

Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 10 Number 8

September 1993

\$3.50

Working on your algebra with

[MATRICES]

An assembly

POTPOURRI

Geneve 9640 sprites with

MY-COLOR

REVIEWS:

AEMS Macro Assembler/Linker

Program Image File Compressor

Membership Manager and

Newsletter Exchange

Minesweeper

The Spider's Guide to Funnelweb
Configurations

RLE+
\$ 9.95



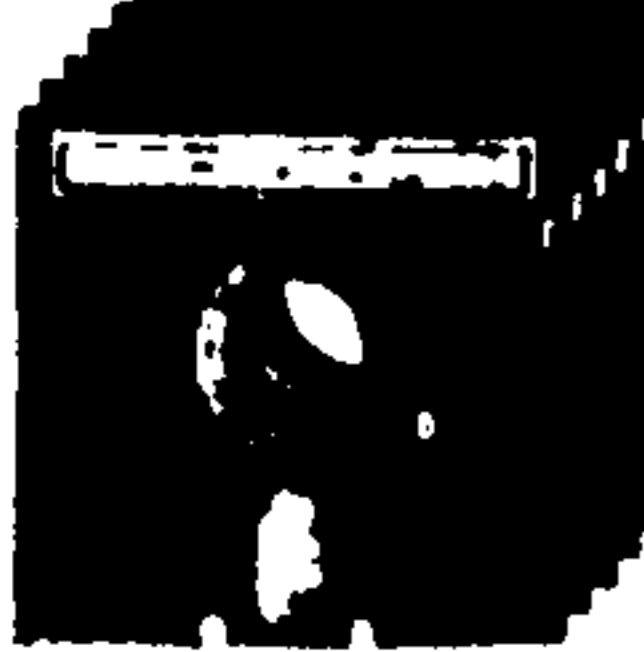
RLE+ IS A GRAPHICS PACKAGE OF OVER 60 RLE GRAPHICS WHICH ARE DRAWINGS, PHOTOGRAPHS AND OTHER WORKS OF ART THAT HAVE BEEN CREATED BY SCANNING PHOTOS, DRAWINGS AND OTHER ARTWORK. THESE FILES ARE FOUND BY THE DOZENS ON MANY BULLETIN BOARDS. WE HAVE TAKEN OVER 60 OF THE BEST ONES AND PACKAGED THEM WITH MAX RLE, A 99/4A PROGRAM THAT LETS YOU DISPLAY RLE FILES ON YOUR SCREEN AND PRINT THEM OUT OR SAVE THEM TO GRAPHX OR TI ARTIST FORMATS. LOADS IN EXBASIC AND IS EASY TO USE. COMES WITH COMPLETE DOCUMENTATION AND EASY TO FOLLOW INSTRUCTIONS. WORKS WITH ANY STAR/EPSON COMPATIBLE DOT MATRIX PRINTER AND PRINTS BIG POSTCARD SIZE PRINTOUTS WITH THE PRESS OF A KEY. ORIGINALLY PLANNED FOR FOUR DISK SIDES, WE HAVE ADDED EVEN ANOTHER DISK TO MAKE THIS A GREAT GRAPHICS PACKAGE.

COMES COMPLETE WITH FIVE DISK SIDES OF RLE FILES--OVER 60 ILLUSTRATIONS AND MAX-RLE. GET READY TO CREATE FANTASTIC PRINTOUTS YOU NEVER THOUGHT WERE POSSIBLE ON THE 99/4A!

\$9.95

Send \$2 for our new 1993 color log

NEW! RLE+ COMPANION DISKS



WE HAVE JUST RELEASED 4 NEW COMPANION DISKS PACKED FULL ON BOTH SIDES (KFLIPPIS) WITH RLE FILES. A FANTASTIC COLLECTION AT A SUPER LOW PRICE! THE TEX COMP RLE COMPANION DISKS 1 THROUGH FOUR ARE ONLY \$4.95 EACH OR ALL FOUR COMPANION DISK (8 DISK SIDES) FOR ONLY \$14.95 AND WE INCLUDE A FREE PLASTIC STORAGE CASE. A GREAT WAY TO GET AN ENTIRE LIBRARY OF GREAT GRAPHICS AT AN INCREDIBLY LOW PRICE!!!

SPECIAL COMBO OFFER
RLE+ AND RLE COMPANION TOGETHER FOR ONLY \$19.95
12 DISK SIDES OF PICTURES WITH FREE STORAGE CASE!

ANNOUNCING

TIPS



THE ULTIMATE GRAPHICS UTILITY FOR THE TI-99/4A COMPUTER

GREETING CARDS*BANNERS*FLYERS WITH THIS ONE UTILITY

ONLY **\$4.95** Per Disk

TIPS IS THE ULTIMATE GRAPHICS UTILITY FOR THE 99/4A. NOW WITH ONE LOW PRICED UTILITY (\$4.95) YOU GET A PROGRAM THAT LETS YOU CREATE YOUR OWN CUSTOM GREETING CARDS, BANNERS, FLYERS, POSTERS AND MUCH MUCH MORE. BEST OF ALL IS THE TIPS LIBRARY OF OVER 2000 GRAPHICS INCLUDING EVERY TYPE SUBJECT MATTER AND BORDERS AND MARGINS. THIS FANTASTIC FREWARE PROGRAM CONTAINS THE LARGEST PRINTSHOP LIBRARY THAT HAS EVER BEEN AVAILABLE FOR THE 4A. NOT ONLY CAN YOU PRINT OUT THIS LIBRARY IN THE SIZES YOU NEED, YOU CAN ALSO CONVERT ALL THE ARTWORK TO THE TI-ARTIST FORMAT AND ALL OTHER POPULAR 4A FORMATS. IN ADDITION TO THE TIPS PROGRAM DISK AT ONLY \$4.95, THE GIANT TIPS LIBRARY OF OVER 2000 GRAPHIC FILES COMES ON TEN TWO SIDED (FLIPPY) DISKS WHICH CAN BE PURCHASED SEPARATELY AT ONLY \$4.95 EACH OR IN A COMPLETE PACKAGE FOR A SPECIAL PRICE OF ONLY \$39.95 WITH THE TIPS PROGRAM DISK, A \$4.95 VALUE, INCLUDED AT NO ADDITIONAL CHARGE.

THE ULTIMATE TI-99/4A GRAPHICS-BANNER-CARD-FLYER PROGRAM NEW FOR 1990
TIPS PROGRAM DISK ALONE AND TIPS LIBRARY DISKS 1-10....\$4.95 +S&H
COMPLETE TIPS PACKAGE OF 11 DISKS INCLUDING ENTIRE TIPS LIBRARY AND TIPS PROGRAM DISK.....\$39.95 +S&H



New - TIPS DISKS 11 Through 16 \$4.95 each or 6 for \$24.95

TEX+COMP

P.O. Box 33084, Granada Hills, CA 91344

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

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% east of Mississippi. 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.

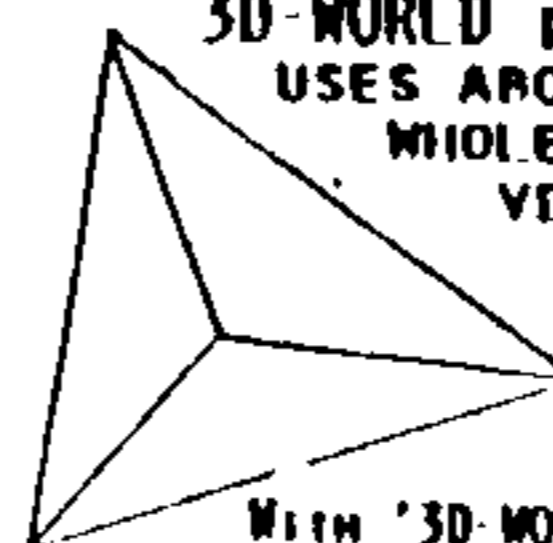
Only **\$4.95**

Software Carousel™ BRAND SOFTWARE GRAPHICS CODE GENERATOR

THE EASIEST TO USE AND MOST POWERFUL GRAPHICS UTILITY FOR YOUR TI 99/4A. GCG ELIMINATES THE NEED TO STRUGGLE WITH TEDIOUS HEX CODE CALCULATIONS. JUST DRAW YOUR GRAPHICS ON THE SCREEN AND GCG DOES THE JOB FOR YOU!!! THE HEXCODES GENERATED BY THIS UTILITY CAN BE DISPLAYED, PRINTED, SAVED AND EVEN AUTOMATICALLY TRANSFERRED INTO YOUR PROGRAM. YOU CAN STORE INDIVIDUAL FILES SUCH AS "TREE", "SPACESHIP", ETC. AND THEN USE THEM AGAIN AND AGAIN IN DIFFERENT PROGRAMS. \$9.95 (specify disk or cassette) Requires Exbasic & Jaytalk

3D-WORLD

CAD (COMPUTER AIDED DESIGN) COMES TO THE TI-99/4A UP TO NOW, 99/4A OWNERS COULD ONLY READ ABOUT THE WORLD OF COMPUTER AIDED DESIGN (CAD) SINCE THE LARGE AMOUNTS OF MEMORY REQUIRED MADE IT IMPOSSIBLE TO RUN ON SMALLER COMPUTERS.

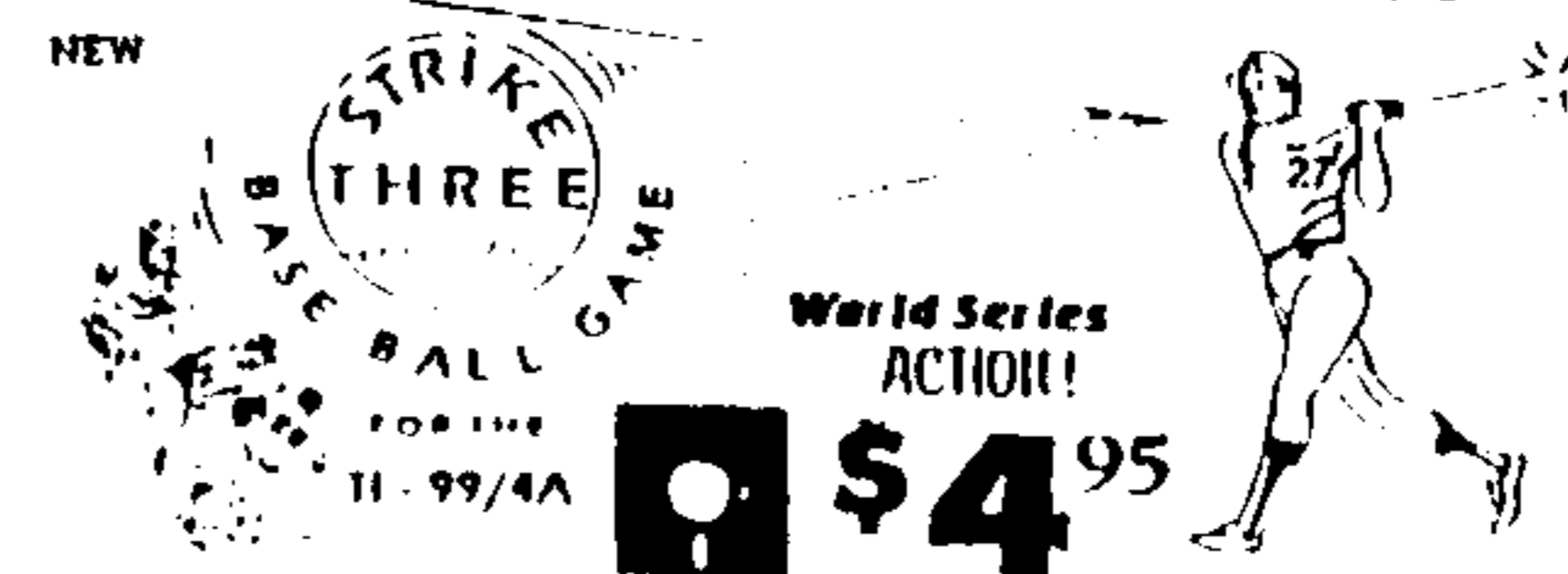


3D-WORLD IS WRITTEN IN THE 9900 MACHINE LANGUAGE WHICH USES ABOUT 8.5K OF PROGRAM SPACE LEAVING ALMOST THE WHOLE 32K RAM FOR DATA-ARRAYS. ABOUT 14K OF THE VDP RAM IS USED TO CREATE THE SCREEN DISPLAY AND THE DSR ROUTINES. THE PROGRAM COMES WITH COMPLETE DOCUMENTATION AND REQUIRES NO SPECIAL PROGRAMMING KNOWLEDGE. TWO DEMO PICTURES ARE ALSO INCLUDED.

WITH '3D-WORLD' YOU HAVE THE ABILITY TO CREATE THREE DIMENSIONAL PICTURES AND ILLUSTRATIONS ON YOUR 99/4A. THIS PROGRAM WHICH WAS WRITTEN IN GERMANY ALLOWS YOU TO NOT ONLY CREATE THREE DIMENSIONAL OBJECTS ON YOUR SCREEN, BUT ALSO TO DISPLAY IT WITH HIGH RESOLUTION GRAPHICS. TO ROTATE THE OBJECT ON THE SCREEN, TO ENLARGE OR REDUCE THE DRAWING AND TO PRINT OUT YOUR CREATION ON A DOT MATRIX PRINTER INCLUDING STAR/EPSON, AXIOM 100 AND 550.

Only **\$4.95**

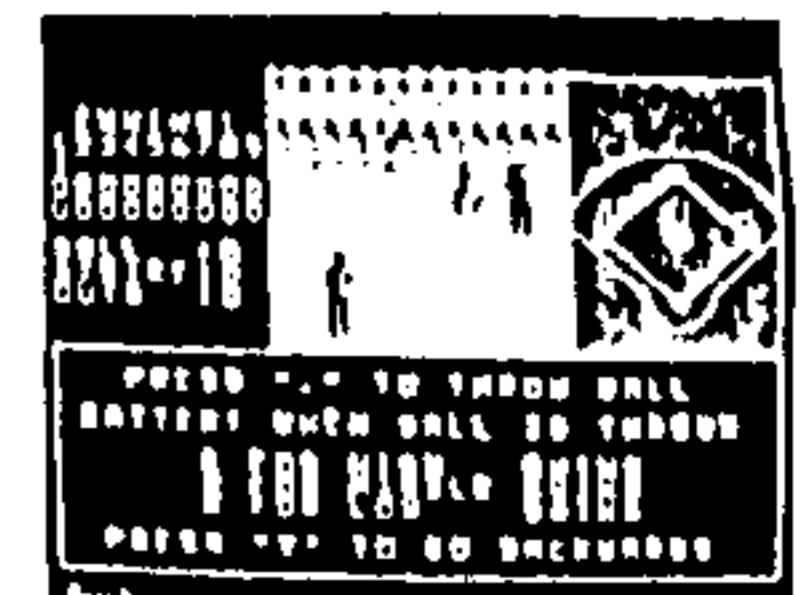
Share the excitement of Baseball



AT LAST! AN EXCITING ALL NEW 99/4A BASEBALL GAME. STRIKE THREE COMBINES IRVING BASEBALL ACTION AND SIMULATION IN A HIGH SPEED ARCADE QUALITY GAME WHICH YOU CAN PLAY AGAINST A FRIEND OR THE COMPUTER.

ON THE PITCHER'S MOUND YOU CAN SELECT YOUR FAVORITE PITCH FROM A FAST BALL TO A CURVE. WHEN AT BAT, USE YOUR BEST BATTER'S EYE TO PICK OUT THE PITCH YOU CAN HIT OUT OF THE BALL PARK. IF YOU GET A HIT, THE 99/4A DETERMINES YOUR OUTCOME BASED ON BIG LEAGUE STATS. A SPLIT SCREEN DISPLAY SHOWS YOUR POSITION ON THE DIAMOND AND ANOTHER SCREEN SHOWS THE PITCHER'S BEING TIMED.

GREAT GRAPHICS, PROFESSIONAL ANIMATION, FAST ACTION AND REALISTIC SOUND COMBINED WITH COMPUTERIZED OUTCOMES BASED ON REAL BIG LEAGUE BASEBALL MAKES STRIKE THREE A HOME RUN. SELF-CONTAINED ON 1 DISKETTE



be part of exciting TI-99/4A Baseball



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

TEX+COMP

DOLLAR DAYS

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

SOFTWARE SUPER SALE!

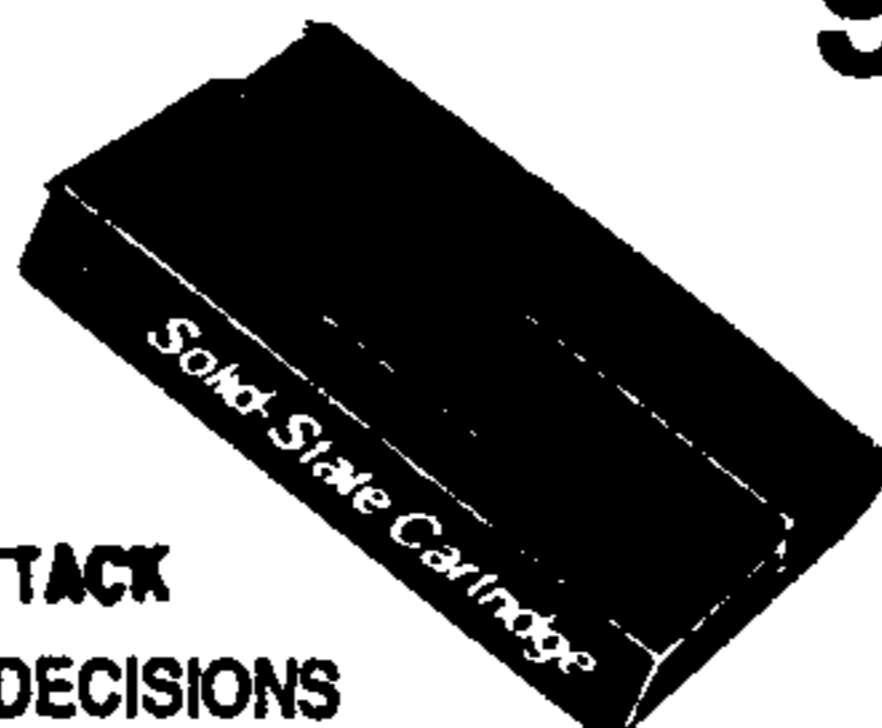
AS LOW AS \$1.95 EACH!!

WHEN BUYING TEN OR MORE OR BUY ANY AMOUNT AT ONLY \$2.95 EACH.

99¢ BOOK BLOWOUT!!!

PUBLISHER'S CLEARANCE ON BOOKS FOR THE 99/4A BUY FIVE OR MORE BOOKS FOR ONLY 99 CENTS EACH + s&h. BUY ANY AMOUNT LESS THAN 5 FOR ONLY \$1.49 EACH + s&h.

- | | |
|--------------------------------------|---|
| PHM 3027 ADDITION & SUBTRACTION I | PHM 3016 TAX INVESTMENT RECORD KEEPING*** |
| PHM 3028 ADDITION & SUBTRACTION II | PHM 3022 PERSONAL REAL ESTATE |
| PHM 3003 MILLIKEN DIVISION | PHM 3029 HUNT THE WUMPUS |
| PHM 3031 NUMERATION II | PHM 3031 THE ATTACK |
| PHM 3004 MILLIKEN INTEGERS | PHM 3220 MICROSURGEON |
| PHM 3110 MINUS MISSION | PHM 3057 MUNCHMAN |
| PHM 3020 MULTIPLICATION I | PHM 3110 CHRISHOLM TRAIL |
| PHM 3002 EARLY LEARNING FUN | PHM 3218 SUPER DEMON ATTACK |
| PHM 3003 BEGINNING GRAMMAR | PHM 3006 HOME FINANCIAL DECISIONS |
| PHM 3112 PARSEC | PHM 3035 TERMINAL EMULATOR II |
| PHM 3007 HOUSEHOLD BUDGET MANAGEMENT | PHM 3052 TOMBSTONE CITY |
| PHM 3044 PERS. REPORT GENERATOR** | PHM 3100 LOGO II* → Plus \$4 additional shipping & handling |
| PHM 3033 TI INVADERS | PHM 3065 EDITOR ASSEMBLER |
| PHM 3104 JAWBREAKERS II | PHM 3010 PHYSICAL FITNESS |
| | PHM 3140 MUNCHMOBILE |



Special purchase

DONKEY KONG featuring Mario \$4.95 (regular \$9.95)

ALSO INCLUDED ARE THE FOLLOWING ARANISOFT MODULES: DEFENDER, PROTECTOR II, PICNIC PARANOIA

*ADD AN ADDITIONAL \$4.00 EACH ON THESE THREE TITLES DUE TO WEIGHT ** FOR USE WITH PERSONAL RECORD KEEPING, *** DISK DRIVE REQUIRED PERSONAL RECORD KEEPING AVAILABLE AT \$14.00

GAMES TIS PLAY THE ELEMENTARY TI COMPUTER PLAYGROUND (TI) INTRO TO ASSEMBLY LANGUAGE PROGRAMS FOR THE TI COMPUTER TI-99/4A GAME PROGRAMS STARTER PACK I STARTER PACK II GAME WRITERS PACK I GAME WRITERS PACK II

SPECIAL PACKAGE AND PRICING ON Adventure THE ADVENTURE SERIES...

SPECIAL PACKAGE AND PRICING ON THE ADVENTURE SERIES... ON DISK OR CASSETTE (specify) THE COMPLETE SCOTT ADAMS SERIES I (ADVENTURELAND, PIRATE ISLAND, THE COUNT MISSION IMPOSSIBLE, VOOBDOO CASTLE, STRANGE ODYSSEY, MYSTERY FUN HOUSE, PYRAMID OF DOOM, GHOST TOWN, SAVAGE ISLAND I, SAVAGE ISLAND II, AND GOLDEN VOYAGE PLUS KIMBLY IRONHEART BONUS ADVENTURE

ONLY \$12.95 FOR ALL 13.

THE SCOTT ADAMS ADVENTURE SERIES II (THE MULE, SPIDERMAN, BUCKAROO BANZAI, AND THE BORGERS OF CLAYBOROUGH CASTLE PLUS 2 BONUS ADVENTURES

ONLY \$9.95 FOR ALL 6.

COMMAND MODULE FOR ABOVE ADVENTURES (reg \$8.95) ONLY 95 CENTS WITH EITHER OF THE ABOVE SERIES

HINT BOOK FOR ALL OF THE ABOVE SCOTT ADAMS ADVENTURES ONLY 95 CENTS WITH EITHER OF THE ABOVE SERIES.

NEW First time on disk — Return to Pirate's Island with Graphics. X BASIC or Editor/Assembler Load

ADVENTURE EDITOR NOW WRITE AND EDIT YOUR OWN GAMES FOR THE TI ADVENTURE MODULE!

TEX COMP is proud to announce the 99/4A Adventure Editor — an all new program which allows you to write your own adventure programs — Scott Adams move over! This disk or cassette based program allows:

1. Any existing adventure for the Adventure command module to be edited, altered, listed and copied from any storage medium to any other storage medium (i.e. tape to disk or vice versa)
2. New adventure games can be created freely using a template game as a start up step. These games can be conceived using the full power and capabilities of the TI Adventure command module.

The program is written for Mini-Memory or Editor/Assembler command modules (cassette or disk) and offers a full screen editor and a special easy to use, mnemonic language called A.P.L. (Adventure Programming Language) and comes with a complete and detailed 75 page manual.

TEX COMP is also organizing a new TI Adventuring Users Group which will allow low cost exchange of user written adventures between members Available NOW From TEX+COMP NEW LOW PRICE Specify Mini-Memory (Cassette) or Editor/Assembler (Disk) Versions



\$4.95



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

HOME ENTERTAINMENT

DISKETTE PROGRAMS

- | | |
|---|------|
| PND 5002 TI-TREE (TE-II req. for speech) | 4.95 |
| PND 5010 Mystery Melody | 4.95 |
| PND 5015 Oldies But Goodies I | 4.95 |
| PND 5017 Oldies But Goodies II | 4.95 |
| ***SPECIAL Oldies But Goodies I & II | 7.95 |
| PND 5025 Sat. Night Bingo (Ex-Basic & Speech) | 4.95 |
| PND 5037 Draw Poker (Ex-Basic) | 4.95 |

CASSETTE PROGRAMS

- | | |
|---|------|
| PNT 6002 TI-TREE (TE-II req. for speech) | 4.95 |
| PNT 6010 Mystery Melody | 4.95 |
| PNT 6015 Oldies But Goodies I | 4.95 |
| PNT 6017 Oldies But Goodies II | 4.95 |
| ***SPECIAL Oldies But Goodies I & II | 7.95 |
| PNT 6025 Sat. Night Bingo (Ex-Basic & Speech) | 4.95 |
| PNT 6037 Draw Poker (Ex-Basic) | 4.95 |

COMPUTER PROGRAMMING AIDS

DISKETTE PROGRAMS

- | | |
|--|-------|
| PND 5007 Teach Yourself 99/4A Basic | 4.95 |
| PND 5019 Teach Yourself Extended Basic | 4.95 |
| PND 5004 Programming Aids I | 4.95 |
| PND 5005 Programming Aids II | 4.95 |
| PND 5077 Programming Aids I, II, III | 9.95 |
| PND 5067 Beginning Basic Tutor | 4.95 |
| PND 5076 Text to Speech (Ex-Basic Speech) | 4.95 |
| PND 5098 TI Forth & manual (Ed/Assem req.) | 19.95 |
| PND 5078 TI Forth Demo Disk (Ed/Assem) | 4.95 |
| PND 5079 TI Forth Source Code (2 disks) | 4.95 |

CASSETTE PROGRAMS

- | | |
|--|------|
| PNT 6006 Programming Aids I | 4.95 |
| PNT 6007 Teach Yourself 99/4A Basic | 4.95 |
| PNT 6019 Teach Yourself Extended Basic | 4.95 |
| PNT 6067 Beginning Basic Tutor | 4.95 |

MATH AND ENGINEERING

specify disk or tape with order

- | | |
|--------------------------------|------|
| Math Routine Library | 4.95 |
| Electrical Engineering Library | 4.95 |
| Graphing Package | 4.95 |
| Structural Engineering Library | 4.95 |
| AC Circuit Analysis | 4.95 |

***SPECIAL---ALL 5 OF THE ABOVE ON DISK OR TAPE---17.95

\$4.95 Per Disk **NEW BONUS SAVE AS MUCH AS 33%** BUY FIVE DISKS GET ONE FREE BUY TEN DISKS GET THREE FREE BUY FIFTEEN DISKS GET FIVE FREE

DISKETTE PROGRAMS

- | | |
|---|------|
| PND 5009 Music Skills Trainer | 4.95 |
| PND 5011 Computer Music Box | 4.95 |
| PND 5018 Market Simulation | 4.95 |
| PND 5030 Speak & Spell II (Ex-Basic req.) | 4.95 |
| PND 5031 Speak & Math (TE-II req.) | 4.95 |
| PND 5042 Spell Writer (TE-II req.) | 4.95 |
| PND 5026 Bridge Bidding I | 4.95 |
| PND 5039 Bridge Bidding II | 4.95 |
| PND 5041 Bridge Bidding III | 4.95 |
| SPECIAL!! BRIDGE BIDDING I, II & III | 9.95 |
| PND 5020 Music Maker Demo (use with module) | 4.95 |
| PND 6010 Mystery Melody (stop the music quiz) | 4.95 |

CASSETTE PROGRAMS*

*see disk versions for requirements i.e. TE-II

- | | |
|---|------|
| PNT 6009 Music Skills Trainer | 4.95 |
| PNT 6010 Mystery Melody | 4.95 |
| PNT 6011 Computer Music Box | 4.95 |
| PNT 6018 Market Simulation | 4.95 |
| PNT 6031 Speak & Math | 4.95 |
| PNT 6042 Spell Writer | 4.95 |
| PNT 6026 Bridge Bidding I | 4.95 |
| PNT 6039 Bridge Bidding II | 4.95 |
| PNT 6041 Bridge Bidding III | 4.95 |
| SPECIAL!! BRIDGE BIDDING I, II & III | 9.95 |
| PNT 6020 Music Maker Demo (use with module) | 4.95 |

MANAGEMENT AND BUSINESS

DISKETTE PROGRAMS

- | | |
|--|------|
| PND 5001 Mailing List (upgraded version) | 4.95 |
| PND 5003 Personal Financial Aids | 4.95 |
| PND 5021 Checkbook Manager | 4.95 |
| PND 5022 Finance Manager | 4.95 |
| PND 5024 Inventory Management | 4.95 |
| PND 5027 Invoice Management | 4.95 |
| PND 5029 Cash Management | 4.95 |
| PND 5038 Lease/Purchase Decisions | 4.95 |
| PND 5075 TI/Multiplan upgrade disk | 4.95 |

CASSETTE PROGRAMS

- | | |
|-----------------------------------|------|
| PNT 6003 Personal Financial Aids | 4.95 |
| PNT 6038 Lease/Purchase Decisions | 4.95 |

TEX+COMP

P.O. Box 33084, Granada Hills, CA 91344



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



Send \$2 for our new 1993 catalog

AUTHORIZED DEALER

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for \$25 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664. Second-class postage paid at Austin, Texas, and additional mailing offices. 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 \$30.25 (Mexico); \$32.50 (Canada); \$30.00, surface mail to other countries; \$42 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 TINET: MICROpendium

GENie: J.Koloen

Internet E-mail: jkoloen@holonet.net

John Koloen.....Publisher

Laura Burns.....Editor

Regena on BASIC

Matrices.....Page 8

The Art of Assembly

Another potpourri.....Page 11

Geneve 9640

Sprites with MY-Color.....Page 15

Hardware project

Modification for CorComp RS232 cards.....Page 18

Reviews

AEMS Macro Assembler/LinkerPage 21

Program Image File CompressorPage 23

MICRO-Reviews: Membership Manager and Newsletter Exchange, Minesweeper, The Spider's Guide to Funnelweb Configuration.....Page 25

User Notes

Transliterator, a hardware fix for shadows, and slowing down the cursor in DM-1000.....Page 26

ClassifiedPage 31

Departments

Bugs and BytesPage 16

CommentsPage 5

FairsPage 19

FeedbackPage 7

Reader to ReaderPage 7

System of the Month.....Page 20

*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.

COMMENTS

Chicago's 10th fair!

Here it is, 10 years since the Chicago Users Group first thought up the idea of having a fair for TI users. They have had one every year since. That's quite a record for longevity.

Attending a fair gives you a clearer idea about how this intrepid little computer can be used. It is a good opportunity to see products "in action," so you can tell whether they are really something you want to add to your system. All in all, it is a good opportunity for both novices and old hands to learn more about their favorite computer.

Don Walden, Chicago user group president, says this year's fair will be modeled after the annual Lima (Ohio) fair that's held every spring. That means vendors will be paying less and the amenities won't be as plentiful as in years past. The pre-fair get-together won't be held and the location isn't as pricey as in years past. However, the fair will bring together TIers who'll be sharing their expertise and experience with all comers. I'm looking forward to it.

CECURE SLOWED DOWN

Don Walden of Cecure Electronics wants to apologize to customers who have received items back late. The company has recently moved into temporary quarters, and Tim Tesch, who has worked as a technician there during the summer, has returned to his studies and can only work part time.

COMPILER COMING ALONG, SLOWLY

Bruce Harrison reports progress on the Herculean task of writing a BASIC compiler for the TI.

"The compiler is coming along, but very slowly, as I expected it would. It's not bad for a product that was supposed to be impossible, but as I'm sure you're aware, things like this get developed one small part at a time, and as of today we're handling FOR-NEXT loops, PRINT, GOSUB, GOTO, END and STOP. The next major hurdle is the IF-THEN structure, and that promises to be one of the most challenging parts.

"Even FOR-NEXT was a struggle, because there are so many possible variations allowed. At present, it can handle loops of the form FOR I=S TO R STEP T, or FOT I=1 TO 20, and permutations of these, including such as FOR I=-S. Nesting of loops has been made 'infinite' by adapting a method suggested by Harry Wilhelm. I'm not sure how many FOR-NEXTs can be nested in Extended BASIC, but there's no limit in the compiler.

LOOKING FOR INFO ON GENEVE LOGO PATCH

A reader called looking for a patch for Logo II that would allow it to run on the Geneve. I've asked around, but haven't found one. I know that Myarc had planned on providing a software patch for Logo II but I haven't found it anywhere. Anyone who has any suggestions is encouraged to write or call.

—JK

HORIZON COMPUTER

NEW HORIZON 4000 RAM DISK

Up To 8 Meg on Single Layer	Built In RAMBO and PHOENIX	
RAMDISK BARE BOARD, MANUAL, ROS 814 F		\$65.00
ZEROk Kit, Above + Parts, NO MEMORY		\$105.00
128k Kit	256k Kit	\$150.00
384k Kit	512k Kit	\$200.00
1 Meg Kit	1.5 Meg Kit	\$425.00
2 Meg Kit	8 Meg Kit	\$CALL
ADD \$30 FOR ASSEMBLY ON RAMDISK OR P-GRAM		
128k Memory Chips NOW only \$25	512k Memory Chips	\$125
P-GRAM 72k Kit	P-GRAM+ 190k Kit	\$175
P-GRAM CLOCK	P-GRAM Upgrade	\$30
HORIZON MOUSE	MEMEX 504k	\$200
M G Eprom for CorComp Disk Controller		\$35
The NEW SCSI Hard Disk Controller is STILL waiting on the DSR		

Order From

800 MILLS SERVICES

166 DARTMOUTH DRIVE

TOLEDO OH 43614-2911

Call 419 385 5946 voice 6am to 8pm EST

419 385 7484 BBS

Shipping is included in pricing for US & Canada.... All other Countries add \$10 to Order

Prices subject to change

Call for latest prices

Ohio Residents ADD 6.5% ST

Visa & M C

Any HORIZON CARD REPAIRS \$50



ELECTRONICS INC.

SALE * SALE * SALE * SALE * SALE

128K X 8 LP 80ns MEMORY HITACHI \$19.95

Look again...\$19.95

Quantities limited to 8 per customer.

SORRY, NO DEALER SALES.

NEW * NEW * NEW * NEW * NEW * NEW * NEW

\$ 1 5 * \$ 1 5 * \$ 1 5

CFORM * C F O R M * CFORM

\$ 1 5 * \$ 1 5 * \$ 1 5

FORMAT YOUR HARD DRIVE FROM MDOS

* USE 34 SECTORS PER CYLINDER *

ADD AN ADDITIONAL 6% STORAGE

1.2 MEG TO YOUR 20 MEG HARD DRIVE

2.4 MEG TO YOUR 40 MEG HARD DRIVE




HARDWARE/SOFTWARE

NEW * NEW * NEW * NEW * NEW * NEW * NEW

MBPII CARD

BARE BOARD, MANUAL, SOFTWARE \$27

ABOVE-PLUS ALL PARTS-YOU SOLDER \$67

COMPLETED ASSEMBLY \$87

CLOCK CHIP MM58167AN \$ 9

CLOCK CRYSTAL 32.768 KHZ \$ 2

ANALOG-DIGITAL CHIP ADC0809CCN \$ 6

VOICE 414-679-4343

FAX 414-679-3736

BBS 414-422-9669

MASTER CARD AND VISA OR
C.O.D. ORDERS USE OUR TOLL FREE

ORDER ONLY PHONE NUMBER:

1-800-959-9640

ORDERS ONLY!

MICRopendium Back Issue Sale

Because of the subscription price increase that went into effect in August, this is your last chance to get back issues at the current rate of \$2.50 each. In fact, depending on how many you buy, the price can be as little as \$2.00 per back issue and we pay the postage on U.S. orders. Residents of Canada and Mexico add 30 cents for each issue, other countries 50 cents each issue (surface mail) or \$1.50 each issue (air mail). All payments must be in U.S. funds.

Here's how to order:

This Sale Ends October 15, 1993

List the month and year of each back issue (or the volume and number) below. If you order 10 or fewer issues, the price is \$2.50 each. If you order more than 10, the price is \$2.00 each.

NOTICE: These issues are not available:

Volume 1, Nos. 1 & 2 (February and March, 1984); Volume 2, No. 1 (February 1985);
Volume 4, No. 9 (October 1987)

Enter the month and year or the volume and number for each back issue.

_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Mail completed order forms to:
MICRopendium Back Issues
P.O. Box 1343
Round Rock, TX 78680

Name _____		Circle payment method		
Address _____		Check/MO	Visa	M/C
City _____		FOR CREDIT CARD ORDERS ONLY		
State _____ ZIP _____		Credit Card # _____		
		Exp. Date _____		
		Signature _____		

FEEDBACK

Praise for Cecure

This letter addresses the recent discussions of repair to Myarc products and Geneves by Cecure Electronics. In May I took my Geneve, HFDC and a 512K memory card to the Lima fair and gave them to Don Walden to be upgraded and repaired. A few weeks later Don called to ask some questions about the repairs and what I wanted and he then told me that the repairs were moving along smoothly. After 30 days I called and asked about the progress and was told that the cards were being shipped that day. Imagine my surprise when the next Tuesday the cards were delivered.

My Geneve now works better than ever. The new Boot-ROM BIOS is the greatest thing since sliced bread. It loads fast and I can even use different versions because of the loading menu.

Anyone thinking of sending something to Don for repair should not have any worries about getting things returned in a timely fashion. Don is always ready to help with questions and problems with anything Myarc. I am using the Geneve and Gen-Tri to write this letter even though I have WordPerfect on a PC next to the Geneve.

Keep on Tling!

Gary Kuehn
Pittsburgh, Pennsylvania

Controlling printer

I do not make a practice of writing letters to the editor, but I wish to express my satisfaction with MICROpendium. I wish that it had been around earlier in the life of the TI99/4A and that I had become aware of its existence sooner. I have gleaned a great deal from the pages of this newsletter (Regena's BASIC, User Notes, et al.)

In the August issue, "Getting More Out of Your Printer" finally showed me how to use my Star NX-1000II. Up to now, I had been able to control printing only with the setup switches on the printer. Sure, I had the manual and saw that control codes could do lots of things. But I could not understand how to enter these codes into my text. With the explanations and especially with the examples to type in

and try, now I know.

I have created a Control File with many printer functions (Double Width, Italicizing, Subscript, Superscript, etc.) that I would have like to have used before. Each Control Code is preceded with .CO and a description. I can merge the Control File into the beginning of any text file and copy the desired code into the appropriate place in the text. This keeps me from looking through the manual for functions I want and has the added advantage that the typing of the codes has already been tested.

I have preformatted business and personal letter samples on my TI-Writer File, as I expect most people do. However, I have my screen margins set to 39 characters, so that I never have to scroll back and forth horizontally to see my text. I never did see what the big deal was with 80-column viewing. For years I worked with IBM main frame terminals and the viewing on those was 72 columns. Of course, the 39-character display is preempted at print time by the left and right margin print settings.

I still have a Seikosha GP-100 on my TI. I use that to run off proof listings and save wear and tear on my Star. That Seikosha earned its keep many times over. One of my jobs in the company where I worked was billing the various departments for their use of our large IBM main frame. More than 90 terminals were scattered throughout three large buildings and in several cities and states. The accumulation of hourly usage was done by *Job Accounting*. But the formulas for allocation of the charges was kept on my TI99/4A here at home. Each month, I would write down the hours used, come home and enter that number in my TI, press enter and the GP-100 would print out the numbers for the accounting department to use.

I had two problems: getting ribbons for the machine and getting 20-pound paper to go through. For a while I was re-inking the ribbons (used numbering machine ink) until I found a distributor in Houston, Texas, that stocked them. I found some 15-pound paper that works okay.

A tip for people who would like to use the & instead of typing *and* all the time. Transliterate: .TL 62:38 will let the shift period (greater than) print the amper-

sand. This beats trying to revise the program coding to change the underline instruction.

Leslie R. Hadley
Levittown, Pennsylvania

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.

Reader to Reader

□ John Collins, 2992 Ribbon Court, S.E., Fort Myers, FL 33905, asks whether he can use a fax machine with the TI99/4A computer. He also notes that he works at Goodwill Industries, and says anyone looking for computer supplies should write to him

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

COMPETITION COMPUTER SOLUTIONS

2219 S Muskego Milwaukee WI 53215 414-672-1600 Fx 672-8977

800-471-1600

(Nationwide & Canada orders only)

**BUY - SELL - TRADE
HARDWARE - SOFTWARE
Dual 1/2 HT drive kit \$65**

**P.E. Box Complete \$125
Star NX-1001 Printer \$165**

**DS/DD 1/2 Ht Drive \$30
2400 Baud Modem \$89**

TI-99/4A Console \$35

**Catalog \$2 Open Daily 9-5 Sat 10-3
Huge Genuine TI Inventory Since 1982
Bankcards, Discover, Checks & UPS/COD
100 Plato Titles in Stock!**

BASIC

Matrices

By REGENA

I have been reviewing a course in Linear Algebra and working with matrices. The course is still traditionally taught with the student multiplying matrix A by matrix B by hand, inverting matrices by various methods, finding the determinant and solving simultaneous equations using step-by-step calculations. The methods are very adaptable to computers. Many versions of BASIC (and other languages) have MAT or similar forms of commands that can do matrix arithmetic. Calculators, of course, also have matrix capabilities. I think we ought to use the capabilities of calculators and computers to do the work and perhaps change the method of teaching linear algebra and mathematics applications courses.

Our TI99/4A does not have simple commands for matrix arithmetic, but the program this month will allow you to enter matrices row by row and multiply two matrices, invert a matrix, find the determinant of a matrix or solve a system of simultaneous equations. I have limited this program to matrices of 10 rows by 10 rows, but you may use DIMension statements and use larger matrices.

The first section of the program is used to multiply two matrices, such as $[A][B] = [C]$. Remember, the order of multiplication does matter; $[A]$ times $[B]$ does not necessarily equal $[B]$ times $[A]$. First enter the number of rows in matrix A, then enter the number of columns in matrix A. The number of rows of matrix B must equal the number of columns in matrix A for multiplication, so that value is not entered again. Next enter the number of columns of matrix B.

The matrix elements are then entered row by row. For example, if you have a matrix of

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

the computer will prompt you, and you will first enter $a(1,1)$, then $a(1,2)$, then $a(1,3)$ and so forth across the row. You will then enter the second row of numbers, $a(2,1)$, $a(2,2)$, $a(2,3)$ and so forth. The process continues for all the rows in the matrix.

For this first section, you need to enter two matrices. The results will be given as a matrix C, and the numbers are listed row by row.

The second section lets you enter a matrix, then the inverted matrix A^{-1} is returned as matrix W.

The third section lets you enter a matrix, and the determinant is returned. In 2x2 and 3x3 matrices, the calculations are multiplications across diagonal paths, but larger sizes use a different method. The matrix is first diagonalized, then the product of the diagonal elements is the determinant. For computer calculations, I actually did an upper triangularization bringing factors of the diagonal outside the matrix.

The fourth section solves a system of simultaneous equations. The $[A]$ matrix is the matrix of coefficients of the unknowns, and the $[B]$ matrix is the constant vector. An example is this system.

$$x_1 + 2x_2 + 3x_3 = 5$$

$$2x_1 + 5x_2 + 3x_3 = 3$$

$$x_1 + 8x_3 = 17$$

In matrix form this system can be written as $[A][X] = [B]$ or

$$\begin{bmatrix} 1 & 2 & 3 & x_1 & = & 5 \\ 2 & 5 & 3 & x_2 & = & 3 \\ 1 & 0 & 8 & x_3 & = & 17 \end{bmatrix}$$

The computer will prompt you to enter the degree of the matrix, which is 3 (a square matrix of three rows and three columns). You will then enter 1 for $A(1,1)$; 2 for $A(1,2)$; and 3 for $A(1,3)$; then

5 for $B(1)$. Continue with the second row entering 2, 5, 3, then 3. The third row entered is 1, 0, 8, then 17. The computer will return the result $X(1)=1$, $X(2)=-1$ and $X(3)=2$.

Lines 120-280 print the main menu screen and branch appropriately in the subroutine. After the subroutine is completed for that problem, you are asked whether you have another problem. Press Y for yes or N for no, and the program branches appropriately.

Lines 290-370 are a subroutine for entering the degree of a square matrix (the number of rows). Lines 380-430 print a sample matrix, then Lines 440-510 are a subroutine to receive the input for a square matrix.

Lines 520-720 are a subroutine to invert the matrix. As the matrix A is entered, an identical work matrix W is defined. It is the actual matrix inverted. Lines 730-880 are a subroutine to exchange rows if a zero diagonal element is encountered. If all remaining rows contain a zero in the needed diagonal element, the determinant is zero and the matrix is not invertible.

Lines 890-960 are a subroutine to keep track of the lines printed on the screen. If you choose a larger matrix, the printing will stop until you press a key, rather than scrolling too quickly.

Lines 970-1470 are the subroutine to multiply two matrices. Lines 1480-1680 are the subroutine to invert a square matrix. Lines 1690-1980 are the subroutine to find the determinant of a square matrix. Lines 1990-2310 are the subroutine to solve a system of equations.

If you would like to save typing effort, you may have a copy of this program by sending \$4 to REGENA, 918 Cedar Knolls West, Cedar City, UT 84270. Please be sure to specify that you need "MATRIX" for the TI99/4A and whether you need cassette or diskette.

(Program begins on Page 9)

**Attend a
TI Faire
this year!**

REGENA ON BASIC—

```

100 REM MATRIX !175
110 REM BY REGENA !071
120 CALL CLEAR !209
130 PRINT " ** LINEAR ALGEBR
A **" !073
140 PRINT : : "CHOOSE:" !207
150 PRINT : "1 MULTIPLY TWO M
ATRICES" !146
160 PRINT : "2 INVERT MATRIX"
!070
170 PRINT : "3 FIND DETERMINA
NT" !249
180 PRINT : "4 SOLVE SYSTEM O
F EQUATIONS" !163
190 PRINT : "5 END PROGRAM" !
137
200 CALL KEY(3,K,S)!190
210 IF (K<49)+(K>53)THEN 200
!222
220 CALL CLEAR !209
230 ON K-48 GOSUB 970,1480,1
690,1990,2320 !062
240 PRINT : "ANOTHER PROBLEM?
(Y/N)" !246
250 D$="" !237
260 CALL KEY(3,K,S)!190
270 IF (K=89)+(K=121)THEN 12
0 !188
280 IF (K=78)+(K=110)THEN 23
20 ELSE 260 !168
290 PRINT "ENTER DEGREE OF T
HE MATRIX" !114
300 INPUT "N = ":N !043
310 IF N<10 THEN 340 !138
320 PRINT : "N MUST BE < 10":
:!203
330 GOTO 290 !114
340 IF N>1 THEN 370 !120
350 PRINT : "1<N<10 TRY AGAI
N": :!170
360 GOTO 290 !114
370 RETURN !136
380 PRINT : "INPUT THE VALUES
ROW BY ROW:" !218
390 PRINT : "A(1,1),A(1,2),A(
1,3),..." !161
400 PRINT : "A(2,1),A(2,2),A(
2,3),..." !164
410 PRINT ".": ".": ".": ".": !232
420 PRINT "A(N,1),A(N,2),...
,A(N,N)": : :!125
430 RETURN !136
440 FOR I=1 TO N !140
450 FOR J=1 TO N !141
460 INPUT "A("&STR$(I)&","&S
TR$(J)&") = ":A(I,J)!213
470 W(I,J)=A(I,J)!188
480 NEXT J !224
490 PRINT !156
500 NEXT I !223
510 RETURN !136
520 REM INVERT MATRIX A !008
530 FOR C=1 TO N !134
540 IF W(C,C)<>0 THEN 580 !1
84
550 GOSUB 740 !054
560 IF D$<>"D" THEN 580 !054
570 RETURN !136
580 W(C,C)=1/W(C,C)!118
590 FOR D=1 TO N !135
600 IF (D-C)=0 THEN 660 !000
610 W(D,C)=W(D,C)*W(C,C)!122
620 FOR E=1 TO N !136
630 IF (E-C)=0 THEN 650 !247
640 W(D,E)=W(D,E)-W(D,C)*W(C
,E)!064
650 NEXT E !219
660 NEXT D !218
670 FOR E=1 TO N !136
680 IF (E-C)=0 THEN 700 !041
690 W(C,E)=-W(C,C)*W(C,E)!06
2
700 NEXT E !219
710 NEXT C !217
720 RETURN !136
730 REM SUB TO SWITCH ROWS !
228
740 FOR F=C+1 TO N !141
750 IF W(F,C)=0 THEN 850 !00
9
760 FOR E=1 TO N !136
770 DW=W(C,E)!088
780 W(C,E)=W(F,E)!191
790 W(F,E)=DW !091
800 NEXT E !219
810 DB=B(C)!054
820 B(C)=B(F)!165
830 B(F)=DB !057
840 GOTO 880 !194
850 NEXT F !220
860 PRINT "DETERMINANT = 0"
!090
870 D$="D" !050
880 RETURN !136
890 PR=PR+1 !189
900 IF PR<20 THEN 960 !077
910 PR=0 !089
920 PRINT "PRESS ANY KEY TO
CONTINUE";!022
930 CALL KEY(3,K,S)!190
940 IF S<1 THEN 930 !174
950 CALL HCHAR(24,3,32,25)!2
26
960 RETURN !136
970 PRINT "MULTIPLYING MATRI
CES" !077
980 PRINT "[A][B] = [C]" !21
8
990 PRINT : "MATRIX A" !086
1000 INPUT " NUMBER OF ROWS:
":MA !016
1010 MA=INT(MA)!022
1020 IF (MA>0)+(MA<11)=-2 TH
EN 1050 !241
1030 PRINT : "ROWS MUST BE FR
OM 1 TO 10": :!223
1040 GOTO 990 !048
1050 INPUT " NUMBER OF COLUM
NS: ":RA !238
1060 RA=INT(RA)!032
1070 IF (RA>0)+(RA<11)=-2 TH
EN 1100 !045
1080 PRINT : "COLUMNS MUST BE
FROM 1 TO 10": :!184
1090 GOTO 1050 !109
1100 PRINT : "MATRIX B" !087
1110 PRINT " NUMBER OF ROWS:
";RA !030
1120 INPUT " NUMBER OF COLUM
NS: ":NB !235
1130 NB=INT(NB)!026
1140 IF (NB>0)+(NB<11)=-2 TH
EN 1170 !109
1150 PRINT : "COLUMNS MUST BE
FROM 1 TO 10": :!184
1160 GOTO 1100 !159
1170 PRINT : : "RESULT WILL B
E MATRIX WITH" !022
1180 PRINT MA;"ROWS AND";NB;
"COLUMNS": :!060
1190 PRINT "INPUT THE VALUES
ROW BY ROW FOR MATRIX A": :
!190
1200 FOR I=1 TO MA !204
1210 FOR J=1 TO RA !210
1220 INPUT "A("&STR$(I)&","&
STR$(J)&") = ":A(I,J)!213
1230 NEXT J !224
1240 PRINT !156
1250 NEXT I !223
1260 PRINT : "INPUT THE VALUE
S ROW BY ROW FOR MATRIX B":
:!116
1270 FOR I=1 TO RA !209

```

(See Page 10)

REGENA ON BASIC—

(Continued from Page 9)

```

1280 FOR J=1 TO NB !207
1290 INPUT "B("&STR$(I)&","&
STR$(J)&") = ":BB(I,J)!025
1300 NEXT J !224
1310 PRINT !156
1320 NEXT I !223
1330 PRINT : : "[A][B] = [C]"
: : !174
1340 PR=0 !089
1350 FOR I=1 TO MA !204
1360 FOR J=1 TO NB !207
1370 T=0 !011
1380 FOR K=1 TO RA !211
1390 T=T+A(I,K)*BB(K,J)!024
1400 NEXT K !225
1410 PRINT "C("&STR$(I)&","&
STR$(J)&") = ";T !064
1420 GOSUB 890 !205
1430 NEXT J !224
1440 PRINT !156
1450 GOSUB 890 !205
1460 NEXT I !223
1470 RETURN !136
1480 PRINT "ENTER A SQUARE M
ATRIX." !108
1490 PRINT "THE COMPUTER WIL
L INVERT" !059
1500 PRINT "THE MATRIX.": : !
220
1510 PR=0 !089
1520 GOSUB 290 !115
1530 GOSUB 380 !205
1540 GOSUB 440 !009
1550 PRINT : : "--INVERTING--
": : !174
1560 GOSUB 530 !100
1570 IF D$<>"D" THEN 1600 !0
54
1580 PRINT "MATRIX IS NOT IN
VERTIBLE" !049
1590 RETURN !136
1600 FOR I=1 TO N !140
1610 FOR J=1 TO N !141
1620 PRINT "W("&STR$(I)&","&
STR$(J)&") = ";W(I,J)!233
1630 GOSUB 890 !205
1640 NEXT J !224
1650 PRINT !156
1660 GOSUB 890 !205
1670 NEXT I !223
1680 RETURN !136
1690 PRINT "FINDING THE DETE
RMINANT OF A SQUARE MATRIX.
": : !158
1700 DET=1 !149
1710 GOSUB 290 !115
1720 GOSUB 380 !205
1730 GOSUB 440 !009
1740 IF N>3 THEN 1810 !032
1750 IF N>2 THEN 1780 !000
1760 DET=A(1,1)*A(2,2)-A(1,2
)*A(2,1)!059
1770 GOTO 1970 !008
1780 DET=A(1,1)*A(2,2)*A(3,3
)+A(1,2)*A(2,3)*A(3,1)+A(1,3
)*A(2,1)*A(3,2)!190
1790 DET=DET-A(1,3)*A(2,2)*A
(3,1)-A(1,2)*A(2,1)*A(3,3)-A
(1,1)*A(2,3)*A(3,2)!095
1800 GOTO 1970 !008
1810 FOR C=1 TO N !134
1820 IF W(C,C)<>0 THEN 1850
!179
1830 GOSUB 740 !054
1840 IF D$="D" THEN 1980 !24
2
1850 DET=DET*W(C,C)!056
1860 FOR D=C+1 TO N !139
1870 W(C,D)=W(C,D)/W(C,C)!12
3
1880 NEXT D !218
1890 W(C,C)=1 !181
1900 FOR E=C+1 TO N !140
1910 FOR D=C+1 TO N !139
1920 W(E,D)=W(E,D)-W(E,C)*W(
C,D)!064
1930 NEXT D !218
1940 W(E,C)=0 !182
1950 NEXT E !219
1960 NEXT C !217
1970 PRINT : : "DETERMINANT =
";DET !036
1980 RETURN !136
1990 PRINT "SOLVING SIMULTAN
EOUS" !098
2000 PRINT "EQUATIONS BY THE
MATRIX" !228
2010 PRINT "INVERSION TECHNI
QUE" !249
2020 PRINT : : "SOLVE [A][X]
=[B]" !008
2030 PRINT : : "ENTER NUMBER
OF EQUATIONS OR" !159
2040 GOSUB 290 !115
2050 PRINT : : "THE COEFFICIE
NTS OF X" !130
2060 PRINT "ARE THE 'A' MATRI
X." !000
2070 GOSUB 380 !205
2080 FOR I=1 TO N !140
2090 FOR J=1 TO N !141
2100 INPUT "A("&STR$(I)&","&
STR$(J)&") = ":A(I,J)!213
2110 W(I,J)=A(I,J)!188
2120 NEXT J !224
2130 PRINT !156
2140 INPUT "B("&STR$(I)&") =
":B(I)!228
2150 PRINT : : !006
2160 NEXT I !223
2170 PRINT : : "-- SOLVING --
": : !090
2180 GOSUB 530 !100
2190 IF D$<>"D" THEN 2220 !1
64
2200 PRINT "NO UNIQUE SOLUTI
ON" !166
2210 RETURN !136
2220 PRINT : : "SOLUTION VECT
OR X:": : !107
2230 FOR I=1 TO N !140
2240 X(I)=0 !197
2250 FOR J=1 TO N !141
2260 X(I)=X(I)+W(I,J)*B(J)!
97
2270 NEXT J !224
2280 PRINT : : "X("&STR$(I)&")
= ";X(I)!206
2290 NEXT I !223
2300 PRINT : : !006
2310 RETURN !136
2320 END !139

```

Cecure in transition

Cecure Electronics is in temporary quarters, according to Don Walden of the company. During this transition period, he advises customers to telephone before sending items to the company or to send to the company's post office box.

He notes that articles sent by United Parcel Service are experiencing significant delays during rerouting.

Cecure's post office box is P.O. Box 132, Muskego, WI 53150. Telephone is (414) 679-4343 (voice), (414) 679-3636 (fax) or (414) 422-9669 (BBS).

Address change

The LA Computer Group, formerly the LA 99ers, has a new mailing address. The new address is LA Computer Group, 7022 Willowtree Dr., Rancho Palos Verdes, CA 90274-2961.

THE ART OF ASSEMBLY — PART 27

Another Potpourri

By **BRUCE HARRISON**
©1993 Harrison Software

We've revisited the Geneve subject on occasion in this series, and today here we go again. Some months back, we had a small crisis on our hands, with people who bought our Word Processor and then found out it wasn't compatible with their Geneve computers. We scrambled about a bit, trying to help our customers. Reworked the source code so that the DSRLNK would be taken from the TI E/A utilities, fixed up the loaders, and so on. We then shipped off a copy to our fellow columnist Stan Krajewski, whom we knew had a Geneve handy for testing.

Stan reported that everything seemed to work okay. Had we at last slain the dragon? It was another of those things that are too good to be true, and the untrue part came when we shipped copies to our two customers. On one of those two machines, nearly everything worked, except that the machine would lose the "index" file when saving a document to disk, so that afterwards the program wouldn't recognize that document as existing on the disk. The page files would show up on the disk, but not the vital index file. Why? We still don't know. This version worked perfectly on our TI. On the other Geneve, many of the important Function and Control key combinations wouldn't work, and it too failed to save the index files for documents.

Much later, we decided to ask Stan a few questions about his Geneve, to see if there was some rationale to this phenomenon. It turned out that Stan's Geneve was using a TI disk controller instead of a Myarc one. That could explain the business of the index files, but not the sensing and processing of keyboard inputs. Sorry, but we have no answers. Stan suggested that it may have had something to do with the EEPROMS, which in his case are the very latest versions. Maybe so, but the inconsistency from one Geneve to another makes it darned near impossible to be confident of making something compatible. We refunded the full purchase price to our customers, and gave up.

STRING MAGIC

Those who program in assembly get used to the idea that we can do pretty much anything with any of the computer's memory, and that's true so long as we need not return to some other program, like Extended BASIC. Your author, of course, did a lot of programming in Extended BASIC before starting to learn assembly. One of the frustrations in XB is the business of using string variables. As many of you already know, XB uses the VDP RAM to store string variables, and provides no other way to store them. You might write a program that takes only 2K of the high memory to load and run under XB, but if it uses lots of strings, XB is likely to run out of what it calls "Stack" space in VDP RAM, and thus your program will not be able to run.

Our good friend Jim Peterson complained to us about this, and we thought about it for a while. Later, one of our customers (Bill

(See Page 12)

SIDEBAR 27

```
* HISTRING/S
* AUX STRING STORAGE ROUTINE
* FOR HIGH MEMORY STRING STORAGE
* CODE BY BRUCE HARRISON
* FOR USE UNDER XB WITH ALSAVE
* 23 AUG 92
*
* EQUATES FOR XB UTILITIES, ETC.
VMBW EQU >2024      VDP MULTI-BYTE WRITE VECTOR
VSBW EQU >2020      VDP SINGLE-BYTE WRITE
VMBR EQU >202C      VDP MULTI-BYTE READ
STRASG EQU >2010    STRING VARIABLE ASSIGNMENT VECTOR
STRREF EQU >2014    STRING REFERENCE VECTOR
NUMREF EQU >200C    NUMERIC VARIABLE REFERENCE VECTOR
NUMASG EQU >2008    NUMERIC ASSIGNMENT VECTOR
XMLLNK EQU >2018    XML LINKAGE VECTOR
CFI EQU >12B8       CONVERT FLOATING POINT TO INTEGER
CIF EQU >20         CONVERT INTEGER TO FLOATING POINT
FAC EQU >834A       FLOATING POINT ACCUMULATOR
KEYVAL EQU >8375    KEY VALUE ADDRESS
KSCAN EQU >201C     KEYBOARD SCAN VECTOR
STATUS EQU >837C    GPL STATUS BYTE
SCRWID EQU 32       SCREEN WIDTH (32 CHARACTERS)
GPLWS EQU >83E0     GPL WORKSPACE ADDRESS
FIRLO EQU >2002     FIRST AVAILABLE LOW MEM ADDRESS
LASLO EQU >2004     START OF DEF TABLE IN LOW MEM
LASHI EQU >8386     HIGHEST AVAIL ADDRESS IN HIGH MEM
*
* END EQUATES, BEGIN SUBROUTINE CODE
*
DEF SETHI, PUTHI, GETHI DEFINE ENTRY POINTS
DEF AVHI
*
*
SETHI
    LWPI WS          LOAD OUR OWN WORKSPACE
    CLR R0           CLEAR R0 - NOT ARRAY
    LI R1,1          POINT TO FIRST PARAMETER
    BLWP @NUMREF     GET THAT NUMBER
    BLWP @XMLLNK     USE XML LINKAGE
    DATA CFI        TO CONVERT TO INTEGER
    MOV @FAC,@MAXNUM STASH AT MAXNUM LOCATION
    JGT SET0A        IF GREATER THAN ZERO, JUMP
    B @BADPRM        ELSE IF ZERO OR NEGATIVE, REPORT ERROR
SET0A MOV @FIRLO,R9 GET FIRST AVAIL ADDRESS LOW MEM
    ANDI R9,>FFFE    INSURE EVEN NUMBER
    INCT R9          POINT TO NEXT WORD
SET0 MOV R9,@STTBL  STASH THAT ADDRESS AWAY
    MOV @MAXNUM,R4   GET NUMBER IN R4
    SLA R4,1         DOUBLE FOR WORD COUNT
    LI R3,>A000      SET R3 TO >A000
OKAY0 MOV R3,*R9+   PLACE R3 NUMBER AT R9 ADDRESS
    C R9,@LASLO      COMPARE TO END OF LOW MEM
    JEQ LOWERR       IF EQUAL, JUMP
    JGT LOWERR       OR GREATER, JUMP
    INC R3           INC NUMBER IN R3
    DECT R4          DECREMENT COUNT BY TWO
    JNE OKAY0        IF NOT ZERO, OKAY
    MOV R9,@ENDTBL  PLACE R9 AT ADDRESS ENDTBL
    MOV @MAXNUM,R4   GET MAXNUM BACK
    MOV R4,R10       PLACE IN R10
    LI R9,>A000      LOAD R9 FIRST HI-MEM ADDRESS
    A R9,R10         ADD TO R10
    C R10,@LASHI     COMPARE TO HIGHEST HI ADDRESS AVAIL.
    JLT OKAY1        IF LESS, PROCEED
```

THE ART OF ASSEMBLY—

(Continued from Page 11)

Harms of Chino, California) wrote us a letter suggesting that something be done about this problem. We thought some more, then started working on source code. Both Bill Harms and Jim Peterson helped in the testing of this stuff, and both found cases where the resulting product could be used.

STASHING IN MEMORY

The idea was simple. If your XB program itself uses very little of the high memory, we create a space in the leftover memory to stash an array of strings. If your XB program is large, so there's little or no high memory available, we provide another utility to stash strings in the available low memory. The TI provides pointers we can use to figure out how much of high or low memory is available, so we don't wind up corrupting anything belonging to your XB program. In today's sidebar is the source code for the utility that stashes strings in high memory. There are four entry points defined in this source code, each of which uses parameters from a CALL LINK in XB. The high memory one shown has entry points SETHI, PUTHI, GETHI, and AVHI. There are lessons to be learned from this code, and we'll try to make this a "learning experience" for you, so don't think of this as a "commercial" for one of our products.

At SETHI, the assembly program does several things: First, it finds out from the CALL LINK parameter how many strings the XB program wants to make room for. Next, it makes a "lookup table" in low memory for the addresses of the strings it will place in high memory. Finally, it creates an array of null strings in the high memory space that's available. While it's doing this, it checks to see whether enough space is there. If you ask SETHI to make room for 6000 strings, it will run out of low memory space for the lookup table, and will report "NOT ENOUGH LOW MEMORY" on your screen. If this is the only utility being used by your XB program, there will be room for about 2900 strings' addresses in the lookup table.

Of course when you start actually assigning non-null strings into the array, more high memory will get used. As each PUTHI linkage is performed, the end point of the array is checked against the limit of available high memory, and if that's going to be exceeded, the error report "NOT ENOUGH HIGH MEMORY" will be issued on screen before that string is assigned. Just to give you the idea how powerful this thing is, we'll take an example from Jim Peterson. Jim used the routine with a multi-column printing program he'd written. Without the routine, this would run out of "stack" space before much of a multi-column page could be composed in a string array. With the routine in place, Jim's program could make 5 columns of 60 lines each at 28 characters per line (for a program listing) without running out of the high memory. With the length bytes, that's 8700 bytes in 300 strings.

The way strings are actually stashed in VDP RAM by XB is somewhat of a mystery. We did a little experiment that stored 300 strings of 28 bytes each in a string array variable under XB, and found that the amount of "stack" space reported by SIZE changed by 10,221 bytes, not 8700. Obviously there is some kind of lookup table space that's also kept in VDP RAM, and that ac-

(See Page 13)

B	@HIGHNG	ELSE ISSUE ERROR MESSAGE
LOWERR B	@LOWNG	ISSUE LOW MEMORY ERROR
OKAY1		
MOV B	R0,*R9+	MOVE A ZERO BYTE, INCREMENT POINTER
DEC	R4	DEC COUNTER
JNE	OKAY1	IF NOT ZERO, REPEAT
MOV	R9,@ENDLST	PLACE R9 AT LIMIT ADDRESS
B	@QEXIT	THEN EXIT ROUTINE
PUTHI		
LWPI	WS	LOAD OUR WORKSPACE
CLR	R0	CLEAR R0, NOT ARRAY
LI	R1,1	POINT FIRST PARAMETER
LI	R2,TEMSTR	POINT AT TEMPORARY STRING STORAGE
MOV B	@MAXLEN,*R2	SET FOR MAXIMUM LENGTH (255 BYTES)
BLWP	@STRREF	GET INCOMING STRING
MOV B	*R2,R3	GET LENGTH BYTE IN R3
SRL	R3,8	RIGHT JUSTIFY
INC	R1	POINT TO SECOND PARAMETER
BLWP	@NUMREF	GET THE NUMERIC VALUE
BLWP	@XMLLNK	USE XML LINKAGE
DATA	CFI	TO CONVERT NUMBER TO INTEGER
MOV	@FAC,R4	MOVE RESULTING INTEGER TO R4
JEQ	NOK4	IF ZERO, NOT VALID
C	R4,@MAXNUM	COMPARE TO MAXIMUM NUMBER SET UP
JLE	OKAY4	IF LOW OR EQUAL, OKAY
NOK4	B	@BADPRM
OKAY4	DEC	R4
SLA	R4,1	THEN DOUBLE TO INDEX BY WORDS
MOV	@STTBL,R5	GET START OF ADDRESS TABLE IN R5
A	R4,R5	R5 HAS TABLE ADDRESS
MOV	*R5+,R6	R6 HAS ADDRESS PRESENT STRING
MOV	R6,R12	STASH ADDRESS IN R12
MOV B	*R6+,R1	R6 POINTS TO CONTENT PRESENT STRING
SRL	R1,8	R1 HAS PRESENT STRING LENGTH
MOV	R3,R7	R7 HAS INCOMING STRING LENGTH
S	R1,R7	R7 HAS DIFFERENCE INCOMING-PRESENT
LENGTH		
JLT	OKAY5	IF A NEGATIVE NUMBER, JUMP
JEQ	NOMOVE	IF ZERO, JUMP
MOV	@ENDLST,R8	ELSE GET END OF STRING LIST IN R8
A	R7,R8	ADD LENGTH DIFFERENCE
C	R8,@LASHI	COMPARE TO MEMORY LIMIT
JLT	OKAY6	IF LESS, PROCEED
B	@HIGHNG	THEN ISSUE ERROR MESSAGE
OKAY5		
A	R1,R6	ADD LENGTH OF PRESENT STRING
MOV	@ENDLST,R4	GET END OF STRING LIST INTO R4
S	R6,R4	SUBTRACT ADDRESS
MOV	R6,R9	SET MOVE SOURCE
MOV	R9,R10	MOVE TO R10
A	R7,R10	ADD LENGTH DIFFERENCE
MOV B	*R9+,*R10+	MOVE ONE BYTE
DEC	R4	DECREMENT COUNT
JNE	MOVBT	IF NOT ZERO, REPEAT
JMP	OKAY7	ELSE JUMP AHEAD
OKAY6		
MOV	@ENDLST,R4	GET END ADDRESS IN R4
MOV	R4,R9	MOVE THAT TO R9
MOV	R9,R10	AND R10
S	R12,R4	SUBTRACT ADDRESS OF CURRENT STRING
A	R7,R10	ADD LENGTH DIFFERENCE
MOVREV	MOV B *R9,*R10	MOVE ONE BYTE
DEC	R9	DECREMENT SOURCE POINTER
DEC	R10	AND DESTINATION POINTER
DEC	R4	DECREMENT COUNT
JNE	MOVREV	IF NOT ZERO, REPEAT
OKAY7		
A	R7,*R5+	ADD LENGTH DIFFERENTIAL TO NEXT TABLE
ADDRESS		
C	R5,@ENDTBL	ARE WE FINISHED?
JLT	OKAY7	IF NOT, REPEAT
A	R7,@ENDLST	ADD LENGTH DIFFERENCE TO END OF STRING
LIST		

THE ART OF ASSEMBLY—

(Continued from Page 12)

counts for the "overhead" of 1521 bytes. That array of 300 28-byte strings would fit in VDP RAM if it were the only string variable used in the program, but it wouldn't take many more string variables to cause the XB program to crash with an "out of memory" error.

Among other things, XB reserves space for string variables in blocks that are larger than the actual string length. If you start in XB command mode with nothing and take SIZE, you find 11840 bytes of stack free. If you then enter a statement like X\$="", you'll find SIZE reports only 11830 bytes free. That null string has eaten up ten bytes. This no doubt makes things simpler for XB to deal with, but not for the programmer who's trying to cram lots of strings into VDP RAM.

In our own routine for storing in high memory, we keep the space occupied by the strings themselves limited to exactly the required amount, so that a string of zero characters occupies only one byte (the length byte), while a ten-byte string occupies exactly 11 bytes, and so on. Each time a string in the array is replaced, the space in memory beyond that one is "closed up" so the string put in place just fits in memory. By keeping the lookup table for addresses of the strings in low memory, we make all of the available high memory useful for strings in the array.

(See Page 14)

Repetitive Strain Injury

By TOM WILLS

(From the Southwest Ninety-Niners newsletter, Tucson, Arizona)

I'm sure everyone has heard of the Repetitive Strain Injury syndrome, better known as RSI. The most common injury is the problem known as Carpal Tunnel Syndrome (CTS), a wrist injury. Another problem is one of the elbow, known as either Tennis Elbow, if you are into sports, or Executive's Elbow.

Carpal Tunnel is by far the most common ailment. The elbow problem can best be described like hitting your "funny bone" and not having it go away. I am currently having problems with Executive's Elbow. (Doesn't that sound like an impressive type of injury?)

Not many people think Tiers can fall victim to RSI, but that is not the case. In fact, Tiers can be more prone to such an injury, especially CTS. The reason for this is in the very design of our faithful computer console. Most people rest the palms of their hands on the desk in front of the console and type from that position. This is bad for the wrists.

There are wrist rests for sale for as little as \$1.50. However, those wrist rests are designed for use with IBM-style keyboards similar to those used by Geneve owners. These wrist resents will not work with the TI99/4A console. I recommend getting two wrist rests and stacking them. This will relieve the pressure on the wrists and make typing easier.

As for the Executive's Elbow, its cause is different. How many are sitting in a chair with arms? Quite a few, I imagine. The problem lies in the fact that, while sitting in such a chair, a person naturally tends to put his elbows on the arms of the chair and lean on them. After enough time, the nerve, which we commonly refer to as the Funny Bone, gets inflamed and causes a tingly numbness in the hand similar to when you hit your "funny bone."

The best way to cure yourself of leaning on the arms of your chair is to get a chair *without* arms. This is especially important for persons who work in an office and sit all day at a desk.

There is no easy way to cure this problem. Sometimes wearing a splint or taking deflatory drugs will alleviate the problem. More often, however, the problem can only be helped by surgery. The surgery, especially for the elbow problem, leaves ugly scars. There is no guarantee that the surgery will clear up the problem. Nor is successful surgery any guarantee the CTS will not come back.

I am bringing this up in an effort to inform Tiers of a serious problem. Hopefully I can, through this article, inform someone who is on the verge of such a problem to avoid it.

```

NOMOVE
MOV R12,R10      GET R12 INTO R10
INC R3           INC TO INCLUDE LENGTH BYTE ITSELF
MOVIN MOVB *R2+,*R10+  MOVE ONE BYTE INTO STRING POSITION
DEC R3          DECREMENT COUNT
JNE MOVIN       IF NOT ZERO, REPEAT
JMP QEXIT      ELSE JUMP TO EXIT

AVHI
LWPI WS        LOAD OUR WORKSPACE
CLR R0         NOT ARRAY VARIABLE
MOV @LASHI,@FAC  GET LAST AVAILABLE HIGH MEM ADDRESS
INTO FAC
S @ENDLST,@FAC  SUBTRACT END OF ARRAY
BLWP @XMLLNK   USE XML LINKAGE
DATA CIF      TO CONVERT INTEGER TO FLOATING POINT
LI R1,1       FIRST LINK PARAMETER
BLWP @NUMASG   ASSIGN THE NUMBER
JMP QEXIT     THEN JUMP TO EXIT

GETHI
LWPI WS        LOAD OUR WORKSPACE
CLR R0         CLEAR R0, NO ARRAY
LI R1,2       POINT TO SECOND PARAMETER
BLWP @NUMREF   GET NUMBER
BLWP @XMLLNK   USE XML
DATA CFI      CONVERT TO INTEGER
MOV @FAC,R3   MOVE TO R3
JEQ NOK       IF ZERO, NOT VALID
C R3,@MAXNUM  COMPARE TO MAXIMUM NUMBER
JLE OKAY3    IF LOW OR EQUAL, OKAY

NOK
B @BADPRM     THEN ISSUE ERROR MESSAGE
OKAY3 DEC R3   ZERO-BASE THE NUMBER
SLA R3,1     THEN DOUBLE FOR WORD INDEXING
MOV @STTBLL,R2  GET START OF ADDRESS TABLE
A R3,R2     ADD INDEX NUMBER
MOV *R2,R2   POINT R2 AT STRING DESIRED
DEC R1      POINT TO FIRST PARAMETER
BLWP @STRASG  ASSIGN THE STRING

QEXIT
LWPI GPLWS    LOAD GPL WORKSPACE
B @>006A     RETURN TO GPL INTERPRETER
* END OF MAIN CODE SECTION
* FOLLOWING IS ERROR-HANDLING CODE
BADPRM
BL @CLR23    CLEAR ROW 23
LI R0,22*SCRWID+4 POINT AT ROW 23, COL 5
LI R9,00RSTR  MESSAGE ADDRESS IN R9
BL @DISLI    DISPLAY MESSAGE
JMP ERREX   JUMP TO ERROR EXIT

LOWNG
BL @CLR23    CLEAR ROW 23
LI R0,22*SCRWID+4 ROW 23, COL 5
LI R9,LMFSTR  MESSAGE ADDRESS
BL @DISLI    DISPLAY THAT
JMP ERREX   THEN JUMP

HIGHNG
BL @CLR23    CLEAR 23
LI R0,22*SCRWID+4 ROW 23, COL 5
LI R9,HMFSTR  MESSAGE
BL @DISLI    SHOW THAT

ERREX
BL @CLR24    CLEAR ROW 24 OF SCREEN
LI R0,23*SCRWID+3 POINT AT ROW 24, COLUMN 4
LI R9,PAKSTR  PRESS ANY KEY MESSAGE
BL @DISLI    DISPLAY THAT
ERRLOO CLR @STATUS  CLEAR STATUS
BLWP @KSCAN  SCAN KEYBOARD
CB @ANYKEY,@STATUS  KEYSTRUCK?
JNE ERRLOO  IF NOT, SCAN AGAIN
LI R0,22*SCRWID  ROW 23, COL 1
LI R1,BOTTOM  SAVED BOTTOM OF SCREEN
LI R2,2*SCRWID  TWO ROWS WORTH

```

THE ART OF ASSEMBLY—

(Continued from Page 13)

Bill, after testing an early version, suggested that the XB programmer should be able to find out how much string storage space our routine still had available while his XB program was running. That seemed like a worthwhile idea, so we added the CALL LINK for AVHI, which can be used at any time either from Command mode or while a program is running. It reports the memory still available into a numeric variable named in the CALL LINK. We mention this mainly to illustrate a point, that having a “third party” try out your program at an early stage is a good idea. In this case it meant a lot of disks shipped from coast to coast, and a number of interesting phone calls, but made a much improved product possible.

At one point, we thought we had the project finished, only to find that one of our error traps didn't work as we thought it should. Bill tried something we'd never thought to try, and sure enough it crashed his TI. That's another advantage to having somebody else try things out. The guy who wrote the code never thinks of doing things that have the potential for disaster. Thanks to Bill, that problem was detected and then solved.

Bill made another important contribution, when he pointed out that our “internal” error traps in the assembly routine would exit to XB as if no error had occurred, thus allowing XB to perhaps keep executing a loop that was needless. What Bill wanted was some way that XB could recognize that an error had occurred in the CALL LINK statement. After some hard thinking, we came up with one of those simple but elegant solutions to this problem. None of the calls to the routine uses more than two parameters. We thus were able to add a BLWP @NUMASG to the error trap's exit, and by setting R1 for a non-existent third parameter, made XB recognize and report an error. Given this, the XB programmer can build in an ON ERROR situation in his XB program, or can simply let the XB program stop when the error happens. That's not all, though. By doing the error report to XB through a third parameter,

we allowed the XB programmer to have XB ignore the error by the simple expedient of putting a third parameter as a numeric variable into the CALL LINK.

It works like this. Suppose XB is getting a string from the array into X\$: CALL LINK(“GETHI”,X\$,Y) will take the Yth string from the array in high memory into X\$. If Y is too big a number, or zero, or there's some other error detected by the Assembly error trap, XB will also report a “BAD ARGUMENT IN ZZZ” when it regains control. To make XB ignore the error, we'd CALL LINK(“GETHI”,X\$,Y,X). Here that third parameter X will get a nonsense value assigned to it, but XB will not recognize any error. Thus the XB programmer can have it both ways, either having XB recognize the error or ignore it. This was one of those “serendipity” results which sometimes happen in programming work, where a simple solution to a problem becomes “elegant programming.” We didn't originally plan it that way, but it's a rather nice outcome anyway.

We point out all these things mainly to emphasize the need for an “independent” test agent. For some of our software projects, my partner Dolores serves in that role. For our Word Processor,

```

BLWP @VMBW      WRITE OLD STUFF BACK
CLR R0          NOT ARRAY VARIABLE
LI R1,3         THIRD PARAMETER
BLWP @NUMASG    ASSIGN WHATEVER IS AT FAC
JMP QEXIT      THEN EXIT

* END OF ERROR-HANDLING CODE
*
* FOLLOWING SECTION ARE SUBROUTINES
CLR23
  LI R0,22*SCRWID ROW 23, COL 1
  LI R1,BOTTOM  DATA SAVING LOCATION
  LI R2,2*SCRWID TWO ROWS WORTH
  BLWP @VMBR     GET CURRENT CONTENT
  LI R9,BLNKLN  32 SPACES
  JMP DISLI     THEN JUMP
CLR24 LI R0,23*SCRWID POINT AT ROW 24, COLUMN 1
  LI R9,BLNKLN  32 SPACES
DISLI LI R10,SCRLI POINT AT CHARACTER BUFFER
  MOV R10,R1    MAKE R1=R10
  MOVB *R9+,R4  GET STRING LENGTH INTO R4
  SRL R4,8      RIGHT JUSTIFY
  MOV R4,R2     PLACE VALUE IN R2 ALSO
DIS1  MOVB *R9+,*R10 MOVE ONE BYTE OF STRING TO BUFFER
  AB @OFFSET,*R10+ ADD >60 OFFSET
  DEC R4        DECREMENT LENGTH COUNT
  JNE DIS1     IF NOT ZERO, REPEAT
  BLWP @VMBW   ELSE USE VMBW TO DISPLAY STRING
DISLIX RT      RETURN
*
* END OF SUBROUTINE CODE
*
* FOLLOWING IS REQUIRED DATA FOR SUBROUTINE
*
WS BSS 32      OUR OWN WORKSPACE
BOTTOM
TEMSTR BSS 256 STORAGE FOR INCOMING STRING
ENDLST DATA >A000 END OF STRING LIST
STTBL DATA 0  START OF TABLE
ENDTBL DATA 0  END OF TABLE
MAXNUM DATA 0  MAX NUMBER OF STRINGS FROM SETUP
SCRLI BSS SCRWID SCREEN LINE CHARACTER BUFFER
MAXLEN BYTE 255 MAX POSSIBLE STRING LENGTH
ANYKEY BYTE >20 HEX VALUE 20
OFFSET BYTE >60 XB CHARACTER OFFSET
LMFSTR BYTE 21
  TEXT 'NOT ENOUGH LOW MEMORY'
HMFSTR BYTE 22
  TEXT 'NOT ENOUGH HIGH MEMORY'
OORSTR BYTE 22
  TEXT 'PARAMETER OUT OF RANGE'
PAKSTR BYTE 25
  TEXT 'PRESS ANY KEY TO CONTINUE'
BLNKLN BYTE SCRWID
  TEXT
  END

```

she served as “guinea pig” almost from its inception, and could always find the bugs in each new version. In many cases, the bugs would never have been found by the programmer. When a person tests his own software, there's a small voice in the back of his head saying “don't try pressing Function 9 when you're at this point”. Something in his intimate knowledge of the source code makes him avoid that keystroke which will crash the software. The “other” user will of course try anything and everything that the software should be able to handle, and will thus find the bugs every time. Some years back, your author was purchasing a rather large and complex system for the Navy. Part of the hardware was

(See Page 15)

Geneve 9640

Sprites with MY-Color

By JIM UZZELL
©1993 DDI Software

The following program illustrates the power of drawing commands to enhance a program that uses sprites. Even though this demo uses only two of the drawing commands, you will see sprites with color, a sprite disappear temporarily, multi-colored sprites, sprites that appear to glitter and change patterns. Probably the most significant part of the demo is the illustration of a multi-colored mouse sprite, a command not documented in the MY-BASIC manual.

There are some do-nothing loops in the program to slow it down so you can see the color changes happen. These loops were designed for MDOS 1.23F. If you are using HDOS 1.50, or you are using Turbo with MDOS 1.23F from 9640 News Vol. 3 No. 1, or you are using MY-FASTVID (available from the 9640 News BBS) with MDOS 1.23F, this demo program will run

faster than designed. You could use some of the routines from the XOP6 series of articles (April-June 1992, MICROpendium) instead of MY-BASIC commands and have the same results.

```
1 !DDI SOFTWARE
2 !COPYRIGHT 1993
100 DATA FEFDFAF5EAD5AA5555A
AD5EAF5FAFD7E7FBF5FAF57AB55A
AAA55AB57AF5FBF7F
110 DATA FFFFFFFFFFFFFFFFFF186
B6B195B6B68FFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFF
120 DATA FFFFFFFFFFFFFFFFFFCEB
5BDCDF5B5CEFFFFFFFFFFFFFFFFF
F30D7D7D1D7D737FF
130 DATA 8EB6B6B6B6B68EFF8BD
BDBDADADADDF38DDDDDDDDDD38F
FB3ADADA1ADAD6DF
140 DATA FFD5AAD5AAD5AAD5AAD
5AAD5AAD5AAF55AB55AB55AB5
5AB55AB55AB55ABFF
150 DATA FFC4A992AFC899AAAA9
```

```
9C8AF92A9C4FFFF239549F513995
5559913F5499523FF
160 CALL GRAPHICS(3,3) :: CA
LL ECOLOR(6) :: RESTORE
170 CALL TCOLOR(4,14) :: CLS
180 READ C$,C1$,A1$,B1$,F$,A
$
190 CALL CHAR(100,C$,104,C$,
108,C$,112,A1$,116,B1$,120,C
1$,124,F$,128,A$)
200 CALL MAGNIFY(4)
210 CALL SPRITE(#1,100,2,64,
161)
220 CALL SPRITE(#2,104,2,64,
225)
230 CALL SPRITE(#3,108,2,64,
289)
240 CALL SPRITE(#7,100,2,96,
161)
250 CALL SPRITE(#8,124,16,96
,225)
```

(See Page 16)

THE ART OF ASSEMBLY—

(Continued from Page 14)

delivered before the software was fully developed, and the customer quickly learned how to crash the program with a single keystroke. This went on for some time, until finally the programmers got the software to be “robust” enough that sitting on the keyboard wouldn’t cause a problem. We knew the project was really finished when the only complaint the customer could come up with was that the little bulbs for the backlit function keys burned out too frequently. Then your author signed the final delivery form so the contractor could be paid his \$5 million.

FINISHING TOUCHES

There’s a tendency sometimes in our work on the TI to rush a project, especially when a deadline is nearing. This happens to us when we are trying to get a new product ready for either the Lima Faire or the Chicago Faire. There’s an urgency that may make us forget some things like proper error traps, or even some of the “nice to have” features that should be added to a program even though they’re not essential to the main function. Many times this also means skipping the step of having someone else try out the software.

Perhaps the most embarrassing example of this happened when we first introduced our Smart Connect disk. We rushed so much to

have that ready for Chicago that we had only tested it with one of our three PC computers. Upon return from Chicago, we found customer complaints waiting for us. Sure enough, what we’d rushed to get ready would only work on that one of our three PCs. After some rather hectic re-working of the PC programs, we were able to get the program fixed, but this became a costly error, since we had to send revised disks to all those customers who’d bought the program at Chicago. Since then, we’ve had very few calls from customers who purchased that package, and in those cases we were able to resolve the problems over the phone.

It’s late September of 1992 as we write this, and this year we’ve resigned ourselves to offering at Chicago only those products that have been thoroughly tested. There are about three or four projects in various stages of incompleteness right now, but none of them is going to be “rushed” for Chicago. Maybe one or two will be ready for Lima in 1993, but we’re making no commitments, even to ourselves.

Last month we promised a “single topic” column, and obviously we’ve missed the mark here, so we are making no promises for next month, except to try to make our column interesting for all our readers, and illuminating for those trying to make programs or routines in assembly language.

SPRITES—

(Continued from Page 15)

```

260 CALL SPRITE(#9,100,2,96,
289)
270 CALL SPRITE(#10,100,2,12
8,161)
280 CALL SPRITE(#11,100,2,12
8,225)
290 CALL SPRITE(#12,100,2,12
8,289)
300 CALL DCOLOR(4,14) :: CAL
L DRAW(1,1,161,1,352)
310 CALL SPRITE(#4,112,4,2,1
61)
320 CALL SPRITE(#5,116,4,2,2
25)
330 CALL SPRITE(#6,120,4,2,2
89)
340 DISPLAY AT(7,32):"COLORI
NG SPRITES"
350 FOR W=1 TO 1000 :: NEXT
W
360 Z=63 :: Z1=161 :: Z2=224
370 FOR X=1 TO 11
380 CALL DCOLOR(X,14)
390 GOSUB 440 :: FOR Y=1 TO
32
400 CALL DRAW(1,Z+Y,Z1,Z+Y,Z
2)
410 NEXT Y :: FOR W=1 TO 100
0
420 NEXT W :: NEXT X
430 GOTO 530
440 IF X=4 THEN Z1=225 :: Z2
=288 :: RETURN
450 IF X=5 THEN Z1=289 :: Z2
=352 :: RETURN
460 IF X=6 THEN Z=95 :: RETU
RN
470 IF X=7 THEN Z=127 :: RET
URN
480 IF X=8 THEN Z1=225 :: Z2
=288 :: RETURN
490 IF X=9 THEN Z1=161 :: Z2
=224 :: RETURN
500 IF X=10 THEN Z=95 :: RET
URN
510 IF X=11 THEN Z1=225 :: Z
2=288
520 RETURN
530 FOR W=1 TO 1000 :: NEXT
W
540 FOR X=2 TO 16
550 CALL DCOLOR(X,6)
560 CALL RECTANGLE(4,95+X,22
4+X,129-X,288-X,129-X,288-X)
570 FOR W=1 TO 500 :: NEXT W
:: NEXT X
580 CALL DCOLOR(9,6) :: CALL
RECTANGLE(4,68,169,92,216,9
2,216)
590 FOR W=1 TO 500 :: NEXT W
600 CALL DCOLOR(8,6)
610 CALL RECTANGLE(4,74,177,
78,208,86,177)
620 FOR W=1 TO 500 :: NEXT W
630 CALL DCOLOR(16,6)
640 CALL RECTANGLE(4,78,185,
72,200,82,185)
650 FOR W=1 TO 500 :: NEXT W
670 MSG$=MSG$&".....INCLUDIN
G MULTI-COLORED SPRITES"
680 MSG$=RPT$(" ",80)&MSG$
690 CALL TCOLOR(4,14) :: LT1
=1+(LEN(MSG$))
700 FOR LT=1 TO LT1
710 DISPLAY AT(21,1):SEG$(MS
G$,LT,80) :: CALL SOUND(20,5
23,0,220,10)
720 IF LT=25 THEN CALL SPRIT
E(#8,124,2,96,225,0,0) :: GO
TO 750
730 IF LT=125 THEN CALL MOTI
ON(#8,10,0) :: CALL SPRITE(#
13,128,16,96,225,0,0)
740 IF LT=LT1-5 THEN CALL SP
RITE(#13,128,16,96,225,0,-10
)
750 NEXT LT
760 CALL TCOLOR(16,6) :: GOS
UB 810
770 DISPLAY AT(21,30):"PRESS
ANY KEY TO EXIT";
780 CALL KEY(0,K,S)
790 IF S<1 THEN 780
800 CALL CHARSET :: END
810 CALL MOTION(#1,3,-10,#3,
10,6)
820 CALL MOTION(#7,-3,10,#9,
-10,-6)
830 CALL MOTION(#10,-10,-6,#
12,10,3)
840 FOR W=1 TO 4000 :: NEXT
W
850 RESTORE 150 :: READ Z$:
: CALL CHAR(252,Z$)
860 CALL DCOLOR(2,16) :: CAL
L RECTANGLE(4,128,60,160,122
,160,122)
870 CALL SPRITE2(#1,252,3,16
,4,5,7,6,8,9,13,11,2,7,3,13,
15,16,128,60,0,0)
880 DISPLAY AT(21,7):"MOUSE
SPRITE";
890 FOR W=1 TO 2500 :: NEXT
W :: CALL MOTION(#1,-3,0) ::
RETURN

```

BUGS & BYTES

SCSI project update

The following item has appeared on a number of electronic services and BBSes. It was written by Jeff White and others. A portion of it consisted of technical materials which have been deleted.—Ed.

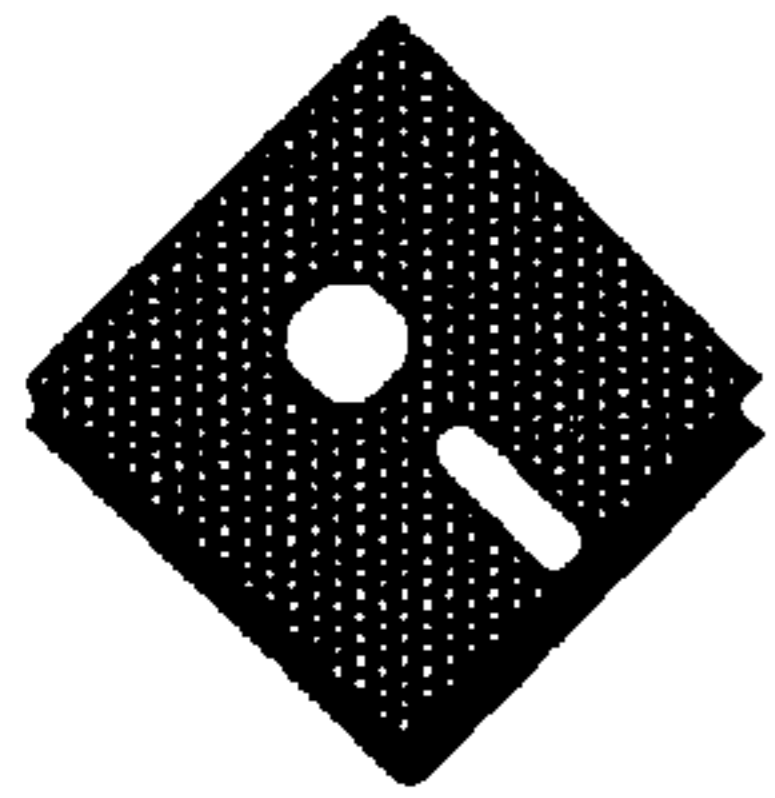
Some people have been concerned about completion of the SCSI DSR. Many of these bought the SCSI host adapter when it was sold at the 1992 Chicago faire.

I will not delve deeply into the history of why the SCSI DSR has been delayed. This article is not an announcement of its completion or impending release. The purpose of this text is to

provide additional insight into the thought processes that have gone into designing the SCSI DSR such that it will be HFDCC-compatible but support greater functionality and flexibility.

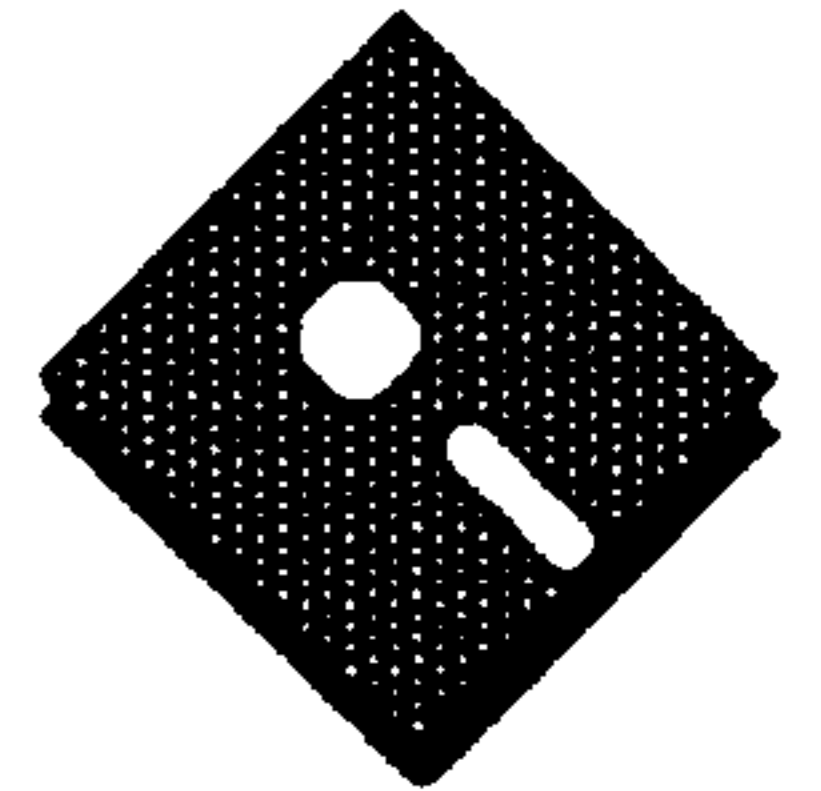
Will McGovern, part of the father-son team that brought us Funnelweb and other programs, wrote the low-level code for SCSI. Will had programmed several SCSI device drivers for Amiga SCSI host adapters. His knowledge of both the 99/4A and the SCSI controller chip on the Western Horizon Technologies SCSI was put to good use.

During Will's development of the low-level code, some deci-
(See Page 18)



MICROpendium

DISK SALE



If you've been waiting for a sale on MICROpendium program disks, this is it! For a limited time (through Nov. 15, 1993) Series 1-6 disks are available for a special price. (Series 6 disks are mailed

monthly starting with the October 1993 edition, programs from April 1993 through October 1993 will be mailed as soon as the order is placed.)

MICROpendium disks

SERIES #	REGULAR PRICE	SALE PRICE	YOU SAVE	DISCOUNT
Series 1 (Apr. '88-Mar. '89)	\$25.00	\$15.00	\$10.00	40%
Series 2 (Apr. '89-Mar. '90)	\$25.00	\$15.00	\$10.00	40%
Series 3 (Apr. '90-Mar. '91)	\$25.00	\$15.00	\$10.00	40%
Series 4 (Apr. '91-Mar. '92)	\$25.00	\$15.00	\$10.00	40%
Series 5 (Apr. '92-Mar. '93)	\$25.00	\$15.00	\$10.00	40%
Series 6 (Apr. '93-Mar. '94)	\$40.00	\$25.00	\$15.00	38%

But that's not all, if you order more than one series, you get an additional discount.

Series 1-5	\$125.00	\$62.50 for all four	\$62.50	50%
Any 4 from Series 1-5	\$100.00	\$50.00 for three	\$50.00	50%
Any 3 from Series 1-5	\$75.00	\$37.50 for two	\$37.50	50%
Any 2 from Series 1-5	\$50.00	\$25.00 for two	\$25.00	50%
Any 1 from Series 1-5 + Series 6	\$65.00	\$32.50 for two	\$32.50	50%
Any 2 from Series 1-5 + Series 6	\$90.00	\$45.00 for three	\$45.00	50%
Any 3 from Series 1-5 + Series 6	\$115.00	\$57.50 for four	\$57.50	50%
Any 4 from Series 1-5 + Series 6	\$140.00	\$70.00 for four	\$70.00	50%
Series 1-5 + Series 6	\$165.00	\$82.50 for all five	\$82.50	50%

(Circle the items you want to order)

ACT NOW — THIS SALE ENDS NOV. 15, 1993

Customer information

Name _____

Address _____

City _____

ST _____ ZIP _____

Please circle the items above and return this entire page (or a copy of it) with a check or money order in payment.

TOTAL REMITTED _____

(U.S. Funds only)

Disk shipping information

Postage is included for any disk sales to U.S. addresses. **Canadian delivery:** add \$2.00 for each series of disks for airmail delivery, \$1.50 for surface. **Overseas delivery:** add \$3.50 for each series of disks for airmail delivery; add \$2.00 for each series for surface

FOR CREDIT CARD ORDERS

Credit Card No. _____

Credit Card: MC Visa Exp. Date _____
(Circle One)

Signature _____

(credit card orders only)

BUGS & BYTES

(Continued from Page 16)

sions he and I made were based on performance issues with the desire to simplify coding the DSR. Then something we discussed resulted in the development of the new DMA GAL which should theoretically double throughput. The DMA GAL had to be replaced to fix a problem with its HOLD-logic used in Geneve systems. The 99/4A does not sense the HOLD pin on the peripheral bus.

Adding the new DMA GAL, cutting a trace, and adding a few jumper wires is what was necessary to double throughput in some instances. The card works as it was designed, and Will did all his development with the original DMA GAL and no jumpers or cut traces on the board.

The new DMA GAL is an improvement — not a fix — of the original design. I tested the original SCSI low-level code from Will on both the Geneve and 99/4A with no trouble. The low-level code has been modified and will be modified further to use the enhancements of the DMA GAL. This means all SCSI cards must be modified even though they would work fine out of the box. Keeping track of the few users that may still have unmodified cards to provide an alternative DSR would be a headache. If you have a SCSI card, when you get the EPROM for it you will get the new DMA GAL if you did not get it earlier.

The SCSI host adaptor was designed by Don O'Neil of Western Horizon Technologies with quite a bit of input from me. Most of the features were outlined by Don with my input. Will McGovern wrote the low-level code based on the low-level software interface of the MYARC HFDCC. Will and I decided on additions to the HFDCC low-level code specifications.

Beery Miller of 9640 News may be able to make SCSI compatible with the Geneve. Geneve compatibility has always been a concern, and every effort will be made to make adapting the SCSI DSR for the 99/4A to the Geneve easy for Beery.

Mike Maksimik will be writing software for SCSI. What this software will be is not definitive, but something similar to the tape backup software, disk management software, etc. that he is writing for use with the HFDCC is possible. CD-ROM software perhaps.

Timothy Tesch and Brad Snyder have been added to the

SCSI development team to expedite completion of the DSR. Every person on the SCSI software development team is capable of writing the DSR, but the time it would take any one person to do it would lengthen the delay. The SCSI DSR is a very ambitious project that the hardware developers did not fully comprehend at the outset.

Tesch, Snyder and I will be working on the SCSI DSR for the 99/4A with the sincere hope that a preliminary version can be released or demonstrated at the Chicago/Milwaukee faire weekend this year.

Because we do not want to alienate third-party programmers who might want to support SCSI, we think it best to get their input before the DSR specifications are finalized. There is no need to splinter the TI community into camps supporting either the Myarc HFDCC or the WHT SCSI. If someone wrote software that supports HFDCC functions, it is our intention that SCSI be compatible.

Tler on top

TI99/4A user Jonathan Guidry of New Iberia, Louisiana, a member (by long distance) of the Dallas TI Home Computer Group, recently placed first in the state of Louisiana in the National Computer Concepts Contest. In the finals in Washington, D.C., he made a 95. Guidry is now a university freshman. Congratulations!

Merger Down Under?

In the August issue of Bug Bytes, published by the Texas Instruments Brisbane User Group, Garry Christensen suggests the option of a single Australian user group (with the present groups as subgroups) with a single newsletter. As another possibility he suggests leaving the groups as they are with a central newsletter.

He suggests this, he says, because local memberships are declining and newsletters are getting smaller. (Bug Bytes, however, is on the large side compared to most U.S. TI groups' newsletters.)

Hardware project

Modification for CorComp RS232 cards

By JOHN F. WILLFORTH

The following article appeared in several user group newsletters. All hardware projects are done at the risk of the reader.—Ed.

Some time ago Frank Foster came to me with an interesting problem. He had purchased a Fastext 80 printer, a special

strobe inverting cable, and in putting these together with his CorComp RS232/PIO card, he sat in disbelief when nothing happened. He tried a regular cable that had been used on an Epson printer, and still no printing occurred. The printer would print a self-test, however.

Since I use a CorComp card, I asked

Frank to bring the printer, card and cables over so we could have a look see.

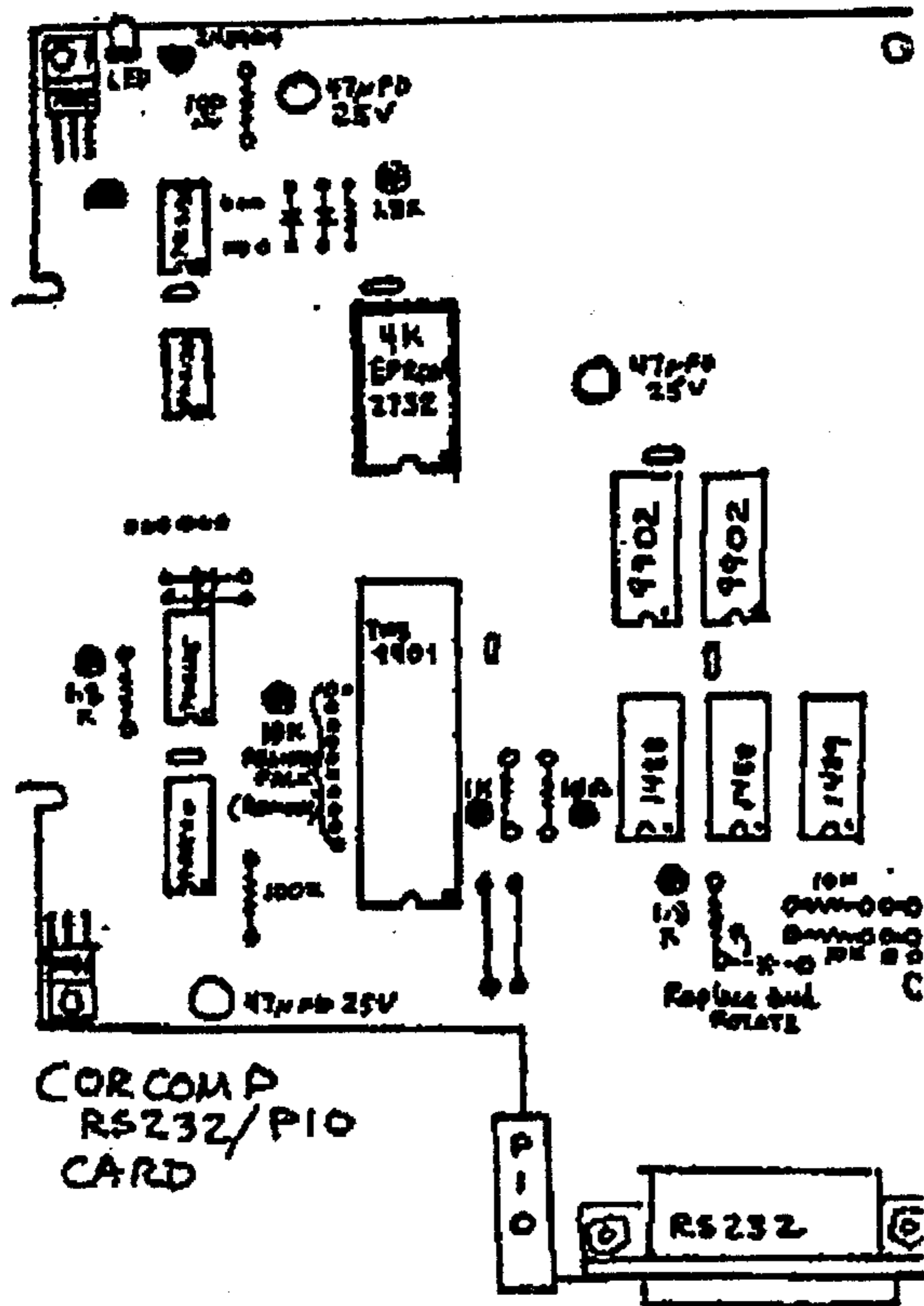
Frank was correct about his cable-card-printer combination. It didn't work on my system either. Since I had his PIO controller in the PEB, I hooked up my print (an Epson) to his card. It worked. It

(See Page 19)

HARDWARE PROJECT —

(Continued from Page 18)

seemed that with the Epson that all the cables worked, and both his and my CorComp cards drove the printers. This was strange.



I had heard that CorComp RS232 cards had undergone some improvements but I couldn't find the information to verify what the improvements were. Well, Frank left and I said I would call CorComp and investigate the problem. Well, I got busy with other things and about two weeks later Frank called. There I was, no further along than when he had dropped off his equipment.

Now for the desperate part. I removed both cards from their shells and noted that the artwork (physical board layout) was the same. However, his card had physical jumpers for multiple board strapped, which mine did not. My board was newer and, therefore, had reaped the benefits of corporate cost cutting that is the plague of all good things. So, looking further and having verified that the ROM information was the same in the two cards, I started checking chip types and vendor types. There were some difference and, if nothing further would show up, I would resort to changing the few of them that were different. I had

heard of a problem in this area.

On closer investigation, I found an astounding difference: the resistors were so different in value that it is hard for me to believe that the card could have worked.

I drew a sketch of the card and on it placed the main components for reference. Five resistors and a resistor pack may be removed, with no apparent problems found, if they are replaced (in the case of the resistors) with the values indicated on the drawing. You will note that the resistor just above the RS232 connection is notated to solder into a hole that exists in a trace already. The resistor pack is a 10K pack, as are the five discrete resistors that I removed. I placed a circle-like symbol at the location where you may have to remove or replace a component.

This may fix problems that you have had with this controller. I've been using mine since early 1984 and have never had trouble with it on any type printer or cable. This is not to say that there have not been some changes to the card. I hope this will help someone who has had strange and hard to define problems with this card. You may need the following resistors for this project: (5) 1.8K 1/4 W, 1 1K 1/4 W, and 1 10 ohm 1/4 W.

1993 TI FAIRS

APRIL

Northeast TI Fair, April 17, Waltham High School, Waltham, Massachusetts. Contact Ron Williams, 14 East St., Avon, MA 02322.

Canadian TI Fest, April 24, Merivale High School, Nepean, Ontario, Canada. Contact Bill Gard, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6 or (613) 523-9396 or Fax (819) 997-2194 Attn: DMES 2.

MAY

Lima Multi User Group Conference, May 14-15, Ohio State University Lima Campus, Lima, Ohio. Contact Dave Szimpl, 4191 Patterson Haplin, Sidney, OH 45365; phone (513) 498-9713 (evenings).

Fourth Annual TI Orphans Reunion, May 15, Zurich Insurance Claims Centre, 9715 Ottewell Rd., Edmonton, Alberta, Canada. Contact Ron Hohman, (403) 456-0862.

SEPTEMBER

Western Washington TI Fair, Sept 18, Tacoma Waterworks, 3506 S. 35th, Tacoma, Washington. Jim Tompkins, (206) 756-0934.

OCTOBER

Annual International TI-Faire, Oct. 8-10, Evangelisches Fe-

rienwaldheim Weidachtal, 7000 Stuttgart 80 (Mörhingen), Weidach Gewann 8, Germany. Contact Hans Huben, Berberitzenweg 6, 7033 Herrenberg, Germany; Wolfgang Bertesch, Helenenburgweg 61, 7120 Bietigheim-Biss, German; or Dierk Warburg, Lilienweg 12, 7141 Benningen, Germany.

Chicago International World Faire, Oct. 30, Holiday Inn, Gurnee, Illinois. Contact Cecure Electronics, P.O. Box 132, Muskego, WI 53150, or Don Walden, (414) 679-4343.

Milwaukee TI Fair, Oct. 31, Quality Inn, 5311 Howell Ave, Milwaukee, Wisconsin. Contact Gene Hitz, 4121 North Glenway, Wauwatosa, WI 53222.

1994 FAIRS

FEBRUARY

Fest-West, Feb. 19-20, Santa Rita Park Inn, Tucson, Arizona. Contact Tom Wills, Fest-West '94 Committee, Southwest 99ers Users Group, P.O. Box 17831, Tucson, AZ 85731 or (602) 886-2460; BJ Mathis, (602) 747-5046; or the Cactus Patch BBS, (602) 290-6277.

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

SYSTEM OF THE MONTH

A system with personality

Our system of the month belongs to Bob Holton of Nashville, Tennessee, a T1er since 1988. He writes:

I bought my first TI console to catalog my videotapes. I had 60

tapes with three movies on each one. It was a pain to keep up with the titles on the legal pads as I copied and erased movies. I taught myself to write a simple program that would scroll five tapes with 15 titles at a time and pause long enough to read them. I learned to save my listings on a tape recorder. When I got my printer, things became much easier for me. As I became more interested in the computer I contacted several

suppliers that TI recommended. I had a terrible time trying to deal with them. The people who took the orders barely knew as much as I about the TI. I wanted a stand-alone drive which I had seen pictures of, but they told me that they either didn't exist or were not sole anymore. They only wanted to sell me an expensive expansion system which I was not ready for. I finally came in contact with Jim Leshar of Dallas, Texas, who talked to me like a human being and listened to my needs. Jim fixed me up with an external drive and controller, and later another drive. Jim taught and explained to me how the TI, its hardware and some of the programs worked during many long phone calls. It is people like Jim Leshar, along with Don Walden and Beery Miller, who make the TI world special. And here locally, Bob Teague, the president of the Music City 99ers, has spent many hours helping me with problems. Their unselfish dedication deserves recognition.

Jim also is responsible for me becoming interested enough to get a modem, the HFDC card and the Geneve — thanks, Jim!

My present system consists of the Geneve 9640 card, HFDC card, Magnavox RGB monitor, Avatex 1200 baud modem, Myarc mouse, RS232 card, Wico Ergostick, Seagate 10 meg hard drive, Miniscribe 20 meg hard drive, black and silver console, Speech Synthesizer, Epson 24-pin printer, Fujitsu 5.25 DSDD floppy drive, Teac 1.44 meg 3.5 floppy drive and PE box. I have installed an IBM P/S2 switching power supply in the PE box and removed the inner compartments of the box. The miniscribe hard drive is mounted inside the PE box with the 720K and 1.44 drives. The HFDC card has 32K memory update for QIC 40 or 80 tape backup. The Geneve card has 32K 0-wait memory, reset switch and boot EPROM BIOS-MDOS V.1.01-1.50H-H1. The ROM BIOS boots directly off the chip or you can choose to boot off the HD, floppy or RAMdisk. The cards were updated by Don Walden of

Cecure Electronics and I am really enjoying the improved performance.

Contrary to many reports in MICROpendium, changing power

supplies does not require modification of any cards in the PE box. I was going to remove or jumper the regulators but Don suggested that I connect the 12-volt output to the 6-volt bus in the PE box and everything is working fine. I put the 10 meg drive in the external power supply case and use it to back up the 20 meg drive. I added a Radio Shack 120 volt fan to the back of the box, blowing in on the cards, and after four hours

of operation the temperature was only 105F at the warmest place tested. The power supply has a fan inside which blows out the back and between the two fans there is plenty of ventilation.

I added a swivel shelf to a cabinet for the monitor and full extension glides to a drawer for the keyboard and mouse that slides across my lap while sitting up in bed. My printer is also on a pull-out drawer. All components are connected to two toggle switches so that everything can be switched on at once or I can elect to leave the modem and printer off if I don't plan to use them.

If anyone has any questions or comments they can reach me at (615) 781-6461 or write to: Bob Holton, 304 Elberta St., Nashville, TN 37210.

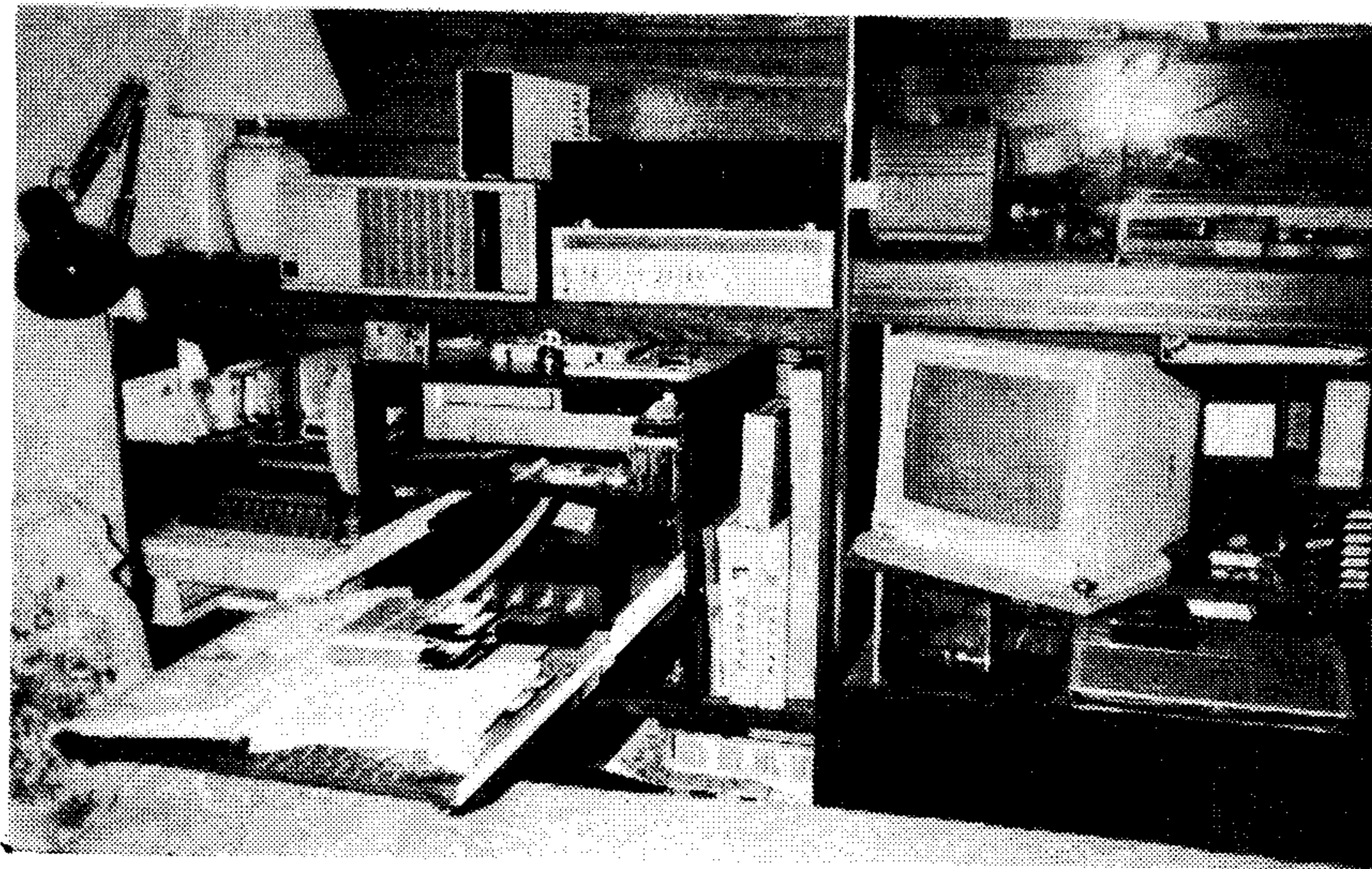
I would like to add that MICROpendium has played a very valuable role in my computer education by providing informative articles and contacts. Keep up the good work! A lot of us would likely have gone down the tubes if it hadn't been for your magazine.

Chicago TI fairgoers can get tourist information

Tourist information for visitors to the Chicago International World Faire Oct. 30 in Gurnee, Illinois is available from the Lake County tourist office.

The Lake County Visitors' Guide contains information on sightseeing as well as a guide to local hotels and restaurants, including prices and discount coupons.

For information, call the Lake County Convention and Visitor Center at 1-800-525-3669.



AEMS MACRO ASSEMBLER/LINKER

A major new tool

By BRUCE HARRISON

When writing a review, there's always the old problem of whether to start with the good news or the bad news. This time the coin came up tails, so your curmudgeonly author will start with the good news.

Art Green still knows how to write an excellent assembler. For some years now, we've been using his RAG Macro Assembler (Ver. 8) as our one and only means of assembling source files. Even for somebody who never uses a macro, Art's assembler has earned its keep in our opinion by the superb way it reports errors on the screen. The new one is, if anything, even better in this respect than his earlier version. As part of our testing, we took one of our own source files and purposely changed the spelling of a label reference. CLR @FKLCNT became CLR @FLKCNT. That seemed a good test case, as transposing letters in a label name is probably our most frequent "detectable at assembly" error. This error is in a file that is "copied" by the main assembler source file, not in the main file itself. Sure enough, the AEMS Macro Assembler reported as follows:

```
LINE 10 COMPCD2 FLKCNT UNDEF
SYMBOL
```

That's really superb error reporting, since it not only tells us what's wrong, but shows the offending symbol and tells in which line of which file the error was found. Nobody who's into assembly programming should have any trouble using this product, once it's up and running, that is.

THE BAD NEWS

You knew this was coming, so here's the first part of it. When Laura Burns first asked me to review this product, I asked about the hardware required to use this assembler, namely the Asgard Expanded Memory System (AEMS). Yes, John had the card, and agreed to loan it to me for the review. There's the first part of the bad news, that this assembler is totally useless to anyone who does not have the AEMS

REVIEW

REPORT CARD

Performance.....A
Ease of Use.....B-
Documentation.....A-
Value*?
Final GradeA-
Cost: \$119.95**

Publisher: Asgard Software, 1423 Flagship Dr., Woodbridge, VA 22192 (703) 491-1267

Requirements: TI-99/4A Console, Monitor, DSDD disk drive, AMS, AEMS or AMS2 memory system.

* Value is hard to assess — see text.

** The 128K AMS card at \$119.95 is the minimum package in which the Assembler/Linker are included "free," and is required to use the software, so that's the "price of entry" into this software.

installed in his P-Box.

The package arrived, with the card and two disks. One disk in the package was an obviously incomplete early version of something, but no assembler. The second disk would not catalog on any of the four disk drives on our two TI machines. That, as we suspected, was because this disk (the "correct" version of the package) was a DSDD disk, which our TI controllers can't handle. Fortunately, in the box with the card and disks was a slip of paper (very elegant looking) with a phone number for Asgard. From the area code, I guessed that this number would be answered by Chris Bobbitt, and I was right.

SOME MORE GOOD NEWS

Chris quickly agreed to send me a set of two DSDD disks, and these arrived after the usual mail delay for this area. These disks turned out to be rather tightly packed with files, most of them archived. A copy of the Archiver had been thoughtfully provided on each disk, and each included a short un-archived README file, so I'd know what I was doing.

Five DSDD disks were required to unpack all the contents of those two disks. Except for an I/O ERROR #6 problem in unpacking the linker ARCFILE, all went fairly smoothly. We found the primary "doc" file that we sought, and found that there was even a "tailored" version of the TI-Writer Formatter provided on the disk, in which the file name for the assembler's User's Guide came up as the default in the File Name field. (A really nice touch, since I always have trouble remembering file names like AEMSMADUG when the formatter is asking me that question. (I'm pretty sure that's not the correct name, but thanks to the pre-defaulted formatter, I didn't need to know.)

GETTING IT STARTED

When the Assembler User's Guide finished printing its modest six pages (a surprise in itself), I started reading. Being the kind of impatient person who never really reads instructions, I skimmed down to the part where it said to run the assembler from file AMAC1. I failed to notice what was in the line just above that, to wit: "... and the AEMS System Software."

Again being impatient, and knowing that the AEMS 128K card was sitting there in my P-Box, I plunged right ahead into Option-5 for DSK1.AMAC1. I/O ERROR 07 happened. Okay, let's try loading it from the RAMdisk menu's RUN option. This loaded up the assembler, and gave me a nice screen for entering source, object, and so on. Since I had a source file right there on my RAMdisk, I typed in the source and object names, and pressed Enter. BOOM! The screen went blank, then nothing. Only the on-off switch had any effect. Why? (Hint: see that part above about "failed to notice.")

In my little brown book, I keep lots of phone numbers for people in the TI community, so I turned to the Gs and called Art Green in Gloucester, Ontario. Art straightened me out in just a minute. One of those other disks had a file called ABOOT, and that has to be run in order to

(See Page 22)

AEMS MACRO ASSEMBLER/LINKER—

(Continued from Page 21)

load and run the assembler. Sure enough, running ABOOT from Option-5 gave me a menu which included the macro assembler, and this allowed me to run the assembler with complete success. Of course none of my source files includes Macros, and none of my programs assembles to anything even close to 128K of memory use, but the assembler seemed to work exactly as expected.

THE LINKER

We said early on in this review that the assembler is very easy to use. The linker, on the other hand, is a good deal more complex, even though the user interface has been made fairly simple. The main reason for this is that the linker is the part of this product which resolves the matter of "paging" memory, so that apparently "seamless" programs of more than 32K length may operate from the AEMS. The linker's documentation is many pages long, but clearly written, and is designed to allow the user to use only what he needs among its many features.

The only real confusion we suffered was that if only a single object file is to be linked, that must be entered at the prompt for "Control File". (Maybe it should say Control/Object) We've always in the past been able to get along without linkers or libraries on the TI, but of course that's because we've been able to make each program we write into just a single object file, and where necessary have written our own "saver" code to make Option-5 program files from our object code.

The new linker bears a striking similarity to Art Green's earlier RAG Linker, which was a companion to his RAG Assembler. It offers so many options that it would be easy to get bogged down even trying to describe the possibilities. Suffice it to say that when you're writing programs large enough to need AEMS, you're going to have to study the linker's docs closely just to get started on the right foot. We've studied the linker docs to some extent, and found that the linker places restrictions on the construction of long programs. In essence, the linker requires a highly structured approach, using the 4K module concept, with a strictly vertical "tree" form between the main module

and its subroutines. As our regular readers know, this is not the kind of approach we follow in our own programs, but many assembly writers do adhere to this method. In the case of programs designed to run on the normal TI, it makes no difference whether the program is organized into modules, nor whether there is any structure at all. Everything will work if the instructions are correctly written. This is very definitely not the case for programs written in the AEMS system.

We tried taking one of our own object modules and linking it with the AEMS Linker. To be fair, this object file was designed strictly for Option-3 operation under E/A, and without some more work on the source file, it wouldn't have run under E/A's Option-5 either. The resulting program file would not run in the AEMS environment. Yes, this was a stupid and unfair test, but we mention it just to point out that the transition from the "normal" TI Assembly program to the AEMS environment is not a trivial matter. There will be a learning curve.

THE DOCS

Copious documentation is included with the AEMS package, including reference material on the 9900 assembly language, detailed instructions for using the linker, etc. As with any new product, there are some errors in the docs, but these are more an annoyance than a real problem. There is some inconsistency about the status of the assembler itself. Its user guide states on the cover page that this manual and the AEMS Macro Assembler are copyrighted products of RAG Software, but the very next page says that "This package is being made available via the Fairware concept." Which is right? We are not fond of the Fairware concept, and how can a copyright be enforced when the manual clearly and explicitly encourages the buyer to "... distribute complete copies to your friends." As we understand copyright law, giving the buyer "carte blanche" to copy the product and distribute it would void any future infringement claim by the author. In any case, the Assembler's User Guide is easy to read and mercifully short.

SYMBIOSIS

John sent a short note with the package containing the AEMS card and disks, in

which he asked that we review the assembler and linker, but not the card, or else make the card a separate review. We're trying to isolate this review to just the assembler and linker, but that's difficult because neither can be used without the card and its "system software." The reverse is not entirely true, because a card purchaser could use the system software and other people's programs without ever using the assembler and linker which were supplied with the card. Of course nobody can write software that makes use of the card without using the assembler and linker, but that's not quite the same thing. As we understand it, Asgard's new word processor (First Draft/Final Copy) has been designed to use the card's memory if that's available, and to run on a "normal TI" if the card is not there. If AEMS becomes very popular, this may become the trend for assembly authors. We are biased on this point, because there's nothing we've ever wanted to do on the TI that wouldn't fit into the 32K memory with room left over.

A LIMITED AUDIENCE

As the years go by, fewer and fewer assembly programmers are available to the TI community. Some subset of this group will be sufficiently interested to purchase AEMS and produce software which takes advantage of its memory capacity. Another subset of the non-programming TI public will be owners of the AEMS, and will thus form a customer base for that subset of the assembly programmers. Our own opinion is that both of these subsets will be small minorities, and that the work of programming for the AEMS will become a labor of love, not a profit center.

MINOR FLAWS

Every new product has to have flaws, and here are a couple we found.

The new assembler starts out with the cursor in the Source File input field, but there are default answers already on-screen for the Macros File and Options fields. Neither of those defaults was to our liking, so we had to back the cursor up the screen and change them. In the documentation, we found that Art provided a "Patcher" program that would allow us to change the defaults in the assembler pro-

(See Page 23)

AEMS MACRO ASSEMBLER/LINKER—

(Continued from Page 22)

gram, by simply editing the contents of a D/V 80 patch file. We tried that, but all we got was a screenful of error messages, the last being "File Not Patched." After several unsuccessful tries, we dug out a copy of Funnelweb 4.4 that had John Birdwell's DSKU on it, and used its sector search-replace function to change the entries for the defaults in Art's assembler. This worked, so that our "working copy" of the new assembler has our choice of defaults in place. On the older RAG Assembler, there was a simpler Install process which worked quite well to tailor the defaults. Perhaps Art should have kept that older concept.

Another minor flaw was noted in the System Software. There's an ability provided to bring programs into the memory on a "stay resident" basis, and then to restart them later. These resident programs are added to the system software's menu, and stay available while you're in the AEMS system. You can, however, exit back to the TI title screen, then re-enter AEMS through a program called ASHOE. The "resident" programs will still be listed on the menu, but will not work.

A final minor annoyance is the presence on the System Menu of certain selections (PEDIT, for example) that are *not* includ-

ed in the package. We were able to edit the script file that provides these as default selections, but why should the user have to delete things that the vendor left out?

Of course these latter two problems should not reflect on the assembler or linker, but people who use these will be exposed to the AEMS system, so we thought it only fair to include them in this review. We said right at the outset that Art Green has made an excellent assembler, and we meant that. The linker will take some significant study to master, but anyone who's writing software large enough to need AEMS should be willing to invest the time to learn the linker's rules. Our minor gripes about the system software are just that, not by any means to diminish the "Artistry" that went into the AEMS Macro Assembler and Linker.

TERMINOLOGY

Throughout this review, we have used the acronym AEMS for Asgard Expanded Memory System, just to be consistent internally. There is a family of such products, so the names AMS, AEMS, and AMS2 may be found to apply to various members of this family of products, and all three of those names will probably appear in advertising for the products. The hardware implementation is somewhat different in the three related products, but

what this review has to say about the assembler and linker should apply whether the product you're considering goes by the name AMS, AEMS, or AMS2, as the software is common to all three.

FINAL THOUGHTS

The idea appears sound. With this new Assembler/Linker and the capability to greatly expand directly-usable memory, the TI can reach new heights in performance. If enough assembly authors can be convinced to buy the AEMS package and produce new programs for it, TI users with AEMS will be able to eclipse the Geneve, and maybe rival the growing PC and Mac trends. It's really a shame that this couldn't have come along ten years ago, but at least now the door is open for "Windows" or "Lotus Symphony" systems running on the TI. Maybe somebody will be willing, given this major new tool, to invest the year or two of effort required to create the "Mac Emulator" for the TI. It's very hard, without a crystal ball, to assess the value of this new Assembler/Linker software, since its principal value lies in what people do with it. By itself, it can't be used on the standard TIs that we own, but given the right environment it has the potential for being the most valuable thing ever created for this machine.

PROGRAM IMAGE FILE COMPRESSOR

It's as if your disks suddenly got bigger

By BOB CARMANY

One sleepy Sunday afternoon not too long ago, I was going through a stack of freeware disks looking for another database program. I think it was about the twelfth or so that I looked at and saw a series of files PFCDOCS, PFC1 and PFC2. It sounded like "Personal File" something-or-other and I set it aside to look at a bit later. Much later that afternoon, I finally got around to looking at the documentation file. It wasn't "Personal File" something but rather "Program File Compressor V.1.0"; just another in the line of archivers, I reasoned.

The deeper I got into the documenta-

REVIEW

REPORT CARD

PerformanceA+
Ease of useA+
DocumentationB
Value.....A+
Final Grade.....A

Manufacturer: Ken Holtman, Hillenraderkamp 6, 6071 EC Swalmen (The Netherlands)

Requirements: TI99/4A, disk system, memory expansion, Editor/Assembler or Funnelweb or program image file folder.

tion, the more promising this little beauty looked. It would compress a program image file and add a runtime decompression code fragment as well. What this meant was that you could compress the program and save it to disk and then run the program any time you wanted without having the additional step of decompressing it first. This was definitely something to be explored!

Performance: The PFC package does exactly what it claims and does it very well indeed. It takes an Editor/Assembler 5 (program image) file and compacts it about 25 percent in most cases. A full 33-

(See Page 24)

PROGRAM IMAGE FILE COMPRESSOR—

(Continued from Page 23)

sector file will be compressed down to between 24 and 27 sectors including the runtime decompression code. The first of the menu options — Compress, Add Runtime Decompressor — is the one that adds this small segment of code (two disk sectors) that decompresses and runs the original program once it has been loaded into memory from disk. Multiple loading EA5 files are automatically processed in turn and saved under the appropriate filenames. The program is written in c99 and is very quick and efficient. The PFC1/2 files can be renamed to something more to your liking if you wish. I have them installed in Funnelweb on my system as PF/PG and they work just fine.

I did find some program image files this utility will not handle. I assume that it is because of some system requirements or other environmental restrictions. Current examples are the V.5.0 F'WEB editors and the V.4.40 FW file. These programs will compress, but there is a problem with the runtime decompressor and they will not execute.

On the other hand, the list of programs that PFC is compatible with is very impressive. It handles DM-1000 V.6.1, Wycove Forth V.3.0, CMINDEX, DISKASSEMBLER, Advanced Diagnostics and even Archiver with ease. In fact, the only programs other than the F'WEB internal programs not successfully processed by PFC were those that used a second file to load the EA utilities separately. Everything else worked just fine.

The only minor drawback is a second or two delay after the program is loaded into memory before it executes while the runtime decompression takes place. It is scarcely noticed in most cases, especially from RAMdisk.

Ease of Use: Everything about this program is simple to operate. The initial boot menu has five choices:

- 1 Compress. Add Runtime Decompressor
- 2 Decompress
- 3 Compress to Non-Executable File
- 4 Copy Program File
- 5 Exit

The only one that might cause confusion is the first, and most valuable, option. When you select it, the program prompts you for the input filename and also for the output filename. From that point, everything is automatic. Both single and multiple files are processed automatically and saved to the appropriate disk drive without further user interaction. In fact, simply by paying attention to the on-screen prompts, you can use the program independently of the documentation.

Since the PFC package is written in c99, final exit asks if you want to rerun the program via the standard c99 prompt. Entering "N" at this point completely exits the program. Options 2-5 do exactly as expected. The second option decompresses a previously compressed file to allow for direct sector editing, for example. Option three compresses an EA5 file but doesn't add the runtime decompressor and hence the program won't execute when selected. The space saving in this case is a bit more than the 25 percent advertised in the documentation because the two sectors of runtime code aren't there.

Option four will copy either a compressed or decompressed file, and the last option exits the program. All in all, it is a very simple program to operate.

Documentation: The documentation is in the form of a four-page text file on disk with the program. Everything is explained quite well, but there could have been a bit more background information differentiating this program from the more commonly encountered archiver programs. This would have especially helped the novice user. However, it is certainly adequate to operate the program, and the lack of additional information didn't detract from the usefulness of the PFC package.

Value: What can you say about a program that is absolutely free! The distribution says that it is released as a public domain program. If there ever was a case of getting "something for nothing," this is it. I don't know how you could find a better value anywhere!

Final Grade: What else remains to be said? Here is a program with the potential

to increase your disk capacity by 25 percent without having to buy the first piece of hardware. It is fast, being written in c99, efficient, easy to operate and absolutely free. There is no way that it could be anything but an "A" program.

Comments: The only caveat that I would place on the use of this program is to test the EA5 files that you want to convert by trying it on a backup copy before you convert any original. You may find that not everything will run properly after it has been converted. Thoroughly test the compressed program by running it several times and checking out all the options. That's just common sense.

The potential disk space savings can be enormous. In my case, I have three 500K Quest RAMdisks in my system, and most of the files that fill the 5,000 sectors of program storage space are EA5 files. The possibility of freeing 1,000-plus sectors of space through this program gets one's immediate attention — that's an area larger than a small Horizon RAMdisk! As a brief test, I went through the first 1,500 sectors of one RAMdisk that was roughly a 50/50 split of EA5 and Extended BASIC programs and, with no real effort, was able to free more than 100 sectors from among the EA5 files without even trying them all!

You don't have to have RAMdisk or other high-powered storage capacity to take advantage of PFC. The idea of saving 60 or so sectors on a single-sided, single-density drive or a couple hundred sectors on a double-sided, double-density drive would be well worth the effort. It would allow one to put a couple more executable programs on disk. For those with hard drives, the potential space savings are astronomical!

If you have difficulty finding this program and want a copy of it, I will gladly furnish a copy of both the documentation and two program files if you contact me at the address below and send along a self-addressed, stamped mailer and disk. Send to: Robert M. Carmany, 1504 Larson St., Greensboro, NC 27407.

MICRO-REVIEWS

Membership Manager and Newsletter Exchange, Minesweeper, The Spider's Guide to Funnelweb Configuration

By CHARLES GOOD

MEMBERSHIP MANAGER and NEWSLETTER EXCHANGE

Last month I reviewed a good "name/address/phone" program. Membership Manager is another such product specifically tailored for 99/4A user groups but potentially useful to any organization that has a membership roster. It is a TI-Base template. First you load TI-Base (Ver. 3.01 or later) and then you type DO MENU to boot Membership Manager

Data stored for each member includes name, address, city, state, ZIP code, nation, membership number, date joined, date membership expires, scratch pad notes (up to 40 characters) unique to each member, and three user-defined fields of information. You can search and sort data in any of these fields, including the user-defined fields, and of course you can print reports and mailing labels.

The design of the software does a good job with foreign addresses and postal codes. Address labels for foreign locations are nicely formatted. Some examples of user defined or scratch pad information might include a member's system hardware (hard drive, Geneve, 80-column card, DSDD floppy capacity, etc.) and stuff the member has checked out of the group's software or hardcover library.

Members can be marked as "inactive" without deleting member information from the database. All user groups have such inactive or former members whose dues have not been paid for some time. You can create sorted lists that include only active members if you want. Group officers do not necessarily want to discard data on inactive individuals. It is, of course, also possible to permanently delete a member's data from Membership Manager's database.

The size of the database is limited only by the size of the storage media. Membership Manager is optimized for use with a hard drive, but works nicely off a RAMdisk (good speed) and can be used with SSSD drives.

One feature I really like is access to TI-BASE functions from within Membership Manager. You can format disks, catalog disks, and copy files all from within Membership Manager.

Newsletter Exchange is a second TI-BASE template designed for 99/4A user groups, separate from and not directly accessible from Membership Manager. You load TI-BASE V3.01 or later and then you load Newsletter Exchange. Some user groups go to great lengths keeping track of exchange newsletters received from other user groups, checking to verify that the other groups are regularly sending out exchange newsletters.

Newsletter Exchange is designed for this sort of record keeping. The database contains the name and address of other user groups. It tracks dates of first and last newsletter received from other groups and dates of the first and most recent newsletter sent by your group to each of the other groups. A group's exchange officer can check how long it has been since a particular group sent an exchange newsletter and the total number of newsletters received or sent, etc.

One indication of the professional quality and attention to detail of both Membership Manager and Newsletter Exchange is the documentation. It comes on disk and prints neatly from DSK1.LOAD into 24 pages of text plus a nifty title page and table of contents page. The resulting printed copy is well written, nicely formatted, and makes good use of double strike and enlarge print to enhance its general appearance. Both databases also have well written on line help screens that you can call up at any time. This is a feature that

should be a part of all application software.

Membership Manager and Newsletter Exchange can be purchased together as a single commercial (not shareware, not for distribution via user group libraries) package from the author for \$25 including postage. Send your money to Bill Gaskill 2310 Cypress Court, Grand Junction, CO 81506. TI-BASE is available Texaments and other 99/4A software dealers.

MINESWEEPER

This is a game for the 99/4A (or Geneve) that has the "look and feel" of the PC Windows game of the same name. The basic Microsoft Windows environment used on many PCs these days includes a few games, and Minesweeper is one of them. Windows can sometimes be used as a multi tasking environment. While your PC at work is doing some number crunching you can play Minesweeper, if the boss isn't looking.

Minesweeper (both the /4A and PC versions) is a logic game that can be quite addictive. The playing field consists of squares, each of which may or may not have a mine underneath. The object of the game is to uncover all of the squares without mines before time runs out. When you uncover a square, you are informed how many mined squares are immediately adjacent. Of course if you uncover a mine you are dead.

The 99/4A version of Minesweeper is written in c and has excellent movement of its cursor in a manner that simulates what you get with a PC's mouse. The joystick or the WERSDFZXC keys move the cursor, and the fire button or "Q" key are used to "click" on a screen object. The "version 1" that I am reviewing does not support a mouse, but the author says he intends to purchase one of the various TI rodents with the initial fairware donations

(See Page 26)

MICRO REVIEWS—

(Continued from Page 25)

received and incorporate mouse control into future versions. Cursor movement is precise and fast, just like using a mouse on a PC. When you begin to move the cursor, it initially moves slowly in the selected direction and then speeds up. I really like the automatic increase in cursor speed which allows you to precisely move the cursor quickly to a point on the other side of the screen.

Across the top of the screen are the number of mines not yet uncovered, the time clock (max 999 seconds) and some words you can click on. Custom allows you to select predefined sizes ("Easy", "Okay", or "Hard") of minefields, or to create a custom mine field of any size and complexity desired. Clicking on Info gives you a fairware notice and address. If you click on Help (available any time in the game) complete instructions are displayed. Scores gives you the highest scores yet obtained (minimum elapsed time until all mines are uncovered) in each of the three predefined levels of play. High scores with player names are saved to disk.

Several of my colleagues at the university play the Windows version of Minesweeper. I have shown two of these people my 99/4A version, (when the dean wasn't around) and they think it is as good as the Windows version.

Minesweeper is fairware, will fit on a SSSD disk, and boots from any EA5 loader such as Funnelweb. It is available on TI related BBS systems, from user group libraries, or directly from the author. No specific donation is specified but you should probably send at least \$5, par-

ticularly if you want the author to send you the game on his disk using his disk mailer and postage. If you send \$10 and request it, the author will send you the source code along with the game. David Orman, 212 S. Nema Pl., Tucson, AZ 85711 (home of the Southwest 99er User Group).

THE SPIDER'S GUIDE TO FUNNELWEB CONFIGURATION

Configuring Funnelweb with its Configure utility confuses many potential Funnelweb users. There are many menus, sub menus, and hardware specifications that Funnelweb needs to know about. Funnelweb is designed to operate in all TI99/4A and Geneve environments, and since there are lots of possible environments (from one SSSD drive on up to more complex stuff) the configuration process is a bit complex. Spider's Guide is a little 24-page book that walks you through the entire process step by step.

Spider's Guide is full of screen pictures. These reproductions of Funnelweb screens are what make the book such a superb tutorial. Most users should be able to follow the book with no trouble as they sit at their computers with one of Funnelweb's introductory menus on screen in front of them. A small sample quote from page one shows you how the book works: Select Configure from the menu of Funnelweb, or enter CF from any option 5 loader and it will load. You should have a screen that looks like this — the book shows picture of intro configure screen. At this point, just press any key. This should be on screen now. — the book shows a picture of the second configure screen.

I highly recommend this book to anyone who becomes confused when trying to configure Funnelweb. It is better than any of the video tapes (including mine) and newsletter articles (including mine) previously published on this subject. The only subject not covered is customizing Funnelweb's V5 text/program editor, which is configured separately from the rest of Funnelweb.

Here is an interesting bit of trivia, in case you don't know: Why is the book called The "Spider's Guide" to Funnelweb Configuration?

Funnelweb (the software) is named for a spider that lives at Funnelweb Farm, the Australian home of Funnelweb's authors Tony and Will McGovern. Funnelweb spiders live in the ground and build funnel shaped webs over the entrance to their holes. They are probably the world's most poisonous spider, with a bite that can kill small mammals and occasionally even humans. The McGovern's have to be careful when they walk out to their roadside mail box.

The Spider's Guide to Funnelweb Configuration book is available directly from the author for \$3.50 + 50 cents postage. Send your money to Larry Tippett, 5826 Buffalo St. Sanborn, NY 14132.

I have pretty much reviewed my supply of 99/4A products I have at my computer shack that were not previously reviewed in MICROpendium. Please send me some more items for Micro Reviews. Products that are never submitted are never reviewed. Send your goods to Charles Good, P.O. Box 647, Venedocia, OH 45894. If you want to chat feel free to phone me evenings at 419-667-3131.

USER NOTES

Transliterater

This comes from Jim Peterson, the Tigercub. He writes:

We all know that the TI-Writer Formatter insists on giving us five blank lines at the top of the page and three at the bottom. If your printer supports a reverse line feed, you can take back those lines at the top by

beginning the page with a line CTRL-U and RJRJRJRJ; but I can't find a practical way to print the three lines at the bottom. I wanted to use the formatter's .TL commands to print out a form 66 lines long, so I wrote this little program. It reads the .TL commands and interprets them just as the formatter does, although somewhat more slowly.

```
100 DISPLAY AT(3,5)ERASE ALL
: "TRANSLITERATER" :: OPEN #2
: "PIO", VARIABLE 254 !209
110 DISPLAY AT(12,1): "Filena
me? DSK" :: ACCEPT AT(12,14)
BEEP:F$ :: OPEN #1:"DSK"&F$,
INPUT :: F=0 !194
120 DISPLAY AT(14,1): "How ma
```

(See Page 27)

USER NOTES

(Continued from Page 26)

```
ny lines per page? 66" :: AC
CEPT AT(14,26)SIZE(-2)BEEP:L
!058
130 DISPLAY AT(16,1):"How ma
ny copies? 1" :: ACCEPT AT(1
6,18)SIZE(-2)BEEP:H !059
140 FOR K=1 TO H :: C=0 :: R
ESTORE #1 !017
150 LINPUT #1:M$ :: IF SEG$(
M$,1,4)<>".TL " THEN 180 ELS
E IF SEG$(M$,1,4)=" .TL " AND
F=1 THEN 150 !156
160 M$=SEG$(M$,5,255):: P=PO
S(M$,",",1):: X=X+1 :: A$(X)
=CHR$(VAL(SEG$(M$,1,P-1)))::
M$=SEG$(M$,P+1,255)&"," !00
5
170 P=POS(M$,",",1):: B$(X)=
B$(X)&CHR$(VAL(SEG$(M$,1,P-1
))):: M$=SEG$(M$,P+1,255)::
IF LEN(M$)>0 THEN 170 ELSE 1
50 !199
180 IF ASC(M$)>127 THEN 220
!184
190 FOR J=1 TO X !151
200 P=POS(M$,A$(J),1):: IF P
<>0 THEN M$=SEG$(M$,1,P-1)&B
$(J)&SEG$(M$,P+1,255):: GOTO
200 !196
210 NEXT J :: PRINT #2:M$ ::
C=C+1 :: IF C=L AND EOF(1)<
>1 THEN PRINT #2:CHR$(12)::
C=0 :: GOTO 150 ELSE IF EOF(
1)<>1 THEN 150 !095
220 PRINT #2:CHR$(12):: F=1
:: NEXT K :: CLOSE #1 :: GOT
O 110 !244
```

Hardware fix for shadows

If you've noticed white shadows around the images displayed on your monitor, you should know that there's a relatively simple fix that can be used to get rid of them.

The problem, which commonly affects less expensive monitors, is that the TI console puts out a spurious signal that causes this "ringing" on the monitor screen. To get rid of it, you'll need to make a trip to Radio Shack and pick up a .005 MFD ceramic capacitor (Cat. #272-130). The capacitor should be placed across the video

input wires to the monitor. The easiest way to do this is to buy an RCA type monaural Y adapter (Cat. #274-304) and an RCA phono plug (RCA #274-339 or #274-321) and solder the capacitor to the phono plug. Then plug the Y adapter into the video input on the monitor. Put the plug with the capacitor into one side of the Y adapter and video output from the console into the other side.

Don't be surprised if the characters on the monitor are easier to read.

Another way to eliminate the white shadows around characters is the following hardware fix. This involves opening the console and is done entirely at the reader's risk.

Referring to the diagram of the console (Fig. 1), locate the VDP Load Resistor (most likely 560 ohm). If your board is not exactly as shown in the diagram, following pin 36 from the VDP chip, through a couple inductors to the 560 ohm resistor, to ground. Replace the 560 ohm resistor with a 330 ohm, one-quarter watt resistor and you can expect to see a substantial improvement in display quality.

Slow down the cursor in DM-1000

If you find that the cursor in DM-1000 is too fast you can slow it down by doing the following:

Copy MGR1 to a newly formatted floppy disk and load your favorite sector editor. Load sector 36 from MGR1, or do a string search for 8000A0FF. The 00A0 is what you want to change. Replace it something in the range of 00A0 to 0700. Try 010C. Write the sector back to the disk and

try DM-1000.

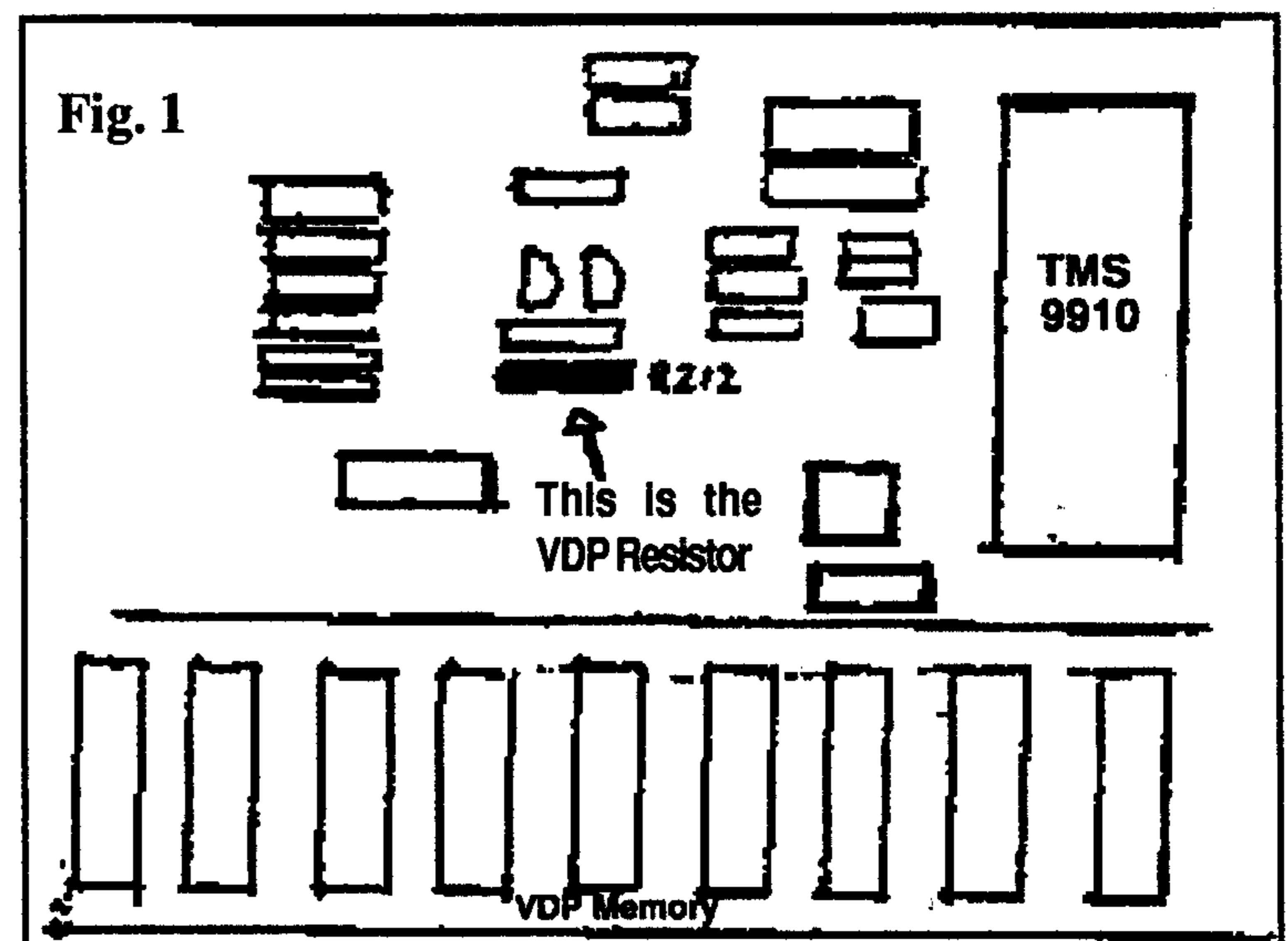
Arrays explained

The following was written by Andy Frueh and has appeared in several user group newsletters.

I have considered myself decent in Extended BASIC programming for a few years. However, one thing eluded me. I could not figure out what "dimensional array" meant. Obviously, it was a powerful thing, but I could not quite figure out how to use them. Why would I want to?

For those who are just starting out, strap yourself in. This could be a bumpy ride.

How can I explain arrays? I suppose the



same way I explained them to myself when I finally understood them. An array is nothing more than a chart in the computer's memory. It has a certain number of rows and a certain number of columns.

I assume most of us understand this BASIC command:

```
10 A=5
```

This sets the variable A to the value of 5. The next line could say:

```
20 PRINT A-5
```

When that program runs, your answer would be zero. A=5 and if you subtract 5 from 5 you get zero. Simple.

An array is a slightly more complicated set of variables:

```
10 DIM A(5)
```

Since there is one number in parentheses, it is a "one dimensional" array. You

(See Page 28)

USER NOTES

(Continued from Page 27)

could say it has 1 row and 5 columns. Your next line could say:
20 A(0)=1 :: A(1)=2 :: A(2)=4 :: A(3)=8 ::
A(4)=16

Hang on! Why did I start with zero? Because, to the computer, counting starts with zero. I realize that this confuses some people. If you would rather start counting with one, place this statement at the beginning of your program:

```
OPTION BASE 1
```

What the heck does that mean? Well, you can use OPTION BASE 0 or OPTION BASE 1 in your programs. You use OPTION BASE only once. This tells the computer whether you are going to start counting with zero or one. If we had OPTION BASE 1 as line five in our little program, the numbers in parentheses in line 20 would each have to be increased by one, so they would be values from 1 to 5.

In our program (let's use OPTION BASE 1), we have set up a one dimensional array, with five columns or elements. You cannot get a picture in your mind? Well, basically, this is what we have:

```

-----
()  1  2  3  4  5
Array
A Value:  1  2  3  4  5
-----

```

We can use more than one row. Here is another little program and its chart:

```
5 OPTION BASE 1
10 DIM A(2,2)
20 A(1,1)=1 :: A(1,2)=2 :: A(2,1)=3 ::
A(2,2)=4
```

```

-----
1   1  2
Array ,  ---
A   2  3  4
-----

```

You may have seven dimensional arrays in Extended BASIC. For example, you might need to write an address program. You decide that each piece of information will be stored separately. For example, although the street address is asked for one line, the number, street and apartment number are all going to be stored in a difference variable. You could do this:

```
5 OPTION BASE 1
10 DIM A$(3,3,1,1,1,7,2)
```

The first set of three would be first name, middle initial and last name. Three pieces of information are in the FIRST DIMENSION of the array. The next set of three would be the street number, street name and apartment number. Next, the user would enter the city. This is the only piece of information in this dimension (dimension 3). The fourth dimension would contain the state, and the fifth would be the zip code. Dimension 6 could be used to store seven pieces of information on the person at that address. Profession, where they work, etc. The last and seventh dimension has two pieces of information to it. The first would be the area code, and the second, the phone number.

How about a "practical" illustration? The program below will load in two full screens of a text file. We have 28 columns available to us normally in Extended BASIC, using the PRINT routine. There are 80 columns in a standard text file (D/V 80). Some quick math tells us that we will need three screen lines to display one text file line. OK, some more quick math. There are 24 lines on a screen, so we can only put eight lines of the text file on one screen. If our program loads in TWO screens, we would need a total of 16 text file lines. Our math is now out of the way ... we know what we need to do.

The program below will load the text into the array TEXT\$(). It is two dimensional. Remember, that means that there will be two major pieces of data. These two pieces will be the number of screens and the lines in each screen. For example, in the program below, I use DIM TEXT\$(2,8). The way I am using it is like this: the two represents the "pages" or number of screens. In TWO screens, I need eight LINES(>)

```
10 OPTION BASE 1
20 DIM TEXT$(2,8)
30 CALL CLEAR::INPUT "Text file?" :A$
40 OPEN #1:A$, INPUT
50 FOR S=1 TO 2::FOR L=1 TO 8
60 INPUT #1:TEXT$(S,L)
70 NEXT L::NEXT S::S=0
80 CALL CLEAR::S=S+1
90 FOR L=1 TO 8
100 PRINT TEXT$(S,L)::NEXT L
110 CALL KEY (0,K,ST)::IF ST=
```

```
0 THEN 110 ELSE IF S<2 THEN 8
0 ELSE END
```

You may get an error message if the text file is less than a total of 16 lines. Watch out for this! You will need to press a key between pages. If all eight lines are not completely filled, neither will the screens be filled.

Testing for 32767

The following was written by Ross Mudie and appeared in the TISHUG News Digest (Sydney, New South Wales).

When writing and developing a program in either BASIC or Extended BASIC on the TI99/4A, the programmer will most likely decide to use the RESequence command to make the line numbers evenly spaced. If the RES command finds a GOTO, IF THEN ELSE, etc. which references a non-existent line number, the line number reference is changed to 32767. If the program does not have a line numbered 32767 then, when the program is RUN, it will stop when the reference to the non-existent line number is encountered. If the program happens to have a line with the number 32767, unexpected results may occur when the program is executed.

When I use the RES command in a large program, I always check for references to line number 32767 before trying to run the program. This saves a lot of effort trying to figure out things the hard way.

To search for one or more occurrences of 32767, the following steps are used:

1. List the program to disk. This creates a DISplay/VARiable 80 file on the disk. I always use the file name LIST, thus when my program which I have just RESequenced has been saved, I use LIST "DSK1.LIST."
2. Load the Editor/Assembler, TI-Writer or Funnelweb editor and then load the file named LIST into memory.
3. Use the Find String command to find any occurrences of 32767 or 327 or 767 in the file. (The reason for specifying 327 or 767 is that in the LIST process it is possible for the 32767 reference to be split over the end of one line and the start of the next.)
4. Note the program line numbers in (See Page 29)

USER NOTES

(Continued from Page 28)

which any bad references occur, then go back to the program and make any necessary corrections.

Some of the programs that I develop create a LIST file which is too big for the editor to load in one go. There are two ways around this problem. If you are using TI- or FW-Writer, load the second half of the file by specifying the line numbers of the file when loading. To do this, use the LoadFile option and specify the start and end line numbers of the file segment to be loaded in front of the file name:

```
250 500 DSK1.LIST
```

This will load lines 250 to 500 of the file named LIST from DSK1.

There may be someone else like me who doesn't always have TI- or FW-Writer handy and still wants to check a LIST file for line number references to 32767. This is where the following little Extended BASIC program comes in handy.

```
100 ! SVE DSK1.TEST32767
110 DISPLAY ERASE ALL:"Test
for 32767, 327 and 767": :
120 OPEN #1:"DSK1.LIST"
130 I=L+1 :: DISPLAY AT(24,2
5):L :: IF EOF(1)THEN 160 EL
SE LINPUT #1:A$
140 IF POS(A$, "32767",1)OR P
OS(A$, "327",1)OR POS(A$, "767
",1)THEN PRINT SEG$(A$,1, POS
(A$, " ",1))
150 GOTO 130
160 CLOSE #1
170 PRINT : "Finished"
```

This little program reads each line of the LIST file. If any occurrences of 32767, 327 or 767 are found, it prints the start of

Long-time Chicago member dies

Floyd Clayton of Des Plaines, Illinois, a long-time member of the Chicago TI Users Group, died Sept. 9.

Funeral services were held in Arlington Heights, Illinois. Burial was in Missouri.

His son, Tom Clayton, is a member of the Chicago Users Group's executive board.

the line in the file containing the possible bad reference. The program is not intended to be foolproof or fancy, but it works for me.

Multiprint outputs multiple copies

The following item appeared in the K-Town 99er, the newsletter of the K-Town 99ers, Knoxville, TN. It was written by John H. Bull.

Each year I have the chore of providing tally cards for a bridge club — 3x5 index cards that show the table and partner for each round for each of eight players. Eight players x 4 parties x 8 months equals 256 cards. As I said, it is a chore.

There are eight different cards and we need 32 copies of each. Some years ago I made a D/V80 file for each cards, with printer codes for my NX1000. Now I load each file with TI-Writer, insert a card in the printer, press PF, enter, PIO, enter, wait for the printer, press F9, then do it again. It takes six key presses for each copy, that many more when I load the next file for the next card, and I have to keep count up to 32 and I keep losing count and have to count the pile of cards manually. The job takes about three hours. There must be an easier way!

The following program saved me about an hour. I still have to insert each index card into the printer, but the program counts them for me and it takes only one key press (enter) per copy.

```
100 ! PRINT MULTIPLE COPIES
!159
110 ! FROM D/V80 FILES !187
120 CALL CLEAR !209
130 DISPLAY AT(2,1): "Put pri
nter on line and insert
card or paper." !031
140 OPEN #1:"PIO" :: S=0 !13
7
150 DISPLAY AT(5,1): "FILE?"&
FN$:TAB(13); "or Q to quit."
:: ACCEPT AT(5,7)SIZE(-15):F
N$ !031
160 IF FN$="Q" THEN CLOSE #1
:: STOP :: ELSE IF S=0 THEN
180 !050
170 IF FN$=PF$ THEN 190 ELSE
```

```
CLOSE #2 !004
180 OPEN #2:FN$ :: C=0 :: S=
1 !013
190 C=C+1 :: DISPLAY AT(20,1
0): "COUNT="&STR$(C) :: PF$=FN
$ !199
200 FOR I=1 TO 60 :: LINPUT
#2:A$ :: IF EOF(2)THEN RESTO
RE #2 :: GOTO 150 !135
210 PRINT #1:A$ :: NEXT I ::
RESTORE #2 :: GOTO 150 !131
```

Here's how it works:

You type the filename for the first card, press enter, and it prints. After that you press enter only once for each copy. You enter a new filename for each card. In my case, that means changing just one digit in the filename — DSK1.TALLY/1, DSK1.TALLY/2, etc.

The program will work with any text files and printer codes but not with format codes. It is designed to print one page or less, but longer documents can be printed by increasing the "60" in line 200 and putting the appropriate page feed codes in the file.

How about making copies with a copier? Well, I don't one handy. Also, my experience is that most copiers don't handle 3x5 cards as well as my printer.

Keyboard reader displays codes

This item has appeared in a number of user group newsletters. It was written by Bob Webb.

This small program is among the ones I use the most. I can never remember the number associated with a key press or ASCII symbol, so I threw this thing together. Let me caution you before I continue — do not run this program until you have saved it. Once you start it, the only way to stop it is to turn your computer off.

Once the program is running, press any key and its associated number will be displayed. If an ASCII symbol is associated with a key press, it will be displayed to the left of the number.

This program doesn't break new ground, however, you might find a part of it to be of use. I have added one of my fav-

(See Page 30)

USER NOTES

(Continued from Page 29)

write little details to it. If no key is pressed for a given amount of time, it jumps to a screen-saver type of subprogram.

The BLANK variable is a counter. This clock ticks away and, if a key is pressed, it is reset to zero and begins counting again. If no key is pressed, it jumps down to line 410 and stays there until a key is pressed. The ascending number in the lower left corner is the value of BLANK as it counts.

```

100 ! KEY TO NUMBER PROGRAM
!112
110 ! EXTENDED BASIC AND 32K
!057
120 ! BY BOB WEBB, 6/91 !140
130 ! CAUTION: YOU WILL HAVE
TO !108
140 ! TURN OFF YOUR COMPUTER
TO END !159
150 !!131
160 ! CALL LOAD DISABLES QUI
T KEY !210
170 CALL INIT :: CALL LOAD(-
31,806,16)!005
180 !!131
190 ON BREAK NEXT !191
200 !!131
210 CALL CLEAR !209
220 BLANK=0 !031
230 DISPLAY AT(5,5):"KEY TES
T PROGRAM" !187
240 DISPLAY AT(7,5):"PRESS A
NY KEY" !215
250 DISPLAY AT(9,5):"ITS NUM
BER WILL" !110
260 DISPLAY AT(10,5):"BE DIS
PLAYED" !169
270 DISPLAY AT(11,5):"ASCII"
:: DISPLAY AT(11,10):" KEY"
!218
280 !!131
290 !!131
300 CALL KEY(0,K,S)!187
310 BLANK=BLANK+1 !073
315 DISPLAY AT(22,4):BLANK !
250
320 IF BLANK>1000 THEN 410 !
077
330 IF S=0 THEN 300 !052
340 DISPLAY AT(12,4):K !220
350 DISPLAY AT(12,12):CHR$(K
)!079
360 BLANK=0 !031
370 GOTO 300 !124

```

```

380 !!131
390 !!131
400 !!131
410 CALL CLEAR !209
420 CALL KEY(0,K,S)!187
430 IF S=0 THEN 420 !172
440 GOTO 210 !033

```

When your Horizon crashes

The following item has appeared in several user group newsletters. The author is unknown.

If your HRD locks up and won't access even the physical drives, DSK1, don't conclude that all is lost. Even though the system seems to work and the disk controller light and HRD LEDs are on, not even turning the console and PEB off seem to get things working.

Here's what to do:

Insert your Editor/Assembler module into the GROM port, then turn the console on, followed by the PEB. While turning on the PEB, hold down the shift key on the console. Then select option 5 from the E/A and load DSK1.CFG to configure your system. You should notice disk access return. The RAMdisk directories are still intact as well, and their contents are still there.

Next, reload the ROS you usually use. Do not throw out the existing information. Exit CFG and everything should be fine.

Avoiding Widget lock-ups

This item appeared in the newsletter of the Portland Users of Ninety-Nines (PUNN). It was written by Paul Herman.

The Widget, or cartridge expander, is a device that allows you to keep up to three cartridges ready for use at the flip of a switch. However, this little jewel has the unnerving habit of locking up the console when it is switched too fast, and without waiting for a few seconds or so.

I've found a simple solution to the problem — hold the FCTN= key (Quit key on the 99/4A) when switching the Widget. This seems to act as an interrupt, allowing you to choose a new cartridge and carry on

in the new cartridge without have to turn the console off or pressing the reset button on the Widget.

I have been using the combination for some time and have had only one or two failures.

Highlighting

The following item was written by Earl Raguse.

I saw a demonstration that allowed one to switch the foreground colors of certain characters to make them stand out from others, like O vs 0, and l vs 1, or for trouble-shooting of bad typing, something I do well.

Aha, you say — I can do that with CALL COLOR.

True, but it's not permanent. I don't like having to embed trouble-shooting routines in my programs if there is an easier way.

The following program called HIGHLIGHT makes permanent foreground/background color changes and can be controlled ON and OFF at will. Once executed, the program can be deleted with NEW before you start entering a new program. I sometimes put this in my LOAD program. It's easy to turn off if you don't want it. I found the basic program idea in the Tacoma 99ers Newsletter of December 1987; the article was by Joe Nolan, who credits Harry Wilhelm of the Twin Tiers UG with the original idea. I don't have any idea how much evolution has gone on, but I added my two cents also.

Lines 130 and 140 do all the work, and if you wish to transfer this effect to one of your own programs, that's all you need. The following tells you how you can change these lines to suit your needs. If you study it a bit, you can see the potential for other purposes.

In line 130,

(1) Change the eighth number, from the address, 17, to the number of the first character set you want to change PLUS 15. The current program is 15+2=17 for character set 2.

(2) Change the eighth number after that, 3, to the number of character sets to change. The current program is 3 for character sets 2, 3 and 4.

(See Page 31)

USER NOTES

(Continued from Page 30)

In line 140,

(1) Load a number (in this case 244) for each character set to be changed. That number is computed as $(16*(16-1))+(5-1)=244$.

(2) The effect is turned ON by CALL LOAD(-31804,63) and OFF by CALL LOAD(-31804,0). This can be done either in a program or from the keyboard. I added the lines 150 and 160 for easy control of the effect on or off. These can be deleted if not wanted.

```
100 ! SAVE DSK1.HIGHLIGHT !2
03
```

```
110 ! By Joe Nolan, Tacoma 9
9ers UG Newsletter Dec 87, O
riginal idea by Harry Wilhel
m of Twin Tiers UG !196
```

```
120 ! Modified by E. Raguse
UGOC 1/87 !022
```

```
130 CALL INIT :: CALL LOAD(1
6128,2,224,38,0,2,0,8,17,2,1
,63,36,2,2,0,3,4,32,32,36,2,
224,131,192,3,128) !032
```

```
140 CALL LOAD(16164,244,244,
244):: CALL LOAD(-31804,63) !
250
```

```
150 PRINT "TURN IT OFF? PRES
S SPACE, ELSE ANY" !189
```

```
160 CALL KEY(0,K,S):: IF S=0
THEN 160 ELSE IF K<>32 THEN
END ELSE CALL LOAD(-31804,0
)!207
```

Myths and truths

The West Penn 99ers printed a number of "Myths and Truths" about magnetic media in their newsletter. They were written by Keith Faulkner of the Lansing Area Commodore Club. Here's a sample:

MYTH

A small magnet such as the holder of a flashlight can damage tapes and diskettes some diskette away.

TRUTH

A one thousand oersted magnet, capable of lifting 40 pounds, was brought to within one inch of a computer tape. No data loss resulted. At the range of $1/2$ inch., the magnet effectively destroyed the data. The read-write head, virtually in contact with the media, has a field strength of 750 to 1,000 oersteds. An extraneous field of 200 oersteds causes a loss of data, but one of 150 oersteds does not.

CLASSIFIEDS

Policy

The cost of classified advertising is 25 cents per word. Classified display (i.e., special formatting or graphics) is \$9 per column inch. Classified advertisements must be paid in advance. Classified advertisers may request a category under which they would like their advertisements to appear, but the final placement decision is the responsibility of the publisher.

Classified deadlines will be kept open for as long as practical. For the purpose of classified advertising deadlines, any classified ad received later than the first day of any month cannot be assured of placement in the next edition. We will do our best to include every advertisement that is submitted in the earliest possible edition.

The publisher offers no guarantee that any advertisement will be published in any particular issue. Any damages that result either from errors in copy or for failure to be included in any particular edition will be limited to the amount of the cost of the advertisement itself. The publisher reserves the right to reject any advertisement.

The advertiser may elect to publish the advertisement in subsequent editions at the same charge, payable prior to publication. The deadline for carryover classifieds is the same as for new advertising.

In submitting an ad, please indicate whether you would like a refund if it is not published in the requested edition or whether you would like us to hold it for the next edition. Cancellations and refunds cannot be made after the second day of the month.

FOR SALE

TI99/4A

PHP1100	TI, 99/4A JOYSTICKS (Pair)	14.95
PHP1260	32K MEMORY CARD	99.95
PHP1500	SPEECH SYNTHESIZER	29.95
PHM3109	TI LOGO II (32K Mem. Req.)	1.95
PHP2500	TI IMPACT PRINTER	249.95
PHM3012	SECURITIES ANALYSIS	12.95
PHM3058	MINI-MEM (W/Assembler)	39.95
PHM3122	PLATO INTERPRETER	49.95

JOY ELECTRONICS INC.—P.O. BOX 542526

DALLAS, TX 75354-2526

(800) 527-7438, OUTSIDE DALLAS AREA

(214) 243-5371, DALLAS AREA

ADD \$5.00 S&H—TEX RES ADD 8 1/2% TAX

10/8 CALL OR WRITE FOR FREE CATALOG

FOR SALE

ENORMOUS TI99/4A INVENTORY. CATALOGS \$2.00. BRAATZS COMPUTER SERVICES, 719 E. BYRD ST., APPLETON, WI 54911.

1-414-731-3478.

10/12

Attend a TI Fair

Buy/Sell Used Hardware and Software

National Used Software Club has buyers looking for TI products, as well as sellers. More than 170 TI items are listed for sale. The membership fee is only \$15/year and entitles you to buy or sell computer items via the NUS/HC database. You will also receive FREE our newsletter (mailed every 8-12 weeks). If you don't buy or sell anything during the

term of your membership we will refund your annual fee.

For a free, no-obligation information packet, write to NUS/HC, P.O. Box 1343, Round Rock, TX 78680; or call 512-255-1512.

Here are some of the items offered for sale:

Horizon RAMdisk (192K)	\$125.00
Advanced Diagnostics	\$25.00
Seagate 251 HD (40 meg)	\$75.00
TI XBASIC	\$20.00

Call our BBS for weekly price and product updates — 512-255-1557. Hours are from 5:30 p.m. to 8 a.m. weekdays and from noon Saturday through 8 a.m. Monday weekends. Supports 300-2400 baud.

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 on the left 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; Vol. 4, No. 9

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

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 1.23F (req. SSSD or larger, for MBASIC).....\$4.00

MDOS 1.50H (req. for MBASIC).....\$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 Cor-Comp.

	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

SECOND CLASS

A T EXPIRES 1/94
CHARLES GOOD
P.O. BOX 647
VENEDOCIA OH 45374