

047
8704
Rocky

ROCKY MOUNTAIN 99ERS

TI-99-4A USERS GROUP

8248 PEARL STREET

THORNTON, COLORADO 80229

PHONE
303-289-7667

24 HOUR BBS
303-288-3692

COMPUTER FAIR: MAY 2 @ 3
ARAPAHOE COUNTY FAIRGROUNDS
DOORS OPEN AT 9:00 A.M.
CLOSE AT 5:00 P.M.
DOOR CHARGE 2.00 PER DAY
NEW TI SOFTWARE DEMO'S

IF YOU HAVE ANYTHING THAT YOU WANT TO SELL, BRING IT WITH YOU, WE WILL SELL IT FOR YOU ON THE CONSIGNMENT TABLE. IF ONLY ONE CART OR CABLE, WE WILL HAVE OUR FULL LIBRARY AT THE FAIR AND WILL BE SELLING DISKS AT 2.00. IF YOU WOULD LIKE TO HELP US AT THE FAIR, YOU MAY DO SO BY CALLING MARK PAYNE.

APRIL MEETING

APRIL 14, 1987

2205 SOUTH MILWAUKEE

7:00 P.M.

DONT FORGET TO RENEW YOUR
MEMBERSHIPS

REGULAR MEMBERSHIP
24.00 PER YEAR
ADDITIONAL FAMILY MBR'S.
6.00 PER YEAR
BBS ONLY MEMBERS
12.00 PER YEAR

GENEVE REVIEW !!

THE FOLLOWING is what we hope will be a continuing ARTICLE on the new GENEVE.

As a users group we have had the great fortune to get a GENEVE SYSTEM that we were able to demo in February. The demo took place at Joe Millers Offices here in Denver. Not having a real good manual to learn how to operate the system from, we encountered several problems with trying to get it to run properly. Not the GENEVE's fault just the fact that all material was on disk and we didn't have the time to run it. But with what we were able to run it was very impressive.

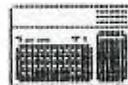
Before the system was sent back to Myarc, I was able to try it out for a day and I was duly impressed with the system. First , the negative things that were found.

A. The manual for the Myarc DOS was hard to understand, but rather than throw that on Myarc's back without comparing, I decided to try out some of the IBM PC's at work to get a feel for DOS. I found that running the IBM DOS was just as involved as Myarc's. A TI-99/4A has a limited DOS and after a while I found that although DOS is involved (as far as the learning process), it also grants the user a greater amount of control over the operating system. For your information the Myarc DOS is supposed to be the same as IBM DOS 2.1, so if you have had some experience using DOS 2.1 then you will be one up on the average 99/4A user who has not.

B. At the demo I heard that tha users could not get into the 512K that comes on the card. I called Myarc and was told that it was there but due to a poorly written test manual (that has since been updated), the explanation for accessing the 512K was not available. A real good way to show off their units huh!

C. The unit that we had did not have a RGB cable so we were not able to check out these features either. It will be included with the new units for sale.

D. And the final problem was that the users were unable to get into the 80 column TI-WRITER that was included as a demo with the system. The reason for this was the manual again. It said to type in EDIT in uppercase but instead it should have been lowercase. I tried this and it worked fine.

*NOW FOR THE GOOD STUFF !!!*

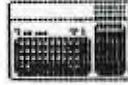
A. The above mentioned 80 column mode for TI-WRITER is as good as any 80 column screen that I've seen on the IBM. Because of the improved speed of the processor I was able to access my files quicker and there was absolutely no lag time in using any of the EDIT and SAVE functions.

B. A big surprise to me was when I was able to access a version of the terminal emulator program. It looked about the same as FAST-TERM.

C. The unit came with a graphic demo that was easily as good as anything I have seen on the AMIGA, plus a little faster in run speed.

D. The GENEVE also has the capability to run any program saved to disk by the GRAM KRAKER. So if you have a KRAKER at home and you are thinking of getting a GENEVE you are ahead of the game for running your cart based software. The unit will also include directions on saving your carts to disk.

SENEVE REVIEW "CONT"



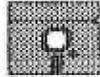
E. For those of you who are interested in the faster run time of software, I can say from experience that there is a difference. I decided to load in some of my wifes business software (YU-CAN), written in EXB to get a good idea of how EXB on the Myarc system would run. I was seeing file sorts that usually took 10 min. to run being ran in half of that time. There were some minor changes that had to be made to the screen color commands but other than that the software ran fine.

In the future I hope to include more information on this wonderful machine which two members of our group have already ordered (not including myself). There is alot that I have left out, but that will be included in future articles.

Until then, keep on computing.....

Mike Miller

BULLETIN BOARD NEWS ::::



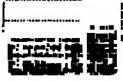
The Rocky Mountain 99ERS' BBS is growing in popularity, as we continue to advertise nationally. We have had Tler's calling from Kentucky, Massachusetts, Texas, California, etc. We are just overwhelmed with the response from one MICROpendium add. We will continue to keep the ball rolling and make sure that we can keep this alive and running strong. I was told by a caller from Texas that he was glad to be able to contact a board in Denver finally as he has tried to call the Front-Range board several times but was unable to get thru. I feel that this is a good sign and will, as your SYSOP ,keep on top of updating these people from out-of-state groups so we can expand our knowledge of new software and some good programming available to us from our counterparts across the good ol' USA. We have had a good bunch of our members logging onto the board, but still not all of you are active as you once were.

In case you haven't been on the board lately we have a new disk full of downloads for you. Please take advantage of these as we increase our libraries, so should you. We are looking to propose a new set-up for the board, IF this Fair makes us the funds. We are considering utilizing a Myarc hard drive controller card with a 20meg hard drive. This will free up the system we now have and we will be able to give the library the support it needs. The new system will cost us quite a chunk of money but we will be able to run a more desirable board so in turn this will give us national communication ability. We will also have to implement a new BBS program as this one is quite limited.

Any one having ideas in any of these areas please drop me a line as to what you want in YOUR BBS. What features would make your log-on life easier. I want our BBS to be the best, and I don't see why it can't be.

SYSOP

TIC-TALK TECH



Dear Editor,

In the February TIC TALK you printed a diagram for connecting Atari/Commodore joysticks to our computer. While it may not matter, there are diodes in the TI joystick that must do something or they would not be there. I built a converter to use one joystick, switchable to either A or B inputs, as desired.

I built mine in the following manner, Obtain : 1-female and 1-male 9 pin "D" subminiature connector, (Radio Shack #276-1537Ĕ-1538; 2-one and a half inch spacers, 2- one and three quarters inch 4-40 or 6-32 machine screws and nuts,(if 6-32 screws are used, it may be necessary to enlarge the holes in the connectors.) 1-micromini SPDT toggle switch. The smaller the better.

CONNECT AS FOLLOWS::

MALE CONNECTOR	THRU	FEMALE CONNECTOR
PIN 1-----	<DIODE	-----PIN 3
PIN 2-----	<DIODE	-----PIN 8
PIN 3-----	<DIODE	-----PIN 5
PIN 4-----	<DIODE	-----PIN 9
PIN 5 NO CONNECTION		
PIN 6-----	<DIODE	-----PIN 4
PIN 7 NO CONNECTION		
PIN 8-----	CT* /	-----PIN 2
PIN 9 NO CONNECTION	\	-----PIN 7

*CT Common terminal of switch.

Note: The banded end of the diodes (cathode) must be toward the male connector. The switch I used had wire terminals so it was easy to use them to support the switch between the connectors.

The connectors are mounted on the screws with the spacers between them, solder connector terminals facing each other. Solder in the diodes and the switch. Plug non-TI joystick into the male connector and plug the female connector into the console. Depending upon which way the switch is positioned you will control the A or B circuit in the computer.

I have used this for two or three years with a WICO command control joystick with no problems whatsoever. If two controllers are needed this will not work.

DAVE SWARTZ
MEMBER RM99ERS

THANKS DAVE!! PRESIDENT



CONFESSIONS OF AN EX-PIRATE
from POMONA VALLEY UG
(edited by Steven DeGeare)

Can you remember way back, when you first bought your computer? You spent a lot of your hard earned money on all these real neat and expensive pieces of hardware. It was great! Lights came on, things made noise, disk drives grunched ... But something was missing ... SOFTWARE!!!

NO big deal. Just run down to your computer store and spend some more money... "WHATINTHEHECKDOYOU MEAN \$99.95 for TI-WRITER? What do you have for under \$40.00? MUNCHMAN, PARSEC, BLASTO, AND TEACH YOURSELF X-BASIC. YOU HAVEGOTTOBEKIDDING!!!"

So you go out and make some friends with questionable reputations. One of them walks up to you and hands you a disk saying, "HAVE I GOT SOMETHING FOR YOU -- DON'T PASS THIS ON TO ANYONE ELSE!"

You run home and crank up your system and catalog it ... "Hmmm, this isn't too bad; couple of public domain programs and ... WHAT'S THIS BUCKROGERS???? How did he get this?"

So you quickly make a back-up copy and return the disk back to your friend. Then you say the magic words, "ANYTHING ELSE?" The next time he gives you 5 disks full of the same great stuff.

A friend stops by your place and complains about the high price of SOFTWARE ... You say, "NO problem!! Have I got something for you. Waddaya want? I got everything!" SOUND FAMILIAR?? All of us has at one time or another done this to some degree. After all, what is wrong with sharing a program with a friend???

The SOFTWARE INDUSTRY has a name for this --"PIRACY"-- Why don't we call it what it really is --THIEVERY-- Stealing SOFTWARE is no different than wading into the store and taking it off the shelf, leaving without paying for it. Except that it is harder for the SOFTWARE authors to get you for stealing his or her product.

If the programmer wanted to give their programs away, they would not have put price tags on them!!!

There are few more backlashes from the act of "PIRACY" (stealing). COPY PROTECTED SOFTWARE and lack of new product development. First let us look at copy protecting. Personally, I would rather see a programmer to spend more time on finding ways to get more out of my machine than how to guard his or her latest masterpiece. Just so they can make a few well-earned bucks.

Most protection schemes are either broken or bypassed within a week or two of the release of a new product. And protection schemes also make your drives work harder (LISTEN WHEN A PROTECTED PROGRAM IS LOADING), thus causing unnecessary wear and tear on your equipment.

How about making back-ups of SOFTWARE you did purchase? Forget it! And for some real fun, try installing protected SOFTWARE on a hard disk or a RAM disk. You CAN'T! Or trash your original (and only) program disk. You need that mail list or report finished by the end of the week? SORRY, 3 to 6 weeks wait. Plus you must send in your original disk. Pity the poor author who doesn't copy protect. WELL, he or she will get famous, not rich, just famous. These are just a few of the benefits of "PIRACY" stealing copyrighted SOFTWARE.

DOES anyone realize just how important new product development is for the continued existence of our TI's? ... Most of our software is produced by small companies. Do you really think that they will continue to bring out quality new programs just to have them go from place to place via "PIRACY"?

Now about FREeware authors. Those people who work or go to school and write SOFTWARE at night and on the weekends, sometimes for months, and then have the gall to ask you to send them 10 or 15 dollars if you like or can use their programs. How many people ever do? According to the authors, not ver many. The crazy part is that most of these programs are really quite good! Some like PRBASE and DM1000 v3.5 (the latest) are very useful. Those like Danny Micheal's "SCREEN DUMP" and "NEATLIST" allow us to do things that we could not do before.

BUT send these people money? Why should I, when I get it for free? What the pirate (thief) doesn't realize is that nothing is 'FREE'. We will all pay for this 'FREE'ware by eventually having a machine that will run only the oldest, out-of-date programs around.

How long do you think they are going to continue to write for this machine? Why should they? Did you know that there is a TIBB's in our area 'up and running' on PIRATED SOFTWARE? (not here in KC as far as I know--editor)

THEN WHAT ARE WE GOING TO DO???

Well, don't accept any PIRATED SOFTWARE. To you are doing this PIRATING please KNOCK IT OFF!! To SOFTWARE producers, charge a reasonable price. And if you must protect your goods, please place two copies in the package that the legitimate users are protected too.

I sucked up SOFTWARE with the best of em. NO MORE!! WHY? I saw that, fo machines that are in production, there are 15 to 30 titles a month being released for them. BUT for our ORPHAN -zip- So far we have been lucky. Are we willing to push that luck? I hope not.

FAST FILE TRANSFER WITHOUT TEII -by Larry Prikockis Cleveland Area 99/4A U.G.

When transmitting files over the phone, the easiest and most reliable method is to use the Terminal Emulator, (but now the use of Xmodem is consider to be as reliable- editor note) The price for this ease of use and error checking is that the process is very slow. Fortunately, there is a way to transfer Basic and Xbasic programs at five times the speed of TEII.

First, both the sender and receiver must select TI BASIC. Then, load teh program to be sent into memory and type "SAVE RS232" but do not press enter YET. The receiver should type 'OLD RS232" When you are ready, switch over to the modems and press enter. The receiver should wait about 5 seconds and then press enter also. boath users should see a munber; "corresponding to the number of block of the program, at the top of the screen. This number wil slowly count down to zero at which point the transfer is complete. Since the program is loaded directly into memory, the receiver must remember to save the program to disk or cassette.

The use of this method is limited to programs that are written in basic or x-basic. To use this method for the transfer of assembly programs is a little more complicated. First, both users must have an E/A module. Select the Editor Assembler and press '1' for the edit section. The receiver should do the same. Then select option 4 for print. When you are asked for a filename, enter "RS232.BA=300.CRLF" The receiver selects option 1 for load (make sure Disk is in the drive) and uses "RS232.BA=300" as a filename. When both sides are ready, switch over to the modems and press enter. Again, the receiver should wait five seconds before hitting enter. With this method, there is not way of telling how much more of the file is left to be sent, so just sit back and wait unti it is finished. The file can be saved to disk by using option 3.

XMODEM File Transfer Protocol

By Keith Amann

[This article was first published in the Capital PC Monitor and PC World magazine during the Spring of 1983, and has been updated for republication in this newsletter.]

When transferring files between computers using the telephone system, there is always the chance that electrical noise will result in data transmission errors. To ensure proper transfer of files it is necessary to detect data transmission errors and to retransmit data that contains errors. Most people think that asynchronous parity error detection provides that capability. It does not. Parity error detection does tell you when a data transfer error has occurred, but it is up to you to retransmit the data to correct errors. The problem is that parity error detection is not actually performed by most TI-99/4A communication packages. If a package does perform the error detection, it may not inform you of errors in such a way that you know to immediately retransmit the data. To ensure "error-free" data transfer you need a protocol file transfer technique.

A protocol is a set of rules and conventions that apply to a specific area of communications that allow participants to properly communicate regardless of the hardware brand or software package being used. The protocol file transfer is a set of rules for transferring files which specifies a set of ASCII handshaking characters and the sequence of handshaking required to perform certain file transfer functions. Protocol handshaking signals allow communication software to transfer text, data and machine code files, and to perform sophisticated error-checking. The handicap in using protocol file transfer techniques is that the computers on both ends of the communications link must be using compatible software; there is no standard that controls these protocols.

The Ward Christensen XMODEM protocol is one specific file transfer protocol that has become a default standard in personal computer communications because of its widespread use on bulletin boards and because of its inclusion in low cost personal computer communication packages such as FAST-TE.

By providing you with this insight into protocol transfer and explaining in detail the operation of the XMODEM protocol, I hope to add momentum to the development of a "standard protocol" whether it be the XMODEM model or some other model. Users of communication software deserve a standard protocol that will allow them to use the technique with any microcomputer regardless of the software packages employed.

The XMODEM protocol is illustrated in Figure 1. As you can see from that figure, XMODEM does not begin the transfer of data until the receiving computer signals the transmitting computer that it is ready to receive data. The Negative Acknowledge (NAK -- ASCII 21) character is used for this signal and is sent to the transmitting computer every 10 seconds until the file transfer begins. If the file transfer does not begin after 10 NAK's are sent, the process has to be manually restarted.

After a NAK is received, the transmitting computer uses a Start of Header (SOH -- ASCII 01) character to signal the start of a data block transfer. The SOH is followed by two numbers. The first number is the number of the data block to be transmitted, and the second number is the ones complement of the data block number. The ones complement is the block number XOR 255 or simply 255 minus the block number.

The block number and the ones complement number are then followed by a 128-byte block of data and an error-checking checksum. The checksum is calculated by adding the ASCII values of each character in the 128 character block; the sum is then divided by 255 and the remainder is retained as the checksum.

After each block of data is transferred, the receiving computer computes its own checksum and compares the result to the checksum received from the transmitting computer. If the two values are the same, the receiving computer sends an Acknowledge (ACK -- ASCII 06) character to tell the receiver to send the next sequential block. If the two values are not the same, the receiving computer sends the transmitter an NAK to request a retransmission of the last block. This retransmission process is repeated until the block of data is properly received or until 10 attempts have been made to transmit the block.

If the communications link is noisy, resulting in improper block transmission after 10 attempts, the file transfer is aborted. XMODEM uses the block number and the ones compliment block number at the start of each block to be sure the same block is not transmitted twice because of a handshake character loss during the transfer. The receiving computer checks the transmitted block to be sure that it is the one requested and blocks that are retransmitted by mistake are thrown away. When all data has been successfully transmitted, the transmitting computer sends the receiver an End of Transmission (EOT -- ASCII 04) character to indicate the end of file. The XMODEM protocol offers the TI-99/4A several advantages over other protocols and file transfer methods. First, the protocol is in the public domain which makes it readily available for software designers to incorporate into a communications package. Second, the protocol is easy to implement using high level languages such as BASIC or Pascal. Third, the protocol only requires a 256-byte communication receive buffer which makes it attractive for TI-99/4A owners who only have 32K systems. Fourth, the protocol allows a user to transfer non-ASCII 8-bit data files between microcomputers because it calculates the end of a file based on file size and uses handshake signals to indicate the end of a file instead relying on an end of file marker character (control-Z) to terminate a file transfer. Fifth, XMODEM error-checking is superior to normal asynchronous parity error checking. The parity method of error-checking is 95% effective if the software on the receiving end checks for parity errors. XMODEM error-checking is 99.6% effective, and the software on the receiving end must check for errors. Parity errors detected also do not result in automatic retransmission of the bad data; XMODEM detected errors result in data retransmission until no errors are detected or until 10 retransmissions have been attempted. Finally, the protocol is used by many bulletin boards and having the protocol in a communications package allows the TI-99/4A user to receive error-checked files from these bulletin boards. XMODEM does not solve all communications problems, but it does provide compatible file transfers between a large variety of computers. Now that we have this common microcomputer link, it is time to apply pressure for a more universal public domain asynchronous communications protocol that can be used between any two computers.

XMODEM Protocol File Transfer

```

Receiving Computer Ready to Receive
*****
Transmitting Computer Ready to Transmit
*****
-----\NAK\----->
|-----|
|<--/SOH/001/254/Good^Data/CkSum/----|
|-----|
|-----\ACK\----->
|-----|
|<--/SOH/002/253/Good^Data/CkSum/----|
|-----|
|-----\ACK\----->
|-----|
|<--/SOH/003/252/Garbled^Data/CkSum/---|
|-----|
|-----\NAK\----->
|-----|
|<--/SOH/003/252/Good^Data/CkSum/----|
|-----|
|-----\ACK\----->
|-----|
|<-----/EOT/----->
|-----|
|-----\ACK\----->
|-----|

```

```

***** 0000PS! HERE'S INST.
ENTER- START/STOPS TEXT.
"E" - PRINTS PREV. LINE.
"X" - PRINTS NEXT LINE.
"P" - PRINT FROM LAST LINE
ON SCREEN TO END OF BUFFER

```

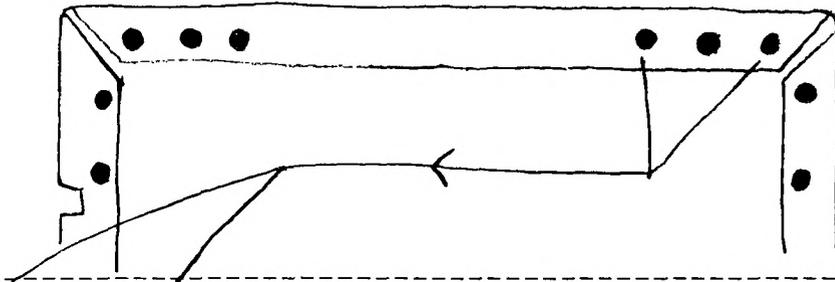
```

100 *****
110 REAC P:LINE:
120 by
130 John Martin
14 Southern Nevada
15 User's Group
16
17 d:plays D-V 80
18 files to screen
19 and can print
20 all or any
21 part to printer
22 or d:
23 *****
24 :W:LAY AT(11,1)BEEP ERASE ALL:"OUTPUT DEVICE? ": :PI0" :: ACCEPT AT(13,1)S
25 :PRTR$
26 :W:LAY AT(15,1)BEEP:"INPUT FILENAME?": :DSK" :: ACCEPT AT(17,4)SIZE(-12):F
27 :L$
28 IF LEN(FILE$)=0 THEN 250 ELSE FILE$="DSK"&FILE$ :: IF POS(FILE$, " ",1)THEN 2
29 50
30 OPEN #1:FILE$,INPUT :: OPEN #2:PRTR$ :: DIM BUFFER$ 255
31 FOR X=1 TO 24 :: IF EOF(1)THEN 400 ELSE LINPUT #1:IF R$(X):: PRINT BUFFER$
32 : : BUFFER$(X+1)=RPT$( " ",28)
33 : : CALL KEY(3,K,S):: IF K=13 THEN 310
340 NEXT X :: GOTO 280
350 CALL KEY(0,K,S):: IF S=-1 THEN 310 ELSE IF K=13 THEN 280 ELSE IF K=81 THEN E
360 ND
370 IF 1+((K=69)+(K=88)+(K=80))THEN 310 ELSE ON POS("EXP",CHR$(K),1)GOTO 330,350
380 370
390 X=X-1 :: IF Y=0 THEN X=24
400 PRINT BU:IF S=-1 THEN X: :: CALL KEY(0,K,S):: IF S>1 THEN 340 ELSE IF K=69 THEN 330 E
410 LSE IF K=80 THEN X=370 ELSE 310
420 Y=Y+1 :: IF Y=24 THEN X=1
430 : : PRINT BU:IF S=-1 THEN X: :: CALL KEY(0,K,S):: IF S>1 THEN 360 ELSE IF K=88 OR K=120 T
440 350 ELSE : :
450 : : CLEAR :: FOR N=X TO 24 :: PRINT BUFFER$(N):: IF BUFFER$(N)=RPT$( " ",28)
460 : : ELSE PRINT #2:BUFFER$(N)
470 : : N
390 IF EOF(1)THEN EN P: ELSE LINPUT #1:X$ :: PRINT X$ :: PRINT #2:X$ :: CALL KEY(0
400 : : IF S=-1 THEN X: ELSE 390
410 : : PRINT : : IF N=0 THEN FILE. " :PRESS:": : (P)review or (A)nother file"
420 CALL KEY(3,K,S):: IF S THEN IF K=82 THEN :W:LAY AT(23,1)" " :: GOTO 310 EL
430 SE IF K=65 THEN CLOSE #1 :: CLOSE #2 :: GOTO 410
440 GOTO 410

```

TIC TALK TECH

IDENTIFICATION OF 5 1/4" DISKS



SEAL	COMPANY	COMMENTS
• • •	MAXELL	
Complete Seal	MEMOREX	ALSO ALBINAR
• •	VERBATIM	
• •	NASHUA	6 DOTS DOWN EACH SIDE
▬ ▬	BASF	
••••••••	ELEPHANT	
▬▬▬▬	3M	2 BARS DOWN EACH SIDE
▬▬▬▬ ▬▬▬▬	WABASH	6 SETS OF 8 SQUARES PER SIDE
▬▬▬▬	FUJI	
▬▬▬	CONTROL DATA	STORAGE MASTER
▬▬▬▬	SYNCOM	3 BARS DOWN EACH SIDE
• •	CERTRON	8 DOTS DOWN EACH SIDE
• •	BONUS	
▬ ▬		

This chart shows the name of the manufacturer who may have produced the disks that you are currently using if you are unsure of make .
 Thanks to N.O.V.A. Users group in Vancouver, Wa. for this chart.



JUST FOR YOU



Modified by Ralph Romans:

- Ver 3.0 Fixes to Ver 2.4
- Incorrect file count when going from 'M' to 'C'
 - File copy would give you a bad copy if the file being copied was stored on the master disk as a non continuous file and the size of the first segment was exactly 39 sectors with additional sectors in another segment on the disk.
- Ver 3.1 Fixes to Ver 3.0
- File copy would give you a bad copy if the master file was a fracter file of exactly 39 sectors and the same file name was on copy disk.
 - When entering a file name in various modes it was possible to mess it up.
- Unfixed bugs in Ver 3.1
- unable to display some dis/var 80 files that are full of control characters. Computer hangs up.
- Ver 3.3
- Changed defaults on sweep and disk initialization.
 - Disk initialization works for Myarc and Corcom.
 - read/write errors gets cleared after 1st use on disk copy.
 - file 'Mgr1' may now be called any name and all features of DM1000 will work.!! This will only work with TI controller and Corcom controller.
 - The loader for Myarc controller is called loadmy.
 - During disk initialization menu, you can use the up arrow to go back to previous question.
- Ver 3.4
- Able to delete/move/copy 1 sector files.
 - Added 'up arrow active' notice when up arrow will take you back to previous question.
- Ver 3.5
- Able to type/print display var 80/fixed 80 files while the file listing is on the screen by pressing a 'T' for type(display) file to screen or 'P' for print to list device with optional control codes sent to printer first.
 - The 'P' and 'T' for print or type are only valid in the left most field.
 - 'EOF' noticed added in the lower lefthand corner of screen.
 - Display var 80/fixed 80 menu removed.

The latest version of Disk Manager 1000 is 3.5 and it is truely remarkable. The above is a listing of the version 3.x history showing the additions, improvements, and the enhancements.

This latest version has two significant enhancements, which involve the displaying and printing of d/v-80 files directly to the printer from the [cmd] prompt as the cursur is sitting there a "t" for type (to display), or a "P" for the printer may be typed instead of the normal copy/move/or delete.



DESIGN YOUR OWN CURSOR.....

I can't give proper credit to whoever originated this program to create a TEXAS cursor, but my goal is not for you to be the proud user of a TEXAS type cursor, but rather you know how to create your own CUSTOM CURSOR!

Whatever program that you use, assembly, or extended basic, you will have to encode the design for your CUSTOM CURSOR. The program will be the vehicle for your own cursor.

```

1 !TEXAS CURSOR from GOTO Newsletter of Columbus, GA. Users Group;
  unattributed, but JIM PETERSONS and DR. RON ALBRIGHT'S names came
  up.
2 CALL CLEAR :: CALL INIT
3 CALL LOAD(8196,63,248) ! REF table pointer at >2004 (3F,F8)
4 CALL LOAD(16376,67,85,82,83,79,82,48,8) ! Indicates that a pro-
  gram named "CURSOR" begins at >3008
5 CALL LOAD(12288,48,48,63,255,254,124,24,12) ! THIS IS WHERE WE
  START THE CUSTOM CURSOR DESIGN
6 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,36,4,91)
7 CALL LINK("CURSOR") ! Links to the cursor program.
    
```

If you are interested in creating your own cursor, please read the rest of this page, and I'll show you how to chart out this TEXAS CURSOR, and how to create your very own, let's say one with your initials, or a square box. The creation is very much the same as charting a sprite in extended basic, but instead of using HEX, you will be using straight BINARY.

BINARY WEIGHT	128	64	32	16	8	4	2	1	
ROW # 1			X	X					= 48
ROW # 2			X	X					= 48
ROW # 3			X	X	X	X	X	X	= 63
ROW # 4	X	X	X	X	X	X	X	X	= 255
ROW # 5	X	X	X	X	X	X	X		= 254
ROW # 6		X	X	X	X	X			= 124
ROW # 7				X	X				= 24
ROW # 8					X	X			= 12

LOOK AT LINE "5" IN THE ABOVE PROGRAM.

If you look at the above chart, you will see the TEXAS CURSOR defined. Now all you will have to do is make a blank chart similar to the one above, and instead of putting the "X"'s in for TEXAS, put in the "X"'s to match the shape of your custom letters, or logo. Add up the numbers (binary weights) across the top of the chart above the place you inserted an "X". Do this for each row, and put the total at the end of each row.

Enter these totals in line 5, AFTER the address in the CALL LOAD(12288,xxx,xxx,xxx,xxx,xxx,xxx,xxx). Now save this program to disk! RUN the program, and you are all set.

The program should stay in the machine until you either:

- * Write over it with another program, not likely with xbasic.
- * Shut the system down.

HAVE FUN!

From J. F. Willforth of the WEST PENN 99'ERS March, 1987

ROCKY MOUNTAIN 99'ERS TIC TALK

This publication is printed monthly for the benefit of the membership of the Rocky Mountain 99'ers Computer Club. The Club and the paper are not for the benefit nor backed by any commercial enterprise. Both are non-profit in nature and are for the sole purpose of computer education. Any fees collected are used to defray any cost to maintain the organization. Neither the paper nor the Club have any affiliation with Texas Instruments. Any statements published in this paper are not necessarily the opinion of the membership.

OFFICERS AND CHAIRMEN

PRESIDENT-----	MARK PAYNE-----	289-7667
VICE PRESIDENT-----	MIKE MILLER-----	288-9251
TREASURER-----	KEN MARBLE-----	986-0371
MEMBERSHIP CHAIRMAN-----	KEN MARBLE-----	986-0371
EDITOR-----	MARK PAYNE (ACTING)-----	289-7667
LIBRARIAN-----	PETE RORHER-----	989-0958
LIBRARIAN ASSISTANT-----	ED HOWARD-----	366-2303
PROGRAM CHAIRMAN-----	JOE MILLER-----	458-5600
RM99ER'S BBS SYSOP-----	MARK PAYNE-----	289-7667
RM99ER'S BBS ASSISTANT SYSOP-----	MIKE MILLER-----	288-9251
RM99ER'S BULLETIN BOARD-----		288-3692

ROCKY MOUNTAIN 99'ERS
8248 PEARL STREET
THORNTON, COLORADO 80229



""NOTE: NEW ADDRESS""

FIRST CLASS MAIL

```

.....:
:Please check the expiration:
:date on your label. This is:
:the date your membership will:
:expire. Renew today so you:
:do not miss a single issue of:
:TIC TALK !!!!!:
.....:

```

Edmonton Users Group
P.O. Box 11983
Edmonton, Alberta
Canada T5L 3L1