

---

Covering the TI99/4A and the Myarc 9640

---

# MICROpendium

Volume 11 Number 5

June 1994

\$3.50

---



## Lima Multi User Group Conference

Inside  
KONO-5  
The Art of Assembly  
Living with spiders  
Using menus with TI-Base

Reviews  
Double Entry Bookkeeping  
Bunnelweb V5.01 Editor  
Bunnelweb Text Sort  
Reformat  
Budget Planner  
Programmable Flash Memory

# 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.

No information published in the pages of MICROpendium may be used without permission of the publisher, Burns-Koloen Communications Inc. Only computer user groups that have exchange agreements with MICROpendium may excerpt articles appearing in MICROpendium without prior approval.

While all efforts are directed at providing factual and true information in published articles, the publisher cannot accept responsibility for errors that appear in advertising or text appearing in MICROpendium. The inclusion of brand names in text does not constitute an endorsement of any product by the publisher. Statements published by MICROpendium which reflect erroneously on individuals, products or companies will be corrected upon contacting the publisher.

Unless the author specifies, letters will be treated as unconditionally assigned for publication, copyright purposes and use in any other publication or brochure and are subject to MICROpendium's unrestricted right to edit and comment.

Display advertising deadlines and rates are available upon request.

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: (512) 255-1512

CompuServe: 75156,3270

Delphi TI NET: MICROpendium

GENie: J.Koloen

Internet E-mail: jkoloen@io.com

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

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

## Lima MUG report

140 TIers have successful conference ..... Page 6

## KONO-5

A challenging board game in Extended BASIC .....Page 8

## The Art of Assembly

We keep learning .....Page 11

## Living with spiders

Writing programs for Funnelweb .....Page 14

## MY-BASIC

Emulating the assembly BLINK .....Page 16

## TI-BASE

Using menus with TI-Base .....Page 18

## A program that writes programs

Automatic disk menu of programs .....Page 20

## Spreadsheet word processing

Multiplan is handy for some text processing tasks .....Page 22

## Reviews

Double Entry Bookkeeping, Funnelweb V5.01 Editor, Funnelweb Text Sort, Reformat.....Page 24

Budget Planner .....Page 25

Programmable Flash Memory .....Page 26

## User Notes

Floppy disk facts, making a modem cable, converting Telco D/V128 files into D/V80 and more .....Page 28

### \*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 DOLLAR DAYS

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

OUR LOWEST PRICES EVER ON THIS ENTIRE PAGE!

Big Price Reduction for Our Full-Featured COMPLETE PRINT SHOP PACKAGE

FOR THE TI-99/4A special price \$4.95\*

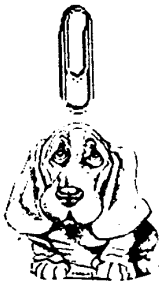
featuring GRAPHX+ with Flip & Rotate & Quick Load (REG \$19.95)



NEW FONTS INCREDIBLE

THE ALL TIME BEST GRAPHICS PROGRAM FOR THE 99/4A HAS JUST GOT EVEN BETTER WITH AN ALL NEW PACKAGE AT A GREAT LOW PRICE. ONLY \$19.95 COMPLETE WITH GRAPHX+ AND TWO DISK SIDES FILLED WITH NEW CLIPART AND FONTS. GRAPHX+ OFFERS EVERY FEATURE YOU NEED TO CREATE THE BEST 4A GRAPHICS YOU HAVE EVER SEEN. GRAPHX+ OFFERS FEATURES SUCH AS FREE HAND DRAWING AND ERASING, ZOOM FOR DETAILED WORK, AUTO FILL, COLORS, TEXT AND TITLES, AUTO CIRCLE DRAWING, CLIP STORAGE AND ANIMATION. 32K, EXBASIC, AND A STAR/EPSON COMPATIBLE DOT MATRIX PRINTER IS REQUIRED.

\*BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF \$4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE FOR ONLY \$1.00.



## GRAPHX DINOSAURS!



\$4.95

**DISK OF DINOSAURS** is a fantastic collection of real and artistic dinosaurs from the amazing \$4.95 artist Pro collection. This special utility for Graphx also contains a bonus disk "Disk of the Dinosaurs" for creating great artwork along with all the other great clip art. Graphx with printout resolution far superior to the best going with it's highly sophisticated "zoom", built-in color copy function, multi-level printer, and always available clipboard is, in the words of three Product of Asgard Software, "The most exciting and available for the price."

Now you can "bring back" anything from dinosaurs to sporting events to sports. Create dinosaur artwork, logos, and pictures faster than you can say "Dinosaur!" Can also be used with TI-99/4A with the convenient remote supplied. Graphx and a 1.2M disk drive system required.



## MAC-FLICK

HIGH RESOLUTION CLIP ART FOR GRAPHX AND TI ARTISTS



TEX+COMP HAS TAKEN THE BEST MAC CLIPART AVAILABLE AND CONVERTED IT TO GRAPHX CLIPART FILES. NOW YOU CAN PRODUCE GRAPHICS WITH YOUR 4A THAT RIVAL THE BEST ANYWHERE. WE FILLED SIX DISK SIDES WITH OVER 100 PIECES OF CLIPART THAT ARE READY TO BE PRINTED OUT WITH GRAPHX OR CAN EVEN BE CONVERTED TO OTHER GRAPHIC SYSTEMS. THE SUPERIOR PRINTING CAPABILITY OF GRAPHX WILL CREATE ART THAT WILL BE OUTSTANDING!!

\$4.95

SIX DISK SIDES GRAPHX, 32K, AND STAR/EPSON DOT MATRIX PRINTER REQ.



ASGARD SOFTWARE HAS PUBLISHED FOUR SETS OF PROFESSIONAL CLIPART AND FONTS FOR USE WITH THE GRAPHX+ PUBLISHING SYSTEM SO YOU CAN CREATE FANTASTIC ARTWORK WITHOUT EVEN HAVING ARTISTIC TALENT. EACH SET IS ONLY \$4.95 (VOL I-IV)

GRAPHX SLIDESHOW BY ASGARD IS A CLIPART COLLECTION AND A SPECIAL PROGRAM FOR DISPLAYING YOUR GRAPHX CREATIONS IN AUTOMATIC SEQUENCE. GREAT FOR DISPLAYS, TRADESHOWS AND PRESENTATIONS.....ONLY \$4.95

- GRAPHX+ SYSTEM (reg. 19.95) special sale price.....\$4.95\*
- ASGARD GRAPHX COMPANION I.....\$4.95
- ASGARD GRAPHX COMPANION II.....\$4.95
- ASGARD GRAPHX COMPANION III.....\$4.95
- ASGARD GRAPHX COMPANION IV.....\$4.95
- ASGARD GRAPHX SLIDESHOW.....\$4.95
- PRINTING TO GO (Clipart for GRAPHX reg. 9.95).....\$4.95
- GRAPHX DINOSAURS (Dinosaur clipart reg. 12.95).....\$4.95
- MAC-FLICK (Hi-res. clipart for GRAPHX reg. 12.95).....\$4.95

\*BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF \$4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE (REG \$19.95) FOR ONLY \$1.00.

**PRINTING TO GO**  
exciting new pieces of Clip Art  
For use with GRAPHX \$4.95  
THE ULTIMATE GRAPHICS SOFTWARE

PRINTING TO GO IS A GREAT COLLECTION OF CLIP ART AND FONTS TO ADD TO YOUR GRAPHX PUBLISHING SYSTEM FOR THE TI-99/4A. THIS GREAT PACKAGE OF FOUR DISK SIDES AT A FANTASTIC PRICE IS ADDING PROOF THAT GRAPHX IS THE BEST BUY IN 4A GRAPHIC PACKAGES AND BY FAR, THE BEST SUPPORTED. THIS COLLECTION CAN BE USED FOR CREATING NEWSLETTERS, ANNOUNCEMENTS OR ANY OTHER APPLICATION WHERE YOU NEED PROFESSIONAL LOOKING ART. THESE CLIPARTS CAN ALSO BE USED FOR GREAT AND PERSONALIZED STATIONERY OR EVEN FOR MAKING T-SHIRTS WITH THE TEX+COMP T-SHIRT KIT. OVER 100 PIECES OF QUALITY CLIPART & FONTS.

TEX+COMP  
P.O. Box 33084, Granada Hills, CA 91344

**CHARGE IT**  
WE ACCEPT VISA or MASTERCARD

VISA and MASTERCARD HOLDERS CALL DIRECT:  
(818) 366-6631

ORDER BY PHONE (818) 366-6631  
24 HOURS A DAY  
7 Days a Week!

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take ten days to clear. Add 3% shipping & handling (with a \$3 minimum). Add 4.5% cost of shipping. All prices shown include a 3% cash discount. Add 3% if paying by Visa or Mastercard. Prices and availability subject to change. Include street address for UPS delivery. California orders add 8.25% sales tax. Canada, Hawaii, Alaska and overseas shipping extra.

Send \$2 for our complete catalog

# COMMENTS

## Bobbitt makes exit

Asgard Peripherals, at one time the hardware end of Asgard Software, has closed its doors. It hasn't actually made an announcement, but it is unlikely that you'll see any more products from the company, though Jim Krych continues to support the AMS card.

The company had great plans for future products, but the lack of success with its AMS (Asgard Memory System) and problems with its XBIII cartridge card doomed it to failure.

It's a sad but inevitable ending to a story that began in the early days of the TI when Chris Bobbitt started the company to produce and market third party software for the TI. During the mid-1980s, it was one of the major players in TI software, marketing an impressive line of software. Several years ago Bobbitt made Harry Brashear a partner in the company, eventually transferring ownership to Harry while concentrating on the newer Asgard Peripherals. Problems with suppliers eventually took their toll on Bobbitt and the result is that the TI community loses another player.

Bobbitt played a significant role in the TI community, and he did it for years. Often a controversial subject, his commitment to the TI was never questioned. I'll miss Chris and his enthusiasm for the TI. I wish him all the luck in the future.

### TALK TO US ON SATURDAY MORNINGS

Readers who'd like to talk to us by phone stand the best chance by calling between 9 a.m. and noon on Saturdays. We try to make call-backs on weeknights, but it just depends on whether we have the time.

—JK

# FEEDBACK

## A good man gone

We were shocked to hear about Jim Peterson's death. It looks like good fellows always have to go away too soon. We'll cherish his programs, his articles, and we'll keep him alive in our hearts.

Norberto Revilla  
T.I.G.R.E.S. de Argentina  
Buenos Aires

## Lima conference is worth the trip

After driving through Thursday afternoon thunderstorms in New Jersey, crisscrossing the east-west divider on Route 80, riding single file for miles and miles in the Red Devil, a 1993 Ford Festiva with a huge 1300 cc engine, I crossed the Pennsylvania-Ohio border. This was the farthest west I had ever been. Where were all

the buffalo?

I was pretty beat by this point but I sensed the allure of all those TMS 9900 series microprocessors. So, onward I went. One-thirty Friday morning and I was in the Land of Oz. Real exciting. Sept till 1:30 Friday afternoon. I regrouped and found my way to the Ohio State University Lima campus. After getting directions from a math professor — there are four buildings and the driveway you're on is tangential to the polygon .... — I found Reed Hall.

I walked in and found Dr. Charles Good directing traffic. Seizing the opportunity to be quite witty, I announced myself as the Connecticut Division of the Lima User Group, chest forward, feet together, three-fingered salute.

Well, he looked at me as though I was a Commodore in TI clothing. So much for being cute! He recognized my name when

(See Page 5)

## READER TO READER

□ David Ormand, 212 S. Nema Place, Tuscon, AZ 85711, writes:

I am doing some "market research" for a product for TI hardware hackers. This is a small circuit card that provides the 60-pin edge connector for the P-Box, circuitry for decoding memory and CRU addresses and buffering the address and data buses and wire-wrap posts for all bus signals. It will bolt onto a general-purpose prototype card that the user can buy from ordinary parts-supply houses and cut to supplied dimensions. Expected price is \$25. If anyone would be interested, please send E-mail to d.ormand1@genie.geis.com, send me a postcard or call at (602) 795-2005 and leave a message.

□ Laurence Topliffe, 1609 Lake Lotela Dr., Avon Park, IL 33825, (813) 452-1599, writes:

Can someone write a program that will read a MAILLIST file made with TI-Writer and print envelopes with both a return address and those addresses while ignoring the asterisks and numbers?

I've tried to but am not knowledgeable enough.

□ Bruce Harrison, of Hyattsville, Maryland, has shared his reply to Phil Van Nostrand, who inquired about finding disks for a 3-inch drive in our April 1994 issue:

I have seen drives that use a smaller than 3.5-inch disk used in the very expensive Yamaha Clavinova electronic keyboards. You might be able to find those 3-inch disks at a music store which sells that line. (Generally, the "high-class" musical instrument stores would sell Clavinovas.) As I recall, the disks themselves look just like the 3.5-inch kind, except, of course, that they're smaller. At the time, I remember being astonished that anyone would introduce yet another size of disk that wouldn't be compatible with anything in the PC or computer realms.

Reader to Reader is a column to put TI and Genev 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.

# FEEDBACK

(Continued from Page 4)

I sheepishly offered it and was magnanimous in his greeting.

He was thinner and taller than I expected. Must be due to the camera angles and such a viewed on my monthly Lima tapes.

I immediately picked up a tape dispenser and some preprinted signs to mark off who got how many tables where. I finally felt like a real member of the Lima UG.

I met many Tiers that I had only spoken to by phone. I started spending money and the Multi User Group Conference had not officially opened.

That night I met some of the big players in the TI community, including Beery Miller and Gary Cox, Don Walden from Cecure and Tim Tesch of S&T Software.

That night I was sitting in Beery's room as he continued to refine MDOS V2.0, and had Walden and James Schroeder demonstrate explain the upgrades available for the Geneve — 384K expanded memory for a total of 917K, Programmable Flash memory and PFM+. I saw this stuff in action and it was impressive. If you want to operate in 80 columns, have the ability to reprogram your operating systems, have a choice of which files to boot up and other options I'm too illiterate to explain, get thee to a Geneve and fill 'er up with all the goodies you can. You still have your trusty 99/4A to play with and, should you have jumped to PC compatibility, a choice of two TI emulators to save your first love in computers from extinction.

The following day's events are a blur. My first observation is you can't tell Tiers by looking at them or reading their resumes. All shapes, sizes and backgrounds were represented. Most vendors are avid Tiers, as well as business people.

If you expect to get a good deal on a cartridge you never managed to get around to buying, you're idea of a bargain and the dealer's are probably not in the same ballpark. Oh, there are deals to be had. These transpire in a number of scenarios. Tiers who are trying to get rid of some stuff because their families are about to throw them out of the house will deal. If you have a lot of money and can make bulk purchases, you can deal. There are a few people with full systems to sell and would

not break them up. You can find someone who wants the parts of the system you don't want and go in for the kill. Keep Michael Millkin, junk bonds, corporate takeovers in the back of your mind and you'll know how to approach your target. Keep some of your stash until the end of the day and watch how the prices plummet.

What was amazing to me were the gadgets that are still available, such as a CorComp Micro Expansion System. I saw a GRAM Kracker. The individual wanted \$100, maybe \$125 for it. I don't know what it actually went for. There were Super Sketches, mice, joysticks and diskettes. All reasonably priced. There were cartridges by Romox, Milton Bradley, Funware, Spinnaker and Imagic. There were direct parallel printer interfaces that didn't require RS232 cards or PEBs. Many books relating to TI-ing and computing in general. My advice is to make up a wish list for next year, start saving and come to Lima next year. Plenty of hotels to stay at. Plenty of people to shuttle you back and forth.

Don't forget the seminars. There are plenty of knowledgeable Tiers to answer all those questions you were afraid to ask. Just to meet Barry Traver and chat for a bit was worth the ride. He approaches the TI with a perspective that is subtle, unique and thoroughly modern. If you think your TI is behind the times, talk to Barry. If a computer were built today that was user-friendly and unique it would have many of the qualities of the 99/4A.

The seminars overlap by a half-hour, so you'll never see and hear all that is going on. But you can get tapes from Charlie (see story on page 7). I have received these tapes for the past three years and there are always many hours of instruction, explanation and demonstration.

Now let's turn to what is NOT at the conference. (I preface this by stating that the fact that anyone even attempts to manufacture hardware or write software for our 10-year-old orphans is beyond my expectations. If some produces are delayed or some goals are never achieved it is not for lack of trying. These community should respect those who have tried.

Bud Mills and Western Horizon Tech-

nology don't have all the bugs out of their SCSI project. Bud has certainly not made a lot of money on TI related stuff and has generously helped those who own a product of his, even if it was not purchased from him. Sometimes he has taken what some would consider a long time to return a product for repair. I once waited three months for my return. I was glad when I got it. If I had the money, I would send Bud a deposit on a SCSI device. He has apologized for the lack of a working model and offered to return his customers' money. Doesn't sound shabby to me. Anytime I want to get angry I can call the manufacturer of my 486SX and get the runaround for weeks on end.

Having said that, there seem to be some companies to avoid when making a TI purchase. There is a company in Canada that builds an 80-column device. I have heard fire and brimstone for the way the company conducts business. I've also heard support and praise for its attempts at projects. Bottom line is that the company can't get it's stuff together. If you want 80 columns, get a used AVPC, a Geneve, a used TIM, or wait and see what WHT comes up with. I would also appreciate straight talk from Asgard Peripherals. If you have left the TI market, just say it.

It is hard on me mentally and physically. Many of the Tiers at the conference made it worth the effort. To meet them is as much a reason to go as to get my hands on all those goodies. Everyone there seemed to enjoy themselves. I hope there is another MUG next year. You ought to go. You'll never know what you're missing.

Gary Fitzgerald  
Nutmeg 99ers, Lima UG, Long Island  
99ers, Ottawa UG

*Feedback is a reader forum. The editor may condense excessively lengthy submissions if necessary. We ask that writers limit themselves to one subject per submission. Our only requirement is that submissions be of interest to those using the TI99/4A, the Geneve 9640 or compatibles. Send items to MICROpendium Feedback, P.O. Box 1343, Round Rock, TX 78680.*

## Lima Multi User Group Conference

# 140 visitors turn annual meeting into big success

By GARY W. COX

About 140 TIers from many states converged in Ohio on May 14 at the annual Lima Multi User Group Conference. Three rooms of vendor and user group tables greeted enthusiastic TIers looking for a bargain. Two additional rooms were used for a variety of seminars that went on continuously throughout the day. According to the sign-in sheet, visitors came from Ohio, Missouri, Indiana, Michigan, Maryland, Pennsylvania, Kentucky, Wisconsin, California, South Carolina, Illinois, Tennessee, New Jersey, Florida and Canada.

Among the many vendors present was Bud Mills of Bud Mills Services. Mills had a variety of products which not only included products such as the Horizon RAMdisk, PGRAM card, Horizon Mouse, Memex 504K and Digi-Port but a new item called an "AT Keyboard and ROM upgrade for the TI99/4A." This upgrade allows the connection of a PC-compatible keyboard to the TI99/4A in a unique way. Where other keyboard interfaces replace the keyboard in the console, this keyboard interface works along with the built in keyboard. With this upgrade it is possible to use either the AT keyboard or the TI console keyboard, as the interface allows the user to input from either keyboard at any time. The upgrade requires a little

Charles Good of the Lima TI User Group organized the conference. (Photo by Gary



work to be done on the TI99/4A motherboard but instructions are included. Or, for an additional fee, Mills will do the installation. Also displayed was the SCSI interface card. The card is apparently complete but work on the software to use the card is still continuing.

Also at the fair was Don Walden of Cecure Electronics selling items such as the PFM and PFM+ memory upgrades, as well as the MBP cards. Walden noted that a new MBP card is in the works when the existing stock of MBP cards has been sold. The new MBP card is expected to provide four analog audio outputs, 32 digital out-

puts, eight digital inputs and eight analog inputs, as well as a clock and Geneve compatibility among other features.

Walden also mentioned that Cecure Electronics is continuing to repair mo. TI compatible products but noted that he prefers that those who need "Texas Instruments" brand equipment repaired to send the equipment to TI for repair since TI still provides those services. However, he noted if you want him to repair original "TI" brand equipment that he would do so.

Beery Miller of 9640 News displayed a variety of products for the Geneve, including 9640 News volumes 1-3, Windows 9640, Barricade, Tetris, Mouse Driver development package and Global War. New was "Myterm," written by Michael Riccio for the Geneve. This is a new terminal program that runs out of MDOS. Among Myterm's features is support for uploading and downloading at 9600 baud. Miller noted production of a CD-ROM of TI software is in the final stages. Cost of the CD is \$75. Only a limited supply will be produced, according to Miller.

Larry Conner of L.L. Conner Enterprises displayed a large array of cartridges, programs and hard-to-find parts, including 9995 CPU chips for the Geneve, MBλ system headsets, Wycove Forth on tape and even a Mini-Myarc PEB.

(See Page 7)



### Exhibit hall

Visitors at the Lima Multi User Group Conference

(Photo by Gary Cox)

## LIMA CONFERENCE—

(Continued from Page 6)

Ron and Ada Markus of Ramcharged computers displayed a collection of software and equipment. New was a program called "Who's Behind the Mexican UFO's?" This game is described as being a "unique text, TI-Artist graphics, animation, sound effects type adventure game using the Missing Link." Ron noted that this is an adventure game that anyone can complete yet, even after completing the game, the user is able to run through it again and obtain different results. He noted that more programs for "The Missing Link" are in the works.

Mark Van Coppenolle and Mike Wright of CaDD Electronics showed their PC99 TI Emulator for the PC, now in version 2A. This version includes bug fixes, speed improvements and a menu driven configuration utility. Wright noted that accuracy was very important in the improvements that he continues to do on PC99. CaDD Electronics also continues to sell RichGKXB, Gramulator kit and Utility disk #1.

### Lima tapes available

Videotapes of the Lima Multi User Group Conference are available to users group representatives, to dealers who attended the conference and to speakers at the conference, according to Charles Good of the Lima Users Group.

Any of these wishing to receive the videotapes should send either \$15 or three six-hour name brand VHS tapes with \$3.75 postage to Good at P.O. Box 647, Venedocia, OH 45894.

Bruce Harrison of Harrison Software displayed a vast array of public domain utilities written by him. These utilities include his BASIC compiler, a password protection program for Horizon RAMdisks, a new EA5 loader, a reformatter for D/V80 files and font utilities. All these programs are public domain and are available from various user group libraries.

Tim Tesch of T&S Software showed a new terminal emulator called "PORT" for use with the Geneve. PORT's unique features include true ANSI color emulation. Tesch noted that he is working on adding ZMODEM upload/download protocol to "PORT" well as Class 2 fax/modem support. He is also working on software to output digital sound using the MBP card. His products can be purchased through Cecure Electronics.

Ken Gilliland of Notung Software had, among his many products, a new program called "Disk of Medieval Times." Like his other disks, it contains a game, artwork, fonts and practically everything that you wanted to know about medieval times.

Mickey Cendrowski and Mike Sealy of MS Software showed a new "TI Software Database" which uses TI-Base to catalog disks. Also new was Page Pro cataloger version 1.6.

Jim Krych of Asgard Peripherals displayed public domain schematics of the Asgard Memory System (AMS) for those  
(See Page 8)

## VENDOR LIST

9640 News, P.O. Box 752465, Memphis, TN 38175-2465; contact: Beery Miller (901) 368-1169.

Akron 99'ers, C/O Jim Harris, 2022 10th St., Cuyahoga Falls, OH 44221; contact: Jim Harris (216) 928-9675.

Asgard Peripherals, C/O Jim Krych (AMS related only), 299 Parkway, East Lake, OH 44094.

Bud Mills Services, 166 Dartmouth Dr., Toledo, Ohio 43614-2911; (419) 385-5946

CaDD Electronics, 81 Prescott Road, Raymond, NH 03077; (603) 895-0119.

Cecure Electronics Inc., P.O. Box 132, Muskego, WI 53150; (414) 679-4343 or 1-800-959-9640.

Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009; contact Hal Shanafield (708) 864-8644.

Cleveland Area TI99/4a User Groups, C/O Harry Hoffman, 3925 Trowbridge Ave., Cleveland, OH 44109; contact John Parken (216) 331-2830.

Competition Computer, 2219 S. Muskego Ave., Milwaukee, WI 53215; 1800-471-1600 or (414) 672-1600.

C.O.N.N.I. (Spirit of 99), 181 Heischman Ave., Worthington, OH 43085; contact John Parkings (614) 891-4965.

Genial Traveler, Barry Traver, 835 Green Valley Dr., Philadelphia, PA 19128; (215) 483-1379.

Great Lakes Computer Group, P.O. Box 152, Roseville, MI 48066-0152; contact June and Leonard Smith (810) 338-0872.

Harrison Software, 5705 40th Place, Hyattsville, MD 20781-1727.

Hoosier Users Group, P.O. Box 2222, Indianapolis, IN 46206-2222; contact Brian Pedigo (317) 255-7381.

K-town TI Users Group, C/O E.M. Smith, 3506 Garden Dr., Knoxville, TN 37918; contact E.M. Smith (615) 687-8869.

Lima TI99/4a Users Group, P.O. Box 647, Venedocia, Ohio 45894; contact Charles Good (419) 667-3131.

L.L. Conner Enterprise, 1521 Ferry St., Lafayette, IN 47901; (317) 742-8146.

Media Ware Software, 2141 N.W. 64th Ave., Suite 15, Sunrise, FL 33313; (305) 749-4690.

Mid-South (Memphis) TI99/4A User Group, P.O. Box 38522, Germantown, TN 38183-0522; contact Gary Cox (901) 358-0667.

MS Express Software, P.O. Box 498, Richmond, Ohio 43944; (412) 265-5201.

Notung Software, 7647 Mc Groarty St., Tujunga, CA 91042; (818) 951-2718.

Pittsburgh TI Users Group, P.O. Box 8043, Pittsburgh, PA 15216; contact Gary Taylor (412) 341-6874.

Ramcharged Computers, P.O. Box 81532, Cleveland, Ohio 44181; (216) 243-1244 or 1-800-669-1214.

RDB Enterprises, 643 Fair Ave., Shelbyville, IN 46176; contact Ricky Bottoms, 1-800-464-8851.

T&S Software; contact Cecure Electronics (see above).

TI Users of Will County, 1400 Caton Ave., Joliet, IL 60435; contact Carl Winterrose (815) 744-5330.

Kawartha 99'ers, c/o Glen Daniels, R.R. 5 Peterborough, Ontario Canada, K9J 6X6; contact John Acheson (705) 743-7751.

West Penn 99ers, c/o Mickey Cendrowski, 100 Pine St., Russellton, PA 15076; (412) 265-5201.

## A board game in Extended BASIC

# KONO-5 is a challenge

The following program, KONO-5, was adapted by Jeffrey Brown of the Ottawa TI99/4A User Group. It came to us via Lucie Dorais. The game was originally written by David Mercer of the United Kingdom for another computer brand. KONO-5 requires Extended BASIC. Documentation is included in the program.

The object of the game is to move your game pieces to the opposite side of the game board while preventing the computer from moving its pieces to your side of the board.

```
100 REM *** KONO-5 *** BY David Mercer, UK; adapted for the TI by Jeffrey Brown, Ottawa !057
110 DIM PM(4):: CALL CLEAR :
: CALL MAGNIFY(3):: CALL SCREEN(11):: FOR A=1 TO 8 :: CALL COLOR(A,2,11):: NEXT A !068
120 DISPLAY AT(1,11):"KONO-5" !243
130 ! WHITE BLOCKS !002
140 CALL CHAR(96,"FFFFC0C0C0C0",97,"FFFF",98,"C0C0C0C0C0C0C0C0")! TOP/LEFT, TOP, LEFT BOARD DIVIDER BLOCKS !071
150 CALL CHAR(99,"00")! EMPTY BLOCK !201
160 DISPLAY AT(3,1):"OBJECT
```

```
OF THE GAME:" !024
170 ! CIRCULAR SPRITE DEFS !049
180 CALL CHAR(100,"000000000010307070F0E0E1E1C1C1C0000000F7FFFF8C08000000000000000")! TOP/LEFT CIRCLE SPRITE !255
190 DISPLAY AT(5,1):"TRY TO GET ALL OF YOUR WHITE" !194
200 CALL CHAR(104,"1C1C1C1E0E0E0F070703010000000000000000000080C0F8FF7F0F0000")! BOTTOM/LEFT CIRCLE SPRITE !249
210 DISPLAY AT(6,1):"MEN TO THE OTHER SIDE OF THE" !132
220 CALL CHAR(108,"000000F8FFF0F010000000000000000000000000C0E0F0F07838383C1C1C1C")! TOP/RIGHT CIRCLE SPRITE !118
230 DISPLAY AT(7,1):"BOARD WHILE PREVENTING THE" !102
240 CALL CHAR(112,"000000000000000000010FFFFF800001C1C1C3C383878F0F0E0C000000000")! BOTTOM/RIGHT CIRCLE SPRITE !103
250 DISPLAY AT(8,1):"COMPUTER FROM GETTING ITS" !071
260 ! YELLOW CHARS !016
270 CALL CHAR(116,"0055AA55A
```

```
A55AA")! THINKING BLOCK !141
280 CALL CHAR(117,"FFFF",118,"C0C0C0C0C0C0C0C0",119,"C0C0")!165
290 DISPLAY AT(9,1):"BLACK PIECES TO YOUR SIDE." !031
300 ! YELLOW ARROW AND "PIECE CHOSEN" SPRITES !017
310 CALL CHAR(120,"000000000103070F1F3F7F7F7F00000000000080C0E0F0F8FCFEFFFFFFF000000")! YELLOW ARROW SPRITE !017
320 DISPLAY AT(11,1):"MOVING:" !101
330 CALL CHAR(124,"0000000000103070707030100000000000000000C0E0F0F0E0C0000000000")! PIECE CHOSEN SPRITE !159
340 DISPLAY AT(13,1):"MOVE THE CURSOR USING THE" !076
350 ! BLACK PIECE !134
360 CALL CHAR(128,"071F3F7FFF7F7F",129,"FFFFFFFF7F7F3F1F",130,"F0FCFEFFFFFFF",131,"FFFFFFFFFFFFEFC")!114
370 DISPLAY AT(14,1):"ARROW KEYS AND PRESS THE" !247
380 ! WHITE PIECE !170
390 CALL CHAR(132,"0718204040808080",133,"8080808040402018",134,"F00C020101000000",135,"000000000101020C")!104
```

(See Page 9)

## LIMA—

(Continued from Page 7)

wanting to build their own AMS card on a protoboard. According to Krych, the AMS card is unique in that instead of pulling data into a page frame the computer uses a pointer to point to where the data is located allowing for very high speed access.

Ricky Bottoms of RBD Enterprises had a variety of hardware and parts including even a Myarc Personality card. He noted that he also repairs TI products and can special order parts.

Other well-known Tiers who were present included Mike Maksimik, Jeff White,

Dan Eicher, Barry Traver, Bill Lucid and James Schroeder. Incidentally, Lucid showed the shareware version 5.01 of Edward Swartz's TI Emulator for PCs. I took a look at this new TI emulator and was impressed.

Numerous user groups attended the conference, including the Mid-South TI99/4A Users Group of Memphis, Tennessee, Chicago TI users group, Great Lakes Computer group, K-Town of Knoxville, Tennessee, TI Users of Will County, Spirit of 99 of Louisville, Akron 99'ers of Akron, Ohio, Hoosier, Kawartha

of Canada, West Penn 99'ers, Pittsburgh 99'ers, Cin-Day, C.O.N.N.I., Indianapolis, Indiana, group, not to overlook the Lima, Ohio, users group.

The entire day was busy with seminars running continuously in two separate rooms, while the other part of the fair went on in three connecting rooms.

In closing, this was my first time to attend the Lima Multi User Group Conference and I was very pleased with the organization not to mention the free pizza party at the end! I congratulate Charles Good and the Lima user group for a job well done.



## KONO—

(Continued from Page 8)

```

400 DISPLAY AT(15,1):"<SPACE
> BAR TO SELECT IT." !218
410 ! COMMON ELEMENTS !233
420 CALL CHAR(136,"000000000
0808080",137,"80808080",138,
"07",139,"F0")!064
430 DISPLAY AT(16,1):"THEN M
OVE TO THE DESIRED":"LOCATIO
N TO MOVE TO, AND":"PRESS IT
AGAIN. A MOVE MAY" !017
440 DISPLAY AT(19,1):"BE CAN
CELLED WITH <BACK> AT":"ANY
TIME BEFORE IT IS":"EXECUTED
. YELLOW TRIANGLES" !073
450 DISPLAY AT(22,1):"INDICA
TE VALID MOVES.":"":"PRESS A
NY KEY TO BEGIN" !008
460 CALL KEY(0,K,S):: IF S<1
THEN 460 ELSE CALL CLEAR !1
01
470 CALL CLEAR :: CALL COLOR
(9,2,16,11,2,11,13,2,16,14,2
,16):: DISPLAY AT(1,11):"KON
O-5" !005
480 ! DISPLAY BOARD !065
490 FOR A=3 TO 22 STEP 4 ::
CALL HCHAR(A,2,96):: DISPLAY
AT(A,1):RPT$( "aaa`",4)&"aaa
v" !059
500 CALL HCHAR(A+1,2,98):: D
ISPLAY AT(A+1,1):RPT$( "cccb"
,4)&"cccv" !042
510 CALL HCHAR(A+2,2,98):: D
ISPLAY AT(A+2,1):RPT$( "cccb"
,4)&"cccv" !044
520 CALL HCHAR(A+3,2,98):: D
ISPLAY AT(A+3,1):RPT$( "cccb"
,4)&"cccv" :: NEXT A !135
530 CALL HCHAR(A,2,117,2)::
DISPLAY AT(A,1):RPT$( "uuuu",
4)&"uuuw" !011
540 ! DISPLAY PIECES !146
550 W$="BBBBBB B W W
WWWWW" :: FOR A=0 TO 24 :: A
$=SEG$(W$,A+1,1):: IF A$=" "
THEN 590 !042
560 R=INT(A/5)*4+4 :: C=(A-I
NT(A/5)*5)*4+3 :: IF A$="W"
THEN B=132 ELSE B=128 !125
570 CALL HCHAR(R,C,B):: CALL
HCHAR(R,C+1,B+2):: CALL HCH
AR(R,C+2,136):: CALL HCHAR(R
+1,C,B+1)!058
580 CALL HCHAR(R+1,C+1,B+3):
: CALL HCHAR(R+1,C+2,137)::
CALL HCHAR(R+2,C,138):: CALL
HCHAR(R+2,C+1,139)!186
590 NEXT A !215
600 CALL DISP(4,"WHO STARTS"
):: CALL DISP(5,"FIRST?")::
CALL DISP(7," PLAYER"):: CA
LL DISP(9," OR"):: CALL D
ISP(11,"COMPUTER?")!106
610 CALL DISP(14," P OR C?")
!138
620 CALL KEY(0,K,S):: IF S<1
THEN 620 ELSE IF K=80 THEN
PL=0 ELSE IF K<>67 THEN 620
ELSE PL=1 !185
630 ! INIT VARIABLES !144
640 LR=4 :: LC=0 !033
650 ! BEGIN GAME !066
660 IF SEG$(W$,1,6)="WWWWW"
AND SEG$(W$,10,1)="W" THEN
W=1 ELSE IF SEG$(W$,20,6)="B
BBBB" AND SEG$(W$,16,1)="B"
THEN W=2 ELSE W=0 !106
670 IF W=0 THEN CALL DELSPRI
TE(ALL):: IF PL=1 THEN 890 E
LSE 720 !033
680 CALL VCHAR(1,23,32,240):
: IF W=1 THEN CALL DISP(9,"
YOU")ELSE CALL DISP(9," CO
MPUTER")!098
690 IF W=1 THEN CALL DISP(11
," WON!!!")ELSE CALL DISP(1
1," WINS!!!")!134
700 CALL DISP(14," PLAY"):
: CALL DISP(15," AGAIN?")::
CALL DISP(17," Y OR N")!10
5
710 CALL KEY(0,K,S):: IF S<1
THEN 710 ELSE IF K=89 THEN
470 ELSE CALL CLEAR :: CALL
CHARSET :: DISPLAY AT(12,10)
:"GOOD BYE!" :: STOP !102
720 ! PLAYER MOVE ROUTINE !0
13
730 PL=1 :: CALL VCHAR(1,23,
32,240):: CALL DISP(11,"PLAY
ER'S"):: CALL DISP(13,"MOVE"
):: CALL BEEP :: R=LR :: C=L
C :: CALL MOVE(R,C)!251
740 CALL KEY(0,K,S):: IF S=0
THEN 740 ELSE IF K<>14 THEN
IF K=32 THEN 820 ELSE 790 !
167
750 CALL DELSPRITE(ALL):: CA
LL VCHAR(1,23,32,240):: CALL
DISP(10,"QUIT GAME?")!005
760 CALL DISP(12," SURE?")
:: CALL DISP(14," Y OR N")!
148
770 CALL KEY(0,K,S):: IF S<1
THEN 770 ELSE IF K=78 THEN
730 ELSE CALL CLEAR :: CALL
CHARSET !049
780 DISPLAY AT(12,10):"GOOD
BYE!" :: STOP !243
790 R2=R :: C2=C :: R=R-(K=6
9)*(R<>0)+(K=88)*(R<>4):: C=
C-(K=83)*(C<>0)+(K=68)*(C<>4
):: IF R2=R AND C2=C THEN CA
LL HONK :: C=C2 :: R=R2 :: G
OTO 740 !159
800 CALL BEEP :: CALL MOVE(R
,C):: GOTO 740 !019
810 ! SELECT PIECE !233
820 IF SEG$(W$,R*5+C+1,1)<>"
W" THEN CALL HONK :: GOTO 74
0 !020
830 LR=R :: LC=C :: CALL SPR
ITE(#9,124,2,LR*32+25,LC*32+
17):: CALL ARROW(LR,LC,W$)!0
01
840 CALL KEY(0,K,S):: IF S=0
THEN 840 ELSE IF K=15 THEN
CALL DELSPRITE(ALL):: GOTO 7
30 ELSE IF K=14 THEN 750 ELS
E IF K=32 THEN 880 !108
850 R2=R :: C2=C :: R=R-(K=6
9)*(R<>0)+(K=88)*(R<>4):: C=
C-(K=83)*(C<>0)+(K=68)*(C<>4
):: IF R2=R AND C2=C THEN CA
LL HONK :: C=C2 :: R=R2 :: G
OTO 840 !004
860 CALL BEEP :: CALL MOVE(R
,C):: GOTO 840 !120
870 ! MOVE THE PIECE NOW!!!
!216
880 A=1 :: CALL VALID(LR,LC,
R,C,W$,A):: IF A=1 THEN CALL
HONK :: GOTO 840 ELSE CALL
MOVEP(LR,LC,R,C,W$):: LR=R :
: LC=C :: GOTO 660 !180
890 ! COMPUTER MOVE ROUTINE
!175
900 Z,PL=0 :: CALL VCHAR(1,2
3,32,240):: CALL DISP(11,"CO
MPUTER'S"):: CALL DISP(13,"M
OVE"):: CALL BEEP !146
910 ! PROCEDURE AUTOKEY !142
920 BM=0 :: GOSUB 1100 ! PIC
(See Page 10)

```

## KONO—

(Continued from Page 9)

```

K BEST MOVE !215
930 MF=BM :: IF MF=0 THEN RANDOMIZE ELSE 990 !160
940 MF=INT(RND*25)+1 :: GOSUB B 1210 ! VALIDATE MOVE (COMP
UTER USES OTHER VALUES) !185
950 IF PM(1)=0 AND PM(2)=0 AND PM(3)=0 AND PM(4)=0 THEN
920 !187
960 IF MF>=1 AND MF<=25 AND PM(1)>0 OR PM(2)>0 AND PM(3)
>0 OR PM(4)>0 THEN 970 ELSE
940 !174
970 IF SEG$(W$,MF,1)="B" AND MF<>MT THEN 990 ELSE 940 !1
92
980 ! NOW BACK IN MAIN PROCEDURE !141
990 GOSUB 1210 ! VALIDATE MOVE !082
1000 ! AUTOBLACKKEY !034
1010 BMTM=0 :: BM=0 :: FOR A=1 TO 4 :: IF PM(A)>BM THEN
BMTM=A !227
1020 NEXT A :: MT=PM(BMTM)!242
1030 ! BACK AGAIN! !085
1040 ! REDUNDANT LINE!!! WORTHLESS!! !032
1050 GOSUB 1210 ! VALIDATE MOVE !082
1060 IF MT=PM(1)OR MT=PM(2)OR MT=PM(3)OR MT=PM(4)THEN 1070
ELSE 1010 !046
1070 IF MF=MT THEN 920 !138
1080 CALL MOVEP(INT((MF-1)/5),MF-INT((MF-1)/5)*5-1,INT((MT-1)/5),MT-INT((MT-1)/5)*5-1,W$)!091
1090 GOTO 660 !229
1100 ! PROCEDURE PICKBESTMOVE !248
1110 FOR PB=1 TO 25 :: IF SEG$(W$,PB,1)<>"B" THEN 1190 ELSE MF=PB :: GOSUB 1210 ! VALIDATE MOVE !084
1120 FOR NM=1 TO 4 :: IF PM(NM)>PB THEN BM=PB !025
1130 NEXT NM !049
1140 IF PB<>16 AND PB<20 THEN 1150 ELSE 1190 !237
1150 IF (PM(1)=16)+(PM(1)>19)*(PM(1)<26)THEN BM=PB !190
1160 IF (PM(2)=16)+(PM(2)>19)*(PM(2)<26)THEN BM=PB !193
1170 IF (PM(3)=16)+(PM(3)>19)*(PM(3)<26)THEN BM=PB !196
1180 IF (PM(4)=16)+(PM(4)>19)*(PM(4)<26)THEN BM=PB !199
1190 NEXT PB :: MF=BM :: RETURN !148
1200 ! PROCEDURE VALIDATEMOVE !237
1210 IF MF=0 THEN PM(1),PM(2),PM(3),PM(4)=0 :: RETURN !110
1220 A=1 :: FOR Q=1 TO 4 :: PM(Q)=0 :: NEXT Q :: R=INT((MF-1)/5) :: C=(MF-1)-INT((MF-1)/5)*5 :: CALL HCHAR(15,23+Z,116) :: Z=Z+.1 :: IF Z>9 THEN Z=9 !079
1230 IF R<>0 AND C<>0 THEN IF SEG$(W$,MF-6,1)=" " THEN PM(A)=MF-6 :: A=A+1 !156
1240 IF R<>0 AND C<>4 THEN IF SEG$(W$,MF-4,1)=" " THEN PM(A)=MF-4 :: A=A+1 !156
1250 IF R<>4 AND C<>0 THEN IF SEG$(W$,MF+4,1)=" " THEN PM(A)=MF+4 :: A=A+1 !154
1260 IF R<>4 AND C<>4 THEN IF SEG$(W$,MF+6,1)=" " THEN PM(A)=MF+6 !037
1270 RETURN !136
1280 SUB MOVE(R2,C2):: CALL DELSPRITE(#1,#2,#3,#4):: R=R2*32+17 :: C=C2*32+9 !105
1290 CALL SPRITE(#1,100,2,R,C,0,0,#2,104,2,R+16,C,0,0,#3,108,2,R,C+16,0,0,#4,112,2,R+16,C+16)!105
1300 SUBEND !168
1310 SUB DISP(R,A$):: DISPLAY AT(R,21):SEG$(A$,1,8):: A$=A$&" " :: CALL HCHAR(R,31,ASC(SEG$(A$,9,1)))!180
1320 CALL HCHAR(R,32,ASC(SEG$(A$,10,1))):: SUBEND !204
1330 SUB MOVEP(R1,C1,R2,C2,W$)! ASSUMES A VALID MOVE HERE! !010
1340 CALL DELSPRITE(ALL):: A$=SEG$(W$,R1*5+C1+1,1):: W$=SEG$(W$,1,R1*5+C1)&" "&SEG$(W$,R1*5+C1+2,255)!218
1350 W$=SEG$(W$,1,R2*5+C2)&A$&SEG$(W$,R2*5+C2+2,255):: C
ALL HCHAR(R1*4+6,C1*4+3,99,2)!112
1360 CALL HCHAR(R1*4+4,C1*4+3,99,3):: CALL HCHAR(R1*4+5,C1*4+3,99,3):: R=R2*4+4 :: C=C2*4+3 :: IF A$="B" THEN B=128 ELSE B=132 !183
1370 CALL HCHAR(R,C,B):: CALL HCHAR(R,C+1,B+2):: CALL HCHAR(R,C+2,136):: CALL HCHAR(R+1,C,B+1)!058
1380 CALL HCHAR(R+1,C+1,B+3):: CALL HCHAR(R+1,C+2,137):: CALL HCHAR(R+2,C,138):: CALL HCHAR(R+2,C+1,139)!186
1390 SUBEND !168
1400 SUB ARROW(R,C,W$):: CALL DELSPRITE(#5,#6,#7,#8):: P=R*5+C+1 :: R2=R*32+25 :: C2=C*32+17 :: IF R=0 OR C=0 THEN 1420 !246
1410 IF SEG$(W$,P-6,1)=" " THEN CALL SPRITE(#5,120,11,R2-32,C2-32)!230
1420 IF R=0 OR C=4 THEN 1430 ELSE IF SEG$(W$,P-4,1)=" " THEN CALL SPRITE(#6,120,11,R2-32,C2+32)!190
1430 IF R=4 OR C=0 THEN 1440 ELSE IF SEG$(W$,P+4,1)=" " THEN CALL SPRITE(#7,120,11,R2+32,C2-32)!200
1440 IF R=4 OR C=4 THEN SUBEXIT ELSE IF SEG$(W$,P+6,1)=" " THEN CALL SPRITE(#8,120,11,R2+32,C2+32)!007
1450 SUBEND !168
1460 SUB BEEP :: CALL SOUND(100,1000,0):: SUBEND !221
1470 SUB HONK :: CALL SOUND(100,219,0):: SUBEND !203
1480 SUB VALID(R1,C1,R2,C2,W$,A):: IF ABS(R1-R2)<>1 OR ABS(C1-C2)<>1 THEN A=1 :: SUBEXIT !222
1490 A$=SEG$(W$,R2*5+C2+1,1):: IF A$<>" " THEN A=1 :: SUBEXIT !029
1500 A$=SEG$(W$,R1*5+C1+1,1):: IF (A=1)*(A$<>"W")+(A=2)*(A$<>"B")THEN A=1 :: SUBEXIT !046
1510 A=0 :: SUBEND !034

```

## THE ART OF ASSEMBLY — PART 36

# We keep learning

By **BRUCE HARRISON**  
©1994 B. Harrison

We've all seen advertisements from institutions of higher learning for their "continuing education" programs. Many of these are aimed directly at people like your author, too old to start over, and too tired to go for the doctorate degrees, yet willing and able to learn some new skill or other. We can resist the temptation to pay anyone money to teach us new things, because we learn new tricks every day in the business of assembly, and that's enough "continuing education" for us.

Sometimes the things we learn are those that we "should have known" all along, but never did learn because we were not paying enough attention, or just failed to make a connection among various facts. Today we'll start with one of those cases.

### DO WE FEEL STUPID?

Yes, today we feel stupid. This is to correct ourselves on something stupid that we did in this column, dating all the way back to Part 8 of the series. In Part 8, we discussed the detection and handling of file errors. In that column, we boldly asserted that the way we used to find an "open" error was the only sure way to do so. Sometimes it pays to go back to the fundamentals, and that's what we did in writing our mini-series for beginners. To briefly refresh your memory, what we did in Part 8 was to show how to detect an error when opening a file by doing this:

```
BLWP @DSRLNK use DSR linkage
DATA 8 requ'd data
STST R14 get status register in R14
ANDI R14,>2000 mask all but Equal bit
JEQ OPNOK if zero, file opened
B @OPNERR else an error in open operation
```

OPNOK (program continues)

**DUMB! That was really DUMB! STUPID! IDIOTIC! WASTEFUL!** The lines with STST and ANDI were not necessary. This accomplishes the same thing:

```
BLWP @DSRLNK
DATA 8
JNE OPNOK if not equal, okay
B @OPNERR else an error in open operation
```

OPNOK (program continues)

All we really needed was that JNE instruction. The DSR linkage routine sets the status register for us by setting bits in its own R15 before executing its RTWP instruction. If an error of any kind has occurred in the DSR operation, the "EQUAL" bit will be turned on, else that bit will be off. Thus JNE is all we needed to do, since having the equal bit turned off means that no error occurred.

We would perhaps never have thought of trying this had it not been for the exercise of thinking through just exactly how the conditional jumps use the status register. We were forced to do that thinking to explain to beginners what those jumps are all about in

detail. Thanks to those of you who asked for a "beginners" lesson, your author learned something himself. Who says "You can't teach an old dog...?" This 53-year-old dog is still learning things.

To prove this to ourselves, we went back to the test routines that were used in preparing Part 8, made the change shown above, and ran some more tests. This worked perfectly. This same test can detect errors in any DSRLNK operation, including read or write operations. Finding the kind of error still requires looking at the PAB+1 byte from VDP RAM, but detecting the presence of an error is as simple as the second snippet shown above. In those new tests, we used the TI DSRLNK that's built into the Editor/Assembler module's utilities, and also the Don Warren/Craig Miller DSRLNK routine that's used when we're operating from Extended BASIC CALL LINK. In both situations, this worked perfectly.

For your convenience, we've put this new method into this month's sidebar, along with the updated error reporting method shown earlier in Part 28 of this series, and even slightly modified that to improve its memory efficiency. The example as given opens a D/V80 file for input, but of course it will work every bit as well for other file types depending on what's in the PAB data. (See part 9 of this series for data on using other file types.)

### THE FREE EYE TEST

Back when your author was "gainfully employed," we used to joke about our employer having offered a free eye test, but "I didn't see the notice." (I also never heard the announcement on the public address system about the free hearing test.) In some ways, the E/A manual is an "eye test." There are things in that book that one may not notice until the 50th reading. Such an experience happened recently, again as part of our work on the "beginner" series.

Some time back, we offered some advice about taking the keystroke from a single keypress into a register, so that it can be compared to various numbers. The example we were showing had to do with Yes/No keypress situations. The recommendation was that the contents of location KEYVAL (>8375) be put into R8 as part of the key input subroutine, then the main code could test for Y or y like this:

```
CI R8,89 compare to ASCII for u.c. Y
JEQ YES
CI R8,110 compare to ASCII for l.c. y
JEQ YES
(else answer is taken as no)
YES (answer was Y or y for yes)
```

What we didn't notice was that we didn't really have to remember the ASCII codes for Y and y. We could have written our source code this way:

```
CI R8,'Y'
JEQ YES
CI R8,'y'
JEQ YES
(and so on)
```

(See Page 12)

# THE ART OF ASSEMBLY—

(Continued from Page 11)

That simple trick was right there in front of our eyes in the part of the manual dealing with the DATA directive, and we should have seen it and connected it to the immediate instructions. What will happen in the above cases is that the assembler will make the immediate values 89 and 110 for us. (We had done this in PC assembly language, and it worked there, but had never thought of trying it on our TI.)

## GETTING EVEN

In most cases, there is no difference between the following two ways of labeling an item:

```
THREE DATA 3
FOUR
```

```
DATA 4
```

There is usually no difference between putting the label on the same line as the directive or instruction, or putting the label on a line by itself. We used that word usually on purpose, because we're about to show an exception. If the location count was left at an odd address before the DATA directive, then there's a difference. Suppose we had the following two cases:

```
THREE DATA 3
ANYKEY BYTE >20
FOUR DATA 4
THREE DATA 3
ANYKEY BYTE >20
FOUR
DATA 4
```

In the first of these two cases, the label four will be assembled as an even address because it defines a word in the memory by the DATA directive. In the second case, the label FOUR will be assembled as the odd address one byte beyond ANYKEY. There will, in both cases, be a "wasted" byte between the byte that's set to >20 and the word that's set at the value 4, but the address of the label FOUR will be different. Let's say that the label ANYKEY is at address >A04E. The memory contents would look exactly the same for both cases: >A04E contains >20, >A04F contains 0, >A050 contains 0, and >A051 contains 4. What's different is that, in the first case the label FOUR is the address >A050, while in the second case that label will be >A04F. This will make a terrible difference if we then use an instruction like this:

```
MOV @FOUR,R3
```

If our data was set up as the first example above, the value placed in R3 will be 4, which of course was our intent. In the second case, the computer will notice that this is a word move, and that the address given is an odd number. It will therefore take the two bytes starting at the next lowest address into R3, so this time R3 will be loaded with the two bytes starting at >A04E, and its value will become >2000. That's very definitely not what we intended.

Having been bitten once by this problem in our Code Breakers program, we are very shy about this kind of problem. There is one other solution to this problem, and that is to put in the EVEN directive before the label FOUR, like this:

```
THREE DATA 3
ANYKEY BYTE >20
```

(See Page 13)

## Sidebar 36

\* SIDEBAR 36

\*

\* PART ONE

\*

\* FILE ERROR HANDLING RE-REVISITED

\* THIS IS NOT COMPLETE CODE, JUST SNIPPETS

\* SIMPLER METHODS FOR BOTH ERROR

\* DETECTION AND ERROR REPORTING

\* CODE BY B. HARRISON

\* PUBLIC DOMAIN

\*

\* THIS OPENS AND READS A FILE RECORD

\*

```
OPNF LI R0,PAB POINT TO PAB IN VDP
LI R1,PABDT AND PAB DATA
MOVB @PABDT+9,R2 GET DESCRIPTOR LENGTH BYTE
SRL R2,8 RIGHT JUSTIFY IN R2
AI R2,10 ADD TEN FOR THE PAB ITSELF
BLWP @VMBW WRITE PAB DATA
AI R0,9 ADD NINE
MOV R0,@>8356 PLACE ADDRESS
BLWP @DSRLNK USE DSR LINKAGE
DATA 8 REQUIRED DATA
JNE RDFI IF NOT "EQUAL", OKAY
B @OPNERR ELSE REPORT OPEN ERROR
RDFI MOVB @READF,R1 SET TO READ
LI R0,PAB POINT AT PAB
BLWP @VSBW WRITE BYTE
AI R0,9 ADD NINE
MOV R0,@>8356 PLACE ADDRESS
BLWP @DSRLNK USE DSR
DATA 8 REQUIRED DATA
JNE READON IF NOT "EQUAL", OKAY
B @FILERR ELSE REPORT ERROR
READON (PROGRAM CONTINUES)
*
*
*
OPNERR
LI R0,22*32+6 POINT AT ROW 23, COL 7
LI R1,FNOTXT FILE NOT OPENED
LI R2,17 17 BYTES TO WRITE
BLWP @VMBW WRITE MESSAGE
FILERR LI R0,PAB+1 POINT AT PAB PLUS ONE
BLWP @VSBW READ A BYTE
SRL R1,13 SHIFT RIGHT 13 BITS
*
*
* AT THIS POINT YOU MIGHT WANT TO INSERT A CI R1,5
* AND JEQ TO SOMEPLACE ELSE FOR AN END-OF-FILE ERROR
*
SLA R1,4 SHIFT LEFT FOUR BITS (MULTIPLY
BY 16)
AI R1,FERMSG ADD START OF MESSAGE TABLE
LI R2,16 16 BYTES IN EACH MESSAGE
LI R0,23*32+7 POINT AT ROW 24, COL 8
BLWP @VMBW WRITE MESSAGE
BL @KEYLOO PAUSE FOR KEYSTROKE
B (SOMEWHERE ELSE)
*
* DATA SECTION
*
PABDT DATA >0014,BUF,>5050,>0000,>000F
TEXT 'DSK1.ANYOLDFILE'
*
FERMSG TEXT 'BAD DEVICE NAME ' EACH TEXT LINE 16
```

# THE ART OF ASSEMBLY—

(Continued from Page 12)

```

EVEN
FOUR
DATA 4

```

Once again, the results in actual memory locations is the same as we showed earlier, except that the EVEN directive makes the assembler's location counter advance to >A050 before it sets the location value for the label FOUR, so this method will result in correct operation for our MOV instruction.

The lesson in all this is that the EVEN directive is almost never necessary. Just making sure that the DATA or BYTE or TEXT directive is in the same source line as its label will insure that the labels which should be at even locations will in fact be assembled as even addresses. In this month's sidebar are partial listings from the assembler, showing the three cases in detail, so you can see for yourself how the addressing works out.

As we've said earlier, our normal practice is to separate all of the data in our programs into one place in the source code, and that's usually at the end of the source, after all the code parts. Within that data section, we normally group all the DATA directives together at the beginning of the data section, so that we need never worry about there being an odd byte made into a wasted byte of zero value. (We also put any BSS directives that have even-numbered arguments into that same part of the data section, as for example our workspace as WS BSS 32.)

To be fair to TI, they did mention this very important difference between the label-only line and the label being on the same line as the DATA directive, but again this was not easily noticed where it was buried in that giant book. (It's in the last paragraph on page 47, after a "However.")

## THE CHALLENGE

This is number 36 of The Art of Assembly, which means that three years' worth of these columns have been written. It's hard to imagine how this can be, since it seems like only yesterday when we were writing Part 1. The challenge for someone who's been doing this for so long is to keep the column both interesting and relevant for the practitioners of this strange and difficult "art." Much material has been covered, and we are still learning things and passing those along to our readers, but we need some more "feedback" to keep going. Your author avidly reads the Reader Feedback column in MICROpendium, as well as the "Reader to Reader" column. That's how the "beginners" parts were inspired. If you're looking for something on our subject, and are too shy to have your concerns aired in print, you can always write to us directly at 5705 40th Place, Hyattsville MD 20781. We answer all letters, and will make every effort to find a solution to your problem. We may not always succeed, but we'll always try.

Your best chance to reach MICROpendium phone is on Saturday mornings. Call us at 512-255-1512 from 9 a.m. to noon. At other times you can leave a message on our answering machine.

BYTES

```

TEXT 'WRITE PROTECTED '
TEXT 'BAD ATTRIBUTE '
TEXT 'BAD OPERATION '
TEXT 'DISK IS FILLED '
TEXT 'END OF FILE '
TEXT 'DEVICE ERROR '
TEXT 'OTHER FILE ERROR '

```

FNOTXT TEXT 'FILE DID NOT OPEN' 17 BYTES LENGTH

\*

\*

\* SIDEBAR PART TWO

\* ASSEMBLY LISTINGS TO SHOW THE "LONE LABEL"

\* EFFECT ON A DATA ITEM'S ADDRESS

\* (THESE ARE PARTIAL LISTINGS, EDITED FOR CLARITY)

\*

\* EXAMPLE ONE - WITH LABEL AND DIRECTIVE

\* ON THE SAME SOURCE LINE

\* (THE RIGHT WAY)

LOCATION CONTENT LABEL OPCODE/DIRECTIVE

```

0048 C0E0 MOV4 MOV @FOUR,R3
004A 0050 '

```

\*

\* NOTE THAT LABEL FOUR IS AN EVEN ADDRESS (>50)

\* AND EQUALS THE LOCATION OF THE DATA ITSELF

\*

```

004C 0003 THREE DATA 3
004E 20 ANYKEY BYTE >20
0050 0004 FOUR DATA 4

```

\* EXAMPLE TWO - WITH LABEL ON SEPARATE

\* SOURCE LINE FROM THE DIRECTIVE

\* (THE WRONG WAY)

LOCATION CONTENT LABEL OPCODE/DIRECTIVE

```

0048 C0E0 MOV4 MOV @FOUR,R3
004A 004F '

```

\*

\* NOTE THAT LABEL FOUR IS AN ODD ADDRESS (>4F)

\* AND NOT THE ADDRESS OF THE DATA ITSELF

\*

```

004C 0003 THREE DATA 3
004E 20 ANYKEY BYTE >20

```

FOUR

```

0050 0004 DATA 4

```

\* EXAMPLE THREE - WITH LABEL ON SEPARATE

\* SOURCE LINE FROM THE DIRECTIVE

\* BUT USING EVEN DIRECTIVE (ANOTHER RIGHT WAY)

LOCATION CONTENT LABEL OPCODE/DIRECTIVE

```

0048 C0E0 MOV4 MOV @FOUR,R3
004A 0050 '

```

\*

\* NOTE THAT LABEL FOUR IS AN EVEN ADDRESS (>50)

\* AND EQUALS THE ADDRESS OF THE DATA ITSELF

\*

```

004C 0003 THREE DATA 3
004E 20 ANYKEY BYTE >20

```

EVEN

FOUR

```

0050 0004 DATA 4

```

## Living with spiders — Part III

# Programming with Funnelweb

By TONY McGOVERN

The previous installments have been on "dual-mode" programs which have to run with or without FWB. Now we'll look at a few aspects of writing programs to work only with FWB.

The first thing is access to GPL routines. If the FWB E/A utilities are loaded then a normal BLWP call to GPLLNK works as usual. The various XML address possibilities should be respected, particularly the >F0 XML at >8300 which is used by the TI-Writer and Mini-mem modules, or from console GROMs. If the E/A utilities are not loaded a briefer than normal form of GPLLNK may be used. The following routine is used to call the beeps and bleeps in DPatch.

```
* FWB system equates
SETGRM EQU >FF28
MODFL EQU >FF5A
*GPLLNK BLWP vector
GPLLNK DATA GPWS,GPLK
GPLK EQU $

MOV @MODFL,R0
LI R1,GPLRT
MOV R1,*R0
MOV *R14+,@>83EC
LWPI GPLWS
B @>0060 Go to GPL
GPLRT LWPI GPWS
MOV @SETGRM,R1
BL *R1
RTWP
```

Here the XML vector has already been determined by the FWB entry code and its target address stored at MODFL (>FF5A). If the E/A GROM is present MODFL will contain the value >2002 for XML >21 as example. The re-entry address is GPLRT is written into the XML address, without bothering to save the existing contents. If you do use >8300 you had better save and restore it though. The GPLLNK data item is written to R6 of the GPL workspace, and then transfer made to GPL. When the linkable GPL routine has finished it returns to GPLRT. SETGRM (>FF28) contains a pointer to a BL routine in FWB which resets all the GPL items. This routine uses only R0. One subtlety is that the GPL stack pointer at >8373 is reset to >80 by the SETGRM routine. The FWB cartridge loader from UL sets it to >7E to help with key-unit problems apparent in some programs. There is no reason why it has to be a BLWP routine and the code could be recast in other forms. FWB contains a similar routine to handle cassette loading, but it is in-line code and not externally accessible.

FWB does however contain an externally accessible DSRLNK with BLWP vector at >FFD4. This routine takes no following data item, and returns with the status <equals> bit set if not found and it is otherwise up to you to dig out any error from the PAB or GPL status byte. This means that it is basically set up for file type access only. As an example here are the load and resave routines

from the CF/CG configuration program.

```
FILSVE EQU $
MOV @SAVEOP,R1
JMP FILOPS
FILOAD EQU $
MOV @LOADOP,R1

* Load-save routine
FILOPS EQU $
LI R0,PAB
BLWP @VSEW
LI R0,PAB+9
MOV R0,@>6356
SB @GPLST,@GPLST
BLWP @DSRLNK
JEQ FAILS
BLWP @VSBRD
DATA PAB+1
SRL R0,13
JNE FAILS
MOVB @GPLST,R0
COC @MASK,R0
JEQ FAILS
INCR11

FAILS RT
MASK DATA >2000
LOADOP BYTE 5
SAVEOP BYTE 6
```

Alternate entry points allow for save or load operations. It is assumed that the PAB and VDP buffers are already set up as needed. I'm never quite sure if the clearing the GPL status byte is strictly necessary but it can't hurt. The exit after the DSRLNK call is taken if the DSR is not found. This particular program uses a data VSBRD to fetch the PAB error byte into R0 from fixed address in VDP. A normal VSBRD would allow a more flexible routine. If no error is indicated here after the first 3 bits have been isolated, the GPL status byte is then checked. The failure exit steps over the word following the BL call. Normally this would be a JMP to an error handler.

Suppose your program needs to do a sector access via a DSRLNK. You could of course use the E/A utility routine which takes a following data item. Just to be different we'll work through the sector reading routine used in QD. The E/A routine is not available so the FWB DSRLNK has to be modified to suit. This is done by temporarily rewriting an internal item from >8 to >A on entry and restoring it on exit. The routine also illustrates the FWB method of coping with high-CRU Horizon RAMdisks. The problem here is that the sector read/write routine has of necessity same name >10 in all disk DSRs and if the drive number doesn't match an error is returned. This means that a normal DSRLNK won't get past the usual disk controller at CRU >1100 (the Myarc

(See Page 14)

## LIVING WITH SPIDERS—

(Continued from Page 13)

RAMdisk at CRU >1000 handles this transparently for normal disk controller accesses). Some programmers solve this problem by using a DSRLNK which starts its CRU search at >1200 which finally loops around to check >1000 and >1100. Atrax Robustus doesn't care for this method as it puts a permanent slug on normal disk access, and insisted on doing it another way which will become apparent.

In the code to follow, extracted from the QD source, only sector reads are performed and any non-null data byte will do for READ. The style of coding reflects a situation where registers are in short supply and length is critical. The drive number is at DR#, the VDP sector buffer address at PBF, and the sector number in R7. The code uses routines available in QD which in turn were written to match TI-Writer routines used by SD. You could do things differently if you chose. The offset values >56, >5C, and >7A refer to particular address offsets into the FWB DSRLNK routine. These will be maintained at their present values, even though I am continually resisting temptation to shave some bytes out of the DSRLNK.

```

DSRLNK EQU >FFD4
FAC EQU >834A
SCNAME EQU >8356
PN BYTE >02, >01, >10
B0F BYTE >0F
B1F BYTE >0F
CTRLC BYTE >83
      EVEN
SECRD EQU $
      MOV R11, R10
      MOV @DSRLNK+2, R11
      INCT @>7A(R11)
SECRN LI R9, PAB
      MOV R9, @SCNAME

      LI R8, PN
      BL @VSTRW
      LI R11, FAC+2
      MOVB @DR#, *R11+
      MOVB @READ, *R11+
      MOV @PBF, *R11+
      MOV R7, *R11
      BLWP @DSRLNK
      JEQ DSERR
      MOVB *R11, R12
      JEQ DSREX
DSERT MOV @DSRLNK+2, R11

```

**In some applications it might be preferable not to reset the CRU starting base immediately but this leads to messier code overall.**

```

A @>5C(R11), @>56(R11)
CB @>56(R11), @B1F
JL SECRN
DSERR EQU $
      BL @VSTRW
      DATA ERMSG
      DATA >2E6
HOLD BL @KSCAN
      CB R12, @CTRLC
      JEQ HLD10
      CB R12, @B0F
      JNE HOLD
HLD10 LI R10, EXIT
DSREX MOV @DSRLNK+2, R11
      DECT @>7A(R11)
      MOVB @B0F, @>56(R11)
      B *R10

```

On entry the return is saved in R10, and R11 used as a working register thereafter. The DSRLNK is then fetched from the BLWP vector and the word offset >7A into the routine is INCTed. This makes it equivalent to a DATA >A call to an E/A DSRLNK. We must then be careful to restore the value at every possible exit from the routine. VSTRW is a BL routine that writes a string with leading length byte from CPU to VDP. While at it we load the PAB pointer at >8356. The next few lines load the FAC area with all the necessary data for a sector read call and leave R11 pointing at >8350 where errors are returned. If the DSRLNK search finally fails, or has found the drive number but can't read the disk, the JEQ sends it to the exit error routine.

The DSR sector routine error byte is then checked. If this is null the sector read has been successful and a normal exit is taken. If it is not null then the sector routine has been executed but the drive number wasn't found at that CRU base and the special error handler at DSERT is entered. This resets R11 to the start of the DSRLNK code and adds the normal increment of CRU base search at offset >5C to the start address for the CRU search. The sector read routine is then re-entered and the whole process started again, but this time from the next CRU slot, so that the last error isn't repeated. Before this repeat is done the CRU base is checked to see if it has reached the end of the line. If so the normal error exit is taken. The virtue of this method is that is the error path that cops the penalties and not your Myarc RAMdisk at >1000 CRU base. I have also heard that it works better with multiple Horizons wit ROM DSRs.

The error exit writes up an error message using an in-line data entry point to the VSTRW routine and then waits for either <fctn-9> or <ctrl-C> to be pressed. The final return address is re-written in R10 as needed. All exits pass through DSREX where the CRU base search start is re-initialized and the DSR type reset for normal file access. In some applications it might be preferable not to reset the CRU starting base immediately but this leads to messier code overall. A good example is right here in in this very disk directory application where the same CRU base could be used until a whole directory worth of file header sectors had been read. In

(See Page 15)

## MY-BASIC

# Emulating the assembly BLINK

By JIM UZZELL  
©1994 DDI Software

The following assembly source code and object code is a demonstration that emulates the BLINK command in MY-BASIC and illustrates how powerful MY-BASIC really is. It is recommended that you have the V9938 manual in order to understand the setup of the VDP registers. The V9938 manual is available from Beery Miller.

Those who do not desire to assemble the source code can type in the object code in MY-WORD and save it as a fixed file. The note in the source code at clear color table can be seen by those that type in the object by changing line 10 to the following (there is a space before and after to highlight what to change, do not include spaces):

```
BOA00B 0000 B010EB0200B015DB0201 7 F326F BEFORE
BOA00B FFFF B010EB0200B015DB0201 8 F326F AFTER
```

The MY-BASIC program is in the header of the source code.

Also included with this article is a hexadecimal-to-decimal converter chart with examples. MY-BASIC has hex\$ and valhex commands which makes it easy to do conversions, but there will be times when you are not in MY-BASIC and this chart will come in handy.

```
*      80 column demo program (TEXT2-26 lines)      02-19-94
*      Copyright 1994 by DDI SOFTWARE
*      100 CALL GRAPHICS(4)
*      110 CALL INIT::CALL LOAD("DSKx.BLINK4")
*      120 CLS::CALL LINK("START")
*      DEF      START
*      REF      VMBW,VSBW, VWTR,VMBR
*      REF      VDPWA,VDPWD
KSCAN EQU >201C
MSTAND DATA >0003 CALL KEY MODE 3
MYWS EQU >F000
MSG1 TEXT 'TEXT2-26 MODE GRAPHICS(4)' 25
MSG2 TEXT 'with blinking' 13
MSG3 TEXT 'Copyright 1994 by DDI SOFTWARE' 30
MSG4 TEXT 'Press any key to continue' 25
*
EVEN
SAVRTN DATA 0000
START MOV R11,@SAVRTN
LWPI MYWS Use super-fast workspace registers
```

```
LIMI 0 Turn off interrupts before we do VDP access
*****
* Set VDP registers and clear tables *
*****
LI R0,>0004 VDP R0 = >04 80 col >00 40 col
BLWP @VWTR
*
LI R1,>0170 VDP R1 = >70
BLWP @VWTR
*
LI R0,>0203 VDP R2 = >03 (Pattern Name table 0
BLWP @VWTR at >1000) Or Screen Image in TI lingo
* 80*27=2160 bytes or >0000 to >0870
*
LI R0,>0402 VDP R4 = >02 (Pattern Generator
BLWP @VWTR table 0 at >1000)
*
LI R0,>032F VDP R3 = >2F (Color table 0 at >0A00)
BLWP @VWTR 80*27/8=270 bytes or >0A00 to >0B0E
LI R0,>0A00 VDP R10 = >00 (continuation of color
BLWP @VWTR table address)
*
LI R0,>0980 VDP R9 = >80 (26 1/2 lines, interlace
BLWP @VWTR off)
*****
*The above is only the first page of a possible 16*
*****
LI R0,>07F5 VDP R7 = >F5 (main color is white on
BLWP @VWTR 1t blue)
*
LI R0,>0C3D VDP R12 = >3D (blink color 1t green/
BLWP @VWTR magenta)
*
LI R0,>0D63 VDP R13 = >63 (blink color on 1
BLWP @VWTR second, main color on 1/2 second)
*
BL @VDPPFIL Clear screen
DATA >0000,>2000,27*80
*
BL @VDPPFIL Clear color table
DATA >0A00,>0000,27*80/8 Change >0000 to >FFFF and
*****
***** screen will blink and will
* Write all text to screen
* help explain the blink routine executed below
*****
LI R0,4*80+29
LI R1,MSG1
```

(See Page 17)

## LIVING WITH SPIDERS—

(Continued from Page 14)

fact this is done in the Editor's SD which uses generally similar code but it really only affects high-CRU Horizons.

Similar problems exist for volume name access to files where DSK. is also a DSR routine name common to all disk type DSRs. As yet the FWB main program code does not contain the special error handling routines as in FWB sector access code as there appears to have been no specific demand for volume-name access to high-CRU Horizon type RAMdisks. Most (the TI-Workshop cartridge is an exception) file-loader DSRLNKs don't support it ei-

ther. Incidentally you can observe the other type of enhanced DSRLNK in action if you have high-CRU Horizons installed. Watch the lights when DSKU (or TI-Workshop or Ottawa issue DM-1000) is accessing low CRU devices, say a Myarc RAMdisk, or regular disk controller. The high CRU Horizon lights will be flashing, but not with similar access by a FWB system program. In this same vein if you catalog a high-CRU Horizon with QD the regular disk controller light will flash with QD but not with SD after the first sector has been read.



# MY-BASIC

(Continued from Page 16)

```

LI R2,25
BLWP @VMBW

*
LI R0,6*80+35
LI R1,MSG2
LI R2,13
BLWP @VMBW

*
LI R0,8*80+25
LI R1,MSG3
LI R2,30
BLWP @VMBW

*
LI R0,25*80+28
LI R1,MSG4
LI R2,25
BLWP @VMBW
    
```

\*\*\*\*\*  
 \* Set up blinking \*  
 \*\*\*\*\*

```

LI R0,>A40 BASE + 64( >40 ) * 8 or 512
LI R1,>1E00 MSG2=515 (Bit=Char) So set these bits
BLWP @VSBW 000x xxxx x equates to "with" in MSG2
LI R0,>A41 + 8 more
LI R1,>FF00 xxxx xxxx = "blinking"
BLWP @VSBW
    
```

\*\*\*\*\*  
 \* Wait for keypress \*  
 \*\*\*\*\*

```

MOV @MSTAND,@>8374
WAIT BLWP @KSCAN Scan keyboard
MOV @>837C,R0 Get status
SRL R0,14 Is EQ bit set?
JNC WAIT No
    
```

\*\*\*\*\*  
 \* Reset tables to default \* \*\*\*\*\*

```

LI R0,>0900 24 lines, interlace off
BLWP @VWTR

*
LI R0,>0D0F Turn off blink routine
BLWP @VWTR This MUST always be done
MOV @SAVRTN,R11
    
```

```

RT Back to mybasic
*****
* VDP Fill subroutine *
*****
VDPFIL MOV *R11+,R0 Get address
MOV *R11+,R1 Get character to write
MOV *R11+,R2 Get count
ORI R0,>4000 Set write flag
SWFB R0
MOVB R0,@VDPWA
SWFB R0
MOVB R0,@VDPWA
SWFB R0 Delay for VDP chip
VDPF1 MOVB R1,@VDPWD
DEC R2
JNE VDPF1
RT Return to caller

*
END
    
```

### Hexadecimal to decimal chart

4th digit		3rd		2nd		1st	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13

(See Page 18)

### Object code

```

00174 A0000B0003B5445B5854B322DB3236B204DB4F44B4520B47527F30DF
0001 A0012B4150B4849B4353B2834B2977B6974B6820B626CB696EB6B69B6E677F28FF
0002 A0028B436FB7079B7269B6768B7420B3139B3934B2062B7920B4444B49207F2D0F
0003 A003EB534FB4654B5741B5245B5072B6573B7320B616EB7920B6B65B79207F2ACF
0004 A0054B746FB2063B6F6EB7469B6E75B6500B0000BC80BC0060B02E0BF0007F293F
0005 A006AB0300B0000B0200B0004B0420B0000B0201B0170B0420C0074B02007F36DF
0006 A0080B0203B0420C007CB0200B0402B0420C0084B0200B032FB0420C008C7F32AF
0007 A0096B0200B0A00B0420C0094B0200B0980B0420C009CB0200B07F5B04207F31CF
0008 A00ACC00A4B0200B0C3DB0420C00ACB0200B0D63B0420C00B4B06A0C01527F2C5F
0009 A00C2B0000B2000B0870B06A0C0152B0A00B0000B010EB0200B015DB02017F326F
0010 A00D8C0002B0202B0019B0420B0000B0200B0203B0201C001BB0202B000D7F349F
0011 A00EEB0420C00E0B0200B0299B0201C0028B0202B001EB0420C00F0B02007F30FF
0012 A0104B07ECB0201C0046B0202B0019B0420C0100B0200B0A40B0201B1E007F32AF
0013 A011AB0420B0000B0200B0A41B0201BFF00B0420C011CB820C0000B83747F302F
0014 A0130B0420B201CBD020B837CB09E0B17FAB0200B0900B0420C00BCB02007F2D1F
0015 A0146B0D0FB0420C0142BC2E0C0060B045BBC03BBC07BBC0BBB0260B40007F28CF
0016 A015CB06C0BD800B0000B06C0BD800C0160B06C0BD801B0000B0602B16FC7F2B9F
0017 A0172B045B7FDA1F
0018 50062START 30110VMBW 30128VSBW 3014AVWTR 40000VMBR 7F326F
0019 30166VDPWA 3016CVPWD 7FA75F
0020 : DDI SOFTWARE 1994 0021
    
```

# Using menus with TI-Base

By RON WARFIELD

The following article appeared in the newsletter of the British Columbia BC 99ers User Group. It is part of an ongoing tutorial.

We will discuss how to make your TI-Base program more useful and a lot easier to use. Everything can be done from menus.

1. Load TI-Base, (XB,EA3,EA5,MINIMEM)
2. When loaded enter date at prompt
3. When the . appears type MODIFY COMMAND DSK1.SETUP

You will see the following screen:

```
*
*****Welcome to TI-BASE
*****QUIT will terminate TI-BASE
*
*PRINTER EPSON
SET CURSOR 2
DISPLAY STATUS
*****FUNCTION (7) for help.
RETURN
```

The \* is for memos only, it does nothing.

The first command that is loaded is SET CURSOR 2. This controls the speed of the cursor (larger number slows down cursor).

DISPLAY STATUS Shows the system stats.

RETURN returns to system operations.

We will modify this command file to a better loader.

Delete the first 4 lines. (function 3-4 times).

Remove the \* from line 5. (function 1).

Delete line 7 and 8.

Press FCTN-8 and the file will be written to the disk. Now when you load TI-Base, the printer will be set up for you, and the screen will be less cluttered. You will notice that when the system loads the commands are printed to the screen as they are executed. If you do not want to see this, add SET TALK=OFF to your setup file. Also if you want to run the system from menus, other commands have to be added to your setup file.

My SETUP file looks like this:

```
SET TALK=OFF ..... stops screen write of commands
COLOR WHITE DARK-BLUE ...sets colors
CLEAR ..... clears screen
CLEAR LOCAL ..... clears all locals
SET RECNUM=OFF ..... turns record numbers off
SET HEADING=OFF ..... turns headings off
SET DATDISK=DSK2.^..... set the data disk to DSK2.
PRINTER EPSON ..... sets up the printer commands
```

```
SET CURSOR 2 ..... sets the cursor speed
DO DSK1.START ..... starts to run the first MENU
```

To make your first MENU file:

1. From the . prompt type MODIFY COMMAND DSK1.START
2. The drive will start and you will see an editing screen 3. Type your commands.
4. Press F8 to write the file.

Below is the structure of my START file. Because of memory limitations, this file cannot be written in TI-BASE. You will have to use TI-Writer and load and save the file START/C and use PF not SF. Don't forget to add the /C. TI-Base does this for you but TI-Writer expects the full file name. Also set the tabs at 39 because TI-Base reads only 40 characters wide in command files.

```
CLEAR
COLOR DARK-BLUE DARK-BLUE ..... color blue on blue (blanks
screen)
LOCAL A C 1 ..... sets local A to 1 character
WHILE A <> *X* ..... while A is not X then return to
here
WRITE 2 9 *^^^^^T.I.^BASE MENU* ..... write menu on screen
WRITE 3 9 *^^^^^=====^* ..... ^
WRITE 4 3 *ENTER:^* ..... ^
WRITE 6 7 *A for^^1st* ..... ^^^This can be longer or shorter
WRITE 7 7 *B for^^2nd* ..... ^^^eg^^A to D or A to W
WRITE 8 7 *C for^^3rd* ..... ^
WRITE 9 7 *D for^^4th* ..... ^
WRITE 10 7 *E for^^5th* ..... ^
WRITE 11 7 *F for^^6th* ..... ^
WRITE 12 7 *G for^^7th* ..... ^
WRITE 13 7 *H for^^8th* ..... ^
WRITE 14 7 *I for^^9th* ..... ^
WRITE 15 7 *J for^^10th* ..... ^
WRITE 16 7 *K for^^11th* ..... ^
WRITE 17 7 *L for^^12th* ..... ^
WRITE 19 7 *X for^^Exit* ..... ^
WRITE 20 30 *Choice?* ..... ^
COLOR WHITE DARK-BLUE ..... ^color white on blue (menu ap-
pears)
READCHAR 20 38 A ..... ^reads a character from screen
into A
^IF A =*X* ..... ^if character = X then
^CLEAR ..... ^clear screen
^CLOSE ALL ..... ^close all databases
^RETURN ..... ^return to command line
^ELSE ..... ^if char = anything else then
^DOCASE ..... ^set up case operations
^^CASE A=*A* ..... ^if A=A then
^^CLEAR ..... ^clear screen
^^WRITE 12 8 *1st* ..... ^write to screen
^^USE 1 ..... ^use database 1
^^DO 1 ..... ^do command file 1
```

(See Page 19)

## MBASIC—

(Continued from Page 17)

E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

### EXAMPLES HEX TO DEC

Hex #	4th	3rd	2nd	1st	=	Dec
>001B	0	0	16	11	=	27
>0100	0	256	0	0	=	256

>0C5A	0	3072	80	10	=	3162
>FFFF	61440	3840	240	15	=	65535

### EXAMPLE DEC TO HEX

Change 2160 to hex.

2160 falls between >8 and >9 third digit.

Use >8 then subtract 2048 from 2160.

Diff=112 which is= to >7 which makes 2160=>0870.

## TI-BASE—

(Continued from Page 18)

```

^^BREAK ..... ^stop
-----
^^CASE A="B" <---*
^^CLEAR..... (repeat for all cases)
^^WRITE 12 8 *2nd* <---*.....use C,D,E,F etc and 3,4,5 etc.
^^USE 2 <---*
^^DO 2 <---*
^^BREAK
-----
ENDIF.....end IF operations
ENDCASE.....end case operations
ENDWHILE.....end while operations
RETURN.....return to command line

```

Above we setup TI-Base and created a START file, which is how we make TI-Base work from menus. Now we will create the database and make some more menus to manipulate the database. To create a database, type at the . command CREATE DSK1.1 which will be our database name. Now just fill in the information you would like. For our tutorial we will have a database as listed below.

```

FIELD^^^DESCRIPTOR^^^TYPE^^^WIDTH^^^DEC
1^^^ONE^^^C^^^10
2^^^TWO^^^N^^^008^^^02
3^^^THREE^^^D^^^008

```

Field 1 contains a "name" of one, will be characters (C) and have 10 spaces to enter data.

Field 2 contains a "name" of two, will be numerical (N) and have 8 spaces and 2 decimal spaces.

Field 3 contains a "name" of three, will be date format (D) and have 8 spaces to enter the date as 01/01/94.

When this has been entered press FCTN-8 to write to the disk.

Now we will make the command file to use this base.

```

CLEAR.....Clears screen
LOCAL B C 1.....Sets a local called B 1 character
WHILE B <> *X*.....while keeps you in menu
WRITE 5 5 *A:^ Change Data*.....writes information to screen
WRITE 6 5 *B:^ Add Data*.....
WRITE 7 5 *C:^ Print*.....
WRITE 8 5 *D:^ Edit Data*.....
WRITE 9 5 *X:^ eKit*.....
WRITE 15 12 *Your Choice*.....
READCHAR 15 24 B.....Waits for input into B
IF B="X".....If input B then
CLEAR.....Clear screen
CLOSE ALL.....Close databases RE-
TURN.....Return to START
ELSE.....If B is not X DO-
CASE.....Start case operation
CASE B="A".....If B = A
DO CHANGE.....Do command file called CHANGE
BREAK.....Stop
CASE B="B".....If B = B
DO APPEND.....Do command file called APPEND
BREAK.....Stop
CASE B="C".....If B = C
DO PRINT.....Do command file called PRINT
BREAK.....Stop
CASE B="D".....If B = D
DO EDIT.....Do command file called EDIT
BREAK.....Stop
ENDCASE.....End CASE operation EN-
DIF.....End IF operation END-
WHILE.....End WHILE operation

```

This is the command file called CHANGE which will ask for information to search for and then change it for you.

```

LOCAL ENTER C 5.....Creates local ENTER, chars, 5

```

spaces

```

LOCAL ONE C 5.....Creates local ONE, chars, 5 spaces
WHILE ONE <> *END*.....Keeps you in the menu
TOP.....Puts database at top or #1 posi-
tion CLEAR.....Clears screen
WRITE 08,02 *Type in the word END to ex;*Write menu to screen, Notice the
it to menu.....ex; Which tells it to go to the
WRITE 12,02 *Enter <INFO WANTED>*.....next line to continue.
WRITE 21,5 *INFO:>.....
READSTRING 21,11 ONE.....Reads your info from screen
IF ONE="END".....If input = END
RETURN.....Return to menu
ELSE.....If not END then Endif and start
ENDIF.....process
FIND ONE.....find information wanted.
IF (EOF).....If end of file then
CLEAR.....Clear screen
WRITE 10 6 *No such info*.....Write to screen
WRITE 12 10 *Press ENTER*.....Prompt to press enter
READSTRING 12 16 ENTER.....Read ENTER
ELSE.....If not end of file
WHILE (.NOT.(EOF));.....and info is not the same as the
^.AND.(INFO <> ONE).....database then
MOVE.....Move one field up (from 1 to 2)
ENDWHILE.....End WHILE process
EDIT.....Edit the database
RETURN.....Return to menu

```

You will notice when you run this file that you will get an error message "Database Not Sorted." For FIND to work, the database must be sorted and FIND only works on the top sorted field. To sort the database, have the database in memory and type at the dot prompt. .SORT ON ONE.

The next command file will be APPEND and is to add data to base.

```

APPEND.....Append adds to the next empty slot
RETURN.....Returns to menu

```

The next command file is PRINT and will print the data.

```

SET PAGE=50.....Sets page length to 50 lines
CLEAR.....Clears screen
LOCAL TEMP C 50.....Creates local TEMP, chars, 50
spaces
TOP.....Go to top of database
WHILE (.NOT.(EOF)).....While not the end of the file
PRINT ONE.....Print the first field (ONE)
REPLACE TEMP WITH TRIM(TWO);.....Put TWO into local TEMP and TRIM
off
^| * | DATE.....blank spaces, add one space and
then
PRINT TEMP.....add DATE, then print local TEMP
MOVE.....Move database up one
ENDWHILE.....End WHILE process
RETURN.....Return to menu

```

Printer codes can be added to make your printout look better. The codes are entered as such: PRINT (DS) (UL) (EX) (CM) (IT) (B) (SPS) (SBS) (NM) then field or text.

This means, in order, double-strike, underline, expanded, compressed, italics, bold, superscript, subscript and return to normal.

An example would be: PRINT (DS)(UL) ONE (NM) would print field one double-strike underlined and then return to normal print.

The last command file is EDIT and will allow you to edit the database.

```

EDIT.....Allows you to EDIT the fields and
RETURN.....page through them with F5 and F6
.....Return to menu

```

When you edit or change data always press Function 8 or press enter to run off the bottom of the screen to make sure you write the changed data.

## A program that writes a program

# Automatic disk menu of BASIC, XBASIC programs

The following program, LOADER100, was written by Jerry Keisler of the Paris (Texas) TI User Group. It has appeared in user group newsletters.

The program writes a load program for Extended BASIC and BASIC programs on a disk. It does this by creating a MERGE file called "CAT" and writing it to a disk that it has cataloged. After quitting the program, enter NEW, MERGE the CAT file into memory and then save it to the disk as LOAD. When you boot up the disk the LOAD program will be loaded into memory with a menu of BASIC and Extended BASIC programs that you can select with one key.

LOADER100 handles up to 100 programs per disk, displaying 20 at a time to the screen. It requires Extended BASIC and a disk system. Complete instructions are included in the program.

### LOADER100

```

3 ! contains fix so LOADING
will appear by program that
is loading. !006
10 !Ever wanted a fast start
ing LOAD program for a disk
that was full and you have n
o intention of changing the
present !012
12 !programs. This program
will build a LOAD program
with a selection list of up
to 100 programs that can be
loaded at the touch of one !
008
14 ! key. !026
16 !THIS PROGRAM WRITES
PROGRAMS!! Due to the nature
of this program, I am
including the explanation in
the program. The new program
!217
18 !is written by sending
the proper ASCII token value
s to a D/V163 file that can
be merged into memory, thus
creating a program. !158
20 !The token values are
sent to a D/V163 file using
CHR$( ). !212
22 !All line numbers use two
CHR$( )'s. I use 0 in the
first one as my maximum line
number is 250. 255 is the
largest value you can put in
!058
24 !one CHR$( ). You must end
each program line with
CHR$(0). The end of program
is CHR$(255)&CHR$(255).
There is no line number or !
179
26 !end of program line in
this line. Everything in a
program line is joined with
the & symbol. !045
28 !You will note symbols
such as , # ( ) :: etc all
have their own CHR$( ). There
are 3 special CHR$( )'s.
CHR$(199) indicates a Quoted
!149
30 !string follows.
CHR$(200) indicates an
unquoted string or number
follows. 199 and 200 must be
followed by the number of
characters in the !037
32 !string. Example CHR$(199
)&CHR$(10)&"hi you all".
CHR$(201) indicates a line
number follows. It is known
two CHR$( )s will follow
this. !150
34 !Example CHR$(134)&CHR$(2
01)&CHR$(1)&CHR$(3) says
GOTO 258. The CHR$(1)=255
and CHR$(3)=3. You might say
you are dealing with a base
255 number system. !237
35 !All CALL routines are
handled as strings. See line
730. !018
36 !Everything else is
documented in the program.
You can delete all remarks
to speed up the !125
38 !initialization of the
program. If the program will
not run there is CAT file on
the disk that is not D/V163.
Change CAT in line 580 to !2
06
40 !another name and
remember to use it in place
of CAT when the program
ends. !016
110 ! A$(0)=DISK NAME !050
120 ! A$( ) =PROGRAM NAMES !0
98
130 ! B =FILE TYPE !036
140 ! C =FILE SECTORS USE
D !087
150 ! CL$ ="CLEAR" !126
160 ! COL =COLUMN # !015
170 ! D =FILE RECORD SIZE
!254
180 ! EN$(0)="X-EX BASIC" !
058
190 ! EN$(1)="X-EX BASIC "
"-MORE PROGRAMS" !186
200 ! I =LOOP !227
210 ! J =FILE COUNTER 127
MAX !202
220 ! K&S =CALL KEY VARIABL
ES !194
230 ! L =# OF PROGRAMS !0
15
240 ! PC =PROGRAM BLOCK !1
18
250 ! PL =LINE # BLOCK !21
0
260 ! LN =PAGE # !058
270 ! Q$ =ACCEPT ALL PROGR
AMS FLAG !035
280 ! QQ$ =ACCEPT THIS PROG
RAM !038
290 ! DEF DI$(R)=DISPLAY AT(
R,COL): !213
300 ! DEF IF$(N)=IF K$="N" T
HEN "RUN DSK1."&A$(I-64+PC)
!173
310 ! DEF LN$(N)=LINE NUMBER
!061
320 ! NOTE LINE NUMBERS WILL
CHANGE DEPENDING ON PAGE OF
(See Page 21)

```

# LOADER100—

(Continued from Page 20)

```

PROGRAM SCREEN. !030
330 CALL CLEAR :: CALL SCREE
N(13):: DISPLAY AT(4,6):"L O
A D E R 1 0 0" !063
340 DIM A$(100)!184
350 DISPLAY AT(6,1):"ACCEPT
ALL PROGRAMS? Y": : "LOAD PRO
GRAMS FROM DISK 1" :: ACCEPT
AT(6,22)SIZE(-1)VALIDATE("Y
N")BEEP:Q$ !120
352 ACCEPT AT(8,25)SIZE(-1)V
ALIDATE("123")BEEP:DISK$ !18
3
353 DISPLAY AT(12,1)BEEP:"PR
OGRAM STATUS.....WORKING"
:: CL$="CLEAR" !004
370 DISPLAY AT(16,5):"DISK N
AME--" :: DISPLAY AT(18,11):
"READING" !062
380 OPEN #1:"DSK"&DISK$&".",
INPUT ,RELATIVE,INTERNAL !06
4
390 ! LINE NUMBER !180
400 DEF LN$(N)=CHR$(0)&CHR$(
N+PL)!162
410 ! DISPLAY AT(R,COL): !05
4
420 DEF DI$(R)=CHR$(162)&CHR
$(240)&CHR$(183)&CHR$(200)&C
HR$(LEN(STR$(R)))&STR$(R)&CH
R$(179)&CHR$(200)&CHR$(1)&ST
R$(COL)&CHR$(182)&CHR$(181)!
240
422 ! IF (K@>64)AND(K@<85)TH
EN DISPLAY AT(K@-62,1)SIZE(-
7):"LOADING" !130
423 DEF LO1$(N)=CHR$(0)&CHR$(
N+PL)&CHR$(132)&CHR$(183)&"
K@"&CHR$(192)&CHR$(200)&CHR$(
2)&"64"&CHR$(182)&CHR$(187)
&CHR$(183)&"K@"&CHR$(191)&CH
R$(200)&CHR$(2)&STR$(I+64)&C
HR$(182)!188
424 DEF LO2$=CHR$(176)&CHR$(
162)&CHR$(240)&CHR$(183)&"K@"
&CHR$(194)&CHR$(200)&CHR$(2
)&"62"&CHR$(179)&CHR$(200)&C
HR$(1)&"1"&CHR$(182)!110
425 DEF LO3$=CHR$(235)&CHR$(
183)&CHR$(194)&CHR$(200)&CHR
$(1)&"7"&CHR$(182)&CHR$(181)
&CHR$(199)&CHR$(7)&"LOADING"
&CHR$(0)!072
430 ! IF K@="N" THEN "RUN DS
K1."&A$(XXX) !153
440 DEF IF$(N)=CHR$(132)&"K@"
&CHR$(190)&CHR$(200)&CHR$(2
)&STR$(N)&CHR$(176)&CHR$(169
)&CHR$(199)&CHR$(LEN(A$(I-64
+PC))+5)&"DSK1."&A$(I-64+PC)
!215
450 ! LOAD PROGRAMS !078
460 FOR I=0 TO 100 !153
470 J=J+1 :: INPUT #1:A$(I),
B,C,D :: IF I=0 THEN 500 ELS
E IF J>=127 OR LEN(A$(I))=0
THEN 550 !223
480 IF A$(I)="LOAD" THEN 470
!152
490 IF A$(I)="LOADER100" THE
N 470 !197
500 IF I=0 THEN DISPLAY AT(1
6,16):A$(0):: GOTO 530 !173
510 IF (ABS(B)=5)OR(ABS(B)=4
AND D=254)THEN DISPLAY AT(2
0,10):USING "### #####
Y":I,A$(I)ELSE 470 !037
520 IF Q$="N" THEN ACCEPT AT
(20,25)BEEP.SIZE(-1)VALIDATE
("YN"):QQ$ :: IF QQ$="N" THE
N 470 !214
530 NEXT I !223
540 ! : "X-EX BASIC" !197
550 EN$(0)=CHR$(181)&CHR$(19
9)&CHR$(10)&"X-EX BASIC"&CHR
$(0)!051
560 ! : "X-EX BASIC " "-MORE
PROGRAMS" !052
570 EN$(1)=CHR$(181)&CHR$(19
9)&CHR$(28)&"X-EX BASIC " "
"-MORE PROGRAMS"&CHR$(0)!190
580 CLOSE #1 :: L=I-1 :: OPE
N #2:"DSK"&DISK$&".CAT",VARI
ABLE 163 !061
590 DISPLAY AT(18,1)BEEP:"RU
N DISPLAY |WRITING" !072
600 FOR PN=0 TO 4 :: PC=20*P
N :: PL=50*PN :: COL=1 !115
610 ! 1 CALL CLEAR !151
620 PRINT #2:LN$(1)&CHR$(157
)&CHR$(200)&CHR$(5)&CL$&CHR$(
0)!129
630 ! 3 DISPLAY AT(1,1):"DIS
K NAME L PROGRAMS" !
129
640 PRINT #2:LN$(3)&DI$(1)&C
HR$(199)&CHR$(28)&"DISK "&A$(
0)&RPT$(" ",14-LEN(A$(0)))-L
EN(STR$(L)))&STR$(L)&" PROGR
AMS"&CHR$(0)!180
650 ! LOAD 20 PROGRAMS AT A
TIME !053
660 COL=8 :: FOR I=1 TO 20 :
: IF (I+PC)>L THEN 710 !230
670 ! LN DISPLAY AT(I+2,8):"
A--PROGRAM" !180
680 DISPLAY AT(20,6)SIZE(-4)
:I+PC :: PRINT #2:LN$(I+3)&D
I$(I+2)&CHR$(199)&CHR$(3+LEN
(A$(I+PC)))&CHR$(I+64)&"--"&
A$(I+PC)&CHR$(0):: NEXT I !1
27
690 ! PRESS FOR MORE !138
700 ! 24 DISPLAY AT(24,1)BEE
P EN$(0) OR EN$(1) !150
710 PRINT #2:LN$(24)&CHR$(16
2)&CHR$(240)&CHR$(183)&CHR$(
200)&CHR$(2)&"24"&CHR$(179)&
CHR$(200)&CHR$(1)&"1"&CHR$(1
82)&CHR$(238)&EN$( (L<21)+1)!
091
720 ! 25 CALL KEY(3,K@,S@) !
073
730 PRINT #2:LN$(25)&CHR$(15
7)&CHR$(200)&CHR$(3)&"KEY"&C
HR$(183)&CHR$(200)&CHR$(1)&"
3"&CHR$(179)&"K@"&CHR$(179)&
"S@"&CHR$(182)&CHR$(0)!103
740 ! 26 IF S@=0 THEN 25 !20
7
750 PRINT #2:LN$(26)&CHR$(13
2)&"S@"&CHR$(190)&CHR$(200)&
CHR$(1)&"0"&CHR$(176)&CHR$(2
01)&LN$(25)&CHR$(0)!056
752 !IF (K@>64)AND(K@<(I+64)
)THEN DISPLAY AT(K@-62,1)SIZ
E(-7):"LOADING" !036
754 PRINT #2:LO1$(27)&LO2$&L
O3$ !044
760 FOR I=65 TO 20+64 :: IF
(I+PC)>(L+64)THEN 810 !190
770 ! 20 EA IF THEN RUN DSK
PROGRAM !248
780 ! LN IF K@=65 THEN RUN "
DSK1.PROGRAM" !096
790 DISPLAY AT(20,1)SIZE(-4)
:I+PC-64 :: PRINT #2:LN$(I-3
7)&IF$(I)&CHR$(0):: NEXT I !
226
800 ! 48 IF K@=88 THEN CALL
CLEAR :: END !179
810 PRINT #2:LN$(48)&CHR$(13
2)&"K@"&CHR$(190)&CHR$(200)&

```

(See Page 22)

## LOADER100—

(Continued from Page 21)

```

CHR$(2)&"88"&CHR$(176)&CHR$(
157)&CHR$(200)&CHR$(5)&CL$&C
HR$(130)&CHR$(139)&CHR$(0)!2
29
820 ! 49 IF K@<>32.THEN 25 !
031
830 PRINT #2:LN$(49)&CHR$(13
2)&"K@"&CHR$(191)&CHR$(192)&
CHR$(200)&CHR$(2)&"32"&CHR$(
176)&CHR$(201)&LN$(25)&CHR$(
0)!207
840 IF PN=0 AND L>20 THEN 90
0!204
850 IF PN=1 AND L>40 THEN 90
0!207
860 IF PN=2 AND L>60 THEN 90
0!210
870 IF PN=3 AND L>80 THEN 90
0!213
880 ! 250 GOTO 1!228
890 PRINT #2:CHR$(0)&CHR$(25
0)&CHR$(134)&CHR$(201)&CHR$(
0)&CHR$(1)&CHR$(0)!118
900 IF (I+PC)>(L+64) THEN 930
!026
910 NEXT PN!052
920 ! END OF PROGRAM LIST !1
95
930 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #2 :: DISPLAY A
T(12,21)BEEP:"COMPLETE" !200
940 DISPLAY AT(24,7):"(Q)UIT
(R)EDO" !187
950 CALL KEY(0,K,S):: IF K<8
1 OR K>82 THEN 950!041
960 IF K=82 THEN 350!153
970 DISPLAY AT(15,1)ERASE AL
L:"ENTER THE FOLLOWING"::"N
EW"::"MERGE DSK"&DISK$&".CA
T"::"SAVE DSK"&DISK$&".LOAD
"!051

```

# Using multiplan as a text processor

## Works well for multiple column output

By PETE PHILLIPS

The following article has appeared in user group newsletters.—Ed.

Word processing with Multiplan?

Why not?

Multiplan has many advantages over TI-Writer and the Editor/Assembler editor. For instance, Multiplan allows you to format a document in a columnar layout and print it in condensed text, providing a larger amount of text on a given page. In addition, Multiplan will center text where desired and allow for the movement of blocks of text in a more flexible manner.

Using Multiplan as a word processor has its drawbacks. Among these are the absence of a global editor. The editing of text is a bit more difficult, as you cannot simply type over your text. Additionally, the fast typist will have to learn to slow down due to the program's relatively slow processing speed. However, despite these drawbacks, for many uses Multiplan may be the easiest way to solve the word processing problem at hand.

I do not propose to go into a full tutorial on the use of Multiplan. For that I refer you to the Multiplan manual. I realize that many users find that to be a formidable document. But only a general knowledge of Multiplan is needed to use it as a general text processor. Therefore, for this discussion, I will cover what I have found to be the easiest steps to follow in setting up

and using the Multiplan worksheet.

Starting with an empty worksheet, your first step should be to select the OPTions command and turn off the "recalc" feature. Since you will be doing no mathematical calculations, this will eliminate the considerable delay incurred as the program searches for numeric cells.

Next, select the Format option, then select Default on the submenu. Finally, select Width on the next menu and set the default width to 30 columns. I know that it is possible to set it at 32 columns, but by setting it at 30 we will later be able to widen for a buffer between columns of text.

The next setup step that is advisable is to again select the Format, Default option. But this time select the Cells feature on the third menu. In the alignment column select "L" for left. Remember, when Multiplan is displaying the alpha/value prompt, hitting a number as the first character in a line will select the value option, rather than alpha. Therefore, if the first character in a line is numeric, you must first hit Enter twice to specifically select the alpha command. However, in case you forget, and the only characters entered on that line are numeric, this will prevent them from being right-justified or otherwise skewed.

The final setup step I use is to select the Window option and place a border around the one open window. You may then use

this border as a line length guide when typing. You may type up to but not including the column containing the right border without having the end of your text cut off.

You are now ready to begin entering your text. Start at row one, column one and enter one line after another in the first column. Don't worry about formatting at this point since this makes it somewhat easier to move data about. Another advantage is that you don't have to worry about keeping track of where you are located on the page.

After entering your text, you're ready to format it into columns. Since the maximum column width on the TI printer is 132, we will divide the text into four equal columns of 32 characters each and have a two-column border on the left and right margins.

Assuming we are working with one page as an example, there are two ways you could format the text. One is to simply divide it into 54 rows per column, assuming the page length is 66 lines, and leave whatever may be left over in the fourth column. You may also make the columns of equal length, in which case you would simply divide the total number of rows by four and make each column that length.

For example, let's assume that the total number of rows, when the document is formatted in one column, is 200. Now, 200

(See Page 23)

## MULTIPLAN—

(Continued from Page 22)

divided by four equals 50. We would then make each column 50 lines long. To do this, copy from rows 51 to 100 and place the copied text into row one, column two. Next, copy the text from rows 101 to 150 and place it in row one, column three. Finally, copy from rows 151 to 200 and place the text in row one, column four.

You now have the entire document in rows one through 50 and columns one through four, but you still have the text use in columns two through four appearing after row 50 in column one. Use the delete command to get rid of these. Now change the default width to 32 to provide spaces between columns.

You are ready to print the file. To do this, first save the file to disk. Next, exit Multiplan, select TI BA-

SIC and enter the following commands (make sure your printer is connected before doing this):

```
OPEN #1:"PIO.CR"
PRINT #1:CHR$(27);CHR$(15);
```

CLOSE #1

Exit TI BASIC. These commands set up the TI dot matrix printer to print in condensed text. If you have a different printer brand, you may need to substitute different commands.

Reload Multiplan and select Print and Options. Enter your printer name in the setup field and return to the Print menu. Now, select margins and set the left margin to two and change the print width to 132. All that needs to be done now is to select the printer command and your document should come out in four even columns on one sheet.

I admit that this procedure sounds tedious, but it is the most flexible means I know of to format text into columns. I have made several attempts to devise a program to translate a D/V80 file into a Multiplan file using the Symbolic Link file format. Unfortunately, my attempts have been fruitless. But I'm still working on it.

### Other multi-column ideas

The procedure outlined here can be simplified by setting your printer to condensed type before loading Multiplan. Also, the same 132-column condensed print setup procedure can be done with Funnelweb. This can also be done through TI-Writer using Transliteration commands.

There are several programs that will format a D/V80 document into three or four columns, although there are fewer options at manipulating the text. Also, the Multiplan file can be output as a D/V80 file and imported into TI-Writer or other word processor. But no program lets you format multi-column text as quickly with as much control over line endings as Multiplan.

## 1994 TI FAIRS

### MAY

**Lima Multi User Group Conference**, May 13-14, Ohio State University Lima Campus, Lima, Ohio. Contact Lima Ohio Users Group, P.O. Box 647, Venedocia, OH 45894.

### NOVEMBER

**The TI International World's Faire**, Nov. 12, Holiday Inn, Gurnee, Illinois. Sponsored by Chicago and Milwaukee users groups. For information, contact Don Walden (414) 679-2336.

## 1995 TI FAIRS

### FEBRUARY

**Fest West '95**, Feb. 18, Fabulous Inn, San Diego, California. Contact Southern California Computer Group, P.O. Box 152535, San Diego, CA 92108, or call the SCCG BBS, (619)263-9125, User No. 25, password FEST.

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.

## MS Express dissolves after Lima fair

MS Express Software has been dissolved as of May 31, according to Mickey Schmitt Cendrowski.

She says this occurred because of "irreconcilable differences" between her and Mike Sealy, her partner in the company, following the Lima Multi User Group Conference May 14.

She notes that all the company's software titles and copyrights remain the property of their respective authors. She says she wants to thank all persons who have supported MS Express Software over the years.

Cendrowski says that Notung Software has exclusive permission to market and distribute the following products previously released through MS Express Software:

- Adventure Hints, Series I and II, by Lynn Gardner.
- Page Pro Cataloger; Sliding Block Puzzles, Series I, II and III; and Sliding Block Solutions, Series, I, II and III; by Norman Rokke.
- Oliver's Twist by Lynn Gardner and Mickey Schmitt Cendrowski.
- Adventure Data Base, Genealogy Plus!, Rattlesnake Bend and TI99/4A Software Data Base by Mickey Schmitt Cendrowski.

For information on purchasing the above titles, write Notung Software, 7647 McGroarty St., Tujunga, CA 91042.

## MICRO-REVIEWS

# Double Entry Bookkeeping, Funnelweb V5.01 Editor, Funnelweb Text Sort, Reformat

By CHARLES GOOD

---

### DOUBLE ENTRY BOOKKEEPING by Harold W. Evans

---

I am not ideally qualified to review this software because I have never had the mundane job of keeping the detailed financial records that are required to run a business. I did show this software to a factory manager who is a member of my local user group and he agreed with me that Double Entry seems quite comprehensive. You can, as the author does, keep financial records for an entire business using this software and your 99/4A computer. Name and address lists, ledgers, payroll records, accounts receivable and accounts payable are all covered by this one integrated package. A peek at the main menu will give you an idea of the broad scope of Double Entry. Here are your choices:

#### Journal Entries

- 1 — Daily or end of month
- 2 — Post to ledger accounts.

#### Ledger Accounts

- 3 — Find, add, edit, delete
- 4 — Sort alphabetically
- 5 — Print all or balances
- 6 — Transfer balances to new disks.

#### Open All Files

- 7 — Name/address & Ledgers.

#### Other

- 8 — Payroll records
- 9 — Financial reports and Accts/Rec statements."

You can print to the screen or to a printer lists of accounts, the contents of an entire account, balances of accounts and financial reports in a variety of formats. Up to 79 accounts receivable and payable can be tracked. A report from the accounts receivable files will age the accounts for 30, 60 and 90 days, which I believe is something no other 99/4A accounting software will do. About the only thing you can't print are checks (for payroll and accounts payable) and invoices.

All the Double Entry software comes on

a DSDD disk, and that is its main limitation. All its parts are written in Extended BASIC, and you can't break it down so that it works effectively in single-density format. You need a double-density floppy controller to use Double Entry and you have to run the software entirely from floppy drives. All records, ledgers, etc are stored on other DSDD disks. The whole system uses six DSDD disks, the program disk and five record disks. You don't have to have all the disks online simultaneously, which is good because TI systems support only four floppy drives. Disks are requested by diskname. For example, the screen may say "Requires diskette PR/ACCTS." You then put the disk named PR/ACCTS into any drive and press Enter. The computer searches all drives until it finds the disk with the proper name and then reads needed data off of that disk. Because of searching for disks by diskname and because of the vast amount of data on all these disks, it would be difficult to run Double Entry from a Horizon RAMdisk.

Double Entry is completely menu driven. I have not managed to crash the program. If it asks for a disk and I put in the wrong disk it will ask again. Ultimately you always get back to the main menu. Execution is slow. This is the fault of Extended BASIC and all the disk switching.

If you have a double-density floppy controller and do business bookkeeping then you should give Double Entry a try. It costs only \$10, which includes postage. There is no other financial package for our computer that is as inexpensive and as comprehensive. Send your \$10 to Dr. Harold Evans, 293 Circle Hills Dr., Grand Forks ND 58201.

And now, three new software packages for manipulating text files. First and most important:

---

### FUNNELWEB v5.01 EDITOR by Tony McGovern

---

The best keeps getting better. This latest

enhanced editor runs out of the Funnelweb V4.4 system. Substitute the new editor files for your existing ED/EE files on your Funnelweb disk. For 40-column users we now have optional column by column left/right scrolling. Just press CTRL= to toggle back and forth between 20-column windowing (the way 40-column TI-Writer has always been) and scrolling one column at a time. Column scrolling is active when you use the arrow keys and when you are typing text. When you column-scroll to the right margin the cursor then drops down a line and starts again at the left margin.

For both 40- and 80-column users we have — ta da — right margin justification directly from the editor. You see right justification on screen and print it out justified with PF directly from the editor. You won't have to use the formatter any more just to get right justification. Justification is done with an alternate type of reformat. CTRL-2 does the normal, non justified, reformat as always. CTRL-R reformats from the cursor position with right justification. This has to be done one paragraph at a time. Position the cursor at the beginning of each paragraph. If you want to maintain paragraph indentation press FCTN-2 (insert). Then press CTRL-R and the whole paragraph is reformatted and justified.

There is a big new feature for 80-column users, and I do mean BIG. You get a 64K text buffer capable of holding in active memory a text file well in excess of 300 disk sectors. Even MY-Word on a Geneve doesn't have a text buffer this big! And you can put an equally big text file into a second 64K view only buffer and then you can cut and paste between these two files. You can mark up to a screen full of text (from 1-23 continuous text lines) from either the view only buffer or the edit buffer and place this text in a "clipboard." The contents of this clipboard can then be inserted anywhere into the edit buffer. I used this editor to create the 200-300+  
(See Page 25)



## MICRO-REVIEWS—

(Continued from Page 24)

sector Sherlock Holmes story text files I described in last month's MICROreviews.

The V5.01 80-column editor works only with systems that have the full 192K video RAM in their 80-column device. All Mechatronics and TIM 80-column peripherals have this. Some AVPC cards and Geneve computers may not.

If you have "only" 128K video RAM in your 80-column system you need to use the previous V5.0 80-column Funnelweb editor. If you are in the unfortunate situation of having a less than fully upgraded 80-column device, take heart in the fact that there is something 80-column V5.0 can do that V5.01 can't — V5.01 has lost the capability of switching between 80- and 40-column displays.

These new editors are shareware. If you like the new features, send the Funnelweb authors some additional money. You can get from me the new V5.01 editors to try in any of three formats:

1 — The complete V5.01 80-column upgrade on a DSSD partially archived disk that you add to your Funnelweb V4.4 system disk.

2 — The complete V5.01 40-column upgrade on an unarchived DSSD disk that you add to your V4.4 system disk.

3 — An unarchived floppy disk that contains on one side a standalone SSSD version of Funnelweb that requires no configuration, has the basic V4.4 system files, disk review, and the V5.01 40-column editor with all chars graphics and three languages. This standalone mini-version of Funnelweb will run directly as

DSK1.LOAD and has the new editor's documentation on the flip side of the disk. I'll send you the new V5.01 editor in any or all of these three formats for \$1 each.

### FUNNELWEB TEXT SORT by Bob Carmany

This is based on code originally published by Bruce Harrison. The code has been modified by Bob to run out of the Funnelweb environment, hence the name. It runs only from a Funnelweb loader or menu and preserves the Funnelweb kernel, including the "Work File" name used with LF/SF in Funnelweb's editor.

Text Sort sorts a D/V80 file alphabetically by treating each text line as a separate record for sorting. Groups of text lines are sorted based on the letter(s) at the start of each text line. You are asked for the input filename and the name of the sorted file.

Text Sort is very fast. Bob uses it to sort his "user" SPELL-IT dictionary. I use it to sort text lines containing student names followed by their grades in the biology courses I teach.

Text Sort is public domain. I will send it to you with fully commented source code on a SSSD disk for \$1, or you can contact the author directly. Bob Carmany, 1504 Larson St., Greensboro NC 27407.

### REFORMAT by Bruce Harrison

Several years ago Jim Peterson wrote an Extended BASIC text reformatter that would format any D/V80 text file to the

user's choice of line width with correct word wrap (no splitting words between lines) and with or without right justification. It works nicely, but it is slow. Bruce Harrison has created a similar D/V80 REFORMATter that runs at assembly speed.

Reformat loads from Extended BASIC or from EA5. Simply enter the input filename, output filename, line width, left margin, and whether or not you want right justification. Press Enter and away you go! Reformat is fast. Text is displayed on screen as it is being reformatted, and it really zips by quickly, particularly if the input and output files are on a RAMdisk. Normally paragraphs are right justified except for the last line, which is recognized by a carriage return control character at the end of the paragraph. Reformat also right justifies reasonably well with files that have no control characters (no CRs) because you can specify that lines ending in a period should not be right justified.

Text files of any length can be reformatted with Reformat because the entire file doesn't have to be in memory all at once. If you take me up on my offer made last month for all the Sherlock Holmes stories in D/V80 format, you can reformat these large Holmes files to 40-column width. The resulting reformatted text is very easy to view on a 40-column display without extra blank lines and without words being split between lines in odd places. Reformat is public domain. Mail me \$1 and I'll send it to you on a SSSD disk complete with source code and Bruce's very user friendly instructions.

## Gordon's Budget Planner

# Designed to meet the budgeting needs of small groups and families

By JOE N. SIMMONS

Jimie Gordon purchased a TI99/4A back in the early '80s, read the manual and began to learn the art of programming. Being an elder of a small congregation of the Church of Christ and needing to create a budget and keep track of expenses was the seed for the following program.

Soon after Jimmie began to understand the computer, he purchased a cassette player and began to use it to store the various portions (sub-programs) which made up the Budget Planner. All of the portions, if put together, far exceeded the available console memory.

The "nitty-gritty" of the program is the

planner which allows you to display by category *whatif's*. The screen display is split in three parts with the upper portion showing the category results by the week, month and year, the middle section showing the cumulative totals of all categories by weekly, monthly and yearly amounts,

(See Page 26)

## BUDGET PLANNER—

(Continued from Page 25)

and the lower portion allowing you to enter whatif's by either weekly, monthly or yearly amounts. Once entered, the results for the category and the cumulative totals are immediately displayed. As you near the end of the budget categories you find that the cumulative total either does or does not exceed income (contributions with a church budget). The planner is helpful in balancing wants with reality.

Keeping track of expenses is the other part of the program. The program allows the printing of a budget sheet with up to 13 columns of category cumulative totals by month, year, budget, average and percent. Actually, if you like to dabble at programming, you can change this budget sheet to suit your needs.

As with most programs, Budget Planner allows you to start from scratch. The welcome screen menu lists the following options:

- 1 — Start a New Year/file
- 2 — Load Existing File
- 3 — Purge Data
- 4 — Load C.B. Mgr. File
- 5 — Catalog
- 6 — Back (quit).

After pressing "1," you are asked to enter the year (last two digits) and the beginning balance. Next you are given the opportunity to load an existing file to edit or create a new one with up to 42 cate-

gories. Subsequently, you are given the opportunity to create your budget (budget planner). After you have your categories and budget entered, you are given the opportunity to save these to either cassette or disk.

After returning to the main menu,

**The program allows the printing of a budget sheet with up to 13 columns of category cumulative totals by month, year, budget, average and percent.**

which is accomplished by entering the menu number representing "Back" and choosing "Load Existing File," you have the opportunity to enter new data, see existing data, delete month or go back to the previous menu. When entering data the syntax is "expense, category." However, you are first prompted to enter the month, year. After you are finished entering data, you will be prompted for the weeks. The syntax is "# of weeks, 38" (38 is the category for weeks). After pressing Enter, you

will be prompted to either "exit" or "back."

Being at that particular time the church treasurer I made several modifications, such as entering the month, year and weeks within the program so that it will not be prompting one to do so. At the beginning of each year, I modified a fix file which I merged with this main program to effect these annual changes.

The reason for entering the weeks is to compute the averages on a monthly basis.

Back in 1988, Jimmie modified his program to read Irving Crowley's "System III — Checkbook Manager" data files. It incorporates Crowley's collations as categories, thus giving one the ability to enter the individual checks and deposits with Crowley's program and to print out the budget sheet using Gordon's program. This enhanced both programs to become more useful.

Having used and modified Jimmie's program, I feel it, with access to Crowley's program which can handle well over 1,000 records, is just what a treasurer for either a small congregation or a family household needs to keep the finances balanced.

To obtain a copy, contact Jimmie Gordon at Route 1, 1221 Kimbrough Avenue, Etowah, TN 37331. Send him at least \$5 to cover the cost of the diskette and mailing expenses.

## Programmable Flash Memory

# Geneve EPROM replacement lets you boot up MDOS in seconds

By CAL ZANELLA

In the July 1993 issue of MICROpendium I submitted a review of the Boot ROM BIOS for the Geneve 9640. This software and hardware modification of the Geneve was offered through Cecure Electronics located in "cheddar country" Wisconsin. Jim Schroeder, along with Don Walden and his crew, did a fine job of bringing this product to market.

The one shortcoming of the Boot ROM BIOS was that once installed on the Geneve, the operating system that was burned into the EPROM was somewhat permanent. In order to change the operating system, one had to return the EPROM to Cecure to have it reprogrammed.

Enter PFM, or Programmable Flash Memory. PFM is a programmable flash memory module which replaces the Geneve EPROM. It is a 128K capacity chip which allows you to upload SYSTEM/SYS to a predefined memory area on the module. Also located in a predefined memory segment on the chip is Jim Schroeder's PFM boot manager and menu system. The boot manager is similar to that of the Boot ROM BIOS, although other features have been added to give the PFM modification more flexibility.

Don Walden shipped my Geneve with MDOS V2.0 preinstalled in the PFM. After inserting the Geneve into the PEB, I booted the

(See Page 27)

## PFM—

(Continued from Page 26)

system and the usual boot screen acknowledging the BIOS version number, copyright notice and serial number appeared. A 15-second delay allowed my hard drive to reach operating speed and SYSTEM/SYS promptly loaded into memory from the PFM device, searched DSK1 for an autoexec file and, finding no floppy in DSK1, returned to the A> prompt. I normally boot from a Horizon Phoenix card, so I grabbed my backup floppy that has all my Phoenix boot files — SYSTEM/SYS sector edited to load Autoexec from DSK6, Autoexec batch file and John Johnson's Setcolor utility file. I copied these files to the RAMdisk for a much speedier load of SYSTEM/SYS to the PFM. I then performed a warm boot (Control/Shift/Shift) and depressed the keyboard space bar until the PFM boot manager menu appeared. The menu options follow:

Select Direct Boot Device  
 B Boot Bios  
 R Ramdisk  
 H HFDC Card  
 F FDC Card

I then selected "R" for RAMdisk and was presented with another screen with the option "Update SYSTEM/SYS program area of BOOTROM (y/n)?" At this point a keyed response of "y" (yes) will load the SYSTEM/SYS file located on the Phoenix boot drive (DSK6) directly into the PFM. After the file is loaded into the PFM the operating system drops back to the MDOS prompt. From that point on the computer will automatically boot from the PFM when reset. Had I keyed in a response of "n" (no) the computer would have booted directly from the RAMdisk without changing the contents of PFM. At this point I could now delete the copy of SYSTEM/SYS from my RAMdisk to regain another 481 sectors for other use.

The other menu choices operate in the same fashion, with the exception of "H" for HFDC (hard and floppy disk controller) If the Myarc HFDC card is the only controller card in the PEB, then this is the appropriate choice for loading from either a hard drive or floppy drive connected to the HFDC. When "H" is selected, a second screen is presented — "Select H for Hard or F Floppy Load." Selecting "H" will load SYSTEM/SYS from hard drive No. 1. Selecting "F" will load SYSTEM/SYS from the first floppy drive connected to the HFDC. Either choice will also prompt to update the SYSTEM/SYS program area of the BOOTROM.

Menu selection "F" for FDC (floppy disk controller) will perform the same functions as the other choices. This would be the appropriate choice for a floppy only system or a system with both an HFDC and FDC in the PEB. For these circumstances, also having an additional floppy drive connected to the HFDC and configured as DSK5, another option would be to select the "H" menu option and follow up with "F" for floppy load from the second option screen. This action would allow booting from the floppy (DSK5) connected to the HFDC. As you can see, several loading options are available.

As expected, the menu "B" option will boot directly from the PFM device. During a cold boot or a warm boot, the menu can be

accessed either by holding down the keyboard space bar or the shift key during reset.

One minor problem I had with this setup is that I was unable to load SYSTEM/SYS from my 42-MB Mitsubishi MR535 hard drive. This drive is a combination MFM/RLL drive. After discussing this problem with Jim and Don on a conference call, I pulled the drive and connected a standard MFM Seagate drive to the HFDC and had no problem booting from it or several other standard MFM drives that I tested. Be forewarned that you may not be able boot from a combination MFM/RLL drive.

Another significant point that I nearly forgot to mention is that MDOS version 2.0 will boot from a 1.44 megabyte floppy disk with PFM and a HFDC located at 1100 CRU.

**The program allows the printing of a budget sheet with up to 13 columns of category cumulative totals by month, year, budget, average and percent.**

For those unaware of other Geneve modifications offered by Cecure Electronics, Don has developed a 384K Static RAM upgrade for the Geneve. This mod adds 384K of addressable CPU memory directly

to the Geneve circuit board, providing a total onboard memory capacity of 896K. I had Don add the mod to my Geneve and at the same time had him repair my Myarc 512K memory expansion card. The combined total memory of this system now stands at 1,376K or 1.34 megabytes. The system runs flawlessly!

Now for the latest news: The "cheddar country" boys are now offering a PFM+ modification. The + indicates an addition to the standard PFM device in the form of an additional 128K flash memory module that is piggybacked onto the PFM and through special software control acts as a very high speed RAMdisk. The speed comes from memory-to-memory data transfer with no Device Service Routine overhead involved. The extra 128K allows you to load your most frequently used software for instantaneous access. A lengthy discussion with Don revealed that the GPL Interpreter files, along with the combined Extended BASIC/Editor Assembler files could be loaded and ready for use in a speedy 2 seconds. Wow! GENTium takes on Pentium!

Don also mentioned that Cecure is planning on an 8-MB PFM++ device, although the chip they plan on using isn't expected to be released until the first quarter of 1995.

For additional information and pricing call or write: Cecure Electronics Inc., S74 W17000 Janesville Rd., P.O. Box 132, Muskego, WI 53150-0132; 414-679-4343 (voice), 414-679-3736 (fax).

**Attend a TI fair this year!**

# USER NOTES

## Floppy disk facts

Many computer users believe that floppy disks are extremely sensitive to magnetic fields. But, according to 3M, a leader American manufacturer of floppy disks, this is not necessarily true. Here are some facts:

- A few inches of space protect against even strong magnetic fields. A refrigerator magnet will erase data if direct contact is made. A magnet on top of a stack of disks will damage only the closest one or two. Magnets stuck on a metal field cabinet containing disks will not do any harm.
- Heat will not cause data loss unless the disk is melted.
- Static electricity will not harm the disk. A close lightning strike could zap data, but the disk would need to be so close that the disk would likely be destroyed anyway.
- X-rays and airport metal detectors will not erase disks.

• Radar and microwaves only damage media that is in front of the antenna.

## Making a modem cable

This item was written by Frank Frankenburger and has appeared in user group newsletters.

Most of us Tiers who already have a modem don't need this little project. However, anyone who wants to get into modems will find that buying one is easy but connecting it to the TI isn't.

Most external modems today will work on the TI system, with the correct cable. You won't be able to find one at a computer store, however. But making your own is a relatively simple project that can be done in 30 minutes even by novices.

In researching this project I came upon three different cable makeups. The one I will discuss here is based on information from the TI RS232 interface card manual. This also applies to the Myarc RS232

card. I don't have a CorComp book to check, but it is probably the same.

The won't discuss the different configurations except to say that one is on my modem that I purchased from a former user group member and the other is from a BBS program author who says that to use his BBS program you may have to make a new cable per his instructions to use his software.

With that out of the way, let's gather our materials. As usual, I will refer to Radio Shack part numbers. I purchased the same parts from a local electronics store and paid about half of what Radio Shack charges. So shop around.

Note that the use of a low wattage soldering iron and flux are required for this project.

### List of Materials

- 2 each, 25-pin male solder D-Sub connectors — Radio Shack 276-1547
- 2 each, 25-position D-Sub hoods — Radio Shack 278-1549

(See Page 29)



**ELECTRONICS INC.**  
P.O. Box 222, Muskego, WI 53150  
**9640 Upgrades**

**384K UPGRADE** EXTRA 384K CPU MEMORY.....\$100.00

- \* 917K+ TOTAL ON BOARD MEMORY
- \* STATIC RAM — NO Refresh — uses MEMORY PAGES CO-EF
- \* NO VALUABLE CARD SLOTS are used
- \* X-TRA LARGE RAM DISK — UP TO 1500+ SECTORS
- \* X-TRA LARGE PRINT BUFFER — up to 400K+
- \* X-TRA LARGE ARRAYS in MDOS BASIC — up to 458k+ default in 64K
- \* KEEP TI-MODE ON and run MDOS BASIC
- \* WORKS with MYARC 480K card or MEMEX 504K card
- \* WORKS with PFM & PFM+ or without them

**PFM Programmable Flash Memory** MDOS on a FLASH ROM.....\$75.00

- \* REPLACES your G.98 BOOT EPROM
- \* BOOTS YOUR SYSTEM without a FLOPPY, HARD DRIVE or RAMDISK
- \* YOU REPROGRAM IT with your modified or latest MDOS
- \* BOOT an alternative MDOS from up to a 3.2 MEG RAMDISK, 1.44 MEG FLOPPY, HARDDRIVE, CorComp, MYARC or TI FDC
- \* LOAD/SYS IS BUILT IN

NOTE: On normal GENEVE, SYSTEM/SYS must be on the 1st 256K on any RAMDISK and LOAD/SYS works on up to 720K FLOPPYS only.

**PFM+ 128K FLASHDISK**.....\$50.00

- (if purchased with PFM — \$60.00 if installed later)
- \* NO BATTERIES are used for back-up
- \* NO DEVICE CRU addresses are used
- \* NO VALUABLE CARD SLOTS are used
- \* YOU PROGRAM IT with the files you want — over 500+ SECTORS
- \* Easy to use menu for reprogramming

**NOTE: PFM+ 128K FLASHDISK REQUIRES PFM UPGRADE!**

MASTER CARD or VISA ORDERS  
CALL TOLL FREE 1-800-959-9640  
VOICE # 414-679-4343 FAX # 414-679-3736

## Chicago fair sets Nov. 12 date

A definite date of Nov. 12 has been set for the combined fair of the Chicago and Milwaukee TI users groups, according to Don Walden, president of the Chicago Users Group.

Scheduled site is the Holiday Inn in Gurnee, Illinois, Walden says. A definite name has not been selected for the fair, he says. Previously, the Chicago and Milwaukee groups held back-to-back fairs — one Saturday, the other Sunday — on the same weekend at different locations.

For further information, contact Walden at Cecure Electronics, P.O. Box 132, Muskego, WI 53150 or (414) 679-4343.

## CardFile 3.0 released

Bill Gaskill has announced the release of CardFile 3.0.

This version combines CardFile 1.0 and QuickFile 1.5 into a single more powerful program, Gaskill says, for users who want to emulate a 3x5-inch index card file electronically.

New features include sorting of insert file entries, more index cards per insert (120 per insert vs. 52 in previous versions), a manual page eject feature when printing, the ability to load an index card template from within the index card editor and the consolidation of the insert and template editors into one program.

CardFile 3.0 comes on two SS/SD floppy disks and may be ordered for \$15 from Bill Gaskill, 2310 Cypress Court, Grand Junction, CO 81506. Gaskill says registered owners of previous versions of CardFile or QuickFile may upgrade for \$5 to cover the cost of disks, mailer and postage.

# USER NOTES

(Continued from Page 28)

2 feet, 6-strand, 24 awg strand wire.— available at any electronics store.

The modem cable is different at each end and, although the books say it doesn't make any difference which end you attach to the modem, I have never seen one that was not clearly marked MODEM on the plug that goes to the modem. I will not break practice here. You should clearly mark each end of the cable *before* going any further.

Solder the strands of the wire to the 25-pin D-Sub connectors according to the table below. Use caution to make sure that each wire is soldered neatly so as to not short across to another pin.

Modem Connector	RS232 Connector
1.....	1
3.....	2
2.....	3
20.....	6
7.....	7
6.....	20

After making sure that the wires are soldered properly and that none of the wires is shorted to another, we can attached the D-Sub hoods. Do not bypass this step. The hood is needed to be able to attach and remove the plugs from the modem and RS232 card. If you don't use them, it won't be long before you have broken or shorted wires or a variety of other little problems. Besides, it dresses up the project and gives you something to write on to mark modem and RS232 on their respective ends.

## File fixer for Telco

This item was written by Gary D. Bishop of the Cedar Valley 99er User Group. It has appeared in user group newsletters.

While I was downloading from a bulletin board, I needed to see an ASCII text file. The board wouldn't let me just list it out. I had to download it somehow. If it could have been listed, I could have used the log-to-disk feature of Telco to capture

it. No such luck for me that night.

I downloaded the file using Xmodem. Of course, Telco didn't know the file was text, so it used the default file specification for all downloads, which is Display Fixed 128. I needed a way to convert the D/F128 file into a D/V80. So I wrote the following program. I also prints the text to screen as it does the conversion.

```
100 ! THIS PROGRAM TAKES A D
/V128 FILE AND CUTS IT UP FO
R EDITING !207
110 ! USE WITH THE XMODEM FE
ATURE OF TELCO TO RECEIVE AS
CII FILES; THEY END UP AS D/
V128 !218
120 CALL CLEAR :: INPUT "INP
UT FILENAME ":F$: :: OPEN #1:
F$,DISPLAY ,FIXED 128 !138
130 INPUT "OUTPUT FILENAME ":
Z$: :: OPEN #2:Z$ !057
140 LINPUT #1:A$ !187
150 START=1 !070
```

(See Page 30)

## MICROpendium disks, etc.

- Series 1994-1995 mailed monthly (April 1994-March 1995)..... \$40.00
- Series 1993-1994 mailed monthly (April 1993-March 1994)..... \$25.00
- Series 1992-1993 (Apr 1992-Mar 1993, 6 disks) .. \$25.00
- Series 1991-1992 (Apr 1991-Mar 1992, 6 disks) .. \$25.00
- Series 1990-1991 (Apr 1990-Mar 1991, 6 disks) ..\$25.00
- Series 1989-1990 (Apr 1989-Mar 1991, 6 disks) ..\$25.00
- Series 1988-1989 (Apr 1988-Mar 1989, 6 disks) ...\$25.00
- 110 Subprograms (Jerry Stern's collection of 110 XB subprograms, 1 disk) .....\$6.00
- TI-Forth (2 disks, req. 32K, E/A, no docs).....\$6.00
- TI-Forth Docs (2 disks, D/V80 files) .....\$6.00
- 1988 updates of TI-Writer, Multiplan & SBUG (2 disks) .....\$6.00
- Disk of programs from any one issue of MICROpendium between April 1988 and present .....\$4.00
- CHECKSUM and CHECK programs from October 1987 issue (includes docs as D/V 80 file) .....\$4.00

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Texas residents add 7.75% sales tax.

Check box for each item ordered and enter total amount here:

Check/MO                      Visa                      M/C  
(Circle method of payment)

Credit Card # \_\_\_\_\_

Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

# USER NOTES

(Continued from Page 29)

```

155 ! FIND LOCATION OF FIRST
CR !079
160 LAST=POS(A$,CHR$(10),STAR
RT)!255
170 IF LAST=0 THEN 180 ELSE
200 !174
180 IF EOF(1) THEN 240 !030
190 LINPUT #1:B$ :: GOTO 160
!045
200 L$=SEG$(A$,START, LAST-2)
!189
210 A$=SEG$(A$,LAST+1,255)!1
37
220 PRINT #2:L$ :: PRINT L$
!071
230 GOTO 160 !239
240 PRINT #2:L$ :: PRINT L$
:: CLOSE #1 :: CLOSE #2 !122

```

## Reducing noise by reversing fan blade

This item was written by Michael Scheller of the Valley of the Sun TI99/ers (VAST) of Tempe, Ariz. It appeared in the group's newsletter, VAST News. Remember, all hardware modifications are taken at the risk of the reader.

Start this project by removing all cards and drives from your PEB. You will need a container for loose screws and washers, phillips and standard screwdrivers, a small Vise-grip, patience and time.

Remove all phillips screws that you see from the external PEB shell. This includes the two screws that hold the clasps that clip when you press the lid of the PEB down. I found simple hand turning with a screwdriver didn't work and had to nudge

the screws with a small Vise-grip.

After removing the screws, you will notice that you can move the outer shell/frame from the base. The back part of the PEB, where the slots, holes and back cards protrude, remains attached to the base. You will have to move the shell by sliding it away from the base. My PEB has a "push" on-off button so, when I removed the shell, the button popped off. I suggest you pull your button off first as a precaution.

You should now have two major components: Outer PEB shell and base with power supply/fan and card slots in plain view. Place your shell/button with screws (don't forget to put them in that container) to the side.

You notice the fan has a shroud around it, but you can remove the fan motor by placing a standard screwdriver through the holes on the back part, thereby removing two bolts and then you can work the plastic fan blades to remove the fan motor. You can now either disconnect the wires and switch them in or continue. I did the latter because I was reluctant to cut and reverse. Furthermore, I couldn't get my hands into the space between the power supply and the fan shroud.

I removed the screws that held the fan to the motor and turned the fan around. I then replaced the screws. Placing the fan motor back into the fan shroud, I replaced the bolts holding the fan motor to the shroud, slid the shell over the base and replaced all screws. It was easier to put the screws back into place than it was to remove them.

There is a noticeable reduction in noise

as a result of reversing the fan blade. The project took about four hours and cost nothing.

## Preachers, lawyers, used car salesmen

The following Extended BASIC program was written by the late Jim Peterson. Documentation is included within the program listing.

```

1 DATA PREACHERS, LAWYERS, USE
D CAR SALESMEN !213
10 GOTO 100 !179
11 J, T$, D, A$( ), SET, M$, CH, X, C
$( ), P$( ), I, XX$, K, ST, R, W, T( ),
C, F, L !100
30 CALL CLEAR :: CALL TITLEP
RINT :: CALL CHAR :: CALL DE
LSPRITE :: CALL SCREEN :: CA
LL COLOR :: CALL KEY :: CALL
HCHAR !179
40 !@P- !064
100 !PREACHER, LAWYER AND US
ED CAR SALESMAN by Jim Peter
son !070
110 !!131
120 CALL CLEAR :: RESTORE 1
:: FOR J=1 TO 3 :: READ T$ :
: CALL TITLEPRINT(5, T$) :: NE
XT J :: CALL CHAR(94, "3C4299
A1A199423C") !174
130 DISPLAY AT(1, 12) : "TIGERC
UB SOFTWARE" :: DISPLAY AT(3
, 12) : "^ TCX-1138" :: FOR D=1
TO 500 :: NEXT D :: CALL DE
LSPRITE(ALL) !252
140 CALL CLEAR :: CALL SCREE
N(4) :: DISPLAY AT(3, 1) : " PR
(See Page 31)

```

## BUGS & BYTES

### Is that by RossetTI or Whittler?

Actually, it's by Myrtle Calkins, the vice-president of the Portland Users of Ninety-Nines (PUNN), and appeared in the group's newsletter, *Wordplay*:

There was a young lady in PUNN  
 Who said to her daughter and son  
 Learn to use your TIs  
 And you'll turn out quite wise

And you also will have lots of fun

### Pulley connects to Internet

Clint Pulley, best known for his authorship of c99, is now the Internet representative on Delphi's TINET. This is a post formerly held by fellow Canadian Gary Bowser, whose name no longer appears as a sysop on TINET and whose departure from that post was unheralded there. (Bowser still retains a presence on TINET, last we checked.)

# USER NOTES

(Continued from Page 30)

```

EACHERS, LAWYERS AND": : "
  USED CAR SALESMEN":RPT$(" *
",28)!075
150 DATA 2,4,6 !240
160 DATA 12,14,16 !134
170 DATA 22,24,26 !137
180 A$(0)="N" :: A$(1)="Y" :
: CALL CHAR(61,"0")!030
190 FOR SET=3 TO 4 :: CALL C
OLOR(SET,16,5):: NEXT SET !0
61
200 DISPLAY AT(7,1):"PREACHE
R  LAWYER  SALESMAN": " 1=2
=3=  4=5=6=  7=8=9=":" Y
Y N    N N Y    Y N Y" !2
18
210 DISPLAY AT(11,1):" The 1
etters Y(es)and N(o)": "tell
you whether that person": "c
laims the numbers above": "th
em are his -" !083
220 DISPLAY AT(15,1):"but yo
u know that preachers": "neve
r lie, lawyers never": "tell
the truth, and used car": "sa
lesmen can't be trusted!" !0
30
230 DISPLAY AT(20,1):" Your
job is to get all the": "numb
ers under the person who": "r
eally owns them." !003
240 M$="" :: FOR CH=49 TO 57
:: M$=M$&CHR$(CH):: NEXT CH
:: FOR J=1 TO 9 :: RANDOMIZ
E :: X=INT(RND*LEN(M$)+1)::
C$(J), P$(J)=SEG$(M$,X,1):: M
$=SEG$(M$,1,X-1)&SEG$(M$,X+1
,LEN(M$)):: NEXT J !237
250 FOR I=1 TO 9 :: J=INT(10
*RND):: IF J<1 THEN 250 :: X
X$=P$(I):: P$(I)=P$(J):: P$(
J)=XX$ !089
260 NEXT I :: DISPLAY AT(24,
8):"Press any key" !171
270 CALL KEY(0,K,ST):: IF ST
=0 THEN 270 !251
280 DISPLAY AT(8,1):" ";P$(1
);"=";P$(2);"=";P$(3);"="
P$(4);"=";P$(5);"=";P$(6);
"=" ;P$(7);"=";P$(8);"=";
P$(9);"=" !254
290 CALL HCHAR(9,1,32,512)::
GOSUB 300 :: R=9 :: GOSUB 3
80 :: GOTO 390 !030
300 W=0 :: FOR I=1 TO 3 :: F
OR J=1 TO 3 :: T(I)=1 :: IF
C$(J)=P$(I) THEN 320 !145310
NEXT J :: T(I)=0 !035
320 NEXT I :: FOR I=4 TO 6 :
: FOR J=4 TO 6 :: T(I)=0 ::
IF C$(J)=P$(I) THEN 340 !129
330 NEXT J :: T(I)=-1 !230
340 NEXT I :: FOR I=1 TO 3 :
: T(I+6)=T(I)+T(I+3):: NEXT
I :: FOR I=1 TO 3 !084
350 IF T(I)+T(I+6)<>2 THEN 3
70 !127
360 NEXT I :: W=1 !112
370 RETURN !136
380 RESTORE 150 :: FOR J=1 T
O 9 :: READ C :: DISPLAY AT(
R,C):A$(ABS(T(J))):: NEXT J
:: R=R+2+(R=23)*16 :: RETURN
!018
390 RESTORE 150 :: FOR J=1 T
O 3 :: READ C :: ACCEPT AT(R
-1,C)SIZE(1)VALIDATE(DIGIT)B
EEP:P$(J):: DISPLAY AT(R-1,C
+1):"=" :: NEXT J :: F=1 ::
L=3 :: GOSUB 430 :: !110
400 IF F=0 THEN DISPLAY AT(R
-1,2)SIZE(6):RPT$(" ",6):: G
OTO 390 !056
410 GOSUB 460 :: IF F=0 THEN
DISPLAY AT(R-1,2)SIZE(6):RP
T$(" ",6):: GOTO 390 !215
420 GOTO 490 !058
430 FOR I=L-2 TO L :: FOR J=
1 TO 9 :: IF P$(I)=C$(J) THEN
450 !164
440 NEXT J :: DISPLAY AT(24,
1):"ILLEGAL CHARACTER" :: F=
0 :: RETURN !251
450 NEXT I :: RETURN !233
460 FOR I=L-2 TO L :: FOR J=
1 TO L :: IF P$(I)<>P$(J) THE
N 470 :: F=F+1 !087
470 NEXT J :: NEXT I :: IF F
<>4 THEN 480 :: RETURN !109
480 DISPLAY AT(24,1):"DUPLIC
ATES - NO FAIR!" :: F=0 :: R
ETURN !081
490 RESTORE 160 :: FOR J=1 T
O 3 :: READ C :: ACCEPT AT(R
-1,C)SIZE(1)VALIDATE(DIGIT)B
EEP:P$(J+3):: DISPLAY AT(R-1
,C+1):"=" :: NEXT J :: F=1 :
: L=6 :: GOSUB 430 :: !056
500 IF F=0 THEN DISPLAY AT(R
-1,12)SIZE(6):RPT$(" ",6)::
GOTO 490 !206
510 GOSUB 460 :: IF F=0 THEN
DISPLAY AT(R-1,12)SIZE(6):R
PT$(" ",6):: GOTO 490 !109
520 RESTORE 170 :: FOR J=1 T
O 3 :: READ C :: ACCEPT AT(R
-1,C)SIZE(1)VALIDATE(DIGIT)B
EEP:P$(J+6):: DISPLAY AT(R-1
,C+1):"=" :: NEXT J :: F=1 :
: L=9 :: GOSUB 430 :: !072
530 IF F=0 THEN DISPLAY AT(R
-1,22)SIZE(6):RPT$(" ",6)::
GOTO 520 !238
540 GOSUB 460 :: IF F=0 THEN
DISPLAY AT(R-1,22)SIZE(6):R
PT$(" ",6):: GOTO 520 !141
550 GOSUB 300 :: GOSUB 380 :
: IF W=1 THEN 560 :: GOTO 39
0 !228
560 DISPLAY AT(24,1):"YOU DI
D IT! PLAY AGAIN?" !080
570 CALL KEY(3,K,ST):: IF ST
<1 THEN 570 :: IF K=89 THEN
140 :: END !137
579 !@P+ !062
580 SUB TITLEPRINT(S,T$):: L
=LEN(T$)!179
584 GOTO 590 !159
586 S,T$,L,J,X,D !124
588 CALL SCREEN :: CALL MAGN
IFY :: CALL DELSPRITE :: CAL
L SPRITE !132
589 !@P- !064
590 CALL SCREEN(S):: CALL MA
GNIFY(2):: CALL DELSPRITE(AL
L)!065
600 FOR J=1 TO L :: X=ASC(SE
G$(T$,J,1)):: CALL SPRITE(#J
,X,16,J*(170/L),10+J*(200/L
)):: NEXT J :: FOR D=1 TO 200
:: NEXT D !054
601 !@P+ !062
602 SUBEND !168

MICROpendium pays $10 for items
sent in by readers and used in the User
Notes column. Send your tips and routi-
nes to MICROpendium User Notes,
P.O. Box 1343, Round Rock, TX 78680.

```

# The ONLY monthly devoted to the T199/4A

## 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      (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.

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 2.0 (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 version(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
<input type="checkbox"/> Series 1	\$9.00	\$7.00	\$5.00
<input type="checkbox"/> Series 2	\$9.00	\$7.00	\$5.00
<input type="checkbox"/> Series 3	\$9.00	\$7.00	\$5.00
<input type="checkbox"/> Series 4	\$9.00	\$7.00	\$5.00
<input type="checkbox"/> Series 5	\$9.00	\$7.00	\$5.00
<input type="checkbox"/> Series 6	\$9.00	\$7.00	\$5.00

SECOND CLASS

4470 W. 11TH ST  
 CHARLES COOK  
 P.O. BOX 497  
 WHEELING, OH 44694