THE NATIONAL NINTY-NINER

VOL I - NO. 12 - NOVEMBER-DECEMBER, 1984

MONTHLY NATIONAL NEWSLETTER OF THE

THE 99ER'S ASSOCIATION
3535 SO.H ST., #93
BAKERSFIELD, CALIF. 93304
(805) 397-4361
DON VEITH - EDITOR/PRESIDENT

CREATED FOR TI 99/4A HOME COMPUTER OWNERS AND USERS GROUPS

ANNOUNCEMENTS

MORNING STAR SOFTWARE ANNOUNCES ADDITIONAL PRICE REDUCTIONS ON CP/M CARD

Scott Swenson, developer of Morning Star's CP/M Card, informed our organization that the list price of the CP/M Card has been reduced to $495.00. The 10% discount to members of a Users Groups, covered in our December issue, is still in effect. This discount effectively lowers the price of the CP/M Card to Users Group members to $445.00.

All the software carried by Morning Star was reduced 10% as announced in our October newsletter. There will be no further reductions beyond the initial offer. Persons having telephone conversations with us in Bakersfield have asked what value the CP/M Card adds to the TI Computer? A whole complete new computer system, that's all!!

The CP/M Card uses the TI system as its host operating environment. You still can use your 99/4 and its system components, as TI designed them to be used, at any time. The added bonus is a whole new operating system available at your fingertips anytime you turn the system on to play or accomplish some serious task. Your additional investment is only limited to the CP/M Card and NOT a whole new unfamiliar computer system. I personally have found these prospects extremely challenging.

NATIONAL USERS GROUP DIRECTORY EFFORT UNDERWAY

Information on Users Groups for each type of computer is being assembled for an upcoming publication entitled "NATIONAL DIRECTORY OF USERS GROUPS". I spoke with Ken Ryder who is coordinating this effort to produce the Directory. Please send Ken a SASE (self-addressed, stamped-envelope) for a questionanaire and information on his project to insure your organization is listed in the Directory. Tell Ken you saw it in THE NATIONAL NINTY-NINER. The address for information is:

KEN RYDER
P.O. BOX 4102
ROME, NY 13440
(315) 339-1069

CORCOMP'S 9900 MICRO-EXPANSION SYSTEM

Our organization has been priviledged to be supplied with four (4) of the prototypes of this unit for evaluation purposes. In-depth articles on our experiences using these units will appear starting with the January, 1985 issue. Two of the units will be used by individuals in ordinary daily use for game playing, word processing, Multiplan and general programming. The other two units will be placed in service operating bulletin boards, one of which operates on a 24 hour per day basis.

The only initial difference one notes upon external inspection of this unit versus the RS-232 version is the connector for the disk drive ribbon cable. Upon opening the case, the upper board becoaes visible for inspection. The upper board, housing the 9900 Disk Controller and 32K Memory Expansion, is literally aounted on top of the RS-232 board. The switches for adjusting your disk drive head step settings are exactly like the 9900 Disk Controller Card for the TI Expansion System. The switches are located at the left rear corner of the unit where the disk controller indicator lamp wires are terminated. All switches are factory adjusted for a 10 millisecond head step time. CorCoap has advised our organization that it is not recommended for the average owner of a stand alone 9900 RS-232 unit to install the upper board. The installation requires changing a PROM and additional wiring connections. In fact, if you have purchased an RS-232 version of the 9900 Micro-Expansion System, the unit will have to be returned to CorCoap if you decide to upgrade it to its full capabilities added by the upper board (D/S, D/D, Disk Controller and 32K Memory Expansion).

To connect a disk drive to the Micro-Expansion unit, the owner will have to purchase a 34 position connector for the disk drive ribbon cable. Radio Shack has such a connector, Archer part # 276-1525, which I purchased for $3.19 plus sales tax. The purchase price may vary slightly from area to area. The connector is fastened onto the end of your disk ribbon cable after removing the board type connector which TI used to connect disk drives to the Expansion System and stand alone Disk Controllers. The connector for a second disk drive was already installed on my standalone disk drive. This type of inline splicing connector, to install a second disk drive, is also available at Radio Shack for $4.95. Check with a local electronics store to purchase the required 34 wire ribbon cable if none is supplied with the disk drive you purchased. Perhaps, a better suggestion is to consult the supplier of your disk drives and have the proper cables and connectors installed at the time of purchase. Another source of assistance is your local Users Group. Each Group I have had the priviledge of visiting or corresponding with had at least one resident electronics expert who gave generously of their free time and knowledge. REMEMBER, it is their time that is being donated and you as the requestor of assistance SHOULD adapt your schedule to that person's available time. One other point, limit your calls for assistance to between 7:00-9:00 pm weeknights and 12:00 am to 4:00 pm on weekends. I can personally assure you this courtesy will be fully appreciated. The individuals who provide this assistance have a family and personal commitments that require their time and attention.

The 9900 Micro-Expansion System plugs directly into Peripheral Expansion Bus of the TI-99/4. The Speech Synthesizer must be the first component plugged into the 99/4, the CorComp 9900 Micro-Expansion unit is installed as the next unit. Disk drive(s) are attached as explained in the previous paragraph. You are then ready to power up and enjoy this substitute for the TI Expansion System. The most noticeable difference one experiences is the absence of the Expansion System's cooling fan and that unit's massive size. The one drawback to the 9900 Micro-Expansion unit is that no further expansion of the unit is possible beyond its present capabilities. The Peripheral Expansion bus is extended through the unit which would permit the atachment of additional units at some future time. Only CorComp or other manufacturers know what may be available to use this option for in the future.

EXTENDED BASIC AVAILABLE!

An announcement arrived in our mail stating that a firm had purchased the rights from TI to manufacture this module. Exceltec Inc. (formerly Sunware Ltd.) is offering a completely interchangeable Extended Basic module. The letter states, "We use TI parts so therefore you can expect our module to have the same high quality performance and characteristics. Our manuals follow the same format as the TI documentation. Our product also includes the handy command reference card."

Some features of the Extended Basic module are:

Over 40 additional commands to control features of the TI-99/4A.

Allows multiple statement lines and automatic loading from disk plus Assembly Language support commands.

Full control of Sprite Graphics, speech, and sound.

Permits control or errors, warnings, and breakpoints plus comands to facilitate debugging.

Exceltic announced a special offer available to TI Users Groups outlined below:

QUANTITY:	1	2	5 OR MORE
PRICE :	$99.95	$89.95	$79.95

Your order may be telephoned to Exceltec at 1-806-794-9161. The firm accepts VISA-MASTERCARD on orders. Funds may also be forwarded to the firm's address: Exceltec Inc., P.O. Box 54380, Lubbock, Texas 79453. The letter was signed by the firm's President, Charles Roberts. Tell the firm you saw the information in THE NATIONAL NINTY NINER.

LOWER DISK PRICES ????

We have seen disk prices keep going lower and believed the bottom had been reached. A letter arrived from a firm in Anaheim, California, which has some outstanding prices on double-sided, double-density half-height disk drives. The units and prices are outlined below. Edward Chang, Sales Manager for the firm, stated the prices quoted were effective until January 31, 1985. The firm ships all drives ordered via United Parcel Service. All orders are shipped C.O.D. within three working days of the order being placed with the firm. The firm does not take Mastercard or Visa. Your payment is made to the UPS delivery agent with a Cashier's Check or cash at the door when the units arrive. UPS does charge $1.65 for their COD service which is paid by the customer. The COD charge will increase to $1.90 effective January 1, 1985. Perhaps a small inconvenience compensated for by the reduced prices. California residents must pay their state's six (6) percent sales tax on all items purchased. All out of state purchases are, of course, exempt from the California Sales Tax.

DISK DRIVE	1 TO 9	10 TO 29	30 TO 49
TOSHIBA FDD5401AOK01	$140.00	$135.00	$132.00
TEAC FD-55B	$115.00	112.00	109.00
PANASONIC JA551-2	110.00	108.00	106.00
HITACHI HFD-505B	110.00	108.00	106.00

The firm has manuals and technical data on all units available for sale. Shipping charges are $3.50 per drive for the East Coast and $2.50 for the West Coast. Please contact the firm at the address listed below. Tell them you saw the information in THE NATIONAL NINTY NINER!!!

PROMETHEUS WORLD ENTERPRISE CO.
P.O. BOX 9426
ANAHEIM, CALIF. 92802
(714) 957-6009

CORCOMP - SOME THOUGHTS

By Don Veith, Editor

Three members of our organization visited CorComp on December 3, 1984. We had extensive conversations with Jackie Sagouspe, former Marketing Director, who is now managing the firm. If you are wondering how they are doing, meetings were held Monday morning in which the participants discussed plans for new product development!

The firm's Chapter 11 Bankruptcy hearings were uneventful with no major problems. In this type of bankruptcy, the court normally appoints an outside individual to act as Trustee of the firm and supervise its reorganization. The court instead choose to appoint Jackie Sagouspe as CorComp's Trustee. A rather unusual tactic in this type of case.

The firm has achieved a major reduction in expenses. All assembly work is now done by an outside contractor literally around the corner from their address. The contract firm hired all of CorComp's previous assembly line personnel. CorComp has thus been able to lower its product failure rates. Quality control is now the assembly firm's concern. Corcomp has instituted a quality control check program on units received from their assembler. Reduced expenses are also reflected in not possessing manufacturing equipment which is subject to breakdowns and required maintenance expenses.

The firm currently has eight employees. The front office staff consists of Jackie Sagouspe, Lorelei Alkier, Sally Beaman, and Moritza. Jackie, President of corComp, welcomes visitors, handles public relations, runs the company, packs product into cartons, answers telephone calls, and does just about whatever is needed. The Accounts Supervisor is Sally Beaman. Her primary purpose is to maintain financial solvency for the firm by monitoring its financial status at all times. Lorelei Alkier is Director of Purchasing and Marketing. Lorelei orders all components to insure a sufficient inventory of parts are on hand to meet production schedules. The receptionist, Moritza, answers all of your telephone calls. Her other duties include Customer Service Relations and handling RMA's, or Return Material Authorizations, which are required prior to return of an inoperative product for repair.

Mike Norton is Chief Project Engineer. He supervises the technical end or back office area of CorComp. Mike is currently working on new product planning and development and assists the Technicians in solving repair puzzles. He also is the person primarily responsible for solving the problems encountered with both the 9900 Disk Controller Card and the Micro-Expansion System. Mike Bonitta, the Chief Technician, was just added to the staff effective December 3, 1984. He possesses a strong background in computers specializing in disk controllers and drives. This gives the firm two technicians to handle the updates and repairs for the firm's products. Yung is currently taking over the shipping and receiving chores from Kelly. Kelly is transitioning from his previous task to become the second Technician to assist in product repair.

The firm's reorganization has placed increased emphasis on serving their customer needs. Their effort is directed towards solving customer complaints, repairing products which have encountered a component failure, and providing upgrades to existing products purchased by customers. The turnaround for products needing component repair should decrease dramatically under the reorganization. Technicians were busy testing 9900 Disk Controller Cards when we visited the firm replacing components as needed and upgrading early versions of the firm's products.

The 9900 Micro-Expansion System prototype units are in the hands of users around the United States at the present moment. I was priviledged to obtain one of these units. The most unique feature, excluding its compact size, is the silence one enjoys not listening to the Expansion System's fan. That factor alone makes it worth the price. All kidding aside, the unit has performed in a flawless manner since we first turned it on one week ago. The ability to use its double-sided, double-density Disk Controller is an unusual experience. It takes a bit of time to adjust to our diskettes being capable of storing 1440 sectors of data or 366,000 plus bytes of storage. The experience of observing the disk sector counter continue increasing beyond 358 sectors is hard to describe. The unit also contains full RS-232 capabilities and 32K of Expansion Memory. Response time to commands typed in is perhaps a bit quicker than with the PES. The various component indicators light up as the computer accesses the particular unit. The POWER lamp stays on constantly during unit operation. The RS-232, 32K RAM, and DISK SYS indicators are the other lamps present on the unit.

CorComp has placed the majority of the 20 test units with actual users in the field for evaluation rather than with distributors. This has given the firm excellent feedback on the unit's capabilities and has assured a testing program of the broadest scope possible. Jackie stated the firm will have the Micro-Expansion System in the distributors hands for customer purchase January 1, 1985. I would advise contracting your distributor now if you are interested in obtaining one of these units. Remember, you must also acquire a stand alone disk drive as the 9900 MES unit does not contain an internal drive.

CorComp has survived its crisis period and the firm's recovery appears definitely underway. I sensed an attitude that the employees were dedicated to keeping the firm alive and financially healthy. The meetings held to discuss new product development and schedules for its orderly planned implementation are excellent signs a lesson was learned from previous errors. Plans are still under consideration to produce a new replacement computer for the 99/4. The largest roadblock for the computer's development is funds to finance the research and development effort required to deliver a NEW computer to the marketplace. The current employees believe such an undertaking is not beyond CorComp's grasp. I also believe the firm will develop the new computer. In the present, drop the folks at CorComp a note, place a telephone call to indicate your support, or simply purchase some of their products. You may contact the CorComp folks at:

CorComp Incorporated
1255 No. Tustin Ave.
Anahiem, Ca 92807
(714) 630-2903

On behalf of all the individuals at CorComp, a very Merry Xmas and a Happy New Year. The staff of THE 99'ERS ASSOCIATION extends our best wishes to each of you for an excellent Holiday Season and a Prosperous New Year.

ARTICLES

WORKING WITH MORNING STAR'S CP/M CARD - PART 1

By Don Veith, Editor

We wish to express our gratitude to Scott and Janet Swenson for making the Morning Star Software CP/M card available to our organization for evaluation purposes. This series of articles would be impossible to perform without this firm's fine product.

The first myth I wish to dispel is that no software is available or sources of CP/M programs are severly limited. These statements are far from the truth. Research in books describing the CP/M language and system operation list several sources of Public Domain software. Two such groups are the CP/M Users Group and the SIG/M Users Group. The first group operates from the address of Lifeboat Associates in New York. The CP/M Users Group has 60 different disk volumes of programs available for $8.00 and a $4.00 registration fee.

The SIG/M Users Group has disks available for $6.00 each including U.S. postage. A self-addressed, stamped envelope to either address below should obtain additional information and a catalog from each group.

CP/M USERS GROUP
2248 BROADWAY
NEW YORK, NY 10024

SIG/M USERS GROUP
BOX 97
ISELIN, NJ 08830

An additional review of THE COMPUTER SHOPPER revealed that CP/M and Osborne Users Groups exist. A review of the Classified Advertisement section of the same publication yeilded the address for a Public Domain group for CP/M programs. A review of the Users Group listings indicated Osborne and CP/M Users Groups are present in each state. This list only represents Users Groups which have contacted THE COMPUTER SHOPPER and placed their name and address on this list. Efforts are being made to locate a list of Osborne Users Groups. A National Osborne group called the First Osborne Group (FOG) does exist. This organization has a program exchange and produces a newsletter for its members. Additional details will be available when this information is received. For those who wish to correspond directly, FOG's address is listed below:

FIRST OSBORNE GROUP (FOG)
P.O. BOX 11683-A
PALO ALTO, CA 94306

An interesting summary of the differences between the TI operating system and CP/M is outlined on page 1-7 of Morning Star's CP/M supplement. The differences are:

1. CP/M disk format is 11% larger to accomodate "System tracks".
2. CP/M waits for a command while TI prompts with a menu.
3. CP/M prevents write operations to disks it does not recognize, TI readies itself to write.
4. CP/M has no built in BASIC though add-ons are available.
5. TI-CP/M functions are generally not in the top row of keys.
6. TI-CP/M makes use of the SHIFT key in conjunction with CTRL and FCTN.
7. To reset CP/M type CTRL-FCTN-ENTER rather than FCTN-=.
8. You must use CP/M special key codes rather than cursor keys.
9. Any cartridge installed is ignored by CP/M.
10. TI-CP/M uses the high resolution graphics mode for text.
11. TI-CP/M allows scrolling up and down over several pages of text.
12. TI-CP/M normally operates in an 80 column mode showing 40 columns at a time.
13. An editor and assembler for CP/M are included.

I would like to spend a few moments reviewing the history of CP/M. Gary Kildall is the developer of the CP/M operating system. In 1973, Intel had developed the first microprocessor and wanted to implement a language called PL/1 for the new chip. PL/1, developed by IBM in 1964, is a combination of ALGOL, FORTRAN, and COBOL. Kildall found he had to develop an operating system to tie the computer together. Memory was extremely limited and an operating system using 4K bytes of RAM was developed. The system contained routines required by a programmer. Kidall was a computer scientist and the operating system was thus naturally oriented towards expert users.

After completing his development work, Kildall offered the system to Intel. The system name, Control Program for Microprocessors or CP/M, was coined at this time. Intel reviewed the system,decided no market for it existed, and declined to purchase it from Kildall. CP/M was completed in the fall of 1973 by John Torode working with Kildall. Torode completed his early development work and made the operating system work. In 1976, Kildall and his wife Dorothy started Digital Research to market CP/M and continue development. The system was offered for sale to computer hobbyists for a price of $70.00.

Kildall had created something unique when he separated the machine specific parts of the system and placed them in one unit called BIOS or Basic Input-Output System. The remainder of the system was the same regardless of the computer system. All a company that wished to offer different options needed to do was change a section of BIOS to implement their design. Thus the rapid growth of CP/M. CP/M has, in addition to BIOS or the hardware interface, CCP or Console Command Processor, and BDOS or Basic Disk Operating System. BDOS supervises applications program execution and input and output. By carefully choosing its commands and utilities, CP/M is a powerful tool. It does require the programmer to possess an excellent knowledge of what the system is doing and memorize command syntax.

Some of the problems associated with CP/M relate to the 8K bytes of memory it occupies. This limited memory does not allow for fail-safe or error checking routines. Digital Research's documentation leaves a bit to be desired. In fact, a couple of authors really believe it is not written, but rather is encrypted. This has created a market for some excellent publications on getting familiar with CP/M and its operating procedures. The 2.0 version of CP/M was released in 1979. Version 2.0 added a table-driven disk driver. A disk system has a delay after each sector is read for the operating system to interpret information read from the disk. The disk rotates several sectors before the operating system is ready to read another sector. A waiting period was thus necessary until the disk rotated to the right location to read the next sector. Version 1.4 cured this problem by making the physical sector seven (7) on the disk sector two (2) for the operating system. This six sector "skewing" or offset factor was set and resulted in increased disk system operating speed. Version 2.0 was table driven with the fixed skew in BIOS which could be set and controlled by the disk manufacturer. The table could be adjusted for any number of tracks and sectors on a disk. This made CP/M adaptable for disks of all types from soft sectored to hard.

CP/M's operating system was designed to load itself into the highest location of the computer's memory. The maximum memory size is 64K for a single user system. The largest number of programs developed for CP/M are for the eight bit 8080 series of processors. Digital Research has not enjoyed the success with its 16 bit version of CP/M enjoyed by the pioneering eight (8) bit version. MS-DOS, implemented by IBM as PC-DOS, has given Microsoft a commanding lead in the 16 bit system battle. The interesting point here, relatively unknown, is that IBM aproached Digital Research to adopt its CP/M operating system for the PC. Gary Kildall was unavailable for a meeting with the IBM executives due to a prior commitment. His being unavailable caused the IBM executives to board a plane for Washington state and their meeting with his competitor Microsoft.

The next subject to cover is the elements of CP/M's operating system. A brief discussion of the CCP, BDOS, and BIOS is our next subject. The CCP or Central Command Processor is only necessary between programs to interact with the user. CCP is loaded when the system boots up and may be overwritten whenever a program is executed. It must be reloaded to interact with the user when the program is finished. Some of the resident commands in CCP are DIR, ERA, REN, TYPE, SAVE, and USER. If you remember the last article, these commands were used for interaction with disk operations.

All CP/M programs and the CCP system commands use BDOS to converse with external system components (disk drives, printer, modem). BDOS establishes the necessary peripheral access to perform a task while BIOS implements that specified task. When the task is finished by BIOS, control is then passed back to BDOS. BIOS is utilized to communicate with the hardware, I/O ports and peripheral units connected to them. A separation of logical I/O functions (BDOS) from the physical I/O functions (BIOS) is perhaps a better explanation. BDOS has a single entry point and uses information in a register to specify a specific task to be accomplished. BIOS has several entry points depending upon the function being performed. Some programmers have modified this procedure and accessed BIOS directly with disastrous results. We are advised never to write programs that bypass BDOS and access BIOS directly.

This concludes my article for this issue. The CP/M unit is enroute to Morning Star Software for PROM updates which make it compatible with CorComp's 9900 Disk Controller Card. Next month, I shall discuss this modification and its operation, additional information on CP/M, my experiences with PERSONAL PEARL, and some programs from Osborne Users Groups.

PASCAL NOTES

By Edgar Dohmann -- JSC Users Group (JUG)

This month we are going to take a look at the P-System Filer and a few related topics. The Filer manages the files on your P-System disks, manipulates them, and has features which allow you to check the surfaces of the disks for physical problems. The Filer is invoked by typing F while in the P-System command mode. Once in the Filer, the following commands are available:

B(ad-blks	Scans the disk for physical problem areas.
C(hng	Renames files on the disk.
D(ate	Allows you to set the date.
E(xt-dir	Catalogs the disk with more detail than L(dir.
G(et	Designates a file as the work file.
K(runch	Moves files on the disk as close together as possible.
L(dir	Catalogs the files on the disk.
M(ake	Creates a file.
N(ew	Clears the work file.
P(refix	Specifies a new default volume.
Q(uit	Exits the Filer.
R(em	Removes (deletes) files.
S(ave	Saves the work file with a name you specify.
T(rans	Transfers copies of your files from one place to another.
V(ols	Lists the system volumes that are on-line.
W(hat	Displays the name of the work file.
X(amine	Attempts to fix physical problem areas of the disk.
Z(ero	Initializes the directory.

The TI implementation of the P-System only supports a subset of the devices defined under the UCSD P-System. This explains why there are gaps in the list of supported devices. Devices #1 (CONSOLE), #2 (SYSTEM), AND #14 (OS) are always on line. Other device availability is determined by what you have installed on your system. The **V(ols** command in the Filer displays a list of those devices and their volume name which are currently available on your system.

The V command is handy to activate a disk drive that did not contain a diskette on power up or I(nitialization from the P-System OS. The OS checks for disk availability during initialization (which takes several seconds). If a disk is empty at this time, it cannot be used except by another initialization or by using the V command if you are in the Filer.

Device number or volume names may be used interchangeably. They are followed by a colon to distinguish them from filenames and to allow them to be used as prefixes for filenames. For example if your system disk is called P and it is in drive 1 (which is device #4) then the Filer file could be referenced as #4:SYSTEM.FILER or P:SYSTEM.FILER. I call my system disk P to minimize typing when I refer to it by volume name. Another handy feature is the ability to refer to the "root" volume (the one in device #4 when the system is booted) with an asterisk. Thus if my P disk is in device #4, I can refer to the Filer as *SYSTEM.FILER.

Another command that gets used often is **Z(ero.** This command initializes the directory of a disk. This command prompts you for the number of blocks on the disk. A P-System block is 512 bytes long which is 2 sectors in TI format. Thus the number of blocks is half the number of sectors on your disk so your answer will normally be 180, 360, or 720 depending on whether you are using single sided or double sided or double density disks.

Sectors 0 through 3 are used in standard TI format so the Disk Manager cartridge will recognize the disk. Sectors 4 through 11 (blocks 2-5) are used to store the P-System directory. If you want a duplicate directory it will be stored in blocks 6-9. There is space for 78 entries in the directory. The first is for the prefix which contains the name of the diskette, the last date entered by the D(ate command, and some other useful information. The other 77 spaces are for file entries.

File directories contain the starting and ending block of the file, the filename, the date the file was created (assuming you used the **D(ate** command), and some other useful information. Filenames may be up to 15 characters long and this includes the period if you use a file type specification. For example a filename of ABC.TEXT is 8 characters long. The P-System recognizes certain file specifications (TEXT, BACK, CODE, DATA, BAD) as special file types. You should only use these specifications for files as described in the Filer manual but you are free to create other specification identifiers for your own purposes.

Unlike the standard TI-99/4A File Manager which allows fragmented disk files, all P-System files must be stored in a contiguous set of blocks on the disk. If your disk has become fragmented because you have deleted and added files a lot, you may want to use the K(runch command to compress out any empty blocks between files. This will put all blank space at the end of the disk and possibly give you more room to add additional files.

The E and L commands provide a means of getting a disk catalog listed. If you use the Disk Manager to get a catalog it only finds one file called PASCAL. This is because it looks at the first 4 sectors which are not used by the P-System except to provide minimum compatibility with the Disk Manager. The Filer's E or L commands must be used to get a real catalog. If you want the catalog printed, put the device name after the volume name and a comma in response to the "Dir listing of ?" prompt. For example, an answer of P:,PRINTER: would dump the directory of my system disk called P to the printer.

These and other commands are all described in the Filer manual. I hope this discussion provides a little more insight into the operation of the P-System files. The last one I want to mention is the Q(uit command which returns you to the P-System OS.

SOME THOUGHTS.....

By REGENA

I have enjoyed visiting various Users Groups and meeting fellow TI-99/4 and TI-99/4A users. In general TI owners are very friendly and willing to help each other. Users Groups will probably be the main source of support for our computer since TI is no longer selling the computer. I hope the Users Groups can continue to communicate with each other and help everyone enjoy this great computer.

I have been very busy these past few months teaching a college course, Introduction To Computers, and really enjoying it. I've tried to show the class how we will be using computers in a practical sense -- no technical stuff. I think most people who take such a beginning course don't really want the boring textbook on how a computer works, but they want to get rid of their computerphobia and see how to use a computer. I've taken my TI several times for various demonstrations. One time was to show how easy it was to load a program -- just shove in a module! I was also able to show them some examples of good educational programs (thanks to the Scott, Foresman courseware).

I'm one of those people that believe this required computer literacy for our students should NOT be a one-semester course in programming but really should be part of the whole curriculum from kindergarten through senior high. I'd like to keep in touch with more of you teachers who are using computer in classrooms.

I got my first computer (TI-99/4) for Christmas of 1980. One of my first programs was putting some graphics with some music for "Auld Lang Syne". We then greeted the New Year with our computer instead of Guy Lombardo music. The next year I had Extended BASIC and had a bit more fun with using sprites for champagne bubbles and fireworks. Last year in COMPUTE! the 1983 versions were printed. For those of you who may want to update those versions, here is the 1984-85 version.

If you already have last year's program, load it in, then you just have a few minor changes. First, I made an error on one of the sounds. Change the first frequency in Line 1400 to 523. Next, instead of 1983 and 1984 we need to change the years to 1984 and 1985. The lines involved are the graphics commands in Lines 620-940. I also changed the ending slightly--Lines 2010-2040--so that you can press any key (instead of FCTN CLEAR) to end the program.

Best wishes to all of you for a nice holiday season--and hope you enjoy your TI throughout 1985!

```
100 REM AULD LANG SYNE
110 CALL CLEAR
120 CALL SCREEN(4)
130 CALL CHAR(96,"0001010303
07070F")
140 CALL CHAR(97,"0F1F1F3F3F
7F7FFF")
150 CALL CHAR(98,"FFFFFFFFFF
FFFFFF")
160 T=600
170 CALL SOUND(T*1.1,262,5)
180 CALL CHAR(104,"00030F1F3
F3F7F7F")
190 CALL CHAR(105,"7F7F3F3F1
F0F03")
200 CALL CHAR(106,"00C0F0F8F
CFCFEFE")
210 CALL CHAR(107,"FEFEFCFCF
8F0C0")
220 CALL SOUND(T*1.5,349,5,2
62,12,175,15)
230 CALL VCHAR(8,5,98,9)
240 CALL VCHAR(8,4,96)
250 CALL VCHAR(9,4,97)
260 CALL SOUND(T/2,349,5,262
,12,196,15)
270 CALL SOUND(T,349,4,262,1
2,220,15)
280 CALL CHAR(108,"7F3F1F070
F1F3F7F")
290 CALL CHAR(109,"FEFCF8C0F
0F8FCFE")
0,15,175,17)
1040 PRINT :::
1050 CALL SOUND(T,587,5,349,
12,175,15)
1060 CALL COLOR(9,5,1)
1070 CALL COLOR(10,5,1)
1080 CALL COLOR(2,7,1)
1090 CALL SOUND(T*1.5,523,5,
349,12,175,15)
1100 FOR I=5 TO 25 STEP 5
1110 CALL HCHAR(6,I,42)
1120 NEXT I
1130 CALL SOUND(T/2,440,6,26
2,15)
1140 CALL SOUND(T,440,6,349,
12,175,15)
1150 CALL HCHAR(4,13,42)
1160 CALL HCHAR(4,17,42)
1170 CALL HCHAR(2,11,42)
1180 CALL HCHAR(2,19,42)
1190 CALL SOUND(T,349,6,110,
18)
1200 CALL HCHAR(4,8,42)
1210 CALL HCHAR(2,6,42)
1220 CALL HCHAR(4,22,42)
1230 CALL HCHAR(2,24,42)
1240 CALL SOUND(T*1.5,392,6,
330,14,131,16)
1250 FOR I=5 TO 25 STEP 5
1260 CALL HCHAR(18,I,42)
1270 NEXT I
1280 CALL SOUND(T/2,349,6,29
```

```
300 CALL SOUND(T,440,5,349,1
2,175,15)
310 CALL HCHAR(8,10,98,3)
320 CALL HCHAR(8,9,104)
330 CALL VCHAR(9,9,98,3)
340 CALL SOUND(T*1.5,392,5,3
30,12,131,15)
350 CALL HCHAR(12,9,105)
360 CALL HCHAR(12,10,98,3)
370 CALL VCHAR(8,13,106)
380 CALL VCHAR(9,13,98,7)
390 CALL SOUND(T/2,349,5,294
,12,131,15)
400 CALL SOUND(T,392,5,330,1
2,131,15)
410 CALL HCHAR(16,13,107)
420 CALL HCHAR(16,10,98,3)
430 CALL HCHAR(15,9,98)
440 CALL HCHAR(16,9,105)
450 CALL SOUND(T,440,5,330,1
2,131,15)
460 CALL HCHAR(8,17,104)
470 CALL HCHAR(8,18,98,3)
480 CALL HCHAR(8,21,106)
490 CALL VCHAR(9,21,98,3)
500 CALL SOUND(T*1.5,349,6,2
20,12,175,15)
510 CALL VCHAR(9,17,98,3)
520 CALL HCHAR(12,17,108)
530 CALL HCHAR(12,18,98,3)
540 CALL HCHAR(12,21,109)
550 CALL SOUND(T/2,349,6,220
,12,175,15)
560 CALL VCHAR(13,17,98,3)
570 CALL SOUND(T,440,4,349,1
2,175,15)
580 CALL VCHAR(16,17,105)
590 CALL HCHAR(16,18,98,3)
600 CALL HCHAR(16,21,107)
610 CALL SOUND(T,523,3,349,1
0,175,13)
620 CALL VCHAR(13,21,98,3)
630 CALL HCHAR(8,25,97)
640 CALL VCHAR(9,25,98,5)
650 CALL SOUND(3*T,587,2,349
,8,233,10)
660 CALL HCHAR(13,26,98,4)
670 CALL VCHAR(11,28,98,6)
760 CALL SOUND(T,587,2,349,8
,233,10)
770 CALL SCREEN(8)
780 PRINT " ^b   hbbbj   hbb
bj   bbbbb"
790 CALL SOUND(T*1.5,523,3,3
49,10,220,13)
800 PRINT " ab   b   b   b
 b   b"
810 CALL SOUND(T/2,440,4,349
,12,175,15)
820 PRINT "  b   b   b   b
 b   b"
830 CALL SOUND(T,440,6,349,1
2,175,15)
840 PRINT "  b   b   b   b
 b   bbbbj"
850 CALL SOUND(T,349,6,220,1
2,175,15)
860 PRINT "  b   ibbbb   lbb
bm      b"
870 CALL SOUND(T*1.5,392,6,3
30,12,131,15)
880 PRINT "  b       b   b
 b      b"
890 CALL SOUND(T/2,349,6,294
,12,131,15)
900 PRINT "  b       b   b
 b      b"
910 CALL SOUND(T,392,6,330,1
2,131,15)
920 PRINT "  b   b   b   b
4,12,131,17)
1290 CALL SCREEN(8)
1300 CALL SOUND(T,392,7,330,
15,131,17)
1310 CALL HCHAR(20,13,42)
1320 CALL HCHAR(20,17,42)
1330 CALL HCHAR(22,11,42)
1340 CALL HCHAR(22,19,42)
1350 CALL SOUND(T,587,6,330,
14,131,16)
1360 CALL HCHAR(20,8,42)
1370 CALL HCHAR(22,6,42)
1380 CALL HCHAR(20,22,42)
1390 CALL HCHAR(22,24,42)
1400 CALL SOUND(T*1.5,523,6,
349,14,131,16)
1410 CALL HCHAR(4,3,42)
1420 CALL HCHAR(2,1,42)
1430 CALL HCHAR(4,27,42)
1440 CALL HCHAR(2,29,42)
1450 CALL COLOR(9,7,1)
1460 CALL COLOR(10,7,1)
1470 CALL SOUND(T/2,440,7,13
1,16)
1480 CALL SOUND(T,440,6,349,
14,175,16)
1490 CALL HCHAR(20,3,42)
1500 CALL HCHAR(22,1,42)
1510 CALL HCHAR(20,27,42)
1520 CALL HCHAR(22,29,42)
1530 CALL SOUND(T,523,5,220,
15)
1540 CALL SOUND(3*T,587,3,34
9,12,233,14)
1550 CALL COLOR(2,16,1)
1560 CALL SOUND(T,698,2,349,
13,233,15)
1570 CALL COLOR(2,12,1)
1580 CALL SOUND(T*1.5,523,3,
349,12,220,14)
1590 CALL COLOR(9,11,1)
1600 CALL COLOR(10,11,1)
1610 CALL SOUND(T/2,440,4,34
9,13,175,15)
1620 CALL SOUND(T,440,4,349,
13,175,15)
1630 CALL COLOR(2,5,1)
1640 CALL SOUND(T,349,5,262,
13,110,15)
1650 CALL COLOR(2,16,1)
1660 CALL SOUND(T*1.5,392,5,
330,13,131,15)
1670 CALL COLOR(9,14,1)
1680 CALL COLOR(10,14,1)
1690 CALL COLOR(2,7,1)
1700 CALL SOUND(T/2,349,5,29
4,13,131,15)
1710 CALL COLOR(2,16,1)
1720 CALL SOUND(T,392,5,330,
12,131,15)
1730 CALL COLOR(2,12,1)
1740 CALL SOUND(T/2,440,5,33
0,13,139,15)
1750 CALL COLOR(2,16,1)
1760 CALL SOUND(T/2,392,5,33
0,13,139,15)
1770 CALL COLOR(2,3,1)
1780 CALL SOUND(T*1.5,349,5,
294,14,147,16)
1790 CALL COLOR(9,16,1)
1800 CALL COLOR(10,16,1)
1810 CALL COLOR(2,16,1)
1820 CALL SOUND(T/2,294,6,22
0,14,175,16)
1830 CALL COLOR(2,6,1)
1840 CALL SOUND(T,294,7,233,
15,117,17)
1850 CALL COLOR(2,14,1)
1860 CALL SCREEN(11)
1870 CALL SOUND(T,262,7,165,
```

```
 b   b   b"
930 CALL SOUND(T,440,6,330,1
2,131,15)
940 PRINT "  b   ibbbk   ibb
bk   ibbbk"
950 CALL SOUND(T*1.5,349,6,2
94,12,147,15)
960 PRINT
970 CALL SOUND(T/2,294,7,220
,12,147,15)
980 PRINT
990 CALL SOUND(T,294,7,233,1
2,117,15)
1000 PRINT
1010 CALL SOUND(T,262,8,233,
14,131,16)
1020 PRINT
1030 CALL SOUND(3*T,349,8,22
15,131,17)
1880 CALL COLOR(2,12,1)
1890 CALL SOUND(4*T,349,6,22
0,15,175,17)
1900 CALL SCREEN(8)
1910 CALL COLOR(9,7,1)
1920 CALL COLOR(10,7,1)
1930 CALL COLOR(2,16,1)
1940 CALL COLOR(2,14,1)
1950 CALL COLOR(2,16,1)
1960 CALL COLOR(2,11,1)
1970 CALL COLOR(2,16,1)
1980 CALL COLOR(2,7,1)
1990 CALL COLOR(2,16,1)
2000 CALL COLOR(2,6,1)
2010 CALL KEY(0,K,S)
2020 IF S<1 THEN 1930
2030 CALL CLEAR
```

TIPS # 16 FROM THE TIGERCUB

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

These Tips are being mailed, together with my new catalog #5, to every Users Group that I know of. I hope that you will make both the Tips and the catalog available to your membership. I am sorry that I cannot take out paid ads in your newsletters, but to advertise in each one of them would cost me more than I have made in the past 6 months, and I would not get enough business to break even. If you would like to continue receiving these Tips, put me on the mailing list for your newsletter, and give me some indication that my Tips are really reaching your members and not going into someone's private file. If I receive enough business from this mailing to pay for its cost, I will then continue to send you my Tips. If not, this will be the last issue of the Tips from the Tigercub.

Copies of my catalog are available for $1.00, which is deductable from your first order. I have over 130 absolutely original quality programs in Basic, many of them now also available in XBasic, on casette or disk for only $3.00 each plus $1.50 per order for casette, package and postage, or $3.00 for diskette, package and postage (higher overseas). I give one-day service, I give bonuses for repeat orders, I give bonus programs on diskette orders. In addition, any User's Group member who mentions his/her users'group when sending me an order before 1 Jan. 1985 may deduct 10% from the cost of the programs.

Tips from the Tigercub #1 thru #14 are now available, with more added, as a diskfull of 50 programs, routines and files for only $15 postpaid. I have also now completed my NUTS & BOLTS disk of 100 XBasic utility subprograms in MERGE format, ready to merge into your own programs, for just $19.95 postpaid.

In the last Tips, I mentioned that I wished I knew who to credit for that remarkable routine to redefine the cursor. Dave Peden has written me that credit should be given to Terry L. Atkinson of 28 Savona Ct., Dartmouth, NS B2W 4R1 CANADA.

And I would like to strongly recommend that you support the 99'ers Users Group Association, 3535 So. H St., #93, Bakersfield CA 93304. They are a strictly non-profit group, devoting a lot of time and effort into helping us all, and they publish a great newsletter..

Every Tips must include a bit of music, and my grandson has requested that I pass this one on to all other two-year olds.

```
100 !ALPHABET SONG - by Jim
Peterson
110 DIM N(21)
120 CALL MAJORSCALE("C",N())
130 CALL SCREEN(5):: DISPLAY
 AT(24,1)ERASE ALL:"READY -
TYPE THE ALPHABET" :: CALL M
AGNIFY(2)
140 CALL KEY(3,K,ST):: IF (S
T<1)+(K<65)+(K>90)THEN 140 :
: CALL SPRITE(#1,K,16,96,120
):: IF K=87 THEN GOSUB 220 E
LSE GOSUB 200
150 IF (K=90)*(FLAG=0)THEN 1
60 ELSE 140
160 FLAG=1 :: M$="C115566D5C
443322D1" :: T=150
165 FOR J=1 TO 18 :: CALL SP
RITE(#J,64+J,INT(11*RND+6),9
6,128,J*5,J*5)
170 X=ASC(SEG$(M$,J,1)):: IF
 X>58 THEN T=150*(X-64):: GO
TO 190
180 X=X-48 :: CALL SOUND(T,N
(X),0)
190 NEXT J :: FLAG=0 :: CALL
 DELSPRITE(ALL):: GOTO 140
200 Y=VAL(SEG$("115566544332
22215543325332",K-64,1))
210 CALL SOUND(500,N(Y),0)::
 RETURN
220 CALL SOUND(500,N(5),0)::
 CALL SOUND(500,N(5),5):: CA
LL SOUND(500,N(4),0):: RETUR
N
230 SUB MAJORSCALE(K$,N())
240 F=VAL(SEG$("110123131147
165175196",POS("ABCDEFG",K$,
1)*3-2,3))
250 C$="10101101010110101101
0101101011010101"
260 FOR J=1 TO 36 :: IF SEG$
(C$,J,1)="0" THEN 280
270 X=X+1 :: N(X)=F*1.059463
094 J
280 NEXT J :: SUBEND
```

Lines 230-280 of that routine are an example of the kind of handy-dandy subprograms you will find on my Nuts & Bolts disk.

We haven't had a Tigercub Challenge for some time, so . . . How can you store a hundred or more values of any size, positive or negative, integer or non-integer, even in exponential notation, without dimensioning an array or opening a file? Now, how can you link your program to another by a RUN statement, thereby losing all data, and recover those values? Yes, I know you can save them on the screen and read them back, but can you find a better way?

Here's a little demo program of how motion can be created by the repetitive redefinition of characters. I call it ETERNITY.

```
100 !ETERNITY! - by Jim Pete
rson
110 CALL CLEAR :: CALL SCREE
N(2):: CALL COLOR(1,16,1)::
CALL CHAR(33,"",34,"",35,"",
36,"")
120 FOR R=1 TO 12 :: CALL HC
HAR(R,R+4,33,26-R*2):: NEXT
R
130 FOR R=13 TO 24 :: CALL H
CHAR(R,29-R,34,(R-12)*2):: N
EXT R
140 FOR C=5 TO 16 :: CALL VC
HAR(C-4,C,35,34-C*2):: NEXT
C
150 FOR C=17 TO 28 :: CALL V
CHAR(29-C,C,36,C*2-33):: NEX
T C
160 FOR J=0 TO 7 :: A$(J+1),
B$(8-J)=SEG$("00000000000000
",1,2*J)"FF" :: NEXT J
170 C$(1),D$(8)=RPT$("80",8)
:: C$(2),D$(7)=RPT$("40",8):
: C$(3),D$(6)=RPT$("20",8)::
 C$(4),D$(5)=RPT$("10",8)
180 C$(5),D$(4)=RPT$("08",8)
:: C$(6),D$(3)=RPT$("04",8):
: C$(7),D$(2)=RPT$("02",8)::
 C$(8),D$(1)=RPT$("01",8)
190 FOR C=2 TO 15 :: FOR J=1
 TO 8 :: CALL CHAR(33,A$(J),
34,B$(J),35,C$(J),36,D$(J)):
: NEXT J :: CALL SCREEN(C)::
 NEXT C :: GOTO 190
```

Next, I would like to share with you a gem of a "why didn't I think of that" routine which John Taylor sent me.

```
100 ! 28 COLUMN TEXT ROUTINE
     ////////////////////////
110 ! EXTENDED BASIC (EASILY
     CONVERTED TO TI BASIC)
     ////////////////////////
120 ! WRITTEN BY JULIE PACK
     B.  U.  G.
     P.O.  BOX 1402
     PALM BAY, FL 32906
     ////////////////////////
130 ! ENHANCED BY JET
     SHOALS 99'ERS
     P.O.  BOX 2928
     MUSCLE SHOALS,AL 35662
     ////////////////////////
140 CALL CHAR(64,"00282828")
150 ! PROGRAM TO COPY STARTS
     HERE
160 CALL CLEAR :: X=-1
170 RESTORE
180 IF X>=21 THEN X=1 :: CAL
L WAIT
190 READ MESS$
200 IF MESS$="P" THEN DISPLA
Y AT(X+2,1):Z$ :: X=X+4 :: Z
$="" :: GOTO 180
210 IF MESS$="ZZZ" THEN DISP
LAY AT(X+2,1):Z$ :: CALL WAI
T :: END
220 IF LEN(Z$)>0 THEN MESS$=
Z$" "&MESS$
230 X=X+2
240 IF X>=21 THEN X=1 :: CAL
L WAIT
250 IF LEN(MESS$)<29 THEN DI
SPLAY AT(X,1):MESS$ :: Z$=""
:: GOTO 180
260 FOR A=1 TO 29
270 I=POS(MESS$," ",A)
280 IF (I=0 OR I>29)AND A=1
THEN A,J=29 :: GOTO 310
290 IF I=0 OR I>29 THEN A=29
:: GOTO 310
300 J,A=I
310 NEXT A
320 IF X>=21 THEN DISPLAY AT
(X,1):SEG$(MESS$,1,J-1):: X=
-1 :: CALL WAIT :: GOTO 340
340 DATA "THIS ROUTINE
 WILL ENABLE YOU TO WRITE LO
NG TEXT MATERIAL IN YOUR DAT
A STATEMENTS SO YOU WON'T HA
VE TO WORRY ABOUT COUNTING"
380 DATA "THE LENGTH OF YOUR
 SENTENCES ALL THE TIME.  TH
  IS ROUTINE WILL AUTOMATICALL
Y EDIT YOUR TEXT TO FIT A 28
 COLUMN SCREEN."
390 DATA "A SUGGESTION- IT I
S A GOOD IDEA TO PUT A QUOTE
 AT THE BEGINNING AND END OF
  THE DATA STATEMENTS SO YOU
WON'T HAVE TO WORRY ABOUT"
400 DATA "COMMAS LIKE THIS ,
,, AND THEY WILL REMAIN IN Y
OUR TEXT PROPERLY."
410 DATA "THIS ROUTINE WILL
ALSO CLEAR THE SCREEN (WHEN
FILLED) AND CONTINUE READING
 YOUR DATA AND DISPLAYING YO
UR TEXT ON THE NEXT SCREEN."
420 DATA P
430 DATA " TO START A NEW P
ARAGRAPH ENTER THE LETTER @P
@ AS A SEPERATE DATA STATEME
NT, THEN INDENT YOUR TEXT ON
 YOUR NEXT NEXT DATA"
440 DATA "STATEMENT 2 OR 3 S
 PACES (IF DESIRED).",P,"TO S
KIP LINES,",P,"JUST ENTER @P
@",P,"WHERE EVER YOU WANT TO
",P,"SKIP."
450 DATA P,"MAKE SURE THAT Y
OUR VERY LAST DATA STATEMENT
  IS @ZZZ@, AND JUST REPLACE
THESE DATA STATEMENTS WITH"
460 DATA "YOUR OWN.",P,"YOU'
LL ALSO FIND THIS ROUTINE IS
 MOST USEFUL WHEN CONCATENAT
ING STRINGS, E.G., @ELIZA@ T
YPE PROGRAMS-",P
470 DATA "AN EXAMPLE:",P,"A$
=@JACK AND JILL WENT UP@","B
$=@THE HILL TO FETCH A@","C$
=@PAIL OF WATER.@","D$=A$B$
&C$&D$","PRINT D$",P
480 DATA "JACK AND JILL WENT
```

```
330 DISPLAY AT(X,1):SEG$(MES
S$,1,J-1)
340 IF SEG$(MESS$,J,1)=" " T
HEN I=1 ELSE I=0
350 Z$=SEG$(MESS$,J+I,163)::
MESS$=Z$ :: IF LEN(Z$)>28 T
HEN X=X+2 :: GOTO 260
360 GOTO 180
370 DATA "THIS SHORT ROUTINE
 UP THE HILL TO FETCH A PAIL
 OF WATER.",P,P,P,"HAPPY PRO
GRAMMING!"
490 DATA ZZZ
500 SUB WAIT
510 DISPLAY AT(24,8):"PRESS
 ANY KEY"
520 CALL KEY(0,K,S):: IF S=0
 THEN 520 ELSE CALL CLEAR
530 SUBEND
```

Thank you, Julie and John. This is becoming one of the most useful routines on my utility disk. I was preparing a disk of PD programs for our U6 library. Some of them needed extra instructions, so I typed them out on TI-Writer, so that people could run them off on their printer. Then I remembered that some folks don't have printers. So -

```
50 CALL CLEAR :: INPUT "FILE
NAME? DSK1."?:F$
60 DIM B$(150):: OPEN #1:"DS
K1."&F$,INPUT, DISPLAY ,VAR
IABLE 80
70 A=A+1 :: LINPUT #1:B$(A)
80 IF EOF(1)=1 THEN B$(A+1)=
"ZZZ" ELSE 70
```

and change line 170 to - 170 @@=@@+1 :: MESS$=B$(@@)

And there you have a quickie program to check out those DIS/VAR 80 files that show up on your disks under filenames that you can't remember using.

MEMORY FULL IN LINE 32767

THE DISK DRIVER

By Curt Purdy - Valley 99'ers

My subject this time is the basics of the disk system. This article applies to single-side, single-density disk systems represented by SS/SD hereafter. The first thing you notice when you go from cassette to disk is that your larger programs may cause a "memory full" error when you run them. This is a result of the disk controller card using 2088 bytes of console memory. The unit requires 534 bytes of memory for its operating system and 518 bytes for each open file. The default, or standard number of open files, is three (3) which requires 1564 of memory.

You will not encounter the memory full message if memory expansion is attached and you are using Extended Basic (X/B). I know, you do not feel like converting all your large Basic files to run in X/B. You CAN free up over 1 K of memory by using the statement CALL FILES(1). Type new immediately after you enter the CALL FILES(1) command. This leaves one (1) of the files mentioned earlier open to operate a single disk drive. If this is still not enough, there are 2 CALL LOAD tricks which require Expansion Memory and the Mini-Memory module. Since this is complicated, I'll suggest you simply purchase X/B instead.

Some of the advantages of disks are that you can:

1) Access programs and more importantly, files by name which permit you to do things like load a file, change the format and rewrite it as a totally different file. This even allows you take a program, convert it to a file for transmission over a modem and then reconvert it back to a program to be used on the other end. (More on the details of this procedure in a future issue.)

2) Run another program totally different from the program currently running in the computer. (A poor-man's CHAIN command). On the last line of the program, enter RUN DSK1.filename and the next program (or next part of your program) will be loaded and run. Although it does not set aside a buffer for transferring variables as in a CHAIN command, your variable values will still be in VDP RAM and can be used by the next program part if you are careful not to reinitialize them.

3) Have several input and output files open at one time which is essential in data manipulation.

4) Use relative files which are essential in database use.

5) Use old disks as frisbees.

Cassette users should despair if you do not have a disk drive. Very few /4A owners started with a disk system. I used cassette for two (2) years prior to obtaining a disk system. You can utilize 90% of what the 99/4A is capable of preforming without owning a disk system. GOOD DRIVING!! Curt Purdy

YOUR OWN SPRITE AUTO MOTION

More on Bit-Map and Interrupts
By John Phillips

Some of my past articles for you die-hard /4A users concerned VDP Interrupts and Bit-Map mode. This month's article concerns a boundary which affects both the previous articles . . . sprite auto-motion.

For those of you who are in love with auto-motion, I'm sure it was a disappointment to read that sprite auto-motion is not available in bit-map mode. Well, I have the answer to your prayers.

With the capability of the interrupt structure, it is possible to write your own VDP interrupt-driven sprite auto-motion. Just because the Editor/Assembler manual says you can't doesn't really mean you can't!

What I have done is translated the GPL interpreter code for sprite auto-motion from GPL into 9900 Assembly code. All you need to do is attach this code to the interrupt cycle (as discussed in a previous article) and define a VDP area for your Sprite Velocity Table. Sounds simple enough?

I have listed a short program below to demonstrate a user-defined auto-motion. You must remember to disable the system auto-motion (as previously discussed) **before** attempting to run your auto-motion code. If you don't, strange things can happen. The code below sets the new interrupt vector, disables system sprite auto-motion, and creates a sprite which is place in a positive Y and X direction. If you have any questions on interrupt structure, refer back to my first article.

Once you have the code successfully running, place the auto-motion subroutine in your subroutine library. Then you can have auto-motion any time you want! Even in bit-map mode! Have fun with your new-found experiment.

. . . John Phillips

```
       DEF AUTO
       REF VMBW,VWTR
********************************************************************************
* R O M   S E T - U P   F O R   S P R I T E   A U T O   M O T I O N      *
*  . . . INTERRUPT DRIVEN, THAT IS!!                                           *
*   VDP REGISTERS FOR BIT-MAP MODE SHOULD BE SET AS FOLLOWS:                   *
*       0 -  >02                                                               *
*       1 -  >E#  # - 0=SS SPRITES 2=DS SPRITES 1=SS MAG 3=DS MAG              *
*       2 -  >06                                                               *
*       3 -  >FF                                                               *
*       4 -  >03                                                               *
*       5 -  >36                                                               *
*       6 -  >03                                                               *
*       7 -  >0#  #-BORDER COLOR (0-F)                                         *
********************************************************************************
VDPREG BYTE >02,>E0,>06,>FF,>03,>36,>03,>01
SALDAT BYTE >60,>80,>80,>0F,>D0
SDLDAT DATA >FFFF,>FFFF,>FFFF,>FFFF
SVTDAT BYTE 10,10
*
PDT    EQU >0000     PATTERN DESCRIPTOR TABLE
SIT    EQU >1800     SCREEN IMAGE TABLE
SAL    EQU >1B00     SPRITE ATTRIBUTE LIST
SDL    EQU >1C00     SPRITE DESCRIPTOR LIST
SVT    EQU >1F00     SPRITE VELOCITY TABLE
FREVDP EQU >1F80     128 BYTES FREE HERE
CT     EQU >2000     COLOR TABLE
INTWS  EQU >83C0     INTERRUPT WORKSPACE
GPLWS  EQU >83E0     GPL WORKSPACE
DISINT EQU INTWS+2   INTERRUPT DISABLE WINDOW
EXTINT EQU INTWS+4   EXTERNAL INTERRUPT ROUTINE ADDRESS
@SAML  EQU RSMOT-SAL FOR AUTO MOTION ROUTINE
GR8LB  EQU GPLWS+17  GPL R8 LOW BYTE

MYWS   EQU >8300     MY WORKSPACE
MOTION EQU >837A     # OF SPRITES TO MOVE
*********************************
* START OF MAIN-LINE 9900 CODE  *
*********************************
       EVEN
AUTO   LWPI MYWS     LOAD MY WORKSPACE
       LIMI 0        INTERRUPTS OFF FOR NOW
       LI R0,SPMOVE  ADDRESS OF MY INTERRUPT ROUTINE
       MOV R0,@@EXTINT HAVE THE ADDRESS SET
       LI R0,>4000    SHUT OFF SYSTEM AUTO-MOTION
       MOVB R0,@@DISINT NOW AUTO-MOTION IS SHUT OFF
*
       CLR R0        START WITH VDP R0
       CLR R1        FIRST BYTE OF DATA
REGL   SWPB R0       SWITCH HI/LO
       MOVB @@VDPREG(R1),R0 GET VDP REGISTER DATA
       SWPB R0       RESTORE
       BLWP @@VWTR   WRITE THE REGISTER
       AI R0,>0100   NEXT VDP REGISTER
       INC R1        POINT TO NEXT DATA
       CI R1,8       DONE?
       JNE REGL      NOT YET
```

```
@ WE ARE NOW IN BIT-MAP MODE
       LI R0,SDL     MAKE A SPRITE PATTERN
       LI R1,SDLDAT  DATA
       LI R2,8       #OF BYTES TO WRITE
       BLWP @@VMBW   WRITE THE SPRITE DATA TO VDP
*
       LI R0,SAL     NOW PUT SPRITE ON SCREEN
       LI R1,SALDAT  DATA
       LI R2,5       INCLUDE >D0 TO DISABLE REST
       BLWP @@VMBW   SPRITE IS NOW ON SCREEN
*
       LI R0,>0100   1 SPRITE IN MOTION
       MOVB R0,@@MOTION ALLOW 1 SPRITE TO MOVE
*
       LI R0,SVT     NOW SET SPRITE IN MY AUTO-MOTION
       LI R1,SVTDAT  DATA
       LI R2,2       2 BYTES TO WRITE
       BLWP @@VMBW   SPRITE IS IN AUTO-MOTION, ALMOST!
*
       LIMI 2        AUTO-MOTION CANNOT WORK WITHOUT INTERRUPTS!
       JMP $         QUIT KEY WILL FUNCTION WITH INTS ON, TOO!
*****************************************************************
* NOW THE AUTO MOTION CODE AS COPIED FROM GPL INTERPRETER CODE   *
* REMEMBER YOU ARE IN THE GPLWS AT THIS POINT. REGISTERS 13, 14 AND *
* 15 MUST BE PRESERVED AS THESE ARE THE LINK BACK TO THE INTERRUPT  *
* HANDLER ROUTINE.                                              *
*****************************************************************
SPMOVE LI R10,>8C02                    JH ONSCRN
       MOVB @@MOTION,R12               MOV R5,R5
       JEQ NOMOT                       JGT $+6
       SRL R12,8                       AI R4,>C000
       LI R2,>8800                     AI R4,>2000
       LI R3,>8C00             ONSCRN CLR R6
       LI R8,SVT                       MOVB *R2,R6
MLOOP MOVB @@GR8LB,*R10                A R7,R6
       MOVB R8,*R10                    ORI R8,>4000
       CLR R4                          MOVB @@GR8LB,*R10
       MOVB *R2,R4                     MOVB R8,*R10
       CLR R6                          MOVB R4,*R3
       MOVB *R2,R6                     AI R8,@SAML+2
       SRA R4,4                        MOVB R6,*R3
       MOVB *R2,R5                     SWPB R5
       SRA R5,4                        MOVB @@GR8LB,*R10
       A R4,R5                         MOVB R8,*R10
       MOVB *R2,R7                     SRL R5,4
       SRA R6,4                        MOVB R5,*R3
       SRA R7,4                        SWPB R7
       A R6,R7                         SRL R7,4
       AI R8,-@SAML                    MOVB R7,*R3
       MOVB @@GR8LB,*R10               AI R8,>C002
       MOVB R8,*R10                    DEC R12
       CLR R4                          JGT MLOOP
       MOVB *R2,R4             NOMOT  LWPI INTWS
       A R5,R4                         RTWP
       CI R4,>C0FF             *
       JLE ONSCRN                      END
       CI R4,>E000
```

REVIEWS

The Bugout Machine Language Monitor

Reviewed by J. W. Vincent

Whether your an assembly language greenhorn, just learning your way around, an experienced hacker, or old pro, this program is for you! **Bugout** is a fantastic new tool for all 99/4A machine language users. It's primary purpose is to help you chase down those elusive **bugs** and get them **out** of your programs but, it's also a very effective way to tap the innermost secrets of the 99/4A's operation or to learn exactly what happens when your machine code executes. Bugout provides an environment and host of functions unriveled by the previously available debugging tools. Bugout provides:

* Four screen "windows" containing:
  - 8 lines of HEX/ASCII dump
  - 9 lines of 9900 machine code disassembled to the proper mneumonics
  - Status/error message area
  - Command input line

* Disassembly or HEX dump output to printer or disk as well as screen with arrow key controlled scrolling
* Four software implemented code execution modes thru RAM or ROM based code:
  Single step, Multi-step, Emulation or Breakpoint execution
* Memory examine/change for CPU RAM/ROM, VDP RAM, GROM/GRAM, or CRU with screen oriented editing
* Memory block moves for CPU RAM/ROM, VDP RAM, or GROM/GRAM
* Memory searches for hex values or literal strings in CPU RAM/ROM, VDP RAM, or GROM/GRAM
* Set Device command for "user-transparent" access to DSR ROMs (addresses >4000 to >5FFF)
* Object Code Loader (your routine gets loaded by Bugout)
* Address input/output in hex, decimal or labeled (assuming label is in REF/DEF) table
* Instant viewing of current workspace registers
* Keystroke generated breakpoints during execution
* Many additional features and functions

Bugout is even user extensible! The author has included a patch file and source examples of how to add your own features to bugout (although I'm pressed to think of what else you could want). My only complaint with Bugout is it's size, it uses the E/A utility routines and CPU RAM >A000 thru >D600 and a stack in VDP RAM at >37D7. If you are writing a large routine you must code and test it in aprox. 8K modules of relocatable code. Other than that, all I can do is rave about how wonderful this program is. I even like the little display it puts on if you try to run a bad copy. Even though its copy protection isn't very tough to get around, don't pirate it, this program is well worth its cost.

The price of $39.95 includes Bugout on disk, a 54 page manual, shipping and handling. If your into assembly language, order it today! Bugout was written by Gregg Wonderly (a senior in computer science) and is marketed by:

The Data Process
PO BOX 1818
Bethany, Oklahoma 73008
(405)728-0158.

FLOPPY-COPY

By John Phillips

A young gentleman by the name of Chris Faherty contacted me a few months back about a product he had created entitled "Floppy-Copy", a disk duplication program. Now, there are several duplication programs on the market, but he insisted that his was better and wanted me to take a look at it. Well, I did get the program, and have looked at it in depth. The product is interesting, so I thought I would do a review on it!

First of all, **Floppy-Copy** is written in TMS9900 Assembly Language, so it requires the use of the TI disk system and the TI memory expansion. This tends to be my biggest complaint about the product. Those of us owning Disk Manager Modules are accustomed to popping in that module and doing all kind of disk access without the need for extra RAM. **Floppy-Copy** does not provide us with such luxuries.

The load time for **Floppy-Copy** is considerable, not to mention dangerous to the disk drive itself. A very sophisticated copy protection is installed which causes the disk drive head to scramble a minimum of 5 times before the program is loaded. This is a **CARDINAL SIN** in my book. I used to use a similar protection on my products, but stopped to due to complaints of drive wear and tear. I, myself, wore out a drive creating this particular type of copy protection method (fortunately, it was TI's equipment).

Once the program does boot, however, you will be pleasantly surprised at its functionality and user-friendliness. An adequate title screen is displayed, followed by a help screen. A prompt asks you if you want help. By pressing the numbers 1 through 4, you are guided through a set of instructions to get yourself up and running.

The instruction or help files are probably the nicest features of the entire product. If you choose a help file, the program loads several screens full of information into auxiliary memory. You read these instructions at your own pace using the arrow keys. What is so neat is that the instructions scroll on the screen depending on which direction you are moving it. You may stop the scroll at any point by simply releasing the arrow key. Nice touch!

The meat of **Floppy-Copy** really consists of three functions:

1) CATALOG

2) BACKUP

3) INITIALIZE

The CATALOG and INITIALIZE work just as the Disk Manager Module does with no major exceptions. It is the BACKUP, however, which adds the "umph" to the product.

Floppy-Copy copies a disk by sector, rather than by file. However, it is very smart in that it only copies the sectors that are used and leaves the blank sectors alone. This creates a very fast, efficient copy of a disk. You may, of course, copy every sector of a disk, but that is just wasted time. They use the term "proportional" to refer to the smart copy.

I did a side-by-side comparison of Disk Manager and **Floppy-Copy** as far as speed goes. **Floppy-Copy** cataloged a disk in 5.69 seconds, initialized a double-sided disk in 34.63 seconds and backed up a single-sided diskette in 1 minute, 13.55 seconds. Disk manager cataloged a diskette in 3.66 seconds (better), initialized a double-sided diskette in 1 minute, 45.08 seconds, and backed up a single-sided diskette in 1 minute, 51.72 seconds.

As you can see, the disk manipulation routines are much faster using Floppy-Copy, but I am not satisfied. As I mentioned earlier, the load time is annoying and dangerous. For my money, I'll stick with my Disk Manager Module.

The documentation is poor, but the built-in help screens more than make up for the lack of hard-copy documentation. There is no way to transer the instruction screens to your printer, so you'll need a good memory.

RATING: *** (out of 5)

Floppy-Copy
The Softspot
P.O. Box 8786
Silver Springs, MD 20907
301-439-8084

ARTS & GRAPHICS with your TI-99/4A
By Thomas A. Thompson - $4.95
Published by Hayden Books
10 Mulholland Dr.; Hasbrouck Heights, NJ 07604

By Darrell Ingold

This paperback book is comprised of 25 graphic programs ranging from a simple screen-sized American Flag to some games. The appeal for this type of programming would probably be limited to those people who would like to see something colorful and interesting on the screen but have not yet reached the programming ability level to accomplish it. While there is some very limited reference to the purpose of each section of the program, the learning application is limited to simply "copy and compare". One may be able to learn how to do many of the graphic features but not without considerable observation and study of the programs as he types, runs and compares the results.

I believe that there are many youngsters who would get a great deal of enjoyment out of this type of programming but I don't feel the buyer should expect any more out of the book than that; but then for the price it could be well worth it to the novice.

FUN & GAMES with your TI-99/4A
By Stephen M. Muncy - $4.95
Published by Hayden Books
10 Mulholland Dr.; Hasbrouck Heights, NJ 07604

By Darrell Ingold

In spite of its name this book is not all fun and games; It takes a serious approach to teaching the needed skills and concepts for worthwhile game programming. I don't mean to imply that the book is not fun. To those who like to play games on their computer as well as program this book is the answer. What I mean by a "serious" approach is that the book does not assume you know all the basics. The first section is devoted to a review of several of the TI Basic commands (those most needed in the book's programs). There is then a section explaining the concept of "How To" develop an arcade style game, followed by a review of graphics and movement in TI Basic using CALL HCHAR, VCHAR & CALL KEY to name a few. The art of "Crashing" & scoring points are also covered.

After covering all these basics the reader is lead by the hand through the development and programming of Meteor Maze and given several miniprograms with which to practice. These miniprograms are designed to be used in larger game programs that the reader may wish to develop later. The remainder of the book consists of 11 games for programming that are relatively sophisticated, including Chopper Rescue and Biorhythms. All considered, the book, which is paperback, is well designed and balanced with teaching and practical application. An excellent investment for the computer game addict!

NUTS & BOLTS (SOFTWARE)

Price:$15.00 FROM:
TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

By Darrell Ingold

I have just spent all evening literally playing with this terrific disk. Although I have never even talked to Jim Peterson it is obvious that he has quite an active imagination and is exceptionally creative as demonstrated in the myriad of over 100 programs that he has produced on NUTS & BOLTS. It is a programmer's delight to see all the nifty programs just waiting to be merged into one's own creative efforts to "polish" them up to perfection! All of the programs on the disk are saved in the merge format (extended basic required) and formated as subprograms. In addition, they are all consecutively numbered beginning at 20000 so that they will not interfere with each other or with most existing programs; several of these subprograms can be used in the same program.

The programs are divided into 15 general catagories: type fonts, text displays, pauses, programming aids, data saving & reading, displays, time & date, music, sorts & scrambles, printer aids, keyboard & joystick controls, math, protection, screenwipes, & miscellaneous. Under the heading of Displays for example is Flag; a beautiful job of a large American Flag. Spriteshow is just exactly that; 10 randomly designed sprites floating all over the screen. Like to type in Russian? No problem. Under Type Fonts you will find Russian as well as Bigchar, Scrunch, Slant & Upside Down, all self-explanatory. Would you like your screen title to appear in large colorful print diagonally across the screen or your instructions to jiggle, flash or fade-out? Then Jim's got the program for you. Want to add a little closing class? Try one of the various screenwipes available in a multitude of colors.
Without a doubt, NUTS & BOLTS has the greatest variety of "nuts and bolts" to put a really professional program together, all on one disk, of anything I have yet seen. A real bargain at $15.00 (and I'm from a long line of penny pinchers!)

HINTS 'N TIPS

TOURE DE BOARDS

By Ron Albright, M.D. -Valley 99'ers

WELCOME TO THE SPIRIT OF 99 TEXAS INSTRUMENTS HOME COMPUTER BBS COLUMBUS OHIO 614-451-0880 24 HOURS 300 BAUD

[C]ALL LOADs And Other Fun Things To Do!

These are CALL LOAD(ADDRESS,VALUE)

Highest Srite In Motion *(-31878,HI6H)
Disable Sprites,Sound,QUI(-31806,128)
Disable Sprites & Sound (-31806,96)
Disable Sprites & QUIT K (-31806,80)
Disable Sprite Motion (-31806,64)
Disable Sound & QUIT Key (-31806,48)
Disable Sound (-31806,32)
Disable QUIT Key (-31806,16)
Enable Above Functions (-31806,0)
RUN "DSK1.LOAD" (-32729,0)
ON BREAK GOTO(Sys Lockup)(-31868,0)
Speed up Sound & Cursor *(-31748,0-255)
Disable Disk Drive *(-31888,63,255)
Enable Disk Drive *(-31888,55,215)
Restart XBasic W/Load Pgm(-31962,255)or
Press Key For Load (-32961,149)
Go from XBasic to Basic (-32116,4)
Sound Registers 0-255 *(-31740,A,B)
Vary Keyboard Response *(-31572,0-255)
Clear Screen For Instant (-32700,0)
Reset To Title Screen (-32730,32) or
CALL PEEK(2,A,B):: CALL LOAD(-31804,A,B)
or (-32961,51) or *(-32630,128)

These Are CALL POKEV(ADDRESS,VALUE)

VDP Modes Are Not Usable From These

Text Mode (-32272,0,"",-30945,0)
Multi-Color (-32280,0)
Bit Map (-32766,0)
Graphics (Normal Mode) (-32768,0)
Blank Screen,Key Restores(-32352,107)

These are CALL PEEK(ADDRESS,VARIABLE)

Random Number Generator0-99(-31880,NUM) VDP Interrupt Timer *(-31879,TIME) Sprite Coincidence, HIT=32 (-31877,HIT) Double Random Number 0-255 (-31808,A,B) Find Speech Synth., YES=96 (-28672,YES) Memory Size *(-31974,A,B)

In Console Basic with the Mini Memory or Editor/Assembler module, type these commands in to get the amount of memory you have left.

CALL PEEK(-31974,A,B) PRINT A6+B-1776

VDP Interrupt Timer value. The timer counts 1 to 100 every 1/30th of a second.

CALL PEEK(-31879,A)

The sound registers may be randomly abused to get different sounds and they will stay on until another sound is made by the normal methods (an error, input beep CALL SOUND, etc.). A and B are values from 0-255. Have fun...

CALL LOAD(-31740,A,B)

Keyboard response can be varied by using different values in this location.

CALL LOAD(-31572,0) Normal
CALL LOAD(-31572,1) Mushy
CALL LOAD(-31572,128) Lock Up

Return to the master title screen without graphics. You can use the commands normally, you just can't see them.

CALL LOAD(-32630,128)

Change the flash rate of cursor. This could be very handy for a program that uses lots of editing because you can see what is hiding under the cursor.

CALL LOAD(-31748,N) N=0-255
1 is the normal setting

So Your Basic Program Doesn't Load Eh?

I bet you discovered that when you upgraded to disk that some of your large programs just would not load, and they gave you all kinds of nasty errors. That is becouse your disk memory system uses around 2KB of console RAM. You can disable the disk system, and regain the lost memory by entering:

CALL LOAD(-31888,63,255)

and then NEW From Extended Basic you must do CALL INIT, but you can also restore the disk system by entering:

CALL LOAD(-31888,55,215)

and then NEW. You can call this from Basic when the the Editor/Assembler or Mini Memory modules are plugged in, but you can only restore the disk system by doing a BYE

While the sprites are disabled, other sprite functions still work.

CALL LOAD (-31878,0)

also stops sprites. Loading the highest numbered sprite in this address restarts them, or you can restart them selectively by number. If you are using the original version of Extended Basic

CALL VERSION(X) :: PRINT X where X=100]

then it behooves you to stop the sprites if you are not using them, as the old version constantly tries to update them, and you will notice a signi- ficant increase in speed if you turn the sprites off.
Concerning Pokes at Protection

The Central Ohio Ninety Niners does not support software piracy. As the master control program from Tron would say. . .

END OF LINE ##

HOW TO FIX DISKS
By Niraj N. Shah and Mike Ballman

This article originally appeared in the Spirit Of 99, the Newsletter & Bulletin Board of The Central Ohio Ninety Niner's The Board Number Is 614-451-0880 Did you ever try to catalog a disk and find out the Disk Controller thinks the disk is NOT initialized? But you know better! What do you usually do with the blown disk? Most people Delete the file giving them the problem. Usually that does correct the problem, but it also gets rid of that file forever. The ultimate solution is to use DISK FIXER by Navarone Industries.

The DISK FIXER enables one to examine and change the contents of any disk on a a sector-by-sector basis. I think it is worth its forty-dollar list price. It is available from some TI retailers or directly from Navarone Industries.

Here is the process to fix a blown-up disk...

First acquire a DISK FIXER from a friend or buy one, they're worth it. Get a hardcopy catalog of the blown disk, or even better, get a complete (old) catalog of what should be on the disk. If a complete catalog is not available try to remember what should be on the disk and write those names down on paper. Once you have a catalog of the disk, you are ready to start using DISK FIXER.

Insert the DISK FIXER cartridge and select option 2 from the Title Screen. Upon doing so you should see the DISK FIXER menu. Do the following if the most recent catalog of the bad disk tells you there are more sectors used/free than is logically posssible: 358 for single sided disks & 718 for double sided disks. For example, if the catalog lists 500 sectors used/free on a single-sided disk THEN do the following ELSE GOTO the paragraph on "SECTOR ONE".

This part tells you how to fix up Sector 0; which is the sector containing the information concerning the disk name and number of sectors used/free on the disk. If the disk catalog tells you the used/free sector information is in error then Sector 0 needs to be fixed. The easiest way to do this is to copy a good Sector 0 from another disk to the blown disk. Here is how to do that:

1) Insert a good disk in drive
2) Read Sector 0 of that disk:
   R 0,1 [ENTER]
3) Put the blown disk in drive
4) Write good Sector 0 to disk:
   W 0,1 [ENTER]

If you catalog the bad disk, you will see that the diskname and the used/free information is the same as the good disk, but do not let that alarm you. We did that to fool the Disk Controller into thinking the bad disk is at least partially restored to normalcy. Now we need to fix up the blown disk as much as we can. This is done by changing Sector 1. Here is how to fix Sector 1. First, get the most complete catalog and the most recent catalog of the bad disk in front of you. Then compare the two catalogs to see which filenames are missing. Next, compile an alphabetical list of all the filenames which are and should be in the catalog.

Then you need to find the corresponding sector for each filename. This is done by using the Find String function of the DISK FIXER

1) Put the bad disk in drive
2) Find a filename by:
   F 0,2D0,1 [ENTER]
   type in the filename [ENTER]
3) Ignore the "ERROR IN SECTOR" message
4) Write down the sector number for that
5) If that filename could not be found, make sure you typed it in correctly and try again; otherwise that file does not exist on the disk.
6) Repeat the process from step two for all of the filenames

You should now have an alphabetical list consisting of two columns: filenames and sectors. With that information in hand you are ready to begin fixing up the bad disk. This is done by modifying Sector 1 of the blown disk. First you have to read Sector 1 from the bad disk by doing this:

1) Put the bad disk in drive
2) Read Sector 1 of disk by:
       R 1,1 [ENTER]

Then you want to alter the contents of sector 1. This is done by using the alter function of the DISK FIXER. This process is best learned by observing a concrete example.

Lets say the blown disk has 14 files (filenames) on it. Thus there should be 14 entries on sector 1; one entry for each file. The rest of the sector should be all zeros. Let's alter Sector 1:

1) Keep the bad disk in drive
2) Enter the Alter function:
       A 0 [ENTER]
3) Type in the following just as shown, including the spaces:

   1 2 3 4 5 6 7 8 9 A B C D E

4) Do not press [ENTER] yet!
5) If you saw a non-zero entry after the E entry in the first column then type in [0] and a [SPACE] and repeat until the first column shows a zero.
6) Press [ENTER]
7) Write the revised Sector 1 to the bad disk:
     W 1,1 [ENTER]

You have just entered a table of pointers to the files on the disk. The table points to the corresponding sector for each file name. This is the table that is updated and sorted if you add/delete files to the disk.

Leave the DISK FIXER by typing [Q] for QUIT and press [ENTER]. Then catalog the disk. Lets call this new catalog the mix ed catalog. You will see the reason once the disk has been cataloged. Notice how the catalog is NOT in alphabetical order It does however contain all of the file names that you hoped and prayed would be on the disk! The next step is to alphabetize the catalog. This is done by first alphabetizing the catalog on paper and carrying along the appropriate sector number of each filename. Here is an example of a Mixed Catalog.

MIXED CATALOG SORTED CATALOG

FILENAME	SECTOR	FILENAME	SECTOR
CAT	1	APPLE	E
SCREEN	5	CAT	1
VOTE	2	DEMO	7
FIRE	6	FIRE	6
APPLE	E	HELLO	9
HELLO	9	JUSTIFY	D
SCROLL	C	LOAD	3
LOAD	3	LOGO	A
TIME	8	PLOT	B
DEMO	7	QUICK	4
QUICK	4	SCREEN	5
JUSTIFY	D	SCROLL	C
PLOT	B	TIME	8
LOGO	A	VOTE	2

The above example shows how you should alphabetize the filenames and the corresponding sector numbers on paper. If you are unsure when dealing with funny characters, the system alphabetizes by lower to higher ASCII values. These values can be found on your TI Basic reference card Once you have done this you are ready to enter this information into Sector 1. You do not have to enter the filenames, just the sector numbers.

Here is how to do that:

1) Put the blown disk in drive
2) Read Sector 1 by entering:
   R 1,1 [ENTER]
3) Enter the Alter function:
   A 0 [ENTER]
4) Type in the sector numbers in the order as shown for the above sorted example catalog. Seperate each number by a space:
   E 1 7 6 9 D 3 A B 4 5 C 8 2
5) Then press [ENTER]
6) Write revised sec tor to disk:
   W 1,1 [ENTER]
7) Put a Write-Protect tab on the disk!

You have now fixed up the disk. For verification quit the DISK FIXER program and catalog the disk. You should have no problems during the cataloging process. But you are not completely done yet! DO NOT add/delete any files or programs to this disk!

Get a fresh disk and inititalize it to the same configuration as the blown disk Then backup the blown disk to the fresh disk. Then catalog the fresh disk and you will see that the used/free sector information is now correct. Thus, the fresh disk is now your working disk and the blown disk is now a disk for your archives.

Keep the blown disk in a safe place just in case you remember a file that was not previously recovered from the blown disk Go through the above procedures to recover that new-but-old file.

If you have any questions on how to fix up blown disks, please leave private mail to MIKE BALLMANN (that's TWO N's)
Happy fixing!

END OF FILE **********

SOURCE: CHICAGO BBS - FORTH PAGE :

Glad to see that some of our members have been experimenting with our TI-FORTH. The following was left on our bulletin board by Jim McCullough. It demonstrates the use of constants and variables. Type the following lines onto an empty screen. It will then function by typing (scr#) LOAD. Different widths between the lines can be made by changing the first number in line 2.

```
0   ( DIAGONALS PRACTICE-VARY THE INTERVAL 15 APRIL 1984 )
1 : HELLO DECIMAL ; -64SUPPORT -FLOAT
2 5 CONSTANT K
3 255 S->F K S->F F/ INT F->S
CONSTANT S
4 191 S->F K S->F F/ INT F->S
CONSTANT FI
5 0 VARIABLE DTC1 255 VARIABLE DTC1
Hit CTRL K to stop - any other key ENTER to continue.
;
6 0 CONSTANT DTR1 191 CONSTANT DTR1
7 : UA DTC1 K + DTC1 ! ;
8 : SCRN DRAW GRAPHICS2 13 SCREEN ;
9 : DLINE UA BS DTC1 DTR1 DTC2 @ DTR2
LINE ;
10 : SPIDER S 0 DO DLINE LOOP ;
11 255 CONSTANT JC1 0 CONSTANT JC2 0
VARIABLE JR 191 VARIABLE JR2
12 : RTD JR1 K + JR1 ! ;
13 : DDIAG RTD LTU JC1 JR1 @ JC2
LINE ;
14 : FIL FI 0 DO DDIAG LOOP ;
15 SCRN SPIDER FIL KEY TEXT FORGET
HELLO
```

The pattern will stay on the screen until any key is pressed. Also as previously promised, here is Dave Wakely's LINES program which was demonstrated in our introduction to TI-FORTH at the March meeting.

For this one load -GRAPH, -SYNONYMS, -TEXT, -GRAPHICS2

```
0 ( DEMO FOR GROUP MEETING OF 3/3/84 1 RANDOMIZE
2 : TRND DECIMAL 255 RND 191 RND 255
RND 191 RND ;
3 : SCRN DRAW GRAPHICS2 8 SCREEN ;
4 : TLINE 30 0 DO TRND LINE LOOP 12
SCREEN ;
SCRN TLINE KEY TEXT
FORGET TRND
```

After the routine has handled sprite motion and auto-sound, and has checked for the system reset key, the timing functions are performed and the VDP status is stored. The interrupt handler then checks whether there is an external routine to be executed.

The existence of an external routine is determined by the word value at CPU RAM >83C4. If this location is zero, there is no external routine and the interrupt handling is complete. A non-zero value in this location is assumed to be a pointer to another interrupt routine. Control is passed to that location.

At this point in the processing, the workspace pointer indicates the GPL workspace (>83E0). If the routine is to use this workspace (a call to KSCAN, for example), the values in certain registers must be preserved as follows:

1. If the routine is operating in the GPL environment, the values in registers 13, 14, and 15 must be preserved.

2. If the routine is operating in the BASIC or EXTENDED BASIC environment (loading a program using the CALL LOAD), the values of registers 9, 10, 13, 14, and 15 must be preserved. Also, register 8 (>83F0) of the GPL workspace must be cleared regardless of whether or not that workspace is used.

An external routine may conclude by returning to the console routine with the GPL workspace active. At this time, register 8 is cleared and control is returned to the point at which the interrupt occurred. If preferred, an external routine may return directly to the point at which the interrupt occurred by loading the interrupt workspace pointer (>83C0) and performing a RTWP instruction. I highly suggest this method!

Here is an example of an interrupt-driven program for you to review and enter in using the EDITOR/ASSEMBLER. This should clear up any confusion from the above article. Until next time . . .

```
      DEF INT
      REF VMBW
*
MYWS    BSS >20
MYWS2   BSS >20
TIMER   DATA 0          TIMER
INTWS   EQU >83C0       INTERRUPT WORKSPACE
GPLWS   EQU >83E0       GPL WORKSPACE
*
MYNAME  TEXT 'JOHN  PHILLIPS'
BLANKS  TEXT '              '



*************************************************************************
* PROGRAM TO DEMONSTRATE EXTERNAL INTERRUPT CODE.                       *
*************************************************************************
INT     LIMI 0          DISABLE INTERRUPTS
        LI R0,INTERR    GET ADDRESS OF EXTERNAL INTERRUPT ROUTINE
        MOV R0,@>83C4   TELL INTERRUPT HANDLER THERE IS ANOTHER ROUTINE
        LIMI 2          ENABLE INTERRUPTS AGAIN
*
INT     LWPI MYWS       LOAD MY WORKSPACE FOR PROGRAM
LOOP    JMP $           SAME AS  "10 GOTO 10" IN BASIC
```

```
*************************************************************************
* EXTERNAL INTERRUPT ROUTINE.  IF YOU FOLLOWED THE ABOVE CODE, YOU WILL *
* REALIZE THAT IT DOES NOTHING EXCEPT AN INFINITE LOOP.  THIS INTERRUPT *
* CODE WILL PRINT MY NAME OUT AND ERASE IT EVERY 1/60th OF A SECOND.    *
*************************************************************************
INTERR  LWPI MYWS2      LOAD ANOTHER WORKSPACE SO MY MAIN ONE ISN'T DESTROYED
        MOV @TIMER,@TIMER IS TIMER ZERO?
        JEQ TIMEUP      NO, SO EXIT THIS ROUTINE
        DEC @TIMER      YES, SO PRINT MY NAME
        JMP INTRTN      AND EXIT THIS ROUTINE
TIMEUP  LI R0,60        RESET COUNTER
        MOV R0,@TIMER   AND SAVE IN RAM VARIABLE
        LI R0,12+9    CENTER OF SCREEN
        LI R1,MYNAME    RAM ADDRESS OF MY NAME
        LI R2,14        14 BYTES TO WRITE
        BLWP @VMBW      WRITE MY NAME
        LI R1,BLANKS    NOW POINT TO BLANKS
        BLWP @VMBW      ERASE MY NAME . . . DON'T BLINK!
INTRTN  LWPI INTWS      RESTORE INTERRUPT WOKSPACE
        RTWP            RETURN TO MY MAIN PROGRAM
*
        END INT         LOAD AND GO OPTION
```

ANYBODY NEEDING JOHN PHILLIPS' ASSISTANCE ON THEIR PROGRAMS OR ANSWERS TO YOUR QUESTIONS, PLEASE WRITE TO THE ADDRESS BELOW:

JOHN PHILLIPS @ A
99'ERS USERS GROUP ASSN.
3535 SO. H ST. #93
BAKERSFIELD, CA 93304

PROFILE - STEVE DAVIS

In the Fall of 1981, Steve Davis knew very little about computers and publishing. Out of curiosity, he enrolled in a community service class in BASIC programming. He was so fascinating by the third class that, he felt he had to have his own computer to practice programming. He rushed out to a department store advertising the TI Home Computer on sale for $400. It was too good of a bargain to pass up. Who ever thought the 99/4A would sell for $50 this last Christmas? At that time, Steve was a salesman representing a company that sold cable television equipment. Previously he had sold Audio-Visual equipment and had been a writer and producer of multi-media shows. College degrees in Radio-TV-Film had led him in this direction.

A search for books of BASIC programs using the 99/4A's features of graphics, music, and speech proved fruitless. Out of frustration, he began writing his own programs. This led to the realization that many other users must also be frustrated at the lack of books for the TI. Thus, in February of 1983, Steve quit his job to publish his first book, PROGRAMS FOR THE TI HOME COMPUTER. It was a tremendous gamble and a lot of work, but worth it. His degree in Journalism came in very handy.

Initially, he decided to distribute the book himself. The book was a tremendous success selling 45,000 copies in the first nine months of publication. Last Fall, arrangements were made with Prentice-Hall to distribute the book. Not bad for a first effort when it made the list of "most wanted computer books".

Dear Fellow 99/4 Users Groups,

This is my second appeal to you to join in supporting the:

"99'er User Group Association" (located in Bakersfield, CA)

This non-profit association of all 99/4 Users Groups in the USA (and othe countries) has the potential of providing all of us 99/4 users with the best "future" source of un-biased (and non-commercial) information concerning 99/4 computer hardware and software.

The monthly National 99/4 Newsletter that Don Vieth is producing contains the best 99/4 information from all of our newsletters and expert knowledge of the 99/4 systems and software by people that know the 99/4 inside-out. These people are all volunteers. Some of them helped TI build and program the 99/4 when they worked in Lubbock.

You can support this effort by sending $10 (or more) to:

99'er Users Group Association
3535 South H Street - #93
Bakersfield, CA 93304

for a one year's subscription for your Users Group (or for a personal subscription). Distribution is also being made via about 25 "Distribution Point" User Groups throughout the USA. (There is one of these in your geographical area.)

I was in Bakersfield in August and met Don and Luci Vieth. Don works for one of the Bakersfield oil companies and is doing this editing/publishing job for us on a part-time basis (nights and weekends). I can assure you that thi is not a commercial venture--these are dedicated people. Don has run into some financial problems with this effort that will be remedied by the support of your 99/4 Users Group.

Please post this letter on your local 99/4 bullentin boards in order to give all 99/4 owners in the USA an opportunity to support this effort.

Sincerely,

John Owen

John Owen, Corresponding Secretary
Johnson Space Center 99/4 Users Group (JUG)

*** BOOK REVIEW ***
*** TI REFERENCE REVIEW ***
*** BY MARK GRAUER ***
*** JACKSONVILLE, ARKANSAS USERS GROUP ***

It was a definite pleasure to discover two excellent books dealing with fundamental operations of the TI-99/4A. The first book is the TI-99/4A BASIC Reference Guide by Carol Ann Casciato and Donald J. Horsfall, published by SAMS and selling for $ 17.95. This reference book should have been included with the computer to begin with. This book goes through each function and command of the console in BASIC and explains each in English rather than "Computer-ese". With each entry, defined alphabetically, is a description of how and when to use the command or function, how to link to other commands or functions and short example programs to demonstrate. One of the nice features is the error message section, which tells you what you did wrong. This is very helpful when you can't figure out how you have messed up, and the TI Users' Reference Guide doesn't seem to help. Are you tired of having to keep flipping to a particular table or chart in the back of the book? That's not a problem with this book, every time you need a particular table (i.e. ASCII) it is right there with the entry you need. For the beginner, there is a helpful section on de-bugging programs, and a well stocked glossary of terms in the back of the book. The only bad thing about the book, is that the spiral binding is a little bit too small for the number of pages packed into the book. All in all, this TI-99/4A BASIC Reference Manual is a definite must for the smart programmer's work table.

The second fine book I have discovered is COMPUTE!'s GUIDE TO TI-99/4A SOUND AND GRAPHICS by Raymond J. Herold, published by Compute! for $ 12.95. This book deals with the basics of sound, graphics, sprites and speech synthesis. For the programmer beginning to work in Extended Basic, or the programmer who wants to improve their techniques, this is a valuable book. I was fascinated by the understandable explanation of how the TMS 9918A VDP operates and how everything is displayed on the screen when your program runs. For the more advanced souls there is a well written section on Advanced Sprite Handling Techniques including how to deal with POSITION, COINC and DISTANCE. The ways of moving a sprite and different ways of controlling it's movement are explained and example programs are given. These example programs are easy to convert to sub-routines in your own programs. Some of the sound programs are similar to those given in Regena's GUIDE TO THE TI-99/4A. But overall this book is well written and organized, and is well worth picking up for your bookcase. I certainly hope that both Compute! and SAMS will keep up their outstanding support of the TI-99/4A users.

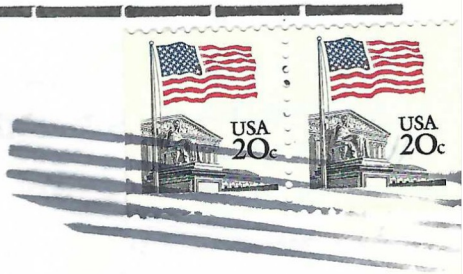
*** SPECIAL PRICE ON HARDWARE ***

CORCOMP MICRO EXPANSION

A few units of the "BASIC" CORCOMP MICRO EXPANSION SYSTEM (RS232 Version) have become available at a VERY LIMITED QUANTITY and SPECIAL PRICE of $ 95.00 (which includes UPS shipping). This unit does not, I repeat, DOES NOT INCLUDE any memory or disk expansion board. These are brand new and in the box with power supply, warranty card and documentation. Send $ 95.00 (Money order or check) to the address below and your unit will be shipped immediately. Checks will delay shipment approximately 10 to 14 days.

THE NATIONAL 99'ER
3535 SOUTH H STREET #93
BAKERSFIELD, CA 93304

AIR MAIL



BILL PECHNIK (133)
1467 CARMI DR.
PENTICTON, BC
CANADA V2A4R9