Covering the TI99/4A and Geneve home computers

MICROpendium

Volume 15 Number 4 July/August 1998 $6

MUG '98

Seminars highlight event
Thanks for the ride, Lima

Programming

Make1From2 in XBASIC
Floating Points in Assembly

Reviews

Schnoz-ola
Freeware Games

V9T9

Downloading files & disks

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published bimonthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Periodical 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.

Foreign subscriptions are $42.50 (Canada and Mexico); $40 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

Internet E-mail: jkoloen@earthlink.net

Home page: http://www.earthlink.net/~jkoloen/

John Koloen Publisher

Laura Burns Editor

THE CHICAGO TI USERS GROUP

presents the

16th Annual Chicago TI International World Faire

supporting TI 994A and 9640

November 14, 1998

NEW LOCATION

NEW LOCATION

AMERICAN LEGION POST 42
Evanston, Illinois
(1033 Central Street)
9:30 a.m. - 4:00 p.m.
Vendors-demonstrations-Seminars-User Groups
Door Prize Drawings, Free Parking,

Bus and Rapid Transportation to Door
for further information contact: **Hal Shanafield** (847)864-8644
Hotel Information Available

All attendees to the Faire will receive a disk of brand new software. This disk will be available only to Faire attendees and is not copyable.

COMMENTS

Peterson recipients named

One of the highlights of the Multi-User Group conference in Lima in May was the announcement of this year's Jim Peterson Award recipients. All four recipients are well-chosen. Winners were:

Charles Good — Community

Mike Wright for PC99 — Software

Michael Becker for HSGPL and 80-column cards — Hardware

Tim Tesch — Geneve

MICROpendium ceases credit card acceptance

Starting Sept. 1, we will no longer accept credit card charges. I realize this may be an inconvenience to some readers, but we can no longer afford to absorb the monthly fees tacked on by our bank and Visa/Mastercard. We already assess a five percent surcharge on credit card purchases, but that is not nearly enough to cover our actual costs. This is due entirely to the low volume of credit card purchases. Were we to recover our actual costs, we'd need to add approximately $6.50 to each purchase, in addition to the five percent surcharge that we already charge.

This change will have its biggest effect on international subscribers who are accustomed to using credit cards. We regret having to do this, but we simply can't afford to underwrite credit card purchases any longer. We recommend the use of international money orders drawn on a U.S. bank. We appreciate your understanding.

—JK

FEEDBACK

A great computer

I seldom use the TI99/4A any longer, but I still enjoy the magazine you have so faithfully published. I have all but the first 10 issues, and check back through some of them ever so often. The new size presents a problem in storing them as an annual unit, though. (I'm not complaining, just challenged.)

I went to an IBM clone when I had trouble getting my Myarc card repaired, and have gradually slipped further away. I still think it's a great computer, though, and wish I had more time to spend with it.

Gordon H. McCaa
Lugoff, South Carolina

Calendar problems

Since I wrote you last year I have found a supplier of MFM hard drives right here in town. The drives are 20 to 40 meg. and from the late '80s IBM clones. They work well but I still have a problem of compatibility where two or more HD are connected. They run

FEEDBACK

OK for maybe a few weeks and then data is corrupted. We have an AGM in Derby once again on May 30 and Ross is going to change one of the chips on my Myarc HFDC and we hope that will do the trick. I've almost given up with my modem but hope that will be sorted as well.

In December of last year I began producing my calendars with Calendar Maker 99 as I had done every year since 1992 when I purchased it direct from Asgard; up until then I had no problems. At first I decided to make a five-year calendar and I needed about six for a few members of our family. The first problem began on printing 1998, the program would break at the end of February and not continue, so 1998 months had to be done separately and with two months to a page had to be lined up spot on to every page, which took some doing, but the worst was to come. The year 2000 had no provision for a leap year. I had to get in touch with the Greenwich Laboratory in London for confirmation of a leap year. So with only 28 days in February 2000 that puts all other years *one* day out of sync each month.

I have just written to Harry Brashear of Asgard in Newfane, New York, to see if he can put me on the right track as to a modification to the program. It does seem to spoil a wonderful application. I have just acquired a Panasonic 24-pin color KX-P2135 printer, I think it's a 1995 model, with no manual; could any of your readers supply me with a copy? How would the programs be altered to take advantage of the four-color ribbon which oscillates inside the printer?

David H. Caine
Crewe, England

Harry Brashear no longer runs Asgard. You might write Ramcharged Computers, 6747 E. Yancey Dr., Brook Park, OH 44142. — Ed.

Likes format

I like the new format in MICROpendium. The print is much easier to read and not as hard on the eyes. The size is much better to fit in your back pocket or in my lunch box. This is a lot of work for you and I just wanted to say thank you.

James Brown
Greendale, Indiana

WNY 99ers still going

I host three meetings each month at my home (second and third Tuesdays, and fourth Wednesdays). The first meeting is for the eight Geneve Users in our group. We usually have about six or seven at those meetings. The second is the Western New York 99ers User Group meeting. We get 12-16 at those. The last is our "99ers SIG" night, with about eight or so attending

I also continue to run a TI bulletin board, "The AM-CAN Friends BBS" at (716) 835-5316. This BBS, running on a TI99/4A, has been running 24 hours every day for many years, and I hope will continue for many more. It uses Tim Tesch's S&T software.

James P. Cavanaugh
Eggertsville, New York

MUG REPORT

Seminars are big hit at 1998 MUG conference

BY CHARLES GOOD

The following report was taken from the MUG conference Web site.—Ed.

The two-day free event, from Friday afternoon May 15 and all day Saturday May 16, seemed to go very smoothly. All the vendors got as many tables as they wanted. All the equipment in the seminar room ran smoothly, and most of the demonstrations worked as planned. Everybody said they were happy with Saturday's on-site food service.

Sixty-one people signed in and I know of at least three folks who gave seminars but did not sign in. I estimate attendance at 70, a far cry from the 300+ that attended the Lima MUG conferences of 1989 and 1990. There were 28 people at the after-the-conference get-together at Lima's best hamburger palace, the Kewpee. People came from 13 states and Canada. One individual from the New England area arrived after a 21-hour Greyhound bus trip, stayed seven hours and then went back to the Lima bus station for the 21-hour trip back. What dedication!

The conference had a "free stuff" table where people could recycle their unneeded hardware and software. Lots of stuff changed hands, including lots of disks, piles of old MICROpendium and Home Computer magazines, several boxed consoles, 3 full Peripheral Expansion Boxes, and several monochrome monitors usable on both 40- and 80-column systems.

John Parkens of Columbus arrived with a free system that included two horizon RAMdisks, which was quickly taken. The happy new owner of this system told me that he was an ex-TIer and would now definitely be getting back into our community.

Bob Carmany discusses upgrading of the TI99/4A.

At the end of the day we had to discard an empty PEB and some console power supplies, which nobody wanted. The fact is that there isn't much market value for routine 99/4A hardware anymore. Even non-routine used hardware seems inexpensive. For example, at one table a complete Mechatronics 80-column peripheral with console was offered for $85.

MUG REPORT

I gave the only Friday seminar, showing off two "old" TI cartridge games that had never been shown to the public. Included was an Atari game called Super Storm and a Funware game called Snoz-ola. These have almost literally been rescued from the dumpster by Competition Computer, which is now selling them to the TI community. Super Storm is the same as the Atari 2600 game called Slime. Super Storm was advertised in 1983 and 1984 for the TI in some computer magazines of that time, but my demonstration was the first time TIers have actually seen the game. It was obvious that I am not a very good arcade game player. There were people in the audience who played with the demonstration system and were soon making more points than I.

Seminars cover many topics

Saturday morning bright and early Dan Eicher gave the first seminar, showing off his complete Tomy Tutor Computer setup. This computer was sold at the same time as the 99/4A and was based on a 9995 CPU. Its BASIC is similar to 99/4A BASIC but much faster. No disk mass storage is available, only cassette. Several game cartridges were shown, all of which were converted to Geneve MDOS format several years ago by Barry Boone. Dan also showed a "Control Data Corporation 99/4A" computer. This is just like the TI version but with a different color bar powerup screen. The screen showed a copyright of 1983. Dan also passed around a copy of the TI FAQ which he will be posting on the Internet. This document answers lost of questions about routine use of the 99/4A.

Charles Good receives Peterson Award from Glen Bernasek.

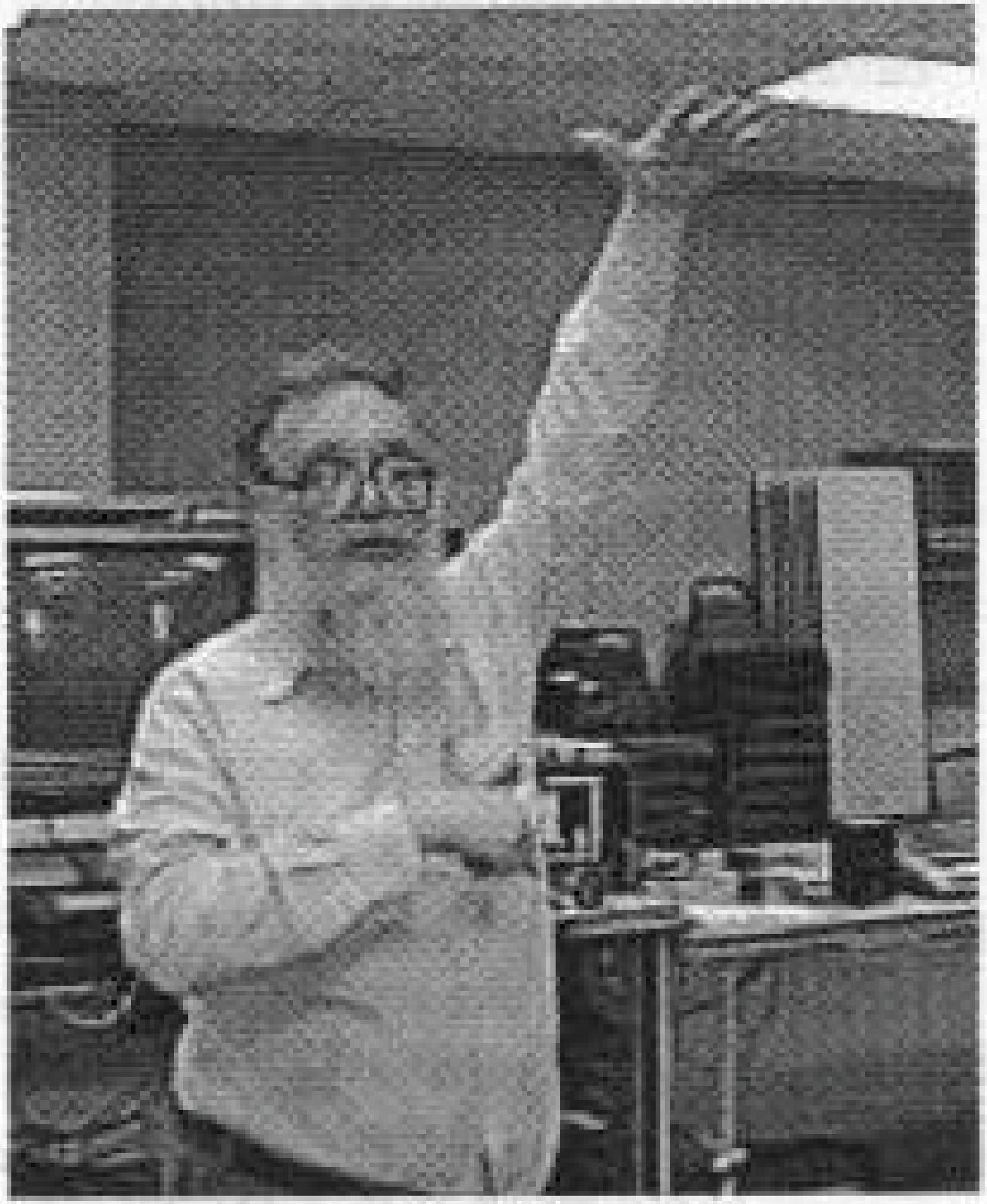
Bob Carmany's seminar was next. Bob provided very detailed information about upgrading 99/4A systems, including which devices use which CRU addresses, which devices conflict with each other, how to set up two RS232 cards, and various combinations of RAMdisks, and how to connect two PE boxes together so that one console can control two PE boxes full of cards. Lots of technical information was provided. I hope some of those present took good notes.

Bruce Harrison then demonstrated all the improvements he has made to Midi-Master software. He had a member of the audience use a Casio keyboard to play a short piece which was stored in the 99/4A's (with Asgard Memory System) memory. The computer than played back the piece and saved it to disk. On

Continued on page 8

Charles Good receives Peterson Award from Glen Bernasek.

Bob Carmany discusses upgrading of the TI99/4A.



Bruce Harrison showing how he has improved Midi Master software for the 99/4A.

Lory Werths, Marcel Barbeau, and Jean-Guy Barbeau are accompanied by a TI99/4A with Midi-Master and a Casio synthesizer.

MUG REPORT

Continued from page 7

rebooting, the piece was loaded off disk and played back again. Bruce's new Midi-Master software will automatically detect and use AMS memory. Long musical pieces can be stored in memory with an AMS card.

Lory Werths, Marcel Barbeau, and Jean-Guy Barbeau are accompanied by a TI99/4A with Midi-Master and a Casio synthesizer.

After Bruce's seminar the MUG Conference was treated to a concert of Renaissance and Celtic music featuring live musicians accompanied by the TI. I have never seen anything like this at any of the many TI fairs I have attended over the years. We were handed printed programs and treated to a series of Celtic, English, and French court and country dances played by Lory Werths, Marcel Barbeau, and Jean-Guy Barbeau. Lory is Bruce's partner and the Barbeaus are her children. They played the recorder, fiddle, bodhran (a traditional Irish drum held by one hand and struck with a short two-headed stick held in the middle with the other hand), and mandolin. All musical selections were accompanied by a Casio keyboard being played by a 99/4A and Midi-Master. The midi music was created by Lory. This very unusual seminar was a thoroughly enjoyable experience.

The next seminar was by Tim Tesch, who gave details of the latest Geneve version 6.0 MDOS. The new features I remember are the ability to use external SCSI Zip drives and the ability to correctly deal with dates in the year 2000 and beyond. Tim also discussed the status of Myarc repairs.

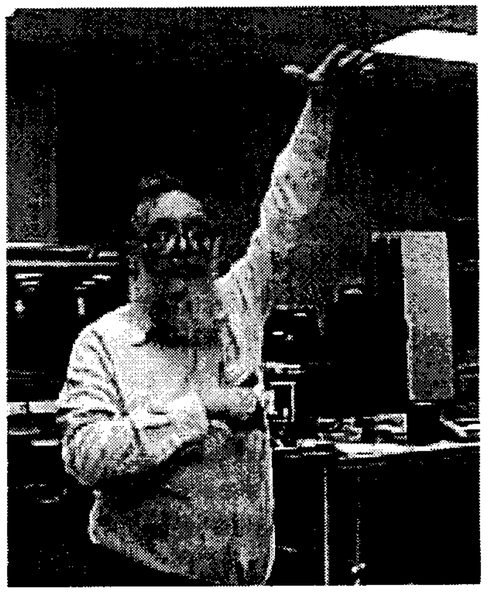
Mike Wright was next. He talked about the next release of PC99 and asked for feedback from the TI community. He said that further development has been delayed because one member of the PC99 development team has been working lots of hours at his "day" job and has had little time to work on PC99.

As I understood his talk, Mike was offering the TI community two alternatives:

One: Release an updated Version 5 now, which includes the following features not found in version 4 — Myarc 512K RAMdisk emulation with Myarc Extended BASIC, 1 megabyte AMS card emulation, an emulated clock card, SOB operating system emulation, and Super Space

MUG REPORT

bank-switching emulation. If this is done, this would probably be the last official version and the source code might be released for anyone to enhance.



Bruce Harrison showing how he has improved Midi Master software for the 99/4A.

Two: Wait on the next release until the PC99 team adds additional features including, hopefully, 9938/9958 VDP (ie. 80-column) emulation. In other words, do PC99 users want the above list of new features now or are they willing to wait, maybe a long wait, for more new features?

As part of the PC99 seminar I showed how easy it is to transfer a TI disk from a 99/4A system to PC99 running on my IBM laptop.

In the next seminar Lew King showed how to access the Internet using his 99/4A. He used Term 80 to dial in to his Internet service provider in Pennsylvania. He read an e-mail message, sent an e-mail message and then, most amazingly, brought up the MUG Conference web page. The web page was nicely formatted but without any graphics. Lew was using a version of the Lynx browser that was resident in the computer of his service provider. The screen display of Term 80 in 80 columns was barely readable. Lew says you can also use Telco to access the Internet in 40 columns. The text is more readable but the screen display is likely be be rather jumbled.

One of the unusual aspects of Lew's seminar was his use of a VGA monitor to display 99/4A video output. Lew connected the video cable of his 99/4A to a device and a cable from this device ran to the VGA monitor. The results were very sharp. Term 80 was easier to read on the VGA monitor than it was on the composite color monitors available in the seminar room. Lew told me that he has tried this VGA device with a Geneve and the results are not good, with lots of color bleeding when the Geneve outputs in composite color.

By e-mail Lew has provided the following information about the composite-color-to-VGA interface device: "This device was made by Proview Technology Inc., 12272 Monarch Street, Garden Grove, CA. 92841; 714-379-4455. Purchased from Tiger Direct 1-800-294-3269 for $120. It will input composite video and output VGA. It will also do the

Continued on page 10

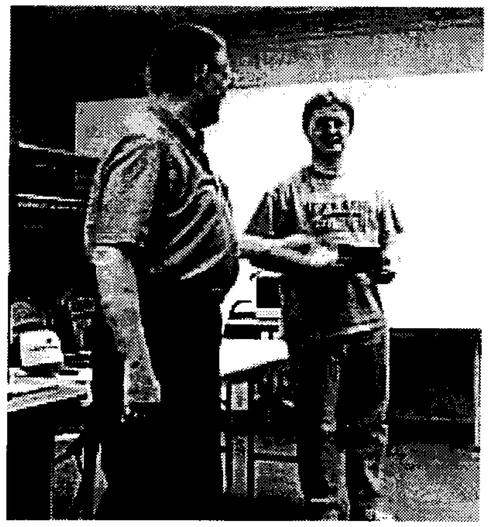
MUG REPORT

Continued from page 9

opposite and input VGA and output composite to a VCR, etc. The audio is stereo input and output, cable-ready with 181 TV channel tuner built in. The TV picture quality on a VGA monitor is excellent. There is a remote control included with on-screen display for contrast, brightness, saturation, hue, volume, and TV channel. Audio and video cables and everything else needed is also included. Input quality Y:U:V 4:2:2 Output quality R:G:B 8:8:8, 24-bit true color."

Ted Zychowicz had a good followup to Lew's seminar. Ted showed how to directly transfer files from an IBM to his Geneve using PORT software on the Geneve.

The final seminar was by Bud Mills. He said that he recognizes that the TI community isn't very big any more but he thinks it is large enough to support more Horizon RAMdisks. He said that he sold an 8-megabyte Horizon earlier in the day and hopes to pay for another limited production run of Horizon boards soon. Bud also stated that he has some PGRAM cards available for sale.



Tim Tesch receives Peterson award from Glen Bernasek.

PETERSON AWARD WINNERS

The conference ended with Glen Bernasek of the TI Chips user group awarding the 1998 Jim Peterson Achievement awards. Recipients were:

Community — Charles Good

Software — Mike Wright for PC99

Hardware — Michael Becker for his HSGPL card and 80-column card

Software — Tim Tesch

Michael Becker was the only recipient not at the conference and he later sent his thanks in an e-mail message as follows: "Thank's to all the people, who voted for me! I am very happy and very glad to get the award, for all the hard work we made all the years for the TI-community! Do not forget the other members of our German community, who made so good work to complete my cards. There are: Juergen Stelter, who made the wonderful layout of all SNUG cards; Harald Glaab, who wrote all the powerful programs (the DSR for EVPC, DSR for HSGPL, the DSR-loader for HSGPL and ASCSI, the EVPC-configuration-program the HSGPL-configurator/loader/saver... — it's too powerful for a single name — and our friends Wolfgang Bertsch and Oliver Arnold, who work on the SCSI-project for WHT and SNUG-card-users."

MUG REPORT

A welcome sight at the vendor area was Larry Conner of L.L. Conner Enterprise. He hasn't been to a TI show in several years.

Other vendors and groups with tables included Lee Bendick (hardware for sale), Tony Knerr (software giveaways), Ramcharged Computers (Asgard and other software), Bud Mills Services, CADD electronics (PC99), Cleveland area user groups, Dave Connery (hardware), Milwaukee area TI User Group, Harrison Software, S&T Software (Tim Tesch), The Fort's User Group of Fort Wayne, Indiana, and HUGGERS User Group of Indianapolis, Indiana. The only vendor who was scheduled but did not show up was Don Walden of Cecure Electronics.

Seminar video tapes available for $10

Video tapes of all conference seminars can be obtained by sending $10 to Charles Good, P.O. Box 647 Venedocia OH 45894 USA. Your money pays for two video tapes with about eight hours of video time on them and postage.

Seminar Speakers

Charles Good "Some never-before-seen 99/4A game cartridges available from Competition Computer"
Dan Eicher "The Tomy Tutor computer, and the Control Data Corporation 99/4A Computer"
Bob Carmany "Upgrading the TI"
Bruce Harrison "Midi Master upgrade"
Dolores P. Werths: "Midi music concert."
Tim Tesch "MDOS, Myarc repair/upgrade status, and future programs"
Mike Wright "PC99 stage 5"
Lew King "Accessing the Internet with a 99/4A using Term 80"
Ted Zychowicz "How to transfer files from the TI to a PC without a modem"
Bud Mills "Products of Bud Mills Services"

Exhibitors

Lee Bendick. Hardware for sale, 1 table
Tony Knerr. Software giveaways. 1 table
L.L. Conner Enterprise. Hardware and software, 4 tables
Ramcharged Computers. Asgard and other software, 4 tables.
Bud Mills Services, 1 table
CADD Electronics (PC99), 1 table
Tim Tesch, 1 table
Cleveland area user groups, 2 tables
Dave Connery, hardware, 3 tables
Milwaukee Area TI User group, 1 table
Harrison Software, 2 tables
S&T Software (Tim Tesch), 1 table
The Fort's User Group, Fort Wayne Indiana
HUGGERS User Group, Indianapolis, Indiana, 2 tables

Tim Tesch receives Peterson award from Glen Bernasek.

MUG REPORT

Thanks for the ride, Lima

BY GLENN BERNASEK

Bernasek is secretary of TI-CHIPS of Cleveland, Ohio.—Ed.

The Multi-User Group Conference (MUG '98) held in Lima, Ohio, in May was the final conference the Lima User Group will host. Charles Good explained that the Lima User Group is no longer considered an Ohio University student activity. Therefore, Reed Hall would not be available for future TI99/4A/Myarc 9640 user group conferences. Besides, Charles said, it's about time that he became a conference visitor rather than an operator. (It must be noted that this year's conference was organized and hosted by two members of what remains of the Lima Users Group.)

As I signed in at the conference on Saturday, I couldn't help but notice the few people who were walking around the main conference room. The usual vendors and user groups were there, but the traffic around the tables was very light. This confused me until I looked into the seminar room. Each seminar had full audience. This accounted for the apparent low attendance in the main room. (More than 70 conference attendees had signed in by early afternoon.)

The campus food service was open, and provided the attendees with a varied and delicious menu. This service was arranged for us by Charles Good. It was excellent and much appreciated.

Other than the Multi-User Group meeting, I was able to attend two of the seminars. Lew King, of the West Penn 99ers, demonstrated how Term-80 enables TIers to surf the net. Lew attempted to dial up the local Internet service provider from the phone without success. However, he was able to quickly get on-line through dialing commands issued by the keyboard.

Once on-line, Term-80 is a powerhouse of Internet communication software. It was noticed how quickly the Web pages, in a readable 80-column mode, came up on the screen with a 9,600-baud modem. This was in part due to the text-only connection ability of the TI99/4A. Therefore, there were no graphics to slow things down. Lew posted a message on the TI list server from the MUG during this seminar using the TI99/4A and Term-80. (It was noted that Vonn Malcuit, of TI-Chips, had volunteered to serve as interim videographer for some of the seminars.)

The other seminar I was able to attend was Ted Zychowicz's presentation on how to transfer files back and forth between a Myarc 9640 and a MS-DOS clone. All Ted did was to hook up a standard 25-pin RS-232 cable from the RS-232 port of the 9640 to the COM port on the back of the clone. Ted explained that this was made possible by the non-standard pin configurations of the TI's RS-232 card. (The TI RS-232 send pin is the MS-DOS COM port receive pin. Therefore a null modem cable between the two systems is not needed.)

Ted used Tim Tesch's PORT for file communication on the 9640 and MS-DOS/Windows communication software such as Microsoft's Exchange on the clone. Ted said the transfers must be made in ASCII text mode and the computers must be set at matching baud rates — 19,200 baud in this demonstration. All the user has to do is initiate the receive command before the send command and the file is quickly and effortlessly transferred. Thanks to Ted's demonstration, we now know that there is life left in those old TI files of ours.

MUG REPORT

As usual, the Multi-User Group meeting was very informative. The meeting was attended by representatives of the Cleveland Area User Groups (TI-Chips), the HUGGERS, the Milwaukee user group, the Chicago user group and the K-Town user group. The HUGGERS have their own BBS, and have received calls from notables such as Michael Becker. The Milwaukee group is in the planning phase of setting up its own BBS.

When asked if any groups have adopted multi-platform (computers other than TI99/4A or Myarc 9640) bylaws or agenda, the Milwaukee user group reported that it has established an MS-DOS special interest group (SIG) that meets on a different day from the regular TI user group meeting day. This avoids meeting and agenda conflicts for time and scheduling. It maintains a pure TI meeting atmosphere at the regular Milwaukee user group meetings.

I explained that TI-Chips had adopted a flexible meeting agenda for both TI and non-TI systems, and that we haven't experienced any real conflicts with this arrangement and have a much more open meeting atmosphere as the result. Comments were made by those in attendance that user groups, by nature, are still problem solvers and communication vehicles regardless of computer system discussed.

The Chicago user group stated that its membership had increased with four new members since the Fest West in Lubbock, Texas. The group said that there wasn't any membership drive. The group just seized the opportunity to sign up new members as the opportunity came along. The Chicago group also said that it serves as a user group information/referral resource in that it answers national inquires as to the location of local user groups for the TI99/4A and the Myarc 9640.

The meeting was closed with an interesting question — "I wonder where Mike Maksimik is?" The user group representatives at the meeting thought it would be interesting to find out where the former illustrious names in the TI community were and what they were doing. Maybe somebody could post a list of well-known TIers in MICROpendium, on the list server and comp.sys.ti with the question, "Where are they now?"

At the conclusion of the Jim Peterson Awards, it was announced that the 1999 MUG will be hosted by TI-Chips in the Cleveland, Ohio, area.

The MUG closed with a round of applause for the Lima User Group for its years of hosting this conference for TIers everywhere. Thanks for the ride, Lima. It was great!

Floating Points

By BRUCE HARRISON

Last month we led off with a bathtub, and this month we're floating, but not on the water in a bathtub. Back in October of 1995, we received a letter from Mr. Greg Knightes, of Coral Springs, Florida. He had noticed that the subject of floating point math operations had been sorely lacking in our columns. A quick check showed that he was right. While we'd mentioned the subject now and then, there was no full discussion about using the floating point operations in any of our columns. As is our usual practice, we answered Mr. Knightes' letter in a couple of days, and included a disk for him with annotated source code to illustrate the use of floating point math. That "demo" program forms the heart of today's sidebar.

Getting the numbers in

The first thing that we had to address is how to input floating point numbers through assembly routines. There are probably many ways to do this, but the easiest way is to use a fairly simple method that's provided by an internal ROM routine available to us through XMLLNK. That routine, which we call CSN, for Convert String to Number, is very powerful. To use the routine, we first employ any of our "string input" routines, such as the CRSIN routine that we developed long ago, or the ACCEPT routine that we showed in Part 53 of this series. This simply allows the user to type on the screen in a manner akin to an ACCEPT AT operation in Extended BASIC. Our ACCEPT routine includes insert and delete character capabilities, field clearing with Function-3, and so on. The key to using it for floating point numbers is that, when it exits, Register 0 of our workspace points to the VDP address of the first character the user enters. If what's typed there is a number, we can take that number by simply putting the address from R0 into the RAM Pad at >8356, then invoking the CSN routine through XMLLNK.

The CSN routine reads the number from the screen image in VDP RAM, and creates from that a floating point number in the eight-byte area starting at >834A. That area of RAM Pad is called the Floating Point Accumulator, or FAC, for short.

The rules of input

The number must start at the beginning of the input field with either a numeric character (0-9), a minus sign, a decimal point, or a plus sign. If none of those is found at the beginning of the field, the result reported to FAC will be zero.

To put that another way, the numeric entry must be left-justified in the entry field. If the field is blank or has leading spaces, the result will be zero at FAC. Operators plus or minus will be accepted as the first character, but not later in the entry field, except as part of an exponent. The multiply, raise to and divide operators will not be accepted anywhere in the entry. In other words, the entry must be purely numbers. The one exception to this rule is the E for exponent operation, as in scientific number notation. If, for example, the field contains 2E3, the E3 will

THE ART OF ASSEMBLY PART 71

be correctly interpreted as meaning that the two gets multiplied by 10 raised to the third power, so the number reported to FAC will be 2,000 in floating point notation. The E must be uppercase, and may be followed by a + or - to indicate the sign of the exponent. In the version of ACCEPT shown in this issue's sidebar, we've included code that will make any alpha character uppercase, so your E for exponent may be typed as lowercase, but will appear in uppercase on the screen.

Differences from XB input

If we use the BASIC or Extended BASIC INPUT routine for a numeric value, as in INPUT N, the operation is different from what happens in our assembly case. To start with, our number may be preceded by leading spaces in BASIC or XB, but the number will still be recognized and reported correctly to the numeric variable in floating point notation. If the input field in BASIC or XB is left blank, a WARNING will be issued, and the value of the variable will not be affected. In our assembly case, a blank field will simply be accepted as zero. Of course, you could put in some assembly code to "strip off" any leading spaces in your input field before using the CSN routine, and thus make your input behave like the BASIC INPUT in that respect. The floating point number generated by CSN will be correct to 14 significant digits, with the last digit rounded if need be. The number is placed at FAC as eight bytes in radix 100 notation. The first of those eight bytes is the power of 100 by which the remaining seven bytes are multiplied. Each of the remaining seven bytes is equal to two significant digits, ranging from 0 through 99 (decimal) in value. The power of 100 in the first byte is offset by >40, so that both positive and negative powers of 100 can be handled. In other words, the first byte being >40 means the number is multiplied by 100 to the 0th power, >41 means the number is multiplied by 100 to the first power, >3F means the number is multiplied by 100 to the -1 power, etc. This way, the powers of 100 can range from ->40 through >7F. The most significant bit (>80) in this first byte is used for the sign of the number, so if the MSB is set, the number itself is negative.

This way of doing floating point numbers is radically different from the way floating point numbers are handled by most other computers. On PC computers, for example, the floating point numbers are handled in only four bytes in binary notation. The method used in the TI yields much more accurate numbers than the PC method. Of course it takes twice as many bytes to store a number, but having 14-digit accuracy can come in handy. Your TI does more accurate math than the PC!

Today's sidebar

It's very long, but is a complete program that demonstrates the use of floating point numbers on your TI. The program accepts two numbers from the user, stashes them away in memory, then performs various math operations on them. Before

Continued on page 16

Continued from page 15

each operation, it copies the numbers back into the RAM Pad memory at locations called FAC and ARG. FAC is the eight bytes starting at >834A, and ARG is the eight bytes starting at >835C. After the numbers are in these two places, we can perform a wide variety of operations on them. For example, we can use XMLLNK to add, subtract, multiply, divide or compare the numbers. For the four main operations, the result of the operation is a number at FAC, and the number at ARG is meaningless after the operation. For the compare operation, both FAC and ARG numbers remain intact after the compare, and the result is indicated by the state of the GPL Status byte at >837C. Please note that the computer's status register is not set by this compare, so we have to examine the byte at >837C to figure out the result of the comparison. If, for example, the numbers are equal, the byte at >837C will equal >20. If they're not equal, then we have to isolate the bits of that byte to determine whether the number at ARG is greater than or less than the number at FAC.

In our example, we've put the status byte into R3, then masked R3 with >4,000, which leaves just the "greater than" bit in R3. If the "greater than" bit was zero, then the number at ARG was less than the number at FAC.

Nasty little details

If you examine the sidebar closely, you'll see some odd little things done, which we'd better explain now. Among the EQUates, you'll see one called VSTACK, set to >1,000. That's used in the code soon after label START to put the number >1,000 into the word at >836E. If we were doing only the math operations that use XMLLNK, we would not have to set a value in >836E. We set this number because later in our program, we perform a SIN function using GPLLNK. That function, and presumably others that are used through GPLLNK, uses a "value stack" in VDP RAM to do its calculations. If we don't initialize the Value Stack Pointer at >836E, the SIN operation will mess up part of our display screen by writing stuff into the screen image portion of VDP RAM.

Operation of the program

The sidebar is a complete program which illustrates many things. First, it prompts for and accepts two floating point numbers. These get placed into memory as eight bytes each at labels NUM1 and NUM2. We've made a little subroutine called MOVNUM to make it easier to move eight-byte numbers from one place to another. For example, when we wish to move NUM1 into the eight bytes at ARG, we simply LI R9,NUM1, LI R10,ARG, then BL @MOVNUM.

You'll notice that we do this each time before an operation, and that except for the first case, we have to put NUM2 into FAC through a MOVNUM operation. In that first case, we didn't have to move NUM2 into FAC because it's still there from the previous Convert String to Number operation.

For the add and multiply operations, it doesn't matter which number is at FAC

and which at ARG. For the subtract and divide operations, however, it's important to remember that the number at ARG gets the number at FAC subtracted from it, and the number at ARG gets divided by the number at FAC. Similarly, in the compare operation, the indication for greater than means that the number at ARG is greater than the number at FAC. In all floating point math operations except compare, the result of the operation is reported at FAC. Thus when we invoke our subroutine at DISNUM, the result number at FAC is what gets converted to a string for display. The subroutine DISNUM puts the number from FAC on the screen at whatever location was set in R0 before calling DISNUM. For positive numbers, the string displayed by DISNUM will have a leading space, while negative numbers will have a minus in that first string character.

We're going to stop at this point because the sidebar this issue is very long. Keep this issue handy, because next issue we're going to continue this discussion with more detailed examination of today's sidebar. See you then.

SIDEBAR17

```
0001  * SIDEBAR 71
0002  * A COMPLETE PROGRAM
0003  *
0004  * DEMO OF FLOATING POINT OPERATIONS
0005  * PUBLIC DOMAIN
0006  * 10/31/95
0007  * BY Bruce Harrison
0008  *
0009         REF  VSBW,VSBR,VMBW,VMBR,KSCAN,XMLLNK  REF UTILS
0010         DEF  START         DEFINE ENTRY
0011  *
0012  * REQUIRED EQUATES
0013  *
0014  STATUS EQU  >837C         GPL STATUS BYTE
0015  KEYADR EQU  >8374         KEY-UNIT
0016  KEYVAL EQU  >8375         KEY VALUE
0017  FAC    EQU  >834A         F.P. ACCUMULATOR (8 BYTES)
0018  FAC11  EQU  >8355         F.P. ACCUM +11
0019  FAC12  EQU  >8356         F.P. ACCUM +12
0020  ARG    EQU  >835C         F.P. ARGUMENT (8 BYTES)
0021  CNS    EQU  >0014         CONVERT F.P. TO STRING W/GPLLNK
0022  CSN    EQU  >1000         CONV. STRING (IN VDP) TO F.P. W/XMLLNK
0023  FADD   EQU  >0600         ADD F.P.NUMBERS W/XMLLNK
0024  FSUB   EQU  >0700         SUBTRACT FAC FROM ARG F.P. W/XMLLNK
0025  FMUL   EQU  >0800         MULTIPLY F.P. NUMBERS W/XMLLNK
```

Continued on page 18

THE ART OF ASSEMBLY — PART 71

Continued from page 17

```
0026  FDIV   EQU  >0900        DIVIDE ARG BY FAC F.P. W/XMLLNK
0027  FCOM   EQU  >0A00        COMPARE ARG TO FAC F.P. W/XMLLNK
0028  SINE   EQU  >002E        SINE OF FAC F.P. W/GPLLNK
0029  VSTACK EQU  >1000        OUR VDP STACK
0030  GPLWS  EQU  >83E0        GPL WORKSPACE
0031  GR4    EQU  GPLWS+8      GPL REG 4
0032  GR6    EQU  GPLWS+12     GPL REG 6
0033  STKPNT EQU  >8373        STACK POINTER
0034  LDGADD EQU  >60          LOAD GPL ADDRESS
0035  XTAB27 EQU  >200E        XTABLE 27
0036  GETSTK EQU  >166C        GET STACK
0037  *
0038  * MAIN CODE SECTION
0039  *
0040  START  LWPI WS           LOAD OUR WORKSPACE
0041         CLR  @KEYADR      CLEAR KEY-UNIT
0042         LI   R0,VSTACK    VALUE STACK ADDRESS
0043         MOV  R0,@>836E    SET VALUE STACK POINTER
0044  RESTR  LI   R0,3         ROW 1, COL 4
0045         LI   R1,N1STR     'ENTER FIRST NUMBER'
0046         BL   @DISSTR      DISPLAY
0047         BL   @ACCEPT      USE ACCEPT SUBROUTINE
0048         DATA 32+2         SCREEN POSITION R2, C3
0049         DATA 28           FIELD LEN
0050         DATA 1            0 - DON'T, 1 - CLEAR FIELD
0051         DATA TEMSTR       STRING DESTINATION
0052         MOV  R0,@FAC12    VDP ADDRESS TO FAC12
0053         BLWP @XMLLNK      USE XML LINKAGE
0054         DATA CSN          CONVERT STRING FROM VDP TO NUMBER
0055         LI   R9,FAC       POINT AT FLOATING POINT ACCUMULATOR
0056         LI   R10,NUM1     MEM ADDRESS FOR 1ST NUMBER
0057         BL   @MOVNUM      PLACE THE NUMBER AT NUM1
0058         LI   R0,3*32+3    ROW 4, COL 4
0059         LI   R1,N2STR     'ENTER SECOND NUMBER'
0060         BL   @DISSTR      DISPLAY
0061         BL   @ACCEPT      USE ACCEPT SUBROUTINE
0062         DATA 4*32+2       SCREEN POSITION R5, C3
0063         DATA 28           FIELD LEN
0064         DATA 1            0 - DON'T, 1 - CLEAR FIELD
0065         DATA TEMSTR       STRING DESTINATION
0066         MOV  R0,@FAC12    VDP ADDRESS TO FAC12
0067         BLWP @XMLLNK      USE XML
0068         DATA CSN          CONVERT STRING TO F.P.
0069         LI   R9,FAC       POINT AT ACCUM.
```


THE ART OF ASSEMBLY — PART 71

```
0070         LI   R10,NUM2     MEM LOCATION FOR NUM2
0071         BL   @MOVNUM      PLACE THE NUMBER
0072         LI   R0,7*32+1    ROW 8, COL 2
0073         LI   R1,ADDSTR    PLUS
0074         BL   @DISSTR      DISPLAY
0075         LI   R9,NUM1      NUM1
0076         LI   R10,ARG      TO ARGUMENT
0077         BL   @MOVNUM      PLACE NUM1
0078         BLWP @XMLLNK      USE XML
0079         DATA FADD         ADD ARG TO FAC
0080         A    R2,R0        ADD LENGTH TO ADDRESS
0081         LI   R4,18        18 CHARS
0082         BL   @BLNFLD      CLEAR THE 18 CHARS
0083         BL   @DISNUM      DISPLAY THE NUMBER AT FAC
0084         LI   R9,NUM1      POINT AT NUM1
0085         LI   R10,ARG      AND ARG
0086         BL   @MOVNUM      PUT NUM1 AT ARG
0087         LI   R9,NUM2      POINT AT NUM2
0088         LI   R10,FAC      AND FAC
0089         BL   @MOVNUM      MOVE NUM2 TO FAC
0090         LI   R0,9*32+1    ROW 10, COL 2
0091         LI   R1,SUBSTR    SUBTRACT
0092         BL   @DISSTR      DISPLAY
0093         BLWP @XMLLNK      USE XML
0094         DATA FSUB         SUBTRACT FAC FROM ARG
0095         A    R2,R0        ADD LENGTH
0096         LI   R4,18        18 CHARS
0097         BL   @BLNFLD      BLANK FIELD
0098         BL   @DISNUM      DISPLAY THE NUMBER
0099  *
0100  * FOLLOWING REPEATS THE PROCESS FOR MULTIPLY AND DIVIDE
0101  *
0102         LI   R9,NUM1
0103         LI   R10,ARG
0104         BL   @MOVNUM
0105         LI   R9,NUM2
0106         LI   R10,FAC
0107         BL   @MOVNUM
0108         LI   R0,11*32+1
0109         LI   R1,MULSTR
0110         BL   @DISSTR
0111         BLWP @XMLLNK
0112         DATA FMUL
0113         A    R2,R0
```

Continued on page 20

Continued from page 19

```
0114          LI    R4,18
0115          BL    @BLNFLD
0116          BL    @DISNUM
0117          LI    R9,NUM1
0118          LI    R10,ARG
0119          BL    @MOVNUM
0120          LI    R9,NUM2
0121          LI    R10,FAC
0122          BL    @MOVNUM
0123          LI    R0,13*32+1
0124          LI    R1,DIVSTR
0125          BL    @DISSTR
0126          BLWP  @XMLLNK
0127          DATA  FDIV
0128          A     R2,R0
0129          LI    R4,18
0130          BL    @BLNFLD
0131          BL    @DISNUM
0132    *
0133    * FOLLOWING COMPUTES AND SHOWS SIN(NUM1)
0134    *
0135          LI    R9,NUM1       1ST NUMBER
0136          LI    R10,FAC       TO FAC
0137          BL    @MOVNUM       MOVE THAT
0138          BLWP  @GPLLNK       USE GPLLNK
0139          DATA  SINE          TO COMPUTE SIN(NUM1)
0140          LI    R0,15*32+1    ROW 16, COL 2
0141          LI    R1,SINSTR     "SIN OF NUM1"
0142          BL    @DISSTR       DISPLAY THAT
0143          A     R2,R0         MOVE POINTER
0144          LI    R4,18         18 CHARS
0145          BL    @BLNFLD       BLANK FIELD
0146          BL    @DISNUM       DISPLAY THE NUMBER
0147    *
0148    * FOLLOWING COMPARES NUM1 AND NUM2
0149    *
0150          LI    R9,NUM1       1ST NUMBER
0151          LI    R10,ARG       TO ARGUMENT
0152          BL    @MOVNUM       MOVE
0153          LI    R9,NUM2       2ND NUMBER
0154          LI    R10,FAC       TO FAC
0155          BL    @MOVNUM       MOVE
0156          BLWP  @XMLLNK       USE XML LINK
0157          DATA  FCOM          COMPARE F.P. NUMBERS
```



```
0158          CB    @STATUS,@ANYKEY IS STATUS BYTE = >20?
0159          JEQ   SEQ           THEN NUMBERS EQUAL
0160          MOVB  @STATUS,R3    MOV TO R3
0161          ANDI  R3,>4000      MASK TO > BIT
0162          JEQ   SLT           IF ZERO, JUMP
0163    SGR   LI    R1,GRTSTR     ELSE SET GREATER
0164          JMP   SHWCMP        THEN JUMP
0165    SEQ   LI    R1,EQUSTR     ARG = FAC
0166          JMP   SHWCMP        THEN JUMP
0167    SLT   LI    R1,LESSTR     NUM1 < NUM2
0168    SHWCMP LI   R0,17*32+1    ROW 18, COL2
0169          BL    @DISSTR       DISPLAY STRING
0170          LI    R0,19*32+5    ROW 20, COL 6
0171          LI    R1,PAK        "PRESS ANY KEY"
0172          BL    @DISSTR       DISPLAY THAT
0173          A     R2,R1         NEXT STRING
0174          LI    R0,21*32+4    ROW 22, COL 5
0175          BL    @DISSTR       DISPLAY "OR FUNCT-8"
0176          LI    R0,23*32+15   ROW 24, COL 16
0177          CLR   @>8378        CLEAR TIMER
0178          MOVB  @CURSOR,R1    CURSOR CHAR
0179          BLWP  @VSBW         ON SCREEN
0180          MOV   @INTLOC,@>83C4 ENABLE USER INTERRUPT
0181          MOVB  @ANYKEY,@ALTKEY ALTERNATE SPACE
0182          BL    @KEYLOO       USE KEY LOOP
0183          CLR   @>83C4        STOP USRINT
0184          MOVB  @ANYKEY,R1    SPACE IN R1
0185          BLWP  @VSBW         WRITE THAT
0186          CI    R8,6          WAS FUNCTION-8 STRUCK?
0187          JNE   EXIT          IF NOT, EXIT
0188          B     @RESTR        ELSE RE-START
0189    EXIT  LWPI  >83E0         GPL WORKSPACE
0190          B     @>6A          TO GPL INTERPRETER
0191    *
0192    * SUBROUTINES
0193    *
0194    ACCEPT MOV  *R11+,R0      R0 HAS START POSITION
0195          JNE   GETLEN        IF NOT 0, JUMP
0196          INC   R0            ELSE POINT AT 1
0197    GETLEN MOV  *R11+,R2      R2 HAS MAX LENGTH
0198          MOV   *R11+,R3      R3 HAS CLEAR FIELD SIGNAL
0199          MOV   *R11+,R9      R9 HAS STRING DESTINATION
0200          CLR   @INSFLG       NOT IN INSERT
0201          MOV   R0,R7         SAVE START POSITION
```

Continued on page 22

Continued from page 21

```
0202          MOV  R2,R4         SAVE LENGTH
0203          DEC  R0            POINT ONE BACK
0204          MOVB @EDGE,R1      EDGE CHARACTER
0205          BLWP @VSBW         WRITE A BYTE
0206          INC  R0            BACK TO START
0207          A    R2,R0         ADD LENGTH
0208          MOV  R0,R6         SAVE THAT POSITION
0209          DEC  R6            LAST ALLOWED
0210          BLWP @VSBW         WRITE EDGE CHAR
0211  CLRSNS  MOV  R7,R0         BACK TO START
0212          MOV  R3,R3         CHECK SIGNAL
0213          JEQ  KEYFRC        IF ZERO, JUMP
0214          MOV  R4,R2         GET LENGTH BACK IN R2
0215          MOVB @ANYKEY,R1    SPACE CHAR
0216  CLRFLD  BLWP @VSBW         WRITE ONE SPACE
0217          INC  R0            MOVE AHEAD ONE
0218        DEC  R2          DEC COUNT
0219         JNE  CLRFLD       IF NOT
0, RPT
0220          MOV  R7,R0         GET START BACK
0221  *
0222  * KEYFRC GETS THE CURRENT CHARACTER
0223  * FROM THE SCREEN, FORCES THE CURSOR
0224  * TO THAT POSITION, THEN ACTIVATES THE
0225  * USER INTERRUPT TO BLINK CURSOR
0226  *
0227  KEYFRC BLWP @VSBR         READ BYTE AT R0 POSITION
0228         MOVB R1,@ALTKEY    PLACE AT ALTKEY
0229         MOVB @CURSOR,R1    PUT CURSOR IN R1
0230         BLWP @VSBW         WRITE CURSOR
0231         CLR  @>8378        CLEAR TIME COUNTER
0232         MOV  @INTLOC,@>83C4 ENABLE USER INTERRUPT
0233  *
0234  * KEYIN IS THE PART THAT GETS KEYSTROKES
0235  *
0236  KEYIN  BLWP @KSCAN         SCAN KEYBOARD
0237         LIMI 2             ALLOW INTERRUPTS
0238         LIMI 0             STOP THEM
0239         CB   @STATUS,@ANYKEY KEY STRUCK?
0240         JNE  KEYIN         IF NOT, REPEAT
0241  *
0242  * FOLLOWING CODE USES THE KEYSTROKE
0243  *
0244         MOV  @KEYADR,R8    KEY AS WORD IN R8
```



```
0245          MOVB @ALTKEY,R1    OLD CHAR IN R1
0246          BLWP @VSBW         WRITE TO SCREEN
0247          CB   @KEYVAL,@ENTERV "ENTER" STRUCK?
0248          JEQ  KEYEX         IF YES, EXIT
0249          CB   @KEYVAL,@BACKUP FUNCTION-S?
0250          JNE  KEY0          IF NOT, JUMP
0251  *
0252  * FOLLOWING IS CODE THAT HANDLES FUNCTION-S
0253  * IT MOVES CURSOR ONE SPOT, THEN GOES TO
0254  * RPTKEY, WHICH DELAYS BEFORE ALLOWING REPEAT
0255  *
0256          DEC  R0            DEC SCRN POSITION
0257          BLWP @VSBR         READ BYTE
0258          CB   R1,@EDGE      EDGE CHARACTER?
0259          JNE  BCKX          IF NOT, JUMP
0260          INC  R0            ELSE INC POSITION
0261          JMP  KEYFRC        THEN BACK
0262  BCKX    B    @RPTKEY        AHEAD FOR REPEAT ACTION
0263  KEY0    CB   @KEYVAL,@FWARD FUNCTION-D?
0264          JNE  KEY1          IF NOT, JUMP AHEAD
0265  *
0266  * FOLLOWING IS CODE THAT HANDLES FUNCTION-D
0267  * IT MOVES CURSOR ONE SPOT, THEN GOES TO
0268  * RPTKEY, WHICH DELAYS BEFORE ALLOWING REPEAT
0269  *
0270          INC  R0            POINT AHEAD
0271          BLWP @VSBR         READ BYTE
0272          CB   R1,@EDGE      EDGE CHAR?
0273          JNE  FWKX          IF NOT, JUMP
0274          DEC  R0            ELSE POINT BACK
0275          B    @KEYFRC       THEN BRANCH BACK
0276  FWKX    JMP  RPTKEY        AHEAD FOR REPEAT ACTION
0277  KEY1    CI   R8,32         COMPARE TO SPACE BAR
0278          JLT  FUNCT         IF LESS, CHECK FOR FUNCT
0279          CI   R8,122        CHECK L.C. z
0280          JGT  CHKINS        IF GREATER, JUMP
0281          CI   R8,97         CHECK L.C. a
0282          JLT  CHKINS        IF LESS, JUMP
0283          SB   @ANYKEY,@KEYVAL ELSE CONVERT TO U.C.
0284  *
0285  * FOLLOWING HANDLES KEY VALUES 32 AND ABOVE
0286  *
0287  CHKINS MOV  @INSFLG,R1    INSERT MODE?
0288          JEQ  KEY1A         IF NOT, JUMP AHEAD
```

Continued on page 24

Continued from page 23

```
0289  *
0290  * FOLLOWING HANDLES INSERT IF IN INSERT MODE
0291  *
0292          C     R0,R6         AT END OF FIELD?
0293          JEQ   KEY1A         IF SO, NO INSERT
0294          MOV   R6,R2         GET LAST POSITION
0295          S     R0,R2         SUBTRACT CURRENT POSITION
0296          MOV   R9,R1         USE ASSIGNMENT SPACE
0297          BLWP  @VMBR         PUT BYTES THERE
0298          INC   R0            POINT AHEAD ONE
0299          BLWP  @VMBW         WRITE THERE
0300  DEC0    DEC   R0            BACK TO OLD POSITION
0301          JMP   KEY1A         PUT IN THE KEYSTROKE
0302  *
0303  * FOLLOWING HANDLES FUNCTION KEYS WITH VALUES BELOW 32
0304  *
0305  FUNCT   CB    @KEYVAL,@DELKEY DELETE KEY?
0306          JNE   FUNCT2        IF NOT, JUMP
0307  *
0308  * FOLLOWING HANDLES DELETE WITH FUNCTION-1
0309  *
0310          MOV   R0,R3         STASH AWAY R0
0311          MOV   R6,R2         GET END OF FIELD
0312          S     R0,R2         SUBTRACT CURRENT POSITION
0313          JEQ   NULDEL        IF ZERO, JUMP AHEAD
0314          INC   R0            ELSE POINT AHEAD ONE
0315          MOV   R9,R1         POINT AT ASSIGNMENT PLACE
0316          BLWP  @VMBR         READ TO THERE
0317          DEC   R0            POINT BACK ONE
0318          BLWP  @VMBW         WRITE TO THERE
0319  NULDEL  MOV   R6,R0         GET END OF FIELD
0320          MOVB  @ANYKEY,R1    SPACE CHAR
0321          BLWP  @VSBW         WRITE A SPACE
0322          MOV   R3,R0         GET OLD POSITION BACK
0323          JMP   KEYFRC        JUMP TO GET NEXT KEY
0324  FUNCT2  CB    @KEYVAL,@INSKEY FUNCT-2 PRESSED?
0325          JNE   FUNCT3        IF NOT, JUMP
0326  *
0327  * FOLLOWING SETS INSERT MODE ON FUNCTION-2
0328  *
0329          INC   @INSFLG       SET INSERT FLAG
0330          JMP   KEYFRC        THEN BACK
0331  FUNCT3  CB    @KEYVAL,@ERSKEY FUNCT-3 PRESSED?
0332          JNE   FUNCT9        IF NOT, JUMP
```



```
0333  *
0334  * FOLLOWING ERASES FIELD IF FUNCTION-3 STRUCK
0335  *
0336  ERSFLD MOVB @ANYKEY,R3    SET R3 NON-ZERO
0337          B     @CLRSNS     BRANCH TO CLEAR FIELD
0338  *
0339  * FUNCTION-9 EXITS FROM ROUTINE
0340  *
0341  FUNCT9 CI    R8,15         FUNCTION-9?
0342          JEQ   KEYEX         IF SO, EXIT ROUTINE
0343  *
0344  * FUNCTION-8 CAUSES ERASE OF FIELD
0345  *
0346          CI    R8,6          FUNCTION-8?
0347          JEQ   ERSFLD        IF SO, ERASE
0348          B     @KEYFRC       ELSE IGNORE KEYSTROKE
0349  *
0350  * FOLLOWING PUTS CURRENT KEYSTROKE ON SCREEN
0351  * THEN MOVES CURSOR TO NEXT SPOT
0352  *
0353  KEY1A   MOVB  @KEYVAL,R1    GET KEY VALUE IN R1
0354          BLWP  @VSBW         WRITE THAT
0355          INC   R0            POINT AHEAD
0356          BLWP  @VSBR         READ A BYTE
0357          CB    R1,@EDGE      EDGE?
0358          JNE   KEY1X         IF NOT, OKAY
0359          DEC   R0            POINT BACK
0360  KEY1X   B     @KEYFRC       THEN BRANCH BACK
0361  *
0362  * KEYEX IS THE EXIT FROM THIS ROUTINE
0363  *
0364  KEYEX   CLR   @>83C4        KILL USER INTERRUPT
0365          MOV   R4,R2         GET LENGTH
0366          MOV   R6,R0         AND LAST POSITION
0367  RDBYT   BLWP  @VSBR         READ A BYTE
0368          CB    R1,@ANYKEY    SPACE?
0369          JNE   RDSTR         IF NOT, JUMP
0370          DEC   R0            ELSE DEC POSITION
0371          DEC   R2            AND CHAR COUNT
0372          JNE   RDBYT         IF NOT ZERO, GO BACK
0373  RDSTR   MOV   R9,R1         GET STRING LOCATION
0374          MOV   R7,R0         AND START POSITION
0375          SWPB  R2            SWAP BYTES
0376          MOVB  R2,*R1+       PUT LENGTH BYTE AT STRING LOCATION
```

Continued on page 26

Continued from page 25

```
0377          JEQ  NULSTR       IF ZERO, JUMP
0378          SWPB R2           SWAP R2 AGAIN
0379          BLWP @VMBR        READ STRING CONTENT
0380  NULSTR  RT                RETURN TO CALLER
0381  *
0382  * UPON EXIT, THE ENTRY IS PLACED AS A STRING WHERE ASSIGNED,
0383  * AND REGISTER 8 HAS THE KEYSTROKE THAT CAUSED THE EXIT
0384  *
0385  *
0386  * FOLLOWING IS THE REPEAT-KEY FUNCTION FOR LEFT AND RIGHT
0387  * MOVEMENT OF THE CURSOR
0388  *
0389  RPTKEY BLWP @VSBR         READ CURRENT CHAR
0390          MOVB R1,@ALTKEY   PLACE AT ALTKEY
0391          MOVB @CURSOR,R1   GET CURSOR
0392          BLWP @VSBW        WRITE THAT
0393          CLR  @INSFLG      CLEAR INSERT MODE
0394          CLR  @>8378       CLEAR TIMER
0395          CLR  @>83C4       DISABLE USRINT
0396  *
0397  * THE LOOP STARTING AT RPT1 DELAYS REPEAT MOTION FOR
0398  * 32/60THS OF A SECOND UNLESS KEY IS RELEASED
0399  *
0400  RPT1    BLWP @KSCAN       SCAN KEYBOARD
0401          LIMI 2            ALLOW INTS
0402          LIMI 0            STOP INTS
0403          CB   @KEYVAL,@NOKEY NO KEY?
0404          JEQ  RPTEX        IF SO, EXIT
0405          CB   @>8379,@ANYKEY COMPARE TO 32
0406          JLT  RPT1         IF LESS, JUMP
0407  RPT1A   CLR  @>8378       CLEAR TIMER
0408          MOVB @ALTKEY,R1   GET ALTKEY BACK
0409          BLWP @VSBW        WRITE
0410          CB   @KEYVAL,@BACKUP BACKWARD?
0411          JNE  RPTF         IF NOT, JUMP
0412          DEC  R0           ELSE BACK ONE
0413          BLWP @VSBR        READ CHAR
0414          CB   R1,@EDGE     IS THAT EDGE CHAR?
0415          JNE  RPTF1        IF NOT, JUMP
0416          INC  R0           PUT POSITION BACK
0417          JMP  RPTEX        THEN EXIT
0418  RPTF    INC  R0           AHEAD ONE
0419  RPTF1   BLWP @VSBR        READ CHAR
0420          CB   R1,@EDGE     EDGE?
```



```
0421          JNE  RPTFA        IF NOT, JUMP
0422          DEC  R0           BACK ONE
0423          JMP  RPTEX        THEN EXIT
0424  RPTFA   MOVB R1,@ALTKEY   STASH CURRENT CHAR
0425          MOVB @CURSOR,R1   CURSOR IN R1
0426          BLWP @VSBW        WRITE CURSOR
0427  *
0428  * THE LOOP AT RPT2 DELAYS 8/60THS UNLESS KEY IS RELEASED
0429  *
0430  RPT2    BLWP @KSCAN       SCAN KEYBOARD
0431          LIMI 2            INTS ON
0432          LIMI 0            THEN OFF
0433          CB   @KEYVAL,@NOKEY NO KEY?
0434          JEQ  RPTEX        IF SO, EXIT
0435          CB   @>8379,@BACKUP COMPARE TO 8
0436          JLT  RPT2         IF LESS, REPEAT
0437  *
0438  * AFTER 8/60THS, CURSOR ADVANCES ANOTHER STEP
0439  *
0440          JMP  RPT1A        ELSE JUMP BACK
0441  RPTEX   MOVB @ALTKEY,R1   OLD CHAR
0442          BLWP @VSBW        WRITE THAT
0443          B    @KEYFRC      THEN BRANCH BACK
0444  *
0445  * FOLLOWING IS THE "BLINK", DONE WITH USER INTERRUPT
0446  * EVERY 20 60THS, THIS WILL BLWP @CHVECT TO CHANGE
0447  * FROM CURSOR TO CHARACTER OR VICE VERSA
0448  *
0449  USRINT CB   @>8379,@TWENTY TIMER=20?
0450          JLT  INTEX        IF LESS, EXIT
0451          BLWP @CHVECT      ELSE CHANGE CHAR
0452  INTEX   RT                RETURN TO INTERRUPT HANDLER
0453  *
0454  * CHVECT CHANGES FROM CURSOR TO CHAR AND VICE VERSA
0455  * EVERY 20/60THS OF A SECOND.  (THAT'S 1/3 SECOND)
0456  *
0457  CHVECT DATA WS,CHG1       OUR OWN WORKSPACE, CHANGE CODE
0458  CHG1    BLWP @VSBR        READ CURRENT BYTE FROM SCREEN
0459          CB   R1,@CURSOR   IS THAT CURSOR?
0460          JEQ  CHG2         IF YES, JUMP
0461          MOVB @CURSOR,R1   ELSE GET CURSOR
0462          BLWP @VSBW        AND WRITE THAT
0463          JMP  CHGX         THEN EXIT
0464  CHG2    MOVB @ALTKEY,R1   PUT OLD CHAR IN R1
```

Continued on page 28

Continued from page 27

```
0465          BLWP @VSBW          WRITE THAT
0466  CHGX    CLR  @>8378         CLEAR TIMER
0467          RTWP                THEN RETURN
0468  *
0469  * BLNFLD CLEARS A SCREEN AREA
0470  * STARTING AT R0 POSITION, EXTENDING R4 SPACES
0471  *
0472  BLNFLD MOV  R0,R3           SAVE R0 IN R3
0473          MOVB @ANYKEY,R1     SPACE CHAR IN R1
0474          MOV  R4,R2          COPY R4 TO R2
0475  BLN1    BLWP @VSBW          WRITE A SPACE
0476          INC  R0             MOVE POINTER
0477          DEC  R2             DEC COUNT
0478          JNE  BLN1           IF NOT ZERO, RPT
0479          MOV  R3,R0          GET OLD R0 BACK
0480          RT                  THEN RETURN
0481  *
0482  * DISSTR DISPLAYS A STRING ON SCREEN
0483  *
0484  DISSTR MOVB *R1+,R2         GET LENGTH BYTE
0485          SRL  R2,8           RIGHT JUSTIFY
0486          JEQ  DISX           IF ZERO, EXIT
0487          BLWP @VMBW          ELSE WRITE STRING
0488  DISX    RT                  RETURN
0489  *
0490  * DISNUM CONVERTS A FLOATING POINT NUMBER TO A STRING,
0491  * THEN DISPLAYS THAT STRING
0492  *
0493  DISNUM CLR  @FAC11          SET FOR BASIC FORMAT
0494          BLWP @GPLLNK        USE GPL LINK
0495          DATA CNS            CONVERT F.P. AT FAC TO STRING
0496          MOVB @FAC12,R2      STRING LENGTH TO R2
0497          SRL  R2,8           RIGHT JUSTIFY
0498          MOVB @FAC11,R1      STRING ADDRESS TO R1
0499          SRL  R1,8           RIGHT JUSTIFY
0500          AI   R1,>8300       ADD >8300 OFFSET
0501          BLWP @VMBW          DISPLAY THE STRING
0502          RT                  RETURN
0503  *
0504  * KEYLOO WAITS FOR A KEYSTROKE, THEN RETURNS
0505  * IN THIS INSTANCE, WE'VE MADE THE CURSOR BLINK
0506  * WHILE KEYLOO IS EXECUTING.
0507  *
0508  KEYLOO BLWP @KSCAN          SCAN KEYBOARD
```



```
0509          LIMI 2              ALLOW INTS
0510          LIMI 0              THEN STOP
0511          CB   @STATUS,@ANYKEY ANY KEY?
0512          JNE  KEYLOO         IF NOT, REPEAT
0513          MOV  @KEYADR,R8     KEY AS WORD INTO R8
0514          RT                  THEN RETURN
0515  *
0516  * MOVNUM MOVES A FLOATING POINT NUMBER FROM
0517  * THE LOCATION POINTED BY R9 TO
0518  * THE LOCATION POINTED BY R10
0519  *
0520  MOVNUM LI   R4,8            EIGHT BYTES TO MOVE
0521  MOVBYT MOVB *R9+,*R10+      MOVE ONE, INC POINTERS
0522          DEC  R4             DECREMENT COUNT
0523          JNE  MOVBYT         IF NOT ZERO, REPEAT
0524          RT                  RETURN
0525
0526  *  GENERAL PURPOSE GPL LINK
0527  *  BY WARREN/MILLER
0528
0529  GPLLNK DATA GLNKWS
0530          DATA GLINK1
0531  RTNAD   DATA XMLRTN
0532  GXMLAD DATA >176C
0533          DATA >50
0534  GLNKWS EQU  $->18
0535          BSS  >08
0536  GLINK1 MOV  *R11,@GR4
0537          MOV  *R14+,@GR6
0538          MOV  @XTAB27,R12
0539          MOV  R9,@XTAB27
0540          LWPI GPLWS
0541          BL   *R4
0542          MOV  @GXMLAD,@>8302(R4)
0543          INCT @STKPNT
0544          B    @LDGADD
0545  XMLRTN MOV  @GETSTK,R4
0546          BL   *R4
0547          LWPI GLNKWS
0548          MOV  R12,@XTAB27
0549          RTWP
0550  *
0551  * DATA SECTION
0552  *
```

Continued on page 30

THE ART OF ASSEMBLY PART 71

Continued from page 29

```
0553  WS      BSS  32           OUR WORKSPACE
0554  INTLOC  DATA USRINT       USER INTERRUPT ADDRESS
0555  INSFLG  DATA 0            INSERT FLAG
0556  NUM1    BSS  8            STORAGE FOR FIRST NUMBER
0557  NUM2    BSS  8            STORAGE FOR SECOND NUMBER
0558  DELKEY  BYTE 3            FUNCTION-1 VALUE
0559  INSKEY  BYTE 4            FUNCTION-2 VALUE
0560  ERSKEY  BYTE 7            FUNCTION-3 VALUE
0561  TEMSTR  BSS  30           TEMPORARY STRING
0562  ALTKEY  BYTE 0            CURRENT CHARACTER FROM SCREEN
0563  ENTERV  BYTE 13           ENTER KEY VALUE
0564  CURSOR  BYTE 30           CURSOR CHAR
0565  BACKUP  BYTE 8            FUNCTION-S
0566  FWARD   BYTE 9            FUNCTION-D
0567  ANYKEY  BYTE 32           SPACE OR COMPARISON BYTE
0568  TWENTY  BYTE 20           CURSOR BLINK NUMBER
0569  NOKEY   BYTE >FF          NO KEY INDICATION
0570  EDGE    BYTE 31           EDGE CHAR
0571  N1STR   BYTE 18
0572          TEXT 'ENTER FIRST NUMBER'
0573  N2STR   BYTE 19
0574          TEXT 'ENTER SECOND NUMBER'
0575  ADDSTR  BYTE 11
0576          TEXT 'NUM1+NUM2= '
0577  SUBSTR  BYTE 11
0578          TEXT 'NUM1-NUM2= '
0579  MULSTR  BYTE 11
0580          TEXT 'NUM1*NUM2= '
0581  DIVSTR  BYTE 11
0582          TEXT 'NUM1/NUM2= '
0583  SINSTR  BYTE 13
0584          TEXT 'SIN OF NUM1= '
0585  EQUSTR  BYTE 24
0586          TEXT 'NUM1 IS EQUAL TO NUM2    '
0587  GRTSTR  BYTE 24
0588          TEXT 'NUM1 IS BIGGER THAN NUM2'
0589  LESSTR  BYTE 24
0590          TEXT 'NUM1 IS LESS THAN NUM2   '
0591  PAK     BYTE 21
0592          TEXT 'PRESS ANY KEY TO EXIT'
0593  OR8     BYTE 23
0594          TEXT 'OR FUNCTION-8 TO REPEAT'
0595          END
```


EXTENDED BASIC

MAKE1FROM2 V.2

Updated program makes combining sorted text files easy

BY W. LEONARD TAFFS

Perhaps one of the most useful uses of the TI99/4A is for library-type files work. Information can be entered in any number of data programs, or through the use of Word processing programs such as TI-Writer, BA-Writer, Funnelweb, etc. These can be saved as Display/Variable files (D/V 80) — perhaps the most common type of file.

All Writer-type programs or databases have their limitation with respect to the size of file that can be maintained. TI-BASE has perhaps the greatest capacity for records. However, the speed with which information can be accessed varies. These various limitations, particularly the buffer limit of word processing programs, led me to create programs that could find an alternative to these problems.

My MASSREADV2 program was perhaps the first major step in this direction. It enabled one to assemble any number of D/V 80 files in tandem fashion, creating one master D/V 80 file. The files this program "tandem-ized" into a larger file had to be sorted files. Furthermore, each file had to be further down the alphabet. For instance, MASSREADV2 would open and read an A to D file, and open and save records from the A to D file in another (master) file. When it finished the A to D file, it would open the next E to H file, appending these records to those already saved from A to D, etc. The only limitation to the size of file that MASSREADV2 could create was the sector limitation of the disks being used. A 1,440-sector disk could accommodate a sizable number of records, the number of records possible depending upon the length of individual records.

Enabling MASSREADV2 to assemble such files required a lot of preliminary work setting up the individual files and being sure to sort them. This could be quite time-consuming.

Ever since the creation of MASSREADV2, I felt there must be a more practical solution which would make the use of MASSREADV2 unnecessary.

Recently I found the means for doing so which is the main listing here. It is perhaps among my most significant contributions to date.

The only requirement for this program is that the files be sorted files and that each file uses the "~" character as an end-of-file (EOF) marker in its last record. If one is processing files that frequently make use of the "~" character (126), then the program and files will have to be edited, to make use of a different EOF

Continued on page 32

Continued from page 31

marker.

MAKE1FROM2 will read any two D/V 80 sorted files and assemble them in one composite sorted file, a process that can be repeated to accommodate as many separate files as disk-sector capacity will allow. Any sorted file used by this program should be *carefully checked* before using, as any possible record out of order will foul up the successful use of this program!

This program, recently published in MICROpendium magazine, has update lines added to the version previously published, to overcome a problem one may run into with various D/V 80 files that sometimes have unrecognizable characters in them, usually character 128. The version previously published in MICROpendium may not copy the final records of a file properly if a file is used as the first file, and the longer file used as the second in order.

MAKE1FROM2

```
1 REM [MAKE1FROM2] Vs. 2
By W. LEONARD TAFFS, SW99ers
      Tucson, Arizona
      3 - 17 - 98 !189
2 !!131
3 ! To Read Two D/V 80 Files
    and Combine in Sorted
    Order as a Composite
    OUTPUT File. !219
4 !!131
5 ! The 2 files must have
    been previously sorted
    and use ~ as EOF marker!
 !082
6 !!131
7 ! OPEN #1:F1$  (File 1)
    OPEN #2:F2$  (File 2)
    OPEN #3:OF$  (Out File)
    OPEN #5:"PIO"(Printer) !
009
8 !!131
9 ! NO CARRIAGE RETURN ADDED
 !016
10 !!131
100 GOTO 150 !229
110 A,A1,B,B1,CL1,CL2,CTA,CT
AT,CTB,CTBT,F3,FCT,K,OF,OK,O
K2,PCT,PR,S,SK,U !090
120 A$,A1$,B$,B1$,D1$,D2$,D3
$,DT$,F1$,F2$,F3$,PR$,U$ !24
9
130 CALL CLEAR :: CALL KEY !
164
140 !@@P- !128
150 CALL CLEAR :: DISPLAY AT
(1,7):"[MAKE1FROM2] vs.2": :
"By W. Leonard Taffs, SW99er
s": : :"Combines Two Sorted
Files in":"  Alphabetical O
rder!" !202
160 DISPLAY AT(9,2):"Both in
put files must have":" been
previously sorted and":"have
 used ""~"" as EOF marker."
!134
170 DISPLAY AT(14,1):"Screen
 Display, Printer, and":"  o
r OutPut File Options" !106
180 DISPLAY AT(18,4):"User "
```



```
"(0/1)"" Response:": :TAB(8)
;"1=YES";TAB(17);"0=NO" :: D
ISPLAY AT(24,2):"Press <Ente
r> to continue" !173
190 U$="User Terminated Prog
ram" :: PR$="Is PRINTER Onli
ne?" :: FCT=1 !147
200 CALL KEY(0,K,S):: IF S<1
 THEN 200 !014
210 CALL CLEAR :: INPUT "Use
 Printer? (0/1) ":PR :: PRIN
T :: INPUT "Open Out File? (
0/1) ":OF :: PRINT :: IF PR
THEN PRINT PR$ :: OPEN #5:"P
IO" :: PRINT #5: :!199
220 PRINT :: INPUT "Date? (0
pt.) ":DT$ :: PRINT :: CALL
CLEAR !246
230 PRINT :: INPUT "Read Fir
st File: ":F1$ :: PRINT :: I
NPUT "From DSK: ":D1$ :: PRI
NT :: F1$="DSK"&D1$&"."&F1$
!174
240 INPUT "Read Second File:
 ":F2$ :: PRINT :: INPUT "Fr
om DSK: ":D2$ :: PRINT :: F2
$="DSK"&D2$&"."&F2$ :: INPUT
 "O.K.? (0/1) ":OK :: IF OK<
>1 THEN 230 !238
250 PRINT :: IF OF THEN INPU
T "Save as: ":F3$ :: PRINT :
: INPUT "To Disk: ":D3$ :: P
RINT :: F3$="DSK"&D3$&"."&F3
$ :: CALL CLEAR !226
260 DISPLAY AT(12,4):"In: 1
";F1$ :: DISPLAY AT(15,4):"I
n:2 ";F2$ :: IF OF THEN DISP
LAY AT(18,3):"Out: 3 ";F3$ E
LSE INPUT "SURE? (0/1) ":OK2
 :: IF OK2<>1 THEN 230 !025
270 INPUT "SURE? (0/1) ":OK
:: IF OK<>1 THEN 230 !203
280 CALL CLEAR :: OPEN #1:F1
$,INPUT :: OPEN #2:F2$,INPUT
 !045
290 IF OF THEN OPEN #3:F3$,O
UTPUT :: OF=1 ELSE OF=0 !249
300 DISPLAY AT(1,3):"~~ make
1from2 VS.2 ~~" !079
310 IF CL1 THEN GOSUB 680 ::
 GOTO 350 ELSE 320 !169
320 LINPUT #1:A$ :: A=LEN(A$
):: IF A>1 THEN A=ASC(SEG$(A
$,1,1)):: DISPLAY AT(3,1):A$
,A;CTAT !248
330 IF POS(A$,CHR$(128),1)TH
EN 320 !242
335 IF (ASC(A$)=128)+(ASC(A$
)=126)THEN CL1=1 :: CLOSE #1
 :: IF (CL1=1)*(CL2=1)THEN 6
10 !115
340 IF (A$=A1$)*(A$=B1$)THEN
 310 ELSE CTA=CTA+1 :: CTAT=
CTAT+1 :: FCT=FCT+1 :: IF CT
A=1 THEN A1$=A$ :: A1=ASC(SE
G$(A1$,1,1)):: CTA=0 !054
350 IF SK THEN 440 !084
360 IF CL2 THEN GOSUB 680 ::
 GOTO 310 ELSE 370 !180
370 LINPUT #2:B$ :: B=LEN(B$
):: IF B>1 THEN B=ASC(SEG$(B
$,1,1)):: DISPLAY AT(7,1):B$
,B;CTBT !006
380 IF POS(B$,CHR$(128),1)TH
EN 370 !037
```

Continued on page 34

EXTENDED BASIC

Continued from page 33

```
385 IF (ASC(B$)=128)+(ASC(B$
)=126)THEN CL2=1 :: CLOSE #2
 :: IF (CL1=1)*(CL2=1)THEN 6
10 !119
390 CALL KEY(0,K,S):: IF S<>
1 THEN 400 :: IF (K=81)+(K=1
13)THEN CLOSE #1 :: CLOSE #2
 :: IF OF THEN CLOSE #3 ELSE
 610 !208
400 IF (B$=B1$)*(B$=A1$)THEN
 350 ELSE CTB=CTB+1 :: CTBT=
CTBT+1 :: FCT=FCT+1 :: IF CT
B=1 THEN B1$=B$ :: B1=ASC(SE
G$(B1$,1,1)):: CTB=0 !106
410 CALL KEY(0,K,S):: IF S<>
1 THEN 430 :: IF (K=81)*(K=1
13)THEN CLOSE #1 :: CLOSE #2
 :: IF OF THEN CLOSE #3 ELSE
 610 !240
420 CALL KEY(0,K,S):: IF S<>
1 THEN 420 !171
430 SK=1 !086
440 IF (A$<B$)*(A1$<B1$)THEN
 DISPLAY AT(12,1):F1$;" Prin
t #";FCT: :A$ !195
450 IF PR THEN IF (A$<B$)*(A
1$<B1$)THEN PCT=PCT+1 :: PRI
NT #5:TAB(5);PCT;A$ !077460
IF OF THEN IF (A$<B$)*(A1$<B
1$)THEN PRINT #3:A$ :: F3=F3
+1 :: DISPLAY AT(24,1):"Savi
ng: ";F3$;F3 !206
470 IF (A$<B$)*(A1$<B1$)THEN
 310 !071
480 IF (A$>B$)*(A1$>B1$)THEN
 DISPLAY AT(18,1):F2$;" Prin
t #";FCT: :B$ !205
490 IF PR THEN IF (A$>B$)*(A
1$>B1$)THEN PCT=PCT+1 :: PRI
NT #5:TAB(5);PCT;B$ !080500
IF OF THEN IF (A$>B$)*(A1$>B
1$)THEN PRINT #3:B$ :: F3=F3
+1 :: DISPLAY AT(24,1):"Savi
ng: ";F3$;F3 !209
510 IF (A$>B$)*(A1$>B1$)THEN
 360 !123
520 IF (A$<B$)*(B1$<A1$)THEN
 DISPLAY AT(18,1):F2$;" Prin
t #";FCT: :B$ !203
530 IF PR THEN IF (B$<A$)*(B
1$<A1$)THEN PCT=PCT+1 :: PRI
NT #5:TAB(5);PCT;B$ !078540
IF OF THEN IF (B$<A$)*(B1$<A
1$)THEN PRINT #3:B$ :: F3=F3
+1 :: DISPLAY AT(24,1):"Savi
ng: ";F3$;F3 !207
550 IF (B$<A$)*(B1$<A1$)THEN
 360 !121
560 IF (B$>A$)*(B1$>A1$)THEN
 DISPLAY AT(11,1):F1$;" Prin
t #";FCT: :A$ !196
570 IF PR THEN IF (B$>A$)*(B
1$>A1$)THEN PCT=PCT+1 :: PRI
NT #5:TAB(5);PCT;A$ !079580
IF OF THEN IF (B$>A$)*(B1$>A
1$)THEN PRINT #3:A$ :: F3=F3
+1 :: DISPLAY AT(24,1):"Savi
ng: ";F3$;F3 !208
590 IF (B$>A$)*(B1$>A1$)THEN
 310 !073
600 GOTO 310 !134
610 REM ** END PROGRAM ** !1
77
620 PRINT :"FILES CLOSED--EN
D OF PROGRAM": :!009
```


EXTENDED BASIC

```
630 IF PR THEN PRINT #5: :TA
B(10);F1$;" had ";CTAT;"Rec
s.":TAB(10);F2$;" had ";CTBT
;" Recs.  ";DT$ !193
640 PRINT "TOTAL READ: ";CTA
T+CTBT !087
650 IF OF THEN PRINT #3:"~"
:: CLOSE #3 :: PRINT "Outfil
e was: ";F3$:TAB(10);F3;" Re
cs." !038
660 PRINT :"FILES WERE:":F1$
;CTAT:F2$;CTBT :: IF U THEN
PRINT U$ !165
670 STOP !152
680 REM ** CHECK CLOSING **
!047
690 IF (CL1=1)*(CL2=1)THEN 6
10 !239
700 IF CL1 THEN 360 !038
710 IF CL2 THEN 310 !245
720 RETURN !136
```

V9T9

Comments on SuperSpace CVAC

Downloading files and disks to V9T9 using XMDM2TI

BY ROGER PRICE

Many people are trying to use XMODEM file transfers to V9T9 without using XMDM2TI and are then getting header file errors. In trying to use the XMDM2TI utility many people are likely not using the correct command line to make the utility work.

This only works if you have a fully configured working V9T9 program.

First you need to prepare a blank disk (file on the hard drive) or have plenty of space on an existing disk file. I then used Hyperterminal with Windows 95 and Telco with the TI using XMODEM to download program files to a real floppy. It seems that the single file transfers work only with program-type files. I did not want intermediate files on my hard drive so I saved them to 1.44mb floppy. The one bugaboo about XMDM2TI was that Ed Schwartz, author of V9T9, gave no example of the actual command line. So here is an example — ARK302 is the name of the program and the target disk is Disk-12:

Go to the V9T9\V6.0 directory with cd\v9t9\v6.0. Now type the following:

>UTILS\XMDM2TI.EXE A:\ARK302 C:\V9T9\V6.0\DISK-12

This should put the file into the disk (Disk-12) as a subdirectory file.

Start V9T9. Change your disk path with shift+ctrl+f9 to the new path, press Enter, then Escape key. If your program is in Extended BASIC, then use the

Continued on page 36

Continued from page 35

usual commands and your program should be there.

The only difference between downloading a file and downloading a whole disk is that you need to archive your disk into a single archived file. When archiving, I answer the question "All files?" No. If you say "yes," the program saves the created file in the wrong type to use. You need the file in Display/Fixed 128 format. Then use XMODEM to download the file to a 1.44 floppy. Use the same command line except that you add /A on to the XMDM2TI part like so — XMDM2TI/A. Now you will be putting the entire disk onto a V9T9 DISK as an archived file.

Screen shot of Arcturus

This will take awhile since it is a long file. When done, just start up V9T9 and load Funnelweb, then Archiver, use extract files and convert your arc file back to a complete disk just as you would if you had real disks. Try planning ahead for the disks you are going to put the archived file on and the disk where you are going to unarc the files to. I would not use real floppies with V9T9 disks because my experience with PC99 and from what I have read, I would keep the files on the hard drive.

My experience with the Transfer program and receive were not good. I do have the Transfer program if someone wants to experiment with it. It is on either 3.5-inch or 5.25-inch floppy. The files I downloaded with Transfer were not usable, seem to be mangled. Someday I may figure out what I'm doing wrong, but really we do not need Transfer unless you are trying to download the GROMs from the console.

Incidently, most of the Atari games do not run on V9T9 but they do on PC99. I have a five-sector BASIC catalog program that I made which I put on each disk. If you are not sure what is on a disk all you need to do is load catalog and whichever disk is in effect will load it and you can search out what disk and files are on each drive.


Samurai, from GIFmania

If you don't have anything I will send you a copy of the

Addatex version of V9T9 with five disks files and some programs, Funnelweb, catalog. For the disk send $1.50 for shipping and handling U.S. For foreign estimate postage. For the disk send to: Roger Price, 1015 N. River Drive, Marion, In 46952-2607.

TI GRAPHICS AND V9T9

Using V9T9 and the program PagePro, Gifmania or TI-Artist (see examples accompanying this article), you can put a TI graphic on the screen, then press CTRL+PRTSCN (printscreen key). This will put the graphic into the clipboard. Start Paint or other graphics program on your PC, paste your graphic, then save the picture to a bitmap file. Load in a Photoworks, Photoshop or other program that will handle graphics. Clip the file to save only the good part then paste the graphic into a page. This only works running V9T9 from Windows 95.


Fish, from GIFmania

SUPERSPACE AND CVAC

Just received the MICROpendium for March/April 98 and in reading the article by Charles Good, I would like to comment that the SuperSpace memory is available in PC99 versions 3 and 4 for the operation of the SuperSpace program CVAC. It is true that SuperSpace does not show up anywhere in the menu. You must have downloaded your own CVAC and your own or purchased programs to load into CVAC. I have in fact loaded Centipede and Defender many times into CVAC in the PC99 program and they both run.

You must start by loading the Editor/Assembler module and you must have only one module loaded. Then load the CVAC program into the E/A with option 3. Program filename is CVAC. Load the program that you have saved from module to disk and downloaded to PC99. Select: Load a cartridge file. Then reset the computer and the cartridge will be on the menu. I do not know if all of the features of SuperSpace work with V3 and V4 of PC99 but CVAC does.

If using the OPA menu you can have the OPA as the first cartridge and E/A as No. 2. After loading the game it will appear on the OPA menu on the right side as:BASIC, Centipede, Editor/Assembler.


City, from Gifmania

Seagate MFM drives listed

If you own a Myarc HFDC card you know that you have to use MFM hard drives. These drives haven't been manufactured for years, but they're still available at swap meets, flea markets and businesses that sell used computer equipment, such as Goodwill.

You can use the following specifications to determine suitability of Seagate drives when you come across them. The following drive sizes are listed here:

3.5 inch drives	Half-height drives	Full-height drives
ST-125	ST-212	ST-406
ST-138	ST-213	ST-506
	ST-225	ST-412
	ST-238/238r	ST-419
	ST-251	ST-425
	ST-251-1	ST-4026
		ST-4038
		ST-4038m
		ST-4051
		ST-4053
		ST-4096

ST - 125

Unformatted	25.6 mb
Formatted (17 sectors)	21.4 mb
Actuator type	Stepper
Tracks	2,460
Cylinders	615
Heads data/servo	4
Discs/type	2/thin film
Recording method	MFM
Transfer rate mbits/sec	5.0
Interface	506/412
TPI (tracks per inch)	824
BPI (bits per inch)	15,500
Average access - ms	40/28 msec*
Single track seek - ms	8 msec
MTBF (hours)	20,000
Power / +12v start-up (amps)	2.0
Power / +12v typical (amps)	0.35
Power / +5v typical (amps)	0.8
Landing zone	Auto park
IBM at drive type	6

*ST125-0/ST125-1

ST - 138

Unformatted	38.4 mb
Formatted (17 sectors)	32.1 mb
Actuator type	Stepper
Tracks	3,690
Cylinders	615
Heads data/servo	6/0
Discs/type	3/thin film
Recording method	MFM
Transfer rate mbits/sec	5.0
Interface	412/mfm
TPI (tracks per inch)	824
BPI (bits per inch)	15,500
Average access - ms	40/28*
Single track seek - ms	8
MTBF (hours)	20,000
Power / +12v start-up (amps)	2.0
Power / +12v typical (amps)	0.4
Power / +5v typical (amps)	0.8
Landing zone	Auto park
IBM at drive type	3

*ST138-0/ST138-1

ST - 212

Unformatted	12.76 mb
Formatted (17 sectors)	10.0 mb
Actuator type	Stepper
Tracks	1,224
Cylinders	306
Heads data/servo	4/0
Discs/type	1/0
Recording method	MFM
Transfer rate mbits/sec	5.0
Interface	506/412
TPI (tracks per inch)	550
BPI (bits per inch)	10,560
Average access - ms	65 msec
Single track seek - ms	23 msec
MTBF (hours)	11,000
Power / +12v start-up (amps)	3.2
Power / +12v typical (amps)	1.0
Power / +5v typical (amps)	1.0
Landing zone	319
IBM at drive type	1

ST - 213

Unformatted	12.8 mb
Formatted (17 sectors)	10.7 mb
Actuator type	Stepper
Tracks	1,230
Cylinders	615
Heads data/servo	2/0
Discs/type	1/oxide
Recording method	MFM
Transfer rate mbits/sec	5.0
Interface	506/412
TPI (tracks per inch)	588
BPI (bits per inch)	9,827
Average access - ms	65 msec
Single track seek - ms	20 msec
MTBF (hours)	20,000
Power / +12v start-up (amps)	2.2
Power / +12v typical (amps)	0.9
Power / +5v typical (amps)	0.8
Landing zone	670
IBM at drive type	N/a

ST - 225

Unformatted	25.6 mb
Formatted (17 sectors)	21.4 mb
Actuator type	Stepper
Tracks	2,460
Cylinders	615
Heads data/servo	4/0
Discs/type	2/oxide
Recording method	MFM
Transfer rate mbits/sec	5.0
Interface	506/412
TPI (tracks per inch)	588
BPI (bits per inch)	9,827
Average access - ms	65 msec
Single track seek - ms	20 msec
MTBF (hours)	20,000
Power / +12v start-up (amps)	2.2
Power / +12v typical (amps)	0.9
Power / +5v typical (amps)	0.8
Landing zone	670
IBM at drive type	2

ST - 251

Unformatted	51.2 mb
Formatted (17 sectors)	42.8 mb
Actuator type	Stepper
Tracks	4,920
Cylinders	820
Heads data/servo	6/0
Discs/type	3/thin film
Recording method	MFM
Transfer rate mbits/sec	5.0

Continued on page 40

SEAGATE MFM HARD DRIVES

Continued from page 39

Interface 506/412
TPI (tracks per inch) 777
BPI (bits per inch) 9,935
Average access - ms ..... 40 msec
Single track seek - ms ..... 8 msec
MTBF (hours) 20,000
Power / +12v start-up (amps) 2.0
Power / +12v typical (amps) 0.7
Power / +5v typical (amps) 0.9
Landing zone ..... Auto park
IBM at drive type 3*
* requires partitioning software

ST-251-1

Unformatted ..... 51.2 mb
Formatted (26 sectors) ..... 42.8 mb
Actuator type ..... Stepper
Tracks 4,920
Cylinders 820
Heads data/servo 6/0
Discs/type ..... 3/thin film
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 412/mfm
TPI (tracks per inch) 777
BPI (bits per inch) 9,935
Average access - ms 28
Single track seek - ms 8
MTBF (hours) 20,000
Power / +12v start-up (amps) 2.4
Power / +12v typical (amps) 0.5
Power / +5v typical (amps) 1.0
Landing zone ..... Auto park
IBM at drive type ..... 3 or 44 *
* requires partitioning software

ST - 406

Unformatted ..... 6.38 mb
Formatted (17 sectors) ..... 5.0 mb
Actuator type ..... Stepper
Tracks 612
Cylinders 306
Heads data/servo 2/0
Discs/type 1/oxide
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 345
BPI (bits per inch) ..... 9,074 (max)
Average access - ms ..... 85 msec.
Single track seek - ms ..... Not listed
MTBF (hours) 11,000
Power / +12v start-up (amps) 3.5
Power / +12v typical (amps) 1.6
Power / +5v typical (amps) 1.1
Landing zone 319
IBM at drive type ..... N/a

ST - 506

Unformatted ..... 6.38 mb
Formatted (17 sectors) ..... 5.0 mb
Actuator type ..... Stepper
Tracks 612
Cylinders 153
Heads data/servo 4/0
Discs/type 2/oxide
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 255
BPI (bits per inch) 7690
Average access - ms 170
Single track seek - ms 3
MTBF (hours) 11,000
Power / +12v start-up (amps) 4.5
Power / +12v typical (amps) 1.8
Power / +5v typical (amps) 7
Landing zone 157
IBM at drive type ..... N/a

SEAGATE MFM HARD DRIVES

ST - 412

Unformatted ..... 12.76 mb
Formatted (17 sectors) ..... 10.0 mb
Actuator type ..... Stepper
Tracks 1,224
Cylinders 306
Heads data/servo 4/0
Discs/type 2/oxide
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 345
BPI (bits per inch) ..... 9,074 (max)
Average access - ms ..... 85 msec
Single track seek - ms ..... Not listed
MTBF (hours) 11,000
Power / +12v start-up (amps) 3.5
Power / +12v typical (amps) 1.7
Power / +5v typical (amps) 1.1
Landing zone 320
IBM at drive type 1

ST - 419

Unformatted ..... 19.14 mb
Formatted (17 sectors) ..... 15.0 mb
Actuator type ..... Stepper
Tracks 612
Cylinders 306
Heads data/servo 6/0
Discs/type 3/oxide
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 345
BPI (bits per inch) 9,074
Average access - ms ..... 85 msec
Single track seek - ms ..... <16.7
MTBF (hours) 11,000
Power / +12v start-up (amps) 3.5
Power / +12v typical (amps) 1.6
Power / +5v typical (amps) 1.1
Landing zone 319
IBM at drive type ..... N/a

ST - 425

Unformatted ..... 25.52 mb
Formatted (17 sectors) ..... 20.0 mb
Actuator type ..... Stepper
Tracks 2,448
Cylinders 306
Heads data/servo 8/0
Discs/type 2/oxide
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 550
BPI (bits per inch) ..... 10,568 (max)
Average access - ms ..... 65 msec
Single track seek - ms ..... 19.67 msec
MTBF (hours) 11,000
Power / +12v start-up (amps) 3.5
Power / +12v typical (amps) 1.6
Power / +5v typical (amps) 1.1
Landing zone 319
IBM at drive type 13

ST - 4026

Unformatted ..... 25.6 mb
Formatted (17 sectors) ..... 21.4 mb
Actuator type ..... Voice coil
Tracks 2,460
Cylinders 615
Heads data/servo 4/1
Discs/type ..... 3/thin film
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 625
BPI (bits per inch) 9,617
Average access - ms ..... 40 msec

Continued on page 42

SEAGATE MFM HARD DRIVES

Continued from page 41

Single track seek - ms ..... 8 msec
MTBF (hours) 12,000
Power / +12v start-up (amps) 4.0
Power / +12v typical (amps) 1.5
Power / +5v typical (amps) 1.5
Landing zone ..... Auto park
IBM at drive type 2

ST -4038

Unformatted ..... 38.2 mb
Formatted (17 sectors) ..... 31.9 mb
Actuator type ..... Voice coil
Tracks 3,665
Cylinders 733
Heads data/servo 5/1
Discs/type ..... 3/thin film
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 750
BPI (bits per inch) 9,617
Average access - ms ..... 40 msec
Single track seek - ms ..... 8 msec
MTBF (hours) 12,000
Power / +12v start-up (amps) 4.0
Power / +12v typical (amps) 1.5
Power / +5v typical (amps) 1.5
Landing zone ..... Auto park
IBM at drive type ..... 20 or 22

ST - 4038m

Unformatted ..... 38.17 mb
Formatted (17 sectors) ..... 31.9 mb
Actuator type ..... Voice coil
Tracks 3665
Cylinders 733
Heads data/servo 5/1
Discs/type 3/sputtered
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 750
BPI (bits per inch) 9,617
Average access - ms ..... 40 ms
Single track seek - ms ..... 8 ms
MTBF (hours) 12,000
Power / +12v start-up (amps) 4.0
Power / +12v typical (amps) 1.5
Power / +5v typical (amps) 1.5
Landing zone ..... Auto park
IBM at drive type 8

ST - 4051

Unformatted 50.88
Formatted (17 sectors) 42.52
Actuator type ..... Voice coil
Tracks 4,885
Cylinders 977
Heads data/servo 5/1
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 960
BPI (bits per inch) 9,720
Average access - ms 40
Single track seek - ms 8
MTBF (hours) 12,000
Power / +12v start-up (amps) 4
Power / +12v typical (amps) 1.5
Power / +5v typical (amps) 1.5
Landing zone ..... Auto park
IBM at drive type ..... 11 or 17 *
* requires partitioning software

ST - 4053

Unformatted 53.3
Formatted (17 sectors) 44.5
Actuator type ..... Voice coil
Tracks 5,120
Cylinders 1,024
Heads data/servo 5/1

SEAGATE MFM HARD DRIVES

Discs/type 3/sputtered
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 1,031
BPI (bits per inch) 9,792
Average access - ms 28
Single track seek - ms 6
MTBF (hours) 15,000
Power / +12v start-up (amps) 4.0
Power / +12v typical (amps) 1.3
Power / +5v typical (amps) 1.3
Landing zone ..... Auto park
IBM at drive type ..... 11 or 17 *
* requires partitioning software

ST - 4096

Unformatted 96.0
Formatted (17 sectors) 80.2
Actuator type ..... Voice coil
Tracks 9,216
Cylinders 1,024
Heads data/servo 9/1
Discs/type 5/sputtered
Recording method ..... MFM
Transfer rate mbits/sec 5.0
Interface 506/412
TPI (tracks per inch) 1,031
BPI (bits per inch) 9,792
Average access - ms 28
Single track seek - ms 6
MTBF (hours) 15,000
Power / +12v start-up (amps) 4.0
Power / +12v typical (amps) .. 1.3 **
Power / +5v typical (amps) 1.3
Landing zone ..... Auto park
IBM at drive type ..... 35 or 12 *
* requires partitioning software
** (1.5 amps for 2-board ST4096 only)

NEWSBYTES

Shaw has Web page

Stephen Shaw of the TI Users Group, UK, has a Web site at http://www.btinternet.com/~shawweb/stephen/book.htm

In the TI*MES newsletter he says that he will be happy to consider any TI-related text on PC disk (preferably in TEXT format) for the Web site, either on a permanent or revolving basis.

Chicago sets Faire

The TI International World's Faire, sponsored by the Chicago Users Group, will be held Nov. 14 at American Legion Post 42 in Evanston, Illinois, according to Hal Shanafield of the group. Address of the Legion Post is 1030 Central.

Already committed to holding seminars are Bruce Harrison, who will also introduce some new products, and Lew King, who will discuss Term 80 on the Internet. Shanafield says he expects both old and new vendors at the event, noting that the price of vendor tables has been reduced to $10.

In addition, attendees will receive a free disk being brought to the Faire by Berry Harmsen of the Dutch TI Users Group, an "unhackable" disk with new European software. Shanafield notes that this will be available only at the

Continued on page 44

NEWSBYTES

Continued from page 43

event as it cannot be copied.

For further information, contact the Chicago Users Group, P.O. Box 7009, Evanston, IL 60204-7009.

Harrison modifies programs

The AMS versions of Bruce Harrison's Slideshow and Video Titler have been modified so they will work with either the SAMS card of the SouthWest Ninety-Niners or the AMS emulator being produced by Michael Becker and his associates in Germany, according to Harrison.

Gerd Weissmann now has the revised editions and is authorized to sell them in Europe. Contact him for pricing at Koenigstr. 17-19, 67655 Kaiserlautern, Germany.

Harrison is selling the new editions for U.S. and Canadian customers. Updates are $1 apiece, and new disks are $5 each including shipping and handling. Contact Harrison at 5705 40th Place, Hyattsville, MD 20781.

HUG offers newsletter

In order to solicit new membership, the Hoosiers Users Group is offering a free sample issue of its newsletter to interested parties, according to Dan Eicher, HUG president. Dues are 20 US dollars per year (for those in the continental US — slightly higher for those outside the USA) and the newsletter comes out bi-monthly.

Interested persons canwrite Hoosiers Users Group, c/o Dan Eicher, 2720 Palo Verde Court, Indianapolis, IN 46227 or e-mail: Eicher@Delphi.com or leave a message via the group's S&T BBS at (317).782.9942.

Harrison releases Transfer for AMS

Bruce Harrison has created Transfer, for people who have both TI and PC computers, designed to work with the AMS card.

Harrison says the program is designed for people who download Internet text files larger than TI editing programs can handle. The program inputs the files via RS-232 (direct connection, no modems) from PC to TI, and stores the incoming records in the AMS. When the file is done, the program prompts for a file name to start saving to TI disk. It will then create a series of D/V80 files, each editable by Funnelweb's Text Editor.

Harrison says that, with a 256K AMS, files of about 240K can be transferred and saved into disk files on the TI. The program auto-increments the last character of the file name while saving. On both input and output cycles, the records of the file are seen on the TI's screen, starting at row 17, column 3.

The public domain SS/SD disk contains complete instructions, and is available for $1 from Harrison at 5705 40th Place, Hyattsville, MD 20781, or from the Lima Users Group, P.O. Box 647, Venedocia, OH 45894.

BUGS AND BYTES

TI for the taking

Curtis Adams has an unexpanded TI system with a lot of software, including a Speech Synthesizer, that he wants to give to someone as he "hates to throw anything away." Contact him at 1255 W-5175 S, Riverdale, UT 84405, or (801) 399-5176.

Crash overcome

Bruce Harrison writes:

At the Lima gathering, I introduced MIDI Play-In as scheduled, but ran into a crash situation in the middle of the demo. A few customer left Lima with copies of the program on the promise that updated copies would be sent out just as soon as the cause of the crash could be found and corrected. Revised copies were sent out before the end of May, labeled as Version 1.1. I think I've sent the new version to everyone who got the original release at Lima, but if I've missed anyone, the missed person should send a card of letter (*not e-mail*) to me at 5705 40th Place, Hyattsville, MD 20781. I'll quickly supply the revised edition at no cost.

MDOS 6.00 released

MDOS 6.00 has been made available to a variety of online sites, according to a notice from Tim Tesch that appeared on the TI list server. MDOS 6.00 includes both the MDOS 6.00 distribution archive and the MDOS 6.00 source file archives. Tesch encourages anyone with a BBS or web site to post the files. However, he cautions that the files should be verified before making them available to others.

The files can be found at ftp://ftp.whtech.com/pub/mdos/.

"Anyone who wishes to download the source may do so, I only ask that it be distributed as-is and that all files be kept together. I don't believe I included instructions on how to reassemble MDOS; my immediate goal was to release the code to the remaining Geneve users. This goal has been accomplished," Tesch said.

Tesch can be reached at ttesch@juno.com or ttesch_myarc@juno.com.

O'Neil considers AT keyboard production

In response to a posting on the TI list server, Don O'Neil of Western Horizon Technologies posted the following message which indicates a willingness to renew production of the WHT AT keyboard interface:

We made, and may be making again, an AT keyboard interface that had RAM as well as EPROM in it that resided in the console ROM area. This device clipped on the 9900 and gave you up to 64K of 16 bit 0-wait RAM, and 64K of 0-wait EPROM. We're currently out of boards, but I've been contemplating making more, but there doesn't seem to be much interest anymore.

Schnoz-ola and Freeware Games

BY CHARLES GOOD

SCHNOZ-OLA by Funware

This is a copyright 1983 8K cartridge game that has only recently been made available to the TI community by Competition Computer. It was never released by Funware and there is essentially no mention of it in 1983/84 computer magazines. All you need to play the game is the cartridge and a 99/4A console. Joysticks are optional.

Your man Schnoz is at the top of a stair-step pyramid. Precious tokens, looking very much like Munchman dots, are lined up along the length of each pyramid step. You are supposed to move Schnoz over all the tokens, picking them up as you go, and then climb back to the top of the pyramid. You can jump to lower levels and you can climb ladders to higher levels. You move the arrow keys or joystick left/right to walk along the pyramid ledges and up/down to change pyramid levels. When you clear all the tokens you can go on to new screens.

The game is not easy. Ola balls bounce down from the top of the pyramid and roll along pyramid ledges. If an ola ball hits Schnoz he is dead. Schnoz can run away from these balls or jump over them, but this is difficult because balls are dropping thick and fast. You get three Schnozes to start the game and bonus Schnozes when you earn enough points.

If you are a cartridge collector you probably don't have this one yet. This fast action game is being offered by Competition Computer for $19.95.

FREEWARE GAMES by Carsten Ziepke

Carsten Ziepke wrote these games and demos between 1983 and 1987. Some were apparently sold commercially to the European market at that time but I have never seen them before. Now they are available for free downloading from Carsten's well-designed Web site at http://members.aol.com/lapinkult/ti/tifrst.htm. The Web site has descriptions and screen shots of most games.

You can download individual games or all of them at once in any of several formats. Just right click on the download link of your choice and select "save as" to store the software on your PC hard drive. The download formats are ASCII, either of the two disk formats supported by the V9T9 emulator, and PC99, a very comprehensive list. If you don't run either of the two PC 99/4A emulators then you should download in ASCII. This gets you a text file listing of each game's Extended BASIC code. You can use PC Transfer to move this text file onto a 99/4A disk and then use

RXB or Text Loader or several other software products to read this text file directly into Extended BASIC where you can save the result on a TI disk as an XB program. Or, if you don't have PC Transfer and a DSDD disk controller you can manually type in the ASCII listing into Extended BASIC and save the resulting XB program on a TI disk. If you use the V9T9 emulator, then the downloaded "file as a disk" or "disk on a disk" will run without further modification with the emulator. The PC99 version also runs directly under PC99 with no modification needed, and you get a nice bonus. The PC99 downloaded disk has a couple of items not found on the other download formats.

There are nine games and one demo in the complete collection. Most have music and some games have optional speech. Games include instructions in REM statements in the early line numbers of the game's code. In some games the text is in German.

The "America's Agent" game comes only with the PC99 version because it requires Mechatronics Extended Basic II Plus which comes with PC99 but which few non-PC99 owners possess. You have to search a house for some microfilm stolen by a Russian agent. The game makes use of the bit map graphics that are part of Mechatronics XB and is very hard to figure out. You have to break the game as it starts, type in a bunch of CALL LOADs , and then run the game again. This is a very cumbersome procedure and is one of the reasons I don't like Mechatronics XB. I think the version of the game I have is incomplete. I can't get it to work. Maybe a finished version will be available by the time you read this.

"Flotten Manoever" is a one-player version of the Battleship board game. The player and the computer each place ships on a grid. This game requires Mechatronics XB.

"Flotten Manoever" is a one-player version of the Battleship board game. The player and the computer each place ships on a grid. The human and computer opponents cannot see the location of one another's ships and take turns shooting at the other guy's coordinate squares. This game requires Mechatronics XB.

"Hoppy V" is a frogger clone that requires joysticks. You have to cross a road and river and you get your choice of two speeds for the moving objects. Joystick action is unusually precise for a game of this type. I like it.

In "Hunchback Rescue" you walk along the walls and into the rooms of a castle to rescue the imprisoned prince.

Continued on page 48

MICROREVIEWS

Continued from page 47

Graphics and music are excellent. There are rolling balls you have to jump over in order to progress. This is a hard game.

"Ski" is your typical "skiing down a slope full of obstacles" game. Only the cursor keys are needed. You can move left/right and you can speed up or slow down your rate of descent. In fact, you can slow down to negative speed so that you appear to move up the slope. Unlike many of these games, this one is easy enough for a klutz like me to play to its conclusion.

In "Treasure Diver" your diver dives from an island at the top of the screen to retrieve treasure at the bottom of the screen and move it back to the island at the top. There are an octopus and various fish which can make this difficult. Graphics are good.

"Spurnasse" is an excellent version of the mind game "Concentration." You look at tiles two at a time and try to match the graphic on the back of the tile. The game is for two to four players. A unique feature of this game is the ability to use two independent computer players as well as human players. You can have a three-player game in which one human plays against both computer players, or a 2 player game in which a human plays against one of the two computer players. The two computer players can also play a two-player game by themselves automatically. I think one of the computer players is designed to play a better game than the other. This is my personal favorite among all the Ziepke freeware games, in part because I enjoy mind games.

"Asteroids" is not what you might think. It is not a clone of TI Invaders. In this game you pilot a flying saucer and observe an asteroid explode into many pieces at the start of the game. You then have to dodge all the asteroid pieces.

"Tuerme von Pompeji" is a Tower of Hanoi clone. You have to move pieces of the pyramid-shaped tower one piece at a time from one side of the board to the other. The only rule is that you can't put a big piece on top of a smaller piece. This is a tough game of logic.

All these games can be downloaded for free from the Ziepke Web site. If you can't do this then send me $1 and I will send them to you on a SSSD TI disk.

ACCESS

Charles Good, P.O. Box 647, Venedocia, OH 45894. Phone: (419) 667-3131; e-mail: good.6@osu.edu.

Competition Computer (source for Schnoz-Ola), 350 Marcella Way, Millbrae, CA 94030. Phone: 800-471-1600 6 a.m.-3 p.m. local time M-F

Carsten Ziepke, Westring 268, D-24116 Kiel, Germany; e-mail: cziepke@ki.comcity.de; Web site for downloading software http://members.aol.com/lapinkult/ti/tifrst.htm.

REVIEW

Panasonic 2135 meets color printing needs of TI users

BY JERRY NOVAK

The following article appeared in Wordplay, the newsletter of the Portland Users of Ninety Nines user group. Novak is a member of the group.—Ed.

With things like Microsoft Word and soft fonts for the "dark side" set, seems like yet another good reason to move on to PCs, right? Well maybe not necessarily. I'm in love with my TI computer and not about to part with it even though I am DOS and Windows 95 literate.

Paging through Computer Shopper a while back I stumbled onto a Tri-State Computer advertisement for printers, specifically a Panasonic 2135 dot matrix printer for $100 (with factory rebate). I did a bit of research and found out that it's a 24-pin color printer, Prowriter and Epson compatible, so I figured I couldn't go wrong and ordered one.

When it arrived, I was a bit disappointed, because the Panasonic folk built the package to cater to PC/Windows users, supplying a Windows setup disk, a DOS setup disk, and a manual that told little more than how to use both disks and the controls on the user panel. However, on the DOS disk is a text file that contains all of the necessary printer codes plus a complete description of the printer's graphics modes. Now we're in business! Downloading a copy of it to an IBM formatted disk on a friend's computer, I took it home, transferred it to n disk via PC-Transfer and printed it.

I found that the printer has two emulation modes — IBM Prowriter and Epson LQ-860, both of which will work with the likes of TI-Writer, Artist and, with some printer set up, PagePro. The Prowriter side is O.K. with color, the six resident fonts in letter quality mode and its other features, but it's the Epson side that really shines!

Besides draft and the same six fonts in letter quality, it has italics, bold, double-strike, outline, shadow, double-high and double-wide highlights, all of which can be used singly or in unison. In addition, one pass can be made with a font on one color, while a second can be in another color in outline, shadow or both if one so desires. It can super/subscript print, set and use vertical and horizontal tabs. It can underline, over-score and strike through text in one pass with solid or broken single or double lines, this of course in addition to the standard IBM character line box set capabilities. Print quality is 24-pin excellent with a decent ribbon.

As if all this isn't enough, the printer will extend TI-Writer's 80-column (pica) limit with its own

Continued on page 50

REVIEW

Continued from page 49

onboard "word processor" mode. There are printer codes that will let one center, right align or right justify text in 80, 96, 120, 137, 160-column, proportional or proportional compressed modes. It can also micro-justify by user selected amounts.

Graphic capabilities far exceed the TI's with 8-pin 60, 120, 240, 80 (CRT 1), 90 (CRT 2) DPI and 24-pin 60, 120, 90 (CRT 3), and 180 and 360 DPI settings. However, I'm sure some enterprising individual will find a way to use the higher than 60 DPI settings.

As an option, for an extra $60 (spendy I know, but worth it) I bought the 64K buffer memory chip, which allows extra document room, and downloading of an extra couple of fonts, one for draft and one for letter quality.

All of the previously mentioned highlights, will work with this LQ font as well. I didn't purchase the other option, a 50-leaf cut sheet feeder, as the 2135 comes with a 15-sheet feeder as well as a back side tractor feed.

Summing it up, I feel that anyone with a bit of programming skills or proficiency in using transliteration codes should be able to do far more with this printer than the creators of TI-Writer, Artist or PagePro ever envisioned for our TIs. This printer has still other features, more than I care to mention here, making it a valuable addition to any TI or Geneve system, even without a conventional manual.

TUTORIAL

The power of CALL KEY

BY STEVE LANGGUTH
OZARK 99'ER USERS GROUP

The CALL KEY command in BASIC and Extended BASIC is one whose complete power may not be appreciated by many programmers. This article and list of examples is an attempt to explain some of the "hidden" capabilities of the CALL KEY statement so that you can get the most out of it in your own programs.

The information in this article was collected from several sources including : an excellent summary of the CALL KEY options, written by Joyce Corker of Waltham, Mass. (the examples that make up the second half of this article are completely hers) which has appeared in several other newsletters recently; and an article by Glenn Davis in the January 1985 edition of the MSP 99 Newsletter.

CALL KEY, as implemented on the TI 99/4A has six possible modes in which to operate. These modes are summarized below.

CALL KEY(0,KEY,STATUS)

When the mode specified is "0", the keyboard is scanned in the same

TUTORIAL

mode it was in previously. (The normal Basic mode is Mode 5 —see below— so when a CALL KEY(0,K,S) statement is used in Basic or Extended Basic, we are really telling the computer to scan using Mode 5 "just like you were doing before".)

CALL KEY(1,KEY,STATUS)

Mode 1 scans the left side of the keyboard only.

CALL KEY(2,KEY,STATUS)

Mode 2 scans the right side of the keyboard only.

CALL KEY(3,KEY,STATUS)

Mode 3 is the "99/4" mode. In this mode values for upper case letters are returned in "KEY" even if a lower case letter is pressed. (In other words, in this mode it doesn't matter whether the ALPHA LOCK key is up or down, all you get is upper case letters.)

This mode is particularly useful where upper case letters are important. For example, it is recommended that disk file names be in all upper case letters. By putting a CALL KEY(3,K,S) statement before the INPUT or ACCEPT statement, the name typed in by the user will be all in upper case letters. (TI Writer uses this mode when accepting file names.)

CALL KEY(4,KEY,STATUS)

Mode 4 (Pascal Mode) allows upper and lower case letters and all control and function keys. However, some of the "codes" are different than in Basic. For example, FCTN 4 will not "break" a program on an INPUT or ACCEPT statement, FCTN S will not backspace, etc. This is because these combinations of key strokes generate different codes in this mode than in Basic. (See the appendix in the User's Reference Guide.)

CALL KEY(5,KEY,STATUS)

Mode 5 is normal Basic mode and allows for both upper and lower case letters.

Examples

Below are several examples of how some of the modes described can be put to use.

Yes or no answers using CALL KEY 0.

```
100 CALL CLEAR
110 PRINT "Y OR N?"
120 CALL KEY(0,K,S)
130 IF K=78 THEN 170
140 IF K<>89 THEN 120
150 PRINT "YES"
160 GOTO 180
170 PRINT "NO"
180 END
```

Space bar or Enter answers using CALL KEY 5.

```
100 DISPLAY AT(3,3)ERASE ALL
:"PRESS SPACE BAR TO CONTIN
UE" :"PRESS ENTER TO PRINT"
110 FOR DELAY=1 TO 600 :: NE
XT DELAY
120 CALL KEY(5,K,S)
130 IF K=32 THEN PRINT "SPAC
E BAR PRESSED" :: GOTO 150 E
LSE IF K<>13 THEN 120
140 PRINT "ENTER WAS PRESSED
"
150 END
```

Continued on page 52

TUTORIAL

Continued from page 51

Alphabet answers that are forgiving of wrong case using CALL KEY 3.

```
100 DISPLAY AT(3,3)ERASE ALL
:"PRESS R TO REPEAT":"PRESS
P TO PRINT"
110 FOR DELAY=1 TO 600 :: NE
XT DELAY
120 CALL KEY(3,K,S)
130 IF K=82 THEN PRINT "HERE
 YOU WOULD GOTO YOUR REPEAT
 SUBPROGRAM" :: GOTO 150 ELS
E IF K<>80 THEN 120
140 PRINT "HERE YOU WOULD GO
 TO YOUR PRINT SUB"
150 END
```

Accessing Function and Control Keys using CALL KEY 5.

```
100 DISPLAY AT(3,3)ERASE ALL
:"PRESS CONTROL KEY AND COM
MA"
110 FOR DELAY=1 TO 600 :: NE
XT DELAY
120 CALL KEY(5,K,S)
130 IF K=128 THEN PRINT "CON
TROL AND COMMA PRESSED" ELSE
 120
140 END
```

or

```
100 DISPLAY AT(3,3)ERASE ALL
:"PRESS FUNCTION 8"
110 FOR DELAY=1 TO 600 :: NE
XT DELAY
120 CALL KEY(5,K,S)
130 IF K=6 THEN PRINT "FUNCT
ION 8 PRESSED" :: GOTO 140 E
LSE 120
140 END
```

As you can see, the CALL KEY command gives you a great deal of control over the input you are accepting.

USER NOTES

Knowing pinouts help drive installation

The following has appeared in several user group newsletters. It was written by Jack Zawediuk, and published here in an edited version. — Ed.

I picked up a 3.5-inch NEC double-sided drive and at first had trouble figuring out the wiring. Then I realized it's the same as any other drive, only it uses a different plug. I did have one wire different. It may be the NEC brand drive or it may be common to 3.5-inch drives. Anyway, I like the smaller drive and disks. It's quite and fast.

Here are the pinouts for TI's disk drives:

Pin 8 Index pulse
Pin 10 DSK1
Pin 12 DSK2
Pin 14 DSK3
Pin 16 Motor control
Pin 18 Stepper motor direction
Pin 20 Step pulse
Pin 22 Write data
Pin 24 Write enable
Pin 26 Track 00
Pin 28 Write protect
Pin 30 Read data

USER NOTES

Pin 32 Side select

Pins 1 to 33, the odd numbers on one side of the plug, are all ground. Pins 2, 4, 6, and 34 are not used on the TI.

As you can see, only 13 pins and ground are used out of a 34-conductor plug.

Other disk drive cards use pint 6 for drive No. 4.

The odd wire I had to hook up

Continued on page 54

NEWSBYTES

Continued from page 53

was pin 4 on the NEC drive to pin 10, 12, or 14, depending on which drive number I wanted the 3.5-inch drive to be.

Adding a fourth drive

The following was written by Jim Wiegand and appeared and has appeared in several user group newsletters. The reader takes full responsibility for the results of this hardware modification. — Ed.

This modification requires some wiring changes with soldering and drilling a small hole. Materials needed are:

SPST miniature toggle switch

dual drive data cable to match your drives (3.5- or 5.25-inch)

power "Y" adapter

These items can usually be found at Radio Shack or computer stores.

Pin No. 14 of the TI disk controller is the Drive Select (DS) line for DSK3. This is the wire that we will be working with. The task here is to install the toggle switch to allow selection of DSK3-A or DSK3-B.

To accomplish this, count the wires in the data cable — the colored wire is No. 1 — to wire No. 14. Cut this wire between the disk controller connector and the disk drive connector, about one inch from the disk drive connector.

Now find wire No. 14 between the two drive connectors and cut it near the center. Strip and tin these two ends and the end selector switch and mount it. With a 3.5-inch drive installed in a 5.25-inch bay, there is ample room.

Cut a suitable length of three-conductor wire (a piece of ribbon cable works well) to connect the switch to the data cable. Solder these wires to the switch with the center wire connected to the center terminal. At the drive connector, solder the center wire to wire No. 14 from the disk controller. The other two wires must be soldered to the cut wires between the disk drive connectors Insulate all soldered junctions and install the drives.

If all went well, the drive selector position (A or B) can now be identified. Type in OLD DSK3.TEST and press Enter. Watch the drive lights and label the switch position appropriately.

Program removes REMs

The following program removes REMarks from BASIC and Extended BASIC programs. The program from which you wish to remove REMs needs to be saved in MERGE format. Then simply run the program and follow the prompts. This program came from one of Jim Peterson's Tigercub disks.

REMREMOVER

```
100 DISPLAY AT(3,5)ERASE ALL
:"REM REMOVER": : :"Program
must be SAVEd in":"MERGE fo
rmat by":"SAVE DSK(filename)
,MERGE"
110 DISPLAY AT(12,1):"FILENA
ME? DSK" :: ACCEPT AT(12,14)
:F$ :: DISPLAY AT(14,1):"NE
W FILENAME? DSK" :: ACCEPT A
T(14,18):NF$
120 OPEN #1:"DSK"&F$,VARIABL
E 163,INPUT :: OPEN #2:"DSK"
&NF$,VARIABLE 163,OUTPUT
130 LINPUT #1:M$ :: A=POS(M$
,CHR$(131),1):: B=POS(M$,CHR
$(154),1):: A=MAX(A,B):: IF
 A=3 THEN 150 :: IF A=0 THEN
 PRINT #2:M$ :: GOTO 150
140 PRINT #2:SEG$(M$,1,A-1)&
CHR$(0)
150 IF EOF(1)<>1 THEN 130 ::
 CLOSE #1 :: PRINT #2:CHR$(2
55)&CHR$(255):: CLOSE #2
```


USER NOTES

CLASSIFIEDS

Internal speech card with speech synthesizer. A must for the Geneve. Includes 9 top games for Geneve. $59 plus $5 S&H. Disks converted to V9t9, PC99, or back to TI99/4A, $1 per disk. 99'er Magazines $1 each. Add Postage. Ph. 765-664-6001.

Download files TI to V9T9. .25 per file. Maximum of 25, TI Assembly Language Manual $5, Power Supply $5, 9 Games for Geneve, $9, add $1 S&H 765-664-6001.

FOR SALE
Fast Mac V34 modem (28.8K), works with any computer, $25 (obo). Call 512-255-1512 or email jkoloen@earthlink.net.

Classified ads are 10 cents per word. Payment (checks, money order, or credit card) must accompany ad. Mail to: MICROpendium Classifieds, P.O. Box 1343, Round Rock, TX 78680.

DISKS/BACK ISSUES

❑ Back Issues,$3.50 each to March 1996, later $6 each. List issues on separate sheet.
No price breaks on sets of back issues. Free shipping USA. Add $1, single issues to Canada/ Mexico. Other foreign shipping 75 cents single issue surface, $2.80 airmail. Write for foreign shipping on multiple copies.
OUT OF STOCK: V1#1-2; V2#1

GENEVE PUBLIC DOMAIN DISKS (SSSD unless specified)
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	$7	$5
❑ Series 2	$9	$7	$5
❑ Series 3	$9	$7	$5
❑ Series 4	$9	$7	$5
❑ Series 5	$9	$7	$5
❑ Series 6	$9	$7	$5