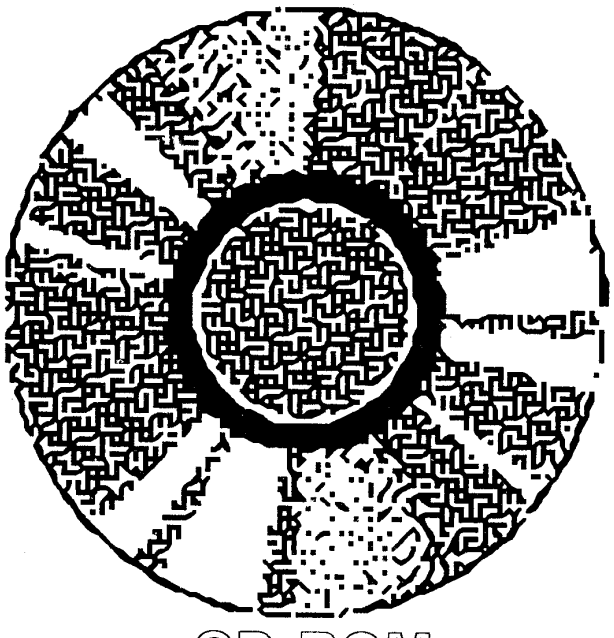
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 5 June 1995 $3.50



CD-ROM

project on again — see page 5

REVIEWED IN THIS ISSUE:

• Geneve SCSI •

• RXB v.1002 (or higher) •

• Funnelweb v.5.2 80-Column Editor •

CONTENTS

MICROpendium

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

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

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

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

Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone & FAX: (512) 255-1512
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@io.com
John Koloen.....Publisher
Laura Burns.....Editor

*READ THIS

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

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

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

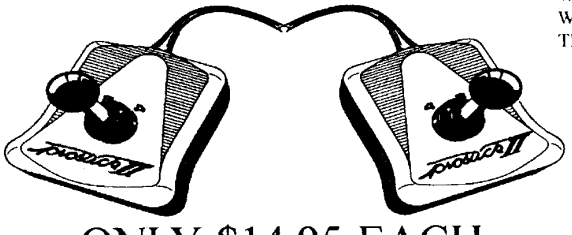
TEX COMP LTD.

TEX - COMP NOW HAS THE REMAINING DATABIOTICS CARTRIDGES IN LIMITED QUANTITIES. SEE LIST BELOW FOR SPECIFIC CARTRIDGES AND PRICES. WE ARE ALSO CARRYING THE PROSTICK 2 JOYSTICK FIRST SOLD BY US IN 1983 AND BACK NOW AT THE UNBELIEVABLY LOW PRICE OF $14.95 EA. DON'T FORGET THAT WE STILL CARRY ONE OF THE LARGEST SHAREWARE LIBRARIES AVAILABLE FOR THE TEXAS INSTRUMENTS HOME COMPUTER.

BARRAGE	$29.95
BEYOND PARSEC	$24.95
BEYOND WORD WRITER	$39.95
BLACK HOLE	$24.95
BOXER	$24.95
BREAKTHGH/ BREAKOUT	$24.95
BURGER BUILDER	$24.95
COMPUTER WAR	$29.95
CONCOLE CALC	$24.95
D-STATION I	$24.95
D-STATION II	$24.95
DRAGON FLYER	$29.95
DESKTOP PUBLISHER	$39.95
ESCAPE	$24.95
FACE CHASE	$24.95
GREAT WORD RACE	$29.95
JUMPY	$29.95
JUNK MAN JR	$29.95
MAGIC MEMORY	$49.95
MICRO PINBALL	$29.95
MICRO TENNIS	$29.95

DBT DataBioTics

MIDNIGHT MASON	$24.95
MINI WRITER	$24.95
MUNCHMAN II	$24.95
PROTYPER	$24.95
QMAZE	$29.95
RED BARON	$39.95
SORGON II	$29.95
SPACE PATROL	$24.95
SPOT SHOT	$29.95
SPY'S DEMISE	$24.95
STARGAZER	$39.95
STAR RUNNER	$29.95
STAR TRAP	$24.95
STRIKE 3	$29.95
TI WORKSHOP	$49.95
WORD WRITER	$29.95
WORD WRITER EXTRA	$29.95



ONLY $14.95 EACH

WE SOLD THESE HIGH QUALITY JOYSTICKS LAST IN 1983. WE HAVE FOUND A GOOD SUPPLY AND ARE CARRYING THEM NOW FOR A LOW $14.95 EA.

prostick II
ARCADE-STYLE JOYSTICK

JOYSTICK FEATURES:

- ***LEFT & RIGHT FIRE BUTTONS***
- ***SWITCH ABLE DIRECTION LIMITER***
- ***ARCADE STYLING***
- ***DURABLE DESIGN***
- ***UNIQUE LEAF SWITCH MECHANISM***

JOYSTICK ADAPTER

FOR CONVERTING ATARI TYPE JOY-STICKS FOR TI USE. USE WITH ABOVE JOYSTICKS

SPECIAL PACKAGE DEAL $34.95
featuring : 2) prostick II joysticks
1) joystick adapter

TEXAS INSTRUMENTS
AUTHORIZED DEALED

ORDER BY PHONE

VISA & MASTER CARD EXCEPTED

(818) 339-8924 FAX (818) 858-2785
24 HOURS A DAY

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

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

COMMENTS

A CD-ROM of TI programs

The on, off CD-ROM project is on again, thanks to Kyle Crichton. You may recall that Fred Moore, of the Los Angeles 99ers user group, had first undertaken the project a couple of years ago only to give up a year later. To refresh your memory, Moore planned to create a huge collection of TI programs and files on a single CD-ROM. What made this a possibility was the development of the Western Horizon SCSI card.

Crichton's says the first CD-ROM will contain the equivalent of 2,000 or more floppy diskettes. That's a lot of software. Just think about how much shelf space you'd gain. Those who are interested in owning the CD-ROM would have to buy a SCSI card, but why not? The SCSI card handles both CD-ROM and hard drives. Visitors at the Lima Multi User Group Conference saw a SCSI running on a TI system that performed such basic functions as loading files from a hard disk. That's all you need with a CD-ROM drive, since you can't save to them. (Rewritable CD-ROMs are available, but they cost thousands of dollars.)

Although there are no plans at this time for a CD-ROM containing Geneve programs, there are probably more SCSI cards running on Geneve systems than TI, simply because the Geneve version of the SCSI card has been available since last year. This is an interesting development and, with a target date of July, could make for a more interesting summer than we are accustomed to.

HISTORY OF THE TI

This month we're starting the longest feature we've ever run — Bill Gaskill's "History of the TI." Actually, this month's installment has little to do with TI. It focuses on the development of the microcomputer industry, especially the early micros. Those who are interested in how things came together will find the series interesting and entertaining. There's a lot of information there.

—JK

BUGS AND BYTES

United Kingdom users going strong

Though we didn't get the information in time to give advance notice, we're glad to report that the TI99/4A Users Group of the UK held its annual meeting June 3 in Derby, England. Richard Twyning, general secretary of the group, has been contacted by the Science Museum in London regarding the group's making a donation of a console.

So, is that frog calling him?

We have it on good authority that the real first name of Bud Mills of Horizon Computer is Clarence. He got that nickname because when he was in the military, he would order only "one Bud" on evenings out.

Term 80 update

The following message was posted on the Internet by Jeff Brown. Brown is developing the Term 80 terminal program.

I've completed the main program config and fixed the last couple bugs in the autodialer config (nice 80-column menu-driven configuration programs). They actually use the same kind of code as Term 80 and take up the same amount of space. Unfortunately, because of the large amount of data needed to run them, especially the 10 menus in the main config program, each takes 134 sectors of disk space! It was unavoidable, unless I felt like reworking an entire memory map.

Term 80 also had a couple additions, a modem init string is now executed on bootup and it allows a couple more options for the direct print mode (do not wait, wait forever, wait/timeout, wait/interruptible).

This is taking MUCH too long. The docs aren't done, and I expect it'll take at least a few days to do them right. There may also be a few more bugs in the program to sort out. Basically, I am going to end up 5-6 days off schedule from the planned release.

This is more than just a little step up from v1.6.6, which was the initially offered version of the program, now v2.7.0.

Basically, in terms of transfers, I fiddled with Ymodem uploads a bit, and couldn't get them going. It reboots. However, Xmodem/TIBBS, Xmodem and X1k work flawlessly and downloads in X/TIBBS, X/X1k and Ymodem are flawless. I achieved a CPS (character per second) rate of 780 between the TI and the Amiga (download to the TI), including reading of sectors from disks, etc. at 9600 BPS. Uploads are a bit slower, though, since Ymodem doesn't work. An upload to the Amiga ran at 650 CPS, 9600 baud.

Transferring to and from disk would be slower, though, since I was using my RAMdisk for testing. I imagine it'd be about the 450 CPS range. Transfers at 19,200 don't change much. The CPS is pretty constant but the number of errors increases because of data lost on one end, likely the Amiga since I was playing a MOD in the background.

TI CD-ROM project revived

Kyle Crichton already has 2,000 disks of software

By LAURA BURNS

A project to create a CD-ROM filled with public domain TI99/4A programs has received new life.

Kyle Crichton of Millbrae, California, estimates that he can start sending CD-ROMs in late July.

Fred Moore of the LA Computer Group began a project to put users group libraries on a CD-ROM in 1993, but abandoned it in 1994, reporting difficulties receiving the programs from other users groups. Moore archived the complete LA99 library on about 100 DSDD disks while working on the project.

Crichton says he is willing to travel to the users groups to copy their libraries.

"If some of them will be nice and send them, that's great," he adds. Users groups who send software will have priority in recording for the project, he says.

He says that, with his brother-in-law, he has invested in a SCSI CD-ROM recorder with a "huge hard drive," noting that this will eliminate some of the problems involved in having the CD-ROM made by a third party. He says he has invested more than $4,000 in equipment for the project, but does not expect to recover all his expenses. He says he wants to do the project on behalf of the TI99/4A community.

Crichton says the CD-ROM will be available on a "subscription" basis, at a price of $100. Users will receive an initial CD-ROM. As new programs are added, the users will be able to return their CD-ROMs to Crichton and in turn receive an updated CD-ROM with additional programs added, he says.

He says that he has "close to 2,000 disks already" on the initial CD-ROM, so "even now they're getting it for less than 10 cents a disk." He says he will provide caddies for the CD-ROMs for users who need them.

The default player for the CD-ROM would be a true XA multisession player, he says. If someone has a non-multisession CD-ROM player, Crichton says he can support it, but would have to add a media charge of $10 for that person's updates. He notes that being able to provide custom service is one advantage of having his own equipment platforms

The CD-ROM is ISO9660 compliant; it will work on Macintosh or IBM CD-ROM platforms and should work with Horizon's SCSI drive for the TI99/4A pending the completion of the SCSI's software, he says. However, it will not work with the Kodak photo multisession format player. The Kodak format is "like Beta vs. VHS" he says. "The problem is that no one but Kodak is supporting it, and then it is being used only for photos."

Crichton says he has several other projects down the line. The next one he hopes to complete is an audio CD project for the TI.

Whether Geneve software will be added to the CD-ROM depends on the customer demand, Crichton notes.

Crichton says he will not cash any checks until he ships the merchandise.

Crichton says he has several other projects down the line. The next one he hopes to complete is an audio CD project for the TI.

"I wanted to do something for the guy who just has a console and cassette player," he says.

He notes that keeping track of programs on cassette tapes is difficult and entails watching a tape counter. In CD format, using the A and B channels, the programs can be found as on any CD he says. The price depends on the media cost, he says, but estimates that the audio CD will cost "around $25."

For further information, or to order, contact Crichton at 350 Marcella Way, Millbrae, CA 94030.

FEEDBACK

Drawing challenge

Bruce Harrison's article, "Drawing a Straight Line," (April 1995 MICROpendium)was fascinating, especially Bresenham's algorithm. BRESH4 was a real challenge. I fiddled with it, simplified the variable names, removed a couple of unnecessary things (line 80 and INT() in line 50), added a display of the "error terms," and revised it until it would accept the input of any two X,Y coordinate positions and draw a line connecting them in either direction. It still worked perfectly, drawing in the best pixels between two points. After a good many hours I have to join Bruce: I can see it work but do not really understand how.

Someone should try to locate Bresenham and find out from him whether he invented a new and clever arithmetic or just fiddled with a promising approach until it worked. In either case my hat is off to him, a very clever guy indeed, and my thanks to Bruce, also a very clever guy, who put the algorithm to work for the TI.

Has anyone figured it out? Perhaps Bruce has by now since I presume that he wrote Part 46 several months ago.

John Bull
Knoxville, Tennessee
bulljh@delphi.com

Send your letters and comments to *MICROpendium Feedback*, P.O. Box 1343, Round Rock, TX 78680.

History of the TI — Part 1

The computer world is a different place after 11 years

By BILL GASKILL

It's been more than 11 years since the last TI-99/4A Home Computer "rolled" off of the assembly lines in Lubbock, Texas and several years since anyone in the media has talked about or even made mention of Atari, Coleco, Commodore, Mattel or any of the other players in what used to be called the Home Computer market.

Things have changed so drastically in the last eight years that few of the early money-makers have been able to keep up. Where the home computer of the early 1980s meant a machine with anywhere from 4K to 48K of RAM running under an 8-bit CPU with the ability to support sound and maybe speech, Home Computers of the 1990s refer mainly to a 486 multimedia PC or a Macintosh that can do anything from play music CDs to show video clips.

The PC Clone explosion that started in mid-1985 has steamrolled right over the competition, virtually wiping out any chance for the Atari STs and Commodore Amigas that showed so much promise. But even the PC clone players have changed. Where the original players in the game were relative unknowns like Blue Chip, Leading Edge, Vendex and other firms headquartered outside the U.S., many of the survivors over the last few years have ended up being inside the U.S. companies. The best players in the game today are firms like IBM, Dell, Compaq, Gateway, AST and the like, all of which are companies headquartered in the U.S.

Only five or six years ago computers like the Apple IIGS and even the Apple IIc were viable options for home computing. The same could be said of the Atari ST and Commodore Amiga computers. Today they are but little-known relics of yesterday's computer industry. Loyal owners of these computers may be angered by my presumptuousness, but visit a bookstore one of these days and tell me there is anything on the book or magazine shelves for Amigas, Ataris, Commodore 64s, Tandys or the non-Macintosh computers from Apple. There just isn't. I went to several book stores and found only two magazines for the Amiga published in the U.S., and a third that is published in Great Britain. I found absolutely nothing for any of the computers that you and I grew up with, in what I call the "Home Computer Era". Gone are all of the great magazines like *InCider* for Apple, *PCM* for Tandy, *Compute! Magazine*, which admittedly went through a real identity crisis before finally folding, *Commodore Powerplay*, *Commodore Microcomputers*, *RUN* and some other wonderful resources for the computing public. But the magazines are gone because the computers are gone, maybe not from our hearts, but certainly from the retail shelves of America.

Today, the typical "Home Computer" is a $1500-$2000 machine that sports multimedia capability, which means if it is of the PC variety, it might be a Compaq Presario, a Gateway 2000 Family PC, the Packard Bell 486SX Multimedia setup, Insight Multimedia System, and AST Advantage! Adventure or one of several others that have hit the shelves in that last couple of years.

All of these machines are at least 486SX CPU chips with 170 or more megabytes of space on their hard disks, 4 megabytes or more of RAM, a CD-ROM drive, a SoundBlaster or compatible sound card & speakers and, in many cases, a fax/modem. On the Macintosh side it would be the Apple Mac Performa series, which runs in the $1795 area at places like Sam's Club, or a Pace outlet.

A new genre of upscale magazines has begun to appear to go with the new machines. From the Ziff-Davis publishing empire, which is just about the Microsoft of computer magazines, meaning if they can't beat you in the market place, they just buy you, comes *Computer Life*, and from CMP Publications out of Manhasset, New York, comes *Home PC*. Both magazines are aimed strictly at the home user, and both do a pretty good job of reporting. But in my opinion, both could take a lesson from Robert Lock and Richard Mansfield of *Compute! Magazine* circa 1983 when it comes to learning how to lay out their magazines. Either my eyesight is failing me at my crusty old age of 46, or their layouts are just really cluttered with glitz.

PC Novice is another magazine that is aimed at the home user, but it has been around for a couple of years (since 1992 to be precise) and its editorial staff still believe there are people in this country who use IBM PCs, PC/XTs, AT&T PC 6300s, 80286 CompuAdds and the like to meet their computing needs. Imagine that?

There is a home computing segment of the market out there today, and it has enjoyed a tremendous resurgence in growth over what it has been over the last few years. However, what has surfaced in the 1990s home computing market is not the same market that you and I grew up in some 10-15 years ago. We are 10 to 15 years older and the home computing market has continued to stay young. When we got into home computing, a lot of it was as a result of the media hype over our kids being left behind if they didn't become computer literate and learn the ways of the electronic future. Well those same kids are young adults today, demanding their own brand of electronic tools to move into the next century.

The computers that we still take our enjoyment from are as old, archaic and foreign to youngsters of the 1990s as the Beatles. Home Computing 1980s style is dead, as is the Home Computing era of the 1980s. But that doesn't mean that we're dead, nor that we've stopped having fun using our machines and being productive with them. Many of the electronic dinosaurs from the "first Home Computer Era" still have a place in this world, even if it is only in our hearts.

THE BIRTH OF THE MICROCOMPUTER INDUSTRY

1939: Stanford University graduates William Hewlett and David Packard create the Hewlett-Packard Electronics Company in Palo Alto, CA, thus giving birth to what would become known as the Silicon Valley in northern California's Santa Clara valley.

1943: Nolan Bushnell is born in Clearfield, Utah. In 1972, at the age of 29, he will become the inventor of the world's first commercial video game.

1947: Bell Labs engineers John Bardeen, Walter Brattain and

(See Page 7)

HISTORY OF THE TI —

(Continued from Page 6)

William Shockley invent the transistor, which paves the way for the creation of smaller computers.

1954: William Shockley moves to Palo Alto, California, where his mother lives, and sets up Shockley Semiconductor, with money from the Nobel Prize he was awarded in 1947 for co-inventing the transistor. He recruits some of the best telectronics engineers in the country and lures them to the Silicon Valley, thereby planting the seed for the creation of northern California as the semiconductor center of the US.

1955: IBM becomes the first computer manufacturer to offer plug-in peripherals for their computers. Although the computers are of the mainframe type, the concept will catch on and become an integral part of microcomputer technology.

1957: Fairchild Semiconductor is formed by eight of William Shockley's best engineers. Some 19 years later Fairchild will become the first company to produce a full-color home video machine that uses replaceable ROM cartridges.

1959: Texas Instruments releases the first integrated circuit after its engineers figure out how to put more than one transistor on the same material and connect them without wires.

1964: John G. Kemeny and Thomas E. Kurtz develop the BASIC programming language at Dartmouth College. BASIC will become a mainstay in the microcomputer world.

1969: Intel, then a one-year old company, releases a 1K-bit RAM chip, which is the largest amount of RAM ever put on an integrated circuit up to that time.

• On June 23, IBM announces that it will no longer sell its mainframe computers bundled with software. Software will now have to be purchased separately. The effect of this announcement is to create a software industry that heretofore did not exist. Software sales will rise appreciably over the next 10-11 years, but will skyrocket with the introduction of the IBM PC in 1981.

1972: Intel introduces the 8008 chip in April 1972. It becomes the first 8-bit microprocessor to hit the market.

• Nolan Bushnell founds Atari and ships the Pong game.

1973: The first "mini" floppy disk is introduced.

1974: Intel introduces the 8080 chip in April 1974. The 8080 is the first microprocessor capable of addressing 64K bytes of memory.

•Texas Instruments releases the TMS 1000 4-bit chip. It becomes an immediate success as over 100 million are sold for use in video games, microwave ovens, calculators and other electronics products.

• In an article appearing in the July 1974 issue of Radio Electronics, author Jonathan Titus tells readers how to build the Mark 8 "personal minicomputer."

• Motorola begins work on the M6800 chip, designed by Chuck Peddle. Peddle would later leave Motorola to join MOS Technology, the creators of the 6502 chip. Peddle ultimately became Commodore's Systems Division Director, responsible for the release of the PET 2001 in October 1977, after Commodore acquired MOS Technology in order to have its own chip source.

• Naval Post-graduate School instructor Gary Kildall creates a new operating system for Intel's 8080 microprocessor called CP/M, an acronym for Control Program for Microcomputers. It sells for $70.

• Creative Computing magazine is founded by David H. Ahl in Morristown, New Jersey.

• Brian W. Kernighan and Dennis M. Ritchie of Bell Labs develop the C programming language.

1975: Texas Instruments introduces the TMS 9900 microprocessor, the first 16-bit chip on the market, but it does not sell.

• Micro Instrumentation and Telemetry Systems, a company founded by Ed Roberts as a vehicle for supporting his experiments in electronics, introduces the MITS Altair 8800 microcomputer in January. MITS becomes the first company or corporate venture into microcomputers for sale to the general public and the Altair becomes the first microcomputer to have software written for it by third-party programmers. Its open bus architecture also allows people to begin making hardware peripherals, making it the first microcomputer to also have third-party hardware add-ons created for it. The whole Altair kit, including the 8080 processor, motherboard, power supply, and 256 bytes of memory sold for $395.

• MOS Technology introduces the 6501 microprocessor, a short-lived predecessor to the famous 6502 that would power the Apple, Atari and Commodore machines from their introduction to their obselescence.

• Byte Magazine publishes its first issue in September.

• Bill Godbout and George Morrow (who would later build the Morrow Computer) build the first 16-bit computer with RAM and a built-in cassette interface. An advertisement for the unnamed computer appears in the first issue of Byte Magazine, but not one of the computers is sold.

1976: Zilog, a computer chip company which is founded by former Intel employee Federico Faggin, introduces the Z80 microprocessor.

• Shugart introduces a 5 1/4-inch floppy disk drive in December that sells for the unheard of price of $390. It is housed in a cast aluminum case. In 1979 the company will enter into an agreement with Matsushita of Japan to produce the now familiar sheet metal enclosed case that would retail for $125 and sell for $50 in OEM quantities. This is the same disk drive that Texas Instruments would sell to 99ers for almost $500 in 1979-83.

• Apple Computer Inc. is formed in April by Steve Jobs and Steve Wozniak.

• Texas Instruments makes the decision to produce a personal computer built around its unpopular TMS 9900 microprocessor. This is Mistake No. 1 according to Joseph Nocera, in his "Death of a Computer" article.

1977: The Radio Shack Division of Tandy Corporation and Commodore Business Machines join the new microcomputer market with introductions of the TRS-80 and PET 2001 (Personal Electronic Terminal) respectively. The TRS-80 is announced in August and the PET in October.

• Computer Shack, later known as Computerland, opens its first store in February.

• Ohio Scientific Instruments offers the first microcomputer with Microsoft BASIC in ROM.

(See Page 8)

HISTORY OF THE TI —

(Continued from Page 7)

• Axiom Corporation of Glendale, California enters the microcomputer printer market with the first low-cost electrosensitive line printer in the industry.

• The research and development process for TI's planned personal computer is in full swing and a corporate decision is made to assign the task of creating the computer to the Consumer Products Group which makes watches and hand held calculators at TI. Chief Operating Officer J. Fred Bucy decides to move the Consumer Products Group from Dallas to Lubbock, Texas, which is only 29 miles from his home town of Tahoca. This is Mistake No. 2 according to Joseph Nocera.

1978: The Plato computer aided instruction system is developed at the University of Illinois. Control Data Corporation would license these applications to Texas Instruments late in 1983, but by then, the fate of the Home Computer was already sealed.

• Machine and operating system-independent UCSD Pascal is released by the Regents of the University of California at San Diego for $200.

• In March, Texas Instruments begins trying to recruit personal computer specialists by running full page ads entitled "Your Experience with personal computers is going to open an unlimited career at TI" in trade publications. The ads seek qualified applicants for Personal Computer Product Marketing Managers, Systems Programmers, Digital Design Engineers, Product Design Engineers, Application Software Specialists and Marketing Support Engineers. The recruitment efforts are largely unsuccessful when potential applicants discover the job is in Lubbock, Texas, rather than close to the center of the microcomputer industry, whic northern California's Silicon Valley, situated only an hour's drive from San Francisco.

• In April, Texas Instruments releases a recreational Solid State Software Leisure Library module for the TI58 and 59 programmable calculators, coining and trademarking the term Solid State Software.

• Intel introduces the 8086 microprocessor.

• In August MICROpro releases Seymour Rubenstein's WordMaster word processor, which is the predecessor to WordStar.

• Illinois residents Ward Christensen and Randy Suess create the first microcomputer bulletin board system, conceived, designed, built, programmed, tested and installed in the 30 day period between January 16th and February 16th 1978.

• The $895 Exidy Sorcerer is released in October by Exidy Computers of Sunnyvale, California. The machine sports 8K RAM, a 64 column by 30 row screen and the ability to use plug-in modules which are the size of 8-track tapes. The Sorcerer appears to be the first "Home Computer" to support ROM cartridge use.

• In December Axiom Corporation introduces the EX-801 printer and EX-820 printer/plotter for $495 and $795 respectively. Both have available interfaces for the Apple II, TRS-80, PET and Exidy personal computers.

• Epson introduces the MX-80 dot matrix printer, shocking the industry with its low price and high performance.

• More than 14 million microprocessors are manufactured year's end, with the 8-bit 6502 chip and TI's 4-bit TMS 1000 chi leading the pack.

NEWSBYTES

M.U.N.C.H. sets TI Fall Faire

M.U.N.C.H. (Massachusetts Users of the Ninety-Nine Computer Hobbyists) has scheduled a TI New England Fall Faire Oct. 14 at the Emanuel Lutheran Church, 200 Greenwood St., Worcester, Massachusetts.

Vendor tables are $15 for a six-foot table. An additional table (if available) will be $10, according to James W. Cox of the group. Vendors will also receive free display advertising in the program guide for the event.

Set-up time begins at 8:45 a.m. Oct. 14. Faire hours are 10 a.m.-4 p.m. Admission to the event is free and refreshments will be served.

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

Orphan BBS upgrades

Harold Mayo, sysop of the Orphanage BBS in Sperry, Oklahoma, says he has upgraded from the 1200-baud Avatex to a 14,400-baud U.S. Robotics and from a standard P-box to a hard and floppy disk controller with two drives, giving the board a total of 40 meg of storage along with a 1.5 meg Horizon card. He says he also uses four DSDD floppy drives.

The board has operated for four years, using S&T software by Tim Tesch and Scott Stasiowski. Mayo says he believes the board to by the only 100 percent TI/Geneve BBS in Oklahoma.

Phone number for the BBS is (918) 288-6708.

Competition closes TI store

Ted Kieper has announced the closing of the TI-oriented store he has operated since 1982 as of June 26.

"I have enjoyed my association with the TI, but it is time to move on," he says.

He says that mail order will still be available at his 800 number. He says either he will continue selling TI products, or he will transfer his entire inventory, including rights to his 800 number, to a buyer for $2,000. For information, call 800-471-1600 or write Competition Computer, 2219 S. Muskego, Milwaukee, WI 53215. Fax 414-672-8977.

THE ART OF ASSEMBLY — PART 48

Another use for CALL FILES

By BRUCE HARRISON

Forty-eight! This can't be! That means we've been turning out these columns for four years! We keep wondering where the trail will end, but just when it seems we've exhausted all the possible subjects, one of our readers comes up with a need we find impossible to ignore. This time it was Ian J. Howle, of Seattle, Washington. Mr. Howle needed some help with bitmap, and in the course of our correspondence he mentioned that he couldn't find any simple way to load TI-Artist pictures. Another challenge was born.

OUR CONFESSION

TI-Artist is a very popular commercial product which we've never owned or used. We're told that the program itself is excellent, but the documentation leaves something to be desired. That's not a fact we can verify, but more than one friend has made that assertion. Our own knowledge, limited as it is, comes from two sources: Harry Wilhelm provided us with some of the salient facts, such as file formats and sizes. We then obtained a copy of Notung's Disk of the Old West, which has some truly excellent examples of TI-Artist pictures made by our friend Ken Gilliland. Armed with this thin veneer of knowledge, we set some goals for ourselves. First, we made our Drawing and Video Titler programs capable of using TI-Artist pictures as inputs. That success meant we understood the files and their relationship to the VDP in bitmap mode.

BUT WHAT IF...

In both the Drawing and Titler cases, we brought the contents of the TI-Artist pictures into the 32K RAM through a DSRLNK and VMBR process, then put the VDP into bitmap and displayed the picture by writing it from the 32K into the VDP again. Mr. Howle's question was how could we load TI-Artist pictures without tying up large chunks of the 32K RAM. Could we perhaps load the files directly into the correct places in VDP RAM directly from the disk, so that only the VDP RAM would ever need to store those files' contents? First, we have to realize that the TI-Artist picture may be one or two 25-sector files. The essential one is the file ending with _P. For example, on Disk of the Old West, there's a set of two files that comprise the Menu, as a TI-Artist picture. They are named MENU_P and MENU_C. The first is just the "black and white" content, while the second contains the color information. Most of the pictures on that disk don't have the _C file, so they have no color portion.

Now let's look for a moment at how we've set up VDP RAM for bitmap operation in our Drawing and Titler programs. The mapping (all numbers in hex) looks like this:

VDP Addresses Use
0000 - 17FF Pattern descriptor table
1800 - 1FFF Screen Image Table
2000 - 37FF Color Table
3800 - Sprite Attr Table, etc.

There would be a conflict if we just tried file operations with this setup of VDP RAM, because the block reserved for the DSR's use begins at >37D8, and that would be within the bitmap's Color Table area. Thus we needed a way to move the DSR reserved block to an address above >3800. CALL FILES(1) does that, placing the reserved area for one file at >3BE4. Since we don't use sprites, this would be just fine. The program never has to deal with more than one file at a time, so CALL FILES(1) met all our criteria.

THIS MONTH'S SIDEBAR

This is a complete program, ready to be assembled. It uses some subroutines from our earlier work, including the SETBM and SETGM routines to get us into and out of bitmap mode, the CRSIN routine to accept file name input, etc. If you get MICROpendium on disk, you can just assemble the file called SIDEBAR48 directly to an object file, then run it under E/A Option-3 or from Funnelweb's Loader. The program occupies only 1,553 bytes in high memory. It uses another block of 776 bytes at >2678 in low memory to stash graphics mode character definitions. The program starts with some "housekeeping" to set up the normal screen conditions. After the colors are set, we capture the character definitions from VDP RAM, and stash them away in low memory at >2678. This makes it easier to get back to Graphics mode when we need to.

Now, using essentially the same technique used in last month's column, we perform a CALL FILES(1). After that, the address kept at >8370 will be >3BE3. We'll use that later, but first we'll get a file name from the user. The file name entered will appear in memory at SPABDT+9, as a string. For TI-Artist pictures, the main file name's last two characters will be _P. In case there's a corresponding color file, we make a copy of the root file name with the _C suffix at SPABD2+9. Once that's done, we put the VDP into bitmap mode by BL @SETBM. Now it's time to go and get that _P file direct from the disk into the Pattern Descriptor Table for bitmap mode.

Notice that the data line SPABDT has zero in its second word. That, you'll recall, is the buffer location for a LOAD operation. Thus the Device Service Routine will place what it finds in the described file into VDP RAM starting at 0, and that's where the bitmap Pattern Descriptor Table is located.

But of course we have to put our PAB data in VDP someplace, then use DSRLNK to access the file. To do that, we first set R1 to point at SPABDT, get the length of the file name from SPABDT+8 into R2, add 10, then get that number from >8370 into R0. Now by subtracting R2 from R0, we will place the PAB into VDP so that it won't overlap the area reserved for the DSR's use. Thus the PAB ends at >3BE3, and the reserved area for DSR operations begins at the next byte, >3BE4. We do our BLWP @DSRLNK,

(See Page 10)

THE ART OF ASSEMBLY —

(Continue from Page 9)

with DATA 8, and the file gets put right where we can see it. When this is done, you'll see the picture come onto the screen sector by sector. After the BLWP to DSRLNK, we perform a JNE LODOK, so that if the file was found the program will jump ahead. If the file was not found, we report the error at the bottom of the screen, then wait for a key press.

If the file loaded correctly, we move on to the code at LODOK, which sets up to access the _C file, if that exists. The PAB Data at SPABD2 has >2000 in its second word, so that the content of the _C file will load directly into the VDP at >2000, which is the color table for bitmap mode. Here we don't have to trap any error, because if the file is not found, the color table will be left black on white. If the file is found, its contents will replace the previous content of the color table, which is of course the desired result.

Now the picture is on-screen, with or without color, and can be viewed. The program keeps checking the keyboard until the user presses a key. When a key gets pressed, the program uses CPDT to clear the picture away, then goes back to graphics mode through SETGM. If Function-9 was pressed, the program will jump to EXIT, else it will go back to LOAD. At that prompt, you can either type a file name or press Function-9 to exit.

The EXIT routine puts a "3" into the CALL FILES data at PABDT+9, then executes a CALL FILES to set things back to the default condition before branching back to the GPL interpreter. That's all for this month. SIDEBAR48 can be assembled into an object file, and run from Option 3 with the program name START. You supply the TI-Artist Pictures, we'll put them on-screen. Once again next month's topic will be a surprise. See you then.

SIDEBAR 48

```
1   * SIDEBAR 48
2   * A COMPLETE PROGRAM
3   * FOR SHOWING TI-ARTIST PICTURES
4   * PUBLIC DOMAIN
5   * CODE BY B. Harrison
6           DEF     START           ENTRY POINT
7           REF     VMBW,VWTR,VSBW,VSBR,VMBR
8           REF     KSCAN,DSRLNK
9   *
10  * REQUIRED EQUATES
11  *
12  GPLWS   EQU     >83E0           GPL WORKSPACE
13  KEYADR  EQU     >8374           KEY-UNIT
14  KEYVAL  EQU     >8375           KEY VALUE
15  STATUS  EQU     >837C           GPL STATUS
16  PAB     EQU     >1000           PER. ACC. BLK
17  CHRTBL  EQU     >2678           CHR DEF STASH
18  CALPNT  EQU     >832C           PNTR FOR CALL
19  PABPNT  EQU     >8356           FOR DSRLNK
20  *
21  * MAIN CODE SECTION
22  *
23  START   LWPI    WS              LOAD WORKSPACE
24          LI      R0,>0733        SCREEN GREEN
25          BLWP    @VWTR           WRITE VDP REG 7
26          LI      R0,>380         COLOR TABLE
27          LI      R1,SAVCLR       OUR COLORS
28          LI      R2,32           32 BYTES
29          BLWP    @VMBW           WRITE COLORS
30          LI      R0,31*8+>800 EDGE CHARACTER
31          LI      R1,SPCURS       BOX PATTERN
32          LI      R2,8            EIGHT BYTES
33          BLWP    @VMBW           WRITE THAT
34          LI      R0,>8F0         CHARACTER 30
35          LI      R1,CHRTBL       BUFFER STORAGE
36          LI      R2,>308         CHR 30 THRU 126
37          BLWP    @VMBR           STASH DEFS
38          CLR     @KEYADR         CLEAR KEY-UNIT
39  * FOLLOWING DOES A CALL FILES(1)
40          LI      R0,>3100        SET '1'
41          MOVB    R0,@PABDT+9     IN PAB DATA
42          LI      R0,PAB          POINT AT PAB
43          LI      R1,PABDT        AND PAB DATA
44          LI      R2,ENDPDT-PABDT     LENGTH
45          BLWP    @VMBW           WRITE TO VDP
46          MOV     R0,@PABPNT      R0 TO >8356
47          MOV     R0,@CALPNT      AND TO >832C
48          BLWP    @DSRLNK         USE DSR LINK
49          DATA    >A              DATA FOR "CALL"
50  * FOLLOWING PUTS PROMPT ON SCREEN
51  LOAD    BL      @CLS            CLEAR SCREEN
52          LI      R0,3*32+10      ROW 4, COL 11
53          LI      R1,FILIN2       PROMPT STRING
54          BL      @DISSTR         DISPLAY
55  LOADFN  LI      R0,10*32+2      ROW 11, COL 3
56          LI      R4,28           28 CHARS
57          BL      @CRSIN          INPUT ROUTINE
58          DATA    SPABDT+9        IN PAB DATA
59          CI      R8,15           FUNCTION-9?
60          JNE     NAMOK           IF NOT, JUMP
61          B       @EXIT           ELSE EXIT
62  * FOLLOWING SETS UP FILE NAME WITH _C
63  NAMOK   LI      R9,SPABDT+8     NAME LENGTH WRD
64          LI      R10,SPABD2+8 SECOND PAB
65          MOV     *R9,*R10+       MOVE THE WORD
66          MOV     *R9+,R4         LENGTH INTO R4
67          JEQ     LOAD            IF ZERO, BACK
68          DECT    R4              REDUCE R4 BY 2
69          JLT     LOAD            IF < ZERO, BACK
70  MOVNAM  MOVB    *R9+,*R10+      MOVE A BYTE
71          DEC     R4              DEC COUNT
72          JNE     MOVNAM          NOT ZERO, RPT
73          LI      R9,UNDC         _C TEXT
74  MOVUDC  MOVB    *R9+,*R10+      MOVE THE
75          MOVB    *R9,*R10        AND THE C
76          BL      @SETBM          BIT-MAP MODE
77  * FOLLOWING ACCESSES THE _P FILE, LOADS
78  * DIRECTLY TO THE VDP STARTING AT >0000
79          LI      R1,SPABDT       PAB DATA FOR _P FILE
80          MOV     @8(R1),R2       LENGTH INTO R2
81          AI      R2,10           10 FIXED DATA
82          MOV     @>8370,R0       HIGH VDP ADDR
83          S       R2,R0           SUBTRACT LENGTH
84          BLWP    @VMBW           WRITE TO VDP
85          AI      R0,9            ADD 9 TO R0
86          MOV     R0,@PABPNT      PUT AT >8356
87          BLWP    @DSRLNK         USE DSR LINK
88          DATA    8               FOR FILE ACCESS
89          JNE     LODOK           IF NE, LOAD OK
90          BL      @SETGM          ELSE GRAPHICS
91          BL      @CLS            CLEAR SCREEN
92          LI      R1,LERMSG       "LOAD ERROR"
93          LI      R0,21*32+4      ROW 22, COL 5
94          BL      @DISSTR         DISPLAY MSG
95          LI      R0,23*32+4      ROW 24, COL 5
96          LI      R1,PAK          PRESS ANY KEY
97          BL      @DISSTR         DISPLAY THAT
98          BL      @KEY            GET KEYSTROKE
99          JMP     LOAD            BACK TO PROMPT
100 * FOLLOWING ATTEMPTS TO LOAD _C FILE
101 * DIRECTLY TO THE COLOR TABLE IN VDP
102 LODOK   AI      R0,-9           SUBTR 9 FROM R0
103         LI      R1,SPABD2       _C PAB DATA
```

(See Page 11)

THE ART OF ASSEMBLY —

(Continued from Page 10)

```
104         BLWP   @VMBW          WRITE THAT
105         AI     R0,9           ADD THE 9 BACK
106         MOV    R0,@PABPNT     PUT AT >8356
107         BLWP   @DSRLNK        USE DSR LINK
108         DATA   8              FOR LOADING
109         BL     @KEY           SCAN KEYBOARD
110         BL     @CPDT          CLEAR BIT-MAP
111         BL     @SETGM         SET TO GRAPHICS MODE
112         MOV    @KEYADR,R8     KEY INTO R8
113         CI     R8,15          FUNCTION-9?
114         JEQ    EXIT           IF SO, EXIT
115         B      @LOAD          ELSE TO PROMPT
116 * FOLLOWING DOES A CALL FILES(3)
117 EXIT    LI     R0,>3300       SET '3'
118         MOVB   R0,@PABDT+9    IN PAB DATA
119         LI     R0,PAB         POINT AT PAB
120         LI     R1,PABDT       AND PAB DATA
121         LI     R2,ENDPDT-PABDT    LENGTH
122         BLWP   @VMBW          WRITE TO VDP
123         MOV    R0,@PABPNT     R0 TO >8356
124         MOV    R0,@CALPNT     AND TO >832C
125         BLWP   @DSRLNK        USE DSR LINK
126         DATA   >A             DATA FOR "CALL"
127         LWPI   GPLWS          GPL WORKSPACE
128         B      @>6A           GPL INTERPRETER
129 *
130 * SUBROUTINES FOR IN AND OUT OF BIT-MAP
131 *
132 * FOLLOWING SETS VDP INTO BIT-MAP MODE
133 *
134 SETBM LI        R0,>206        REGISTER 2
135         BLWP   @VWTR          SIT TO >1800
            LI     R0,>403        REG. 4
            BLWP   @VWTR          PDT TO >0000
138         LI     R0,>3FF        REG. 3
139         BLWP   @VWTR          CT TO >2000
140         LI     R0,>607        REG 6
141         BLWP   @VWTR          SDT to >3800
142         LI     R0,>570        REG 5
143         BLWP   @VWTR          SAT to >3800
144         LI     R0,>58         SIT AT >1800
145         MOVB   R0,@>8C02      LOW BYTE ADDR
146         SWPB   R0             SWAP R0
147         MOVB   R0,@>8C02      HIGH BYTE ADDR
148         LI     R0,3           THREE TABLES
149         CLR    R1             START WITH ZERO
150 SIT     MOVB   R1,@>8C00      WRITE TO VDP
151         AI     R1,>100        INC HIGH BYTE
152         JNE    SIT            NOT ZERO, RPT
153         DEC    R0             ELSE DEC COUNT
154         JNE    SIT            NOT ZERO, RPT
155         MOV    R11,R14        STASH RET. ADDR
156         BL     @CPDT          CLEAR PDT
157         LI     R0,>3800       SPRITE 0
158         LI     R1,>D000       USE >D0, DELETE
159         BLWP   @VSBW          ALL SPRITES
160         CLR    @>837A         NO MOTION
161         LI     R0,2           SET TO WRITE 2 TO VDP
162 REGISTER ZERO
163         BLWP   @VWTR          VDP TO M3 MODE (BIT MAP)
164         B      *R14           RETURN
165 CPDT    LI     R0,>60         CT AT >2000
166         MOVB   R0,@>8C02      LOW BYTE ADDR
167         SWPB   R0             SWAP R0
168         MOVB   R0,@>8C02      HIGH BYTE ADDR
169         LI     R0,>1800       >1800 BYTES
            LI     R1,>1F00       BLACK ON WHITE
    CT      MOVB   R1,@>8C00      WRITE ONE BYTE
172         DEC    R0             DEC COUNT
173         JNE    CT             NOT ZERO, RPT
174         LI     R0,>40         PDT AT >0000
175         MOVB   R0,@>8C02      LOW BYTE ADDR
176         SWPB   R0             SWAP
177         MOVB   R0,@>8C02      HIGH BYTE ADDR
178         LI     R0,>1800       >1800 BYTES
179         CLR    R1             ALL ZEROS
180 PDT     MOVB   R1,@>8C00      WRITE ONE
181         DEC    R0             DEC COUNT
182         JNE    PDT            NOT ZERO, RPT
183         RT
184 *
185 * FOLLOWING SETS VDP TO GRAPHICS MODE
186 *
187 SETGM   LI     R0,>1E0        VDP REG 1
188         BLWP   @VWTR          WRITE
189         LI     R0,>200        VDP REG 2
190         BLWP   @VWTR          WRITE
191         LI     R0,>401        VDP REG 4
192         BLWP   @VWTR          WRITE
193         LI     R0,>30E        VDP REG 3
194         BLWP   @VWTR          WRITE
195         LI     R0,>600        VDP REG 6
196         BLWP   @VWTR          WRITE
197         LI     R0,>506        VDP REG 5
198         BLWP   @VWTR          WRITE
199         LI     R0,>380        COLOR TABLE
200         LI     R1,SAVCLR      COLOR DATA
201         LI     R2,32          32 BYTES
202         BLWP   @VMBW          WRITE COLORS
203         LI     R0,>8F0        CURSOR
204         LI     R1,CHRTBL      STORED PATS
205         LI     R2,>308        30 THRU 126
206         BLWP   @VMBW          WRITE THOSE
207         LI     R1,>D000       CANCEL SPRITES
208         LI     R0,>300        ATTR TABLE
209         BLWP   @VSBW          WRITE THAT
210         CLR    @>837A         STOP MOTION
211         CLR    R0             VDP REG 0
212         BLWP   @VWTR          WRITE
213         RT     RETURN
214 DISSTR   MOVB   *R1+,R2       GET STRING LEN
215         SRL    R2,8           RIGHT JUSTIFY
216         JEQ    DISX           IF ZERO, SKIP
217         BLWP   @VMBW          WRITE STRING
218         A      R2,R1          ADD LENGTH
219 DISX    RT                    RETURN
220 *
221 CLS     CLR    R0             START OF VDP
222         LI     R4,24          24 ROWS
223         LI     R1,BLNKLN      SPACES
224         LI     R2,32          32 PER ROW
225 LOOP    BLWP   @VMBW          WRITE 32 SPACES
226         A      R2,R0          ADD 32 TO ADDR
227         DEC    R4             DEC ROW COUNT
228         JNE    LOOP           NOT ZERO, RPT
229         RT     ELSE RETURN
230 *
231 * FOLLOWING IS A MODIFIED VERSION
232 * OF OUR CRSIN ROUTINE AND ITS KEY
233 * INPUTS KI2 AND KI2A
234 *
235 CRSIN
236         MOV    R11,R15
237         CLR    @INSFLG
238         MOV    R0,@PGNUM
239         DEC    R0
240         MOVB   @EDGE,R1
241         BLWP   @VSBW
242         INC    R0
243         A      R4,R0
244         BLWP   @VSBW
245         MOV    R0,@ENDOC
246         S      R4,R0
247         MOV    R4,@SAV4
```

(See Page 12)

THE ART OF ASSEMBLY —

(Continued from Page 11)

```
248 CRSI0A  BLWP  @VSBR
249         MOVB  R1,@ALTKEY
250 CRSI0   BL    @CURFRC
251         BL    @KI2
252         CI    R8,9
253         JEQ   CRSRT
254         CI    R8,8
255         JEQ   CRSBK
256         CI    R8,10
257         JLT   CRSC4
258         CI    R8,15
259         JEQ   CRSDMY
260         CI    R8,13
261         JLT   CRSDMY
262 CRSC4   CI    R8,4
263         JNE   CRSENT
264         INC   @INSFLG
265         JMP   CRSI0
266 CRSENT  CB    @KEYVAL,@ENTERV
267         JEQ   CRSDMY
268         CI    R8,3
269         JEQ   CRSDEL
270         CI    R8,32
271         JLT   CRSI0
272         CI    R8,122
273         JGT   CRSI0
274         CI    R8,97
275         JLT   CRSI1
276         SB    @ANYKEY,@KEYVAL
277 CRSI1
278         MOV   @INSFLG,R1
279         JEQ   CRSI1A
280         MOVB  @ALTKEY,R1
281         BLWP  @VSBW
282         MOV   @ENDOC,R2
283         S     R0,R2
284         LI    R1,TEMSTR
285         BLWP  @VMBR
286         DEC   R2
287         JEQ   CRSI1A
288         INC   R0
289         BLWP  @VMBW
290         DEC   R0
291 CRSI1A  MOVB  @KEYVAL,R1
292         BLWP  @VSBW
293         INC   R0
294         BLWP  @VSBR
295         CB    R1,@EDGE
296         JNE   CRSI0A
297         DEC   R0
298         JMP   CRSI0A
299 CRSRT   MOVB  @ALTKEY,R1
300         BLWP  @VSBW
301         CLR   @INSFLG
302         INC   R0
303         BLWP  @VSBR
304         CB    R1,@EDGE
305         JEQ   CRSRT1
306         MOVB  R1,@ALTKEY
307         BL    @CURFRC
308         BL    @KI2A
309         CB    @KEYVAL,@RITEV
310         JEQ   CRSRT
311         CB    @KEYVAL,@NOKEY
312         JEQ   CRSRT2
313 CRSRT1  DEC   R0
314 CRSRT2  MOVB  @ONOFF,@KI2A+2
315         MOVB  @ALTKEY,R1
316         BLWP  @VSBW
317         JMP   CRSI0
318 CRSBK   MOVB  @ALTKEY,R1
319         BLWP  @VSBW
320         CLR   @INSFLG
321         DEC   R0
322         BLWP  @VSBR
323         CB    R1,@EDGE
324         JEQ   CRSBK1
325         MOVB  R1,@ALTKEY
326         BL    @CURFRC
327         BL    @KI2A
328         CB    @KEYVAL,@LEFTV
329         JEQ   CRSBK
330         CB    @KEYVAL,@NOKEY
331         JEQ   CRSRT2
332 CRSBK1  INC   R0
333         JMP   CRSRT2
334 CRSDMY  JMP   CRSIX
335 CRSDEL  MOV   R0,R7
336         CLR   @INSFLG
337         MOV   @ENDOC,R2
338         S     R0,R2
339         INC   R0
340         DEC   R2
341         JEQ   CRSD1
342         LI    R1,TEMSTR
343         BLWP  @VMBR
344         MOV   R7,R0
345         BLWP  @VMBW
346 CRSD1   MOVB  @ANYKEY,R1
347         MOV   @ENDOC,R0
348         DEC   R0
349         BLWP  @VSBW
350         MOV   R7,R0
351 CRSD0   B     @CRSI0A
352 CRSIX   MOVB  @ALTKEY,R1
353         BLWP  @VSBW
354         MOV   @ENDOC,R0
355         DEC   R0
356         MOV   @SAV4,R2
357 CRSIX1  BLWP  @VSBR
358         CB    R1,@ANYKEY
359         JNE   CRSIXX
360         DEC   R0
361         DEC   R2
362         JGT   CRSIX1
363 CRSIXX  MOV   @PGNUM,R0
364         MOV   *R15+,R1
365         SWPB  R2
366         MOVB  R2,*R1+
367         SWPB  R2
368         JEQ   CRIX
369 CRSIX2  BLWP  @VMBR
370 CRIX    B     *R15
371 *
372 KI2     CLR   @STATUS
373         BLWP  @KSCAN
374         LIMI  2
375         LIMI  0
376         DEC   R4
377         JEQ   CHNG
378         CB    @ANYKEY,@STATUS
379         JNE   KI2
380         MOV   @KEYADR,R8
381         RT
382 CHNG    CI    R1,>1E00
383         JEQ   L1
384         LI    R1,>1E00
385         BLWP  @VSBW
386         MOVB  @ONOFF,R4
387         JMP   KI2
388 L1      MOVB  @ALTKEY,R1
389         MOVB  @ONOFF+1,R4
390         BLWP  @VSBW
391         JMP   KI2
```

(See Page 13)

THE ART OF ASSEMBLY —

(Continued from Page 12)

```
392 *
393 KI2A    LI      R5,>0280
394 KI2B    CLR     @STATUS
395         BLWP    @KSCAN
396         CB      @KEYVAL,@NOKEY
397         JEQ     KI2C
398         LIMI    2
399         LIMI    0
400         DEC     R5
401         JNE     KI2B
402         MOVB    R5,@KI2A+2
403 KI2C    RT
404 *
405 CURFRC  LI      R1,>1E00
406         LI      R4,>0100
407         BLWP    @VSBW
408         RT
409 *
410 KEY     BLWP    @KSCAN          SCAN KEYBOARD
411         LIMI    2               INTERRUPTS ON
412         LIMI    0               THEN OFF
413         CB      @ANYKEY,@STATUS     KEY STRUCK?
414         JNE     KEY             NOT, SCAN AGAIN
415         RT                      ELSE RETURN
416 *
417 * END SUBROUTINES
418 * START DATA SECTION
419 *
420 WS      BSS     >20             OUR WORKSPACE
421 NOKEY   BYTE    >FF             COMPARISON BYTE
422 SAV4    DATA    0               STORAGE FOR R4
    TEMSTR  BSS     30              TEMP STRING
    LERMSG  BYTE    21              LENGTH
425         TEXT    'ERROR IN LOADING FILE'
MESSAGE
426 ANYKEY  BYTE    >20             COMPARISON BYTE
427 *
428 * SAVCLR IS JUST 32 BYTES OF >13 TO SET
429 * THE GRAPHICS COLORS TO BLACK ON GREEN
430 *
431 SAVCLR  DATA    >1313,>1313,>1313,>1313
432         DATA    >1313,>1313,>1313,>1313
433         DATA    >1313,>1313,>1313,>1313
434         DATA    >1313,>1313,>1313,>1313
435 PGNUM   DATA    0
436 ENDOC   DATA    0
437 EDGE    BYTE    31              EDGE CHARACTER
438 FILIN2  BYTE    12
439         TEXT    'INPUT NAME '
440 PAK     BYTE    25
441         TEXT    'PRESS ANY KEY TO CONTINUE'
442 ENTERV  BYTE    >0D             ENTER KEY
443 LEFTV   BYTE    >08             FUNCT-S
444 RITEV   BYTE    >09             FUNCT-D
445 UPKEY   BYTE    11              FUNCT-E
446 DNKEY   BYTE    10              FUNCT-X
447 DELKEY  BYTE    3               FUNCT-1
448 INSKEY  BYTE    4               FUNCT-2
449 INSFLG  DATA    0               INSERT ON-OFF
450 UNDC    TEXT    '_C'            UNDERLINE C
451 BLNKLN  TEXT    '               '
452 ONOFF   DATA    >0201           CURSOR ON-OFF
453 ALTKEY  BYTE    0               SAVED CHARACTER
454 SPABDT  DATA    >0500,>0000,0,>2A00,>0000
455         BSS     28              FILE NAME SPACE
456 SPABD2  DATA    >0500,>2000,0,>2A00,>0000
457         BSS     28              FILE NAME SPACE
458 SPCURS  DATA    >007E,>4242,>4242,>7E00
BOX
459 PABDT   BYTE    5               NAME LENGTH
460         TEXT    'FILES'         NAME
461         BYTE    >B7,>C8,1,'1',>B6   TOKEN (1)
462 ENDPDT  EQU     $
463 END
```

Netiquette for the Internet

Common sense helps smooth the edges of telecommunications

By TOM WILLS

(This article originally appeared in the newsletter of the SouthWest Ninety-Niners)

I'd like to take total credit for the following, but all I did was type it in and revise it where needed to soften the IBM PC references. The Net Guidelines and Netiquette was compiled by Arlene H. Rinaldi of the Florida Atlantic University in 1994.

The complete Internet User Guidelines and Netiquette is much longer and involved than is listed here. The complete thing is in a six-page pamphlet. I've included only a couple of subjects which I felt affected everyone. Information on netiquette for other subjects can be found on many locations throughout the Internet.

I decided to put this into our newsletter as surfing the net has become popular even in TI circles. It may seem to be slanted towards the IBM world, and that's because it is. However, it applies to all computer users. Please read and remember:

This applies mainly to Internet email, but can also be associated with the sending of BBS email and email on information services such as CompuServe, GEnie and Delphi.

PREFACE

The formulation of this guide was motivated by a need to develop guidelines for all Internet protocols to insure users at Florida Atlantic University realize the Internet capabilities as a resource available, with the provision that they are responsible in how they access or transmit information through the Internet (a.k.a. the Net).

It is assumed the reader has some familiarity with the terms and protocols that are referenced in this document.

Please send additions, comments, suggestions and requests for revisions to email RINALDI@ACC.FAU.EDU.

INTRODUCTION

It is essential for each user on the network to recognize his/her responsibilities in having access to vast services, sites, systems and people. The user is ultimately responsible for his/her action in accessing network services.

The "Internet" or "The Net," is not a

(See Page 14)

NETIQUETTE—

(Continued from Page 13)

single network; rather, it is a group of thousands of individual networks which have chosen to allow traffic to pass among them. The traffic sent out to the Internet may actually traverse several different networks before it reaches its destination. Therefore, users involved in this internetworking must be aware of the load placed on other participating networks.

As a user of the network, you may be allowed to access other networks and/or computer systems attached to those networks. Each network or system has its own set of policies or procedures. Actions which are routinely allowed on one network/system may be controlled, or even forbidden, on other networks. It is the user's responsibility to abide by the policies or procedures of these other networks/systems. Remember, the fact that a user can perform a particular action does not imply that he or she should take that action.

The use of the network is a privilege, not a right, which may temporarily be revoked at any time for abusive conduct. Such conduct would include:

1. The placing of unlawful information on a system;

2. The use of abusive, or otherwise objectionable, language in either public or private messages;

3. The sending of messages that are likely to result in the loss of recipients' work or systems;

4. The sending of "chain letters" or "broadcast messages" to lists or individuals and any other type of use that would cause congestion of the networks or otherwise interfere with the work of others.

(There are many other offenses that could be included in the above list, but I think you get the idea.)

Permanent revocations can result from disciplinary actions taken by a judiciary board called upon to investigate network abuses.

THE TEN COMMANDMENTS FOR COMPUTER ETHICS

1. Thou shalt not use a computer to harm other people.

2. Thou shalt not interfere with other people's computer work.

3. Thou shalt not snoop around in other people's files.

Each network or system has its own set of policies or procedures. Actions which are routinely allowed on one network/system may be controlled, or even forbidden, on other networks. It is the user's responsibility to abide by the policies or procedures of these other networks/systems.

4. Thou shalt not use a computer to steal.

5. Thou shalt not use a computer to bear false witness.

6. Thou shalt not use or copy software for which you have not paid.

7. Thou shalt not use other people's computer resources without authorization.

8. Thou shalt not appropriate other people's intellectual output.

9. Thou shalt think about the social consequences of the program you write.

10. Thou shalt use a computer in ways that show consideration and respect.

ELECTRONIC COMMUNICATIONS

(Email, LISTSERV groups, Mailing Lists and Usenet)

1. Keep paragraphs and messages short and to the point.

2. Focus on one subject per message and always include a pertinent subject title for the message. That way the user can locate the message quickly.

3. Don't use academic networks for commercial or proprietary work.

4. Include your signature at the bottom of email messages. Your signature footer should include your name, position, affiliation and Internet addresses and not exceed four lines. Optional information could include your address and phone number.

5. Capitalize words only to highlight an important point or to distinguish a title or heading. **Asterisks** surrounding a word can also be used to make a stronger point. Capitalizing whole words that are not titles is generally termed SHOUTING!

6. Limit line length and avoid control characters.

7. Follow chain of command for corresponding with superiors. For example, don't send a complaint via email directly to the "top" just because you can.

8. Be professional and careful what you say about others. Email is easily forwarded.

9. Cite all quotes, references and sources, and respect copyright and license agreements.

10. It is considered extremely rude to forward personal email to mailing lists or Usenets without the original author's permission.

11. Be careful when using sarcasm and humor. Without face-to-face communications, your joke may be viewed as criticism.

12. Abbreviations can be used; however, messages filled with them can be confusing and annoying to the reader.

Examples:

IMHO = In my humble opinion
FYI = For your information
BTW = By the way
TTYL = Talk to you later
Flame = Antagonistic criticism
:-) = Happy face for humor
:-(= Sad face
<G> = Big grin

Cor-Comp repairs have new address

David Lynch, repairer of Cor-Comp products, has a new address, according to the North County 99ers newsletter.

Contact Lynch at 1146 N. Gilbert, Anaheim, CA 92801, (714) 772-1740.

Working with the SCSI controller and hard drive

By MIKE DOANE

(Reprinted from the May 1995 South-West Ninety-Niners newsletter)

I have been working with my SCSI (small computer systems interface) controller and hard drive for the last month and a half and I thought I would give you a rundown on my progress. If you are not familiar with the terms I am using, don't despair, I will explain them throughout this article.

MDOS commands and "built-in" utilities work fine with the exception of the "delete." Jack Mathis has been looking into this and believes he may be able to come up with a solution or at least a "fix."

I have been able to create directories and subdirectories with no problem but I cannot remove them. The only way I have managed to "clean up" (remove unneeded or unnecessary files or directories) has been to reformat the hard drive (HD) and copy the files I want back to the HD. This is not exactly a *fun* job and takes some time to accomplish. I have learned not to copy or save files unless I *really, really* need/want them. These "extra" files cause no problems but they don't need to be there and they irritate me.

MDOS is the operating system used by the Geneve computer. It is a file that must be "loaded" (stored in the Geneve's memory) for the Geneve to function. It is a type of "road map" or manager. It has numerous utilities within it. You can "format" a disk, delete files, "assign" disk drives and some other "stuff" (please forgive me the use of the technical term "stuff"; I know it may not be understood by the average "joe," but there is no other way to describe its various functions) from the "prompt." Instead of loading up DM-1000 or any other disk manager, you can do these functions from the prompt. You can also "run" programs from the prompt with the use of "loaders" which are needed with the TI.

A "directory" is a "file" created to store other files. It is much like creating a "disk" within a disk. It is usually referred to as a "folder" in which you store "pages" (programs or files). The HD can be thought of as a "filing cabinet" (or in my case a "filing pile") and the "directories" as the folders within. If you want to run a program then you must "open" the cabinet (access the Hard Drive), select the "folder" (open the directory) and "take" the page (filename or program). When I "run" Page Pro the command (path) I use is "SCS1.PP.UTIL1"; the path name is broken down as "SCS1" (the name I have "assigned" to my HD — the period is a "separator" used to tell the computer where to interpret the command), "PP" (the name I call the "directory/folder" where I store my Page Pro files — the period is once again inserted to split the command), "UTIL1" (The actual name of the Page Pro program). *No* period follows this final statement.

I have been able to create directories and subdirectories with no problem but I cannot remove them. The only way I have managed to "clean up" (remove unneeded or unnecessary files or directories) has been to reformat the hard drive

These "path" names may sound complicated but they are usually included in your "menu" program and they automatically "search" for the proper drive. If you use the "BOOT" menu instead of having the menu "look" for the DSK#.PROGRAM you insert the path name you wish the computer to find. I use the "GPL" menu, but I have used BOOT. I do not use BOOT because there is no provision to "save" the changed menu back to "SCSI" (the name I have assigned to my SCSI HD). I could save the changed file to a floppy and then copy it to the HD. It was a bit awkward but it worked fine. If you are familiar with BOOT and you are comfortable with it then this is the way for you to go.

Telco works perfectly from the SCSI. You need to "install" the program and not just copy the files to the HD. You can hold down the space bar while it is loading and it will ask you for the new path name. There has been a problem reported with "capturing" the log to the SCSI. I do not use this function and I do not copy any files to my HD that I do not need. This is something to use your floppy drive for. I download any files to my RAM disk and "un-arc" them there. I pick out what I need and *then* save them to the HD. Once you go through the tedious job of "cleaning up" your HD from a few months of randomly saving files you will understand.

DM1000 runs from the SCSI, but, obviously, it will not recognize the HD. It is still the best way to manage your floppy drives and RAM disks, but it is not capable of handling the HD.

Disk Utilities loads and runs from the SCSI but with the same problem as DM1000. The problem with DSKU is there is no option for selecting SCSI. The only option is for which disk number! There may be a way around this, but I have not had the time to go into this further. It is still the easiest way to do any "sector" editing on programs to make them load from the SCSI. I use it quite often and am pleased with its performance.

(See Page 16)

SCSI—

(Continued from Page 15)

Archiver works, but with the same limitations as the two previous programs. I always archive and unarchive my files from/to a floppy anyway, so this does not bother me.

Multiplan works with a few minor problems. The MP program needs to be "sector edited" to install the path name. This is no big deal, because I had to do the same thing to run the program from my HRD. MP will not save files to the "second" level of the SCSI, i.e., a "subdirectory," but it will save them to a "directory." You cannot save MP worksheet back to a subdirectory but you can save them to a separate directory. You cannot save the worksheet to "SCS1.MP.MPFILES.Filename" but you can save them to "SCS1.MPFILES.Filename"; this is a very minor problem and easy to adjust to. You can always just save and load from a floppy (which is where you should have a "backup" copy anyway!). The SCSI loads and saves the worksheets so fast it is worth the extra effort!

MY-Word (the Geneve version of TI-Writer) works fine. You need to install it instead of simply copying the files, but you can do the same thing you did with Telco. I do not know whether (ya know, I have always wondered about the spelling of whether. I thought it was spelled "wether," but I just looked it up in the dictionary and it defines wether as, "A gelded male sheep." *Ouch*! I don't think I will have any problem remembering the proper spelling from now on!) it will save the files back to the HD but I *always* save my writing files to either the Horizon RAMdisk (HRD) or to a floppy.

I use Page Pro from my HD with ease. It is nice to have *all* my Page Pro files accessible to me from *one* disk! I have *all* my PP pictures, fonts, borders *and* line characters available to me *without* swapping disks! Man, oh, man! This made the SCSI purchase worth it all by itself. The only problem I have now is remembering which picture, etc., is which. I have so many of them available that I must refer to my catalog to decide which one I want! They say that some people look for potholes in the streets of Heaven. Hmmmm, I wonder what they mean by that?

I manage my HD with Clint Pulley's excellent Directory Manager (file names "DM and DN") which came with the SCSI Utilities disks. This program allows you to manage your HD with ease. You can rename your directories, copy files from one directory to another, create directories, and, when they get the "delete" problem corrected, you will be able to "move" files from one directory to another (you can do this now, but due to the delete problem it does not remove the file from the first directory) and remove unneeded directories/files.

I have managed to get First Draft/Final Copy to work on my HD with one problem. It will not save the file I am working on to the HD nor will it "catalog" the HD. I get around this problem by using either my HRD or a floppy to save/load files from. I like to use FD/FC because of the built-in spell checker program. This program works flawlessly from the HD. The program was written by Art Gibson and was sold through Asgard Software. It was designed to be used in conjunction with the AMS memory system but it works with just a 32K memory card. It does have a few bugs but its advantages far outweigh them.

These are some of the programs I have managed to run from my SCSI HD. The only program I have completely failed with is "Picasso." I never managed to get it to run from my HRD either so I was not surprised when it did not run from my HD. It was a disappointment because I use it so much. Rod Stallard has an idea what might be wrong with it and he is going to try to find and solve the problem. There is nothing like putting someone's name in print to gently apply the heat to them, ehhh, Rod?

1995 TI FAIRS

APRIL

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

SEPTEMBER

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

OCTOBER

TI New England Fall Faire, Oct. 14, Emanual Lutheran Church, 200 Greenwood St., Worcester, Massachusetts. Contact Jim Cox, 905 Edgebrook Dr., Boylston, MA 01505 or (508) 869-2704.

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

1996 TI FAIRS

FEBRUARY

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

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

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

YOUR NAME:_____

ADDRESS:_____

(STREET) (CITY) (STATE) (ZIP)

CDROM PLAYER AND PLATFORM INFORMATION:

MY CDROM PLAYER IS MADE BY:_____MODEL#:_____

I HAVE IT CONNECTED TO: IBM/COMPATIBLE___MAC___HORIZON SCSI____

OTHER:_____(BE SPECIFIC)

IF YOU CHECKED HORIZON SCSI: IT IS CONNECTED TO:TI99/4A___9640___

I USE THE FOLLOWING OPERATING SYSTEM WITH MY CDROM PLAYER:

WINDOWS___DOS___VERSION:_____(3.1 FOR WINDOWS FOR EXAMPLE)

SYSTEM 7.X___MYARC DOS(MDOS)___VERSION_____ TI99/4A____

____I HAVE BEEN ABLE TO VIEW KODAK PHOTO CDS THAT WERE RECORDED IN 2 OR MORE SEPARATE SESSIONS

____I HAVE BEEN ABLE TO READ CD ROM XA DISKS

____MY CD ROM MANUFACTURER ASSURES ME THAT I CAN READ ISO MULTISESSION DISKS

_____DON'T KNOW(PLEASE INCLUDE YOUR CDROM PLAYERS SPEC SHEET)

LI'L TUTOR

Helping kids learn letters and numbers

By ED MACHONIS

(This program has appeared in several users group newsletters. We reprint it from the February 1995 Tic Toc, *newsletter of the Rocky Mountain 99ers of Aurora, Colorado)*

Li'l Tutor started life as a one-liner, spawned a couple of sibling one-liners, merged with them into a TIny Gram, then, flushed with success, just kept on growing. Fortunately, a publication deadline finally checked its growth. This is contrary to the way my programs usually grow.

This program is designed to help preschoolers recognize the letters of the alphabet and digits of our number system. Except for the initial menus, all keyboard entries are single key presses and an incorrect key press is ignored. Alpha lock can be in either position.

This program is designed to help preschoolers recognize the letters of the alphabet and digits of our number system.

Only a cassette recorder and a Speech Synthesizer are required, although the program will run without speech, albeit very slowly. In this event, all CALL SAY statements should be deleted to speed things up.

The initial menu allows a choice of letters or numbers. I know the program could have been rewritten to teach both at the same time, but I wouldn't relish the job of explaining to a child the difference between the number 0 and the letter O. I think it's best if letters are learned separately from numbers, as different concepts are involved.

On the next menu, Option 1 provides for the sequential display of the characters selected as double-sized stationary sprites. Tex pronounces the name of the character.

Option 2 is similar except that the characters are displayed in a random order. A time delay is introduced before Tex speaks so that the tot has a chance to name the character before Tex does. As the child grows proficient, the delay can be reduced by changing the values for "T" in line 30. Different values are used for letters and numbers because it takes Tex longer to think of the number names.

Option 3 lets the kid have at the keyboard. When a key is pressed, the character is displayed on the screen and named. Again, a pause lets the kid beat Tex to the punch.

Option 4 randomly displays and names the character on screen. If the child presses the corresponding key, Tex speaks rewarding words and a new character is displayed and named. This option is also of value for after-schoolers who have trouble locating the letter keys.

Type in the following program.

LI'LTUTOR

```
10 ********************
* by Ed Machonis *
* QB99ers Bayside NY *
**********************
!190
20 DISPLAY AT(10,1)ERASE ALL
:"1) LETTERS";:"2) NUMBERS"
:;:"CHOICE?" :: ACCEPT AT(14
,9)VALIDATE("12")SIZE(1)BEEP
:C :: RANDOMIZE !126
30 IF C=1 THEN L=65 :: H=90
:: R=26 :: T=500 :: M$="A TO
Z" :: N$="LETTERS" ELSE L=4
8 :: H=57 :: R=10 :: T=250 :
: M$="0 TO 9" :: N$="NUMBERS
" !024
40 DISPLAY AT(8,1)ERASE ALL:
"1) ";M$:;:"2) RANDOM ";N$:;
:"3) KEYBOARD ENTRY":;:"4) K
EYBOARD MATCH":;:"CHOICE?" :
: ACCEPT AT(16,9)VALIDATE("1
234")SIZE(1)BEEP:C !084
50 CALL CLEAR :: CALL SCREEN
(5):: ON C GOTO 60,70,90,110
!086
60 FOR K=L TO H :: A$=CHR$(K
):: CALL MAGNIFY(2):: CALL S
PRITE(#1,K,16,85,120):: CALL
SAY(A$):: FOR D=1 TO T :: N
EXT D :: NEXT K :: GOTO 60 !
247
70 J=INT(RND*R)+L :: IF J=K
THEN 70 ELSE K=J :: A$=CHR$(
K):: CALL MAGNIFY(2):: CALL
SPRITE(#1,K,16,85,120)!194
80 FOR D=1 TO T :: NEXT D ::
CALL SAY(A$):: GOTO 70 !169
90 CALL KEY(3,K,S):: IF K<L
OR K>H THEN 90 ELSE A$=CHR$(
K):: CALL MAGNIFY(23):: CALL
SPRITE(#1,K,16,85,120)!198
100 FOR D=1 TO T :: NEXT D :
: CALL SAY(A$):: GOTO 90 !18
9
110 FOR I=1 TO 10 :: READ SP
$(I):: NEXT I !096
120 FOR D=1 TO 200 :: NEXT D
:: J=INT(RND*R)+L :: IF J=K
THEN 120 ELSE K=J :: A$=CHR
$(K):: CALL MAGNIFY(2):: CAL
L SPRITE(#1,K,16,85,120):: C
ALL SAY(A$)!017
130 CALL KEY(3,X,S):: IF X<1
OR X<>K THEN 130 ELSE CALL
SAY(SP$(INT(RND)+1)):: GOTO
120 !071
140 DATA FINE,GOOD,VERY GOOD
,GOOD WORK,RIGHT,THAT IS COR
RECT,YOUR RIGHT,THAT IS RIGH
T,THAT IS EXACTLY RIGHT,YOUR
DOING FINE !121
```

Compress It Anyway

Extended BASIC program compresses object code

By MARK SCHAFER

(This originally ran in Bytemonger, the newsletter of the Bluegrass 99ers)

It's been a long time since there has been an Extended BASIC program in Bytemonger. Lately, I've been known for my assembly language programming, but this program has to do with assembly language.

This program takes object code and compresses it while still allowing it to be loaded by the same loaders as before. It will work on all object code created by an assembler.

"Wait a minute," you might say, "XB cannot load compressed object code." Compress It Anyway. "What if it is already compressed?" Compress It Anyway.

Technically, this program doesn't really compress. It eliminates the waste in object code to make it shorter. Compress It Anyway sounds better than Optimize It Anyway, and besides, I like the abbreviation (C.I.A.).

There are four forms of waste in assembly language object code. The names I give to theme are: excessive load address tags, line numbers, checksums and not packing. Two of these come in two forms, so you might say there are six. Let me discuss each one.

To understand excessive load address tags, you must know that object code is made up of tags, most of which are followed by one or more parameters, which are either ASCII or hex. The tags are listed on page 14 of the Editor/Assembler quick reference card. Tags 9 and A are the load address tags. They are used excessively in two different ways; 9 or A is used at the beginning of each record in the body of the code. I figured out that the loader doesn't need to be reminded what the load address is all the time. So CIA gets rid of them unless deemed necessary. Also, sometimes they are used in succession. That's like telling the loader, "Load code here; no, on second thought, load code here." CIA removes all but the last one when they're used successively.

At the end of every line in object code is a four-character line number that is not needed, so CIA gets rid of that to make room for more tags.

Near the end of every line is either a checksum tag (tag 7) or a checksum ignore tag (tag 8). The loader doesn't have to check the sum, so CIA removes these tags which makes room for another tag.

This program takes object code and compresses it while still allowing it to be loaded by the same loaders as before. It will work on all object code created by an assembler.

The most puzzling is the fact that the assembler does not put as many tags as will fit in each record. There's room in every record for one more tag. CIA corrects this. I call this process "packing." The other form of this waste is the fact that the assembler starts a new record when it reaches the end of the code before doing the tags that come at the end. This is not necessary, so CIA continues on the same record.

As it turns out, compressed object code doesn't have line numbers or checksums, so CIA can't do as well on them as it does on uncompressed object code. They still benefit from the other two processes.

The most effective among all these processes is packing. Most of the space saved will be due to packing. The least effective is line numbers. Without line numbers, there are four blank spaces at the end of almost every record. That's not enough room for another tag, since they're usually five characters long, so you might as well put a line number there. The first record, however, only has two spaces, so leaving out the line number saves you one tag on the first record. Big deal. I was hoping it would make a bigger difference than that. The chance of it saving you a record is about one chance in 16. The chance of saving a sector: one chance in 48.

That's one of the reasons why this program makes some of the processes optional. You don't have to eliminate the line numbers if you don't want to.

Now, let's talk about running it. It will ask you for the names of the input and output files, as so many other programs do. You must include the device name (DSK1, DSK2, etc.), so this gives you flexibility in file processing.

It will then give you a list of the optional processes and ask you which ones you *don't* want to use. If excessive load address tags is the only one it lists, then that means the input file is compressed. Type the letter in front of the name to *not* use that process. I did this originally, just in case there was a loader that didn't like you eliminating everything, but, so far, I haven't found one. You can just hit return to use all of them. You can also hit "Q" to go back to the file questions.

Packing is not an optional process because it would be so much trouble. Since it works, there should be no problem with that. So if you specify all the processes (in other words, don't do any of them), it will still reduce the object code because of the packing.

The program will bomb out if the input

(See Page 20)

COMPRESS IT ANYWAY—

(Continued from Page 19)

file is not a D/F 80 file (cheap program). If it is a D/F 80 file but doesn't look like object code, it will tell you so and give you the option to continue anyway. I do this because so often I've wanted to say to a program, "Do it anyway!" And since this program has the word "anyway" in its name, I felt it was appropriate to have this option.

It will print the new object code on the screen as it goes. If you don't like that, you can change line 550. If it comes to an unrecognized tag, it will tell you what it is, print the line it occurred on and give you the option to continue anyway.

When it's done, it will give you statistics, so you can measure the effect. It will tell you how many tags, records and sectors the code had before and after. The tag reduction is more a matter of interest. The record reduction will give you an idea of how much time will be saved when loaded. The sector reduction tells you how much disk space is saved.

From the studies I've done, CIA reduces compressed object code by about 10 percent and uncompressed object code be abut 25 percent. The savings aren't as big as I had hoped because compressed object code doesn't have two of the forms of waste, and eliminating the line number turned out to be so minor. But it never hurts, so you might as well Compress It Anyway!

COMPRESSIT

```
1 ! COMPRESS IT ANYWAY 10/24
/92 !120
2 ! REV 2.1.1 10/24/92 !080
3 ! BY MARK SCHAFER !165
4 ! BLUEGRASS 99 COMPUTER SO
CIETY !172
90 T$="123456789ABCFI" !061
100 DIM LA(4),NA(4)!045
110 PRINT "ENTER FILE NAMES
WITH DEVICENAMES." !165
120 INPUT "INPUT FILE NAME:
":IF$ :: IF IF$="" THEN END
ELSE IF POS(IF$,".",1)=0 THE
N 110 !168
130 INPUT "OUTPUT FILE NAME:
":OF$ :: IF OF$="" THEN 120
ELSE IF POS(OF$,".",1)=0 TH
EN 110 !210
140 IF IF$=OF$ THEN PRINT "S
ORRY, FILES ARE NOT ALLOWEDT
O MATCH." :: GOTO 120 !028
150 OPEN #1:IF$,FIXED 80,INP
UT !034
160 LINPUT #1:R$ :: K=ASC(R$
):: IF K=1 OR K=48 THEN 190
!079
170 PRINT "FIRST RECORD IS "
;R$:"THAT DOESN'T LOOK LIKE
OBJECT CODE." :: INPUT
"CONTINUE ANYWAY? ":Y$ :: IF
Y$="N" OR Y$="NO" THEN CLOS
E #1 :: GOTO 120 !096
180 INPUT "TREAT AS COMPRESS
ED? ":Y$ :: IF Y$="" THEN 17
0 ELSE IF Y$="Y" THEN K=1 EL
SE K=48 !071
190 PRINT "A EXCESSIVE LOAD
ADDR TAGS" :: IF K=48 THEN P
RINT "L LINE NUMBERS":"C CHE
CKSUMS" !029
200 PRINT :: INPUT "WHICH DO
YOU NOT WANT TO ELIMINAT
E? ":E$ :: A1=POS(E$,"A",1):
: L1=POS(E$,"L",1)AND K=48 :
: C1=POS(E$,"C",1)AND K=48 !
016
210 IF POS(E$,"Q",1)THEN CLO
SE #1 :: GOTO 120 !060
220 LA(1)=-1 :: IF K=1 THEN
B=2 :: HD=256 ELSE B=4 :: HD
=16 !209
230 P=B+10 :: CL=P-1 :: O$=S
EG$(R$,1,CL):: T,R=1 :: L=79
:: IF L1 THEN L=75 !064240
IF C1 THEN L=L-B-1 !190
250 OPEN #2:OF$,FIXED 80,OUT
PUT !142
260 TN=POS(T$,SEG$(R$,P,1),1
):: T=T+1 :: IF TN>0 THEN 28
0 !127
270 PRINT "FOUND TAG";ASC(SE
G$(R$,P,1));" IN RECORD ";R$
:: INPUT "CONTINUE ANYWAY?
":Y$ :: IF Y$="" THEN 270 EL
SE IF Y$="Y" THEN TN=11 ELSE
420 !169
280 ON TN GOTO 330,330,360,3
60,360,360,370,370,290,290,3
40,340,390,380 !236
290 LAT=TN :: K=0 :: LA$=SEG
$(R$,P+1,B):: FOR A=1 TO B :
```

(See Page 21)

COMPRESS IT ANYWAY—

(Continued from Page 20)

```
: NA(A)=ASC(SEG$(LA$,A,1))::
IF B=4 THEN IF NA(A)>57 THE
N NA(A)=NA(A)-55 ELSE NA(A)=
NA(A)-48 !116
300 IF NA(A)=LA(A)THEN K=K+1
!121
310 NEXT A :: IF K=B THEN 37
0 !020
320 FOR A=1 TO B :: LA(A)=NA
(A):: NEXT A :: IF A1=0 THEN
IF POS(T$,SEG$(R$,P+B+1,1),
1)=TN THEN 370 !127
330 A$=SEG$(R$,P,B+1):: GOSU
B 480 :: GOTO 260 !026
340 A$=SEG$(R$,P,B+1):: GOSU
B 480 :: A=B :: K=2 !141
350 LA(A)=LA(A)+K :: IF LA(A
)>=HD THEN LA(A)=LA(A)-HD ::
K=1 :: A=A-1 :: GOTO 350 EL
SE 260 !063
360 LAT=0 :: A$=SEG$(R$,P,B+
7):: GOSUB 480 :: GOTO 260 !
058
370 P=P+B+1 :: GOTO 260 !242
380 A$=SEG$(R$,P,7):: GOSUB
480 :: GOTO 260 !029
390 R=R+1 :: LINPUT #1:R$ ::
K$=SEG$(R$,1,1):: IF K$=":"
THEN 410 !120
400 P=1 :: GOTO 260 !222
410 LAT=0 :: A$="" :: GOSUB
530 :: NR=NR+1 :: GOSUB 620
:: PRINT #2:SEG$(R$,1,76)&LN
$ !248
420 PRINT USING 470:"","," OL
D","," NEW" !236
430 PRINT USING 470:"TAGS",T
,NT !052
440 PRINT USING 470:"RECORDS
",R,NR !022
450 PRINT USING 470:"SECTORS
",1-INT(-R/3),1-INT(-NR/3)!0
27
460 CLOSE #1 :: CLOSE #2 ::
END !190
470 IMAGE ####### ##### ####
# !017
480 P=P+LEN(A$)!198
490 K$=O$&A$ :: IF LEN(K$)>L
THEN 530 !013
500 O$=K$ !160
510 CL=CL+LEN(A$)!068
520 NT=NT+1 :: RETURN !199
530 IF C1 THEN 610 !213
540 O$=O$&"F" :: NT=NT+1 ::
NR=NR+1 :: IF L1 THEN GOSUB
620 :: O$=O$&RPT$(" ",75-CL)
&LN$ !153
550 PRINT #2:O$ :: PRINT O$
!077
560 IF A1 THEN 580 !181
570 O$=A$ :: CL=LEN(O$):: GO
TO 520 !245
580 IF LAT=0 THEN 570 !209
590 O$=SEG$(T$,LAT,1):: FOR
A=1 TO B :: IF B=4 THEN IF L
A(A)>9 THEN K=LA(A)+55 ELSE
K=LA(A)+48 ELSE K=LA(A)!199
600 O$=O$&CHR$(K):: NEXT A :
: CL=A :: NT=NT+1 :: GOTO 49
0 !078
610 O$=O$&"80000" :: CL=CL+B
+1 :: NT=NT+1 :: GOTO 540 !1
06
620 LN$=RPT$("0",4-LEN(STR$(
NR)))&STR$(NR):: RETURN !008
```

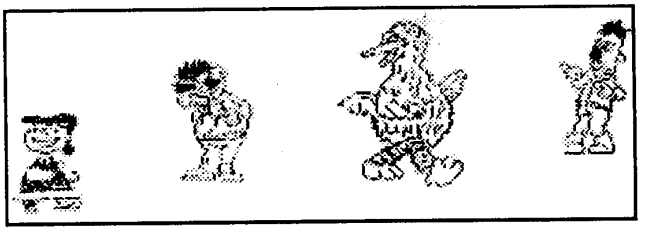
Using TI-Artist with FWB Editor v5.0

By JACQUES GROSLOUIS

(Reprinted from the May 1995 Bits, Bytes & Pixels, *newsletter of the Lima [Ohio] 99/4A Users Group)*

How do you like Lucy, Big Bird and Bert and Ernie? It has always been possible to print TI-Artist instances using TI-Writer. This was accomplished by creating a transliterate file from an instance file.

Printing of this conversion required setting the printer to PIO.CR when processing through the formatter. The above instances were processed through a conversion program and the resulting D/V80 files were imbedded in this document file. The document was printed using PIO in ALL-CHAR mode of v5.01 on an FX-80 Epson printer. In order to put instances side by side as shown, your printer must support reverse feed. A reverse feed of two inches was placed between each instance. With the FX-80 printer this is invoked by including <ESC>"j"CHR$(n) in your document where n represents a reverse feed of n/216 inch. If your printer does not reverse feed, you will have to print the document in sections and roll the paper back for each instance. The program allows you to set the tab position in the document which is produced. I usually set the tab position at 10 in the program and change the code once the document has been inserted into my final document. The FX-80 code is <ESC>"l"CHR$(n) where n sets the left margin in the range from 0 to 78. The instances above were set at tabs 10, 25, 40 and 60. You will notice that the program includes the tab setting in the file name.



There are a few limitations in the use of this program. First, the instance that you wish to convert should be less than 10 columns wide. This is necessary because when a file is saved in D/V80 format a carriage return (CR) and a line feed (LF) are placed at the end of each 80 characters in the file. This plays havoc with the use of the single density graphics mode. Since each row of the instance appears on only one line of the file when fewer than 10 columns are used the problem is avoided.

(See Page 22)

TI-ARTIST WITH FWB 5.0—

(Continued from Page 21)

A second problem is encountered when code 255 is sent to the printer in graphic mode. This should fire all eight dots in a vertical line. In Funnelweb, character code 255 is reserved for another purpose and will not print properly. My solution was to change all occurrences of 255 to 254. A side effect is that this puts a part in Ernie's hair. A similar problem occurs with code 32 which has been replaced by code 16. If anyone has a better solution, please let me know.

I have left some lines and numbers in the printing of the instances above to show the spacing that must be considered when using tab and reverse feed. The printing of Lucy begins after 5 is printed and a two-inch reverse feed is called for between each instance. I found it difficult to properly adjust the vertical spacing for the placement of instances. A bit of trial and error was called for.

Instances that are wider than nine columns can be converted and sent to the printer by setting the printer code to PIO.CALF. The program handles instances that are less than 10 columns wide differently by placing a CR at the end of each line of graphic code. This permits the use of PIO to print the line. The graphic code for instances that are wider than nine columns are ended with LF. The only problem that this creates is that if you try to print an instance that is less than 10 columns wide with PIO.CRLF it will print double spaced.

The following program was adapted from a public domain program by David Dhein distributed with PLUS! v2.0 by Jack Sughrue. My changes were to send printer codes to a disk file instead of sending transliterate codes.

TIA2FW5

```
100 ! SAVE DSK1.TIA2FW5
110 E$=CHR$(27):: Z=0 !048
120 CALL SCREEN(11):: DISPLA
Y ERASE ALL AT(1,4):"TI-ARTI
ST TO FWB v5.0": :"CONVERSIO
N PROGRAM" !104
130 DISPLAY AT(5,1):"INSTANC
E file name:" :: ACCEPT AT(5
,21)VALIDATE(UALPHA,DIGIT)SI
ZE(8):NAME$ !238
140 DISPLAY AT(7,3):"The fil
e is on drive 1" :: ACCEPT A
T(7,24)SIZE(-1)VALIDATE(DIGI
T):FD !178
150 DISPLAY AT(8,1):"Which d
rive for new file? 1" :: ACC
EPT AT(8,27)SIZE(-1)VALIDATE
(DIGIT):SD !176
160 DISPLAY AT(11,1):"Print
at Tab Position? 10" :: ACCE
PT VALIDATE(DIGIT)SIZE(-2)AT
(11,24):T :: IF T>78 THEN 16
0 !075
170 A$="DSK"&STR$(SD)&"."&NA
ME$&STR$(T)!101
180 NAME$="DSK"&STR$(FD)&"."
&NAME$&"_I" !013
190 OPEN #1:NAME$,INPUT :: O
PEN #2:A$,OUTPUT !178
200 INPUT #1:X,Y :: A=X*8 ::
IF A>255 THEN A=X*8-255 ::
Z=1 !053
210 K$=CHR$(27)&CHR$(75)&CHR
$(A)&CHR$(Z)!101
220 IF X*Y>25 THEN DISPLAY A
T(20,4):"This may take awhil
e." :: DISPLAY AT(21,4):"Ple
ase be patient..." !095
230 PRINT #2:E$&CHR$(108)&CH
R$(T);:: DISPLAY AT(14,1):"R
OW:";Y;"COL:";X !211
240 PRINT #2:E$&CHR$(65)&CHR
$(8)!222
250 FOR K=1 TO Y :: PRINT #2
:K$;!135
260 FOR L=1 TO X :: INPUT #1
:C(Y),C(6),C(5),C(4),C(3),C(
2),C(1),C(0)!251
270 FOR I=7 TO 0 STEP -1 ::
A=C(I)!037
280 FOR J=7 TO 0 STEP -1 ::
IF 2^J>A THEN 310 !109
290 A=A-2^J :: B(J)=B(J)+2^I
:: IF B(J)=255 THEN B(J)=25
4 !249
300 IF B(J)=32 THEN B(J)=16
!002
310 NEXT J :: NEXT I !065
320 FOR I=7 TO 0 STEP -1 !17
1
330 PRINT #2:CHR$(B(I));:: D
ISPLAY AT(16,1):"ROW:";K;" C
OL:";L;" Char:";B(I):: B(I)=
0 !054
340 NEXT I :: NEXT L :: IF X
>9 THEN PRINT #2:CHR$(10)ELS
E PRINT #2:CHR$(13)!005
350 NEXT K :: PRINT #2:E$&CH
R$(108)&CHR$(0);!028
360 PRINT #2:CHR$(27)&CHR$(6
5)&CHR$(12)!023
370 CLOSE #1 :: CLOSE #2 ::
DISPLAY AT(23,3):"Another (Y
/N)?N" :: ACCEPT AT(23,17)VA
LIDATE("YNyn")SIZE(-1):Q$ ::
IF Q$="Y" OR Q$="y" THEN 10
0 !062
```

> Instances that are wider than nine columns can be converted and sent to the printer by setting the printer code to PIO.CALF. The program handles instances that are less than 10 columns wide by placing a CR at the end of each line of graphic code.

MICRO-REVIEWS

Geneve SCSI, RXB v.1002 (or higher), Funnelweb v.5.2 80-Column Editor

By CHARLES GOOD

This month I am reviewing very significant new products I demonstrated at the recent Lima MUG Conference.

GENEVE SCSI by Western Horizon Technologies and Bud Mills Services.

Since the day of the Lima conference I have been using a small computer systems interface (SCSI) card and a "Slow SCSI-2" 40-meg hard drive in my Geneve system. I am delighted with its elegance, performance and low cost, and highly recommend a SCSI hard drive system to any Geneve user

The system is elegant because everything can be mounted inside the P-box. No external power supply is required. My SCSI hard drive takes so little power that it can be operated using an unmodified P-box power supply in addition to one regular half-height or two very low power half-height floppy drives controlled by a TI or other floppy drive controller. The hard drive, which is the size of a 3.5-inch floppy drive, can be mounted in the drive bay next to a half-height DSK1 using a 5.25-inch mounting frame available for about $5 at most computer stores. I chose instead to put my hard drive on a slot inside the main P-box compartment one slot away from my SCSI card. My hard drive is mounted on the back half of a TI clamshell. This is the half of the clamshell with the little wings that allow it to snugly fit into a slot in the P-box. I used the clamshell that covered my firehose 99/4A cable, which most Geneve owners should have as surplus since the firehose cable is not needed for a P-box containing a Geneve. All you need to do is drill a couple of holes in the clamshell and bolt on the hard drive. Then thread one end of a "Y" power cable out of the floppy drive compartment and connect it to the internal hard drive and connect the cable that comes with the SCSI card to the hard drive. If you have a pair of very low power floppy drives and a second "Y" power cable, you can run both floppy drives and the SCSI hard drive at the same time off of the P-box power supply. Many will find that only one floppy half height drive and the SCSI hard drive can be powered directly from the P-box.

Performance is fantastic compared to using just floppy drives or a combination of floppy drives and Horizon RAMdisks. My 40-meg drive has more than 160,000 256 -byte sectors of file storage space. Based on some timed tests I have done, using my Geneve's hard drive to load the new v5.2 Funnelweb word processor, and loading and saving a 360-sector DV80 document to and from the word processor is *as fast as or faster than* the same operations using a Horizon RAMdisk on a 99/4A system. Saving large text files is significantly faster. I really find this amazing, because I thought you couldn't get anything faster than a Horizon RAMdisk. The "MDOS v2.50S delta" Geneve operating system I am using seems full featured. I have experienced no bugs. You can copy and delete files, create and remove subdirectories, etc., using the same syntax used for hard drives controlled by a hard and floppy disk controller (HFDC). In fact, if you really want, you can have HFDC-controlled MFM hard drives and SCSI card controlled hard drives working together in the same Geneve P-box. Support software allows you to format hard drives as well as search sectors for strings edit sectors. For sector operations you have to know which sectors to edit or search or you have to search the entire hard drive. With the software I now have available you can't just put the cursor next to a file name and automatically look at the sectors of just that file.

A really nifty fairware "Directory Manager" program by Clint Pulley is included. It is full featured and user friendly, operating somewhat like Funnelweb's Disk Review. A display shows all the file names in a selected directory. From this display you can view (in ASCII or hex), copy, move, delete, rename, protect or unprotect single or groups of files. You can easily bounce back and forth between one directory and another and you can delete rename and create directories. If you move the cursor next to a text file and press "E" you can edit that file with a text editor. Clint's requested donation for this software is only $10, an excellent value!

LIMITATIONS

• You cannot *yet* use a SCSI hard drive on a "Genmod" Geneve. If you have a Memex memory expansion card with more than 512K of extra memory then you have a genmod Geneve.

• You need to have a second 32K fast RAM chip mounted on your Geneve card to use the SCSI interface card. You remove the existing 32K socketed RAM chip, purchase or make a stacked set of two 32K chips already soldered together, put this stack back in the socket and make one solder connection with a little wire. Either BMS or WHT will sell you the presoldered stacked two-chip set and will mount them for you if you send them your Geneve. The operation takes little time if you do it yourself. Many Geneve owners already have this modification.

• You can't *yet* entirely boot the SCSI version of MDOS from Cecure's PFM (programmable flash memory) flashdisk boot ROM. The 128K SCSI DOS is too big to fit on the 120K PFM boot ROM. I have software from Cecure Electronics called SCSI4PFM that lets you put most of SCSI MDOS on a PFM and boot the last 8K of MDOS from some other device. This eliminates the extra keypresses normally needed to bypass the PFM when booting SCSI MDOS from some other device like a floppy. Unfortunately, current versions of SCSI MDOS do not recognize the PFM+ flashdisk drive. This means that currently a PFM+ flashdisk is useless on a Geneve with SCSI and can't be used to contain the last 8K of SCSI MDOS. Instead you have to put this last 8K of SCSI MDOS on a floppy disk or Horizon RAM-

(See Page 24)

MICRO REVIEWS—

(Continued from Page 23)

disk. The limitations discussed in this paragraph are of concern only to those who have PFM or PFM+ installed on their Geneve.

• You can't *yet* make the SCSI drive act as DSK1. The first SCSI drive's device name is "SCS1" (that's a "one," not capital I) and cannot be reassigned to DSK-something. DSK1 emulation is supposed to be available in future versions of SCSI MDOS.

• You can't *yet* boot MDOS automatically from the hard drive. This will require the as yet uncompleted SCSI DSR EPROM. As of this writing the only way to boot SCSI MDOS is from a floppy in DSK1, from a Horizon RAMdisk, or (partially) from a PFM boot ROM.

• You can't *yet* control a SCSI floppy drive or other type of SCSI peripheral with the SCSI interface card. The ability to do this awaits further refinements of MDOS. As of now you need to have a separate floppy drive controller. On the Lima Conference videotape Mike Maksimik, author of the SCSI version of MDOS, says support for SCSI hard drives and CD-ROM drives will definitely be forthcoming.

Cost: From BMS the SCSI card with cables, MDOS and support software costs $170. A 40-meg (available storage after formatting) low power slow SCSI-2 hard drive like mine costs $100. The presoldered stack of two 32 chips with a little wire already attached at one end is $25. There is a $5 per order shipping charge. WHT has similar prices. SCSI hard drives of other sizes are available from BMS or WHT and can also be purchased on the open market. There is no extra cost software required other than the requested fairware donation for Directory Manager. The combined cost of the SCSI card and 40-meg hard drive and the extra 32K (if needed) is probably less than you would pay now for a new HFDC card alone from Cecure. Originally you had to pay about $350 for the HFDC plus the cost of a compatible MFM hard drive (most of which were slow and of relatively low capacity) plus the cost of an external power supply for the hard drive. These high costs kept me and many other 99/4A and Geneve users from enjoying the benefits of a hard drive system. Now many serious Geneve users (and soon 99/4A users as well) can afford to move up to the really massive mass storage capacity of a relatively fast hard drive.

If you (like me) are a total klutz when it comes to drilling holes and soldering then BMS (and maybe also WHT) offers a complete plug and play Geneve package deal. You send in your Geneve. BMS will mount the extra 32K and return your Geneve along with a SCSI card, cables, software, and a 40-meg drive already mounted on either a half clamshell or a 5.25-inch frame. The hard drive will come formatted, loaded with support software, tested on your very own Geneve. Just pop your Geneve and SCSI card back into your P-box, plug in a few cables, and you're ready to go! The complete package price including shipping is $325. The only thing BMS can't guarantee is that this setup will work with *two* floppy drives in your P-box because the two floppy drives and the hard drive may eat up too much power. If you hear your hard drive drop out when two floppy drives are in use you will have to disconnect the second internal floppy drive. This is what I had to do, and that is why you might want to play it safe and use the 5.25-inch frame option. This plug and play system package will definitely work with one internal half height floppy and maybe with two. Call BMS or WHT for product availability.

RXB v1002 (or higher)
by Richard Gilbertson

This product has a long history and has been reviewed in MICROpendium several times. I am reviewing it again because a significant new feature has lately been added.

RXB is an "extended Extended BASIC," with many enhancements added to the original TI Extended BASIC. It started out several years ago as "Windy XB." Then it became an enhanced version of "GRAM Kracker XB" that was sold by Miller Graphics on their Utility 1 disk for users of their "GRAM Kracker" GRAM emulator. RXB, as it is now called, has been revised and updated 1,002 times. There are about 50 XB commands and subprograms that have either been added or enhanced compared to the original TI XB, and the product is under continuous revision. In the few weeks since I started evaluating RXB for this review it has advanced from v1000 to v1002. It is probably the most enhanced of the many "extended Extended BASICs" that have appeared over the years. RXB is fully compatible with any existing XB program, even XB programs containing assembly code. I have found no exceptions to this.

Right now you need either a GRAM emulator (GRAM Kracker, Gramulator, P-GRAM or GRAM Karte) or a Geneve to run RXB. It comes on disk as a set of GRAM files along with a massive on-disk user guide and lots of useful programs that take advantage of RXB's special features. If enough people are interested in an RXB cartridge, WHT will make the cartridges. Projected cost is about $60 per cartridge, and 25 advance reservations are needed for a production run. As I understand it, WHT will not require money up front but will require a firm agreement to pay on delivery. Phone or email WHT for more details.

When you first boot RXB you have many choices, and you have to make a choice quickly. If you press the space bar RXB immediately defaults to XB command mode and bypasses the DSKx.LOAD search. If you press a number from 1 to 9, RXB will go directly to that drive number and search for LOAD on that drive. If you press the period key RXB goes to its editor/assembler screen. If you do nothing, after about 20 seconds RXB automatically begins searching all drives from 1 to 9 for an XB program call LOAD. If LOAD isn't found on any drive RXB eventually defaults to XB command mode.

The Editor/Assembler portion of RXB is exactly like the TI E/A module with the addition of a disk directory option. Internally the E/A portion of XB is coded just like the TI module, so that *all* software that runs from the TI E/A module will run from RXB. I know of no exceptions. If you bring up a disk directory you can move the cursor next to the name of a runnable assembly file and run the software directly

(See Page 25)

MICRO REVIEWS—

(Continued from Page 24)

from the disk directory.

The main new feature of RXB, and the reason for this review, is batch file capability. You can use a D/V80 text file created with a text editor to control RXB command mode. The syntax from command mode is CALL USER("DSKx.FILENAME") where FILENAME is a DV80 file. When you press <enter> FILENAME takes over complete control of RXB. Yes, it is possible to write an XB program using a text editor and then load the resulting text file into RXB and make it a runnable XB program, but this is only one of many batch file possibilities.

RXB takes the text file and reads the file, one character at a time, into the key scan of RXB command mode. You see your text appear on screen just as if you were typing on the keyboard, and the text is immediately executed. This means that *anything* that can be done in command mode from the keyboard can also be done under the control of a text file.

RXB assumes that a carriage return in a text file is the same as pressing <enter> from the keyboard. If you don't have a CR at the end of a text line RXB goes directly on to the next text line and considers it to be part of the same line of input. You can have one RXB program line with more than 500 characters. You can create an entire runnable RXB program as a single large paragraph starting with the program's one and only line number, with double colon command separators within the paragraph, and with a single carriage return placed at the end of the paragraph.

The main new feature of RXB, and the reason for this review, is batch file capability.

The RXB package comes with a USER-DEMO text file that really puts CALL USER through its paces. As you sit back and watch, RXB under the control of USER-DEMO creates runs merges saves and deletes (from disk and from memory) programs, runs TRACE, opens files and reads data into the files, and brings up disk directories to show you what is or is not currently on the disk that originally only contained the file USER-DEMO.

I wrote an RXB batch file. modified from one originally written for use with Super BASIC. My batch file is designed to automate the process of adding checksums to an XB program saved to disk under the name PROGRAM before publishing the code in MICROpendium or in a newsletter. You put my DV80 batch file saved under the name BATCH, Tom Freeman's CHECKSUM program (available on disk from user group libraries, from MICROpendium or from me as described below), and the XB file PROGRAM on a disk in DSK1. Then from RXB you enter CALL USER("DSK1.BATCH") and watch the magic. My batch file actually produces INPUT into the running CHECKSUM program. You end up with an on-disk publishable DV80 listing of PROGRAM with checksums suitable for publication in a newsletter. You also get a copy of PROGRAM saved as a runnable XB program with checksums added to each line of XB code.

Here is my Checksum batch file, complete with REM statements to tell you what is going on:

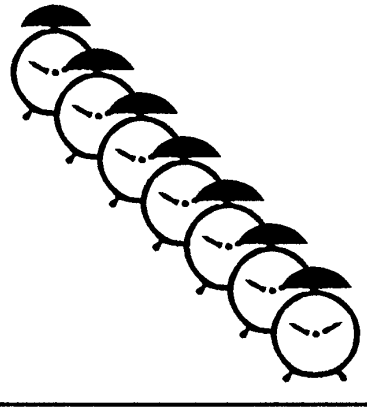
```
! Batch file using RXB v1001
to add checksums to XB PRO-
GRAM and print a listing
of PROGRAM for a newsletter.
OLD DSK1.PROGRAM
SAVE DSK1.CKSMINPUT,MERGE
RUN "DSK1.CHECKSUM"
! The next two lines add IN-
PUT into the running CHECKSUM
program
1.CKSMINPUT
1.CKSMOUTPUT
CALL NEW
MERGE DSK1.CKSMOUTPUT
! Delete unneeded file
DELETE "DSK1.CKSMOUTPUT"
! Prints PROGRAM as DV80 file
for use in newsletter
LIST "DSK1.NEWSLETTER"
```

(See Page 26)

READER TO READER

James F. Uzzell would like to hear from European Geneve users, especially those near Naples, Italy. Write him at HQ Airsouth, PSC 813, Box 105, FPO AE 09620, USA.

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



It's time to start thinking about attending your next TI fair.

MICRO REVIEWS—

(Continued from Page 25)

```
! Saves PROGRAM back to disk
with checksums added to each
line of code.
SAVE DSK1.PROGRAMCHK
! Delete unneeded file
DELETE "DSK1.CKSINPUT"
```

The only thing similar to RXB's batch file capability is Super BASIC. This is a commercial disk based program that does not need a GRAM device and runs with the regular TI XB module. Super BASIC is protected by a gizmo that plugs into the joystick port. You can easily copy the Super BASIC disk, but you have to have the physical gizmo plugged in for Super BASIC to run. Batch files written for Super BASIC should run on RXB of you add a CR to each batch file line.

My only complaint about RXB is that under certain circumstances it is not recognized by Funnelweb's Disk Review. Problems occur only if you have RXB on line and load Funnelweb using some method other than DSKx.LOAD, such as autoload from a Horizon RAMdisk or as an E/A5 program from RXB's editor assembler menu. Under these circumstances if you bring up a disk directory from Funnelweb's Disk Review, move the cursor next to a runnable XB program and press "R" to run the program, Funnelweb will respond with "XB not found." If you have a regular TI XB cartridge plugged into the console and do this Funnelweb will run the XB program. Many Funnelweb users will find this to be only a slight annoyance. The RXB author says this minor incompatibility between RXB and Funnelweb is because of a bug fix that was made in TI XB years ago in GRAM Kracker XB. RXB is based on GRAM Kracker XB.

Super Basic is commercial and may no longer be available. RXB is now *free!* RXB used to cost $25 from CADD Electronics, but when the author sent me a copy he wrote "freeware" on the envelope. I talked to Rich Gilbertson on the phone and said, "You mean 'shareware,' don't you?" He replied, "No, its freeware." I said, "You mean you don't expect people to send you money for this?" and he replied, "That's correct." Nevertheless, you should send Rich some money. Because RXB v1001+ is free, *every* Geneve user and GRAM device owner should try it out. Send some money to Rich and he will send you the very latest version. Or send me $2 and I will send you on two DSSD disks the latest RXB version I have together with on-disk copies of Checksum and my batch file.

FUNNELWEB V5.2 80-COLUMN EDITOR by Tony McGovern

This latest edition of the Funnelweb family of 80-column editors is the most flexible and has the largest text buffer of any word processor ever written for the 99/4A or Geneve. When the editor boots you get your choice of one 128K text buffer, or two 64K text buffers, or four 32K text buffers. If you use multiple text buffers you can have a different document in each buffer, up to four documents in active editable memory simultaneously. You can edit each of the documents in each of the buffers separately and you can cut and paste between the different buffers. You can block up to 22 lines of text from one document, put these lines in a separate clipboard buffer, and paste the contents of the clipboard anywhere into any of the edit buffers. There is also a View buffer (in addition to the 1-4 edit buffers). You can put text from the View buffer into the clipboard and paste this text into any of the edit buffers.

This 128K text buffer size is, by any 99/4A or Geneve standard, enormous. The edit buffer of most TI-Writer clones is about 22K. MY-Word on the Geneve has a buffer of about 50K. You can load the single 347-sector v5.2 documentation file into the 128K text buffer and still have 48 percent of the buffer free. When you invoke ShowDirectory you get a display showing the percent of the current buffer still empty and the number of text lines still available. The Funnelweb edit buffers will only accept a finite number of lines (about 4,500 for the 128k buffer) because the computer has to keep track of line numbers.

Help screens are generated from a single DV80 file that loads into memory when the editor is loaded. This help file is editable by the user using the Funnelweb editor. The editor keeps track of all the path and file names you use in a session when you LF, SF, PF, in each of the multiple edit buffers. When you do any of these operations a window pops up showing you all these file names and gives you quick access to these files by pressing a number, or you can type in the path of a new file. The program also keeps track of the names of each file currently in each text buffer, the number of lines in each buffer, and whether each buffer has been edited since its file was loaded. All this information can be quickly displayed in a pop-up window when desired.

This 80-column editor requires Geneve or a 99/4A system with an 80-column adapter. You add the files on the v5.2 upgrade disk to your Funnelweb v4.4 system disk. You need the v4.4 FW and/or LOAD files to make this editor work. To use the multiple buffers or the single 128K buffer you need 192K of VDP RAM. If the editor senses that your system has only 128K of VDP RAM (such as an unmodified Geneve) it will default to a single 64K edit buffer. If you haven't already done so, this Funnelweb editor gives you a good reason to have Cecure Electronics add the needed additional VDP RAM to your Geneve. The editor comes with command line language files and character sets for American and British English, French, German and Swedish. Italian and Spanish may be added later if consumers help Tony McGovern to do so. The complete extended IBM graphics set (ASCII 0-254) can be typed on screen and printed WYSIWYG directly from the editor with most printers.

Like the rest of the Funnelweb system, the 80-column v5.2 editor is fairware. If you use it regularly you should send Tony McGovern and additional donation. If you want to try it out send me $1 and I will send you the v5.2 upgrade files on a DSSD disk.

ACCESS

BMS — Bud Mills Services (SCSI cards and hard drives, extra 32K fast RAM added to Geneve), 166 Dartmouth Drive, Toledo, OH 43614. Phone (419) 385-5946.

WHT— Western Horizon Technologies (same products as BMS, also RXB

(See Page 27)

MICRO-REVIEWS —

(Continued from Page 26)

cartridge), 3297 Woody Lane, San Jose, CA 95132. Phone (408) 848-5947, ask for Don O'Neil. Internet doneil@hooked.net

Richard Gilbertson (RXB author), 1901 H. Street, Vancouver, WA 98663, (360) 737-7963.

Tony McGovern (FWB v5.2 author), 215 Grinsell St., Kotara, NSW 2289, Australia. or on the Internet at phpam@cc.newcastle.edu.au Delphi GLOBAL01

Charles Good (your humble reviewer and sender of $1 disks), P.O. Box 647, Venedocia OH 45894. Phone (419) 667-31431. Internet cgood@osulima1.lima.ohio-state.edu (preferred) or good.6@osu.edu

Selecting a surge protector

Finding a good one is worth the extra effort

By GARY W. COX

Reprinted from the May 1995 newsletter of the Mid-South 99ers)

WHAT IS A SURGE PROTECTOR/SUPPRESSOR?

A surge protector/suppresser is simply a device which attempts to prevent unwanted voltage surges from reaching your electronic equipment. The surge protector operates by providing an alternative pathway for the additional electrical current by either absorbing the energy or diverting the additional energy to ground, thus, away from your equipment.

WHAT IS A VOLTAGE SURGE?

A voltage surge is any voltage above the 110/115 volts in a normal electrical outlet. The power supplies in most electronic equipment can filter out momentary power surges under about 300 to 500 so volts, but anything above that can damage your equipment.

WHAT CAUSES VOLTAGE SURGES?

The most obvious cause of a voltage surge is lightning/electrical storms, which can cause serious damage to electronic equipment, especially computers! Other causes are not so obvious, such as your air-condition unit kicking on, causing a drop and then a surge in electricity, which sometimes can be seen by your lights blinking. Motors hooked into the electrical line can cause surges or work being done by the power company can cause a surge. However, power surges are not to be confused by loss of power, as a surge protector protects *only* against overvoltage, not undervoltage. However, losing power usually doesn't damage equipment, whereas too much voltage will damage equipment.

HOW CAN I PROTECT AGAINST VOLTAGE SURGES?

That's what surge protectors are for. Most surge protectors plug into your wall outlet and then the equipment that you wish to protect plugs into the surge protector.

All computer equipment should have some sort of surge protector on it, but the quality of the surge protectors greatly varies, as does the price!

HOW DO I TELL A GOOD SURGE PROTECTOR FROM A NOT-SO-GOOD ONE?

Underwriters Laboratories has developed a minimum standard for surge protectors called UL listing 1449. Any good surge protector should carry this UL 1449 listing for absorbing surges between any pair of the three wires in an electrical circuits, which are the ground wire, the "hot" or line wire and the neutral wire. Some surge protectors only protect as little as one of these lines when all three should be protected.

A good surge protector should have an indicator light indicating that the device is working, as it is important to know if the surge protection is still operating. Some protectors have only a light to indicate the status of the protection, while other protectors will have an additional circuit which shuts down the entire unit when protection is compromised, to prevent any unprotected power from reaching your equipment.

Many protectors have a site wiring fault indicator to indicate whether the outlet that you are plugged into is wired OK or if the outlet is improperly grounded or if the polarity is reversed, which can prevent the surge protector from working properly.

Read and understand the documentation on the surge protector. Even the cheap surge protectors have fancy packages, so don't be fooled by a pretty package.

Some surge protectors only have MOVs (Metal Oxide Varistors) to protect against surges. MOVs will absorb a surge, but MOVs provide only the minimum protection necessary. The surge protector should have more protection than just MOVs. The more that is between the power line and your equipment, the better protection that you have against surges!

Some sophisticated surge protectors have what is called sine wave tracking, which tracks the AC sine wave and cuts a surge off the top of the wave. In other words, this feature can provide for better protection.

Most surge protectors have EMI/RFI filtering which filters out any noise in the line and it is a nice feature to have in a surge protector.

In looking at the specifications on the surge protector look at how much voltage it will let through in a surge. I have seen some surge protectors let through as little as less than 40 volts while others let through 500 volts or more! The UL 1449 standard states that a surge protector should not let through more than 330 volts to comply with the standard. Most power supplies will filter out a 330 volt surge but around 500 volts or more seems to me a little

(See Page 28)

SURGE PROTECTORS—

(Continued from Page 27)

high. Also look at how high a surge the protector is capable of handling. Some surge protectors fail at 3,000 volts while others will go much higher and, of course, the higher the better.

Many surge protectors offer a guarantee whereby if you equipment is damaged by a surge they will pay for the repair up to a certain amount. This amount varies from anywhere around $2,000 5o $25,000.

Another feature to look for in a surge protector is modem/fax protection, as a voltage surge can come through your phone line just as easily as your electrical line. Many surge protectors have a phone line protector built in or a phone line protector can be purchased separately.

Most of the good surge protectors tend to sell for $50 or more, so just remember you get what you pay for. If you buy a $6 surge protector you probably do not have much protection.

Surge protectors typically come in various forms. The most common is in the form of an outlet strip with multiple outlets. Another form is in a complete unit that plugs directly into an electrical outlet covering up both outlets yet providing for several plugs. Another type unit is in the form of a single cube where your existing outlet strip or any electrical device plugs into it and it plugs into the wall. Surge protectors can be purchased in the form of outlets themselves replacing the standard in-the-wall electrical outlet or they can be purchased in the form where they go into your electrical feed box. Phone line protectors can come in the form of a single unit which plugs into your electrical outlet (for grounding) or many of the electrical surge protector units also provide a phone jack for protecting the phone line.

Additionally, if protection against loss of power is needed in addition to surge protection, then a UPS (Uninterruptable Power Supply) is the answer. UPSes will provide power to equipment when loss of power from the power company occurs without disturbing the operation of that equipment. However, UPSes are only for temporary power so that a user will have time to finish what he or she is doing and shut down the system properly without data loss. Most UPSes also provide surge protection and can react to losses of power of a duration of less than a second. Your computer equipment must have continuous power as an interruption of less than a second can reboot your system and possibly cause data loss. The UPS also must be matched with the amount of equipment that you wish to protect as too much equipment attached to a UPS can overload it. The UPS must be large enough for the amount of equipment connected to it. Furthermore, the larger the UPS the longer it will continue to provide power in a power failure. Depending on the size of the UPS, it can supply power for a duration of typically 15 minutes to several hours. Of course, the larger the UPS, the more it will cost, and many manufacturers also provide a guarantee that if your equipment is damaged, they will pay for the repair up to a certain amount.

When shopping for a surge protector, be sure to read the package for the specifications, and do not be fooled by the fancy ta$4.0 just pay attention to the actual specifications.

I hope that this helps a little while you are out shopping for a surge protector.

USER NOTES

HRD users: change the module first

The following item was written by Leonard Taffs of the SouthWest Ninety-Niners user group.

If you are using a Horizon 4000 RAMdisk and you have lockup while using Extended BASIC or Super Extended BASIC or while in Writer programs, etc., and you get only a blank screen when you turn the computer back on — even after turning the whole system off and waiting — you have the classic symptom of having lost your ROS (RAMdisk operating system), at least.

Instead of trying to find your ROS disk to reload it, try changing the module you may have in your cartridge port. I have found this be a fast way to make certain whether the ROS is gone or not. More times than not, I have found my ROS still there after this module switching trick. It might be a good idea to clean the offending module's contact points before re-inserting it.

XBASIC program prints graphs

The following program was written by Jim Peterson and was included among his Tips from the Tigercub No. 35. It requires Extended BASIC.

The program, called GRAPHPRINT, prints a graph based on up to 30 values that the user enters. It's designed to output to a parallel printer but can easily be configured for a serial port. Additional information is included at the beginning of the program.

GRAPHPRINT

```
100 !TIGERCUB GRAPHPRINT by
Jim Peterson !162
110 !Will output to printer
a line graph of 31 items of
data as, for instance, the t
emperature for each day of a
month. !175
120 !Values must be positive
integers within a range of
75 from minimum to maximum !
202
130 M$=RPT$("|_",65):: DIM T
$(31),D$(75):: MN=10000 !127
140 DISPLAY AT(12,1)ERASE A
L:"Input data -.maximum 31":
"items. Enter to finish" !16
```

(See Page 29)

USER NOTES

(Continued from Page 28)

```
150 FOR X=1 TO 31 :: DISPLAY
AT(14,1):X;TAB(4);CHR$(1)::
ACCEPT AT(14,4)VALIDATE(DIG
IT)SIZE(-5)BEEP:T$(X):: IF T
$(X)=CHR$(1)THEN X=X-1 :: GO
TO 170 !049
160 T=VAL(T$(X)):: MX=MAX(MX
,T):: MN=MIN(MN,T):: NEXT X
!173
170 RN=MX-MN :: IF RN>75 THE
N PRINT "EXCEEDS MAXIMUM RAN
GE OF 75" :: STOP !081
180 IF MX>75 THEN AD=MX-75 !
175
190 OPEN #1:"PIO",VARIABLE 1
32 :: PRINT #1:CHR$(15);CHR$
(27);CHR$(51);CHR$(12):: PRI
NT #1:RPT$("_",132)!009
200 DISPLAY AT(12,1)ERASE AL
L:"Wait, please...": :".....
.this takes time" !061
210 LM=LEN(STR$(MX)):: FOR J
=1 TO 75 :: J$=STR$(76+AD-J)
!252
220 IF J>66+AD THEN J$=J$&"
" !244
230 IF J/2=INT(J/2)THEN D$(J
)=RPT$(" ",LM)&SEG$(M$,1,132
-LM)ELSE D$(J)=J$&SEG$(M$,1,
132-LM)!127
240 NEXT J :: PRINT #1:RPT$(
" ",LM)&SEG$(M$,1,132-LM)!18
2
250 J=1 :: T=VAL(T$(J))-AD :
: T=76-T :: D$(T)=SEG$(D$(T)
,1,J*4+4)&CHR$(239)&SEG$(D$(
T),J*4+6,255):: J=J+1 !227
260 T2=T :: T=VAL(T$(J))-AD
:: T=76-T :: FOR N=T2 TO T S
TEP (T2>T)+ABS(T>=T2):: D$(N
)=SEG$(D$(N),1,J*4+2)&CHR$(2
53+(T<T2))&SEG$(D$(N),J*4+4,
255):: NEXT N !172
270 J=J+1 :: D$(T)=SEG$(D$(T
),1,J*4)&CHR$(239)&SEG$(D$(T
),J*4+2,255):: IF J<=X THEN
260 !014
280 FOR J=1 TO 75 :: PRINT #
1:D$(J):: NEXT J :: PRINT #1
!213
290 T=8 :: FOR J=1 TO 31 ::
PRINT #1:TAB(T);STR$(J);:: T
=T+4 :: NEXT J !139
```

Using EXPMEM2 for data

The following item has appeared in several user group newsletters. It was originally compiled by Scott Darling.

If you need to work with quite a bit of data or would like to change programs but save the data after you press CALL QUIT, then you can set up the 24K of high-memory in the 32K memory expansion as a single data file called "EXPMEM2." You open this file just as you would a disk file, with one exception — you must precede

(See Page 30)

USER NOTES

(Continued from Page 29)

the OPEN statement with a CALL LOAD to the location -24574 as follows:

For INT/VAR files — 24
For DIS/VAR files — 16
For INT/FIX files — 8
For DIS/FIX files — 0

EXAMPLE

If you want to open up the expansion memory for display, variable 80 files, this is what you'd do:

```
100 CALL INIT
110 CALL LOAD(-24574,16)
120 OPEN #1:"EXPMEM2",RELATI
VE,UPDATE,DISPLAY,VARIABLE 8
0
```

Then continue on as you normally would.

If you want to store both data and assembly language routines at the same time, do this:

```
100 CALL INIT
110 CALL LOAD(-24574,16)
120 OPEN #1:"EXPMEM2"
130 CALL LOAD("DSK1.ASSM1")
140 CALL LOAD("DSK2.ASSM2")
150 CALL LINK ("START")
160 REM CONTINUE REST OF PRO
GRAM
```

In the above example, the 24K of high-memory was saved for use as a DATA file (DIS/VAR 80 format), then the assembly routines were loaded. The computer will look for the best place to put the routines and will adjust the pointer accordingly. After the routines are loaded, a LINK statement starts the first routine and off we go.

If that's not enough for you, you can also use the Mini-Memory Module for 4K more of storage for assembly routines. Now that's 16K of program space and 12K of assembly routine space.

Converting files to D/V 80

This tip is by Andy Frueh, of the Lima (Ohio) 99ers. We got it from the May 1995 issue of the *PUG Peripheral*, the newsletter of the Pittsburgh Users Group:

If you have a D/V 40, 132 or whatever text file that you need to edit with TI-Writer, you can change it to a D/V 80 file by using a sector editor. TI-Writer will load *only* D/V 80 files. Find the header sector of the D/V file. Edit that sector in HEX mode. Go to byte>11. If the file is D/V 40, you should see the number 28 ("(" in ASCII) here. To change it to an "80" (or "P" in ASCII), type the number 50 over the 28.

UUENCODE and UUDECODE

The following programs, UUENCODE and UUDECODE, are used to decode and encode files commonly transmitted over the Internet. ENCODE is used to encode a file and DECODE is used to decode an encoded file. The programs work with non-program image files. Program image files, such as Extended BASIC programs, must be archived prior to running them through UUENCODE.

These programs require Extended BASIC and a memory expansion. Author of the program is unknown.

UUENCODE

```
1 PRINT "":"":"":"" :: FOR D
=0 TO 14 :: CALL COLOR(D,3,1
):: NEXT D :: CALL SCREEN(2)
:: CALL CHAR(140,RPT$("F",16
))!114
2 INPUT "Path.filename of fi
le to encode:":SF$ :: INP
UT "Path.filename of text fi
le :":TF$ !069
3 OPEN #5:SEG$(SF$,1,5),INPU
T ,RELATIVE,INTERNAL !067
4 INPUT #5:A$,A,B,C :: PRINT
"":"selected disk:"&SEG$(SF
$,1,4):"Sectors free:";B-C:"
Sectors available:";C:"":"**
* SEARCHING ***":"" !117
5 INPUT #5:A$,A,B,C :: IF A$
=SEG$(SF$,6,255)THEN PRINT A
$ :: GOTO 7 !006
6 IF A=9 THEN CALL SOUND(200
,217,0):: PRINT "":"":"*** F
ILE NOT FOUND ***" :: RUN EL
SE 5 !044
7 A=ABS(A):: IF A=5 THEN PRI
NT "":"":"":"Memory image (P
ROGRAM) filescannot be direc
tly encoded! Please archive
the file.":"" :: CALL SOUND(
4250,217,0):: STOP !069
8 PRINT "":"File matched O.K
.              Encoding
Started
" !021
9 OPEN #2:TF$,DISPLAY ,VARIA
BLE 80 :: PRINT #2:"*TI UUEN
CODE*" :: IF A=1 THEN OPEN #
1:SF$,INPUT ,DISPLAY ,FIXED
C :: PRINT #2:"DIS/FIX" !015
10 IF A=2 THEN OPEN #1:SF$,I
NPUT ,DISPLAY ,VARIABLE C ::
PRINT #2:"DIS/VAR" !07411 I
F A=3 THEN OPEN #1:SF$,INPUT
,INTERNAL,FIXED C :: PRINT
#2:"INT/FIX" !174
12 IF A=4 THEN OPEN #1:SF$,I
NPUT ,INTERNAL,VARIABLE C ::
PRINT #2:"INT/VAR" !17013 P
RINT #2:C:SEG$(SF$,6,255)!01
5
14 CALL GETCH(C):: IF C=3276
7 THEN PRINT "":":" *** DON
E ***" :: CLOSE #1 :: CLOSE
#2 :: CLOSE #5 :: END !089
15 IF C=3000 THEN PRINT #2:
a" :: GOTO 14 !021
16 PRINT #2:CHR$(64+((A AND
240)/16))&CHR$(64+(C AND 15)
);:: GOTO 14 !203
30000 SUB GETCH(C):: IF F$<>
"" THEN 30002 ELSE IF EOF(1)
THEN C=32767 :: SUBEXIT ELSE
INPUT #1:F$ :: C=3000 :: SU
BEXIT !169
30002 C=ASC(F$):: F$=SEG$(F$
,2,255):: SUBEND !255
```

UUDECODE

```
1 PRINT "":"":"":"" :: FOR D
=0 TO 14 :: CALL COLOR(D,3,1
):: NEXT D :: CALL SCREEN(2)
:: CALL CHAR(140,RPT$("F",16
))!114
2 INPUT "Path.filename of fi
le to decode:":SF$ !192
3 OPEN #1:SF$ :: INPUT #1:A$
:: IF A$="*TI UUENCODE*" TH
EN 4 !016
4 INPUT #1:A$ :: INPUT #1:RI
:: INPUT #1:N$ :: PRINT "":
"Use original filename (Y/N)
```

(See Page 31)

USER NOTES

(Continued from Page 30)

```
?" !057
5 CALL KEY(3,K,S):: IF S=0 T
HEN 5 !076
6 IF K=ASC("Y")THEN INPUT "O
.K., what pathname (inc .) :
":P$ :: N$=P$&N$ :: GOTO 8 !
038
7 INPUT "Enter path.filename
:":N$ !049
8 IF A$="DIS/FIX" THEN OPEN
#2:N$,DISPLAY ,FIXED RL !121
9 IF A$="DIS/VAR" THEN OPEN
#2:N$,DISPLAY ,VARIABLE RL !
116
10 IF A$="INT/FIX" THEN OPEN
#2:N$,INTERNAL,FIXED RL !21
5
11 IF A$="INT/VAR" THEN OPEN
#2:N$,INTERNAL,VARIABLE RL
!210
12 PRINT "":"* Decoding *" !
156
13 CALL GETCH(C):: IF C=3276
7 THEN 50 ELSE IF C=97 THEN
PRINT #2:"" :: PRINT :: GOTO
13 ELSE C2=C :: CALL GETCH(
C)!007
14 PRINT #2:CHR$((C2*16)+C);
:: GOTO 13 !248
50 PRINT "":"":" *** DONE **
*" :: CLOSE #1 :: CLOSE #2 !
029
30000 SUB GETCH(C):: IF F$<>
"" THEN 30002 ELSE IF EOF(1)
THEN C=32767 :: SUBEXIT ELSE
INPUT #1:F$ !110
30002 C=ASC(F$):: F$=SEG$(F$
,2,255):: SUBEND !255
```

Ormand was writer

Tom Wills didn't write last month's article about the Southwest 99ers taking over production of the Asgard Memory System, Wills reports. To set the record straight, David Ormand wrote the article.

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

CLASSIFIEDS

FOR SALE

T.O.D. FANS!

New and challenging Tunnels of Doom Adventure, with the flavor of Middle Earth! It's called **Halls of Lost...MORIA!** and takes place in the dawn of the Fourth Age.Battle Orcs, Cold Drakes, Wights, Goblins, Trolls and Balrogs while exploring Moria for lost treasures. Unique graphics and ploys. Send $4.95 ($5.95 outside USA) for disk and booklet to: Michael Veprauskas, 32 Vaillancourt Dr., New Ipswich, NH 03071. v12n5

TI99/4A

TI software & hardware. Joy Electronics Inc., P.O. Box 542526, Dallas, TX 75354-2526. 214-243-5371. Call or write for free catalogue. v12n5

FOR SALE

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

TIPS

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

Classified ads are 10 cents per word

If you want to sell or buy something,
advertise it in MICROpendium Classifieds.
Simply write your ad on a separate sheet of paper,
count the word (a phone number counts as one word)
and send it, along with payment, to:

MICROpendium Classifieds,
P.O. Box 1343
Round Rock, TX 78680.

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$35** ❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50** ❑ 12 issues, other countries surface mail, **$40.00**

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

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

Address Changes

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

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

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

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

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

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

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

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

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS

A T EXPIRES 1/96
CHARLES GOOD
P.O. BOX 1343
ROUND ROCK TX 78664