



# QB MONITOR

QB-99'ERS U.C. NEWSLETTER

101 8706  
07  
08

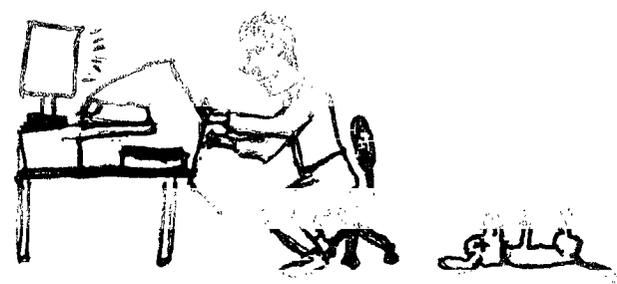
Summer 1987

The MONITOR is the Newsletter of the QB-99'ers User Group, is printed Sept. thru June and sent in exchange for other User Group Newsletters. Send Exchange Newsletter to Frank Cotty, Queensborough Community College, Bayside, NY 11364. Credit original sources.

The QB 99'ers meets the second Saturday of each month September through May, at Queensborough Community College, Bayside New York, room S225. See the calendar at the end for the dates

August 1987							September 1987						
S	M	T	W	T	F	S	S	M	T	W	T	F	S
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26

Are you receiving our newsletter regularly? If you check the top line of the address label on the envelope containing this newsletter the months listed are those we have received your newsletter. To continue to benefit from exchange with our group your group must send a newsletter within three months beginning September 1, 1987.



NO SPOT, ACCORDING TO THE COMPUTER  
IT'S NOT TIME FOR SOME LUNCH YET

Contents	Page
Editor's note .....	2
Library updates .....	2
Newsletter Editors	
Disk Controllers .....	16
WORTH .....	
Brain Teasers - Computer Quiz .....	2

Articles for the Sept issue must be in by Aug 10

# QB MONITOR ~ QB-99'er NEWSLETTER

## Editor's note

As we are now fully into summer there seems to be less time for indoor activities. Less time for our TIs. But let me assure you that the TI world has not stopped. New versions of Funlwriter have come out. The Funlwriter diskette has now been renamed Funnelweb. It will still contain the TI writer updates added by the McGoverns of Australia, but in addition the loaders for our most used programs are included. You can now load in through Funnelweb Masstrans, Dbase, E/A, DM1000 and cassette saved E/A programs. In addition Tony and Will have included DISKPATCH (DISKO), access to your c99 programs (REL3), and a user list of other programs can be made to be loaded through Funnelweb when you prepare a FW diskette for loading by running a ULINSTL program and saving the results to your new program diskette.

All of this programming is impressive and as Tony and Will state "...as a result of TI's misbegotten marketing policies, the capabilities of the TI-99/4A are only now being fully explored."

Full credit and information on payment to the authors of the fairware programs provided and/or suggested for use will be found in the documentation on a separate disk called FW-DOCS V3.5 now in our library with the FW V3.5 Diskette.

## Newsletter tidbits

Newsletters are coming in from all over the country and Canada. We are now exchanging with 50 TI User Groups and looking for more. If you have not been looking through the monthly newsletters you are missing out on some of the best computer entertainment, information and diversion around. Recent issues included a computer crossword puzzle, cartoons and characteratures of famous persona, instructions on adding 32K of memory to the console, a piggyback module project to combine often used modules into one pack, and lively discussion of the future in new computer hardware; there are offers for used TI Equipment, bargains on modules, new software offerings, new hardware offerings. In short, the newsletters provide the information outlet to the consumer that was taken away with the demise of general interest computer magazines.

Since the library was underutilized in the past we have set up a round-robin to give wheels to our newsletters. Contact Ed Machonis or Frank Cotty to be included in the program.

## Brain teasers answers

Answers to the "Brain Teasers" you found in the April issue of the "QB Monitor" are as follows:

11. 90 degrees in a right angle
12. 200 D. for P. G. in M.
13. 8 sides on a stop sign
14. 3 blind mice see how they run
15. 4 quarts in a gallon
16. 24 hours in a day
17. 1 wheel on a unicycle
18. 5 digits in a zip code
19. 57 Heinz varieties
20. 11 players on a football team
21. 1000 words that a picture is worth
22. 29 days in February in a leap year
23. 64 s. on a c.
24. 40 Days and nights of the great flood

Did anyone get #'s 12 and 23 yet?

The "Computer Quiz" answers are:

1. ENIAC stands for Electronic Numeral Integrator and Calculator
2. First to use punch cards? Jacquard 1752-1834 on weaving looms
3. First use of punchcards on computers?....Babbage 1791-1871
4. Commercial application of computers began? .....1953
5. First switching devices? mechanical relays, then vacuum tubes
6. National commission on Electronic Funds Transfer established?.....1974
7. First hand held computer? HP 65 cost \$795 1974 Hewlett-Packard
8. Meaning of LSI? Large scale integration. Chips made with these replaced the earlier integrated circuits (IC's) when they found that other electronic components could be added to the chip besides the diode memory arrays.
9. a. Illiac IV produced by?  
...Burroughs  
b. Star-10 by?....Control Data  
c. ASC by? ....Texas Instruments  
... (yes them)  
d. STARAN by? Goodyear Aerospace  
What do these have in common?  
... Each attempted to process information at the rate of 150 MIPS (million of instructions per second) and the costs were between 15 and 40 million dollars each to develop.

DISK CONTROLLERS - from TI to MYARC

right Jerry Coffey, January 1987

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 PEX 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 (2560K 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 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 substantial 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

DISK CONTROLLERS - Cont'd. Pg 2

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  $20 \times 39 = 780ms$ , almost four revolutions of the drive. The TI interlace lays the sectors down on a track in the order 075318642. 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 can be 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 format 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.

## DISK CONTROLLERS - Cont'd.

Pg 3

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 558s 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.

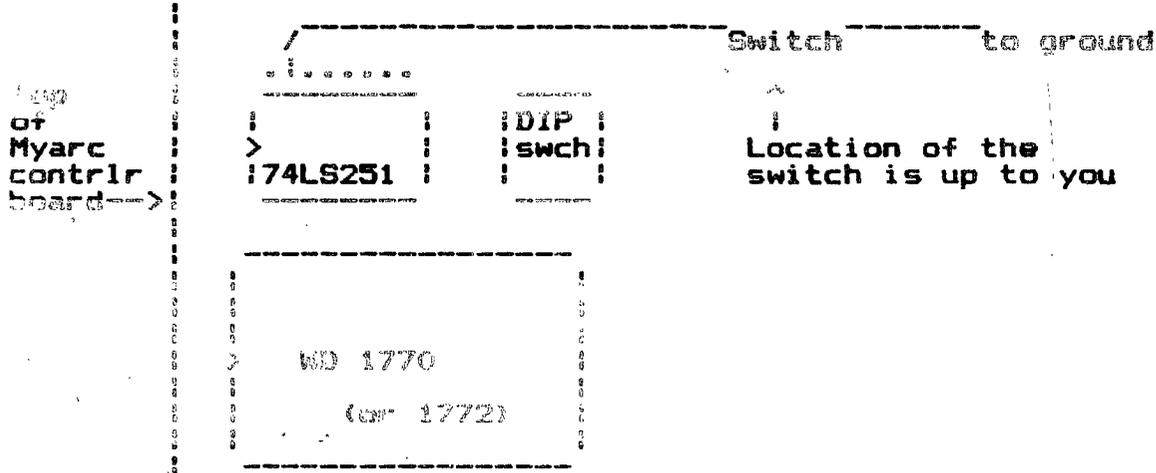
Sect/trk	Interlace	Pattern (dashed line is time available for head step)
9	4 *	0 7 5 3 1 8 6 4 2
9	2	0 5 1 6 2 7 3 8 4
18	5 **	0 11 4 8 15 1 12 5 9 16 2 13 6 10 17 3 14 7
18	4	0 9 5 14 1 10 6 15 2 11 7 16 3 12 8 17 4 13
18	3	0 6 12 1 7 15 2 8 14 3 9 15 4 10 16 5 11 17 none
16	5	0 13 10 7 4 1 14 11 8 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

Disk Controllers -- an Addendum March, 1987

I mentioned a "turbo" modification to lock out the "read after write" (write verify) routine usually performed by the controller. Here are the details:

Find the 74LS251 chip at the top center of the controller board, above the DIP switches and beside the large FDC chip (marked WD1770). Solder a wire from the number 2 pin of the 74LS251 through a switch to ground (e.g. the wide trace of the DIP switches or any trace connected to that wide trace). It looks about like this from the bottom (non-component side) of the board.



As always you proceed at your own risk. (One person has told me this did not work on his 40 track system, but I haven't verified that.) You can tell it is working if your controller writes as fast as it reads (normally the write takes twice as long).

The above article was downloaded from Bob & Bill's BBS, (BBBBS) Clinton, MD (301) 292-1482 - Jerry Coffey 74716,3525 can also be contacted thru this BBS

TI FORTH PART 2

by Hank Ellerman

Downloaded from TI-RALIEGH 919/833-3412

Now that you have had some time to play with the system and maybe become comfortable with FORTH, it's time to take the options you like to use and turn them into a fast loading BSAVE program. We will add a couple of words that I use all the time, PAGE and FREE. PAGE is nice to clear the screen and sets the cursor at upper left. FREE is a good one, it will tell you how much memory you have free. Maybe by now you have defined some of your own words.

Lets define the word PAGE:

```
: PAGE 0 0 GOTOXY CLS ;
```

Just as you see it on the above line. That's all there is to it. You have a a new word.

Lets define the word FREE:

```
: FREE SP HERE - . ;
```

Now you can see the amount of free memory anytime.

Now we have to find a place on a scr for these new words. You can pick a empty scr or add them to scr#3 later, because we have to update scr#3 any way. The only option you need to put these words in the vocabulary is -SYNONYMS. So load -SYNONYMS and define these words and see how they work.

Lets do the job now. BSAVE. The first thing is to put the backup disk in drive 1. Better yet make another copy of the backup disk, because we are going to write on this disk and on some of the scrs that you may want to look at in the future. Now we have a 3rd copy. The first is the original with the write protection tab on it. The second is the backup that has scr72 corrected "the bug moved and set for PIO". Now lets work the third.

Put it in drive 1 and load FORTH. If you are already in FORTH, go to command mode and type:

```
TEXT COLD
press enter
```

This will reboot the FORTH system to make sure we start from scratch.

Now we have the flashing cursor under the word TI FORTH.

Lets load our options. The ones you like to use or these; types

```
-GRAPH -DUMP -VDPNODES -COPY -PRINT
-BSAVE -64SUPPORT
press enter
```

The -64SUPPORT maybe -EDITOR which one you like better. REMEMBER you can not use both.

In my system I have loaded everything but -EDITOR -TRACE -ASSEMBLER -CRU -FLOAT, to save memory. They can be loaded after the BSAVE is in memory. After the ok is displayed and the cursor is flashing again we are ready to do our BSAVE type:

```
* TASK 51 BSAVE
press enter:
```

( NOTE the first character in the line is the apostrophe)

Now we have our fast loading program on disk. We just have to tell FORTH where it is at. Type:

```
EMPTY-BUFFERS 3 EDIT
press enter
```

Now we have to change scr#3 to tell FORTH where the BSAVE is at and also put in a little more information.

SCREEN#3 is the welcome scr. If you are using the 40 column editor you have line numbers, if -64SUPPORT you have no line numbers and have to count them for yourself. This is what scr#3 should look like after all changes.

```
0( WELCOME SCREEN) BASE->R HEX 10 SYSTEM
( Clears Screen)
10 0 GOTOXY ." TI FORTH BOOTING" CR 10
83C2 C! ( QUIT off)
2DECIMAL 51 BLOAD 16 SYSTEM MENU
31 VDPMODE ! ( Tells FORTH you'r in TEXT
mode)
40 DISK_LO ! ( Allows EDIT/COPY on all
SCREENS)
589 DISK_HI ! ( Sets highest scr number)
689 DISK_SIZE ! ( Set single sided disk)
7: FREE SP HERE - . ; ( Free memory)
8: PAGE 0 0 GOTOXY CLS ; ( Clear screen)
9 FREE
10
11
12
13
14
15R->BASE
```

Now don't get the line numbers mixed up with the text, and don't type in the line numbers. The line numbers are in column 1, they are just there for reference.

Now hold down the function key and press 9. Back to the command mode. Type:

```
FLUSH press enter
```

Our new scr is written and we can try it out. Type:

```
TEXT COLD press enter
```

That was quick, we have just loaded everything you saved with BSAVE.

Try it again, this time turn the computer off and start from the master screen. FAST ha?

Now all you need is the program "CLONE" and you can copy ANYTHING in TI language. You can do it also with the FORTH command "FORTH-COPY".

Most of this information I read from MILLERS GRAPHICS, "The Smart Programmer", 1475 W. Cypress Ave. San Dimas, CA 91773. \$12.50 a year., It is a monthly publication, maybe.

Lets see some of your programs now!  
HANK

WHY FORTH? - PART 2  
-----

Downloaded from TI-RALIEBH 919/833-3412

An article on the pro's and con's of the Forth language, from the standpoint of someone who does not use it.

-By Jim Ness ----- Nov. 1984

From chapter one, you must have got the feeling that I think Forth is a waste of time. I do. For me, that is.

If you are familiar with assembly language programming, and are comfortable using it, then Forth is a waste of your time. You can come up with assembly language routines that are cleaner and faster than the same in Forth, and with about the same amount of effort.

But if you haven't yet become familiar with assembly language, Forth has a number of routines that are not possible in Basic or Xbasic, that you may wish to use.

Forth gives you such better access to the video display area of your computer. You can use all 4 display modes with Forth (The standard graphics mode, 40 column text mode, Multicolor mode, and Bit Graphics mode).

You also have more control of peripherals. Faster file access, ability to use modes of file handling not available to Basic programmers.

Much faster arithmetic routines, both floating point and 8/16/32 bit numbers. Faster loops, for instance count to

10,000 in a loop in about 5 seconds.

Even program in a Forth version of assembly language. It runs at Forth speeds, but uses a vernacular similar to real assembly language.

In short, Forth allows you to do anything that CAN be done on your machine, anything that you could do in assembly language. The difference is just speed.

If you don't feel comfortable with assembly language, but want to get more out of your machine, try Forth. Don't expect to learn Forth as fast as you learned Basic but with a little work, you should be able to do some unbelievable things.

If you ARE comfortable with assembly language, you may want to play around with Forth anyway, just to say you did it. Then you can look down your nose at Forth, just like I do ("Forth? Ah, that's just for amateurs who can't handle Assemble Language!").

Pay attention to Hank's tutorials. You will need them, because the TI manual stinks. Remember that TI released this product after they announced their withdrawal from the market. The product was not yet finished. So the software has a couple of bugs, and the manual is not a friendly piece of literature. There is a disclaimer at the front of the manual.

If you are really interested in Forth, there is also an independent product out that was produced by WYCOVE. I think it is supposed to be a good product.  
-JIM NESS