basic Interpreters. Another set of rules controls the way disk and file information are saved on a disk. Disk parameters are stored in sector 0, while sector 1 must have a two byte "pointer" (a hexadecimal sector address) for each block (one sector) containing the bookkeeping data for a file. It is these blocks that are scanned in order to display the disk directory.

Since the Myarc controller must read sector zero to determine the number of sectors per track, the other parameters in that sector are available to control other variables such as number of tracks. But there were other limitations to overcome. The number of files on a disk is limited by the space available for pointers. 256 bytes at 2 bytes per pointer would give 128

files -- except the pointer list must end with a null word (>0000) so directory routines know where to stop -- so we get 127 files per disk. The pointer itself can address sector numbers as high as 65535, so this is no problem. The real limitation is the bit map in sector 0. It begins at byte 56 leaving only 200 bytes or 1600 bits available to map the disk. Since a bit must be turned on for each sector used, the 1440 sector DSDD 40 track disk is already near the limit. The answer devised for the 80 track DSDD system is to map two consecutive sectors with each bit. It wastes some space but no more than systems that use a standard 512 byte sector.

Making the Quad System Work

So now lets say we have new code in the disk controller EPROM (an "erasable" version of the PROM chip used by TI) that does all the proper tricks with the bit map and has the FDC commands to control the new 80 track drives we have added to the system. We still have to tell the controller which drives are 80 track and find a disk manager program that can use the new commands. The selection problem can be taken care of using the DIP switches on the card (but in the process you lose their original function -- setting step speed). Since the Eprom responds to standard GPL calls, most functions can be handled by the TI Disk Manager 2 cartridge. The exception is the disk formatting process -- the formatting works OK, but the initial data written into sector zero is for the standard bit map. (This can be fixed by changing byte 56 from >03 to >01 with a sector editor.) Read/write operations from XB or TI Writer work fine since they use the GPL protocols. Myarc has an excellent disk manager program that works beautifully with 40 track drives, but it has suffered from a number of subtle bugs in 80 track mode. This program, like many others designed for high speed I/O, uses assembly language code to handle the FDC -- bypassing some of the routines in the EPROM. Differences in bit map handling, even slight differences in execution times can affect the performance of 80 track drives. The code in the 80 track EPROM has had a lot of attention to proper timing -- the price you pay for higher performance.

Fine Tuning the Myarc Disk System

Before you start using the Myarc system routinely, there are some experiments that can get maximum performance from your drives. Use the Myarc disk manager to try different interlace settings -- first with your 40 track drives, then with the 80 track drives. Watch for hesitations as each formatted disk is verified, then use the Test option to read the sectors you have layed down. Look and listen for "retries" -- when the sector number pauses with a head seek noise. Use the best disks you have and note the combinations that test smoothly. With fast drives in good condition, you should be able to run 9 sector (single density) format at interlace 2 and 16 or 18 sector double density format at interlace 3. Don't worry if 18/3 pauses at the end of each track -- this is just the extra revolution forced by having sectors 17 and 0 adjacent on the disk.

When you try this with 80 track drives, don't be surprised if the results are different. The time required for the head to settle into a wide standard track may not be adequate to get it reading properly from the narrow tracks on the quad drive. Such subtleties as erase delays and disk quality are also more critical on the skinny, low power tracks. My Mitsubishi 4853s (96 tpi) will support both 16/3 and 18/3 but are unreliable at 18/4, while my TEAC 55Bs support all three at 48 tpi. Don't take chances with any setup that is marginal. The error rate may be low, but it always seems to happen to a file that isn't backed up.

Hot Rodding

If you want to try for a little more speed, there are two more tricks you can use. The faster WD1772 FDC chip is pin compatible with the standard WD1770 supplied by Myarc. It will try to step the head at 2ms rather than the 6ms setting of the standard chip. (The 80 track EPROM automatically uses the fastest step speed available.) Many of the latest drives can step at 2ms or 3ms even though they are conservatively rated at 4ms or 5ms. The change is noticeable but may not be worth the high price of the WD1772 (it is not a commonly used chip and is rarely discounted). The second fix is cheap and very useful for producing large quantities of copies. The FDC chip's automatic "write verify" function can be defeated by shorting one pin on the controller card to ground. This is best done with a switch so the verify can be enabled for normal operations. The effect of this modification is equivalent of the "turbo" option on the Corcomp controller and should be used only after testing.

Interlace Patterns

Note: The configurations marked * and ** are the standard interlace patterns for TI and Corcomp formats. The end-of-track intervals are only approximate since the 9 and 16 sector formats include more buffer space than the 18 sector format.

Sect/trk Interlace Pattern (dashed line is time available for head step) -------- ---------

9 4 * 0 7 5 3 1 8 6 4 2 ---------- 9 2 0 5 1 6 2 7 3 8 4 ------

18 5 ** 0 11 4 8 15 1 12 5 9 16 2 13 6 10 17 3 14 7 --------- 18 4 0 9 5 14 1 10 6 15 2 11 7 16 3 12 8 17 4 13 ------ 18 3 0 6 12 1 7 13 2 8 14 3 9 15 4 10 16 5 11 17 none

16 5 0 13 10 7 4 1 14 11 8 5 2 15 12 9 6 5 ---------- 16 3 0 11 6 1 12 7 2 13 8 3 14 9 4 15 10 5 ------- 16-sector patterns are not precisely to scale

FORTH TUTORIAL

By Lutz Winkler

FORTH TO YOU, TOO! SESSION 5

When we set up our autobooting system disk I stated that I always include -DUMP. This utility provides a lot more than what I am going to cover here since I want to keep things as simple and understandable as possible. As you may have gathered by now, Forth is a 'stack-oriented' language. There are several 'stacks' but whenever there is reference made to 'the stack' it means the PARAMETER stack. (Parameter = argument.) The stack's main function is for temporary storage of arguments, i.e., numbers. Every time you enter anything from the keyboard (or a word is encountered in a pro- gram) Forth first looks for it in the dictionary. If it is not there, it is converted to a number and put on the stack (If found in the tionary it is executed). By now you know the stack concept: What went on the stack last will be taken off first. Because there is a limit to your computer's memory capacity, there is also a limit as to what the stack can hold. Good programmers make sure that the stack holds only what is needed and don't let 'garbage' accummulate there. If by chance you define words which leave junk on the stack, it will eventually reach its limit and the program will stop with a ?FULL STACK message on your display. Conversely, if a words needs to fetch a parameter from the stack and nothing is there, you'll get a ?EMPTY STACK error message.

Bring on .S (dot-S). It let's you look at the stack content without touching it otherwise, i.e., it neither adds nor removes anything. For example, let's enter 15 and then 22. Now enter .S and see what you get on the display. It should show : 15 22. The : symbolizes the bottom of the stack. In other words, if we use . (dot) then 22 should be printed to the display because it is the top (first-out) item on the stack. Now use .S and see what's left. Only the 15. Another . (dot) will fetch it and if you use . once more Forth will respond with ? EMPTY STACK. (Usually preceeded by a number.) In order to program in Forth you must understand the whys and hows of the stack.

Speaking of stacks, there is another one, though it is never called a stack. It goes by the name of DICTIONARY, but just like nearly everything in Forth it is also a stack. Every time you define a new word it ends up on top of the dictionary (stack). On the bottom reside - you guessed it - the Forth resident words. Our autoboot then piles the words from the load options on top and finally you add your words (or your program's words). Large programs can use up almost all of the memory. Say you have loaded the AAA SUPER-DUPER XY CALCULATOR and there are now 1,500 bytes free. You are through calculating and wish to install the PARAGON XY PLOTTER. You may not be aware of the fact that it takes 4,000 bytes, so as it boots there comes the point where your TI has reached its limit. ?DICTIONARY FULL will be the message to let you know that there is no way you can run the XY PLOTTER with SUPER-DUPER CALCULATOR still in memory. Well, there's always COLD to start over. Not necessary. FORGET is easier and faster. FORGET cccc (as it is stated in the manual) wipes everything out of memory starting with cccc and every word which was added after it.

One way to always know what to FORGET to get rid of a program, but not the autoboot, is to include on the WELCOME screen a do-nothing word. It should be added as the last word just before the R->BASE word. It can be anything you like, most people use their initials to help them remember to FORGET. (How is that for logic?) Every word compiled prior to my : LW ; remains in the dictionary, every word added afterward is dropped by FORGET LW <ENTER>. In the case of very short routines which I may load on top of another one I usually include a : XX ; or similar do-nothing and display a prompt upon exiting to

remind me what to forget. In this fashion I leave the underlying program in memory.

RECAP:

The .S word displays the parameter stack's content without adding or removing anything. "!" denotes the bottom of the stack.

FORGET cccc let's you clear from the dictionary entries beginning with cccc and every word added since cccc was compiled.

Placing a do-nothing word on screen 3 makes forgetting easy.

SUGGESTION: Study Chapter 7 of STARTING FORTH.

AN OPEN LETTER

Guilford 99'er UG

Dear Members:

It is with deep regret that I must announce that, effective with the July meeting I tender my resignation as Vice President of the Users' Group. I find that due to excessive and demanding personal obligations, I will not longer be able to effectively carry out the duties of the office. I will continue to assist George in getting the newsletter out on time and I will be glad to assist with an occasional program at our meetings. However, I simply do not have enough time to administer the office as I think that it should be administered.

I feel certain that you can find someone with more time to spare and who will be able to devote more time to planning programs and doing those things that are necessary to keep the Users' Group running smoothly.

Bob Carmany

FOR SALE

Dan Post has a 300/1200 baud external modem for sale. The asking price is $55 $60. Contact Dan at 621-2552 or leave a message on the ROS board.

ROLL YOUR OWN

By Bob Carmany

Have you ever heard the expression "roll your own"? Weel, besides cigarettes, it has found a new meaning when applied to our dedicated band of fanatics--the diehard TI user. The VIC 20 has long since vanashed from the scene and TI (or most anyone else in the commercial world) no longer considers the 99/4A to be a significant part of the market. The result is that, very often, we end up making our own hardware, cables, and other accessories. A couple of months ago, George Von Seth and I bought a couple of AVATEX 1200e modems. The price was just too good to resist. The only problem is that they come without cables! Well, we still couldn't resist the temptation and went ahead and ordered them anyway. The next step (after they arrived) was to sit down and construct a couple of cables so that out new-found prizes would work. Fortunately, the folks at AVATEX were more than friendly and a call to their "800" number led to a friendly chap who was more than willing to help us out. With a piece of paper in hand with scribbled pin-outs, George and I proceeded to "roll our own" cables. Let's see, pin 1 on the RS232 side went to pin 1 on the modem side---then you had to switch pins. . .

At any rate, some 20 minutes later, we had the cables completed and it was time for the proverbial "moment of truth". Everything was plugged in and, with our fingers crossed, we tried the modem. To our delight, everything worked just fine!

Another example is a fellow in the outback of Australia. An Opal miner by trade, Ron Kleinschafer of the Hunter Valley UG has "rolled his own" modem, Eprommer, and 3-slot P.E. Box. Do you want the plans? Ron will probably send them to you. All you have to do is write and ask. In fact, that is the case with most TIers.

What does this prove? Notning really except that there is no reason why ANY prudent person can make cables and other minor hardware items with a minimum of experience and effort. For projects of a greater magnitude, there are resources

available right here in our UG. Mack Jones has considerable experience with a soldering iron. In fact, sometimes you would think that he is an expert swordsman when you see him in action. Herman Geschwind is our resident expert "on just about everything". Others in the club can help you in areas of their particular expertise. WHY BE BASHFUL!!! If you have a question about hardware or software, ASK SOMEONE!! Someone in the UG can answer the question or probably has access to information on it. Remember, now that TI is no longer actively supporting the TI, your best source of information is your UG.