LA 99ers TopIcs

Vol 11 Number 11 NOV,1992

CONTENTS

Remember Next Meeting

Wed Nov 11 7:30pm

Pacific Heritage Bank

BE THERE!

GIVE THANKS ON THANKSGIVING

NEXT MONTHS MEETING ON

WED. DEC 9th

RAMBLIN'
THOUGHTS FROM THE PRESIDENT

OUR OCTOBER,1992 MEETING WENT WELL AND WAS AGAIN SAVED BY EARL RAGUSE'S EVER CONVENIENT SUPPLY OF DEMO DISKS WHICH SEEMS TO BE ALWAYS IN HIS CAR. THERE WERE ABOUT 9 PERSONS WHO SHOWED UP. HE DEMONSTRATED A FORTH FRACTALS PROGRAM DRAWING CARTOID DESIGNS.

PERSONAL NEWS ON MIDI:

AIN'T GOING TOO WELL SWEETHEART. TOO MUCH CLUMSEY TYPING AND NOT ENOUGH THINK'N BEFORE EXECUTION. LAST MONTH I STARTED TO TEXT CODE INTO SNF FILES, PEACE INT HE VALLY. WELL AFTER 100 HOURS AT THE COMPUTER IT STILL AIN'T DONE. I THNK I MADE EVERY MIXTAKE I POSSIBLY COULD. YOU HAVE TO BE VERY CAREFUL WHEN USING THE COPY FUNCTION, I WUZ NOT. I THINK I TRIED TO COPY LINES THAT WERE'NT THERE AND TOLD THE TI EDITOR TO COPY IN THE WRONG PLACE THE WRONG BLOCK OF TEXT TO THE WRONG LINES. BUT I AM STILL HAVING FUN??? PLACING THE WRONG SYNTAX FOR FLAT WAS A BIG MISTAKE, SO WHEN IN DOUBT READ THE DOCUMENTATION. ALLTHO JIM PETERSON WARNED ME OF THE FACT THAT I WAS USING THE WRONG SYNTAX BY A FRIENDLY LETTER.

WE NEED A BOARD MEETING DESPERATELY...

ELECTIONS COMING UP FASTER THAN YOU CAN SAY "JACK ROBINSON". SO IF YOU WANT US TO HAVE AN EXCITING FUTURE AS WE HAVE HAD AN EXCITING PAST, AS THEY SAY ; "BE THERE OR BE SQUARE". WE NEED TO AMMEND THE BY LAWS IF WE WISH TO STAY IN THE COMPUTER WORLD IN GENERAL. WILL LET YOU KNOW MORE LATTER ABOUT THAT.

THE LINES AT THE SIDE THAT APPEAR, CANNOT HIDE THE FACT, THAT THIS LETTERS', END CAME AND WENT AS FAST AS THIS YEAR.

YOUR PRESIDENT,

CONVERSION AND OTHER FILE FUN

XB MISCELANY #15 by Earl Raguse

I wondered what would happen if I used my previously mentioned (Feb 92 ROM) CONVERT program to convert a Display Variable 80 (DV80) file to an Internal Fixed 20 (IF20) file. I deliberately wrote a DV80 file with quite a bunch of long words and a lot of short words. I wanted to see what happened when the file was manipulated by the CONVERT pgm. I first made the file with an 80 column width, then 40 columns. That file is included here as TESTFILE. Have you ever tried to write sensible statements using only 1, 2 or 3 letter words. You can't even say "See Spot run". Its a lot easier when you use longer words, at least for me it is.

What I intended to do was to convert the DV80 file to a IF20 file, but, since one can't look at a IF20 file with TIW/FW, I then intended to convert it back to DV80 to see what havoc I had wreaked.

Ok, so what happened when I tried to convert the file? Well, for one thing, I found out, the hard way, that you can not use XB's LINPUT statement to read Internal files, its just plain not allowed, I don't know why, but that is what the manual says, and its right, you can't.

Well I couldn't make the latter conversion. That's when I started reading and found out about the LINPUT limitation. So I changed LINPUT to INPUT instead and used an IF40 file, that seemed to work, sort of, the trouble now was that parts of the original 40 column file were lost when it got back to DV80.

That turned out to be because I had put commas in the original DV80 file. When INPUT sees the comma it stops reading, because a comma is used as a record separator. So why did it not then read the part after the comma on the next read, like in a DATA statement. I don't know, but it read the next record. In DV and DF files, each line is a record. If a sentence is longer than one line, it is a multiple record. If you want to read two variables in one statement, you must provide two variables to put them in. That is impractical here, one never knows how many commas there might be in the next file. Also what would happen if there are no commas? The best solution is to get rid of the commas, or stick to Display type files and use LINPUT.

What happens to the record length? You will remember that with fixed length records the controller will pad out the record if it is not of max length. But what happens if the input file record length is longer than the output file record length. I thought it would just truncate the record, but no, it does not, if it is a Display type file, it just writes the overflow as another record, nothing is lost. A word is divided whenever the specified record length is reached, just like in an XBASIC program listing on the screen. However, if it is an Internal type file, and the record is too long, it crashes.

I originally did not think this would happen, so I tried to avoid a problem by writing into the program a test for input records longer than the output record length. If found, the record was broken into parts of output record length, without regard as to whether it was breaking up words. Then these were written to the output file as two records. Well this did not seem to work too well, probably because, I was trying to outwit the disk controller. In the end I did nothing, and just let the controller worry about it, except you must use common sense with Internal files.

In addition to converting files back to DV80 to look at them, I also used Disk Utilies, and I also modified the original CONVERT program allow the option of Converting or Reading any type file.

I got some amazing, to me at least, results. All file types, with the same information, look almost alike, there are some minor differences in appearance. The following is my observations:

1) Display Fixed files pad out the record length with "Hex 20" (spaces).
2) Internal Fixed file pad with "Hex 30" (0's).
3) Display Variable files start a record with a length byte.
4) Internal Variable files seem to have two length bytes. I do not understand the function of the second one. I can not find out what it does. Cont'd Pg 3

XB MISCELANY #15 by Earl Raguse

5) Variable type files will break up records which are too long and write additional records.
6) Internal type files will not tolerate records that are too long.
7) Variable type files have an End Of File (EOF) marker, which is FF.
8) Fixed length files do not have an EOF marker.

If anyone knows more about this, or if I am in error somewhere, please set me straight.

Now the reason I was looking at the files with DSKU was to find out why I sometimes could not load the DV80 files made with CONVERT, or any XB program I wrote for that matter, into FW. Everything appears normal, until it comes to the place where the file is supposed to suddenly appear on the screen. At that point FW hangs up and I have to exit the program to regain control.

My guess is that FW can not find a marker or something that tells it that it is time to stop reading and start displaying. I think that XB and the disk controller DSR should be taking care of all this. Anyway, even with DSKU, I did not find the answers. All files had the required EOF marker in the right place. The working files look excatly the same to me as the non-working ones. The problem still plagues me occasionally, if anybody has ideas, please contact me.

I had originally not planned to publish the CONVERT program, because it is a rather long program, but since I have modified it so it is also an ANY TYPE file reader, it is much more useful, therefore, I am going to publish it next month as, as a type in program. I will also explain how it works. You may also get it from me on disk, just ask. You will probably learn more if you type it in and try understand each statement.

TI WRITER TIP
From Nutmeg News

Here is a tip for TI-Writer users. Some of you are probably familiar with the Replace String function, however here are a few tricks to its use that can increase its effectiveness for you.

1. Before using RS, make sure to turn off the word wrap mode. Do this by pressing function 0 until you get a hollow cursor. This will prevent TIW from reformatting the whole document. The only time you would want this type of reformatting is when you are replacing a short string with a much longer one, e.g. FW replaced by Funnelweb Farm Utility Disk. If you must leave word wrap on because of this, remember that each paragraph must end in a carriage return, and any line that has special spacings or indentations should end in a carriage return as well.

Try this on something like a doc file from a fairware disk. Load in a large file, press function 9, and then type RS enter. Now type / a / q / and press enter. Yes, there are spaces before and after the letters a and q, and both letters should be framed by slashes as shown. TIW will find the first occurence of a, and then will ask Yes No All Stop. Typing Y and enter will change just this "a" to a "q" if you are in fixed mode (hollow cursor), however if you are in word wrap mode TIW will replace the "a" and then reformat the rest of the paragraph. Take this one step further and type A enter. This will change every occurrence of "a" to "q". Time this and then hit function 0 and change "q" back to "a" by changing the RS string to / q / a /. You should find the fixed mode to be many times faster than the word wrap mode.

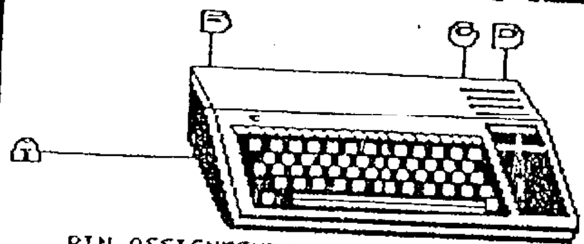
2. RS is sensitive to columns, meaning that you do not have to have every word in the document checked if you want to change "2]" to "2)". Use this string to replace a string occurring between columns 8 and 10:

8 10 /]/)/

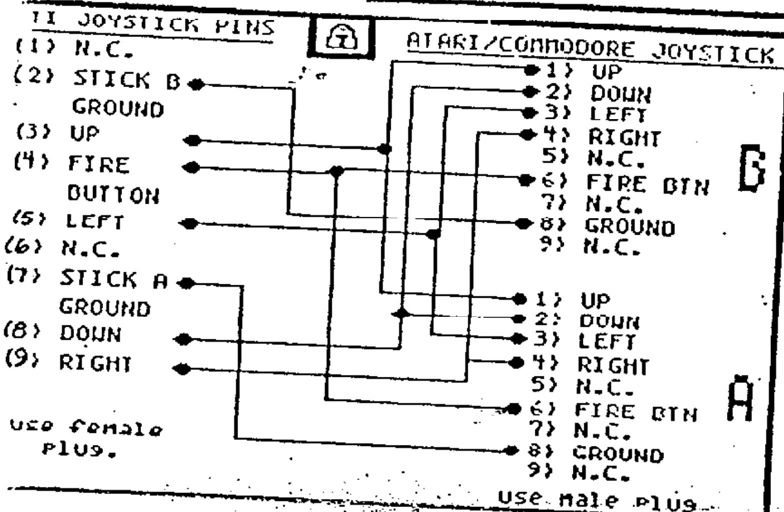
Remember, always back up your files and work with the backup. Doing some of these exercises can be fun, but only if there is no risk involved. Until next time, keep well and enjoy.

From
TI RUG

PIN ASSIGNMENTS



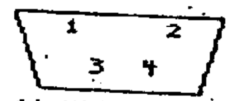
PIN ASSIGNMENTS APPEAR AS THEY WOULD IF YOU LOOKED DIRECTLY INTO THE PLUGS.



FORT USER GROUP

FEBRUARY, 1987

POWER SUPPLY PLUG

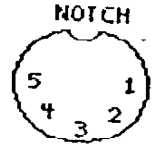


1) NOT USED
2) 8 VOLTS AC
3) 16 VOLTS AC
4) GROUND

MODULATOR PLUG

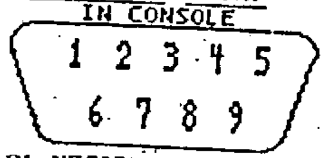
1) RED
2) TRANSPARENT
3) WIRE WRAP (not a ground)
4) BLACK (ground)
5) WHITE or YELLOW

VIDEO/SOUND PLUG

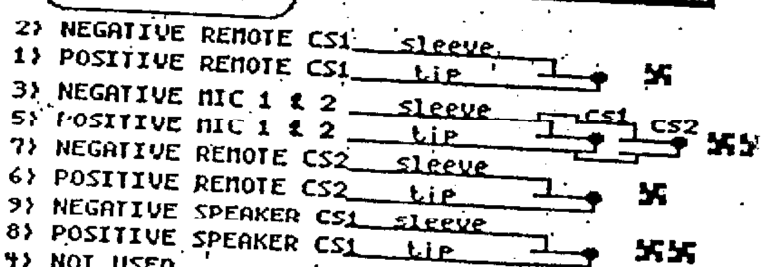


1) +12 volt
2) VIDEO OUT
3) SHIELD
4) GROUND
5) SOUND OUT

CASSETTE PORT IN CONSOLE

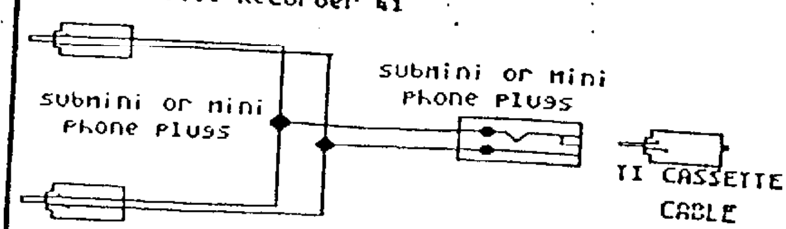


CASSETTE CABLE BELOW



* SMALL REMOTE PLUG 3/32"
** PHONE PLUG 1/8"

Use Y splitter to record to two tapes



PAGE PRO PAGE COMPOSER

BY DEANNA SHERIDAN
NORTHCOAST 99'ERS

I have had some difficulty in getting into this program, so please bear with me if I ramble. First of all, the way to REALLY learn a program is to either do a demo at the monthly meeting or a review for the newsletter. They are educational experiences, to say the least!

Asgard's flyer on this software states that combined with PagePro, you can now create and print multi-page documents in either landscape (sideways) or portraite mode, with columns of 68, 80 or 128 wide. Reading the documentation gave me very little idea of how this was to be accomplished. So, I decided to load and print the demo files to see what was really happening. An extra paragraph explaining the types of files Page Composer will handle and why would have cleared up the matter in short order

First to use Page Composer effectively, you definitely need Page Pro, lots of graphics and lots of fonts and whatever else you want to enhance your document, such as borders, etc.

A two-column portraite document from Page Composer is actually TWO full-page documents from Page Pro printed side by side in high resolution. In other words, the documents are "shrunk" to fit the page. As you can see from this page, the one-character high font is reduced to what would normally be called "condensed" print. The printing is done in a manner that does not distort graphics by elongating them as some high-density printing does.

You will definitely want a version of Page Pro that will save a Page Pro page as a picture. This "seems" to be the only type of file that Page Composer will load. So it takes awhile (or at least it did for me) to catch on to the fact that I would have to prepare each part of my page in PagePro, save it as a picture, and then place the pictures where I want them in Composer to assemble the page or pages. You cannot type text on the screen in the Editor...only place pre-formatted files in the designated areas.

The program uses a "Mac"-type point and shoot interface that supports either keyboard, joysticks or mouse. I have found that the joystick option has very slow response and I have to click on some of the options several times before they respond. Also, the joystick must be in "exactly" the right place to get a response. Keyboard is a little slower to move around, and the response "enter" key is almost as bad. I would hope that a mouse user has better luck

A file must be named and opened before anything else can be accomplished. After that, you load your pictures to print, either full pages or individual as prepared in Page Pro. Depdening on the complexity of the page, you should "draft" a mockup of where the pictures are to be placed.

After the pictures have been placed, the file can be printed or saved. The other menu options are "next page", "previous page" and "go to". The docs say you can have up to 999 pages in a document. However, due to the large size of a full-page picture file from Page Pro, and two of these files to a page, you had better have a hard disk for those files.

We complain and complian that our TI's won't accomplish this or that other computers can do. I think we expect a lot from our little 2 megahertz system with about 48K of working memory. When we do get programs that bring us into that IBM world, we complain again that they are too hard or too slow. I think the finished product from this program is extremely professional. However, I need to stress that it will take a lot of patience and time to create these pages.

It helps a lot to have either a ram disk or a hard drive because you have to switch between Page Pro, Funnelweb, and Composer frequently. If you don't have your favorite graphic or font in Page Pro format, you will have to take time to do some converting. If you want special headlines with additional fonts (Page Pro only supports two at a time), you will need to do some creative work in TI-Artist and then convert to Page Pro Format.

Lastly, because the different effects in Composer are created through resolution rather than size of fonts, printing is extremely slow. The documentation states that the speed of Composer is tied to the disk drives and printer, and the only way to speed it up is to use a faster drive and printer. It also stated that running Page Composer from a RAM-disk can reduce the amount of time it takes to print a single-page 960 resolution portrait document with 10 pictures from 28 minutes to 5 minutes. Well, I put my files on my ram disk and used my laser jet for printing, and found that it took a lot more than 5 minutes to print the left side of this article so that I could get a good idea of my placement. Therefore, if the finished product isn't what you expect the first time around, it will take time to reformat and reprint. Patience is the key.

Also, if after you print the first time, as I did, and find that you prefer to use a different resolution, you will need to create an entirely new file, as you can only choose the resolution at the time you create the file and cannot be changed after it is saved and called back to edit. A separate utility file is included entitled "Clipix" which will clip graphics from large files for use with Page Pro. Most of the "clipping" programs will only save a graphic the size of the screen area, or the largest size that TI-Artist will support. With Clipix, you can page down and get pictures that were too large before. However, the picture must be in Page Pro format BEFORE it can be clipped. That means that it was have to previously been converted from another format, usually PCX or MAC by another conversion program. It would have been much more convenient if it would have loaded a PCX or MAC format and saved it format and saved it to Page Pro.

DISK CONTROLLERS from TI to MYARC.
Copyright Jerry Coffey Jan '87
Reprinted from CPUG April '90

The views expressed in this article reflect the authors personal experience with the 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 TI 99 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 could support three double sided 40 track drives (540K). Corcomp introduced their four double density system (1440K), followed by Myarc 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 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 devided 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 is 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 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 the disk 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 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 the 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 the drive makers requires only 6250 bytes per track and includes several hundred additional "buffer" bytes to compensate for difference 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 that there are about 25 thousand magnetic pulses (bits) passing beneath the head every second. In single density format, the majority of these pulses are timing or filler bits -- in double density, many of the timing bits are surpressed 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 into the target track, verify the track number (written in the format operation), detect the sector identifier as it flies past, then immedeiately read 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/second = 2048 bits/sector into 250Kbits/second). . read a 125 sector file in one second? Well, first many of these bits are not data bits, they are overhead to keep things synchronized and allow for timing variation between drivers. 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 time available for reading data. By the time one sector is safely tucked away in the 32K card, several sectors have passed by the drives 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 head settled down for some serious (250K bps) reading.

INTERLACE AND HEAD STEP TIMES

Life was simple with the TI controller. Both the interlace and the head step time were locked into the controllers PROM (that's the programmable chip that contains the control programs for the card). The head step time is the built-in delay between the 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 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 a 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 the VDP 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 provides a foretaste of the speed that MYARC would achieve with its double density controller.

The MYARC controller bypasses the VDP RAM to load directly to CPU RAM. This technique coupled with the buffer RAM chip on the controller card provideda quantum jump in disk I/O speed. The MYARC card reads the TI single density interlace at 11.25 sectors per second (the same as Corcomp) and reads the CC 18 sector/track interlace at 18 sectors/secon (the same speed Corcomp reads itsMANAGER program), but this is only the beginning. Since the hardware emptiesits second 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 Corcomp controller. The MYARC 16 sector format can be read at interlace "3", 26.67 sectors/second -- 3 times as fast as the TI 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, intelace "3" reads the track in three 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, intelace "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 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 controllor'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 the 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 zero. It begins at byte 56 leaving only 200 bytes of 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 that we have new code in the disk controler 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 the 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 the 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 the sector editor.) Read/write operations from XB or TI Write 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 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 interlacing 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 laid down. Look and listen for "retries" -- when the sector number pauses with the head seek noise. Use the best disk 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 another revolution forced by having sectors 17 and 0 adjacent on the disk.

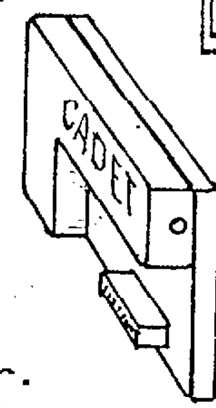
When you try this with 80 track drives, don't be surprised if the results are different. The time required for the head to settle into a wide standard track may not be adequate to get it reading properly for the narrow tracks on the quad drive. Such subtleties as erase delays and disk quality are also 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 noticable 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 to the "turbo" option on the Corcomp controller and should be used only after testing.

CADET
CONSOLE EXPANSION

TI BRISBANE USER GROUP
P.O. BOX 3051
CLONTARF M.D.C.
QLD AUST 4019.



Plugs into I/O port on right of the console.
Compatable with all software from tapes and modules.
Features:-
PRINTER PORT. Drive a printer directly from Basics or print text files.
SOFTWARE IN ROM:
Word Processor like the original TI-Writer. Type and print letters, assignments etc.
Text Formatter like original TI version. Add "class" to your printed text.
Tape Games Loader to load and run specially taped assembly games.
32K MEMORY EXPANSION for use by the above software.
ADDITIONAL 32K RAM divided into two devices to store text for later reuse. Supercap backup maintains power to memory for weeks.
I/O EXTENSION. Attach a speech synthesizer or ramdisk. Ramdisk requires optional adaptor card.

Limited number available to order at $100 each.
Ramdisk adaptor card $20.
Contact Col Christensen on (07)2847783.

128K MEMORY SYSTEM from ASGARD

This information is excerpted from Micropendium Oct 1992
The Asgard Memory System (AMS) is designed to be used exclusively as memory for programs and data. The AMS functions as a 32k card with the- standard TI994A software when installed in the PEB box. According to Asgard it will not conflict with any hard or floppy disk controllers and is compatible with some, but not all, RAMdisks. It is not guaranteed to work with the Myark or CorComp RAMdisks or the TI , CorComp or Myark 32k cards. No problems have been encounterd with Horizon RAMdisks, according to Chris Bobbitt.

Programs designed to work with the card can access up to 128k of CPU memory simply and with a minimum of restrictions on program design. Memory can be banked within a few clock cycles anywhere within the standard 32k memory space.

The AMS requires a TI994/A with PEB and a disk system. It is compatible with all disk controllers and all video cards as well as virtually all other cards for the 4A. Cost is $119.95 + $10 S&H. Delivery 4-6 wks. Programmers may receive a free packet containing programming information on request.

LA 99er's OFFICERS

PRESIDENT	Edgar May	310 644 6241
V.PRESIDENT	Earl Raguse	714 847 5875
Treasurer	Al Whiteman	310 538 1953
Secretary	Bob Wheeler.	310 378 5092
Editor	Joe Fierstein	310 377 9834
Membership	Hal Jeffery	310 372 7795
Equipment	Gail Fair	310 326 6660
Program	Steve Mehr	805 379 2937
Market Place	Fred Moore	310 670 4293
User Group	Fred Moore	310 670 4293
Library	Sy Silver	310 370 2145

LA 99ers
A TI User Grp
7730 Emerson Ave
Los Angeles,Co. 90045

* First Class *

Exchng * Sep 92
DALLAS TI COMPUTER GROUP
P. O. Box 29863
Dallas TX 75229