Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 13 Number5 September/October 1996 $6

http://www

Home pages and FTP sites — Page 15

A TI list server — Page 21

Utility bill audit

Assembly for the Geneve

Tips on debugging

Using the p-Code system

New software for the AMS Card

CONTENTS

MICROpendium

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

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

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

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

Display advertising deadlines and rates are available upon request.

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

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

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

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

Extended BASIC

The Art of Assembly

Telecommunications

Geneve 9640

Newsbytes

Pascal

AMS Card

Reviews

User Notes

*READ THIS

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

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

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

TEX COMP LTD. FREEWARE ONLY

Public Domain and Shareware programs to meet your every need.

XB & 32K Req. for most programs

$2.95 PER DISK

LISTED BY SUBJECT

GRAPHICS, MUSIC, & ANIM.

#1 THE SINGING TI VOL.1 (S)
#4 PRINT ART (P)
#5A MUSIC/GRPAHICS
#6 XB MUSIC
#7 SPACE SHUTTLE MUSIC/GRAPHICS
#9 MONA LISA PRINTOUT (P)
#11 ANIMATED XMAS (WOODSTOCK)
#14 FIGURE STUDY (P)
#32 XB XMAS MUSIC
#41 VIDEO GRAPHS
#52 ANIMATION 99
#69 PLAYER PIANO
#93 KBGB GIRLIE CALENDAR (P) *
#103 SORGAN (ORGAN)
#107 STARTREK MUSIC ALBUM
#111 POPMUSIC & GRAPHICS
#114 PANARAMA
#115 GRAPHICS DESIGN
#120 BITMAC (P)
#230 THE SINGING TI VOL. 2 (S)
#231 THE SINGING TI VOL. 3 (S)
#246 THE SINGING TI VOL. 4 (S)
#313 3-D WOLRD

COMPUTER & PRINTER UTILS. PROGRAMMING LANG.

#3 DUMPIT (E/A)
#15 SRAT/EPSON DEMO (P)
#16 SIDEWAYS PRINTOUT
#18 TI DIAGNOSTIC (MM)
#28 LOADERS / CATALOGERS
#30 HOUSEHOLD BUDG. PRINTOUT (P)
#35 PROGRAMMING AIDS/UTIL VOL. 1
#42 FUNNELWEB FARM
#53 HACKER CRACKER
#55 SCREEN DUMP (P)
#62 DISK MANAGER II (M)
#75 DISK CATALOGER
#76 PROGRAMMING AIDS/UTIL VOL. 2
#78 ARTCON GRAPHICS CONVERTER
#79 DISK MANAGER 1000
#80 BIRDWELL DISK UTILITY
#85 AUTOBOOT UTILITY
#86 COLUMN TEXT III (P)
#87 ARCHIVER III
#89 PROCALC DEC/HEX CONV.
#96 STATS & SORT ROUTINES
#97 MEMORY MANIPULATOR
#101 ENHANCED DISPLAY PACK.
#104 C99 COMPILER LIBRARY
#108 FULPLUS
#110 DISK + AID
#117 UNIVERSAL DISASSEMBLER
#119 RAG LINKER
#127 PIX GRAPHICS UTILITY
#243 OS/99 (Gram Device)
#251 PC TRANSFER TI/IBM (DD)
#254 NIBBLER/TURBO
#256 EXPLORER / DM1000
#260 TI FORTH (DISK ONLY)
(Add $12 for manual)
#17 TI FORTH DEMO
#116 TI FORTH TUTORIAL
#307 GRAPHICS CODE GENERATOR

BUSINESS, HOME, ACCOUNTING, DATA BASE, & WORDPROCESSING

#10 GOTHIC PRINTOUT (P)
#19 TI WRITER/MULTIPLAN UPGRADE
#20 ACCOUNTS RECEIVABLE
#21 DATA BASE DEMO
#23 WILL WRITER
#29 LABLE MAKER (P)
#36 STRICTLY BUSINESS
#56 SPREAD SHEET
#58 PR BASE
#59 GRAPH MAKER
#74 LABEL MAKER II (P)
#77 MICROdex 99
#81 HOME ACCOUNTING SYSTEM
#83 HOME APPLICATION PROGRAM.
#90 JET CHECKBOOK MANAGER
#92 HOUSHOLD INVENTORY
#109 TI WRITER MINI MANUAL
#112 INVOICE PACK
#113 LABEL MAKER III (P)
#129 CASH DRAW "P.O.S"
#130 THE ORGANIZER
#147 CALENDER-NOTPAD
#177 HOUSEHOLD BUDG. MANG. (M)
#30 H.B.M PRINT OUT (P)
#221 PERSONAL REAL ESTATE (M)
#249 MAPMAKER
#252 99 WRITER II (P)
#253 AMA MAIL LIST
#255 TRAK-A-CHECK
#257 DAILY DIARY
#258 EXER-LOG
#310 SELF HELP TAX CUT

TELECOMMUNICATIONS

#57 TELCO
#57D TELCO (for systems with DS drives)
#118 FAST TERM

EDUCATION & PERSONAL DEVLP.

#22 ASTROLOGY
#24 ENGINEERING CALCS.
#25 MEDICAL ALERT!
#27 KIDS LEARNING VOL. I
#31 MORSE CODE TRAINER
#37 LAPD COOKBOOK
#40 ARTIFICIAL INTELIG.
#54 ASTRONOMY
#66 HEBREW TYPEWRITER
#67 GENEALOGY
#71 KIDS LEARNING VOL. II
#95 WEATHER FORECASTER
#138 FIREHOUSE COOKBOOK
#142 TOUCH TYP. TUTOR (M)
#184 FACE MAKER (M)
#194 ST. NICK (M)
#195 TINY LOGO
#196 JOTTO
#199 MILLIKEN ADDITION (M)
#200 MILLIKEN DECIMALS (M)
#203 MILLIKEN FRACTIONS (M)
#204 MILLIKEN INTEGERS (M)
#205 MILLIKEN LAWS MATH (M)
#211 MIND CHALLENGERS (M)
#212 MINUS MISSION (M)
#213 MILLIKEN PERCENTS (M)
#214 STORY MACHINE (M)
#215 BEGINNING GRAMMAR (M)
#216 METEOR MULTIPLE. (M)
#217 HANGMAN (M)
#222 MUSIC MAKER (M)
#223 PHYSICAL FITNESS (M)
#225 ALIEN ADDITION (M)
#226 ALLIGATOR MIX (M)
#227 DEMOLITION DIVISION (M)
#228 DRAGON MIX (M)
#261 ASTRONOMY 2
#262 ASTRONOMY 3
#263 ASTRONOMY 4
#264 HIGHER MATH
#265 HIGHER MATH 2
#266 ELECTRONICS MATH
#268 CALCS. & CONVERTERS
#269 TRIGONOMETRY
#306 SPEAK & SPELL II (S)

GAMES

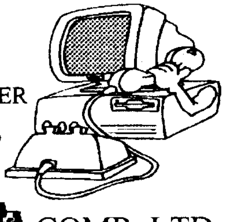
#2 WHEEL OF FORTUNE
#3 TI TREK
#8 LOOTO PICKER
#13 STRIP POHER "PG"
#26 R-RATED GAME
#33 CHECKERS & BACKGAMMON
#34 SOLITARE & SCRABBLE
#38 GREAT GAMES VOL. 1
#39 GREAT GAMES VOL. 2
#43 BEST OF BRITAIN VOL. 1
#45 BEST OF BRITAIN VOL. 2
#46 SUPER TRIVIA 99
#47 INFOCOM RAPID LOADER
#48 GHOSTMAN
#49 DEMON DESTROYER
#50 OH MUMMY!
#51 BERLIN WALL
#60 FREDDY
#61 THE MINE
#63 ASTROBLITZ & MAZOG
#64 MAJOR TOM & SPACE STATION
#65 PERFECT PUSH
#68 CHESS (SARGON)
#70 TI RUNNER II
#72 CERBERUS
#73 CRYPTO (Gram)
#82 CROSSWORD PUZZLES
#84 GALACTIC BATTLE/SPY ADV.
#88 AUSSIE GAMES VOL. I
#91 THE MAZE OF GROG
#94 GREAT TI GAMES VOL. 3
#98 DAYS/DOORS OF EDEN (ADV.)
#99 GREAT TI GAMES VOL. 4
#100 ASSAULT THE CITY (TOD.)
#102 COLOSSAL CAVES (ADV.)
#105 KINGS CASTLE
#106 QUEST "D&D ADV."
#121 SUPER YAHTZEE & WHEEL II
#122 ADULT ADV. (ADV.)
#123 GREAT TI GAMES VOL. 5
#124 GREAT TI GAMES VOL. 6
#125 BLACKJACK & POKER (M)
#126 VIDEO CHESS (M)
#128 TETRIS
#131 COMPUTER CRAPS
#135 ARTURUS
#137 CROSS-FIRE (M)
#139 MOON MINE (M)
#140 MASH (M)
#141 MOONSWEEPER (M)
#144 STAR TREK (M)
#145 BUCK ROGERS (M)
#148 KENO & SLOTS
#149 GREAT TI GAMES VOL. 7
#150 ULTIMATE TRIVIA
#176 AMAZING (M)
#178 DEMO DEDYTROYER (M)
#179 POPEY (M)
#180 QUEBERT (M)
#181 METOER BELT (M)
#182 BLASTO (M)
#183 CARWARS (M)
#185 SUPER FLY (M)
#186 SPACE BANDITS (M)
#188 KILLER CATAPILLER (M)
#190 BLACK HOLE (M)
#191 GREAT TI GAMES VOL. 8
#192 GREAT TI GAMES VOL. 9
#193 SPYS DEMISE (M)
#197 PRO TENNIS (M)
#198 TI INVADER/TOMSTONE (M)'S
#202 CONNECT FOUR M)
#206 SLTMOIDS (M)
#207 OTHELLO (M)
#208 PARSEC (M)
#209 SOCCER (M)
#210 SEWERMANIA (M)
#218 HUSTLE/FOOTBALL (M)
#219 CHISOLM TRAIL (M)
#220 ZERO ZAP (M)
#224 ATTACK (M)
#229 4/A FLYER (M)
#232 TUNNEL OF DOOM (M)
#233 MS ADVINTURE
#244 HOPPER (M)
#245 TREASURE ISLAND (M)
#247 STRIKE THREE (M)
#248 GREAT TI GAMES VOL. 10
#250 BARRAGE/SPOT SHOT (M),S
#291 MAZE MANIA
#292 THE CASTLE
#319 ARCADE SPECIAL

KEY

P = PRINTER REQ.
G = GRAPHEX REQ.
S = SPEECH REQ.
M = MODULE BACKUP
MM = MINI MEMORY
E/A = EDITOR ASSEMBLER
ADV= ADVENTURE REQ.
* = TITLE IS $4.95
(To cover added costs of disk and label)

DON'T WAIT FOR YOUR COMPUTER

TO CALL



ORDER BY PHONE

ORDERS *ONLY* (800) 846-3474
TECH HELP (818) 339-8924
(Call First) FAX (818) 858-2785

VISA & MASTER CARD EXCEPTED

ADD 5% SHIPPING {$5.00 MINIMUM}
OUTSIDE USA INCLUDE 10% SHIP

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

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

COMMENTS

TI and the Web

The new TI listserver is a big step forward for those who have Internet accounts and want to keep up with activities on the TI newsgroup. The listserver forwards all items posted on comp.ti.sys. The server is administered by Tom Wills and sponsored by the Southwest Ninety-Niners User Group.

The way it works, anyone who sends a request to the listserver can have the postings automatically forwarded in the form of email messages. How much easier can it get?

Check out the article on page 21 for more information.

And another article of interest to Web surfers, and those who'd like to try it, is a listing of World Wide Web home pages compiled by John Van Weelie. The article, starting on page 15, lists three pages of Internet addresses of interest to TI and Geneve users. Many are home pages maintained by TI users. As you will see, the list of home pages related to the TI continues to grow.

However, the success of the Web has led to the decline of BBSes. Over the years we've published listings of TI-related BBS sites. In the past year or so, these sites have declined, not so much because the Sysops lost interest so much as that users stopped calling on them, apparently turning to the Web instead. It makes sense economically. Unless the BBS you called was in your city, you'd have to pay long distance phone charges to go online. This gets expensive very quickly. With the Web, the call is local and the online charge is 50 cents and hour, or less.

Of course, the commercial online services are having the same problem as many of their long-time users abandon them for the Web. Even if users don't abandon the online services, they're reducing the amount of online time and that hits the services hard since for them time is money. You have to ask the question why someone would pay $3 to $5 per hour for an online service when you can surf the Web for 50 cents.

One place I've seen the decline in online services is GEnie, where the TI special interest group used to be robust. It's now a shadow of what it was as the new owners consolidate their operation. They've increased the cost of service while decreasing the level of service. The TI software archive, for example, has been slashed to about 40 percent of what it was six months ago. It's gotten so pathetic that I now think of the GEnie TI archive as an inadequate collection of software. I wouldn't be surprised if many TI users have better collections at home.

The way I look at it, the expansion of the Web means that the potential for diversity has increased. It may also mean that there won't be a single site you can go to to find everything you want about the TI. I don't know of any TI software archives on the Web that count 5,000 or more files, such as the old GEnie collection did. But with more and more home pages blossoming, the more likely that TI Web surfers will be able to find the information and software that they want. It just may take a little longer, but they'll be able to do it a lot cheaper.

In no way should BBS operators view this column as a shot at them. I hope BBSes thrive alongside the Web. After all, the more the merrier.

—JK

READER TO READER

Kate Burkard, treasurer/librarian for the Manasota TI99/4A User Group has written with a question. Her address is 2814 Hermitage Blvd., Venice, FL 34292:

I have two TI systems with CorComp Disk Controllers with two half-height double-sided drives in the Peripheral Expansion Box. I have been trying to hook up two half-height double-sided drives in their own TI Power Supply Box as Drives 3 and 4, but I've had no success. The drives work fine as 2 and 3 with a full-height single-drive system. But when I set the drives to 3 and 4 and try to attach them to the back of the controller card and access them, it disables Drives 1 and 2. I have tried these on both systems that I have and neither system works. I think maybe the cable isn't right even though it works on the full-height, single-drive setup.

I have checked with many articles I have on chaining disk drives, etc., and I can't figure out what I'm doing wrong. Could someone please help me?

Second item I need help with is Page Pro Composer. I have an Epson 24-pin printer and the line spacing needs to be changed. I made the change in Page Pro from an article I read, but I don't know what the change needs to be for Page Pro Composer, or how to do it.

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

COMPETITION COMPUTER 800-471-1600

(Nationwide & Canada ORDERS ONLY)

350 Marcella Way Millbrae, California 94030

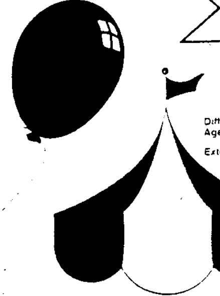
STILL NO EXTRA CHARGE FOR CREDIT CARD ORDERS
NO "HANDLING" CHARGES
NON-PROFIT SHIPPING CHARGES. YOU ONLY PAY WHAT WE PAY
WE ACCEPT: PERSONAL CHECKS(4 BUSINESS DAYS FOR CLEARANCE)
MONEY ORDERS
MASTERCARD AND VISA

MasterCard VISA

TI-99/4A™ CD ROM

CD ROM UPDATE: CAN NOW BE USED WITH PC99 STAGE 3A USING NEW PC99 UTILITY DLCONV.EXE! 5 ISSUE SUBSCRIPTIONS $100
SINGLE (120 MB) ISSUES NOW AVAILABLE $35⁰⁰

CELEBRATES ITS 1ST ANNIVERSARY



FUN HOUSE

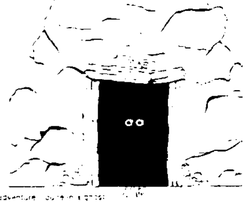
Difficulty: Advanced
Ages 12 to adult

Extended Basic Language
TI 99/4 TI 99/4A

In this adventure challenge after challenge will face you. Can you make it out of the mysterious fun house? Watch out! Only the best adventurers can get out of the room of mirrors. Sound effects galore. Lots of fun.

TIventures

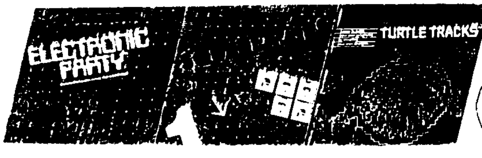
Miner 49'er



In this adventure you're in a ghost town that once was a prosperous mining town. Enter the mine; there remain many treasures. Who put those treasures there? Rumors of a monster in the mine scared off the people in the town and they left their treasures behind. Find the treasures and [illegible]. Now with sound effects.

Difficulty: Intermediate
Ages: 12 to adult

Extended Basic Language
TI 99/4 TI 99/4A

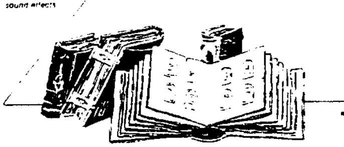


In Search Of: The Four Vedas

A party from your tribe went on an expedition. They never returned and now it's your turn. Go on! During this adventure you must find the ancient books of your tribe that were lost many years ago. They contain great knowledge of the past and of magic. There are some fairly easy obstacles and some extremely hard ones that require a lot of thinking. Now with sound effects.

Difficulty: Advanced
Ages: 12 to adult

Extended Basic Language
TI 99/4 TI 99/4A

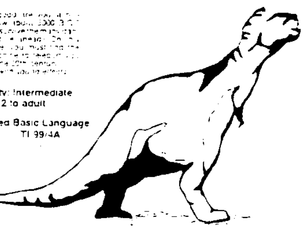


STONE AGE

How good are you at [illegible]? [illegible] 3000 B.C.? Can you survive the many dangers that lie ahead? On this adventure you must find the time machine to take you back to the 20th century. Now with sound effects.

Difficulty: Intermediate
Ages: 12 to adult

Extended Basic Language
TI 99/4 TI 99/4A



Order any game for only $5

Fireball

An arcade style action game which combines math practice with entertainment.

The students are in command of the last anti-meteor defer system on the planet Terra. As the meteors approach, a ma problem is displayed on the screen. A quick and correct answ is required to destroy the meteor. Incorrect answers allow t meteors to destroy sections of the last remaining city; the fa of Terra is in their hands.

The student can select the type and complexity (+, −, ×, of the problems and the speed of play before starting; a co stant display of the level of difficulty, points earned and pe centage of correct response is shown. A final summary of th information is presented at game s end. Level of difficulty automatically adjusted during play.

Grade levels 2-8; Special Education, all levels.

Turn The Water Off!

A set of 2 programs designed to provide customized spelling practice combined with an entertaining "Hang Man" style game: colorful graphics accompany a spelling drill which challenges students to save Simon Spellbinder by correctly spelling the "mystery word" before the overflowing sink fills the room with water.

Allows creation of an endless number of spelling word files. Easy to follow instructions, combined with a powerful filing system, guide the user in setting up, saving and editing word files for use with the game. This allows for an infinitely varied and exactly targeted file of spelling words, challenging students at any level. Files work on tape or disk and are very fast-loading.

For grade levels 2-6. Also included in the game is the option for students to input their own lists of spelling words, rather than use the customized file.

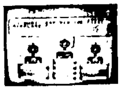
The game-style format provides a basis for identification of a state and subsequently, its capitol from pieces of information provided as hints. Player must then indicate geographical location by using Uncle Sam s hand to point to the state from the outline map of the U.S. displayed on the screen. Scoring is based on the number of hints and related tries to identify state/capitol and the time required to locate the state. Music and color graphics are used as motivation for correct answers.

The Everything Teacher

The Everything Teacher is a powerful combination of four dynamic, highly motivating games and a mini-authoring system in a user-friendly format. This system allows even non-programmers to quickly create customized educational games which will teach and/or review any information on any subject at any level.

All games feature excellent and noteworthy use of graphics and sound, as well as a complete summary of user performance. Players may select the game program of choice and incorporate an information file from cassette or disk. Complete instructions and suggestions for home and classroom use are included with all programs.

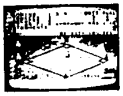
The four games included in this useful educational package are:



T.V. Sweepstakes

"Hello Ladies and Jellybeans ... I'm your host, Ferd Ferblebraster, and I'd like to welcome you to T.V. Sweepstakes!"

This fast-paced, two contestant game features a humorous announcer quizzing material in a T.V. game show format designed to encourage quick thinking and accurate response.



Baseball

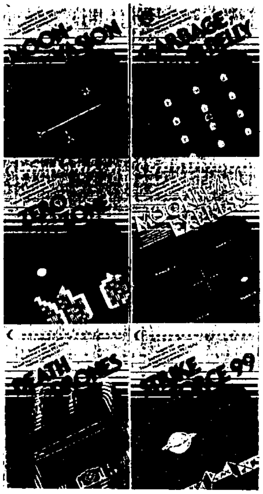
Takes you ... out to the old ball game ... (popcorn and peanuts not provided). Two players, or two teams take turns at bat as they answer questions to move around the bases and score runs. Separate game score and performance information is provided.

Space Patrol—Lost!

Commander, you and the ships under your command are lost in a hostile sector of space. Sensors have detected an incoming meteor storm ... Alternating rounds of questions and game play are highlighted by excellent color graphics and sound. Correct answers accumulate ... ammunition ... for meteor defenses. Both performance and game score are displayed at the end of each game.

Last Jellybean on Earth

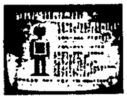
Life's tough when you're born a Jellybean in the land of the Horribly Hungry Munch Monsters. Rounds of game play and quiz material alternate in an arcade-style format to provide the setting for this ... sweet ... game accompanied by creative use of graphics and sound. The higher the score, the more ... tooth decay ... is earned for use in the struggle against the Munch Monsters.



INTELLESTAR LIFE SCIENCE SERIES

CELLS - The Building Blocks of Life

This three-part set is an introduction to Intellestar s Life Science Series. Extensive use of realistic, animated graphics and accompanying text combine to provide a clear insight to the innermost workings of life's building blocks, the cells.



CELLS, Part 1 explores the chemical nature of protoplasm as the basis of all life. The rudiments of cells, cell types and cellular movement are vividly portrayed in a high-quality visual format.

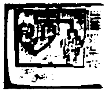
CELLS, Part 2 presents "a view from the inside" detailing cellular structures. Animated graphics present the material in a clear, concise form, leading to a concrete understanding of the inner workings of cells.

CELLS, Part 3 delivers a step-by-step "live action" model of cellular reproduction. The main stages of cellular division are demonstrated in a well-organized format, aided by realistic animation.

Aimed at ages ten to adult, this learning package will provide the "aspiring scientist" with a highly visual, well-defined view of life at a microscopic level.

Heart Attack

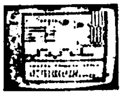
Age 11 Adu



Our heart is literally in your hands with the latest simulation/game in Intellestar s Life Science Series. Vividly displayed on your T.V. screen is a detailed working model of your heart and circulatory system.

Players learn by experimentation, the effects of twenty controlling mechanisms on oxygen levels, body temperature, heart rate, cardiac output and functioning of major blood vessels and major types of blood cells.

Inside Frankie Stien



An animated "living simulation" of cellular interaction between the systems of the body, this challenging program provides interaction necessary for real learning and understanding.

Player "becomes the brain" of little Frankie Stein, monitoring information in the blood and using the information to decide which body system must be stimulated or suppressed, keeping Frankie's body "in balance" and alive as long as possible. Various events, some correlating with age and other health factors, influence responses of certain body systems. Animated screen graphics vividly portray various bodily functions in an interrelating format, providing an in-depth understanding of system interdependence. Score is measured by Frankie's age: the longer he is kept alive, the higher the score.

Included with the program is complete documentation providing background information on bodily systems, a glossary of terms, complete operating instructions and material for classroom duplication.

Super Game Special!

FEEDBACK

Reader reports on UK TI meeting

I'm very pleased to hear our mag has gone bimonthly against probable closure; even so, this does effectively double its price and I wonder how long not only you but the user can sustain price increases such as this.

Anyway, since I wrote you last year I have again expanded my system with two 80-track drives and hopefully eliminated the problems with my Hard and Floppy Disk Controller and Seagate Hard drive. Our user group electronics engineer Ross Bennett did all the hard work of changing chips, testing and putting full-length heat sinks on my HFDC and also supplied me with two 80-track half-height floppy drives each holding 720K and one a 3.5-inch which runs very quietly. All this he put together at our user group Annual Group Meeting and Fair on May 18k 1996, St. John's Ambulance Center, Trinity Street, Darby, England. Our chairman as usual was Trevor Stevens.

It was all good fun, and about 10 attended, with demos from three TI machines plus my own, a transferring of files from an Amiga with hard disk to a TI, also an IBM-type showing TI programs on its own screen

D.H. Caine
Crewe, England

Books hard to find

I have spent a lot of time making use of the c99 tutorials that started in the 1987 MICROpendiums and have learned a lot.

One trouble with getting the basic books on programming (recommended in articles) is that the used book stores around here don't bother with computer books written before 1990. All the newer books are concerned with ANSI C or C++, which has many differences from c99. I have looked for the recommended book *C Primer Plus* but have found only *The NEW C Primer Plus*, printed in 1990.

P.C. Van Nordstrand
Seabrook, Texas

Program has error

I just received the July 1996 issue of MICROpendium, and, as usual, I tried out some of the programs. Unfortunately, there is an error in W. Leonard Taffs' program "Recurring Words" on page 28. In line 40, the printer is opened as #3, but in line 170 it is referred to as #2. The "ON ERROR 10" in line 100 sends you right back to line 10. A small problem and easy to fix, but it might cause some newcomers to the TI99 world a few bad moments.

Woodrow A. "Woody" Wilson
Southern California Computer Group
San Diego, California

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

EXTENDED BASIC

Calculating energy costs

By HARLEY RYAN

E-G/UTE is an Extended BASIC program that will calculate the days in each month's gas or electric bill. Also it will calculate the kilowatts or cubic feet used per day, along with the cost per day for each month.

When you start the program you will be asked if you need instructions. Press "Y" if you do need them or Enter if you do not want them. Next press either "E" or "G" depending on which utility you want to process. After that press "D" to save to a disk, "P" to print out the list, or Enter to output to the screen.

You now will need to enter the year of the bills you are working on. Follow the screen prompts to continue. If you make a mistake in any of the above entries, press "N" to return to the first month and re-enter them. If you intend to save your work to a disk or printer, you should enter a zero in front of each single digit number. This will produce a more legible printout.

After the beginning year is entered, the year will advance by one when needed. On each leap year one day will be added to give February 29 days. If you are saving this to a disk and then stop the program, you must give the new file a different name than the first file. You will be notified of this on screen. The program rounds off the numbers for KW/CF per day and cost per day. After a great many entries (35 + years), the calculations were off by only one on a few when compared to the utility companies' calculations.

There may be times under certain conditions that the year will not change to what it should be, although this did not happen to me. If it does, the list may be edited if it is on disk by any word processor program.

I have tried to make this as goof-proof as possible, but it is still possible to get an error that will make you restart the program where you left off. There are a great many variables in this program, so it was slow to load until I pre-scanned it. This speeded it up a lot. (If you place the pre-scan on line number one, it must be removed before running the program. If it remains, the program will crash.)

If you enter enough data, you will be amazed at how much the cost of electricity and gas has increased over the years. This gives you a good reason to see where you can save some energy. A new gas furnace and electric air conditioner made a world of difference on my bills.

If you enter enough data, you will be amazed at how much the cost of electricity and gas has increased over the years. This gives you a good reason to see where you can save some energy. A new gas furnace and electric air conditioner made a world of difference on my bills.

EG/UTE28

```
1 DATA 31,28,31,30,31,30,31,
31,30,31,30,31 !110
2 GOTO 100 :: CF$,D1$,D2$,E$
,F$,GC$,KW$,M1$,M2$,R$ :: C,
CFS,D,D1,D2,E,ES,G,GCS,J,K,K
W,M1,M2,S,TOT,V,X,Y :: CALL
CLEAR :: CALL COLOR :: CALL
KEY :: CALL SCREEN :: !@P- !
078
100 ! ELECTRIC and GAS BILL
PROGRAM,by H.Ryan of CONNI,A
pril 1996 !151
110 ON ERROR 750 !249
120 CALL CLEAR :: FOR C=0 TO
 14 :: CALL COLOR(C,16,1)::
NEXT C :: CALL SCREEN(5):: E
=0 :: G=0 :: J=0 :: V=1 :: X
=0 !122
130 IMAGE ####   ##/##/##/##
  ##     ##.##  ##.## !030
140 DISPLAY AT(1,3):"ELECTRI
C & GAS BILL PROGRAM" :: DIS
PLAY AT(4,2):"ELECT BILL> E\
": :" GAS BILL--> G/" !087
150 ACCEPT AT(5,16)BEEP VALI
DATE("EG")SIZE(1):R$ :: IF R
$="" THEN 150 :: IF R$="E" T
HEN GOSUB 730 ELSE GOSUB 740
 !248
160 ON WARNING NEXT :: DISPL
AY AT(4,5):"NEED INSTRUCTION
S ?": :TAB(12);"(Y/N)N" !219
170 ACCEPT AT(6,17)BEEP VALI
DATE("YyNn")SIZE(-1):R$ :: I
F R$="N" THEN 180 ELSE GOSUB
 580 !106
180 DISPLAY AT(6,3)ERASE ALL
:"SAVE TO DISK or PRINTER ?"
: :TAB(6);"PRESS 'D' FOR DIS
K":TAB(6);"PRESS 'P' FOR PRI
NTER" !049
190 DISPLAY AT(10,6):"PRESS
'N' FOR NEITHER": :TAB(13);"
N" !183
200 ACCEPT AT(12,13)BEEP VAL
IDATE("DNP")SIZE(-1):R$ :: I
F R$="D" THEN GOSUB 650 ELSE
 IF R$="P" THEN GOSUB 700 !0
52
210 DISPLAY AT(2,8)ERASE ALL
:"ENTER DATA" :: DISPLAY AT(
6,2):"FIRST YEAR>" !147
220 ACCEPT AT(6,14)BEEP VALI
DATE(DIGIT)SIZE(4):Y :: DISP
LAY AT(7,2):"1st MONTH->
  '0' TO END" !110
230 ACCEPT AT(7,14)BEEP VALI
DATE(DIGIT)SIZE(2):M1$ :: IF
 M1$="0" THEN 560 :: IF M1$=
"" THEN 230 :: M1=VAL(M1$)::
 IF M1<1 OR M1>12 THEN 230 !
204
240 FOR C=1 TO M1 :: READ D
:: NEXT C !110
250 IF Y/4=INT(Y/4)THEN IF M
1=2 THEN D=D+1 !206
260 DISPLAY AT(8,2):"1st DAY
--->" :: ACCEPT AT(8,14)BEEP
 VALIDATE(DIGIT)SIZE(2):D1$
:: IF D1$="" THEN 260 :: D1=
```

(See Page 8)

ENERGY —

(Continued from Page 7)

```
VAL(D1$):: IF D1<1 OR D1>31
THEN 260 !095
270 DISPLAY AT(9,2):"2nd MON
TH->" :: ACCEPT AT(9,14)BEEP
 VALIDATE(DIGIT)SIZE(2):M2$
:: IF M2$="" THEN 270 :: M2=
VAL(M2$):: IF M2<1 OR M2>12
THEN 270 !234
280 DISPLAY AT(10,2):"2nd DA
Y--->" :: ACCEPT AT(10,14)BE
EP VALIDATE(DIGIT)SIZE(2):D2
$ :: IF D2$="" THEN 280 :: D
2=VAL(D2$):: IF D2<1 OR D2>3
1 THEN 280 !205
290 IF G=1 THEN 320 !061
300 DISPLAY AT(11,2):"KWH US
ED-->" :: ACCEPT AT(11,14)BE
EP VALIDATE(DIGIT)SIZE(4):KW
$ :: IF KW$="" THEN 300 :: K
W=VAL(KW$)!188
310 DISPLAY AT(12,2):"ELECT
COST>" :: ACCEPT AT(12,14)BE
EP VALIDATE(DIGIT,".")SIZE(6
):E$ :: IF E$="" THEN 310 ::
 ES=VAL(E$):: GOTO 340 !169
320 DISPLAY AT(11,2):"CF USE
D--->" :: ACCEPT AT(11,14)BE
EP VALIDATE(DIGIT)SIZE(4):CF
$ :: IF CF$="" THEN 320 :: C
FS=VAL(CF$)!139
330 DISPLAY AT(12,2):"GAS CO
ST-->" :: ACCEPT AT(12,14)BE
EP VALIDATE(DIGIT,".")SIZE(6
):GC$ :: IF GC$="" THEN 330
:: GCS=VAL(GC$)!115
340 DISPLAY AT(14,4):"OK (Y/
N)> Y" :: ACCEPT AT(14,14)BE
EP VALIDATE("YN")SIZE(-1):R$
 :: IF R$="N" THEN RESTORE :
: GOTO 230 !228
350 IF M2-M1>1 THEN 360 ELSE
 380 !166
360 DISPLAY AT(18,1):" " ::
DISPLAY AT(18,3):"CANNOT PRO
CESS";M2-M1;"MONTHS" :: DISP
LAY AT(23,8):"PRESS ANY KEY"
 !195
370 CALL KEY(0,K,S):: IF S<1
 THEN 370 :: DISPLAY AT(18,1
):" " :: DISPLAY AT(23,1):"
" :: RESTORE :: GOTO 230 !12
6
380 IF M1=M2 THEN TOT=D2-D1
ELSE TOT=D-D1+D2 !057
390 KW=KW/TOT :: ES=ES/TOT !
232
400 CFS=CFS/TOT :: GCS=GCS/T
OT !230
410 DISPLAY AT(16,2):USING "
TOTAL/DAYS>     ##":TOT :: IF
 G=1 THEN 440 !035
420 DISPLAY AT(17,2):USING "
K-WATT/DAY> ##.##":KW !141
430 DISPLAY AT(18,2):USING "
COST/DAY--> ##.##":ES :: GOT
O 460 !253
440 DISPLAY AT(17,2):USING "
CF/DAY----> ##.##":CFS !076
450 DISPLAY AT(18,2):USING "
COST/DAY--> ##.##":GCS !164
460 DISPLAY AT(23,8):"PRESS
ANY KEY" !009
470 CALL KEY(0,K,S):: IF S<1
 THEN 470 !029
480 DISPLAY AT(16,1):" " ::
DISPLAY AT(17,1):" " :: DISP
LAY AT(18,1):" " :: DISPLAY
AT(23,1):"" !078
490 IF V=1 THEN 510 !010
500 RESTORE :: IF M1>M2 AND
D-D1<D2 THEN Y=Y+1 ELSE IF M
1=1 AND M2=2 AND D-D1>D2 THE
N Y=Y+1 !018
510 IF J=1 AND E=1 THEN PRIN
T #1,USING 130:Y,M1$,D1$,M2$
,D2$,TOT,KW,ES !008
520 IF J=1 AND G=1 THEN PRIN
T #1,USING 130:Y,M1$,D1$,M2$
,D2$,TOT,CFS,GCS !137
530 IF X=1 AND E=1 THEN PRIN
T #2,USING 130:Y,M1$,D1$,M2$
,D2$,TOT,KW,ES !023
540 IF X=1 AND G=1 THEN PRIN
T #2,USING 130:Y,M1$,D1$,M2$
,D2$,TOT,CFS,GCS !152
550 DISPLAY AT(6,13):Y :: DI
SPLAY AT(22,1):" " :: V=0 ::
 RESTORE :: GOTO 230 !066
560 IF J=1 THEN CLOSE #1 ::
IF X=1 THEN CLOSE #2 !043
570 CALL CLEAR :: END !222
580 DISPLAY AT(1,1)ERASE ALL
:"This program will calculat
e":"the days per  month in e
ach":"gas or electric bill.A
lso it" !072
590 DISPLAY AT(4,1):"will ca
lculate the Kilowatts":"or C
ubic  feet used  per day":"L
astly,the cost per day will"
:"be calculated and  shown
onthe screen" !114
600 DISPLAY AT(8,12):"and sa
ved to disk":"or a printer i
f so desired.":"The year wil
l advance by one":"after the
 first years entry." !165
610 DISPLAY AT(12,1):"Every
leap year,one day willbe add
ed to February. If youare  s
aving the  output to a":"dis
k  and the  program stops" !
031
620 DISPLAY AT(16,1):"you MU
ST give the  next file":"a
different name than  the":"f
irst file. To get a neater p
rintout,enter a zero before"
 !050
630 DISPLAY AT(20,1):"each s
ingle digit entry." :: DISPL
AY AT(23,6):"PRESS ANY KEY"
!217
640 CALL KEY(0,K,S):: IF S<1
 THEN 640 :: RETURN !210
650 DISPLAY AT(19,1)ERASE AL
L:"Give your output file a n
amethat is different than  t
he first filename." !114
660 DISPLAY AT(10,3):"ENTER
OUTPUT FILENAME": :TAB(6);"D
SK " :: ACCEPT AT(12,9)BEEP
SIZE(12):F$ :: IF F$="" THEN
 660 !168
670 IF E=1 THEN OPEN #1:"DSK
"&F$,OUTPUT,DISPLAY ,VARIABL
E 80 :: J=1 :: PRINT #1:" YE
AR       DATE       TOT    KW/D
AY  COST/DAY" :: PRINT #1 !2
41
680 IF G=1 THEN OPEN #1:"DSK
"&F$,OUTPUT,DISPLAY ,VARIABL
E 80 :: J=1 :: PRINT #1:" YE
AR       DATE       TOT    CF/D
AY  COST/DAY" :: PRINT #1 !2
18
690 RETURN !136
700 IF E=1 THEN DISPLAY AT(2
2,6)ERASE ALL:"TURN ON PRINT
ER!" :: OPEN #2:"PIO",OUTPUT
 :: PRINT #2:" YEAR      DAT
E     TOT    KW/DAY  COST/DA
Y" !068
```

(See Page 10)

THE CHICAGO TI USERS GROUP

presents the

14th Annual Chicago TI International World Faire

supporting TI 994A and 9640

November 9, 1996

EVANSTON PUBLIC LIBRARY
Evanston, Illinois
(corner Church & Orrington)
9:00 a.m. - 4:00 p.m.
Vendors-demonstrations-Seminars-User Groups
door prize drawings

----

for further information contact: **Hal Shanafield** (847)864-8644

Hotel Information for the Official Hotel

Skokie Howard Johnson
9333 Skokie Blvd
Skokie Illinois 60077

Standard Queen	$68
Standard Twin Double	$80

Tel: 1-800-1GO-HOJO (1-800-446-4656) mention the TI Faire

Includes

Hot Buffet Breakfast

Indoor Pool, Whirlpool, Sauna Fitness Center

Free Shuttle Service to Faire Site in Evanston

THE ART OF ASSEMBLY — PART 60

A Moving Experience

By BRUCE HARRISON

This month we mark yet another year's worth of columns with what we consider a genuine "breakthrough" in Assembly programs. We've overcome another of those "can't be done" things from the Editor/Assembler manual. The manual says we can use sprites in bit-map mode, but not their automatic motion. Well, we've done that!

THE FIRST STEPS

We started this effort right where last month's column left off, in the Enhanced Graphics Mode (a.k.a. half-bit-map). At first, we tried using TI's built-in automatic sprite motion, but this soon proved troublesome, because the routine in the console ROM which provides that service is hard-coded to look for the Sprite Attribute Table at >0300 and the Sprite Motion Table at >0780 in VDP RAM. Both of those locations are within the area we were using for our Pattern Descriptor Table. We could get automatic motion for our sprites, but only by giving up the use of 32 character definitions, 16 of them at >0300, and another 16 at >0780. That, we felt, was too great a price to pay, since many programmers would want to be able to use all 256 character definitions at will. There just had to be another way.

THE BETTER WAY

Our solution was to ignore the ROM's Sprite Motion service, and instead provide our own through a User Interrupt. To do that, we disassembled TI's ROM code, then rewrote it so that it could use the Sprite Attribute Table and Sprite Motion Table that we'd assigned in our own program. This way, we're able to put the Attribute and Motion Tables at any convenient location in VDP RAM. In the half-bit-map case, we set the Pattern Table for sprites at >0800. When we start up, the standard character definitions are already in place at >800, but we will redefine some of them for demo purposes.

To make sure that the ROM sprite motion is disabled, we set the byte at >837A to zero before putting our sprites in place. We use our own memory location (at label SPMOT) to indicate how many sprites are to be in motion.

After writing the data for our four sprites into VDP RAM through subroutine SPRITE, we activate the motion by placing the address of USRINT at >83C4. After that, we enter a "key loop" that includes LIMI 2 and LIMI 0, and the user interrupt routine takes care of the desired motion for our sprites.

The interrupt routine (at label USRINT) performs a BLWP to the vector at SPRINT. The code section starts at label SPRVEL, and there is a separate workspace at label INTWS just for that code. This saved us some memory by allowing R1, R2, and R3 of that WS to be pre-loaded with addresses. In this particular case, we've set the Sprite Attribute Table at >1000 and the Sprite Motion Table at >1080. Other locations in VDP can be used for those, but these locations are handy. Doing it this way leaves all of VDP RAM from >2800 onward available for other things.

IN THE SIDEBAR

Once again the sidebar is a complete program. It's very similar in the early parts to last month's sidebar, but there are two added EQUates for the ATTLST (Sprite Attribute Table) and the MOTBL (Sprite Motion Table) Otherwise it's pretty much the same down to six lines before label CLRLP. There we put a solid block character definition into the 0th and 255th spots in the Sprite Definition space. These will be used to show that all 256 Sprite Character Patterns are available for our use. A little later we display another string on the screen that says "With Sprites in Motion." Just to illustrate that it can be done, we've put the sprites in magnified form by setting R0 to >01E1 and then doing a VWTR. This way each sprite will use just one character, but will be twice the height and width of the normal screen characters.

Now the program uses the subroutine SPRITE four times. First it places a sprite at dot-row 148 and dot-column 120 that's a red "A" moving upward at speed 10 and to the right at speed 20. Next it sets Sprite 1 at the same position, but this time it's a yellow "C" moving upward at speed 10 and to the right at speed 40. Sprites 2 and 3 are solid blocks using the characters 0 and 255,and these move downward at speed 20. Number 2 is green, and 3 is magenta.

As soon as the data for these sprites has been sent to VDP by the subroutine, the main program enables the User Interrupt by placing the address from INTLOC at >83C4. From here on, we just sit at the key loop and wait for a key press. While waiting, you get to see these four sprites moving around the screen just as if we were in regular graphics mode. Of course you can tell from the individually colored characters on the screen that we're actually in half

(See Page 11)

ENERGY —

(Continued from Page 8)

```
710 IF G=1 THEN DISPLAY AT(2
2,6)ERASE ALL:"TURN ON PRINT
ER!" :: OPEN #2:"PIO",OUTPUT
 :: PRINT #2:" YEAR      DAT
E     TOT    CF/DAY  COST/DA
Y" !045
720 PRINT #2 :: X=1 :: RETUR
N !048
730 DISPLAY AT(2,3)ERASE ALL
:"ELECTRIC BILL PROGRAM" ::
G=0 :: E=1 :: RETURN !228
740 DISPLAY AT(2,6)ERASE ALL
:"GAS BILL PROGRAM" :: E=0 :
: G=1 :: RETURN !114
750 DISPLAY AT(20,1):"ERROR
CODE ";C;" HAS OCCURED": :TA
B(6);"PRESS ANY KEY" !219
759 !@P+ !062
760 CALL KEY(0,K,S):: IF S<1
 THEN 760 :: GOTO 210 !227
```

THE ART OF ASSEMBLY —

(Continued from Page 10)

bit-map, or Enhanced Graphics mode. Given a key press, we reset everything to "normal", kill the User Interrupt by CLR @>83C4, then exit to the GPL Interpreter. That puts us at the "PRESS ENTER TO CONTINUE" prompt. For those of you who get MICROpendium on disk, we've included the Object file SIDE60/O, so you can try this out immediately.

THAT USER INTERRUPT

As we said, the code in our User Interrupt, starting at label SPRVEL, was derived partly from disassembly of the ROM code starting at >095C and going through >09E8. This required extensive modification to get out the hard coded addresses and set up to use the Attribute List and Motion Table where we'd put them. It's written so that as long as the Equates are correct, and the correct value has been sent to VDP register 5, the motion of sprites will proceed without trouble. The main program has to be executing LIMI 2 and LIMI 0 for the sprites to remain in motion, but that's the same in normal graphics mode.

We start by pointing R9 and R10 at the Motion Table and Attribute List, respectively. Next we get the number of sprites that are in motion into R12. If that is zero, we skip the rest of the subroutine. In this particular case, the number at SPMOT is 4, so sprites 0, 1, 2, and 3 are in motion. The number in motion is right-justified in R12. Now the address from R9 is placed in R8, then is set into place as the VDP address in the Motion Table. The routine reads the Y velocity into R4 and the X velocity into R6. It reads the first auxiliary byte into R5 and the second into R7. Each of these four registers then gets shifted right by four bits, effectively dividing the numbers by 16. It adds R4 to R5 and R6 to R7.

Now the routine shifts over to the Attribute Table, and reads the Y position of the sprite into R4 and the X position into R6. R5, which contains the sum of the Y velocity and Y auxiliary value, each divided by 16, is added to the position in R4. The vertical is checked by comparing R4 to certain limit values, and its value is adjusted if necessary. This allows a sprite to wrap around correctly from top to bottom and vice versa.

Now the routine writes the new position bytes to the attribute table, and the new auxiliary values to the Motion Table. That finishes its work for one sprite. Now we use a trick code to add four to both R9 and R10, so that these point to the locations in the tables for the next sprite. The trick is written C *R9+,*R9+. This just causes R9 to be incremented by two twice, and the results of the comparison itself are ignored. Finally, we DECrement R12, and go back to service the next sprite if it's not zero. When R12 becomes zero, we Return with Workspace Pointer by RTWP. That sends us back to the second instruction at USRINT, which returns to the interrupt handler.

WORDS OF CAUTION

If your program accesses files during execution, you'll need to be very careful about the placement of the various tables in VDP RAM. You might even need to reload them after file accesses, depending where your PABs and buffers are placed. Also, before using a file, you'll need to put back the six bytes we stashed at ANYKEY+1 into the location pointed to by >8370, or else file access won't happen. In earlier columns, we've covered how to execute a CALL FILES(1) from Assembly, and you may want to do that. That will mean your "file" space reservation will move to >3BE4 (>8370 will contain >3EB3).

The process we've shown here allows for the use of up to 32 sprites, each having a choice of any of 256 character patterns. In our own work, we've never found it necessary to use that many of either sprites or character patterns, but you should have the maximum capability available, so we've arranged it that way for you.

This month's column is being kept short, because the sidebar is so long. This new capability may be the breakthrough you've all been waiting for. Next month, we'll do moving sprites in full bitmap mode, again with a complete program for you to experiment with. See you then.

SIDEBAR60

```
       DEF  START
       REF  VMBW,VWTR,VSBW,VMBR
       REF  KSCAN
STATUS EQU  >837C
SCRORG EQU  >1800          HALF BIT SCREEN
MOTBL  EQU  >1080          SPRITE MOTION TABLE
ATTLST EQU  >1000          SPRITE ATTRIBUTE TABLE
START  LWPI WS             LOAD OUR WORKSPACE
       LI   R0,>380        POINT AT COLOR TABLE
       LI   R1,SAVCLR      AND AT STORAGE SPACE
       LI   R2,32          32 BYTES TO GET
       BLWP @VMBR          READ COLOR TABLE INTO STORAGE
       MOV  @>8370,R0      GET VDP ADDR FROM >8370
       LI   R1,ANYKEY+1    POINT AT STORAGE BUFFER
       LI   R2,6           SIX BYTES TO READ
       BLWP @VMBR          READ THOSE INTO BUFFER
       LI   R0,>800        POINT AT CHARACTER TABLE
       LI   R1,CHRTBL      AND AT BUFFER STORAGE
       LI   R2,128*8       128 CHARACTER DEFINITIONS
       BLWP @VMBR          STASH CHARACTER DEFS
       A    R2,R1          MOVE TO SECOND HALF CHRTBL
       BLWP @VMBR          DUPLICATE CHARS IN SECOND HALF
       BL   @SETHB         SET TO HALF-BITMAP
       CLR  R0             START OF CHAR DEF TABLE
       LI   R1,CHRTBL      THE SAVED CHARACTERS
       LI   R2,256*8       ALL 256 DEFINITIONS
       BLWP @VMBW          WRITE THOSE
       LI   R0,>800        SPRITE PATTERN TABLE
       LI   R1,BLOCK       SOLID BLOCK PATTERN
       LI   R2,8           EIGHT BYTES
       BLWP @VMBW          WRITE THAT TO SPRITE CHAR 0
       LI   R0,255*8+>800 LAST PLACE IN SPRITE PAT. TBL.
       BLWP @VMBW          WRITE BLOCK TO SPRITE CHAR 255
       LI   R0,SCRORG      POINT AT SCREEN ORIGIN
       LI   R1,>2000       SPACE IN LB R1
       LI   R2,768         768 CHARS IN SCREEN
CLRLP  BLWP @VSBW          WRITE A SPACE
       INC  R0             NEXT SPOT
       DEC  R2             DEC COUNT
       JNE  CLRLP          REPEAT IF NOT ZERO
```

(See Page 12)

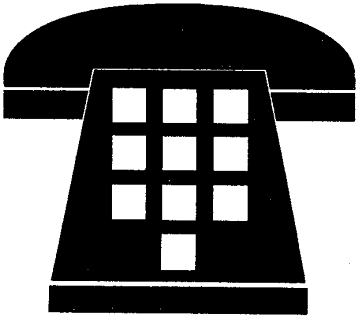
THE ART OF ASSEMBLY —

(Continued from Page 11)

```
       LI   R0,>2000     COLOR TABLE
       LI   R1,>4F00     BLUE ON WHITE
       LI   R2,>400      HALF OF COLOR TABLE
COLSET BLWP @VSBW         WRITE A BYTE
       INC  R0           NEXT POSITION
       DEC  R2           DEC COUNT
       JNE  COLSET        REPEAT IF NOT ZERO
       LI   R1,>F400      WHITE ON BLUE
       LI   R2,>400       2ND HALF OF TABLE
COLSE2 BLWP @VSBW         WRITE
       INC  R0           NEXT BYTE
       DEC  R2           DEC COUNT
       JNE  COLSE2        LOOP IF NOT ZERO
       LI   R0,>1E0       UNBLANK SCREEN
       BLWP @VWTR         BY VWTR
       LI   R0,SCRORG     POINT AT >1800
       LI   R4,2         TWO CYCLES
WRTOM  LI   R1,>2100      START WITH CHAR 33 (!)
       LI   R2,127-33     ALL DEFINED CHARS THRU 126 (~)
WRTCH  BLWP @VSBW         WRITE ONE
       AI   R1,>100       NEXT CHAR
       INC  R0           NEXT SCREEN LOCATION
       DEC  R2           DEC COUNT
       JNE  WRTCH        RPT TIL ZERO
       LI   R1,>A100      SET FOR SECOND SET'S (!)
       LI   R2,127-33     SAME NUMBER
       AI   R0,66        DOWN 2, OVER 2
WRTCH2 BLWP @VSBW         WRITE ONE
       AI   R1,>100       NEXT CHAR
       INC  R0           NEXT SCREEN LOCATION
       DEC  R2           DEC COUNT
       JNE  WRTCH2        RPT TIL ZERO
       AI   R0,66        DOWN 2, OVER 2
       DEC  R4           DEC R4 COUNT
       JNE  WRTOM        ANOTHER IF NOT ZERO
       LI   R0,33*8+>2000  COLOR FOR FIRST !
```

```
       LI   R1,NEWCOL     NEW COLOR SCHEME (WHITE ON
GREEN)
       LI   R2,8         EIGHT BYTES
       BLWP @VMBW         RE-COLOR THE !
       LI   R0,48*8+>2000 POINT AT ZERO CHARACTER'S COL
OR
       BLWP @VMBW         RE-COLOR THE 0
       LI   R0,57*8+>2000 POINT AT NINE CHAR'S COLOR
       LI   R1,RBOW       RAINBOW
       BLWP @VMBW         RE-COLOR THE 9
       LI   R0,122+128*8+>2000 SECOND SET'S L.C. Z COLO
       LI   R1,MAG        MAGENTA ON WHITE
       BLWP @VMBW         COLOR THAT
       LI   R3,HBSTR      TITLE STRING
       LI   R4,19         19 CHARACTERS
       LI   R2,8          8 BYTES PER DEF
       CLR  R13           START AT 0
LEGLP  MOVB *R3+,R1        GET A CHARACTER FROM STRING
       SRL  R1,8          RT. JUSTIFY
       SLA  R1,3          MULT. BY 8
       AI   R1,CHRTBL      ADD TABLE OFFSET
       MOV  R13,R0         GET R13 INTO R0
       BLWP @VMBW          WRITE CHAR DEF
       LI   R1,RBOW        RAINBOW COLOR
       AI   R0,>2000       COLOR TBL OFFSET
       BLWP @VMBW          WRITE COLORS
       A    R2,R13         ADD 8 TO R13
       DEC  R4            DEC CHAR COUNT
       JNE  LEGLP          RPT IF NOT 0
       LI   R0,20*32+6+SCRORG ROW 21, COL 7
       CLR  R1            ZERO IN R1
       LI   R2,19          19 CHARACTERS
LEGWRT BLWP @VSBW          WRITE A CHAR
       AI   R1,>100        INC HIGH BYTE R1
       INC  R0            NEXT SPOT
       DEC  R2            DEC CNT
```

(See Page 13)



THE ART OF ASSEMBLY —

(Continued from Page 12)

```
      JNE  LEGWRT        BACK IF NOT 0
      LI   R0,22*32+5+SCRORG ROW 23, COL 6
      LI   R1,SIMSTR      "SPRITES IN MOTION"
      LI   R2,22         22 CHARS
      BLWP @VMBW         WRITE THAT
      LI   R0,>01E1       ENLARGED SPRITES
      BLWP @VWTR          TO VDP REG 1
      LI   R1,>0400       FOUR SPRITES MOVING
      MOVB R1,@SPMOT       AT SPMOTION
      BL   @SPRITE        SET SPRITE
      DATA 0             NUMBER ZERO
      BYTE 148,120,65,6,-10,20  PARAMETERS (A RED "A")
      BL   @SPRITE        SET SPRITE
      DATA 1             NUMBER ONE
      BYTE 148,120,67,10,-10,-40 PARAMETERS (A YELLOW
"C")
      BL   @SPRITE        SET SPRITE
      DATA 2             NUMBER TWO
      BYTE 148,120,0,3,20,20  PARAMETERS (GREEN BLOCK)
      BL   @SPRITE        SET SPRITE
      DATA 3             NUMBER THREE
      BYTE 148,120,255,13,20,-40 PARAMETERS (MAGENTA
BLOCK)
      MOV  @INTLOC,@>83C4 ENABLE USER INTERRUPT
KEY   BLWP @KSCAN         SCAN KEYBOARD
      LIMI 2             ALLOW INTS
      LIMI 0             STOP INTS
      CB   @ANYKEY,@STATUS KEY PRESSED?
      JNE  KEY           IF NOT, RE-SCAN
      BL   @SETGM         RESET TO GRAPHICS MODE
      LI   R0,>1E0        UNBLANK SCREEN
      BLWP @VWTR          BY VWTR
      CLR  @>83C4         DISABLE USER INTERRUPT
      LWPI >83E0          GPL WORKSPACE
      B    @>6A          TO GPL INTERPRETER
SETHB LI   R0,>01A0       BLANK SCREEN
      BLWP @VWTR          BY VWTR
      LI   R0,>206        SET TO WRITE VDP REGISTER 2
      BLWP @VWTR          SIT TO >1800  (SCREEN IMAGE
TABLE)
      LI   R0,>400        SET TO WRITE TO VDP REG. 4
      BLWP @VWTR          PATTERN TABLE - SPECIAL VALUE
      LI   R0,>39F        SET TO WRITE TO VDP REG 3
      BLWP @VWTR          COLOR TABLE -SPECIAL VALUE
      LI   R0,>601        SET TO WRITE VDP REG 6
      BLWP @VWTR          Sprite descriptor table to >0800
      LI   R0,>520        SET TO WRITE VDP REG 5
      BLWP @VWTR          Sprite attribute table to >1000
      LI   R0,2           SET R0 TO WRITE 2 TO VDP REGISTER
ZERO
      BLWP @VWTR          SET TO M3 MODE (BIT MAP)
      CLR  R1            CLEAR R1
      MOVB R1,@>837A       NO SPRITES IN MOTION
      LI   R1,>D000        SPRITE DELETE
      LI   R0,>1000        AT ATTRIBUTE TABLE
      BLWP @VSBW          WRITE THAT
```

```
      RT                 RETURN
SETGM LI   R0,>1A0        SET TO WRITE VDP REG 1 (BLANK
SCREEN)
      BLWP @VWTR          WRITE
      LI   R0,>200        SET TO WRITE VDP REG 2
      BLWP @VWTR          WRITE
      LI   R0,>401        SET TO WRITE VDP REG 4
      BLWP @VWTR          WRITE
      LI   R0,>30E        VDP REG 3
      BLWP @VWTR          WRITE
      LI   R0,>600        VDP REG 6
      BLWP @VWTR          WRITE
      LI   R0,>506        VDP REG 5
      BLWP @VWTR          WRITE
      LI   R0,>380        POINT AT COLOR TABLE
      LI   R1,SAVCLR      AND AT SAVED COLOR DATA
      LI   R2,32          32 BYTES
      BLWP @VMBW          WRITE THE COLOR TABLE BACK
      LI   R0,>800        POINT AT GRAPHICS CHAR TABLE
      LI   R1,CHRTBL      AND AT STORED CHARACTER DATA
      LI   R2,256*8       256 CHARACTERS
      BLWP @VMBW          WRITE CHARACTER DEFS BACK
      LI   R1,>2000       SPACE IN LB R1
      LI   R2,768         768 CHARS
      CLR  R0            SCREEN ORIGIN
CLRGM BLWP @VSBW          WRITE ONE
      INC  R0            NEXT SPOT
      DEC  R2            DEC COUNT
      JNE  CLRGM          NOT 0, RPT
      MOV  @>8370,R0       ADDRESS FOR FILE STUFF
      LI   R1,ANYKEY+1     SAVED DATA
      LI   R2,6           SIX BYTES
      BLWP @VMBW          WRITE
      CLR  R0            PREP TO WRITE VDP REG 0
      BLWP @VWTR          WRITE THAT TO REMOVE BIT MAP
      RT                 RETURN
SPRITE LI   R0,ATTLST      POINT AT ATTR. TABLE
      MOV  *R11+,R3       GET SPRITE #
      SLA  R3,2           MULTIPLY BY 4
      A    R3,R0          ADD TO R0
      LI   R2,4           4 BYTES DATA
SPLP   MOVB *R11+,R1       GET A BYTE
      BLWP @VSBW          WRITE THAT
      INC  R0            NEXT BYTE
      DEC  R2            DEC COUNT
      JNE  SPLP           RPT IF NOT 0
      LI   R1,>D000        "DELETE" BYTE
      BLWP @VSBW          WRITE THAT
      LI   R0,MOTBL        POINT AT MOTION TABLE
      A    R3,R0          ADD NUMBER X 4
      MOVB *R11+,R1        GET Y VELOCITY
      BLWP @VSBW          WRITE THAT
      INC  R0            NEXT BYTE
      MOVB *R11+,R1        GET X VELOCITY
      BLWP @VSBW          WRITE THAT
      CLR  R1            ZERO IN R1
```

(See Page 14)

THE ART OF ASSEMBLY —

(Continued from Page 13)

```
       INC  R0          FIRST AUX BYTE
       BLWP @VSBW       WRITE 0 TO 1ST AUX BYTE
       INC  R0          SECOND AUX BYTE
       BLWP @VSBW       WRITE 0 TO SECOND AUX BYTE
       RT               RETURN
USRINT BLWP @SPRINT     USE BLWP TO VECTOR
       RT               THEN RETURN
SPRINT DATA INTWS,SPRVEL WORKSPACE AND CODE
SPRVEL LI   R9,MOTBL    POINT AT MOTION TBL
       LI   R10,ATTLST  AND ATTR. TABLE
       MOVB @SPMOT,R12  HOW MANY IN MOTION?
       JEQ  INTEX       IF ZERO, EXIT
       SRL  R12,8       RIGHT-JUSTIFY
STR8   MOV  R9,R8       MOTION TABLE
       MOVB @INTWS+17,*R1 LOW BYTE R8 TO VDP ADDR
       MOVB R8,*R1      HIGH BYTE R8 TO VDP ADDR
       CLR  R4          CLEAR REG 4
       MOVB *R2,R4      GET Y VELOCITY FROM TABLE
       CLR  R6          CLEAR REG 6
       MOVB *R2,R6      GET X VELOCITY FROM TABLE
       SRA  R4,4        SHIFT RIGHT 4 BITS WITH SIGN
       MOVB *R2,R5      GET AUX DATA BYTE
       SRA  R5,4        SHIFT RIGHT WITH SIGN
       A    R4,R5       ADD R4 TO R5
       MOVB *R2,R7      GET 2ND AUX BYTE
       SRA  R6,4        SHIFT RIGHT 4 WITH SIGN
       SRA  R7,4        SAME FOR R7
       A    R6,R7       ADD R6 TO R7
       MOV  R10,R8      ATTRIBUTE ADDR TO R8
       MOVB @INTWS+17,*R1 LOW BYTE R8 TO VDP ADDR
       MOVB R8,*R1      HIGH BYTE R8 TO VDP
       CLR  R4          CLEAR REG 4
       MOVB *R2,R4      Y POSITION TO R4
       A    R5,R4       ADD R5 TO R4
       CI   R4,>C0FF    COMPARE TO LIMIT
       JLE  AC          JUMP IF LOW OR EQUAL
       CI   R4,>E000    COMPARE TO >E000
       JH   AC          JUMP IF HIGH
       MOV  R5,R5       MOVE R5 TO ITSELF
       JGT  AD          IF POSITIVE, JUMP
       AI   R4,>C000    ADD >C000 TO R4
AD     AI   R4,>2000    ADD >2000 TO R4
AC     CLR  R6          CLEAR REG 6
       MOVB *R2,R6      X POSITION TO R6
       A    R7,R6       ADD R7
       ORI  R8,>4000    SET >4000 BIT IN R8 (WRITING)
       MOVB @INTWS+17,*R1 LOW BYTE R8 TO VDP ADDR
       MOVB R8,*R1      HIGH BYTE R8 TO VDP ADDR
       MOVB R4,*R3      WRITE NEW Y POSITION
       MOVB R6,*R3      WRITE NEW X POSITION
       MOV  R9,R8       GET MOTION ADDR
       INCT R8          ADD TWO FOR AUX BYTE
       ORI  R8,>4000    SET FOR WRITE
       MOVB @INTWS+17,*R1 LOW BYTE R8 TO VDP ADDR
       MOVB R8,*R1      HIGH BYTE R8
       SWPB R5          SWAP R5
       SRL  R5,4        SHIFT R5 LEFT 4 BITS
       MOVB R5,*R3      WRITE NEW AUX BYTE #1
       SWPB R7          SWAP R7
       SRL  R7,4        SHIFT LEFT 4
       MOVB R7,*R3      WRITE NEW AUX BYTE #2
       C    *R9+,*R9+   ADD 4 TO R9
       C    *R10+,*R10+ ADD 4 TO R10
       DEC  R12         DEC COUNT IN R12
       JGT  STR8        IF NOT ZERO, DO ANOTHER SPRITE
INTEX  RTWP             RETURN WITH WORKSPACE POINTER
SPMOT  BYTE 0           NUMBER OF SPRITES IN MOTION
INTLOC DATA USRINT      ADDRESS OF USER INTERRUPT
CHRTBL BSS  8*256       CHAR DEF STORAGE
SAVCLR BSS  32          COLOR TBL STORAGE
WS     BSS  32          OUR WORKSPACE
INTWS  BSS  2           USER INTERRUPT WORKSPACE R0
       DATA >8C02,>8800,>8C00  PRE-LOADED R1, R2 AND R3
       BSS  24          R4 THRU R15
NEWCOL DATA >FCFC,>FCFC,>FCFC,>FCFC WHITE ON GREEN
MAG    DATA >DFDF,>DFDF,>DFDF,>DFDF MAGENTA ON WHITE
RBOW   DATA >F6F8,>F9F3,>F2FC,>F5F4 "RAINBOW" COLORS
BLOCK  DATA >FFFF,>FFFF,>FFFF,>FFFF BLOCK CHARACTER
HBSTR  TEXT ' HALF-BIT MAP MODE ' MESSAGE 1
SIMSTR TEXT 'With Sprites in Motion' MESSAGE 2
ANYKEY BYTE 32          BYTE FOR COMPARISON
       BSS  6           STORAGE FOR FILE BYTES
       END
```

World Wide Web

A listing of TI and other home pages, FTP sites

The following list of World Wide Web home pages and FTP sites was compiled by John Van Weelie, president of the 9T9 Toronto User Group. The listing includes sites for the TI-99/4A or by 99/4A users. He can be reached at jvweelie@mgl.ca. The listing is current as of Aug. 16, 1996.

TI 99/4A RELATED WEB PAGES

The TI-99/4A Home Computer Page — Rich Polivka (Ohio, USA) http://w3.gwis.com/~polivka/994apg.html

TI Parallel Computing Page — Henry Koplien (Germany) http://sys00.ti6.tu-harburg.de/~ti6hk/hobby/

Kerry's TI-99/4A Page — Kerry High (Missouri, USA): http://www.umr.edu/~khigh01/994a.html

Emperor Ken's World — Ken Gilliland — Notung Software (California): http://ourworld.compurserve.com/homepages/notung

Don Shorock's Home Page — Don Shorock (Kansas, USA): http://homepage.midusa.net/~shorock/don.html

TI-99/4A Page — Owner ??? (Minnesota, USA): http://www.umn.edu/nlhome/m235/walt0101/index.html

Comp.Sys.TI News: http://www.tile.net/news/ti.html

The Virtual TI-99/4A Page — Red Wolf: http://www.he.net/~schmeli/ti.html

Gary W. Cox Home Page (Tennessee, USA): http://www.netten.net/~garycox/

Math Facts — Math Games B5 Software Co.:http://sjuvm.stjohns.edu/~lenbc/software/softm/softm005.html

Bruce Tomlin's Home Page: http://www.crl.com/~btomlin/

Eric LaFortune's Home Page (Rock Runner Author): http://www.cs.kuleuven.ac.be/~ericl/rock/

OPA — Gary Bowser's Home Page: http://www.io.org/~opanit/

Western Horizon Technologies Home Page (Don O'Neil): http://www.sonyx.com/wht/

TI-99/4A Classic Computer Page (Rob Patton): http://www.sundial.net/~rob/ti994a.htm

TI MBX Page (Rob Patton): http://www.sundial.net/~rob/ti_mbx.htm

Aurora's Homepage: http://firewall.fh-rosenheim.de:8888/~aurora/ti99.html

Dallas 99 Interface (DTIHCG): http://www.startext.net/today/news/opinions/starcol/dal99er.htm

Dallas 99 Interface (DTIHCG): http://www.arlington.net/today/news/opinions/starcol/dal99er.htm

Sam and Mary's Home Page — TI-99/4A Business SIG: http://members.aol.com/smorab/smhp.htm

Amiga TI Emulator — T. Brouwer:
http://src.doc.ic.ac.uk/aminet/misc/emu/TI4Amiga.readme
http://src.doc.ic.ac.uk/aminet/misc/emu/TI4Amiga.lha

Mark Wacholtz Home Page: http://www.paradise.net/~wacholtz/

Bobbitt Home Page — Chris Bobbitt Former Asgard Software owner: http://www.cais.com/fmg/bobbitt/

Larry Conner — L.L. Conner Enterprises (Indiana, USA): http://www.holli.com/%7Elconner/

Graphics and Picture Stuff Home Page — Bob Wray: http://ai.eecs.umich.edu/people/wrayre/explain.html

George Lin's Emulator Home Page: http://www.gml.com/pub/george/

Dean Dierschow's Homepage: http://www.xocolatl.com/dean/

Dean Dierschow's Cartridge Lists: http://www.clark.net/pub/vgr/dierschow.html

TI Fest West Home Page — Tom Wills: http://www.theriver.com/TheRiver/Cafe/Calendar/fest.html

Emulation Sites and Software: http://www.unix-ag.uni-kl.de/%7Esteve/emulation/emulation.html

Misc Video Game Stuff: http://www.dstc.edu.au/BDU/staff/dennis/vgames.html

Obsolete Computer Museum (Tom Carlson): http://www.ncsc.dni.us/fun/user/tcc/cmuseum/cmuseum.htm

TI Frequently Asked Questions (Novak): http://www.io.com/~vga2000/faq/ti.faq

Emulators — TI-99/4A: http://members.aol.com/chrissalo/emu2.htm#eti994a

TI-99/4A User Group list — 12-aug-90: http://web.state.ut.us/bbs/IBM/DL01/TI_USERS.TXT

TI-99/4A User Group list — 12-aug-90: http://folio.state.ut.us/bbs/IBM/DL01/TI_USERS.TXT

Mike's EMULATOR List: http://goliat.eik.bme.hu/~korn/emulator.html

TI-99/4A Comp.Archives (Nova Scotia, Canada): http://www.cs.dal.ca/ftp/pub/comp.archives/comp.sys.ti/

Phillip's CQ-PAN Home Page: http://cq-pan.cqu.edu.au/students/phillih1/

Gopher/game_archive/cartList/ti99.lst: http://gopher.spies.com/Gopher/game_archive/cartList/ti99.lst

Gopher/game_archive/cartList/ti99.lst: http://gopher.spies.com/ftp.items/game_archive/cartList/ti99.lst

Frequently Asked Questions: http://stekt.oulu.fi/~jopi/link/cem.faq

The Machine Room: Index of Micros: http://www.dcs.ed.ac.uk/~axc/MACHINE-ROOM/

JerryG's Classic Video Games: http://www.hevanet.com/jerryg/Games.html

The 3rd Dimension: http://www.hsv.tis.net/~jonecool/index.html

iNTERdINE dESIGN: http://www.interdine.com/

Emulator FAQ's: http://www.jmas.co.jp/FAQs/emulators-faq/part2

Adventure Game History File: http://www.lysator.liu.se/~unicorn/adv/agh.html

TI Frequently Asked Questions: http://www.officialsports.com/~vga2000/faqs/ti.faq

PHOAKS: Resources (comp.sys.ti): http://www.phoaks.com/phoaks/comp/sys/ti/index.html

Classic Games Cart List Server: http://www.serc.nl/People/zeist/d.atari/carts.lst

Nerd World : COMPUTER SCIENCE: http://www.tiac.net/users/dstein/nw402.html

(See Page 16)

HOME PAGES —

(Continued from Page 15)

The 3rd Dimension: http://www.traveller.com/~jonecool/

Virtual Computer Library Computer Technology: http://www.utexas.edu/computer/vcl/comptech.html

Flippette (Mike Brent): http://www2.awinc.com/users/mbrent/flippet.htm

Harborside (Scott Stasiowski): http://www2.corenet.net/utility/

Erik Olson's Home Page — Author of Mass Transfer, Etc.: http://www.users.interport.net/~olsone/junk.html

Brad Snyder's Home Page (Author of XB Packer) http://www.lehigh.edu/~bls3/bls3.html

Decatur 99ers Home Computer User Group: http://www.datalynxil.com/orgs/99ers.htm

Michael Zapf (TI/Geneve Userin Germany): http://www.vsb.cs.uni-fankfurt.de/~zapf/

Hans-Christian Alberts: http://www.informatik.uni-bremen.de/~hca

D.Thorburg — TI-99/4A Color Logo: http://www.tu-graz.ac.at/

TI-99/4A FTP SITES

OPA/NIT FTP Site — Gary Bowser (Toronto, Ontario, Canada): ftp://ftp.io.org/pub/users/opanit

Solutions FTP Site — T.M. Pederson: ftp://solutions.solon.com/pub/ti99

Clark Net Services FTP Site: ftp://ftp.clark.net/systems/systems/ti99

Bruce Tomlin's TI-99/4A Cartridge List: ftp://ftp.crl.com/users/bt/btomlin

V9t9 Emulator Source Code FTP Site — Edward Swartz: http://www.io.com/~ftp/usr/edswartz/v9t9/

TEXAS INSTRUMENTS Home Page

http://www.ti.com/

OTHER HARDWARE

Home Automation Systems — X10 Powerhouse System (Q&A): http://www.smarthome.com/smarthome/x10tech.html

BUY, SALE OR TRADE OF TI-99/4A COMPUTER

Marlin Bates in Fuji-dom!: http://sonnet1.sonnet.com/Fujidom/

Marlin Bates in Fuji-dom!: http://www.stocko.com/Fujidom/

Joe's Classic Sales Page: http://watt.seas.virginia.edu/~jap5r/

Video Game Advantage: http://www.io.com/~vga2000/

Zach D's Virtual Flea Market: http://www.hub.ofthe.net/~zachd/flea1.html

Classic Video Games Nexus: http://iquest.com/~lkseitz/cvg_nexus.html

Games & Gallery: http://pantheon.cis.yale.edu/~maxwell/index.html

Nathan Page's Home Page: http://talisa.acad.cai.cam.ac.uk/~nmp20/

Nathan Page's Home Page: http://www-gcsu.cai.cam.ac.uk/~nmp20/

Surplus TI Equipment — incl. 99/4A: http://www.73.com/a/0126.shtml

Computers: http://www.iaehv.nl/users/gigawalt/computer.html

Jeff Coleburn's Home Page: http://www.netaxs.com/~vsp

Keith Koch's Home Page: http://www.witness.com/%7Ekkoch/

OTHER RESOURCES

Maine Computer User Groups: http://www.state.me.us/msl/mgusrgrp.htm

Computer History & Emulation — Under Old Computers http://www.freeflight.com/fms/comp/

About Infocom Games: http://www.helikon.com/personal/pete/Advents/Files/Inform/HowTo.txt

Conference Room (Buckaroo Banazai Info): http://bbs.annex.com/relayer/confrnce.htm

The Home Computer Timeline — 1983: http://dspace.dial.pipex.com/town/square/chriss/timeline/1983.html

MIND LINK! BBS List: http://giant.mindlink.net/bbslist.html

The Old Computers List (Mark Robinson): http://mercury.ee.man.ac.uk/special/computers_body

Chicago CBBS List: http://techknow.com/wolfsden/cbbs.htm

TWIG — Picture of the Week: http://twig.hypercon.com/

Emulation (Brian Hanechak): http://www-unix.oit.umass.edu/~hanechak/emu.html

Logon — Sacramento BBS List: http://www.calweb.com/~logon/bbsrec/Bantoom_BBS.html

Computer Bits Online: User Groups in Oregon and SW Washington http://www.computerbits.com/ugc.htm

Jaguar: http://www.devolution.com/jaguar-l/1921.html

GamePen PC: http://www.gamepen.com/softwarepen/pc/zork.html

Chip directory: http://www.geo.mtu.edu/chipdir/chipdir.html

Jeremy Fitzhardinge's Home Page: http://www.progsoc.uts.edu.au/mlists/progsoc-1994/0215.html

PSCU User Group Directory: http://www.pscu.com/html/pscu/usrgrp.html

Technology Overload: http://www.rpi.edu/~boothj/80s/techno.html

Emulator FAQ: http://www.smartpages.com/faqs/emulators-faq/part1/faq.html

The Giant List of Classic Game Programmers http://www.sponsor.net/~gchance/Designers

Computer Shopper BBS/User Group List: http://www.zdnet.com/cshopper/bbs/filters/bbsdir.html

Charles Meadows Home Page: http://www.calweb.com/~cmeadows/bbs/b4874057.html

Chronology of Events in the History of Microcomputers http://www.i
landnet.com/kpoulson/comphist.htm

Comprehensive Computer Catalogue: http://plato.digiweb.com/~hansp/ccc/

Great Micropressors of the Past and Present http://www.cs.uregina.ca/~bayko/cup.html

Care & Feeding of Elderly Computers: http://www.inmos.co.uk/~iainf/computers.html

ARTICLES MENTIONING THE TI-99/4A

Colecovision: http://quartz.dcs.warwick.ac.uk:8080/~ross/Text/Coleco/coleco.html

Video Games Stuff (Dennis Brown): http://www.owlnet.rice.edu/~dgb/vg.html:

(See Page 17)

HOME PAGES —

(Continued from Page 16)

http://www.owlnet.rice.edu/~dgb/museum/texts/compbest.txt

Academic Programmers — A Spotter's Guide
http://dcpu1.cs.york.ac.uk:6666/pete/scribbles/progtype.html

Vintage Commentary: http://moliere.byu.edu/digital/vintage.html

Thomas Reimer's Home Page:
http://www.con.wesleyan.edu/~triemer/comp.html

A little info on Clint: http://www.grfn.org/~ccraven/me.htm

INside Magazine: http://www.monitor.ca/monitor/issues/vol2iss10/

Dennis' Home Page in English: http://www.on-luebeck.de/~dbrenke/english.html

Weiner's Home Page: http://www.physics.rutgers.edu/~bweiner/computer2.html

David Borgo's Home Page: http://www.pitt.edu/%7Edjbst57/

Saiche Me: http://www.xnet.com/~catena/me.html

Trincoll Journal:
http://www.trincoll.edu/tj/tj10.20.94/letters/letter4.html

OWNERS OR FORMER OWNERS OF TI-99/4A COMPUTER

Ray Dyer — Wizard Wolf: http://www.io.org/~tibm/

Greg Ward's Home Page: http://www.mni.mcgill.ca/users/greg/

Daniel Dreilinger's Home Page:
http://202.241.76.129:www.cs.colostate.edu/~dreiling/

·Daniel Dreilinger's Home Page: http://www.cs.colostate.edu/~dreiling/

Home Page for SHicks4000 (Steve Hicks):
http://home.aol.com/shicks4000

Kevan's Computer Bits...: http://www.ftech.net/~kevan/collection/

My Cheezy Art Page: http://ally.ios.com/~spider69/art.html

Tim Kent's Home Page: http://students.cs.byu.edu/~tk/tk.bio

The Castle Lord (Daniel Craigg):
http://csugrad.cs.vt.edu/~dcraigg/throneroom.html

CS 3604 — Professionalism in Computing:
http://ei.cs.vt.edu/~cs3604s/topic1.html

Agee's Home Page: http://godzilla.eecs.berkeley.edu/~agee/

Jack Martin's (Webmeister in Training) Home Page http://home.earthlink.net/~ojmartin/index.html

The Cat's Basket: Computing and Comms: http://homepages.enterprise.net/leigh/cp1.html

Rutt's Palmtop Cafe — Calculated Risk:
http://people.delphi.com/rrutt/calcrisk.htm Rick

Jason Gabler @ Work:
http://quadrophenia.ucdavis.edu/work/work.html

More About Michael:
http://www.access.digex.net/~mcheek/MLIFE.html

Den of Antiquity: http://www.acsu.buffalo.edu/~rconnor/den.html

Dave Warner's Home Page:
http://www.albany.net/~dwarner/dwinfo.html

Work Bio: http://www.america.net/~mouse/vita.html

Classic Video Games:
http://www.auburn.edu/~gibsowc/games/games.html

HTML, CGI, and PERL — Oh My!:
http://www.colorbus.com/01cbus/users/brian/code.html

Neil W. Van Dyke Home Page:
http://www.cs.brown.edu//people/nwv/personal.html

Old School Gaming: http://www.delta.edu/~jmgerris/

BX Page: http://www.diemme.it/sline/link/testi/BX/BX0827.TXT
http://www.diemme.it/softline/link/testi/BX/BX0827.TXT

Paul's Home Page: http://www.ecn.uoknor.edu/~peberrym/

Clark's Classic Video Games: http://www.eden.com/~cw/classic.html

>From Hell: http://www.freenet.hamilton.on.ca/~ad329/from-hell.html

Matt Banta's Home Page: http://www.gl.umbc.edu/~mbanta1/

Programming:
http://www.gsgis.k12.va.us/users/jcrocket/program.html

Brad Ackerman's Home Page: http://whitestar.resnet.cornell.edu/

Kenhn's Home Page: http://www.ici.net/cust_pages/kenhn/author.html

Rick Wilson's Home Page: http://www.infinet.com/~rwilson/rick.html

HCA-Computers: http://www.informatik.uni-bremen.de/~hca/computer.html

Alexander Bochmann's Home Page:
http://www.infra.de/~bochmann/Welcome.html

Eric Smith's Home Page: http://www.ior.com/~ebsmith/eric/computer.html

Tandy 102 Home Page:
http://www.kippvisual.com/ex/Tandy102.htmlB

Mikael's Home Page: http://www.kuai.se/~hengus/Mikael.html

Michael Kamprath's Home Page: http://www.leonardo.net/kamprath/home.html

Dubois's Home Page: http://www.lirmm.fr/~dubois/possess_eng.html

Charles Wilson's Home Page: http://www.mnsinc.com/charlesw/

Back Slash's Home Page:
http://www.nicom.com/~bakslash/arthur.html

John Weise's Home Page: http://www.probe.net/~jwiese/john.html

The Formative Years:
http://www.realtime.net/~alramsey/formtive.htm

Chris Swindle's Home Page: http://www.rose-hulman.edu/~swindlcg/comp/comp.html

Leigh Benson's Home Page: http://www.smallpenis.org/bio.html

Get To Know Us?: http://users.uniserve.com/~blevis/welcome.html

Rick Nyman's Home Page: http://www.serve.com/rnyman

Office of Information Systems — Scott S. Street
http://www.state.de.us/webmast/sstreet.htm

Alex Francis' Home Page: http://www.t-n-c.com/homealex.htm

HTML document for the World Wide Web:
http://www.techline.com/~rag/rag.htm

Walter Gibson's Classic Video Games: http://www.auburn.edu/~gibsowc/games/games.html

Classic 1980's Computers/Video Game Systems
http://rmet.com/~bfrandse/classic.html

SEARCH ENGINES

Canoe: http://www.canoe.com/

Yahoo: http://www.yahoo.com/

Lycos Search Engine: http://www.lycos.com/

Opentext Search Engine: http://search.opentext.com/

Alta Vista Search Engine: http://www.altavista.digital.com/

WebCrawler Search Engine: http://query.webcrawler.com/

Infoseek Search Engine: http://guide-p.infoseek.com/

excite Search Engine: http://www.excite.com/

DejaNews Search: http://search.dejanews.com/

ASSEMBLY ON THE GENEVE

Basics start with understanding of assembly on the TI

By MICHAEL ZAPF

After its emergence in the late 1980s, no one really expected the Geneve to draw the majority of the TI users from their good old TI to its predestined successor. Designed as an expansion card for the Peripheral Expansion Box, the possible audience could be at most a subset of the TI community.

In contrast, the Geneve had its difficulties gaining acceptance, partly by its price — who really dared to spend hundreds of dollars for a computer compatible to an orphan? — but also because of Myarc's questionable reputation, the low quality of some products, erratic support, and lack of cooperation to resolve problems.

Another point of criticism was brought up by some programmers. The Geneve, they said, abandoned the DSR principle. For those of you who are not that familiar with assembler programming, the DSR technique was introduced by Texas Instruments for the TI99/4A and guarantees its simple expandability by referencing external devices by name. To find the correct device, be it the disk drive, the hard drive, an interface, or a software device like the text-to-speech utility of TE-II, only one central lookup routine was necessary. Even when the Myarc Hard and Floppy Disk Controller was introduced, no change needed to be made to the system — although no one had thought of connecting a hard drive when the TI was designed in the late 1970s.

The Geneve can still use all standard peripheral cards. But Myarc obviously decided to drop the pure DSR technique. Instead, all known device names were included in the MDOS operating system that served as substitutes for the real names. The decision might have been based on the desire to enhance some functions of the disk drive, e.g. include time stamps, or to implement a print spooler. The calls to the respective devices must be intercepted and routed through an MDOS system routine. That means when a new card comes up, the operating system must be updated. While this would not pose any problem for the Geneve (because its system is loaded from disk), no new cards except the SCSI controller were seen so that this problem remained more or less theoretical. (Let's avoid a discussion of the role of cause and effect.)

But several outstanding and well-written programs came into existence. The best contributions were the c99, TIC, and Fortran compilers, some painting and designing programs, and a bunch of utilities. However, most of these programs serve as utilities to build other projects, especially the compilers. That means that there are simply too few real applications: few heavyweight editors (more than just to enter programs), few games, few data processors, and so on. Albeit it offers stunning features — large memory, improved graphics, standard mouse support, even multitasking support by MDOS — the Geneve did not receive the attention it deserved. Other than all the different expansion cards that can be found for the TI, the Geneve could have introduced a new standard that all users could have relied upon. It did not, and this is very, very disappointing.

Programming the Geneve was never too easy. While a TI programmer has nearly all information he needs for writing good programs, the Geneve's operating system is far more complex but additional information is hard to get, especially for those who do not have access to BBSs or, now, the Internet. My intention is to encourage more people owning a Geneve to get their papers, pencils and keyboards ready to write all those programs every other computer user cannot even imagine doing without.

In this article I will start with the basics of assembler programming, but I will presume a good understanding of it on the TI. The first goal is simply to demonstrate how to build working assembler programs on the Geneve using TASM and LDR. And they will run in MDOS, not in GPL mode! (TASM, TASN, TASO, and LDR are part of the c99 program package, which is available on various online services.)

FIRST STEPS

As I have already said, we do not need to reinvent the wheel, so let's jump right into the cold water. We will use TASM as assembler; I think almost everything that is said here applies to any other MDOS assembler.

TASM produces relocatable object code as does the original TMS9900 assembler. Other than with the Editor/Assembler, the loader of these files is not included in the system. Its name is LDR, and it also comprises the SAVE utility.

TASM and LDR allow exporting and importing of symbols of separate object files, that is, usage of DEF and REF. The E/A programming environment keeps the defined symbols until they are cleared; as there is no such underlying environment in MDOS, all symbols are lost when LDR exists.

This has several consequences. First, you cannot "install" programs for later use. After loading, no other program will have access to any defined symbols. (In the MS-DOS world an installable program is called TSR: terminate and stay resident. There could be ways to simulate this in MDOS, but I have not used it so far.)

Second — and as a consequence — there are no utility routines that can be invoked by the BLWP @ symbol, as the well-known VSBW, VMBW, etc. To offer a little comfort to the programmer, MDOS implements all common utility routines (and many, many more!) by XOP. The routine below shows a little example of how to use some utility XOPs.

Third, you must load all modules of a program in one turn. All necessary files need to be mentioned in the LDR command.

E/A allows the generation of program code that is very compact and fast-loading. The drawback is that it is not relocatable any more. In MDOS, this feature is provided by appending another file name to the LDR command line. After loading the object modules, a series of about 16-kilobyte program files are generated (with the last character increasing). Again we lose relocatability but, as we will always have all memory available, the program will reside on the same locations anyway.

The example following this text illustrates each mentioned point. It is assumed that all files reside in the same directory on the hard or floppy disk. Please make sure you have the following files: TASM, TASN, TASO, LDR.

(See Page 19)

ASSEMBLY ON THE GENEVE —

(Continued from Page 18)

MAKING THE FIRST SAMPLE PROGRAM

After entering the source code, save the file as SAMPLE_S. At the MDOS command line enter:

TASM SAMPLE_S

If there are no typing errors, the assembler should exit with "Assembler done. Errors: 0" and produce a file called SAMPLE_X. This file is the relocatable object code which you can load and run by using LDR SAMPLE_X.

However, each time you want to run it you will have to use the LDR. You can get an independent (but also non-relocatable) program file with LDR SAMPLE_X SAMPLE.

Now the program is not executed but saved in the given file. It can now be started at the MDOS command line simply by entering "SAMPLE."

If you look at the source code you should notice that the usage of XOPs is like macro programming, because all you have to do is to provide the necessary data. You do not need to know how the routines exactly work. In this example the usage of XOPs is recommended, but when you have time-critical program sections you should use your own routines than those offered by MDOS. This applies especially to the video routines, and in the next parts of my MDOS programming lessons we will talk about that in detail.

Beside the advanced video access, we will also discuss how to provide some useful routines, e.g. DSR access and mouse processing in graphics and text mode by taking a glance at the routines used in my program Fractals!2.0.

If there are any questions feel free to send an email to: zapf@vsb.cs.uni-frankfurt.de. Or even better, post your message to the Usenet newsgroup comp.sys.ti.

I'm looking at this newsgroup almost daily so that there is a good chance that I will get your message very quickly, and many other users may profit by the discussion.

SAMPLE PROGRAM

```
**** Sample program ****

*                 DEF isn't needed. The first word MUST be exe-
cutable.
*                 REF neither, because we don't have separate
modules
*                 and we can't reference any utility routine as
in E/A

B @START

SAMPLE TEXT 'This is a TMS9995 program.'
BYTE >0D,>0A
TEXT 'Press any key.'
BYTE >0D,>0A
BYTE 0

KEY DATA 5
VIDEO DATA 6

START LWPI >F000 choose Workspace in on-chip RAM

*                 LIMI 0  Uncomment this and see what happens!
*                 Your conclusion?

LI R0,>0027       WriteTTY video routine (like PRINT)
LI R1,SAMPLE address of string
CLR R2            which is null terminated (non-zero = length)
XOP @VIDEO,0      branch to XOP routine

LI R0,>0033       MakeSound video routine (like CALL SOUND)
LI R1,>0578       1400 Hz sound (familiar 'beep')
CLR R2            no second generator (not needed when at-
ten.=15)
CLR R3            no third generator (applies for this, too)
LI R4,>010F       turn volume1 up, make gen2 mute
LI R5,>0F00       gen3 mute
LJ R6,>000F       no noise setting; make noise mute
LI R7,12          12 sixtieth of a second (0.2 sec)
XOP @VIDEC,0      do you 'see' the sound?

LI R0,5           keyboard mode 5
KLOOP XOP @KEY,0  invoke key scan
JNE KLOOP         EQ set when a new key has been pressed

BLWP @>0000       exit to MDOS (just to the command line!)

END
```

NEWSBYTES

1987 MUG conference set

The next all 99/4A and Geneve Multi User Group Conference will be May 23-24 at the Lima Campus of Ohio State University, according to Charles Good of the Lima Users Group.

The conference is free to attendees and vendors. Good says the space has been booked for the Memorial Day Weekend from 4 p.m. Friday through 7 p.m. Saturday. For information, contact Good at P.O. Box 447, Venedocia, OH 45894. Phone (419) 667-3131. E-mail address good.6@osu.edu.

Cactus Patch goes offline

The venerable Cactus Patch BBS went offline on Sept. 1, according to Tom Wills. In a posting on the Internet, Wills said, "I have been involved with this BBS for the past 7 1/2 years. Prior to that I ran the Nearer The Lake BBS for the Sheboygan (WI) Area 99ers User Group for several years." The Cactus Patch BBS is the official BBS of the SouthWest Ninety Niners User Group of Tucson, AZ.

"In the past year usage has plummeted and the value of keeping Cactus Patch has continually diminished," Wills said. "When we can go days between calls, and there are only 66 users still logged on, the question needs to be raised "is it worth the cost to keep the BBS online?" Of the 66 users still logged onto the BBS, 6 are non-TIers. Of the remaining 60, less than half have been online this calendar year."

Pascal Primer

Using the TI p-Code system

By STAN KATZMAN

This is reprinted from the Chicago TImes, *the newsletter of the Chicago TI Users Group. It is the firsttin a series.*

A while back John Willforth asked me if I would write a column on the Pascal/p-Code system. I was hesitant because I don't know too many people that have p-Code or people that want (or can get) p-Code, so I don't know how much of a demand there is for knowledge of this system. I decided to do it anyway because this is the least covered system in the TI family.

Let us first discuss the hardware and software needed. In order to run Pascal on the TI one needs:

1. the p-Code card
2. at least *two* disk drives (I am running two DS/SD drives)
3. RS-232 card and a printer (this is necessary because the programs can get to be *very* large and debugging without a print-out would be very difficult)
4. 32K memory
5. p-Code Editor
6. Pascal compiler
7. Filer
8. Utilities
9. Assembler and Linker

I do not use the Assembler and Linker, but if you can obtain these pieces of software, do so, and you can grow into them. You will also need the documentation (which is very extensive).

There are some books that I would also recommend:

- Any Pascal text that covers Standard Pascal and/or U.C.S.D. Pascal (not Turbo Pascal)
- *Introduction to the U.C.S.D. p-System,* by Charles W. Grant and Jon Butah, Sybex, 1982
- *Advanced U.C.S.D. Pascal Programming Techniques,* by Eliakim Willner and Barry Demchak, Prentice Hall, 1985 and 4) *The U.C.S.D. Handbook,* by Randy Clark and Stephen Koehler, Prentice-Hall, 1982.

In p-Code the No. 1 drive is called No. 4, the No. 2 drive is called No. 5 and the No. 3 drive is called No. 9

After getting the hardware and software together the first thing that has to be done is to back up the software; this is very important, please do nothing else until this is done.

We have to format disks in order to back up the software, so let us discuss how to do this in the p-Code system. If you only have SS/SD drives you can use the DFORMAT program on the Utilities disk. This software does *not* work for DS or DD disks; the disk will come out SS/SD in spite of what the menu prompts say in DFORMAT. To format a disk other than SS/SD go into XBASIC and using DM-1000 format a series of disks.

Now shut off the console and turn on the p-Code card using the switch on the back and put the Filer disk in the No. 1 drive and a disk formatted from XBASIC in the No. 2 drive. (In p-Code the No. 1 drive is called No. 4, the No. 2 drive is called No. 5 and the No. 3 drive is called No. 9; we will use these numbers from here on.) Now turn on the console and the drives will turn on and after about a minute you will see across the top a series of prompts and in the middle of the screen a greeting message. The cursor will be at the top of the screen and at this point pres "F" (for Filer) and then press "Z" (Zero).

You will now see "Zero dir of ?" Now enter "#5:"; now it will say "Duplicate dir?" Press "Y"; next it says "Are there 180 blks on the disk? (Y/N)" (if the disk is DS/SD or DS/DD enter "N"; if SS/SD enter "Y"), then the screen reads "# of blocks on the disk" (if DS/SD enter 360, i DS/DD enter 720).

Next, the screen reads "New volume name?" At this point enter a disk name (the one from BASIC will not count) with a max of eight characters starting with a letter.

Next the screen reads "(new volume name) correct?" At this point enter "Y" o "N" ("N" lets you re-enter the volume name). When the Filer has finished initializing the disk it displays "(new volume name) zeroed"; do this until all of you disks are initialized. These are formatte disks. The zeroing process is found o page 52 of the Filer.

If you only have SS drives, put the Utilities disk in No. 4 and the blank disks i No. 5. Follow the direction on page 15 o the Utilities documentation. Then take ou the Utilities disk and put the Filer disk i No. 4 and go through the "Zero" proces described above.

To back up an entire disk, put the File in No. 4 and empty formatted disk in No 5. Call the Filer by entering "F" and the enter "T" (Transfer). You will then se "Transfer?" At this point take the Filer ou of No. 4 and put the disk you want to b backed up in No. 4, then enter "#4:,#5:" you will now see "Destroy (diskname) ("diskname" is the name of the disk in No 5). At this point press "Y" and the copyin will be automatic. Do this for all of th disks that came with the system.

This article has been a bit long so w will quit here. More next time.

TI list server

Online TI community can keep in touch automatically

For the first time, online TI users can keep up with TI newsgroups automatically by subscribing, free of charge, to a new TI list server.

For those who do not know what a list server is, it is a email address that is used to send messages to everyone who has "subscribed" to that server. The "subscription" to the new TI list server is free. However, you will need access to an internet provider. The cost of this type of service can be had for nothing, if you use a Freenet site or internet from your work place to a fairly substantial sum, depending on what's available in your area. There are some national internet providers who can be used for as little as $9 a month.

The list server is administered by Tom Wills and is an extension of the SouthWest Ninety Niners User Group of Tucson, AZ.

To subscribe to the TI List Server, send an email to the following address: "majordomo@theriver.com" (without the quotes)

Leave the subject line BLANK and include the following as the message BODY: "subscribe ti99" (without the quotes of course)

Please remember not to include your signature file in the message body.

After Tom has had a chance to review the subscription requests, and approve them, you will receive an email message telling you that you've been approved to use the server. At that time you will start receiving messages that others are sending to the server. You will then be able to use the TI99 list server to your hearts content.

> This service is not a replacement for the TI news group, COMP.SYS.TI, but a supplement to it. The big difference is that subscribers to the server will be sent all messages as soon as the server receives them.

This list server will not be moderated, as the TI community is a very reliable group of people. However, it is not an open list server (that is the ability to subscribe to it without approval). This is done so as to keep nosy non-TIers and advertisers off the list server.

After you are approved, just send your messages to ti99@theriver.com. Everyone who has subscribed to the list server will receive your email. It is just that simple.

An added feature will be the ability to find out who has subscribed to the server. To do so, send a message to majordomo@theriver.com (with a BLANK subject line) and "who ti99" (without the quotes) in the main body. Again, please do not include your signature file in the message. The server will then send you an email listing the addresses of everyone currently subscribed to the ti99 list server.

This service is not a replacement for the TI news group, COMP.SYS.TI, but a supplement to it. The big difference is that subscribers to the server will be sent all messages as soon as the server receives them. To see what is in the news group, you must specifically attach to it and decide which messages you wish to read. Each method has its advantages and disadvantages.

One thing this new service will replace is the TI Internet Mailing List that Tom Wills was maintaining. Maintaining the mailing list was very difficult for several reasons; the constant changing of addresses by people on the list, the internet recourses themselves, and the time involved. It will be much easier to pay $5 a month and maintain the list server than it ever was to maintain the TI Internet Mailing List.

To directly contact Tom, send an internet email message to twills@theriver.com.

Reduce monitor glare to avoid eyestrain

Eyestrain is most common, according to business product marketer Quill Corporation, among computer users who:

- Have glare on monitor screen from natural or artificial light resources.
- Sit too close to their screens.
- Have their monitors facing a window.
- Do not practice resting the eyes throughout the day.
- Have overhead lighting too bright.
- Have monitor screen brightness set too intensely.
- Position the monitor screen above eye level.

The first defense against monitor glare is to identify the source and eliminate it if possible. If the source of glare cannot be completely eliminated, try tilting the monitor screen away from the light source. Overhead lighting can be reduced, and desktops illuminated with task lighting without creating additional glare. Positioning the monitor slightly below eye level allows the eyelids to cover more of the eyes, which helps prevent eye dryness and reduce strain on eye muscles.

A range of glare filters are available:

- Mesh filters reduce general glare.
- Optically coated filters have an anti-reflective surface to reduce glare and improve character contrast.
- Privacy filters provide security by preventing viewing from either side. They also are anti-reflective, reduce glare and improve character contrast.
- Antistatic/radiation filters suppress the electric (E-field) component of electromagnetic radiation emitted from monitor screens. The filters also dissipate static electricity with a ground wire.

An adventure with AMS from two points of view

By BRUCE HARRISON
and LEW KING

People are buying the Advanced Memory System (AMS) now that it's being produced and sold by the SW99ers of Tucson, AZ. One of those customers is Lew King, of Industry Pennsylvania, who's co-authoring this article. His perspective as an owner will follow, but for the moment it's the viewpoint of an experienced Assembly programmer.

There are two potential kinds of applications for the AMS memory card. One is the creation of massive programs, in which the code to operate the program needs more than 32K of RAM. That's not the forte of this particular Assembly programmer. The second kind of application is one in which the program itself occupies only part of the 32K memory, but the program needs large blocks of memory to handle data. For example, you might have a word processor such as Funnelweb where the text buffer could be expanded to perhaps 16 times what Funnelweb could normally handle. Now that's the kind of application that gets Bruce Harrison's attention.

At the M.U.G. in Cleveland, Lew King bought his AMS card, with 256K capacity. When I asked what he'd do with all that memory, he quickly came up with an idea for me to pursue. Some time back, I produced my Video Titler program, which uses bit-map images. Just one such image (for example a TI-Artist picture with color included) occupies 12K bytes of memory. Two such images completely fill the high portion of the 32K memory.

Thus, in the original Video Titler, the program code and internal data had to fit entirely in low memory, starting at >2678, so that all of high memory from >A000 through >FFFF could be used to store two complete bit-map color pictures. That meant that after just one "wipe" transition, the user would have to pause the VCR, load the next frame from disk, then proceed to the next transition. That works, but it's labor intensive and slow work for the user. Lew immediately saw the possibility for pre-loading many frames into the AMS, so that pauses for reload from disk would be minimized.

Having this idea to work with was just the start. To make a working program, we had to have some definitive words on how to take advantage of the AMS in terms of Assembly source code. Nobody at the Cleveland gathering had that kind of information at his fingertips. There is no published source (until now) for that kind of information. After a couple of unsuccessful attempts to get the needed information, I asked Lew to print out and send me the "docs" that were delivered on disk with his AMS card. Lew complied, and within two days I had printed docs plus several disks. As it turned out, it was fortunate indeed that Lew included those disks, because one of them contained a source file called AMSINI. That sounded like it could be important, and it was. AMSINI contained the key source code segments to make use of the card. There was still a problem, however, in that some of the material in the printed docs contradicted parts of the AMSINI source file.

(See Page 23)

Bruce Harrison releases AMS application, Speed Read

Bruce Harrison has released a new version of his Video Titler Version which takes advantage of the memory of the Super AMS Card. He has also released a new commercial software product called Speed Read.

Harrison say the new Video Titler versions has all 23 "nifty wipes" of the old Video Titler, but uses the AMS card to store frames of bit-map data so sequences of many titles can run without interruption. For example, he says, on a 256K card, a user can run 20 complete frames in sequence without reloading.

The commercially released AMS Video Titler's price is $10, including shipping and handling. It will work with or without AMS, though Harrison notes that "its big advantage is when used with AMS in place." Harrison says the program discovers for itself whether an AMS card is present and automatically sets itself for the amount of memory that is there. The disk contains complete instructions, source code and sample title files. Harrison says the program uses files created either with TI-Artist or the public domain Harrison Drawing Program.

Speed Read is designed to improve the user's reading speed by displaying the contents of D/V80 text files at a controlled scroll rate.

Harrison says any D/V 80 file can be used, even one too large to edit on the TI. The user controls the rate of scrolling, while the program indicates the current speed in words per minute. Harrison says the product is compatible with both TI and Geneve computers. He says the scroll speed is independent of the CPU speed, so rates are consistent on TI or Geneve. A "bookmark" feature allows the user to re-enter large files at any point without having to read from the beginning, he notes.

The program is available in two versions, one for U.S. and Canadian users (60 Hz vertical rate) and the other for European users (50 Hz vertical rate). For U.S. and Canadian users, the price is $5, for overseas users, $7. The price includes shipping and handling.

Contact Harrison at 5705 40th Place, Hyattsville, MD 20781. For information on the AMS card, contact the SouthWest Ninety-Niners, P.O. Box 17831, Tucson, AZ 85731-7831.

ADVENTURE WITH AMS —

(Continued from Page 22)

After working through a chain of informants, I got in touch with Rich Gilbertson, who was making a version of his Rich Extended BASIC (RXB) for the AMS card. Rich filled in the gaps, telling me that the printed docs were wrong and the AMSINI file correct. He pointed out all the non-obvious things, like the fact that in its power-up condition certain pages of the AMS are preassigned to the low and high portions of 32K memory. Pages 2 and 3 are the low memory, and pages 10 (>A) through 15 (>F) are assigned as the high memory. In other words, if your program loads in low memory, it will reside in pages 2 and 3 of the AMS, and if it writes to or reads from high memory, that will go into and come from pages >A through >F of the AMS. (One page is 4K of memory. pages >A through >F will hold 24K bytes.)

Among other things, this means that pages 0 and 1 and the group of pages from 4 through 9 are unused in the power-up configuration. Pages beyond 16 (>10) are also unfilled on power-up. Some programmers (including Gilbertson) prefer to leave pages below 16 (>10) unused, reserving them for future use with an alternate operating system. Being an old curmudgeon who doesn't believe any such operating system will ever emerge, I decided that pages below 16 would not be left alone by my Video Titler program.

But first, since AMS cards come in various configurations, with capacities of 128K, 256K, and even 512K, I had to have a way for my program to measure the amount of memory that was present, and adapt its operation accordingly. Fortunately, that AMSINI file on the disk contained just such a program segment. I adapted that slightly, so it would measure page by page starting at 16 and going to whatever was the highest page available. Since pictures occupy three pages each, I divide the number found by three to find the number of "frames" that can be stored in the memory above page 15. In the Video Titler case, we can use pages 10 through 15 to store two more frames, since the entire program remains in low memory, pages 2 and 3.

Now let's suppose we've got a card with only 128K memory. There will be only 16 pages above 15, and using them in groups of 3 will mean only 5 frames there. Adding the two frames in pages 10 through 15 means seven frames can be stored. If we use that unused space from page 4 through 9, two more frames can be stowed, for a total of nine. Thus in the program we take the number of pages beyond 15, divide that by three (ignoring any remainder) and add 4 to come up with the total number of frames we can keep in memory. For Lew King's card, with 256K capacity, there are 48 pages past 15, so a total of 20 frames can be stored. For 512K capacity, a whopping 41 frames could be put into memory.

In the sidebar for this article are a couple of "snippets" of source code that may be helpful to anyone trying to make use of the AMS card for data-intensive applications. First in that is a "complete program" section called AMST. This was the first thing I sent to Lew King for testing. This little program, if run on my own machine, which has no AMS card, will simply report that fact. On Lew's machine, it reports 48 pages and 20 frames as the capacity of his AMS card.

This source can be modified easily to report in different ways. For example, you could take the number of pages above 15, add 12 for the pages from 4 through 15, then multiply by 4 to indicate available capacity in kilobytes.

ACCESS AND MAPPING

To use other than the default setup, you have to do two things. First, you have to "turn on" the card's memory in the >4000 block and write to the mapping registers there. Second, you have to "turn on" the mapper function to make what you've written into the >4000 block take effect.

The first step in all this is to set Register 12 to the value >1E00, which is the CRU address of the AMS card. Now doing a SBO 0 instruction will enable the >4000 memory block for reading and writing. In that block, each word starting with >4000 and ending with >401E acts to control what page maps into what memory location. The subroutine AMSINI (see sidebar) initializes all the mapping registers to power-up condition. For example, >4004 is set for page 2, >4006 to page 3, etc. In our own program, we're mapping things into the >A000 through >F000 blocks. Thus the registers we write to in >4000 are at >4014 through >401E. To make a mapping selection, we write the desired page number to the high order byte of the control register. Let's take a concrete example. Suppose we want pages 4, 5 and 6 of the AMS to behave as if they were at >A000, >B000, and >C000. It's this easy:

```
LI   R12,>1E00 CRU ADDRESS
SBO  0         TURN ON CARD'S >4000 BLOCK
LI   R1,>0400  PAGE NUMBER IN LEFT BYTE
LI   R3,>4014  MAPPING REGISTER FOR >A000 BLOCK
MOV  R1,*R3+   WRITE TO >4014 AND ADD 2 TO R3
AI   R1,>100   LEFT BYTE OF R1=5
MOV  R1,*R3+   WRITE 5 TO >4016 AND ADD 2 TO R3
AI   R1,>100   LEFT BYTE OF R1=6
MOV  R1,*R3    WRITE 6 TO LEFT BYTE AT >4018
```

This sets the registers, but does not actually start the mapper working. To put the mapping in effect, we have to perform one more operation:

```
SBO     1      TURN ON THE MAPPER
```

Now that the mapper is turned on, anything written to >A000 through >CFFF will actually be written into pages 4 through 6 of the AMS memory instead of into pages >A through >C. Similarly, any memory reading from >A000 through >CFFF will actually read the contents of pages 4 through 6 of the AMS memory. Thus your program can write and read as if to >A000 through >CFFF, but the mapper will make things actually access the pages 4 through 6 in the AMS card. You can turn off access to the card's >4000 block and still leave the mapper in effect by the instruction SBZ 0. (Provided R12 still contains >1E00.) You can turn off the mapper by SBZ 1, and then the areas >A000 through >CFFF will revert to "normal," reading and writing to pages >A through >C of the AMS card.

In the finished AMS version of Video Titler, the program works by frames, where each frame accounts for three pages of the AMS

(See Page 24)

ADVENTURE WITH AMS —

(Continued from Page 23)

memory. We use an odd-even scheme, so that odd-numbered frames (1,3,5,...19) go into pages mapped to >A000 through >CFFF, and even-numbered ones go into pages mapped to >D000 through >FFFF. This way the program always has immediate access to two frames at any given time.

Let's say, for example, that the user asks to load a TI-Artist picture (with color) into frame 1. We take the desired frame number, multiply that by 3, add 1, then put that in the left byte of a workspace register. Since this is an odd frame, we use the mapping registers at >4014 through >4018, and put 4, 5, and 6 into the left bytes of those mapping registers, respectively. (For an even page, we'd use >401A through >401E.)

Now when the program reads the file, it dumps the contents of the pattern and color parts into >A000 through >CFFF, which really goes into pages 4 through 6 of the AMS memory. In similar fashion, frame 2 goes into pages 7 through 9, frame 3 into pages 10 through 12, frame 4 goes to pages 13 through 15, and so on until frame 20, which goes into pages 61 through 63. Clear as mud? Look at part two of the sidebar, which may help. Meanwhile, how about a little different perspective?

THE AMS OWNER'S VIEWPOINT

The Super AMS card is a most welcome and needed addition to the TI99/4A. The physical aspects are a well-made, high-quality card which inserts in place of the 32K card. In normal use, it duplicates the 32K card exactly, except for the absence of an indicator light. In expanded mode, the card provides from 128K to 1 megabyte of paged memory, depending on the chips installed.

Three disks of software are provided with the card, of which two are archived. Included is an excellent memory tester that writes and reads to every bit. This assures the user that everything is functioning properly. Also included is an excellent disk copy program, AMS Copy. The program reads an entire 720-sector disk into memory, then writes to the copy disk in one pass. Contents of the master disk remain in memory, allowing as many copies to be made as needed without re-reading. Also there were two games, which I could not get to run. The remainder of the software is mainly for programmers and of little consequence for the end user. So, aside from testing the memory and copying disks, what can be done with the Super AMS card? Well, before the middle of July, the answer was "not too much."

Enter Bruce Harrison. When I first purchased the Super AMS card back in May at Cleveland, Bruce dismissed it, saying that he could write any program for the TI and keep it well within the 32K space. No argument here. Then I posed the hypothetical question — how many titles could the Video Titler program hold in 256K of memory? To which Bruce answered, "I'll think about it."

Well, Bruce did a lot of thinking and a lot of programming. The result is that the Video Titler can now hold up to 20 color graphic files in memory on a 256K card. Super. They are instantly available using any of the many different wipes in the program. If you make a mistake, or change your mind, any frame can be reloaded without changing the others. The sequence can also be started at any frame.

Anyone who has previously used the Video Titler will most certainly appreciate the new features, added versatility, speed, and power of the new AMS Video Titler. The entire operation of paging 20 titles into memory, then using them is totally seamless and transparent to the user. It is as though the TMS9900 had 20 address lines. Substantial documentation is included with the program, but I dare say that most folks won't read them. Aside from possibly looking up which keys to use for what wipes, the Titler is that easy to use. When you are ready to quit, the Titler returns nicely to the Editor/Assembler menu screen.

It's rather ironic. To the best of my knowledge, Harrison is the first programmer outside of the SAMS development team to make any real productive software for the Super AMS card. Yet, he is the one programmer that never received a card to use. The AMS Video Titler beta copy worked flawlessly the very first time. This is a real tribute to outstanding programming ability.

PROGRAMMER'S UPDATE

Since the above was written, I have received a set of the AMS documentation and an offer of a card from David Ormand of the SW99ers. David has been sent a copy of the AMS Video Titler and the SW99ers have sent me a card to use for programming and testing my new products designed for use with AMS.

AMS SIDEBAR

```
* SIDEBAR PART ON - A TEST PROGRAM
* COMPLETE AS SHOWN
*
* TEST PROGRAM FOR AMS CARD
* SPECIAL BY B. HARRISON
* AMST/S
       REF  KSCAN,VMBW
       DEF  START
KEYADR EQU  >8374         Key-unit address
KEYVAL EQU  >8375         Reported keystroke
STATUS EQU  >837C         GPL STATUS BYTE
GPLWS  EQU  >83E0         GPL workspace
GR4    EQU  GPLWS+8       GPL Reg 4
GR6    EQU  GPLWS+12      GPL Reg 6
STKPNT EQU  >8373         stack pointer
LDGADD EQU  >60           load grom address address
XTAB27 EQU  >200E         a storage location
GETSTK EQU  >166C         get stack

START  LWPI WS            LOAD OUR WORKSPACE
       BL   @AMSINI       USE AMS INITIALIZE SUBROUTINE (BE-
LOW)
       C    R1,@>401E     AMS CARD PRESENT? 0=NO
401E=0F0F=YES
       JNE  NOAMS         IF NO AMS CARD, JUMP AHEAD
       SBO  1            TURN ON MAPPER
       CLR  R1           R1=0
       CLR  R4           R4=0
       MOV  R1,@>401E     SET >F000 TO PAGE >00
       MOV  R1,@>F000     SET FIRST WORD PAGE 0 TO 0
       LI   R3,>8000     R3 TO PAGE >80
       MOV  R3,@>401A     SET >D000 TO PAGE >80
       LI   R1,>0F00     START AT PAGE 15
AMSLP1 AI    R1,>100      ADD ONE PAGE
       INC  R4           INC NUMBER FOUND
       MOV  R1,@>401C     SET >E000 TO PAGE 16, 17, 18, ETC.
       MOV  R1,@>E000     LOAD >E000 WITH PAGE #
       C    R1,@>E000    DID IT LOAD?
       JNE  AMSDO         IF NOT, WE'RE BEYOND LAST PAGE
```

(See Page 25)

ADVENTURE WITH AMS —

(Continued from Page 24)

```
       C     @>F000,@>E000      CHECK PAGE 0 FOR CHANGE
       JEQ  AMSDO          IF EQUAL, JUMP
       CI   R1,>9000       LIMIT NUMBER?
       JL   AMSLP1         IF LESS, REPEAT
       C    @>D000,@>E000      CHECK PAGE 80 FOR CHANGE
       JNE  AMSLP1         IF NOT EQUAL, REPEAT
AMSDO  DEC  R4             ONE LESS (R4 EQUALLED PAGE NOT
FOUND)
       MOV  R4,@AMS        SAVE NUMBER OF PAGES ABOVE 15
       BL   @AMSINI        AMS INITIALIZE (BACK TO "NORMAL")
       SBZ  0             TURN OFF CARD - MAPPER STILL ON
       SBZ  1             TURN OFF MAPPER
       JMP  REPORT         JUMP TO REPORT SECTION
NOAMS  LI   R0,11*32+3     ROW 12, COL 4
       LI   R1,NOSTR       NO AMS MESSAGE
       BL   @DISSTR        DISPLAY THAT
       JMP  KEYEX          JUMP TO EXIT ROUTINE
REPORT LI   R0,10*32+3     ROW 11, COL 4
       LI   R1,PGSTR       PAGES STRING
       BL   @DISSTR        DISPLAY THAT
       A    R2,R0         ADD LENGTH TO POINTER
       MOV  @AMS,R3        GET NUMBER OF PAGES PAST 15
       MOV  R3,@>835E      PUT AT >835E
       BL   @SHWINT        SHOW NUMBER ON SCREEN
       LI   R0,12*32+3     ROW 13, COL 4
       LI   R1,FRMSTR      FRAMES STRING
       BL   @DISSTR        SHOW THAT
       A    R2,R0         ADD LENGTH
       CLR  R2            CLEAR R2
       LI   R1,3          SET R1 TO 3
       DIV  R1,R2         DIVIDE R2-R3 BY 3
       AI   R2,4          ADD 4 FOR PAGES 4 THRU 15
       MOV  R2,@>835E      QUOTIENT PLUS 4 (FRAMES) TO >835E
       BL   @SHWINT        SHOW THAT NUMBER (1/3 OF PAGES NUM +
4= FRAMES)
KEYEX  BLWP @KSCAN         SCAN KEYBOARD
       LIMI 2             ALLOW INTS
       LIMI 0             STOP INTS
       CB   @ANYKEY,@STATUS KEY STRUCK?
       JNE  KEYEX          IF NOT, REPEAT
       LWPI GPLWS          LOAD GPL WORKSPACE
       B    @>6A          BACK TO GPL INTERPRETER
AMSINI LI   R12,>1E00      AMS CRU BASE
       SBO  0             TURN ON AMS
       LI   R1,>FEFF       (THIS IS ->0101)
       LI   R0,>4000       START OF MEMORY
AMSLP  AI   R1,>0101       ADD 1 PAGE
       MOV  R1,*R0+        MOVE 2 BYTES TO MEM-MAPPER
       CI   R0,>4020       ALL DONE?
       JLT  AMSLP          NO, INIT MORE
       RT                RETURN
DISSTR MOVB *R1+,R2        LENGTH BYTE TO R2
       JEQ  DISX           IF ZERO, SKIPIT
       SRL  R2,8           RT JUST
       BLWP @VMBW          WRITE STRING TO SCREEN
       A    R2,R1         ADD LENGTH TO POINTER
DISX   RT
SHWINT BLWP @GPLLNK        USE GPLLNK VECTOR
       DATA >2F7C          DATA FOR INT TO STRING
       MOVB @>8361,R2      LENGTH TO R2
       SRL  R2,8           RT JUST.
       MOVB @>8367,R1      ADDR TO R1
       SRL  R1,8           RT JUST
       AI   R1,>8300       ADD OFFSET
       BLWP @VMBW          WRITE STRING
       RT
GPLLNK DATA GLNKWS
       DATA GLINK1
RTNAD  DATA XMLRTN
GXMLAD DATA >176C
       DATA >50
GLNKWS EQU  $->18
       BSS  >08
GLINK1 MOV  *R11,@GR4
       MOV  *R14+,@GR6
       MOV  @XTAB27,R12
       MOV  R9,@XTAB27
       LWPI GPLWS
       BL   *R4
       MOV  @GXMLAD,@>8302(R4)
       INCT @STKPNT
       B    @LDGADD
XMLRTN MOV  @GETSTK,R4
       BL   *R4
       LWPI GLNKWS
       MOV  R12,@XTAB27
       RTWP
WS     BSS  32             OUR REGISTERS
AMS    DATA 0              PAGES BEYOND 15
NOSTR  BYTE 19
       TEXT 'NO AMS CARD PRESENT'
PGSTR  BYTE 14
       TEXT 'PAGES FOUND = '
FRMSTR BYTE 9
       TEXT 'FRAMES = '
ANYKEY BYTE >20
       END
*
* END OF COMPLETE PROGRAM
*
```

PART TWO

```
* PART TOW OF SIDEBAR - A "SNIPPET"
*
* SUBROUTINE SETFRM (FROM AMS VIDEO TITLER)
* ON ENTRY, R3 CONTAINS THE CURRENT FRAME NUMBER (1 THRU
XX)
* WHERE XX CAN BE 9 (128K CARD), 20 (256K CARD), OR 41
(512K CARD)
* PAGE NUMBERS START WITH 0, SO FOR EXAMPLE ON A 256K CARD,
PAGES
* ARE NUMBERS 0 THRU 63, FOR A TOTAL OF 64 PAGES
*
* THE PROGRAM "KNOWS" HOW MANY FRAMES ARE AVAILABLE, AND
* WON'T USE THIS SUBROUTINE BEYOND THE AVAILABLE FRAMES.
*
SETFRM MOV  R3,R1          COPY CURRENT FRAME # TO R1
       MPY  @THREE,R3      MULTIPLY R3 BY 3
       INC  R4            ADD 1 TO PRODUCT IN R4 (PAGE NUMBER
ON AMS = 4 THRU YY)
       SWPB R4            SWAP TO LEFT BYTE
       COC  @ONEWD,R1      CHECK CURRENT FRAME FOR ODD/EVEN
       JNE  SETEVN         IF NOT 1 IN LSB, EVEN
SETODD LI   R8,>4014       MAP FOR >A000
       LI   R1,>A000       SET R1=>A000
       JMP  SETPGS         JUMP
SETEVN LI   R8,>401A       MAP FOR >D000
       LI   R1,>D000       SET R1=>D000
SETPGS LI   R12,>1E00      AMS CRU ADDR
       SBO  0             TURN ON CARD
       SBO  1             TURN ON MAPPER
       LI   R5,3          THREE WRITES TO MAKE (3 PAGES PER
FRAME)
SETLP  MOV  R4,*R8+        SET CURRENT PAGE AND ADD 2 TO  R8
       AI   R4,>100        NEXT PAGE
       DEC  R5            DEC COUNT
       JNE  SETLP          RPT IF NOT 0
       SBZ  0             TURN OFF CARD MEMORY (>4000)
       RT                RETURN
THREE DATA 3               THREE AS A WORD
ONEWD DATA 1               ONE AS A WORD
```

Extended BASIC

Tips on debugging will save you time

By JIM PETERSON

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often.

So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps to debug better than some that cost 10 times as much.

There are really three types of bugs. The first type will prevent the program from running at all — it will crash with an error message. The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a BASIC program, which uses character sets 15 and 16, in Extended BASIC.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand — such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance, that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and give you an error message— probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

> There are really three types of bugs. The first type will prevent the program from running at all. The second type will allow the program to run, but will give the wrong results. And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware.

VARIETY OF ERROR MESSAGES

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark).

```
100 PRINT "Program must be s
aved in:"merge format."
```

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program; that line may actually be correct, but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that: Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended BASIC manual list almost all the error codes, and almost all the causes of each one — it will pay you to consult these pages rather than guessing what is wrong.

VARIABLE NAMES

You may create some really bad bugs when you try to modify a program that was written by someone else — especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray — you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-looking command !@P- up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name

(See Page 27)

DEBUGGING —

(Continued from Page 26)

or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the programs to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE in 680, and you bring line 680 to the screen and find it reads:

CALL HCHAR(R,C,CH)

Just type PRINT R;C;CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended BASIC, you can even enter and run a multistatement line in immediate mode (that is, without a line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by:

```
FOR J=1 TO 100::PRINT A(J);:
: NEXT J
```

Or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped, but A will have a value of 100.

You can temporarily stop a program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and restart it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as:

```
1 BREAK 960,970,980
```

Note that in this case the program breaks just *before* those listed line numbers. You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

USING TRACE

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement, which will cause each line number to print to the screen as it is executed. If used as a command, it will trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multistatement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE( ) with the line number between the parentheses, and a subprogram after everything else:

```
30000 SUB TRACE(X):: DISPLAY
AT(24,1):X :: SUBEND
```

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping any line may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen — it may have invisible characters at one or both ends. Try PRINT LEN(M$) to see if it contains more characters than are showing; or PRINT "*"&M$&"*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything — and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program.

BUGS AND BYTES

More farewells

A TI user seeking to donate items wrote us for recipients. Following our suggestions, he wrote to Eunice Spooner, who has run a TI program for elementary school children, and Sister Pat Taylor, who has run one in a nursing home. Each wrote him back to state that she is no longer doing her TI project.

Web site for MBX

Charles Good notes that the Milton Bradley MBX system now has its own web site thanks to Rob Patton. A message he posted says, "It has been expanded in the last few days with a couple of my own MBX newsletter articles and info about a never released Milton Bradley command module game called Card Sharp." The site address is www.sundial.net/~rob/ti_mbx.htm.

TI Technical Tips

Don't forget to remove regulators when replacing PEB power supply

This is reprinted from the July 1996 TIsHUG News Digest, *published by the Sydney, Australia, users group.*

It has become alarming to hear of the amount of Texas Instrument users who are creating problems with one specific hardware update.

I am speaking of the conversion of Peripheral Expansion Box power supplies. I am a strong advocate of changing the internal power supplies to an "IBM" style of transformer. It is an improvement which is long overdue and I am glad to see it being advanced.

There is one major fact which has been overlooked by every article I have read, however. I have also noticed the fact almost everyone who has "converted" a PEB claims no depreciable reduction in heat in their "cards." The problem is created by the modification they have done.

THE PROBLEM

The regulators on the cards themselves are still being activate. They are being "back-fed" through the output leg by the cards themselves. This in effect "tells" the regulators they are still active.

The purpose/object of a regulator is to restrict voltage to a specific level and dissipate the excess voltage in the form of heat transfer. In the normal state of things the unregulated current enters the input leg, is regulated and exits the output leg to supply the "card" with constant, regulated voltage.

What is done, in essence, with the "jumpering" (connecting a small piece of wire between the "input" and "output" legs) is to tell the regulator it is being activated (through the "input" leg) and then telling it it is "not" working through the "output" leg.

It has become alarming to hear of the amount of Texas Instrument users who are creating problems with one specific hardware update! I am speaking of the conversion of Peripheral Expansion Box power supplies.

Regulators are not very smart. They are designed to work by discerning the difference between power-in and power-out. We in effect "lobotomize" their "thinking" process with the "jumpering" process. They are not designed to be activated without a specific amount of voltage supplied to the "input" leg. This is usually a figure of approximate one-and-one-half volts and above its voltage rating (in the case of a +5 volt regulator this figure would be six to six-and-one-half volts), a "mental breakdown" by introducing power to both legs. The regulator is still "active" even though you jumper it. It is still attempting to regulate the power and is creating heat.

I speak from experience on this. We (Tom Wills, Jack Mathis and I) managed to physically explode a 7805 regulator and blow a quarter-inch hole completely through a Myarc Hard and Floppy Disk Controller. The cause was from jumpering the regulator. It was an expensive and frustrating lesson.

THE FIX

The fix for this is simple. *Remove* the regulators and *then* jumper across the input/output legs. Why would you want them on there anyway? If the power supply you have installed "surges," your card is toast anyway. Since you "jumpered" the regulator the power is going right past it. Don't tell me it is so you can simply remove the jumper and install it in a non-altered box. You are better off *never* "swapping" your system cards between different systems (between a "standard" TI PE-Box and an "updated" one). I don't care how many *WARNING* labels you put on the card, you are asking for trouble.

I, in fact, recommend *all* your PE-Boxes be converted to the "IBM" style power supplies. The cost is so low and the price of repairs to your cards is so high you really can't afford not to; $30 for a power supply as opposed to $50 minimum repair cost (if shipped to an outside repair facility.

As usual, I must make the disclaimer of: *Any and all modifications/repairs to your system and its related components are done with the full assumption of owner/repair person's risk.*

Chicago TI fair set for Nov. 9

The 14th Annual Chicago TI Faire will be held November 9, 1996 at the Evanston Public Library located at 1730 Orrington Ave., Evanston, Illinois.

According to Hal Shanafield of the Chicago TI Users Group, the Skokie Howard Johnson's about two-and-one-half miles from the site will offer rooms with queen-sized beds at $68 per night and rooms with twin double beds for $80 per night, including a buffet breakfast. The Howard Johnson's will also offer free shuttle service to the Faire site. Shanafield says the group is working on plans for a Friday night get-together before the event and a banquet afterwards.

For more information contact Hal Shanafield at (847) 864-8644 or write to the Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009.

User groups listed

A publication listing computer users groups, including TI users groups, is *User Group Connection*. For information, write P.O. Box 67249, Scotts Valley, CA 95067-7249.

NIMROW

The last person able to move wins

By BARRY TRAVER

Martin Gardner's book *Wheels, Life and Other Mathematical Amusements* (New York, 1983), describes "a pleasant counter-moving game on a chessboard" (see pages 145-146). That "game without a name" appeared in the Genial TRAVelER in the following format:

One white counter and one yellow counter are placed randomly in each row of a checkerboard, the only restrictions being that the white counter be to the left of the yellow counter and that the two counters be separated by at least one empty square. Players alternate moves, a move consisting of moving one of your counters horizontally any number of empty squares in its row, forwards or backwards. Jumping your opponent is not allowed, and the last person to be able to move wins. It's that simple!

My version of the game is called NIMROW, and after the program randomly sets up the pieces on the board, you are allowed your choice of any one of four different options:

1. You can go first (against the computer).
2. You can go second (against the computer).
3. You can play against a friend.
4. You can watch the computer play against itself.

Since you get to choose whether to go first or second *after* the pieces are set up, it *should* be easy to beat the computer, but I think you'll find it otherwise! (I haven't had so much fun writing a game since I wrote "Giant and Dwarfs," which appeared way back in the April 1983 issue of *99'er Home Computer Magazine*.)

A minor variation you may want to try involves modifying line 250 in the program, which presently reads like this:

```
250 CH2=INT(RND*8)+1 :: IF
ABS(CH1-CH2)<2 THEN 250
```

The following alternative does away with the rule requiring that set-up of the counters involve at least one empty square between counters at the start of the game:

```
250 CH2=INT(RND*8)+1 :: IF
ABS(CH1-CH2)<1 THEN 250
```

Whether you like this particular variation or not is up to you.

You can beat the computer at this game, but the only hint I will give as to how to analyze it is contained in the name I chose to give to the game, viz., NIMROW. (See Gardner if you want more details of how this game is really another popular game somewhat in disguise.)

NIMROW

```
0 ! COPYRIGHT (C) 1989 by Ba
rry Traver, 835 Green Valley
 Drive, Philadelphia, PA 191
28 (phone: 215/483-1379) --
ALL RIGHTS RESERVED! !194
100 GOTO 121 :: DIM LC(16)!0
01
110 A2$,B2$,BC,BE,C(),C1,CF,
CH1,CH2,CTR,CW,D(),DD(),FF,F
F$,FLAG,I,II,J,JJ,K,M,R(),R1
,RS,RS$,RW,RWT,S,SF,SG,ST,T1
,T2,T3,T4,TYPE,TYPE$,X(),XI,
XJ,Z(),ZF,ZH !152
120 CALL ACCKEY :: CALL CHAR
 :: CALL COLOR :: CALL DELSP
RITE :: CALL KEY :: CALL LOC
ATE :: CALL MAGNIFY :: CALL
SCREEN :: CALL SPRITE !013
121 !@P- !064
122 DISPLAY ERASE ALL :: CAL
L SCREEN(13):: CALL COLOR(1,
16,1,2,16,1,3,16,1,4,16,1,5,
16,1)!042
123 CALL COLOR(6,16,1,7,16,1
,8,16,1,9,16,1,10,16,1,11,16
,1,12,16,1)!005
124 DISPLAY AT(1,1)ERASE ALL
:"NIMROW":" COPYRIGHT (C) 19
90":"  BY BARRY TRAVER": :"
 NIMROW is GENIALWARE.  If"
!106
125 DISPLAY AT(6,1):"you lik
e the program and are":"a cu
rrent subscriber to the":"Ge
nial TRAVelER or to GEnie,"
!033
126 DISPLAY AT(9,1):"you are
 not expected to send":"me a
nything for it (you have":"a
lready paid for it!).  If":"
you like it and are _not_ a"
 !118
127 DISPLAY AT(13,1):"GT or
GEnie subscriber, you":"are
requested to send $3.00":"to
 Barry Traver, 835 Green":"V
alley Drive, Philadelphia,"
!010
128 DISPLAY AT(17,1):"PA 191
28.  (Or, if you like,":"you
 can send $10.00, and I":"wi
ll send you a CONEYGAMES":"d
isk containing eight games"
!065
129 DISPLAY AT(21,1):"plus w
inning strategies for":"each
.)  Enjoy!": :"(Press any ke
y to continue.)" !026
130 CALL KEY(0,K,S):: IF S<1
 THEN 130 !200
131 DISPLAY ERASE ALL :: CAL
L SCREEN(8):: CALL COLOR(1,2
,1,2,2,1,3,2,1,4,2,1,5,2,1)!
239
132 CALL COLOR(6,2,1,7,2,1,8
,2,1,9,2,1,10,2,1,11,2,1,12,
2,1)!139
140 ! Version 1.04 of NIMROW
 COPYRIGHT (C) 1989 by Barry
 A. Traver, 835 Green Valley
 Drive, Philadelphia, PA 191
28 (phone:  215/483-1379). !
116
150 DISPLAY ERASE ALL :: RAN
DOMIZE :: TYPE$="1" !051
160 FOR I=1 TO 8 :: R(I)=16*
I+17 :: C(I)=16*I+49 :: NEXT
 I !134
170 CALL MAGNIFY(3):: CALL C
OLOR(14,9,2):: CALL CHAR(136
,"0000000000000000",138,"FFF
FFFFFFFFFFFF")!179
180 CALL CHAR(140,"00030F1F3
F3F7F7F7F7F3F3F1F0F030000C0F
0F8FCFCFEFEFEFEFCFCF8F0C000"
)!036
190 A2$="      "&RPT$(RPT$(C
HR$(136),2)&RPT$(CHR$(138),2
),4)!154
200 B2$="      "&RPT$(RPT$(C
HR$(138),2)&RPT$(CHR$(136),2
```

(See Page 30)

NIMROW —

(Continued from Page 29)

```
),4)!155
210 DISPLAY AT(1,2)ERASE ALL
:" NIMROW:  A GAME OF SKILL"
:"(C) 1988,1990 BY B.A. TRAV
ER" !047
220 DISPLAY AT(5,1):B2$:B2$:
A2$:A2$:B2$:B2$:A2$:A2$:B2$:
B2$:A2$:A2$:B2$:B2$:A2$:A2$
!114
230 DISPLAY AT(4,8):"1 2 3 4
 5 6 7 8" :: FOR I=1 TO 8 ::
 DISPLAY AT(2*I+4,6):STR$(I)
;:: NEXT I !211
240 BE=0 :: FOR I=1 TO 8 ::
CH1=INT(RND*8)+1 !141
250 CH2=INT(RND*8)+1 :: IF A
BS(CH1-CH2)<2 THEN 250 !179
260 LC(I)=MIN(CH1,CH2):: LC(
I+8)=MAX(CH1,CH2):: NEXT I !
156
270 FOR I=1 TO 8 :: CALL SPR
ITE(#I,140,16,17+16*I,C(LC(I
)),#I+8,140,12,17+16*I,C(LC(
I+8))):: NEXT I !148
280 DISPLAY AT(22,1):"
    TYPE? ";TYPE$:"1-YOU GO
1ST  2-YOU GO 2ND":"3-NO COM
PUTER 4-ALL COMPUTER" !067
290 CALL ACCKEY(22,17,"14",T
YPE):: TYPE$=STR$(TYPE):: DI
SPLAY AT(22,1):"":"":"" :: S
F=8 !075
300 IF SF=8 THEN SF=0 :: FF$
="  LEFT" :: CF,T1,SG,BC=1 :
: T2=3 :: T3=2 :: T4=4 :: SG
=1 :: BC=1 :: GOTO 320 !114
310 SF=8 :: FF$=" RIGHT" ::
CF=18 :: T1=2 :: T2=3 :: T3=
1 :: T4=4 :: SG=-1 :: BC=8 !
019
320 IF SF THEN DISPLAY AT(21
,9):"RIGHT TO MOVE" ELSE DIS
PLAY AT(21,9):"LEFT TO MOVE"
 !254
330 FOR I=1 TO 8 :: D(I)=LC(
I+8)-LC(I)-1 :: NEXT I :: IF
 BE THEN 380 !039
340 I=1 !001
350 IF D(I)<>0 THEN 380 !236
360 I=I+1 :: IF I<9 THEN 350
 !243
370 DISPLAY AT(24,1):"PLAY T
O BITTER END (Y/N)? N" :: CA
LL ACCKEY(24,27,"YN",RS):: I
F RS THEN BE=1 :: GOTO 380 E
LSE 410 !169
380 DISPLAY AT(24,1):"" :: F
LAG=1 :: FF=MAX(1,SF):: I=1
!141
390 IF D(I)<>0 OR LC(I+SF)<>
FF THEN FLAG=0 :: GOTO 450 !
111
400 IF I<8 THEN I=I+1 :: GOT
O 390 !160
410 FOR I=1 TO 8 :: DISPLAY
AT(2*I+4,2):"   ";:: DISPLAY
 AT(2*I+4,24):"   ";:: NEXT
I !227
420 DISPLAY AT(22,4):FF$;" S
IDE HAS LOST!": :TAB(4);"ANO
THER GAME (Y/N)? Y" :: CALL
ACCKEY(24,24,"YN",RS)!210
430 IF RS THEN CALL DELSPRIT
E(ALL):: DISPLAY AT(21,1):""
:"":"":"" :: GOTO 240 !088
440 CALL DELSPRITE(ALL):: DI
SPLAY ERASE ALL :: CALL SCRE
EN(13):: CALL COLOR(1,16,1,2
,16,1,3,16,1,4,16,1,5,16,1,6
,16,1,7,16,1,8,16,1)!235
441 CALL COLOR(9,16,1,10,16,
1,11,16,1,12,16,1):: DISPLAY
 AT(1,1):"I hope you enjoyed
 the game":"NIMROW.  If so,
you may want" !048
442 DISPLAY AT(3,1):"to orde
r the disk CONEYGAMES":"from
 Barry Traver, 835 Green":"V
alley Drive, Philadelphia,"
!238
443 DISPLAY AT(5,1):"Valley
Drive, Philadelphia,":"PA 19
128 ($10).  Included on":"th
at disk are _EIGHT_ GAMES":"
AND WINNING STRATEGIES FOR"
!232
444 DISPLAY AT(9,1):"EACH:
3 different versions":"of th
e classic con game ""31""":"
(using matchsticks, playing"
 !008
445 DISPLAY AT(12,1):"cards,
 or dice), ""The Game":"of G
ale"" (a.k.a. ""Bridg-it"")"
:"""Nimrow"" (checkerboard g
ame)" !182
446 DISPLAY AT(15,1):"""Penn
eyToss"" (coin tossing),":""
"Shutout,"" and ""Tic-Tac-To
d":"(Philadelphia Style).""
 All" !021
447 DISPLAY AT(18,1):"of the
se games are copyright":"by
Barry Traver (except for":"T
ICTAC/PHI, which I released"
 !238
448 DISPLAY AT(21,1):"into p
ublic domain).": : :"(Press
any key to conclude.)" !128
449 CALL KEY(0,K,S):: IF S<1
 THEN 449 ELSE DISPLAY ERASE
 ALL :: STOP !032
450 ZF=1 :: IF TYPE=T1 OR TY
PE=T2 THEN 650 ELSE GOSUB 74
0 !223
460 ZF=0 :: ON CTR+1 GOTO 47
0,590,600,620 !202
470 RW=1 !097
480 IF D(RW)=0 THEN IF RW<9
THEN RW=RW+1 :: GOTO 480 !09
6
490 IF RW<>9 THEN 540 !069
500 II=1 :: JJ=0 !023
510 IF LC(II+SF)<>BC THEN JJ
=JJ+1 :: DD(JJ)=II !028
520 IF II=8 THEN 530 ELSE II
=II+1 :: GOTO 510 !080
530 RWT=INT(RND*JJ+1):: RW=D
D(RWT):: M=-1*(INT(RND*(LC(R
W+SF)-BC)*SG)+1):: GOTO 580
!236
540 II=1 :: JJ=0 !023
550 IF D(II)<>0 THEN JJ=JJ+1
 :: DD(JJ)=II !235
560 IF II=8 THEN 570 ELSE II
=II+1 :: GOTO 550 !161
570 RWT=INT(RND*JJ+1):: RW=D
D(RWT):: M=INT(RND*D(RW)+1)!
051
580 CW=LC(RW+SF)+M*SG :: GOT
O 650 !031
590 RW=Z(1):: CW=LC(Z(1)+SF)
+X(Z(1))*SG :: GOTO 650 !026
600 IF X(Z(2))>X(Z(1))THEN Z
H=Z(2):: Z(2)=Z(1):: Z(1)=ZH
 !137
610 GOSUB 730 :: RW=Z(1):: C
W=LC(Z(1)+SF)+M*SG :: GOTO 6
50 !143
620 IF X(Z(2))>X(Z(1))THEN Z
H=Z(2):: Z(2)=Z(1):: Z(1)=ZH
 !137
630 IF X(Z(3))>X(Z(1))THEN Z
```

(See Page 31)

NIMROW —

(Continued from Page 30)

```
H=Z(3):: Z(3)=Z(1):: Z(1)=ZH
 !140
640 GOSUB 730 :: RW=Z(1):: C
W=LC(Z(1)+SF)+M*SG !050
650 IF TYPE=T1 OR TYPE=T2 TH
EN DISPLAY AT(22,3+CF):"ROW?
";ELSE 690 !136
660 CALL ACCKEY(22,8+CF,"18"
,R1):: DISPLAY AT(23,CF):"TO
 COL?";:: CALL ACCKEY(23,8+C
F,"18",C1)!021
670 IF SG*(C1-LC(R1+8-SF))>0
 OR C1=LC(R1+8-SF)OR C1=LC(R
1+SF)THEN DISPLAY AT(24,1)BE
EP:" ROW";R1;"COL";C1;"IS NO
T ALLOWED!" :: GOTO 650 !208
680 GOTO 710 !023
690 IF TYPE=T3 OR TYPE=T4 TH
EN DISPLAY AT(22,3+CF):"ROW?
";!142
700 DISPLAY AT(22,8+CF):RW;:
: R1=RW :: DISPLAY AT(23,CF)
:"TO COL?";:: DISPLAY AT(23,
8+CF):CW;:: C1=CW !171
710 II=C(LC(R1+SF)):: JJ=C(C
1):: IF II<JJ THEN ST=4 ELSE
 ST=-4 !115
720 FOR I=II TO JJ STEP ST :
: CALL LOCATE(#(R1+SF),R(R1)
,I):: NEXT I :: LC(R1+SF)=C1
 :: GOTO 300 !249
730 M=1 !005
740 FOR I=1 TO 8 :: X(I)=D(I
):: NEXT I :: IF ZF THEN 750
 ELSE X(Z(1))=X(Z(1))-M !081
750 FOR I=1 TO 7 :: IF X(I)=
0 THEN 820 !185
760 FOR J=I+1 TO 8 :: IF X(I
)=0 THEN J=8 :: GOTO 810 !20
4
770 XI=X(I):: XJ=X(J):: IF X
I=4 OR XI=5 OR XI=6 THEN IF
XJ=4 OR XJ=5 OR XJ=6 THEN XI
=XI-4 :: XJ=XJ-4 !099
780 IF XI=2 OR XI=3 OR XI=6
THEN IF XJ=2 OR XJ=3 OR XJ=6
 THEN XI=XI-2 :: XJ=XJ-2 !11
9
790 IF XI=1 OR XI=3 OR XI=5
THEN IF XJ=1 OR XJ=3 OR XJ=5
 THEN XI=XI-1 :: XJ=XJ-1 !11
3
800 X(I)=XI :: X(J)=XJ !094
810 NEXT J !224
820 NEXT I :: IF ZF=0 THEN 8
30 !246
830 CTR=0 :: FOR I=1 TO 8 ::
 IF X(I)<>0 THEN CTR=CTR+1 :
: IF ZF THEN Z(CTR)=I !245
840 NEXT I :: IF ZF THEN RET
URN !189
850 IF CTR<>0 THEN M=M+1 ::
GOTO 740 !095
860 RETURN !136
870 !@P+ !062
880 SUB ACCKEY(R,C,FL$,CH)::
 CALL GCHAR(R,C+2,DF):: DISP
LAY AT(R,C)BEEP:CHR$(DF);::
CTR=0 !137
890 CALL KEY(0,K,S):: CTR=CT
R+1 :: IF CTR=5 THEN CALL HC
HAR(R,C+2,30)!004
900 IF CTR=10 THEN CALL HCHA
R(R,C+2,DF):: CTR=0 !166
910 IF S<1 THEN 890 ELSE IF
K>96 THEN K=K-32 !196
920 IF K=13 THEN K=DF !254
930 IF FL$="YN" THEN IF CHR$
(K)<>"Y" AND CHR$(K)<>"N" TH
EN 890 ELSE CALL HCHAR(R,C+2
,K):: CH=K :: IF CH=89 THEN
CH=1 :: SUBEXIT ELSE CH=0 ::
 SUBEXIT !006
940 IF CHR$(K)<SEG$(FL$,1,1)
OR CHR$(K)>SEG$(FL$,2,1)THEN
 890 !122
950 CALL HCHAR(R,C+2,K):: IF
 K>64 THEN K=K-64 !126
960 IF K>48 THEN K=K-48 !193
970 CH=K :: SUBEND !190
```

Copying cartridges to disk

No special hardware required to download ROM

By OLDEN WARREN

This article appeared in the May 1996 issue of Bytemonger, *the newsletter of the Bluegrass 99ers of Lexington, Kentucky..*

Back in June 1995 I published an article detailing how to transfer cartridge programs to disk. After trying it, I decided that I could improve upon the technique and did so, writing my own companion programs to aid in the process. Some time later I demonstrated this process at our users group meeting.

Recently, there has been some interest in transferring cartridges to disk. I read a column in another group's newsletter showing another method for doing this. Not only do I show you how to load it to disk, I show you how to make the program load into memory or to a Supercart when it is run. Since I never shared my method with the public at large, I thought I would do it now. This is not to say it is any better than any other method; it is just an offering of a different method that may be of some help to someone else.

All of the programs mentioned in the instructions can be found on a disk I've created called "Cartridge Copier." All of the programs contained on it are not mine (i.e., E/A Debug, Disassembler), so I cannot sell the disk. I would be more than happy to provide anyone who wants copies of my programs, including a file with the instructions shown below.

BASIC INSTRUCTIONS

You need an Editor/Assembler (E/A) cartridge, Extended BASIC, the "COPIER" disk and a cartridge with ROM chips only. You can find the ROM chips by opening it up and looking at the chips. If it has all large chips, then it is ROM only. If it has any small chips then you have GROM, which cannot be copied with this method. A Widget can be used to switch between cartridges or you can cover Pin 1 of the cartridge being copied with tape. (Pin 1 is the rightmost pin on the bottom side). Also, a printer would be convenient to have, but not necessary.

1) Cover Pin 1 of the cartridge being

(See Page 32)

COPYING CARTRIDGES —

(Continued from Page 31)

copied (not necessary if you have a Widget).

2) If you have a Widget, plug E/A in Slot 1, the cartridge you want copied in Slot 2 and Extended BASIC in Slot 3.

3) Make sure you have a totally blank disk to copy the cartridge on.

4) Switch over to the E/A cartridge (either with the Widget or just insert it), insert the "COPIER" disk into Drive 1, select "LOAD AND RUN" and enter "DSK1.DEBUG"; when it is loaded, press Enter and type "DEBUG" for the program name.

5) When the debugger prompt appears, switch over to the cartridge being copied (Slot 2 on the Widget or insert it with Pin 1 covered). Make sure the computer doesn't reset. If it does, make sure Pin 1 is deactivated and start over. Now, make sure the debugger is still operative (press Enter a couple of times).

6) Now type in "M6000 8000" and press Enter. The screen should start scrolling upward and display different kinds of numbers. If all you get is "6000=00 00 00 00 00 00 00 ************" or zeroes only, make sure you have a ROM-only cartridge and that you are switched over to the cartridge being copied (or that it is inserted). You will probably see the title of the cartridge in the first few lines on the right. Now, let the screen display the numbers for a while and make sure that everything is OK. Make a note of the last address used in the program (zeroes will trail the last used address).

7) Type in "M6000 6080" and press Enter. Address >6006 contains the pointer to the program's starting address. Add 2 to this number (remember it is in HEX) and write down the number in that location. This will be used in Step 15. Press QUIT to go back to the main title screen.

8) Load the E/A, select "LOAD AND RUN" again and enter "DSK1.DISASM"; when it is loaded, press Enter and type "START" for the program name.

9) The disassembler title screen should appear. Now, switch over to the cartridge being copied (or plug it in again). Make sure it doesn't reset or otherwise stop the machine. For the starting address enter "6000"; for the ending address enter "67FF"; insert the blank disk, type "DSKx.CART1" for the device name (where "x" is the drive number that the blank disk is in) and press Enter to start the disassembly process. The disassembled code should appear on the screen. To pause the disassembling, press any key; press it again to continue. When the number on the left equals >6800 (or around that number) the screen will stop scrolling. Press Enter, QUIT, then Enter twice for the E/A menu.

A Widget can be used to switch between cartridges or you can cover Pin 1 of the cartridge being copied with tape.

10) Press QUIT. Switch to the Extended BASIC cartridge, load it and run the program "DSK1.TRIM" from the "COPIER" disk. For the source file input "DSKx.CART1"; for the object file input "DSKx.CART11"; if you receive an error, check to see if you have the right disk inserted and that the source and object files are valid. If that isn't it, check to see if you have enough disk space available. When done, press QUIT.

11) Switch to the E/A cartridge, select the Editor and load in the CART11 file.

12) When loaded press FCTN-9 to get the Editor command list. Select "R" for replace, then type in "V,1000/6/A/" and press Enter. Now, press "Y" whenever the cursor stops on an instruction (i.e., LI, JMP, SB, CB, B, A, BLWP) that has a "6???" in it. Don't change DATA statements or ones that have only two digits (e.g., "6?"). After completing the last one, repeat this step using "V,1000/7/B/" as the replacement string. When finished, save this file in VARIABLE 80 format on the Editor command list.

13) Do Steps 8-13 again, changing the starting address in the disassembly step to 6800, the ending address to 6FFF and the filenames to "CART2" and "CART22"; the third time, use 7000, 77FF, "CART3" and "CART33"; the fourth time use 7800, 7FFF, "CART4" and "CART44"; you only need to do this enough times to cover the last used address (e.g., if the last used address is >6980 then you only have to do it twice).

14) Switch to the E/A cartridge, select "LOAD" and type in "DSK1.REASM"; when it is loaded, select "EDIT" and change all of the lines that contain "DSK2" to "DSKx" where "x" represents the disk drive that contains the CART files. Also, delete any "COPY" lines not needed (i.e., if you only created CART11-CART33, then delete the COPY "DSK2.CART44" line). Modify the "START" line to say "START EQU >Axxx" where "xxx" is the last three numbers of the address you wrote down from Step 7. Save this as "DSKx.CARTS" in VARIABLE 80 format. Load the Assembler, enter "DSKx.CARTS" as the source file and "DSKx.CART" as the object file and assemble under the "R" option. When it is finished, press Enter to return to the E/A menu, select "LOAD AND RUN" and enter "DSKx.CART"; the program should start automatically.

If it doesn't look right, use the editor of the disassembled code to see if lines are left out. Correct any problems, reassemble.

USING A SUPERCART

If you have a SuperCart (8K RAM and E/A chip in a cartridge) you can reduce the number of cartridge switches necessary in the basic instructions. This is particularly helpful if you do not have a Widget. After Step 2 follow the steps below:

a) Switch over to the E/A, select "RUN PROGRAM FILE" and enter "DSK1.COPYCART".

b) When it is loaded, insert the cartridge to be copied (or switch to it with the Widget) and press Enter.

c) At the next prompt, insert the SuperCart (or switch to it with the Widget) with its write protection OFF and press Enter.

d) At the next prompt press Enter to

(See Page 33)

COPYING CARTRIDGES —

(Continued from Page 32)

confirm that the program is complete, enter "N" at the "RERUN" prompt and press Enter once more to return to the E/A menu.

The cartridge has now been copied into the SuperCart. To ensure the process worked properly, return to the computer title screen with the SuperCart inserted. The cartridge name should appear as one of the program options. Select the program to see if it runs normally. If not, make sure that there are no GROM chips in the cartridge being copied. Proceed from Step 3. Any time the instructions ask for the E/A or the cartridge being copied the SuperCart can be left in place.

MAKING PROGRAM LOAD INTO A SUPERCART

If you want the copied program to load into a SuperCart, adjust the instructions as follows:

a) Skip Steps 11 and 12.

b) In Step 14, when modifying the "START" line change ">Axxx" to "6xxx" where "xxx" is the last three numbers of the address you wrote down from Step 7. After assembling the program, select the "LOAD AND RUN" option and enter "DSKx.CART" as the filename with the SuperCart inserted (with write protection OFF). After it has loaded, the program should start automatically. When you return to the title screen the cartridge name should appear as one of the program options.

The author, president of the Bluegrass 99 Computer Society, can be reached at 4016 Weber Way, Lexington, KY 40514.

MICRO-REVIEWS

Loadmaster 2.1, Extended BASIC 2.4g, Speak DV80, ET and His Adventure on Land, Archiver 4.0g, Touchdown 96

By CHARLES GOOD

This month some of my reviews will be updates of software I have previously reviewed. These updates were acquired during my visit to the Cleveland MUG conference in May. The Cleveland groups did a really great job organizing this event. Attendance was high and everything was free, including the food. It is a good thing we are not fans of the ADAM computer. ADAM enthusiasts are having a computer show, AdamCon VIII, in the Cleveland area in September. The cost of admission to the entire three-day event, which includes food and lodging, is $260!. Daily admission with lunch is only $19. The Adam home page says that 42 people showed up for the equally expensive AdamCon 7. The Cleveland MUG attendance was well over 100. TIers are very lucky to have groups willing to organize free or low-cost TI computer shows.

LOAD MASTER updated to v2.1 by Mickey Cendrowski

My review of v1.2 a few months ago was kinda lukewarm. That is because at that time the product didn't seem very special. It would display a disk directory, view text files from the displayed directory and run Extended BASIC software from the directory. Now there is much more.

The new feature added to v2.1 that makes this a really useful product is file identification. No other 99/4A product does this so extensively. When Load Master reads a disk directory it compares the characteristics of each file to its internal database and makes a very good guess as to the specific nature of the file. Here are the file types recognized: Character file (as in CHARA1), TIPS font, Infocom game, FORTH screens, Editor/Assembler option 5, Artist border (refers to TI Artist), Artist picture, Artist instance, Artist slide, Artist movie, Artist vector, TI-Base command, TI-Base help, TI-Base structure, XB merge type runnable program, Editor/Assembler option 3, Archived file, Page Pro banner, Page Pro large font, Page Pro small font, Page Pro line font, Adventure game file, TOD (Tunnels of Doom) adventure file, JP border (I am not sure what "JP" is), JP font, JP image, JP pattern, Firstbase dictionary, Firstbase data and Firstbase index.

The on-screen disk directory reports a file type from the above list next to each file name in addition to the usual type of file information (length, name, etc.). You can print these directories in various ways. For example, you can make a disk jacket with all directory info including file type printed on the jacket. Or, you can print an insert designed to slip inside or be pasted on the outside of a disk jacket. You can also print directory labels or save directory information to a disk file, all with file types. Speaking of printing, you can also print labels with four lines of user defined text, and you can have the software control two printers. Load Master is designed to work with a serial and a parallel printer, sending some output to one and some to the other if you want.

Load Master is almost entirely written in Extended BASIC and determines file type by examining the kind of file (PROGRAM, DV80, IF128, etc.), file name, and file length. It does not look at file headers, which can't be done from straight Extended BASIC, and thus has trouble identifying many types of "program" files. Most of these are identified as "XB-BAS-EA5-ADV" because it can't tell them apart. If the file is program and 34 sectors long then "GROM" replaces EA5 in the above list. Thus, Load Master isn't perfect in its file type identification. It is, however,

(See Page 34)

MICROREVIEWS —

(Continued from Page 33)

the only TI product currently available that attempts such detailed file identification. A much older software product, IDENTAFILE, by J. P. Hoddie, was published before many of Load Master's identifiable file types ever existed. If you want to distinguish between "program" files that are XB, E/A5 or neither, then I recommend Funnelweb's Disk Review.

Load Master is quite slow and thus probably not really suitable for a quick disk catalog. Error checking is supposed to be added to future versions. Right now it is sometimes possible to crash the program. The Load Master author suggests that you run your entire disk collection through Load Master and make disk inserts or jackets. I think this is a good idea, and the main reason that most folks will have for using Load Master. Load Master v2.1 is "send the author what you think it is worth" shareware. Send me $1 and I will mail it to you to try out on a SSSD disk.

SPEAK DV80 assembly language version by Tony Knerr and Charles Good

I have previously reviewed my DV80 file speaker. Written in Extended BASIC, it loads TI's Text-to-Speech, asks you for the name of a text file to read, and then speaks the text file. Tony made an assembly language version of my program that is much faster in all respects but otherwise identical to my original.

On my Geneve, from disk, my original takes 33 seconds to load itself and all the Text-to-Speech files. Tony's version takes 9 seconds. My original XB version loads in a line of text and then pauses because the text has to be converted to upper case so it can be spoken by Text-to-Speech. These pauses come at the end of each text line, which does not necessarily correspond to a natural break in the middle or end of a sentence. Thus, the pauses in speech can be annoying. Similar pauses occur in Tony's version, but they don't last nearly as long, so the speech sounds more natural.

Send me $1 and I will send you Tony's very fast DV80 file speaker on a SSSD disk.

EXTENDED BASIC updated to v2.4g, by Tony Knerr

I have previously reviewed v2.3, which is mainly for use by 99/4A systems that have a GRAM device. V2.4g is only for Geneve users. This Geneve version has everything found in version 2.3 plus some Geneve-specific niceties. These include:

• *CALL HFDCON and CALL HFDCOF*. These turn on and off the HFDC EPROM, or the EPROM in the Cor-Comp and Myarc floppy disk controller. This gives you access to the CALL subprograms found in these EPROMs such as DIR, ILR, LLR, LR, FILES, MDM and DT in the HFDC EPROM.

• *CALL SPEED1 and CALL SPEED5*. I really like these! They let you set normal 99/4A speed (SPEED1) from within Extended BASIC either in command mode or from a running program. You no longer have to exit XB and go to the Geneve's GPL interface to set speed. This is very useful for games. The Geneve makes most software run much faster then it would on a 99/4A. For some games this is too fast. You can slow the game down to normal 99/4A speed with CALL SPEED1.

• *The BREAK* (fctn/4) now works on the Geneve. It also works in XB v2.3 and in the latest version of RXB, but doesn't work from the GROM file version of the official TI Extended BASIC.

The following nice enhancements to regular Extended BASIC carry over from v2.3:

— CALL CHARSET resets both upper *and* lowercase characters to original TI characters and default foreground and background colors.

— CALL ALSET resets to original TI XB characters but not colors.

— CALL CHARA1 resets to v2.4g resident characters, with true lower case characters.

— CALL LRGCLPS loads the large capital char set displayed on the TI title screen.

— CALL BYE and CALL QUIT both exit to the title screen from within a running program or command mode.

— CALL HELP shows these enhanced commands on screen.

— CALL CRASH, CALL NYANYA, CALL HONK, CALL CHIME, and CALL BEEP produce sounds.

— CALL GPEEK reads GROM memory.

— CALL MLOAD loads and optionally runs an E/A5 program.

— CALL MOVE moves blocks of memory.

— CALL MSAVE saves a memory image block of memory to disk.

— CALL SPROF and CALL SPRON turn off and on sprite motion.

— CALL VPEEK and CALL VPOKE will read and write to VDP memory.

— CALL WAIT causes a user-defined delay.

— CALL XB restarts Extended BASIC and looks for DSK1.LOAD, which you can bypass by holding down any key.

This software should be in the library of every Geneve owner. If you have a hard disk you should put it there and make it available as one of the several flavors of extened basic now available to Geneve users. The software is public domain. Send me $1 and I will mail it to you on a DSSD disk.

ET AND HIS ADVENTURE ON LAND by Hank Mishkoff

This is a long-lost piece of 99/4A command module software that existed only as source code buried in the author's closet. It has recently been assembled into a more or less runnable educational game. In 1983 TI produced its own ET frogger-like game and contracted with Looking Glass Software to produce a series of educational command module cartridges based on the movie character ET. None of the TI or Looking Glass ET games were ever officially released. Published references from that time refer to games called ET, ET at Sea and ET and His Adventure on Land. The first two were finished and exist in user group libraries as GROM files playable with a GRAM device and as slightly buggy assembly files playable from Extended BASIC. Until very recently, nobody had ever seen the "on land"

(See Page 35)

MICROREVIEWS —

(Continued from Page 34)

game and it was assumed to have been lost or never completed.

One day last fall Hank Mishkoff logged onto the Internet newsgroup comp.sys.ti and indicated that he had been involved in software development for the 99/4A. It turns out that he wrote several educational software command modules, and wrote the music found in the Music Maker module. We exchanged a number of e-mail messages which formed the basis of a really interesting interview published in the November 1995 issue of the Lima newsletter. As a result of this exchange Hank dug into his closet and produced the original software development disks and notebook for the ET and His Adventures On Land project, long forgotten since 1983. Hank wrote the game as an independent contractor for Looking Glass Software, but when TI quit the 99/4A Hank was not paid for his efforts, so his disks went in the closet and were forgotten. The disks contain assembly source code and, as indicated in the Lima newsletter article, Hank gave me permission to freely distribute the material.

This is an ecology/nature game. The code is incomplete, not implementing all the details described in the project development notebook, and there is no sound. (The notebook indicates that there was also going to be an "ET and His Adventures in Air" game.) ET wants to get picked up by his spaceship. The spaceship needs a place to land and there are several possible landing sites, each occupied by animals. ET has to move all the animals through a maze of trails to their proper homes and give them some food. The proper animal has to go to the proper habitat and be given the correct kind of food before ET's spaceship can land. There are three habitats, forest, jungle and desert. Within each habitat a submenu asks you to: 1-take the animals home, 2-feed the animals, and 3-clear the forest. You have to do all these things correctly in at least one habitat before ET's spaceship can land and take him away to his home planet. There are three skill levels. Some of the animals you encounter are snake, skunk, frog, spider, lion, giraffe, fish, parrot, elephant and monkey. As you move the animals with the arrow keys or joystick they make realistic motions. The frog hops, for example.

Some of the details just aren't quite right, probably because the code isn't complete. The proper food for the frog, for example, sometimes seems to be a rabbit. I don't think so! Some parts of the game just don't work at all.

The title screen says c1993 Texas Instruments. This is not true. Neither TI nor Looking Glass Software ever paid for the code, so it remains the property of Hank Mishkoff, who has authorized its unrestricted distribution. The code has been assembled into files that run as E/A5. Send me $1 and I will send you these E/A5 files along with the Lima newsletter article on the same SSSD disk that contains Archiver v4.0g.

ARCHIVER updated to v4.0g by Tim Tesch

This is the same old Barry Boone archiver we all use and love, updated by Tim Tesch to support long path names. Contrary to what you might expect from the suffix "g," this software is not Geneve-specific. It will work on a 99/4A or Geneve system with or without hard drive. It is, however, specifically useful on systems with hard drives. It works fine on my SCSI hard drive and Tim has developed and tested it on a system with an HFDC hard drive controller.

The only parts of the new archiver that allow input of long path names are "Extract Files" and "Catalog Ark File." You are still limited to "DSKx" when you want to pack an archive. When you select "Extract Files" you are first prompted for a path name of the file to unpack. This means only the path that contains the file, without the file name. The second prompt asks you for the file name. This fooled me for awhile. I kept getting errors when I gave the complete path name at the first prompt. The third prompt asks for the path where you want to put the unarchived files.

Archiver was originally written by Barry Boone, and most of us have already sent Barry a shareware donation. Tim asks for a "whatever you think it is worth" additional donation if you have already paid Barry Boone. This software should be in the hands of all hard drive users. Send me $1 and I will mail it for you to try out on a SSSD disk along with the ET game described above.

TOUCHDOWN 96 by Gene Hitz

This is a really unusual software product. I know of nothing similar. If you are lucky, the cost of this commercial software can be taken as an itemized deduction on your income tax. The previous sentence is true.

Touchdown 96 predicts the outcome of National Football League games week by week, complete with point spread. It will also give you a list of team ratings based on past performance. When you purchase the product, the current season game schedule and team ratings, correct to the week the software is mailed to you, are already entered on your disk by the software author. Each week you update the program's data bank with the scores of the previous week's games. This accumulating data allows the program to predict the outcome of the next week's games. The program can be used indefinitely season after season. The previous season's ratings are used as a starting point for a new season. For 1997 all you have to do to get started is input the 1997 week-by-week game schedule when it becomes available.

In making its predictions the program uses win/loss records, point spread, home vs. away and values which it calculates for itself called momentum and rebound factors. At any time you can get screen displays and printouts of predictions for any week's games, the current team rankings as calculated by the program and the conference standings as determined by win, loss and tie records. Predictions are most valid for the next week's games, but can be extended to future weeks.

The program has a very professional on-screen look. There is lots of error checking, so it is difficult to make an incorrect entry. It makes heavy use of John Bull's XB windows routines and lots of

(See Page 36)

MICROREVIEWS —

(Continued from Page 35)

spiffy looking overlapping windows pop up all over the screen. There is also a neat looking football-shaped cursor. There are on-line help screens, which you really need sometimes. I find the documentation kind of confusing, perhaps because I am not much of a sports or gambling fan.

Here is the sequence of program operation in order to generate a list of predictions in case you can't figure it out from the docs. Go to the Setup window and Load data, then exit the Setup window by pressing Q. Go to the Files window and Load file. This is the schedule file. Pick the season week number. When the schedule for that week is loaded press Q to get back to the main menu. Go to the Program Operations window and select Predictions. The predictions will be displayed on screen and you will be given the opportunity to make a hard copy.

How well does it work? I don't know, but we will find out. I am writing this review at the end of July 1996. Most of you will receive your September/October 1996 MICROpendium magazines the last week of September. Below is the predicted outcome of the first week's NFL games, most of which will be played Sunday, Sept. 1, 1996. These predictions are based completely on data from the 1995 season: San Francisco by 7.83 over New Orleans. Dallas by 7.63 over Chicago. Denver by 6.23 over New York. Indianapolis by 4.86 over Arizona. Pittsburgh by 4.8 over Jacksonville. Miami by 4.31 over New England. Green Bay by 3.03 over Tampa Bay. San Diego by 2.94 over Seattle. Kansas City by 1.25 over Houston. New York by 0.75 over Buffalo. Baltimore by 0.6 over Oakland. Minnesota by 0.58 over Detroit. St. Louis by 0.42 over Cincinnati. Washington by 0.37 over Philadelphia. Carolina by 0.01 over Atlanta.

Here are the predictions for week 2, most games played Sunday, Sept. 8. These predictions, based only on last season's data, would be more accurate if data were available from the week 1 games, but that is not possible since I am writing this in July: San Francisco by 10.22 over St. Louis. Green Bay by 10.19 over Philadelphia. Detroit by 6.98 over Tampa. Dallas by 6.64 over New York. Pittsburgh by 6.07 over Baltimore. Kansas City by 5.02 over Oakland. San Diego by 4.38 over Cincinnati. Buffalo by 3.61 over New England. New Orleans by 3.19 over Carolina. Miami by 1.94 over Arizona. Houston by 1.89 over Jacksonville. Indianapolis by 1.7 over New York. Seattle by 1.36 over Denver. Atlanta by 1.06 over Minnesota. Washington by 0.49 over Chicago.

Touchdown 96 costs $10, which includes shipping and handling and the most recent week's data entered on your SSSD disk. Annual updates with any added or improved options and containing current data are available for $5 to registered owners. The software is Geneve, 99/4A, and RAMdisk compatible. It is entirely written in Extended BASIC.

Oh yah. About that income tax stuff. The feds require that any net gambling profits, even illegal gambling profits, be listed as miscellaneous income on your federal income tax report. However, any direct expenses incurred in obtaining these winnings, such the purchase cost of Touchdown 96, can be deducted from these winnings.

ACCESS

- Tim Tesch (Archiver 4.0g), 1856 Dixie Rd., Port Washington, WI 53074.
- Tony Knerr (XB v2.4g, Speak DV80 assembly version), 17 Marshall Circle, Downington, PA 19335. Phone (610) 269-7447.
- Mickey Cendrowski (Load Master v2.1), 100 Pine St., Russellton, PA 15076.
- Gene Hitz dba Program Innovators (Touchdown 96), 4122 W. Glenway, Wauwatosa, WI 53222-1116.
- Charles Good (your humble reviewer and distributor of $1 disks), P.O. Box 447, Venedocia, OH 45894. Phone (419) 667-3131. Preferred e-mail address good.6@osu.edu (other previously published e-mail addresses still work).

USER NOTES

ABASIC cursor

Jack Mathis, of the Southwest 99ers, has advice for Geneve users who want to change the shape of the Advanced BASIC cursor.

Using a sector editor on the ABASIC1 file, search for hex >1962 015F. Change it to >1962 011E. This changes the cursor from the underline (character 95) to character 30, a standard TI-BASIC cursor character.

Another way to load

Phil Van Nordstrand, of Seabrook, Texas writes:

As I have been playing around with c99 recently, I was very interested in Dave Ormand's article in the November 1995 MICROpendium. There is one change that could be made in Ormand's program loading recommendations. That is where he says to load the library function CFIO because the program uses the STDIO file. That apparently is not necessary when you use the PRINTF file. If you convert the DF80 E/A3 program to an E/A5 program, you will notice an extra four sectors are required when using the CFIO file. I discovered this by accident when I converted one version of a program to the E/A5 program format using CFIO and another version of the same program without CFIO and found they both worked properly even though one was four sectors shorter.

Quick change to D/V80

The following was written by Andy Frueh.

If you have a D/V40, 132 or other display/variable text file that you need to edit with TI-Writer, you can change it to a D/V80 file by using a sector editor. (Remember, TI-Writer loads only text files with 80-column line lengths.)

(See Page 37)

USER NOTES

(Continued from Page 36)

Find the header sector of the D/V file you want to change. Edit that sector in hex mode. Go to byte >11. If the file is D/V40, you should see the number 28 — an open-parenthesis "(" in ASCII. To change it to 80, type 50 over the 28 — "P" in ASCII.

The importance of pitch and slope with speech and TEII

The following was written by Don Grin. We found it in The Computer Voice, the newsletter of the Southern California Computer Group.

Here is an example of automatic speech of an imaginable words with just two lines:

```
10 OPEN #1: "SPEECH",OUPUT
20 PRINT #1: "THANK YOU AND G
OD BLESS THE UNITED STATES O
F AMERICA"
```

Set the pitch (highness 0-63) and slope rate (0-255):

```
10 OPEN #1: "SPEECH",OUPUT
20 PRINT #1:"..50 160"
30 PRINT #1: "SPEAKING AT 5 0
PITCH AND 1 6 0 SLOPE"
```

If pitch and slope are not indicated, the computer will speak at 43 pitch and 128 slope. Zero pitch will produce a whisper. The ideal slope is defined in the manual as 10 percent of pitch, round it, then multiply by 32.

To avoid garbled speech, stay below the following maximum slope (p is pitch). Slope is not to exceed:

POS(16xPOS(32-p)-496)

where POS is a positive value.

Here is an example: If pitch is 50, the slope shouldn't exceed:

POS(16xPOS(32-50)-496) =
POS(16x18-496) = 208

Test the program for pitch, slope, and phrase:

```
10 OPEN #1:"SPEECH",OUTPUT
20 INPUT "INPUT PITCH":A$
30 INPUT "INPUT SLOPE":B$
40 PRINT #1:"//"&A$&" "&B$
50 INPUT "INPUT PHRASE":C$
60 PRINT #1:C$
70 GOTO 20
```

Inflection symbols can precede words for special stressing of words:

Use ^ to stress primary word (1 per line).

(See Page 38)

Fig. 1

Program line	Comment
`10 OPEN #1:"SPEECH",OUTPUT`	
`20 PRINT #1:"^SUPER"`	primary word emphasis
`30 PRINT #1:"^>SUPER"`	primary word and syllable emphasis
`40 PRINT #1:"WHAT TIME IS IT"`	statement format
`50 PRINT #1:"_WHAT TIME IS_IT"`	question format
`60 PRINT #1:"_WHAT TIME^IS_IT"`	another question format

1996 TI FAIRS

MARCH

1996 TI Workshop, TI99/4A User Group U.K., March 16, Wheatsheaf Public House, Sandbach, Cheshire, England. Contact Trevor Stevens, chairman, 249 Southwell Rd. East, Rainworth, Notts, NG21 OBN, UK, or call the MOBB BBS at 01623 491282.

Dutch TI Users Group Annual Meeting, March 23, Buusthuis Kremerstraat 241 Utrecht, The Netherlands. Contact Berry Harmsen, chairman, 1E Oosterparkstraat 141E, 1091 GZ Amsterdam, The Netherlands, (phone) (31) 20-6941047.

NOVEMBER

11th International TI99/4A and Geneve Computer-Treffen, Nov. 1-3, Freizeitheim Vorsfelde, Am Sportplatz 5, D-38448 Wolfsburg, Germany. Contact Martin Zeddies, Haupstr. 28, D-38448 Wolfsburg-Reislingen, Germany. Phone/fax number +Germany-5363-71125.

14th Chicago TI International World Faire, Nov. 9, Evanston Public Library in Evanston, Illinois. Contact Hal Shanafield, Faire chairman, at (847) 864-8644, or Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009.

1997 TI FAIRS

APRIL

Fest West '97, April 5, San Jose Civic Auditorim, San Jose, California. Contact Fest West '97 c/o Don O'Neil, 3297 Woody Lane, San Jose, CA 95132, or call (408)-934-0352.

MAY

Multi Users Group Conference, May 23-24, Ohio State University, Lima Campus. Contact Charles Good, P.O. Box 447, Venedocia, OH 45894. Phone (419) 667-3131. Preferred e-mail address good.6@osu.edu.

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.

USER NOTES

(Continued from Page 37)

Use _ to stress secondary words (no limit per line).

Use > to stress a syllable of a word.

See Fig. 1 for inflection examples.

You can use the allophone option to create words right from original sounds. Here is a program you can try:

```
10 OPEN #1:"SPEECH",OUTPUT
20 OPEN #2:"ALPHON",INTERNAL
30 PRINT #1:"I WILL CREATE SK
ULL."
40 A$=CHR$(120)&CHR$(103)&CH
R$(70)
50 PRINT #2:A$
60 PRINT #1:"I WILL CREATE PU
LL."
70 B$=CHR$(109)&CHR$(71)
80 PRINT #2:B$
```

Here is a program to print allophone codes from a word:

```
10 OPEN #1:"SPEECH",OUTPUT
20 OPEN #2:"ALPHON",INTERNAL
30 INPUT "PHRASE :":A$
40 PRINT #1:A$
50 INPUT #2:B$
60 PRINT "INTERNAL CODE IS: "
:BK$
70 PRINT "ALLOPHONE CODES FOR
";A$;" ARE; "
80 L=LEN(B$)
90 FOR I=4 TO L
100 PRINT ASC(SEG$(B$,I,1))
110 NEXT I
120 GOTO 30
```

There are fewer allophone speech codes needed for a word than speech codes with SPGET. The manual lists character codes for various allophone sounds as well as various stress, pitch, and slope options.

Finding E/A start words

Have you ever tried to run an Editor/Assembler program but couldn't because you didn't know the "start" word? Here's one way to find that start word:

From Editor/Assembler, load the program into memory. Go back to the title screen and enter E/A BASIC. Type and run the following program:

```
10 FOR I=16128 TO 16383
20 CALL PEEK(I,A)
30 PRINT CHR$(A)
40 NEXT I
```

Among the words, symbols, and other items that appear you should find the start word you are looking for. It's worth a try.

Uploading text files using Telco

The following was written by Glenn Bernasek of the TI-Chips User Group of Cleveland.

This article is in response to a request from the multi-user group officers confer-

(See Page 39)

USER NOTES

(Continued from Page 38)

ence at the 1996 TI99/4A and Geneve Multi-User Group Conference in May.

Uploading ASCII text files onto the Internet with Telco is a simple, two-step process.

Step 1: Configure the Telco file transfer setup routine (Choose "Setup options" from the main menu, then choose "File transfer setup" and set the defaults as follows:

A)	Abort downloads	:	Discard
B)	Default error check	:	CRC
C)	Echo locally	:	Off
D)	Blank lines	:	Off
E)	Character spacing	:	000
F)	Line pacing	:	010
G)	Pace character	:	000
H)	Strip leading space	:	Off
I)	Line by line send	:	Off
J)	Send at end of line	:	CR
K)	CR translation	:	Strip
L)	LF translation	:	Strip

Press Fctn-9 and save the changes.

Step 2: Upload a text file to the Internet while in Telco. Follow these instructions after you've gone online:

1. Press Fctn-6.
2. Choose ASCII option by pressing Enter.
3. Type in full text filename (DSKn.FILENAME) and press Enter.
4. Text file is then uploaded. You should see the line numbers counted at the bottom of the screen as the file is uploaded.
5. After the file is uploaded, Telco will prompt you to press a key.
6. The last line of your text file will be seen at the bottom of the screen. Type ### and press Enter to tell the Internet that you're finished sending.
7. Choose the post or send option and you're done.

That's all there is to uploading ASCII files to the Internet. Let's start sharing and talking.

Alternatives to CALL CLEAR

This comes from the Northern Nevada Ninety-Niners newsletter. It was written by Matt Patrovsky.

Are you tired of the CALL CLEAR statement? Sure, it's effective, but it doesn't seem very exciting or different. Try these out, they should spice things up just a bit.

It does help if you having something on the screen, so type CALL HCHAR(1,1,64,768) before you run each example.

Example 1

```
10 CALL HCHAR(1,1,32,768)
```

Example 2

```
10 CALL VCHAR(1,1,32,768)
```

Example 3

```
10 FOR I=24 TO 1 STEP-1
20 FOR Y=32 TO 1 STEP-1
30 CALL HCHAR(X,Y,32)
40 NEXT Y
50 NEXT X
```

Example 5

```
10 FOR X=1 TO 24
20 CALL HCHAR(X,1,32,32)
30 NEXT X
```

Example 6

```
10 FOR X=24 TO 1 STEP-1
20 CALL HCHAR(X,1,32,32)
30 NEXT X
```

Example 7

```
10 FOR Y=1 TO 32
20 CALL VCHAR(1,Y,32,24)
30 NEXT Y
```

Example 8

```
10 FOR Y=32 TO 1 STEP-1
20 CALL VCHAR(1,Y,32,24)
30 NEXT Y
```

Example 9

```
10 X=1
20 Y=1
30 XXZ=24
40 YYZ=32
50 CALL HCHAR(X,Y,32,32)
60 CALL VCHAR(X,Y,32,24)
70 CALL HCHAR(XXZ,Y,32,32)
80 CALL VCHAR(X,YYZ,32,24)
90 X=X+1
100 Y=Y+1
110 XXZ=XXZ-1
120 YYZ=YYZ-1
130 IF X>12 THEN 140 ELSE 50
140 END
```

CLASSIFIED

FOR SALE

SPEED READ

A new program for TI or Geneve owners improves reading speed. Can use *any* D/V 80.text files. Two versions: For U.S. and Canada (60 Hz) it's $5.00 including S&H. For overseas (50 Hz) $7.00. Send cash, check or Money Order to Bruce Harrison, 5705 40th Place, Hyattsville MD 20781. v3n5

NEW FOR AMS

Harrison's Video Titler for AMS. Uses the AMS to store video frames so that many frames can be "wiped" in sequence to your VCR. Priced at only $10.00 including S&H. (Overseas $12.00.) Send cash, check or money order to Bruce Harrison, 5705 40th Place, Hyattsville MD 20781. v3n5

CHRISTMAS MUSIC

Christmas music still available. The 1995 Christmas Music from Harrison is on the market again. For the set of three disks, send $10.00 to Bruce Harrison, 5705 40th Place, Hyattsville MD 20781. v3n5

Call 512-255-1512 to place a classified ad. The cost is 10 cents per word.

Devoted to the TI99/4A since 1984

Subscription Fees

❑ 6 issues, USA, $35 ❑ 6 issues, Mexico, $40.25

❑ 6 issues, Canada $42.50 ❑ 6 issues, other countries surface mail, $40.00

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

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

Address Changes

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

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

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

Card No. ________________

Expiration Date ________________
(Minimum credit card order is $9)

Signature ________________
(Required on credit card orders.)

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

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

Name ________________

Address ________________

City ________________

State ________________ **ZIP** ________________

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

Disks, Etc.

❑ **Back Issues**,$3.50 each to March 1996 (monthly issues, later $6 each (bimonthly issues). List issues: ________________

No price breaks on sets of back issues. Free shipping USA. Add $1, single issues to Canada/Mexico. Other foreign shipping75 cents single issue surface, $2.80 airmail. Write for foreign shipping on multiple copies.

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

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

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

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**....................$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

PERIODICAL