Covering the TI99/4A and the Myarc 9640

# MICROpendium

Volume 12 Number 6 July 1995 $3.50

# AMS

## Customizing your system

•••

## An initial review

THE HOME COMPUTER ERA CONTINUED

**The role of Apple Computer in the early years**

MICRO-REVIEWS

**TI Workshop, Halls of Lost Moria**

# CONTENTS

**MICROpendium**

**MICROpendium (ISSN 10432299) is published monthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Second-class postage paid at Round Rock, Texas.** POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.



All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680. We cannot take responsibility for unsolicited manuscripts but will give consideration to anything sent to the above address. Manuscripts will be returned only if a self-addressed stamped envelope is included.

Foreign subscriptions are $40.25 (Mexico); $42.50 (Canada); $40.00, surface mail to other countries; $52 airmail to other countries.

All editions of MICROpendium are mailed from the Round Rock (Texas) Post Office.

Mailing address: P.O. Box 1343, Round Rock, TX 78680.

Telephone & FAX: (512) 255-1512

Delphi TI NET: MICROpendium

GEnie: J.Koloen

Internet E-mail: jkoloen@io.com

**John Koloen.....Publisher**

**Laura Burns.....Editor**



### *READ THIS

Here are some tips to help you when entering programs from MICROpendium:

1. Most BASIC and Extended BASIC programs are run through Checksum, which places the numbers that follow exclamation points at the end of each program line. Do not enter these numbers or exclamation points. Checksum is available on disk from MICROpendium for $4.

2. Long Extended BASIC lines are entered by inputting until the screen stops accepting characters, pressing Enter, pressing FCTN REDO, cursoring to the end of the line and continuing input.

# TEX COMP LTD.

TEX - COMP NOW HAS THE REMAINING DATABIOTICS CARTRIDGES IN LIMITED QUANTITIES. SEE LIST BELOW FOR SPECIFIC CARTRIDGES AND PRICES. DON'T FORGET THAT WE STILL CARRY ONE OF THE LARGEST SHAREWARE LIBRARIES AVAILABLE FOR THE TEXAS INSTRUMENTS HOME COMPUTER.

TI WORKSHOP WAS INTRODUCED BY DATABIOTICS IN 1988 AND FAIRLY UNKNOWN TO THE TI COMMUNITY AT LARGE. IT INCORPORATES DISK MANAGEMENT, EDITOR ASSEMBLER, DISK DISASSEMBLY, DEBUGGING, AND MUCH MORE ALL IN ONE HANDY CARTRIDGE .

# TI WORKSHOP FEATURES ONLY $49.95

• **MEMORY MANAGER**
*SHOW, DUMP, EDIT, INIT, COPY, & SEARCH MEMORY*

• **DISK MANAGER**
*FILE & DISK UTILITIES, COPY, PROTECT, UNPROTECT, & DELETE ALL FILES, SECTOR EDITOR, WITH EDIT, READ, WRITE, & FIND*

• **PROGRAM LOADER**
*LOAD & RUN, LOAD, READ, SAVE, & RUN PROGRAM IMAGE FILES, LOAD & RUN OR LOAD D/F 80 FILES TO*

• **DEBUGGER**
*SET BREAK POINTS, SINGLE STEP, SLOW EXECUTE, EXECUTE PROGRAMS, SET AND VEIW REGISTERS, OR DISASSEMBLE*

• **EDITOR/ASSEMBLER**
*LOAD TEXT FILES, EDIT TEXT, SAVE D/V 80 OR D/F 80 FILES, PRINT TEXT FILES, PURGE MEMORY, ASSEMBLE, AND CROSS REFERENCE*

• **USER GUIDE**
*HANDY 58 PAGE OWNERS MANUAL THAT DETAILS ALL OF THE ABOVE FEATURES*

• **SYSTEM REQUIRMENTS**
*CONSOLE, 32K EXPANSION MEMORY, & DISK SYSTEM, (PRINTER OPTIONAL)*

**REMEMBER ANY ONE OF OUR OVER 300 SHAREWARE DISKS IS JUST A PHONE CALL AWAY !!**

DBT DataBioTics

| | |
|---|---|
| 4/A FLYER | $24.95 |
| BARRAGE | $29.95 |
| BEYOND PARSEC | $24.95 |
| BEYOND WORD WRITER | $39.95 |
| BLACK HOLE | $24.95 |
| BOXER | $24.95 |
| BREAKTHGH/ BREAKOUT | $24.95 |
| BURGER BUILDER | $24.95 |
| COMPUTER WAR | $29.95 |
| CONCOLE CALC | $24.95 |
| D-STATION I | $24.95 |
| D-STATION II | $24.95 |
| DRAGON FLYER | $29.95 |
| DESKTOP PUBLISHER | $39.95 |
| ESCAPE | $24.95 |
| FACE CHASE | $24.95 |
| GREAT WORD RACE | $29.95 |
| JUMPY | $29.95 |
| JUNK MAN JR | $29.95 |
| MAGIC MEMORY | $49.95 |
| MICRO PINBALL | $29.95 |
| MICRO TENNIS | $29.95 |
| MIDNIGHT MASON | $24.95 |
| MINI WRITER | $24.95 |
| MUNCHMAN II | $24.95 |
| PARAPRINT | $69.95 |
| PROTYPER | $24.95 |
| QMAZE | $29.95 |
| RED BARON | $39.95 |
| SORGON II | $29.95 |
| SPACE PATROL | $24.95 |
| SPOT SHOT | $29.95 |
| SPY'S DEMISE | $24.95 |
| STARGAZER | $39.95 |
| STAR RUNNER | $29.95 |
| STAR TRAP | $24.95 |
| STRIKE 3 | $29.95 |
| TI WORKSHOP | $49.95 |
| WORD WRITER | $29.95 |
| WORD WRITER EXTRA | $29.95 |

ORDER BY PHONE

VISA & MASTER CARD EXCEPTED

TEX COMP LTD.
AMERICAS NUMBER ONE TI COMPUTER RETAILER
**425 E. ARROW HWY. #732**
**GLENDORA, CA 91740**

**(818) 339-8924 FAX (818) 858-2785**

ALL PRICES ARE FOR CASH/CHECK, ADD 3.5% FOR CREDIT CARD ORDERS

# COMMENTS

# Crichton buys Competition

Good news for TI users. Competition Computer will remain in business under new ownership. Kyle Crichton, who is marketing a CD-ROM of programs and files for the TI, bought out Competition. He will continue to operate the company under the Competition name. The company offers a free buyers guide to TIers who call its toll-free line, 800-471-1600.

**EUROPEAN DEVELOPMENTS**

Several projects are under development in Europe, including a Teletext card for the TI and Geneve, an assembly language version of Risk and a program that gives speech to Infocom adventures. I don't have much more information at this point, but you can email for more at the following address: OLIVER@THORIN.SWB.DE.

**24 PAGES THIS MONTH**

We've cut back to 24 pages this month. But don't be alarmed. We just wanted to reduce the workload for a month. We'll be back to 32 next month.

**SCSI DSR UPDATE**

David Nieters reports progress on the SCSI DSR for the TI. "I am able to save and load files to the SCSI. I can overwrite files. I can even do relative access reads.What I have left is to add the RESTORE I/O op code and do relative access writes." The DSR will let 99/4A users utilize the WHT SCSI card.

**—JK**

# FEEDBACK

## Review gets thanks, with corrections

I am writing to thank MICROpendium and specifically Dr. Charles Good, for taking the time and the effort to provide the objective review of my Mailing List Manager program which appeared in the April 1995 issue. As a result of the review I have received and shipped 14 orders for the program so far. (I'm writing this in June.) That's a pretty respectable response when you figure that I've sold only 154 copies of all the other shareware programs I've written and distributed over the last 10 years. Thanks!

I am also writing to let readers know that there are a couple of typos in the review that affect people placing orders for a copy of MLM,l and I would like to let folks know that a couple of things have changed since the review was first put together, which was January 1995, for the April 1995 issue of MICROpendium.

1. My address is 2310 Cypress Ct., Grand Junction, CO 81506, not 10 Cypress Court as appears in the review.

2. MLM has a 28-column text editor, not a 40-column one as is stated in the review. the original version did have a 40-column, assembly language-based editor, but I was forced to abandon it when I ran out of disk space on a SS/SD disk. So the 28-column, Extended BASIC editor is now the editor of choice because it takes up fewer sectors on disk.

There are *no printed docs* available. My original intent was to provide a printed manual, but the cost to print instruction manuals is much higher than I would ever be able to recover in sales of the program, so no printed manual is offered.

**Bill Gaskill**
**Grand Junction, Colorado**

## Deleting SCSI files

In the June 1995 issue you published two reviews of SCSI (small computer systems interface) hard drive use on a Geneve, one written by me. The two reviews complement each other nicely because they mostly cover different aspects of SCSI use. There is one area in which the two reviews directly contradict each other. My review specifically states that there is no problems creating or deleting files or directories. The other states that files and directories cannot be deleted. If this is true, then SCSI hard drives for the Geneve would be a very incomplete and imperfect product. But it isn't true.

I have been using MDOS "v2.50S beta" and "v2.50S delta" on my Geneve for several months and have no trouble deleting files and directories both directly from the MDOS command line as well as from the directory manager. In a recent conversation with Bud Mills I was told that there is an older version of SCSI MDOS that would not allow files to be deleted. It is likely problems with the deletion of files from SCSI hard drives are due to either an outdated or currupted MDOS.

**Charles Good**
**Venedocia, Ohio**

## Struggles with HD

Since I wrote you last year my standalone system has been made redundant and relegated to the attic. I purchased a second-hand peripheral expansion box complete with (two) half-height double-sided drives and on June 3 of this year I went along to our AGM/Show in Derby and purchased a Myarc hard and floppy disk controller and hard disk. I would have liked the SCSI (small computer systems interface) card but at the moment for me a bit too expensive.

I'm having a struggle at the moment coming to terms with the complexity of a hard disk; one of the quirks is I reformatted the HD and now the MDM5 will not load. It loaded okay first time but not now. I'm hoping one of the user group chaps can sort it all out.

**D.H. Caine**
**Crewe, England**


IMAGINE ALL THE PUBLIC DOMAIN SOFTWARE WRITTEN FOR THE TI99/4A AVAILABLE ON ONE CDROM . THAT IS WHAT YOU WILL HAVE AT THE END OF YOUR SUBSCRIPTION TO THE TI 99/4A CDROM. FOR THE PRICE OF 30 -50 PUBLIC DOMAIN DISKS YOU WILL RECEIVE 1900 DISKS INITIALY AND MANY MORE IN PERIODIC UPDATES UNTIL YOU WILL HAVE 650 MEGABYTES OF TI SOFTWARE. THATS A LOT OF SOFTWARE FOR YOUR COST OF ONLY $100 . YOUR FIRST CDROM WILL BE SHIPPED TO YOU FOUR WEEKS AFTER YOU HAVE SENT YOUR ORDER FORM WITH A CHECK OR MONEY ORDER PAYABLE TO KYLE CRICHTON , 350 MARCELLA WAY, MILLBRAE, CA 94030. USER GROUPS SHOULD CONTACT ME ABOUT THE USER GROUP PARTICIPATION PROGRAM. NOTICE TO SOFTWARE AUTHORS I WILL NOT KNOWINGLY PUBLISH PROGRAMS NOT IN THE PUBLIC DOMAIN PLEASE CONTACT ME IF YOU HAVE PROGRAMS WHICH ARE BEING DISTRIBUTED ILLEGALLY AS PUBLIC DOMAIN BUT WHICH YOU DID NOT RELEASE AS SUCH. MESSAGES:(415) 697-1108 ,FAX:(415)697-7406

---

YOUR NAME:_____
ADDRESS:_____
(STREET) (CITY) (STATE) (ZIP)
CDROM PLAYER AND PLATFORM INFORMATION:
MY CDROM PLAYER IS MADE BY:_____MODEL#:_____
I HAVE IT CONNECTED TO: IBM/COMPATIBLE___MAC___HORIZON SCSI____
OTHER:_____(BE SPECIFIC)
IF YOU CHECKED HORIZON SCSI: IT IS CONNECTED TO:TI99/4A___9640___
I USE THE FOLLOWING OPERATING SYSTEM WITH MY CDROM PLAYER:
WINDOWS___DOS___VERSION:_____(3.1 FOR WINDOWS FOR EXAMPLE)
SYSTEM 7.X___MYARC DOS(MDOS)___VERSION_____ TI99/4A____
____I HAVE BEEN ABLE TO VIEW KODAK PHOTO CDS THAT WERE RECORDED IN 2 OR MORE SEPARATE SESSIONS
____I HAVE BEEN ABLE TO READ CD ROM XA DISKS
____MY CD ROM MANUFACTURER ASSURES ME THAT I CAN READ ISO MULTISESSION DISKS
_____DON'T KNOW(PLEASE INCLUDE YOUR CDROM PLAYERS SPEC SHEET )

## 1980s home computer era — Part 2

# Players in the game: Apple Computer

**By BILL GASKILL**
**©1995 B. Gaskill**

Although I never considered Apple to be a player in the home computer market, they probably were, at least on the high end of the scale. They are of course still in business today, but the Macintosh line, which is all that is left of the desktop computers that Apple once produced, and the new Power Mac with the Apple/IBM/Motorola Power PC Reduced Instruction Set Chip (RISC) are struggling to find a larger market share in both corporate and residential America. According to an article in the Nov. 7, 1994 issue of *Time Magazine*, Apple reported profits of $114.7 million during the third quarter of 1994, more than 40 times the level it saw just two years ago, but it still has the same market share today, that the company did in 1984, which is a paltry 8-10 percent.

The Apple IIe that grew out of the Steve Wozniak, Alan Baum computer that was created in a Silicon Valley garage in 1977 is long gone, and the Apple IIc, introduced in June 1984 as the closest thing to a home computer Apple ever produced, have disappeared under the weight of the PC Clone explosion of the mid to late 1980s. The Apple IIGS that was officially released in January 1987, and designed to compete with the Atari ST and Commodore Amiga, has also faded away without fanfare or circumstance.

Similarly, the players in the game at Apple have changed. Cofounders Steve Jobs and Steve Wozniak are no longer with Apple. Woz pretty much left the picture when Apple went public and his stock made him $150 million richer. Jobs was of course ousted in a power play after John Scully was brought in from Pepsi to run things. Steve Jobs then formed the NeXt Computer Company using some of his own money. Unfortunately for Jobs, NeXt never found a market.

In 1994 NeXt announced that it would cease manufacturing the NeXt computer and would instead concentrate on marketing the company's multi-taking operating system. A few ads for the operating system appeared in a few magazines for a few months after Jobs' announcement, but it's not been heard of since. Now of course, even John Scully has been replaced at Apple.

In 1985 Apple Computer was riding high, as witnessed by this newsbyte in the February 25, 1985 issue of *Infoworld*.

"Apple Computer announced $698.3 million in net sales for the first quarter of 1985, a 121% increase over the $316.2 million in sales for the first quarter of fiscal 1984. John Scully, president and chief executive officer, says that Apple sold approximately 500,000 computers during the last quarter of 1984."

To top that off, Appleworks, the easy-to-use, integrated database, word processor, and spreadsheet software program written by Rupert Lissner, surpassed Lotus 1-2-3 in sales. This marked the first time that any application had outsold Lotus since 1983.

As we will see later in this series of articles, by 1990 Apple would have all but abandoned the Apple II line that made up so much of its 1980s profit and the company would focus its efforts on the Macintosh by expanding the line and dropping the price. That decision is probably why Apple is still a financially healthy company in 1995, where Commodore and many other players of the 1980s home computer market are gone.

In 1993 Apple sold a whopping 2.5 million Macintosh computers, but there were over 12.5 million PCs sold that year. Although I don't own an Apple Computer, I really want them to stay in business because there are no other choices anymore and I don't like not having choices. I was saddened when American Motors went under in 1987 for the same reason. I have the same feelings in the software arena where once mighty Borland International is struggling to keep itself afloat in a sea of Microsoft sales and Microsoft bucks.

Apple Computer has now become part of the Power PC group, which is a consortium made up of Apple, IBM and Motorola. Their goal is to pool their resources in order to stave off the monopoly that seems to be developing between Intel and Mi

**(See Page 7)**

## BUGS AND BYTES

### Black + silver = green?

According to columnist L.M. Boyd, an estimated 10 million old personal computers are thrown out yearly and the number is rising. Experts predict 150 million such old computers to be added to the United States' landfills in the next 10 years. Hobbyists with TI99/4As and other "classic" computers, however, are doing their part to help save the earth!

### X-Files, move over

The Portland Users of Ninety-Nines (PUNN) have taken the first step towards a television series on local cable access. Club members are scheduled to meet with Columbia Cable officials Aug. 11, and according to Ted Peterson, club president, tapes could air as early as the first week of October. Club officers are trying to plan meetings around demos that will show off the TI to good advantage on the air.

# HOME COMPUTER ERA —

(Continued from Page 6)

crosoft where Intel is controlling the hardware side of the PC industry and Microsoft is controlling the software side. I suppose only time will tell how successful the Power PC effort will be?

One final note about Apple Computer. Byte Magazine did a really neat series on Apple from Steve Wozniak's point of view in the December 1984 and January 1985 issues. If you have access to these two older magazines, the articles are worth reading.

Following is a list of computers Apple has produced over the years.

• APPLE I-The original "built-in-the-garage" computer. About 200 were built and 175 of the 200 were sold. It was a text only computer that was built around the 6502 chip only because other processors of the day cost hundreds and the 6502 could be purchased for $25.

• APPLE II — Released in June 1977 at a cost of $1298 for the 4K RAM version or $2638 for the maximum 48K RAM ver-[illegible]. It supported cassettes, game paddles, [illegible] color graphics. Later releases, such as the II+ and IIe would add enhancements such as a lower case letter set, an improved keyboard, fewer chips required to make the computer operate, interfaces and system support for disk drives, more memory, 80-column displays, hard disks and the like.

• APPLE IIc — Released in May 1984, the IIc was what Apple referred to as a "focused product", meaning it was created specifically to compete against IBM's PCjr. The basic system without optional monochrome monitor was around $1295. The computer had 128K RAM, a SS/SD 5.25 inch floppy drive built in and it came with Applesoft BASIC. It could be hooked to a TV via an RF modulator and a second disk drive was available for it. When first announced there was also to be a flat panel LCD display offered as an option, but I don't think it ever became a reality.

My wife Jacque and I had the oddest-ever experience with with the IIc. We [illegible]ked the computer very much when it first [illegible]he out and found a dealer who gave away a free 10-speed bicycle with the purchase of an Apple IIc. We bit on the sales promo and took our bike and computer

The introduction of the Lisa and Macintosh was supposed to spell the end of the Apple II line, but the II and its fans refused to die. Ultimately, Apple got the message and decided to adopt the company slogan "Apple II Forever."

home only to find that the IIc would not boot up. I took it back to the dealer and it worked perfectly for him. Nonetheless, he gave me another IIc and I happily took it home too, only to find that it too would not boot up when plugged in to the electrical outlets in my four year old house. I took it back and it worked properly at the dealers shop, but we still couldn't get it to boot at my house. Each time I would plug it in and insert the Apple DOS disk I got some kind of error message that I don't recall, but at the dealer's shop it worked perfectly? We ended up giving the bicycle and the computer back, and that was the last time I have ever messed with an Apple computer.

The Apple IIc also seems to be among the first of the Apple II family to fall victim to mass liquidation. The introduction of the Lisa and Macintosh was supposed to spell the end of the Apple II line, but the II and its fans refused to die. Ultimately, Apple got the message and decided to adopt the company slogan "Apple II Forever" in support of the Apple II, II+, IIe, IIc and the existing customer base for these machines. But by March 1987 COMB liquidators (now known to us as Damark Industries) were carrying the Apple IIc and an Image writer printer on the back cover of their catalog for under $1000. This was a radical departure from Apple's up-to-then very strict policy against mailorder sales of its computers.

• APPLE IIGS — Released in July 1986, it came with 256K RAM expandable to 8 megabytes, 2 super hi-res graphics modes, external disk drives, a 16-bit 6502 CPU, a 32 voice sound chip. The GS in the computer's name stood for Graphics and Sound. Price was $995 without a monitor or a disk drive. Sheldon Leemon tells an interesting story about the II GS and Leemon's conversation with Good Morning America's David Hartman in the March '87 issue of Compute!. It's on page 83 on the magazine if you're interested.

• APPLE III — Released in May 1980 as a replacement for the Apple II. It never did, because the computer buying public spurned it in the marketplace. It appeared a year after originally promised and at first had a very high failure rate which all but ruined its chances of becoming the next generation Apple that Apple Computer wanted to be. For some reason there aren't many photos of the Apple III around. I did find an artist's rendition of one in an Apple Computer ad in Compute! magazine, January 1983 issue, on page 13 (if you're curious).

• LISA — Released in January 1983 as the business solution, it sold for $9995 and was an abject failure in the marketplace. Despite its new and innovative approach to computing (it had a mouse and it came with bundled software), it was slow and over priced even with the optional 5mb hard disk. By 1987 mass liquidation houses were hawking the Lisa for $799 in an effort to get users to buy them after Apple unloaded the computer at bargain basement prices to these outfits. A pretty usable Lisa for business purposes could be purchased with 20mb of hard disk space and 1mb of RAM for $1995.00. Considering the Lisa came bundled with basic business software, that was a fairly good price.

• MACINTOSH — Announced in January 1984 at $2495, the Mac came with 128K RAM, a 9 inch, hi-res, built-in black and white monitor, and a single 3.5 inch floppy drive. A Mac Plus was introduced in November 1986 that provided some nice upgrades such as the ability to boot from a hard disk, and 800K floppy disk drives instead of the original 400K ones that users complained about.

(See Page 8)

# HOME COMPUTER ERA —

(Continued from Page 7)

One of the less remembered issues marring the Mac's reputation was Apple's decision to force Digital Research to alter the look and feel of its GEM Desktop because Apple felt the GEM product encroached upon the look and feel of the Macintosh. Pretty gutsy for a company that had themselves taken the entire Mac idea from the Xerox Star computer that Steve Jobs saw in Xerox's Palo Alto Research Center (PARC).

• MACINTOSH SE — The SE stood for System Expansion. It was basically, a Mac Plus with an internal expansion slot and open architecture that allowed third-party peripheral attachment. Introduced during the 1st quarter of 1987.

• MACINTOSH CLASSIC — Part of Apple's new pricing structure introduced in the Fall of 1990 (announced at Benelux in The Netherlands in September, and here in the U.S. in October 1990). The Classic was housed in a Mac SE case and chassis, which means it had the built-in monochrome monitor, 3.5" floppy disk drive, and it came with 1mb of RAM. With 2mb of RAM and the optional 40mb hard drive the unit sold for right at the $1500 mark. A similarly configured Mac SE would have sold for $3595.

• MACINTOSH LC — The LC, which Apple said stood for "Low Cost", had the look of the Apple IIGS rather than the classic Macintosh design. It was housed in a long, thin chassis that weighed less than 9 pounds fully loaded. It was a color Macintosh that used the Motorola 16mhz 68020 chip.

• MACINTOSH II — The first Mac to support color and the first Macintosh to be housed in a chassis that did not look like a Macintosh. The Mac II ultimately would be sold in various models (IIci, IIcx, IIfx, IIsi etc.) and at various pricing levels as Apple realized the need to compete against the invasion of Windows 3.0 based PCs that were depriving the Mac of its ease of use and what you see is what you get advantage. When it was released in the 1st quarter of 1987, it was supposed to compete against the 386 class of PCs.

• MACINTOSH PERFORMA — A dizzying number of Macintosh models exist under the "Performa" banner to[illegible] all of which seem outwardly to be the same computer. The number after the "Performa" name and the varying prices tell you there is something different about each one, you just have to figure it out. All are extremely attractive computers and look as if they would fit anywhere in a person's home. The models with full multimedia capabilities make a tempting purchase.

• POWERBOOK — A portable or laptop Macintosh computer.

• POWER MAC — These are what Apple sees as the future of Apple, if Apple is to remain in business. When "Plug and Play" becomes a reality in the PC world this year, the last real advantage of the Macintosh (easy setup and installation of components) will have been overcome by the big guns in the PC arena. This means there will be no reason left to buy an Apple product over any other desk top or personal computer on the market. But the Power PC chip is supposedly going to make the Mac able to compete with the Pentium chip from Intel.

## More Extended BASIC Myths

# Stacking it up

**By BRUCE HARRISON**

It may surprise some of our readers to hear that we really read those user group newsletters. We get them from Lima, West Penn, and occasionally other groups. Sometimes we find from this reading that many people are confused about certain aspects of Extended BASIC.

For example, in the October 1994 issue of the SW99ers newsletter, we found the question "What is the difference between capacity of Stack Bytes space and Program Space bytes?" An interesting question, for which there was no answer given. Well, here's the answer, folks. Stack Space is not even in the main 32K memory. It's space in VDP RAM, which is used by XB for various purposes. From a "cold" start, SIZE will yield 11,840 bytes of Stack free, and 24488 bytes of Program space. That 24488 bytes is in the high section of the 32K memory. The Stack space reported is located in the VDP RAM, and runs from >0997 through >37D7. This is not usable as program storage when the 32K memory is in use. (Note that under certain circumstances you can fool XB into believing that the 32K is not there, but that's another story.)

To see how that stack space gets used, try this little experiment in XB Command mode without any program present: Type SIZE <Enter>. The screen says 11840 bytes of stack free, 24488 bytes of program.

Now type A=10 <Enter>. Then type SIZE <Enter>. The screen says 11831 bytes of stack free, 24480 bytes of program.

What does this mean? When you assigned a value to a numeric variable, XB took up eight bytes from Program space to store the number (10) in floating point format. It also used up some stack space in VDP RAM to store the lookup table for this variable, so that XB can find the value of A when it needs to. Thus if you type in PRINT A <Enter>, XB will look up A in the lookup table in VDP RAM, get the address of the 8 bytes in Program space, and then will fetch and print that number. You can't see this happening, so you'll have to take our word for it. Now try this:

Type A$="" <Enter>. Type SIZE. The screen says 11821 bytes of stack free, 24480 bytes of program.

Thus 10 more bytes of Stack are used, but Program space is unaffected. The 10 bytes of stack used are the lookup table space for the A$ variable.

Now try this: Type A$="HELLO" <Enter>. Type SIZE <Enter>. The screen says

(See Page 9)

# MORE EXTENDED BASIC MYTHS —

(Continued from Page 8)

11812 bytes of stack free.

Program space stays the same, but giving A$ the value "HELLO" has eaten up another 9 bytes of Stack. Note that this is four bytes more than the length of "HELLO." From here on, the amount of Stack free (as shown by SIZE) will decrease by one byte for each character added to A$. For example, making A$="HELLOO" will yield 11811 bytes of Stack.

Okay, now before we leave this, try the following experiment: Type CALL CLEAR <Enter>. Then type SIZE <Enter>. The screen says 11798 bytes of stack free.

What? Yes, we lost some more stack space. Another 14 bytes gone because we told XB to CALL CLEAR. Why is that? It happens because XB also uses Stack space to store a lookup table for all CALLs used. Try for example a CALL INIT, and another 12 bytes of Stack gets used up.

**STACK DYNAMICS**

When XB is running a program, the ...ck space gets initialized during the pre...n so that all variables and CALLs used in the program have space reserved in the lookup tables. The string variables are all set to null strings. Numeric variables have their lookup table space reserved, and also have Program space set aside for their values. Each numeric variable used gets an eight-byte block of Program space assigned for its value in floating point notation. Thus an ordinary numeric variable needs eight bytes of Program space and eight bytes plus the length of the variable name of Stack space.

As the program runs, each new assignment of a non-null value to a string variable uses a fresh area in VDP RAM for the new value of the string. The storage space for its "old" value is marked as what we'll call a "dis-associated" string. From time to time, XB does a "garbage collection" process on the Stack space, so that all the dis-associated strings get removed and the space used gets "closed up." This is why XB programs will seem to pause execution sometimes, while the Stack is being ...eaned up. If a SIZE operation is performed when the program is stopped, XB performs a garbage collection before taking the SIZE, so that will never show the

String functions, like STR$ and SEG$ eat up stack space for temporary string storage. Stack is also used for parameter-passing operations in CALL LINK.

effect of the leftover string values that are in the Stack space. Some leftover strings will still be present in VDP RAM, but won't count as "used Stack space."

**THE STRUCTURE OF A STRING**

When variables have been assigned, the address of the start of the variables' lookup table will be at >833E in memory as a word. You can see what that address is from XB by: CALL PEEK(-31938,A,B):: PRINT A*256+B

This address won't do you much good, since there's no PEEKV available in ordinary Extended BASIC. On our own system, we have a P-GRAM card, so we can perform a CALL PG from XB and then examine the stuff that's in VDP RAM. Let's say we've just performed A$="HELLO" in command mode. After CALL PG, we get into the Memory Edit mode from the P-GRAM menu. The word at >833E tells us that the variable lookup table starts at >37CE in VDP RAM. There we find the ten-byte entry for the variable A$. The ten bytes (in Hex) look like this: 80 02 00 00 37 D6 37 C8 41 24.

Their meaning is as follows: 80 means this item is a simple string variable; 02 is the length of the variable's name; 00 and 00 means this is the last item in the list, otherwise these bytes would point to the start of the next variable's listing. 37 and D6 is the pointer to the start of the variable's name. 37 and C8 points to the actual content of this variable. (The H in HELLO) If A$ is a null string (A$="") then these two bytes will be 00 and 00, and no content will be stored. The 41 and 24 are the variable name A$ in hex.

The actual string in VDP doesn't have just the word HELLO, but has four additional bytes starting at >37C5. It looks like this: 37 D4 05 48 45 4C 4C 4F 05.

The first two of these bytes point back at the two bytes that give this string's location in the lookup table. The 05 is the length of the word "Hello". The 48 45 4C 4C 4F are the letters in "HELLO" in hex, and finally there's another 05, which repeats the length of "HELLO". If a new value had been assigned to A$, then the two bytes 37 and D4 would both be 00, marking this as a dis-associated string to be eliminated on the next garbage collection. We're sure there's a reason for having two length bytes, one before and one after the string's content, but we don't know what that reason is.

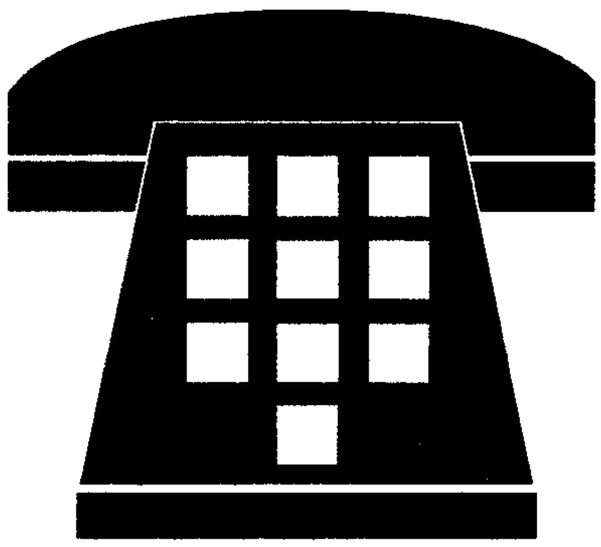
In summary, a simple (non-array) string variable takes up eight bytes plus the length of its name for the lookup table entry, plus four bytes plus its length if it's a non-null string.

The lookup table entries for numeric variables are similar, except that the first byte is 00 instead of 80, and the pointer to the content points to an address in the 32K memory, not in VDP RAM. Thus in terms of Stack use, a numeric variable uses eight bytes plus the length of its name. Of course the numeric variable's value uses eight bytes, but that's taken from Program memory (32K), not Stack.

**OTHER USES FOR STACK**

Extended BASIC makes other uses of Stack space during program operation. For example, math operations that involve parentheses, such as X=(Y+3)*D, use a "value stack" in the VDP RAM to store temporary values as floating point numbers. String functions, like STR$ and SEG$ eat up stack space for temporary string storage. Stack is also used for parameter-passing operations in CALL LINK. File operations use Stack space too, for Peripheral Access Blocks and buffers to accept file inputs and outputs. If your program runs out of Stack while executing, it will stop with an "OUT OF MEMORY"

(See Page 10)



Want to talk to someone at MICROpendium? You'll need to called between the hours of 9 a.m. and noon Saturdays. If you call at other times, you will probably get an answering machine. But don't let that bother you. We listen to the answering machine at least once a day and return calls as soon as possible, usually that day.

**Call us at 512-255-1512**

# MORE EXTENDED BASIC MYTHS —

**(Continued from Page 9)**

error. Taking SIZE just after that happens will usually just confuse you more, because neither Stack nor Program space will show up as zero. That's because Extended BASIC had something to assign to stack, did a garbage collection, and still didn't have enough room for that item, so it gave up and stopped with the error report without actually assigning the item to memory.

If your XB program is running out of stack space when it runs, there is something you can do to add more stack space. You can perform a CALL FILES(1) or CALL FILES(2) before running the program. The "normal" number of files in XB is 3. This ties up the VDP RAM space from >37D8 almost to the end of VDP RAM. If you do a CALL FILES(2), you'll free up another 518 bytes of stack for yourprogram's use. Doing a CALL FILES(1) will free up 1036 bytes for stack. Of course this will limit the number of files your program can have open at any given time, so the CALL FILES solution may not be practical.

While we're on this topic, let's point out that we have made an Assembly routine that will allow the program to perform a CALL FILES from within itself. That routine, complete with instructions, is available from the Lima Users' Group for anyone who needs it. The disk is called CALLFILES, and it is number 957B in the Lima Public Domain library. You can get it by sending $1 to Dr. Charles Good, P.O. Box 647, Venedocia, OH 45894. The Assembly routine can be used in several ways, and need not tie up any of your XB Program space.

### A MEMORY MYTH

In that same issue of the SW99ers newsletter, there was an item about a CALL LOAD that would reportedly "free up" an additional 8K of program memory! Now that would really come in handy, wouldn't it? The way to "free up" memory was given like this: CALL INIT :: CALL LOAD(-31866,30,0). After doing this, SIZE will show 32,193 bytes of Program space as being available.

Oddly enough, that does work, so that it appears that 7705 extra bytes are available in Program space. Unfortunately, this extra memory is simply an illusion. What's really happened here is that you've "artificially" changed a number in the CPU RAM Pad area at >8386, and have thus fooled SIZE into giving a false report. You can prove that by just bringing in a program with OLD, then taking SIZE again. The mythical 7705 bytes will disappear, and the SIZE will report the same Program space as if you hadn't done that CALL LOAD. There is no such thing as a free lunch, and there's no way to magically expand the 32K memory.

This is the third in a series of one article! Ordinarily we don't dabble into Extended BASIC, but there were some things we've not seen covered anywhere else in XB articles that might provide useful information to readers of MICROpendium. This may be the last in that "series of one," unless reader questions indicate that more are needed. We hope you've all learned something useful. See you in the Art of Assembly.

# VAST BBS back on line.

The VAlley of the Sun TI Users group is back on line with a new telephone number, according to Ralph E. Rees, VAST president.

New number for the board is (602) 267-1419. Additionally, Rees says, the BBS now has online 420 megs of space on the hard drive with a tape backup system that backs the data up nightly.

Rees invites all MICROpendium users to call and visit the board and says he would also like all TI users group officers to inform their members and ask them to call.

"We already have approximately 20 megs of TI megs of TI files on our system," he says, "but we have room for a whole bunch more."

THE ART OF ASSEMBLY — PART 49

# Gripes and delays

By BRUCE HARRISON

It's November 1994 as we write, and things don't look so good for the future of the TI99/4A. One by one, vendors are dropping by the wayside. Asgard is gone, along with many software authors. Bud Mills Services still exists, but there have been no ads appearing for some months now. Western Digital is still "talking about" the SCSI card and how soon its software will be ready. We're not holding our breath! One must ask, "When it's ready, will there be any customers left?" Part of the problem is the penny-pinching TI owner himself. In recent months, Dr. Charles Good has offered many of our public domain disks through the Lima Users' Group at $1 per disk, which includes the disk itself, the software with unlimited license, and the mailing. Still, people have called me directly to ask if I can offer these disks for less! Less than $1? In all probability, it has cost more than $1 to call me long distance just to ask the question. (NO, I DON'T!) At the 1994 Lima Faire, we were not selling anything, but merely allowing people to make copies at our table if they brought along blank disks. Business was brisk all day. Obviously $0.00 was the correct price for our goods.

## COULD BE WORSE...

I like using lines from old movies, and I'm reminded here of one from the movie "Young Frankenstein." In this case, the young doctor and his assistant Igor are in the process of robbing a grave for the body they'll use in the experiment. They're standing neck-deep in the dug-up grave. The young Doctor Frankenstein (Gene Wilder) remarks what a horrible situation they're in, and Igor (the late Marty Feldman) quips, "Could be worse." The Doctor, annoyed, points out the details of their situation and asks "How could it possibly be worse?" Igor answers in a very matter of fact tone, "Could be raining." This is followed immediately by a stroke of lightning, a clap of thunder, and a drenching rain.

Bad as our situation is, it could be worse. At least the TI community still has active user groups who know about each other, and of course also has MICROpendium to keep in touch with even the lone user. Back when I acquired my first Radio Shack Color Computer (CoCo), I was looking around for other users to answer my many questions. Our friend Barry Traver got me a point of contact in the Philadelphia area for a CoCo users group. The man was available by phone, and eagerly helped with many questions, but when I asked whether there was a group in the Washington, DC area, he didn't know! There's no forum for the CoCo users of the world, just isolated pockets of users. We TI people should count our blessings!

## SLOWING IT DOWN

Several times in this column we've discussed the matter of creating controlled delays in an Assembly program. In many cases Assembly's biggest strength, its speed, is a curse. In this month's sidebar, we're showing some "near perfect" methods of doing a delay in any Assembly program. The first part is a simple subroutine which uses the user interrupt process to control the amount of the delay. Except for Register 11, this method does not change any registers, so it will not interfere with what's happening in the main code. Its only drawback is that it makes delays only in increments of 1/60th second.

As you'll see in the sidebar, the subroutine is invoked with a BL @DELAY followed by a DATA line. The DATA is the number of 60ths of a second for the desired delay. The DATA number can run from 1 through 32767. That means the delay itself can range from 1/60 second to 546 seconds, which works out to 9.1 minutes! In the first example shown in the sidebar, the DATA is 30, so the delay will be 1/2 second. Since this delay uses the interrupt timing, the delay is independent of the processor or memory speed, so that the delay will be the same on bus-modified TI machines or on Geneves as it is on a standard TI.

## THE EUROPEAN PROBLEM

If your program is to be used primarily in Europe or Australia, where the vertical interval is in 50ths of a second, the delay timing would have to be adjusted in the source code, as for example 25 instead of 30 would be used in the DATA to make a 1/2 second delay. If that's not done, all delays in your program would actually become 6/5ths of their desired value when run on a 50 Hz machine. In some cases, we've made two versions of a program, one for U.S. use, and another for European use.

## VARIATIONS ON THE THEME

In the second part of the sidebar, there's a variation on the subroutine, which shows a method by which the machine can be made to do other things while the delay is running. Here, we've added a BLWP @KSCAN within the loop, so that the user can terminate the delay by pressing a key. This requires one more byte added to the DATA section of the program so a check can be made after the BLWP @KSCAN. Using your imagination, you can see how other functions could be combined into the delay loop. (Another exercise for the student.) Given the speed at which Assembly executes, the computer can do lots of things in each 1/60th second, without affecting the accuracy of your delay.

## HOW IT WORKS

The key to this is the little section of code at label USRINT. That performs only one simple task, incrementing the word of memory at label TIMCNT, then returning to the interrupt servicing routine. The interrupt servicing routine is always there in the console, and activates itself once each 1/60th second during the vertical interval, provided only that a LIMI 2 is being performed now and then. Thus TIMCNT gets incremented once each 1/60th of a second, which is of course the desired case. At the label DELAY, we first take the DATA that follows the BL instruction into a word of memory at DLYTIM. This sets the "target" number for

(See Page 12)

# THE ART OF ASSEMBLY —

(Continued from Page 11)

the delay. Next, the subroutine clears the word at TIMCNT, so we start with zero in that variable. The third operation puts the address of USRINT into the word at >83C4, so this interrupt will be serviced on each vertical interval. The Interrupt servicing routine uses its own workspace, so the RT in USRINT returns to the servicing routine, and does not affect your main-code registers.

Once those things are done, the subroutine enters a loop at DLY1. This loop allows interrupts briefly during each pass by the LIMI 2 and LIMI 0 instructions. It then compares the words at TIMCNT and DLYTIM to see whether the desired delay has been completed. So long as the word at TIMCNT is less than the word at DLYCNT, the loop simply repeats. Once TIMCNT becomes equal to (or greater than) DLYTIM, the subroutine clears >83C4 to de-activate USRINT, then returns to your main code after the DATA item.

**WHAT HAVE WE DONE?**

The Interrupt driven DELAY was inspired by two things. We had previously done a timeout routine for use with Extended BASIC, so that an ACCEPT AT could be limited to a certain amount of time, measured accurately in 1/60th second increments. That routine used the User Interrupt to advance a counter, thus making an accurate timeout possible. We'd also been looking at some source code written by one of our readers. The source code was for a game program that needed delay loops, and there were several different delay subroutines because different registers needed to be preserved.

After giving it a little thought, we figured that the delay subroutine could be made using the Interrupt idea, and that only Register 11 of the program's workspace would be affected. This takes more memory than one of our reader's original subroutines, but is still more efficient than having three or four different subroutines just to do delays.

This method was then used in one of our public domain products, to serve as a timer. That product is a Slideshow program which uses TI-Artist picture files. In that application, the user can select to have each slide in a sequence appear on-screen until a key is pressed, or he may choose to use a timed sequence, in which case each slide appears for a precise amount of time before being replaced. The user of that program enters his number in seconds, ranging from 0.1 second through 300 seconds (5 minutes). We also wanted to give the user the ability to stop the show at any time by pressing FCTN-9. Thus a variation of the second method in today's sidebar was applied and tested. It worked very nicely.

**BUT THEN AGAIN...**

Having been trained as an electronics engineer, and having worked in that profession for 30 years, I can testify to the fact that engineers are never content to leave well enough alone. Left to themselves, engineers would re-design a product for 100 years before sending it into production. Back then, I would have said that nobody could do more re-thinking of things than engineers. Then I discovered programmers. Programmers can re-work one product for much longer than the engineers I'd worked with. That includes your author.

In the fine tradition of never leaving well enough alone, we've added Part Three to this month's sidebar. This does not need a User Interrupt, but simply depends on one that's already there in the form of the screen timeout counter. For icing on the cake, we made this one so that it works as a BLWP vector, without affecting any of the main program's registers. Like the previous ones, this uses a DATA item that can range from 1 through 32767, and produces a delay of that number of 60ths of a second. Since this is a BLWP operation, the subroutine has its own workspace, and gets the data using its own R14. Because the screen timeout counter increments by twos, we have to double the desired number before starting the delay loop. Of course we also want to start with the counter at zero, so we CLR @>83D6 before entering the loop.

For the convenience of our readers, this PART 3 of the sidebar is a complete program that can be typed in, assembled, and run. It'll start by putting the message DELAYING TEN SECONDS on the screen. It will then invoke the delay by doing a BLWP @DELAY, followed by DATA 600. After the ten seconds delay, the program puts a "DELAY IS FINISHED" message on the screen, then awaits a keystroke and exits when it gets one. By my digital watch, this seems to be quite accurate.

Note that the number has to be doubled, which we do by a simple SLA R0,1 operation. Doing it this way insures that the number in R0 is always even, so it will match the number at >83D6 when the desired time has expired. All the subroutine needs to do, then, is just keep allowing interrupts once on each pass through the loop, and then compare the number in R0 to the screen timeout counter at >83D6.

As in the previous case, we could include a BLWP @KSCAN within the loop, so that the delay would be stopped by any keypress. This method, with its need for another set of workspace registers, is not exactly the most efficient in memory use, but it works as advertised, so we'll forgive ourselves for taking an extra 32 bytes in this case. It could turn out that your program needs to preserve all its own registers while the delay is happening, so those 32 bytes could be worth having.

So, once again, dear readers, we leave you with the choices to make yourself. Parts one, two, and three of the sidebar all work, so you can use any of them to advantage in your programs. You can combine features, add other possibilities, and carry this stuff beyond our wildest dreams. In any case, we hope you're successful in your pursuits.

That's all for this month's lesson. The programs we're making are all being made available through our dear friend Dr. Charles Good and the Lima Users' Group. So long as people are still interested, we'll keep trying new things. We're open to suggestions from readers at any time, either for needed "products" or for help with their own work. Next month's topic is again undecided, so watch this space for another surprise.

## SIDEBAR49

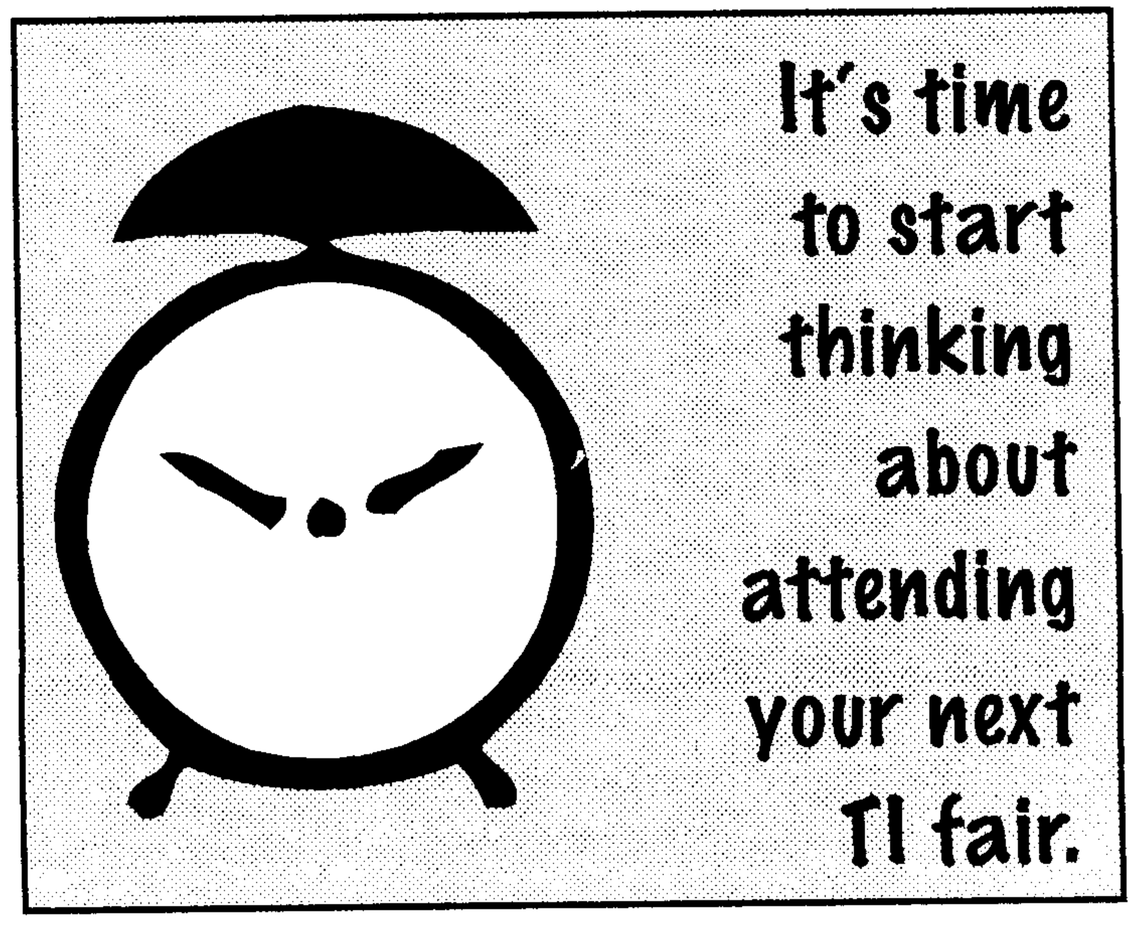
```
0001  * SIDEBAR49
0002  *
0003  * OTHER DELAY METHODS
```

(See Page 13)

# THE ART OF ASSEMBLY —

(Continued from Page 12)

```
0004  * PART ONE - WITH A USER INTERRUPT
0005  * USES NO REGISTERS EXCEPT R11
0006  * MAKES POSSIBLE DELAYS FROM
0007  * 1/60TH SECOND THROUGH 546 SECONDS (9.1 MINUTES)
0008  * THE EXAMPLE SHOWN WOULD GIVE 1/2 SECOND
0009  *    CODE BY:  Bruce Harrison
0010  *    PUBLIC DOMAIN
0011  *    25 November 1994
0012  * IN MAIN CODE, DELAY IS INVOKED THUS:
0013  *      BL   @DELAY       USE DELAY SUBROUTINE
0014  *      DATA 30           NUMBER OF 60THS OF A SECOND TO DE-
LAY
0015  * THE DATA ITEM ABOVE MAY RANGE FROM
0016  * 1 THROUGH 32767
0017  *
0018  * THE DELAY SUBROUTINE IS:
0019  *
0020  DELAY  MOV  *R11+,@DLYTIM GET DESIRED DELAY
0021         CLR  @TIMCNT      CLEAR TIME COUNT
0022         MOV  @INTLOC,@>83C4 SET USER INTERRUPT
0023  DLY1   LIMI 2            ALLOW INTERRUPTS
0024         LIMI 0            STOP THEM
0025         C    @TIMCNT,@DLYTIM  COMPARE TO DESIRED DELAY
0026         JLT  DLY1         IF LESS, REPEAT
0027         CLR  @>83C4       ELSE CLEAR USER INTERRUPT
0028         RT                THEN RETURN
0029  *
0030  * THE INTERRUPT ROUTINE IS THIS:
0031  *
0032  USRINT INC  @TIMCNT      INCREMENT TIME COUNT
0033         RT                THEN RETURN
0034  *
0035  * THE FOLLOWING GOES INTO THE DATA SECTION
0036  *
0037  TIMCNT DATA 0
0038  DLYTIM DATA 0
0039  INTLOC DATA USRINT
0040  *
0041  * PART TWO
0042  * ANOTHER VERSION OF DELAY SUBROUTINE
0043  * THIS TERMINATES AS SOON AS A KEY GETS PRESSED
0044  *
0045  DELAY2 MOV  *R11+,@DLYTIM GET DESIRED DELAY
0046         CLR  @TIMCNT      CLEAR TIME COUNT
0047         MOV  @INTLOC,@>83C4 SET USER INTERRUPT
0048  DLY21  LIMI 2            ALLOW INTERRUPTS
0049         LIMI 0            STOP THEM
0050         BLWP @KSCAN       SCAN KEYBOARD
0051         CB   @>8375,@NOKEY NO KEY PRESSED?
0052         JNE  DLYEX        IF NOT, TERMINATE SUBROUTINE
0053         C    @TIMCNT,@DLYTIM  COMPARE TO DESIRED DELAY
0054         JLT  DLY21        IF LESS, REPEAT
0055  DLYEX  CLR  @>83C4       ELSE CLEAR USER INTERRUPT
0056         RT                THEN RETURN
0057
0058  * ADD THIS TO DATA SECTION WITH THAT
0059  * SHOWN ABOVE
0060  NOKEY  BYTE >FF          A BYTE OF -1
0061  *
0062  * PART THREE - YET ANOTHER METHOD
0063  * USING BLWP VECTOR, AND AFFECTING NONE
0064  * OF THE MAIN PROGRAM'S REGISTERS
0065  * THIS USES THE SCREEN TIMEOUT COUNTER
0066  * AT >83D6, WHICH COUNTS BY TWOS
0067  *
0068  * DELAY3/S
0069  * WITHOUT USER INTERRUPT
0070  * FOLLOWING IS A COMPLETE PROGRAM
0071  * THAT CAN BE ASSEMBLED & TESTED AS IS
0072  * PRODUCES A TEN SECOND DELAY
0073  *
0074         REF  VMBW,KSCAN   REF UTILITIES
0075         DEF  START        DEFINE ENTRY
0076  START  LWPI WS           LOAD WORKSPACE
0077         LI   R0,12*32+5   ROW 13, COL 6
0078         LI   R2,20        20 CHARACTERS
0079         LI   R1,DLYTXT    DELAY MESSAGE
0080         BLWP @VMBW        WRITE THAT
0081         BLWP @DELAY       USE VECTOR
0082         DATA 600          NUMBER OF 60THS OF A SECOND TO DE-
LAY
0083  *
0084  * AS IN PREVIOUS CASES, THE DATA NUMBER
0085  * MAY RANGE FROM 1 THROUGH 32767
0086  * HERE, IT'S 600, FOR A 10 SECOND DELAY
0087  *
0088         LI   R0,14*32+7   ROW 15, COL 8
0089         LI   R2,17        17 CHARACTERS
0090         LI   R1,DOVTXT    DELAY DONE MSG
0091         BLWP @VMBW        WRITE THAT
0092  KEY    BLWP @KSCAN       SCAN KEYBOARD
0093         CB   @>837C,@ANYKEY KEY STRUCK?
0094         JNE  KEY          IF NOT, REPEAT
0095         LWPI >83E0        LOAD GPLWS
0096         B    @>6A         BACK TO E/A
0097  *
0098  * THE DELAY SUBROUTINE IS:
0099  *
0100  DELAY  DATA DLYWS,DLY0   BLWP VECTOR
0101  DLY0   MOV  *R14+,R0     GET DESIRED DELAY
0102         SLA  R0,1         DOUBLE THE NUMBER
0103  *
0104  * WE DOUBLE THE NUMBER BECAUSE THE COUNTER
0105  * ADVANCES BY TWO EVERY 1/60th SECOND
0106  *
0107         CLR  @>83D6       CLEAR TIMEOUT COUNTER
0108  DLY1   LIMI 2            ALLOW INTERRUPTS
0109         LIMI 0            STOP THEM
0110         C    R0,@>83D6    COMPARE NUMBERS
0111         JNE  DLY1         IF NOT EQUAL, REPEAT
0112         RTWP              ELSE BACK TO MAIN PROGRAM
0113  *
0114  * DATA SECTION
0115  *
0116  WS     BSS  32           MAIN WORKSPACE
0117  DLYWS  BSS  32           DELAY WORKSPACE
0118  DLYTXT TEXT 'DELAYING TEN SECONDS'
0119  DOVTXT TEXT 'DELAY IS FINISHED'
0120  ANYKEY BYTE >20          KEYSTROKE COMPARISON BYTE
0121         END
```



# Customizing your AMS to fit your system

By BOB CARMANY

If you bought one of the AMS cards from the Southwest 99ers as I did, it was with great anticipation that you began the adventure of prodding, poking and customizing the system to conform to your specifications.

Everything worked fine until you tried to run ZAEMSPAT (RAGPATCH) to customize the printer (for a Star 10X or NX-1000) or alter the screens of the MacroAssembler and Linker. It seems that the patch files supplied with the AMS don't work as advertised. It appears that the programs were updated after the patch files were created, which resulted in the invalidation of the patch addresses. Never fear, all is not lost!

Having a Star NX-1000,I was determined to alter the printer codes by hook or crook! The first order of business was to print out the appropriate patch files PRTDIST and SCRDIST to have a point of reference. The PRTDIST patch file is the distribution version of the printer codes and SCRDIST is the distribution version of the screen display. Next, print the PRTNX1000 file if that is your designated printer.

The first order of business is to determine which file of the series contains the printer codes. The "FILE.DSK1.AMAC2" directive at the beginning of the patch file gives us that information. Copy the file from your AMS disk to a freshly initialized disk. *Remember to make your changes on a copied file — not the original!* Once you have the AMAC2 file copied. crank up your favorite disk sector editor (DISKREVIEW is excellent). The first byte of the distribution copy of printer codes is shown in "PATCH >00,>0037". The byte value is hex 0037. Simply do a file search for the hex value 0037. You will find it right after the six-byte file header. It is neatly commented in the patch file as the number of lines per page. Edit that byte to >004B as per the PRTNX1000 patch file. Two bytes further on "PATCH >020>0050" is the line length. Change it also to the appropriate value of >0060. Another two bytes brings you to the margin value, etc. Simply go through the file changing the appropriate bytes until you are finished and the job is done.

## Where to get the AMS

The Asgard Memory System (AMS) is available from the Southwest 99ers, P.O. Box 17831, Tuscon, AZ 85731. The group may also be contacted on the Cactus Patch BBS, (520) 290-6277. Prices are $85 for the 128K version, $100 for the 256K version and $25 to upgrade the 128K version to a 256K version.

If you have a printer that uses other control codes than the ones in the commented patch files, substitute the corresponding hex codes to customize the print codes.

The procedure for altering the printer codes in the Linker is exactly the same. You just perform the operations of a copy of the ALNK2 file. When you are done, save them back as the AMAC2 and ALNK2 files in your working copy of the MacroAssembler and Linker, respectively.

Now that we are warmed up, let's take a look at the screen displays of both the MacroAssembler and Linker. This is a bit more complex. This time, you are operating on the AMAC1 and ALNK1 files, though.

Once again. the code starts right after the six byte header. The difference is that you are not starting at the very first byte. The other problem is that there is a whole heap of "00" for the empty fields in the distributed copy. Look at the first patch directive — "PATCH >04,>00". That you are actually starting at the 105h byte of the file (6-byte header plus 4 bytes). The easiest way to find a point of reference is to search for ">0460". From there, go to the 10th byte of the file (hex OA). The patch at that location is a length byte and is for the source file. If your source code normally resides on disk drive No. 1, you can enter that as default. The length byte would he changed to >05 (for the number of characters in DSK1.), The following byte would begin the customized prompt (i.e., 44534B312E). This is the hex equivalent of "DSK1." From there, go through the file changing the codes as needed. Remember that the first byte of each of the patches is the length byte of the following entry. Include periods (ex. DSK1.) in the count of the length. All of the entries are the hexadecimal equivalents of the alphanumeric characters. A good hint is to have a printer manual handy since most have an alphanumeric/hexadecimal table in the appendix.

I save my assembled code to one of my Quest RAMdisks which I have designated as DSK9. I moved down to byte >27 (6-byte header plus >21) for the "PATCH .21,>00" for the object file default. That byte was changed to >05 and the next bytes to 44534B392# or "DSK9." for my chosen assembled code storage. You can also alter your printer if it isn't "PIO" and put in a date default like "00/00/95"; the last patch to make is for the options at absolute byte >88 in the file. I changed mine to "R" after changing the length byte to >01, since I rarely assemble code into compressed object code.

The ALNK1 file is altered in exactly the same way. Simply change the appropriate bytes to reflect the changes that you want to appear on the screen when the program loads and save the file to your working disk.

The whole process should take considerably less than an hour and you are left with customized versions of the MacroAssembler and the Linker. It took me about 45 minutes, but that was starting from scratch and working all of this out for the first time. If you have any questions, please feel free to contact me on Delphi — TOOHEYS, or write to me at my home address: 1504 Larson St., Greensboro, NC 27407.

CAT-TO-DSK

# Keeping those files straight

By BRUCE RODENKIRCH

I have been keeping myself busy with my TI-994a and my Geneve assembling library disks for the Greater Akron Group and the Northcoast UG in Cleveland. The Lima Library, Genie, Packet Radio, BBS's and Internet News Groups have been a good source of files on TI correspondence and programs, humor, recipes, science and other files of general interest. When I write my library update page for our user group newsletter I find it helpful to include a list of all the programs on the disk. One of the reasons for this listing in addition to informing the club members what is on the disk is that it is very helpful to be able to use the Find String routine of TI-Writer to locate individual programs.

Rather than type all of the file names, numerous is some cases, I have been using DM-1000 to transfer the disk catalogs to a D/V80 file. Just in case you don't know how to do that, here is the procedure: From any DM-1000 screen press FCTN-3 until the screen appears that says "Enter List Device." Beneath this line is the printer designation, which will be PIO in most cases. Replace this with "DSKn.diskname" or whatever is appropriate. Press enter for the other prompts except for the prompt "Save to disk (y/n)." Enter "n" and then catalog the disk as usual and press FCTN-7. This will print the catalog to disk as a D/V80 file.

For my purposes, where all I want is a list of file names, the catalog has to be edited to delete the unneeded information. I also add a comma after each program name and a carriage return at the end to aid in formatting.

The above method was helpful but it seemed to me the computer could be made to do more of the work. I remembered using Jim Peterson's Menuloader program which printed a catalog to screen and would also "RUN" selected XB programs. I studied the program and found I could delete much of it and retain the parts I wanted. After some other embellishments I came up with the CAT-TO-DSK program listed below. This produced a D/V80 file with commas after each program name and a carriage return at the end for neat reformatting. Each program name starts a D/V80 line, which would be handy when one wishes to add a file description on that line. A sample of this printout is as follows:

DS-1115: (no cr)
3/COL,"
ARROWDYNAM,"
BOOT,"
LOAD,"
etc."
(cr)

As you can see it is all set up for reformatting or adding file descriptions. This would also be handy to create a library file of all your disks because it so easy to add new disks later and edit.

As you can see it is all set up for reformatting or adding file descriptions. This would also be handy to create a library file of all your disks because it so easy to add new disks later and edit. For example, one would not need all of the miscellaneous work files (such as LOAD in the above example) that would show up on a catalog of the Funnelweb disk for example. The name Funnelweb could be used and the other names deleted. If you were looking for a Funnelweb disk that would be the name you would use in the Find String search. You can also change the program names to more than 10 letters for clarity. For example: PNUTBUTCKY could be change to PEANUT BUTTER COOKY, which would make your search easier, too, when you can't remember what name was used. Don't forget to keep the names in capital letters because FS is case sensitive.

The program is a simple one and it should not be too difficult to modify it to suit your needs. You may want to do away with the disk prompts and assign disk numbers to speed up the cataloging of large numbers of disks. All you have to do is change Line 210 to say D$="DSKn." where n is the disk drive number you wish to use as a source disk. The prompt will be ignored but will serve as a pause as you load the next disk. Change line 220 to show the number of the disk drive where the list will be printed. I left the Tigercub logo as the cursor to remind me of the original author of the program but you could change that if you prefer.

That's pretty much it. Thanks to Jim Peterson for Menuloader and all the other programs he wrote for the TI. When you use this program say a little prayer for Jim and tell him he is not forgotten.

## CAT-TO-DSK

```
100 ! CAT-TO-DSK (Public Dom
)by Bruce Rodenkirch, 1514 J
ulian St, Cuyahoga Falls, OH
 44221. May 1995
110 ! Based on MENULOADER by
 Jim Peterson
120 ! Will print disk catalo
g file names only to disk as
 a dv/80 file.
130 ! can be used to create
a library file of your disks
 which can be easily searche
d by FS in TIWriter.
140 CALL PEEK(8198,A):: IF A
<>170 THEN CALL INIT
150 CALL CLEAR :: CALL LOAD(
8196,63,248):: CALL LOAD(163
76,67,85,82,83,79,82,48, 8)
160 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
170 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
```

(See Page 16)

# CAT TO-DSK —

(Continued from Page 15)

```
6,4,91):: CALL LINK("CUR SOR
")
180 CALL CLEAR :: FOR S=1 TO
 14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2 ,16
)
190 CALL SCREEN(5):: CALL VC
HAR(1,31,1,96):: DISPLAY AT(
1,8):"CATALOG TO DISK"
200 DISPLAY AT(10,4):"TARGET
 DISK IS DSK2."
210 DISPLAY AT(12,4):"SOURCE
 DISK ? (1-6):" :: ACCEPT AT
(12,25)SIZE(-1)VALIDATE( "12
3456"):D$ :: D$="DSK"&D$&"."
220 OPEN #1:D$,INPUT ,RELATI
VE,INTERNAL :: INPUT #1:N$,A
,J,K
230 OPEN #2:"DSK2."&N$,DISPL
AY ,VARIABLE 80 240 PRINT "D
ISK ";N$;" IN PROCESS"
250 PRINT #2:N$&": "
260 FOR X=1 TO 127 :: INPUT
#1:P$,A,J,B
270 IF P$="" THEN 300
280 PRINT #2:P$&", "
290 NEXT X
300 PRINT #2:CHR$(13) :: ! P
RINT C/R
310 CALL CLEAR :: DISPLAY AT
(20,6):"INSERT NEXT DISK" 32
0 CLOSE #2
330 CLOSE #1
340 GOTO 190
```

# NEWSBYTES

## Date changes for TI Fall Faire

M.U.N.C.H. (Massachusetts Users of the Ninety-Nine Computer Hobbyists) has moved up the date of its TI New England Fall Faire because of a schedule conflict, according to James W. Cox of the group. The even t is now scheduled for Sept. 30 at the Emanuel Lutheran Church, 200 Greenwood St., Worcester, Massachusetts.

Set-up time begins at 8:45 a.m. Sept. 30. Faire hours are 10 a.m.-4 p.m. Admission to the event is free and refreshments will be served.

For information, contact Cox at 905 Edgebrook Dr., Boylston, MA 01505, or call (508) 869-2704 after 6 p.m. Eastern Time.

## Groups get new addresses

New mailing address for the TI99 Users of Perth is Secretary TIUP, 20 Hudson St., Bayswater 6053, Western Australia.

New mailing address for the Pittsburgh TI Users Group is P.O. Box 1212, Munhall, PA 15120-1212.

## Classic car parts found on-line

Car-Trek, a dial-in national listing of available classic auto parts and cars, is available via computer modem on a BBS which can also be accessed through the Internet. Users can key in the name of the needed part and allow the computer to search the database, according to Kevin Price, Car-Trek president. Car-Trek is offering free listings to sellers through August 1995.

For more information, call 1-800-728-3240 (voice) or (614) 842-9213 (modem, setting 8-N-1).

## Austrian users change address

Alfred Slovak of the TI- and Geneve User Group Vienna says the group will move to a new location as of fall 1995. The new address is TI- and Geneve User Group Vienna, Grossbauerstr. 24, A-1210 Vienna, Austria. Telephone numbers are (office) 0043 1 4004 30343; (fax) 0043 1 4004 30399.

## 1995 TI FAIRS

### APRIL

**Lima Multi Users Group Conference**, April 29, Reed Hall, Ohio State University at Lima. Contact Lima Users Group, P.O. Box 647, Venedocia OH 45894, or call Charles Good (evenings) at (419) 667-3131 or Internet cgood@osulima1.lima.ohio-state.edu.

### SEPTEMBER

**10th International TI-Meeting**, Sept. 22-24, Wohlfahrtsgebäude der Wiener E-Werke (Welfare Building of the Vienna Electricity Board), Wachaustr. 28, A-1020 Vienna, Austria. For information write Kurt Radowisch, TI- and Geneve User Group Vienna, Fugbachgasse 18/17, A-1020 Vienna, Austria.

**TI New England Fall Faire**, Sept. 30, Emanual Lutheran Church, 200 Greenwood St., Worcester, Massachusetts. Contact Jim Cox, 905 Edgebrook Dr., Boylston, MA 01505 or (508) 869-2704.

### OCTOBER

**Chicago International TI Faire**, Oct. 28, Evanston Public Library. Contact Hal Shanafield, (708) 864-8644.

## 1996 TI FAIRS

### FEBRUARY

**Fest West '96**, Feb.17, Ramada Inn, 1601 Oracle Dr., Tucson, Arizona. Contact SouthWest Ninety-Niners User Group by sending e-mail to twills@primenet.com. Or call the Cactus Patch BBS at (520) 290-6277.

**This TI event listing is a permanent feature of MICROpendium. User groups and others planning events for TI/Geneve users may send information for inclusion in this standing column. Send information to MICROpendium Fairs, P.O. Box 1343, Round Rock, TX 78680.**

# READER TO READER

Alfred Slovak, Fugbachgasse 18/17, A-1020 Vienna, Austria, writes:

When using Archiver 3.03g, problems occur when I archive to a 2880-sector disk. I have been using Archiver 3.03g on my Geneve with great success on 1440-sector disks. I am equipped as follows: Geneve, HFDC at >1000 with a Quantum Q540 drive, Cor-Comp controller at >1100 with one 3.5-inch 720K drive (80 tracks, 18 sectors) as DSK1 and one 5.25-inch 360K drive (40 tracks, 18 sectors) as DSK2. And that's the point. Archiver works without any problem on the 5.25-inch drive. But on the 3.5-inch drive there occur problems as follows:

When I archive files from DSK2 to DSK1, everything seems to be normal. Archiver treats one file after the other and in the end shows the cyan screen with black foreground saying, "Finished, press any key!" But, then, when cataloging or extracting an arced file, sometimes I get the response "This is not an ARCfile," even though Archiver created that very file just a minute ago. In the catalog, these files appear as IF128 like any other ARCfile.

I first threw my suspicion on the disk drive, but I had it tested by a hardware specialist at my work who found it error-free. Now I use a new version of MDOS, V.2.21, which seems to be okay. First I tried to archive a bunch of files and extract them again from DSK2 to DSK1 and vice versa, and everything seemed okay. I used — of course — new disks for these trials, but new disks are usually relatively empty, only a few hundred sectors used. Ah, I thought, now it works with MDOS 2.21, perhaps 2.00 had a few bugs. And then I found out, by trial and error, that the message "This is not an ARCfile" appears only when the file is located at sectors with numbers above 1440, meaning on the back side of the disk. So the problem never occurs if the disk is not full enough.

When I copy a "good" file from the low sectors to the high ones, it changes into a "bad" file, and it remains a "bad" file when I copy it back to the low sectors.

As well, when I copy a "good" file from my 5.25-inch drive to the 3.5-inch drive, to high sectors, it changes into a "bad" file and remains "bad" wherever I copy it.

I do not understand enough about programming or disk editing (I program only in BASIC) to find out what the problem is or even to solve it. I suppose there is a byte somewhere that is not written correctly on the back side of the disk.

I experience this problem only with ARCfiles; all other files (programs, text files, etc.) can be accessed, no matter where they are located on the disk.

I wonder if anyone can help me with this problem. I could work around this problem by using only the first 1439 sectors of disks or archiving only on disks with a maximum of 1440 sectors. But I arced a few annual volumes of MICROpendium disks on 3.5-inch disks, as the 12 files of a complete year won't fit on a 1440-sector disk, but they fit well on a 2880-sector disk. So I arced all my MICROpendium disks till the message, "Finished, press any key," appeared, and relied on the files to be okay. Now I am afraid (do I have to be?) that about half of these arced files are lost. I hope readers can help me to regain these files, as I don't have the original disks any more.

I wrote to Barry Boone in February concerning this matter, but so far he hasn't responded.

By the way, has anyone a documentation of Archiver 3.03g. I have only one of version 2.0; I found it by accident in the stuff of a friend who gave up his Geneve.

Any help or information would be greatly appreciated.

Next are my problems with ABASIC, 2.99A as well as 3.00. I have difficulties when using Break (F4) to interrupt a program. F4 (or Alt4 or CtrlC or Page Down) works only when the program is halted by an INPUT or ACCEPT. Now, I found out that this only occurs when a LOAD-program is found and executed on loading ABASIC. If no LOAD-program exists in the current path, ABASIC will respond * READY * and wait for further inputs. Programs run "by hand" from this point will react on each of the above Break keys at any time in the program. Even a LOAD-program can be accessed by OLD or RUN without affecting the reaction of Break keys, as long as it is not in the current path (thus, not found by the computer on loading ABASIC) or it is saved to the current path at a late time (thus, not present on loading ABASIC).

It seems to me that the execution of LOAD on loading ABASIC changes a bit somewhere, that prevents the proper reaction of Break keys. Not even entering NEW after LOAD has been executed automatically will reset this bit. The trick I knew on the 99/4A to bypass the execution of the LOAD-program by holding down the space bar won't work on the Geneve (at least mine).

Any help ... (see above).

By the way: I have now (as I believe) the newest versions of MDOS (2.21) and ABASIC (3.00) But what I need urgently are full documentation of these programs. I feel there are a lot of fine features and possibilities, which I can't use, as I don' know of them.

Any help ....

**Reader to Reader is a column to put TI and Geneve users in touch with other users. Address questions to Reader to Reader, c/o MICROpendium, P.O. Box 1343, Round Rock, TX 78680. We encourage those who answer the questions to forward us a copy of the reply to share with readers.**

# Competition changes ownership

Competition Computer has been taken over by Kyle Crichton, who is also producing the TI99/4A CD-ROM (see June 1995 MICROpendium).

The company continues with its 1-800-471-1600 toll-free order number.

New address for the company is 350 Marcella Way, Millbrae, CA 94030. The other voice phone number is (415) 697-1108; fax is (415) 697-7406.

Crichton notes that in addition to Competition's merchandise purchased in the takeover, the company also has stocks of TI-compatible hardware and software he acquired previously.

MICRO-REVIEWS

# TI Workshop, Halls of Moria

By CHARLES GOOD

## TI WORKSHOP by DataBioTics

When this module was released in 1987 by DataBioTics few purchased it, and it disappeared into obscurity. It was listed in the spring/summer and the fall 1988 Triton catalogs under the name Magic Memory. Now Tex-Comp Ltd. (the new Tex-Comp) is again offering this unique product. The module is unique because it is the largest module software ever produced for the 99/4A. Inside is 64K of ROM, bank-switched 8K at a time into the console's cartridge ROM (or RAM) area at >6000 - >7FFF.

TI workshop is a combination disk manager, sector editor, and assembly language program development environment. Program development software includes a text editor designed specifically for writing assembly source code, an assembler, a disassembler, a debugger, and a memory viewer/editor. All of this is in the cartridge, accessible from a well-designed menu system. It is easy to bounce back and forth between one workshop application and another, for instance, between the memory editor and the debugger. No disk-based software (such as the TI E/A editor or assembler or Superbug debugger) is used. You do, however, need a disk system to load and save source files and assembled code and you need a 32K memory expansion. A printer is nice too. From anywhere in the workshop you can do a screen dump. You can, of course, print entire source files and there is a large variety of printing options relating to the assembly and disassembly of assembler code.

The disk manager is really first rate, resembling DM1000 in its capabilities but better than DM1000 because it links directly with a sector editor. Even normal 99/4A users who are not assembly programmers will find this cartridge disk manager convenient and useful. I use it when my Horizon RAMdisks crash to help me reload the RAMdisk ROS and, if necessary, load my files back on to the RAMdisks. I have disk managers on my RAMdisks, but they are useless if the RAMdisk locks up.

TI Workshop's Disk Manager is compatible with drives 1-9 and RAMdisks that are at any CRU address. You can page back and forth on screen within a directory listing without first having to move the cursor to the top or bottom of the displayed page. You can mark all files for copying, protecting, unprotecting, or deleting with a single keystroke, or you can do this individually from a disk directory. The directory lets you separately mark each file on the disk with up to three commands and then execute all the commands. File commands include Copy, Delete, Edit (loads DV or DF80 file into editor), Find (switches to sector editor and lets you view/edit sectors containing the file), Load (an EA5 or EA3 file which will then run), Move (copies, then deletes file from source), Output (text file to printer), Protect, Rename, Temp (temporarily unprotects a file so an action such as Rename can occur then protects file again) and Unprotect. Disk Utilities of the disk manager include Copy (sector copying with or without bitmap), Erase (sweep), Initialize and Format A Bunch (box format).

Memory manager lets you Show Dump Edit Initialize (fill designated memory with the same data byte) Copy (from one area of memory to another) and Search. You can do these things with VDP and CPU memory, but you can't check on GROM memory. You can also set a CRU bit at any CRU address to activate the peripheral at that address. This allows you to manipulate the memory of the device's DSR. For example, I have a Cor-Comp disk controller. If I select "CRU Bit Control" from memory manager (or from several other places within TI Workshop) I can activate the disk controller's DSR. First, I type 1100, the address of the DSR. Then I change the first address at this location to >0001 to turn on the DSR. I can then page through the Cor-Comp disk controller's DSR and get a display and/or printout in both hex and ASCII. I notice that part of this DSR code says in ASCII "c Miller Graphics", which verifies that Craig Miller wrote the Cor-Comp disk controller DSR code.

You can also manipulate extended VDP memory if you have an 80-column card. Such cards have much more than 16K of VDP memory with the extra memory going beyond the normal >3FFF where console VDP memory ends. TI Workshop lets you manipulate this extra memory if you have it.

The Debugger is a menu based-modification of Edgar Dohman's Superbug II. In fact, Mr. Dohman is credited as the author of most parts of TI Workshop. You can view and set all the registers, set and list breakpoints, single step or slow execute (three instructions per second), and execute at regular speed based on where you set the program counter register.

Disassembly has all sorts of screen display and/or printer options. You can activate a DSR, copy the DSR code to a safe memory location, and then disassemble the DSR code as if this code was still resident at the DSR location (>4000 - >5FFF). The instruction book has a Debugger and Disassembler tutorial. You are given some sample code to enter and told what to expect as the Debugger and Disassembler do their thing.

The source code editor resembles the TI Editor/Assembler module editor with some enhancements. Tab settings are more convenient, there are more convenient text manipulation features and there is a true lowercase character set. An automatic mode is available that gives auto tabs and capitalization in columns 1-25, handy for distinguishing comments from actual assembly source code.

The assembler has some print options besides the usual R, C. and L choices of the TI assembler. The most interesting of these is Cróss Reference. This gives you a printed list of program labels, line num-

(See Page 19)

# MICRO-REVIEWS —

(Continued from Page 18)

bers where the labels occur and line numbers where the label is referenced.

The cartridge comes with a detailed 60+-page user guide. One of the appendixes in this manual is unusual. It tells you how to permanently back up and customize TI Workshop. Just trot out your TI eprommer (everybody has one, right?), open up the cartridge, remove the EPROM from its socket and copy it to disk. Now you can make a few changes here and there as suggested in the appendix and then save your customized TI Workshop back to a new EPROM.

This is an expensive cartridge, $49.95, but it is an expensive cartridge for Tex-Comp Ltd. to manufacture. The 64K EPROM isn't cheap. Also, copying costs for the large user guide are not insignificant. TI Workshop is a product that all assembly programmers should consider. It offers the assembly programmer seamless access to a combination of features found nowhere else in a single software package, part of like Miller Graphics Advanced Diagnostics, Explorer, and Diskassembler all combined into one easily accessible menu system. Only PC99 running on an IBM compatible can offer a similar set of features all in one package. TI Workshop is by far the best cartridge-based disk manager/sector editor available for the 99/4A and because of that may be worth the expense for mere mortal 99/4A users as well as programmers.

## HALLS OF LOST MORIA
## by Michael Veprauskas

This is a Tunnels of Doom game requiring the TOD module. Mr. Veprauskas wrote it for his children. I know, you are probably thinking, "Ho hum, yet another TOD game." That is what I thought, too, when I was first asked to review Halls. But this game really is different and enjoyable. My 14-year-old son Colin took Halls for a test drive and really liked it. He has experimented with other TOD games in our user group library and claims Halls is the best of the lot.

One thing different about Halls is that it is winnable! There is an end point, and with a little persistence you will get there. Colin did it in one long afternoon of play. It took me a little longer. I am an adventure game klutz and here publicly confess that I have not, without aid from cheat books, solved any of the Scott Adams or Infocom adventures. It is really good for my ego to know that I can actually solve Halls. In addition to the game file, there is a little TI BASIC program that lets you know your score based on how many treasures you collect, how many floors you go down, etc. TI BASIC lets you run this program without removing the TOD module.

The second thing different about Halls is its documentation. It is massive and professionally produced. This is not your usual on-disk short doc file or pile of poorly copied typewritten sheets. You get an 18-page book with lots of nice black-and-white graphics giving you the whole story of Moria, a description of the 10 underground levels, a summary of stats and abilities of the various types of people in your party and an extensive glossary of names and nouns you are likely to encounter in the game. You also get some "Stat Sheets" to help you keep track of things. You write on the stat sheets for each adventurer a name, class, hit points, wounds, level, weapons with their damage and available ammo, armor and shield protection and magical items. This is a lot of stuff to remember. Of course you can have the game display this info, but this is a bit cumbersome. It is really handy to have all these data neatly organized on a single sheet of paper.

If you want to start your adventure right away there is a predefined game with predefined adventurers at a predetermined location described in the user guide. A "Quick Stat Sheet" is included with the game that lists these predefined adventurers along with their names, hit points, and class. You just select "Continue Current Game" from the TOD module menu after you load the game. I really like this because I am too lazy to construct new dungeons or restock old ones when starting a new type of TOD game. The predefined game is the one I finished. You can, of course, make new dungeons and replay Halls once you solve the predefined game.

Halls of Moria comes on a SSSD disk. It is commercial and available from Ramcharged Computing for $5. Included, of course, is all the fancy documentation.

**ACCESS:**

Tex -Comp Ltd (for TI Workshop). 425 E. Arrow Hwy. #732, Glendora, CA 91740. Voice phone (usually answered by a real live human) (818) 339-8924. Fax (818) 858-2785.

Ramcharged (for Halls of Lost Moria), 1-800-669-1214.

Charles Good (send products for review in this column). P.O. Box 647, Venedocia OH 45894. Voice phone (419) 667-3131. Internet email cgood@osulima1.lima.ohio-state.edu or good.6@osu.edu

# The AMS card revisited

**By BOB CARMANY**

The first week in June marked the long-awaited arrival of the AMS card that I had ordered at the Lima TI Fair. After a couple of weeks of prodding, poking and experimenting with the card, I am reminded of the old rhyme:

*There was a little girl*
*Who had a little curl*
*Right in the middle of her forehead.*
*When she was good,*
*She was very, very good,*
*But when she was bad she was horrid.*

My experience with the AMS card has been a lot like that rhyme. What is good about the hardware and software is really superb but there are other aspects that are very bad. The hardware portion of the system is very good. No problems slipping it into an appropriate slot in the P-Box. It

(See Page 20)

# AMS REVISITED—

(Continued from Page 19)

tested out fine as a 32K card and the pass mode worked well when I loaded the 72K TI-Nopoly program into the card. The number of 4K pages added up to the full 256K. So the entire hardware aspect of the system was outstanding. Even the card construction was the best that I have seen in a long while. In fact, the 256K (or 1 meg with 512K S-RAMs) gives the 99/4A almost unlimited programming potential. And, for once, the programming tools are available.

As I started to modify the basic system, I found that some things worked and others didn't. It was easy enough to modify ABOOT and ASEA with the respective patch files and ZAEMSPAT (actually RAGPATCH). Just make the changes in the patch files, run the program and everything works just fine.

The script file AEMSSYS was another matter entirely. I have yet to get the Load command to work properly. It is supposed to duplicate "Load and Retain" from the main menu but I have not been able to get it to load a file from either physical disk drive or RAMdisk. Ironically, everything else worked, though!

The Library Manager, MacroAssembler and Linker (all supplied with the card) are three really superb programs. I have tried all three and they all work flawlessly. They are going to be very valuable tools in the generation of new software for the AMS system. I have always liked the MacroAssembler and the Linker even adds the necessary code segments to enable the pass mode for long programs automatically which eliminates any user calculations or other extra effort.

On the other hand, none of the patch files for these three programs work. It appears that the programs were upgraded, which changed the patch addresses, but the data in the patch files weren't changed. I was disappointed that I couldn't use the PRTNX1000 file to customize my version for my NX-1000 printer. The screen display patch files didn't work either.

I can live with all the failings of the software that I have discussed so far. The generic printer profile will probably work with my NX-1000 with little modification and I can live without having my favorite programs automatically loaded when I boot the AMS system from my RAMdisk. I can even live without ASHOE and ASEA, which summarily "trash" my Quest RAMdisk at CRU >1000. They are all relatively minor nuisances.

The really "horrid" problem is the DSRLNK that is used with the AMS system. It doesn't leave the usual data at >83D0 and 83D2 that virtually all of the others do. The result is that *any* program that uses boot tracking will load but not execute properly. There is no real reason not to write to those two addresses. They are infrequently used in programming applications and there is no logical reason not to use those standard addresses for a "data trail." I fear that until the DSRLNK is rewritten, software development will be slow. I hope Art Green will take it upon himself to make this modification now that the AMS card is once again in production. This is a modification that should be made as soon as possible! Try to load and run Funnelweb from the AMS card and you will see what I mean!

Does this mean that I am disappointed with the AMS card? Not by a long shot! Would I buy one if I had it to do all over again? Of course I would! Just imagine the possibilities — multiple text file storage and swapping from the 40-column version of F'WEB, much larger and more complex utilities and games (e.g., TI-Nopoly) and virtually unlimited programming space. The only limitation is the imagination and creativity of our programmers. You see, there is no more 32K limit!

# SCSI installed, AMS has new utilities

**By MIKE DOANE**

*(Reprinted from the July 1995 newsletter of the SouthWest Ninety-Niners)*

It is now officially summer and the *heat* is on! I now have the correct version of MDOS installed on my Geneve and now my SCSI (small computer systems interface) is fully operational. I have been busy loading up my PagePro pictures and font files. I haven't printed a picture from the hard disk yet because I got into the habit of working from my RAMdisk and I haven't broken the habit. Still no luck with Picasso, but I'm still trying.

TI-Base is my next project to tackle from the SCSI. I truly enjoyed using my Horizon RAMdisk to run TI-Base, but I have always been a little leery about committing a large database file to an electronic memory storage medium. It becomes counterproductive to use the speed of the HRD to do any "sorting" if you have to save and load the database from a floppy. I'll keep you informed of my progress.

I am not trying to rub your noses into the fact of my SCSI being operational while the TI version is not, but rather I am trying to keep you informed of programs you can use when the TI EPROM is released.

I talked to Bud Mills on June 20 and he informed me there has been more progress on the EPROM. He is hoping to release it *soon*! (Was that a big groan I just heard?) I also talked to him concerning the FC1 card. This is the floppy controller card which will work in conjunction with the SCSI controller. The "semi-official" word on this from Bud is with the new EPROM we *should* have the ability to "read/write" disks using the "IBM" format. This will be a large step toward compatibility for you "C" programmer out there. It sure will make it a lot easier for you to "port" your programs back and forth from that nasty ol' PC you have at work! Bud sounded very optimistic about the forthcoming release of the EPROM. I must ask all you TIers out there to have a little more patience. Bud has never let us down and I don't think he intends to break his record with this project.

The AMS (Asgard Memory System)

(See Page 21)

ge 23

# SCSI AND AMS UPDATE —

(Continued from Page 20)

has been completed and released. The cards have been received with mixed reviews. We have received reports that there might be a problem with two of the cards we sent out. Those cards are being shipped back to us for troubleshooting. We have also received reports that others who have received theirs are experiencing no problems. We had a problem with copying the second disk of the three-disk set of programs/utilities. We did not find out about the copy problem until after we sent out the first 10 orders. We will be reissuing those disks.

Some people have also reported a problem with using the "TIWForm" program included on all three of the disks. This is a program which was made to print out the documentation on each disk. I must take the blame for this oversight. Since I never take the time to follow the instructions, I usually don't have to read the documentation! I never used this supplied utility to print out the documentation so I was not aware of the problem. The TIWForm program is a formatter written by RAG Software (R.A. Green) and since most of the documentation and programs were written by R.A. Green there are special formatter commands that only work with his formatter. The TIWForm program *should* be the same for all three disks. You must, however, turn the console off after printing each document as the formatter doesn't allow you to "back" out or "redo."

Jack Mathis and Shawn Baron have been hard at work creating new utilities for the AMS. At last report, Shawn has written a new testing program and has several other projects imagined. Jack has been working on a "disk copier" which will enable you to "read" 720 sectors from a disk, "store" them into the AMS memory and then "format" and "write" those sectors to another disk. This will allow you to copy disks in *one* pass! Those sectors are still in memory so if you want to copy another disk then you can do so *without* "reading" from your master disk! This program also will automatically determine what size AMS card you have. If you have a 384K or larger AMS you can "copy" 1440-sector disks. (You can't do a 1440-sector disk on a 256K AMS because 1440 sectors equals 360K of memory; each "K" equals four sectors approximately.) The Super AMS Disk Copier is *only* a disk duplicator, it is not a file copier. I imagine Jack will hear from you librarians on this program.

There has been no final decision regarding a "battery backup" system
AMS but I consider it a dead i
would be nice to have an "insta
menu program, but if the memo
corrupted or a bad program gets in
then you would have to "drain" the
ry to get rid of the problem. I sugg
stick with the Horizon RAMdisk for y
"boot" or "menu" programs or buy a "Super-cart" cartridge for this ability. We have been considering a switch to "clear" the AMS memory in case of corruption, but it will not be sold as a kit or as an "add-on" bulletin. It will more than likely be a "technical tip" article such as the conversion to a switching-type power supply series.

CHIP
PR
BAD

Jack and David Ormand have been doing some investigation on the "mapper" of the AMS board and they have come to the conclusion that this board *should* be capable of up to 16 megabytes of memory. I don't have the slightest idea why you want a program this large (especially considering the fact it would take you a week to load it in from a floppy drive) but the capability is there. This just shows you what will happen if you leave those two guys alone without proper supervision!

# USER NOTES

## Tips on Myarc controller and Artist Cardshop

The following items were written by Ron Warfield of the British Columbia 99ers User Group. We found them in the West Penn 99ers News. The first tip is in reference to problems encountered when a Myarc disk controller is used to read a disk that was formatted using a CorComp controller.

The problem is that the Myarc controllers have difficulty reading CorComp disks. CorComp controllers put the disk information at the very top of the read sector and TI and Myarc leave space at the top.

TI controllers had no problems with this because they are slow, but Myarc controllers are so fast that they miss this information.

All I do is "REDISK" — copy the disk on a TI system and the problem goes away.

**ARTIST CARDSHOP**

I use Artist Cardshop a few times during the year and enjoy it very much. After I got my 24-pin printer, I noticed that the cards were much longer than they used to be.

To fix the problem, I used a sector editor to search for the printer codes to see what they were set at. I found that the codes were for 8/72-inch linefeeds on an Epson printer.

Well, when you have a 24-pin printer, the same code sets the printer to 8/60-inch linefeeds, hence the larger picture. To convert this to a better linefeed you have to find something a little closer to 8/72, only in 160ths or 360ths linefeeds.

Well, 20/180 is almost perfect, so I changed the codes and it worked perfectly. Be sure to make this change to a backup copy of your original Artist Cardshop disk. Then, working with a sector editor, edit the file PRINTU. Search for the string 184108 and change it to 183314.

## Subprogram primer

The following item orignally appeared in the newsletter of the User Group of Orange County. It was written by Jim Swedlow.

Extended BASIC has four (or more) ways of writing utility routines: subrou

(See Page 22)

# ER NOTES

(Continued from Page 21)

bprograms, assembly language and DEFinitions. Each of these and cons.

est way to describe a subprogram lain how it differs from a subrou-

ubroutines are called by line number nile subprograms are CALLed by a name.

• Subroutines use the same variables as the main program. In subroutines, variables are entirely different unless noted in the CALL statement.

• Subroutines can be anywhere in the program while subprograms must be at the end.

Since you call subprograms by name, you do not lose their identity when you RESequence. However, you lose the power of ON GOSUB, and passing variables can be tricky if many are involved.

Enter and run this sample program:

```
10 A,B=10 :: PRINT A;B;C ::
CALL TEST
20 CALL NEWS(B) :: PRINT A;B;
C
30 SUB TEST :: A=5 :: PRINT
A;B;C :: SUBEND
40 SUB NEWS(C) :: PRINT A;B;
C :: C=4 :: SUBEND
```

You should get this output:

| | | |
|---|---|---|
| 10 | 10 | 0 |
| 5 | 0 | 0 |
| 0 | 0 | 10 |
| 10 | 40 | 0 |

If you think this through, you will see that there are nine variables in thjis program: A, B and C in the program; A, B and C in TEST; and A, B and C in NEWS.

When you called NEWS(B) the value of B was passed to NEWS and processed as C in that subprogram. Play with this program (including inserting BREAK statements) to test this out.

## Memory expansion chip checker

The following item first appeared in the *The Front Ranger*, the newsletter of the Colorado Springs TI user group. It was written by Joe Nuvolini.

This program checks the 16 4116 chips on your 32K memory expansion card, reporting the condition of each chip. The program doesn't tell you if other parts of the card are defective, but if you are having trouble it could save a lot of time pinpointing a specific chip.

The program requires Extended BASIC and a memory expansion.

### MEMCHECK

```
1 !*************************
*  MEMORY EXPANSION CHECK  *
*     by Joe Nuvolini      *
* VAST News Vol 11 *7 1995 *
```

(See Page 23)

# USER NOTES

(Continued from Page 22)

```
**************************
 !081
100 CALL CLEAR :: PRINT "MEM
ORY EXPANSION CHECKER": : :!
176
110 PRINT "SINCE PROGRAMS LO
ADED FROM  DISK IN XB LOAD I
NTO THE    32K MEMORY, THIS
PROGRAM    SHOULD BE KEYED I
N IF YOU   SUSPECT YOUR 32K
CARD IS BAD": : :!050
120 PRINT "TO UTILIZE THE IN
FORMATION  PROVIDED BY THIS
TEST,ORIENTYOUR MEMORY EXPAN
SION BOARD WITH THE TWO ROWS
 OF 4116   CHIPS AT THE TOP.
": : :!086
130 PRINT "ENTER:":"  1 TO T
EST TOP ROW/CHIPS     2 TO T
EST BOTTOM ROW/CHIPS  3 TO E
ND" !025
140 CALL KEY(0,K,S):: IF S=0
 THEN 140 !208
150 R=K-48 !083
160 IF R=1 THEN A=-12288 ELS
E IF R=2 THEN A=12287 ELSE I
F R=3 THEN CALL CLEAR :: END
 !000
170 IF K<49 OR K>51 THEN 150
 !201
180 IF R=1 THEN N=35 ELSE N=
27 !060
190 CALL CLEAR !209
200 IF R=1 THEN PRINT "TEST
 OF TOP ROW OF 4116'S." ELSE
 PRINT "TEST OF BOTTOM ROW O
F 4116'S" !098
210 PRINT :"READING FROM RIG
HT TO LEFT..": :!244
220 FOR I=0 TO 7 !061
230 IN=2 !080
240 CALL INIT !157
250 CALL LOAD(A,IN)!129
260 CALL PEEK(A,D)!051
270 IF IN=D THEN PRINT "CHIP
 U";STR$(N);" IS OK" ELSE PR
INT "CHIP U";STR$(N);"IS BAD
" !225
280 N=N-1 !022
290 NEXT I !223
300 PRINT !156
310 PRINT "PRESS ENTER TO CO
NTINUE ":X$ !063
320 CALL CLEAR :: GOTO 110 !
016
```

**MICROpendium pays $10 for items sent in by readers and used in this column. Send them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.**

# CLASSIFIEDS

## FOR SALE

### FOR SALE

TI99/4A and Myarc Geneve systems. Two P.E. boxes with Myarc cards — floppy controller, HDFC, 512K memory card, RS232, 9640 card, mouse — TI99/4A (blk-&-silver), dual Teac floppy drives, 20 meg Seagate hard drive, Prostick & EPYX joysticks, Speech Synthesizer, Navarone cart expander, 9 cartridges, numerous books, magazines, software. All for $600 plus UPS shipping. Phone 403-276-4716. v12n6

### TIPS

It took over 340 hours but now all TIPS files have been categorized, over 7,000 pictures by subject. You can NOW locate anything in seconds. Set consists of 51-S/S or 27-D/S brand new disks, in sleeves, labeled. Printable index is included. Get a set so your members may copy and reimburse your cost. Check or M.O. for $50.00 includes copying and shipping. Raymond Frantz, 502 No. 51st St. #2, Phoenix, AZ 85008-6649. (602) 275-4813. v12n6

# The ONLY monthly devoted to the TI99/4A

## Subscription Fees

❑ 12 issues, USA, **$35**

❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50**

❑ 12 issues, other countries surface mail, **$40.00**

❑ 12 issues, other countries, air mail, **$52.00**

**Outside U.S., pay via postal or international money order or credit card; personal checks from non-U.S. banks will be returned.**

## Address Changes

Subscribers who move may have the delivery of their most recent issue(s) delayed unless MICROpendium is notified six weeks in advance of address changes. Please include your old address as it appears on your mailing label when making an address change.

Check each item ordered (or list on separate page) and enter total amount here:_____

Check/MO ❑ MasterCard ❑ VISA ❑ (check one)

Card No. _____

Expiration Date _____

**(Minimum credit card order is $9)**

Signature _____

**(Required on credit card orders.)**

**No sales tax on magazine subscriptions. Texas residents add 7.75% sales tax on other items, including back issues and disk subscriptions. Credit card orders add 5%.**

**Mail to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680**

**Name** _____

**Address** _____

**City** _____

**State** _____ **ZIP** _____

**The set of numbers at the top of your mailing label indicates the cover date of the last issue of your subscription.**

## Disks, Etc.

❑ **Back Issues**, $3.50 each. List issues:_____

_____

No price breaks on sets of back issues. Free shipping USA. Add 30 cents, single issues to Canada/Mexico. Other foreign shipping 50 cents single issue surface, $1.50 airmail. Write for foreign shipping on multiple copies.

**OUT OF STOCK: Vols. 1, No. 1-2; Vol. 2, No. 1**

❑ **MICROpendium Index** (2 SSSD disks, 1984-1992), Extended BASIC required .....$6.00

❑ **MICROpendium Index II** (9 SSSD disks — 1 for each year — 1984-1992), XB required .....$30.00

❑ **MICROpendium Index II with MICROdex 99** (11 SSSD disks), XB required.....$35.00

❑ **MICROdex 99** (for use with MP Index II, 2 SSSD disks), XB required .....$10.00

❑ **MICROpendium Index II** annual disks ordered separately (1 disk per year, 1984-1992); each.....$6.00

**MICROdex 99, by Bill Gaskill, is a collection of programs that allow users of MP Index II to modify their index entries, as well as add entries. MICROdex 99 supports many other functions, including file merging, deletion of purged records, record counting and file browsing.**

### GENEVE DISKS (SSSD unless specified)

❑ **MDOS 2.21** (req. DSSD or larger (for floppy & hard drive systems) .....$4.00

❑ **GPL 1.5** .....$4.00

❑ **Myarc Disk Manager 1.50**.....$4.00

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** .....$4.00

❑ **Menu 80** (specify floppy or hard disk versions(s); includes SETCOLR, SHOWCOLOR, FIND, XUTILS, REMIND .....$4.00

### GENEVE PUBLIC DOMAIN DISKS

These disks consists of public domain programs available from bulletin boards. If ordering DSDD, specify whether Myarc or CorComp.

| | SSSD | DSSD | DSDD |
|---|---|---|---|
| ❑ **Series 1** | $9.00 | $7.00 | $5.00 |
| ❑ **Series 2** | $9.00 | $7.00 | $5.00 |
| ❑ **Series 3** | $9.00 | $7.00 | $5.00 |
| ❑ **Series 4** | $9.00 | $7.00 | $5.00 |
| ❑ **Series 5** | $9.00 | $7.00 | $5.00 |
| ❑ **Series 6** | $9.00 | $7.00 | $5.00 |

**SECOND CLASS**

A T ENTIRE 176
CHARLES BOOD
P.O. BOX 947
VENETIA PA 15367