Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 2 | March 1995 | $3.50

SCSI

(Small Computer Systems Interface)

Here at last (for the Geneve)

RXB 1000 in beta test version
More of those Extended BASIC myths
Tips to win at Parsec
Handy Dandies by Jim Peterson
MICRO-Reviews of Video Titler, Horizon RAMdisk Password,Checktime, Ultimate Delay and Random Number

CONTENTS

MICROpendium

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

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

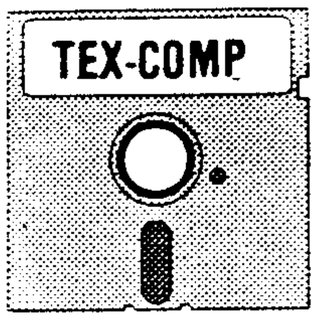
Foreign subscriptions are $40.25 (Mexico); $42.50 (Canada); $40.00, surface mail to other countries; $52 airmail to other countries.

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

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

*READ THIS

Here are some tips to help you when entering programs from MICROpendium:
1. Most BASIC and Extended BASIC programs are run through Checksum, which places the numbers that follow exclamation points at the end of each program line. Do not enter these numbers or exclamation points. Checksum is available on disk from MICROpendium for $4.
2. Long Extended BASIC lines are entered by inputting until the screen stops accepting characters, pressing Enter, pressing FCTN REDO, cursoring to the end of the line and continuing input.



ANNOUNCING NEW LOW PRICES ON FREEWARE DISKS FROM $2.95 TEX COMP $2.95

TEX-COMP HAS SLASHED PRICES ON THE BIGGEST & BEST COLLECTION OF FREEWARE FOR THE 99/4A. NOW ONLY $2.95 PER DISK WITH A 5 DISK MINIMUM. CHOOSE FROM HUNDREDS OF GREAT PROGRAMS. NO EXTRA CHARGE ON PROGRAMS THAT REQUIRE MORE THAN ONE DISK SIDE. ALL PROGRAMS HAVE BEEN TESTED AND ALMOST ALL HAVE BEEN PROVIDED WITH EXBASIC AUTOLOAD. CHOOSE FROM THE BEST IN GAMES, UTILITIES, GRAPHICS, HOME & BUSINESS AND DISK BACKUPS OF DOZENS OF YOUR FAVORITE MODULES THAT ARE NOW OUT OF PRODUCTION.

GAMES • BUSINESS • GRAPHICS • WORD PROCESSING • UTILITIES • DATABASE • MUSIC • COMMUNICATIONS • HOME

GRAPHICS, MUSIC & ANIMATION
#1 THE SINGING TI VOL. 1 (S)(2)
#4 PRINTART (2)(P)
#5A MUSIC/GRAPHICS
#6 EXBASIC MUSIC (2)
#7 SPACE SHUTTLE MUSIC/GRAPHICS
#9 MONA LISA PRINTOUT (P)
#11 ANIMATED XMAS (WOODSTOCK)
#14 FIGURE STUDIES (P)(PG)
#32 EXBASIC XMAS MUSIC (2)
#41 VIDEO GRAPHS (M)
#52 ANIMATION 99' (2)
#69 PLAYER PIANO/KEYBOARD ANALYSIS
#93 KBGB GIRLIE CALENDAR (4)(P)
#103 SORGAN THE TI ORGAN
#107 STARTREK MUSIC ALBUM
#111 POP MUSIC & GRAPHICS
#114 PANORAMA
#115 GRAPHICS DESIGN SYSTEM
#120 BITMAC (2)(P)
#230 THE SINGING TI VOL. 2 (S)(2)
#231 THE SINGING TI VOL. 3 (S)(2)
#246 THE SINGING TI VOL. 4 (S)(2)
#313 3-D WORLD (CAD for the TI)

COMPUTER UTILITIES, PRINTER UTILITIES & PROGRAMMING LANGUAGES
#3 DUMPIT (E/A)
#15 STAR/EPSON PRINTER DEMO (P)(2)
#16 SIDEWAYS PRINTOUT (P)(2)
#18 TI DIAGNOSTIC (MM)(2)
#28 LOADERS & CATALOGERS
#30 HOUSEHOLD BUDGET PRINTOUT (P)
#35 PROGRAMMING AIDS & UTILITIES I
#42 FUNNELWEB FARM (SHELL UTILITY)
#53 HACKER CRACKER
#55 SCREEN DUMP (P)
#62 DISK MANAGER II (M)
#75 DISK CATALOGER
#76 PROGRAMMING AIDS & UTILITIES II
#78 ARICON GRAPHIC CONVERSION
#79 DISK MANAGER 1000
#80 BIRDWELL DISK UTILITY (2)
#85 AUTOBOOT UTILITY
#86 COLUMN TEXT III (P)
#87 ARCHIVER III
#89 PROCALC DECIMAL/HEX CONVERTER
#96 STATISC & SORT ROUTINES
#97 MEMORY MANIPULATOR
#101 ENHANCED DISPLAY PACKAGE
#108 FUNLPLUS (PLUS!)
#110 DISK+ AID
#117 UNIVERSAL DISASSEMBLER
#119 RAG LINKER CONVERSION
#127 PIX GRAPHICS UTILITY
#243 OS/99 (GD)
#251 PC TRANSFER TI/IBM (DD)
#253 THE EXPLORER/DM1000
#254 NIBBLER/TURBO
#260 TI FORTH (DISK ONLY- add $8 for manual)
#17 TI FORTH DEMO
#116 TI FORTH TUTORIAL
#5079 FORTH SOURCE CODE
#104 C99 COMPLIER & LIBRARY
#5007 TEACH YOURSELF TI BASIC
#5019 TEACH TOURSELF EX-BASIC
#5067 BEGINNING BASIC TUTOR
#307 GRAPHICS CODE GENERATOR
#3912 SUPER BUGGER
#3913 BETTER BANNERS
#3914 CERTIFICATE 99
#3915 HOROSCOPE MAKER
#3916 GRAPHX+ PRINT SHOPPE
#3917-3720 GRAPHX COMPANIONS 1-4
#3721 MAC FLICK (G)
#3722 PRINTING TO GO (G)
#3723 GRAPHX DINOSAURS (G)

BUSINESS, ACCOUNTING, WORD PROCESSING, DATA BASE HOME OFFICE & HOME
#10 GOTHIC PRINTOUT (P)
#19 TI WRITER/MULTIPLAN UPGRADE
#20 ACCOUNTS RECEIVABLE (2)
#21 DATA BASE DEMO
#23 WILL WRITER
#29 LABEL MAKER I (P)
#36 STRICTLY BUSINESS (2)
#56 SPREAD SHEET
#58 PR BASE (database)
#59 GRAPH MAKER
#74 LABEL MAKER II (P)
#77 MICRodex 99 (database)
#81 HOME ACCOUNTING SYSTEM
#83 HOME APPLICATION PROGRAMS (2)
#90 JET CHECKBOOK MANAGER
#92 HOUSEHOLD INVENTORY
#109 TI WRITER MINI MANUAL
#112 INVOICE PACK
#113 LABEL MAKER III
#129 CASH DRAWER (point of sale)
#130 THE ORGANIZER
#147 CALENDAR-NOTEPAD
#177 HOUSEHOLD BUDGET MANAGEMENT (M)
#30 HBM DATA PRINTOUT (P)
#221 PERSONAL REAL ESTATE (M)
#249 MAPMAKER
#252 99 WRITER II (TI WRITER)(P)
#253 AMA MAILING LIST
#255 TRAK-A-CHECK
#257 DAILY DIARY
#257A EXER-LOG
#23 WILL WRITER
#310 SELF HELP TAX CUT

TELECOMMUNICATIONS (MODEM)
#57 TELCO
#57D TELCO (FOR SYSTEMS WITH DS DRIVES)
#118 FAST TERM

EDUCATION & PERSONAL DEVELOPMENT
#22 ASTROLOGY
#24 ENGINEERING CALCULATIONS (2)
#25 MEDICAL ALERT
#27 KIDS LEARNING I (2)
#31 MORSE CODE TRAINER
#37 LAPD COOKBOOK (2)
#40 ARTIFICIAL INTELLIGENCE (ELIZA)
#54 ASTRONOMY
#66 HEBREW TYPEWRITER
#67 GENEALOGY (2)
#71 KIDS LEARNING II (2)
#95 WEATHER FORECASTER +
#138 FIREHOUSE COOKBOOK
#142 TOUCH TYPING TUTOR (M)
#184 FACE MAKER
#194 ST. NICK/GHOSTLY SPELLING (M)
#195 TINY LOGO
#199 MILLIKEN ADDITION (M)
#200 MILLIKEN DECIMALS (M)
#203 MILLIKEN FRACTIONS (M)
#204 MILLIKEN INTEGERS (M)
#205 MILLIKEN LAWS OF MATH (M)
#211 MIND CHALLENGERS (M)
#212 MINUS MISSION (M)
#213 MILLIKEN PERCENTS (M)
#214 STORY MACHINE (M)
#215 BEGINNING GRAMMAR (M)
#216 METEOR MULTIPLICATION (M)
#217 HANGMAN (M)
#222 MUSICMAKER (M)
#223 PHYSICAL FITNESS (M)
#225 ALIEN ADDITION (M)
#226 ALLIGATOR MIX (M)
#227 DEMOLITION DIVISION (M)
#228 DRAGON MIX (M)
#270 TRIGONOMETRY
#271 CALCULATORS & CONVERSIONS (2)
#272 HIGHER MATH (2)
#273 ASTRONOMY 2
#306 SPEAK & SPELL II (S)(EX)

GAMES
#2 NM. OF FORTUNE, BLKJACK, POKER
#3 TI TREK
#8 LOTTO PICKER
#13 STRIP POKER (PG)
#26 R-RATED NOVELTY GAME
#33 CHECKERS & BACKGAMMON
#34 SOLITAIRE & SCRABBLE
#38 GREAT TI GAMES VOL 1
#39 GREAT TI GAMES VOL 2
#43 BEST OF BRITAIN GAMES VOL 1
#45 BEST OF BRITIAN GAMES VOL 2 (LEGEND OF CARFAX ABBY GRAPHIC-INTERACTIVE ADVENTURE)
#46 SUPER TRIVIA 99
#47 INFOCOM RAPID LOADER
#48 GHOSTMAN (from U.K.)
#49 DEMON DESTROYER (from FRANCE)
#50 OH MUMMY! (HIT from GERMANY)
#51 BERLIN WALL (from CANADA)
#60 FREDDY (HIT from GERMANY)
#61 THE MINE (from GERMANY)
#63 ASTROBLITZ & MAZOG
#64 MAJOR TOM & SPACE STATION BETA
#65 PERFECT PUSH (HIT)
#68 CHESS (SARGON)
#70 TI RUNNER II (HIT)
#72 CEBERUS (HIT SPACE GAME)
#73 CRYTO (GRAM)
#82 CROSSWORD (PUZZLES)
#84 GALACTIC BATTLE & SPY ADVEN.
#88 AUSSIE GAME COLLECTION VOL 1
#91 THE MAZE OF GROG (WOODSTOCK II)
#94 GREAT TI GAMES VOL 3
#98 DAYS/DOORS OF EDEN (BIBLE ADV)(AM)
#99 GREAT TI GAMES VOL 4
#100 ASSAULT THE CITY (TD)
#102 COLOSSAL CAVES (ADVENTURE)
#105 KINGS CASTLE (M)
#106 QUEST (D&D)
#121 SUPER YAHTZEE & WHEEL II
#122 ADULT ADVENT & GAMES (PG)
#123 GREAT TI GAMES VOL 5 (2)
#124 GREAT TI GAMES VOL 6 (2)
#125 BLACKJACK & POKER (M)
#126 VIDEO CHESS (M)
#128 TETRIS (HIT from RUSSIA)
#131 COMPUTER CRAPS
#132 AMBULANCE (M)
#133 DRIVING DEMON (M)
#134 ROTO-RAIDER (M)
#135 ARTURUS (HIT-ZARGON)
#136 ANT-EATER (M)
#137 CROSS-FIRE (M)
#139 MOONMINE (M)
#140 MASH (M)
#141 MOONSWEEPER (M)
#143 CONGO BONGO (M)
#144 STAR TREK (M)
#145 BUCK ROGERS (M)
#148 KENO & SLOTS
#149 GREAT TI GAMES VOL 7 (with HIT BLOCKBUSTER)
#150 ULTIMATE TRIVIA
#151 JUNGLE HUNT (M)
#152 POLE POSITION (M)
#153 DONKEY KONG (M)
#154 PROTECTOR II (M)
#155 PAC MAN (M)
#156 CENTIPEDE (M)
#157 DEFENDER (M)
#158 SHAMUS (M)
#159 MS. PAC MAN (M)
#160 DIG-DUG (M)
#161 PICNIC PARANOIA (M)
#162 MOON PATROL (M)
#3711 ARCADE SPECIAL (4 GAMES)
#3712 THREE GREAT GAMES
#3713 PRO TENNIS+

INFOCOM ADVENTURE BACKUPS
#163 ZORK I
#164 ZORK II
#165 ZORK III
#166 HITCHIKER'S GUIDE
#167 WITNESS
#168 ENCHANTER
#169 INFIDEL
#170 PLANETFALL
#171 SORCERER
#172 DEADLINE
#173 CUTTHROATS
#174 SUSPENDED
#175 STARCROSS
****
#176 AMAZING (M)
#178 DEMON DESTROYER (M)
#179 POPEYE (M)
#180 QUEBERT (M)
#181 METEOR BELT (M)
#182 BLASTO (M)
#183 CAR WARS (M)
#184 FACE MAKER (M)
#185 SUPER FLY (M)
#186 SPACE BANDITS (M)
#188 KILLER CATERPILLER (M)
#190 BLACK HOLE & SPACE AGRESS. (M)
#191 GREAT TI GAMES VOL 8
#192 GREAT TI GAMES VOL 9
#193 SPY'S DEMISE (HIT) (M)
#194 ST. NICK+ (M)
#196 JOTTO
#197 PRO TENNIS+ (HIT)(M)
#198 TI INVADERS/TOMBSTONE CITY (M)
#202 CONNECT FOUR (M)
#205 HOPPER (M)
#206 TREASURE ISLAND (M)
#206B SLYMOIDS (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 4A FLYER (FLIGHT SIM.)(M)
#232 TUNNELS OF DOOM (MOD BACKUP PLUS 2 NEW ADVENTURES)(M)
#233 MS ADVENTURES (3 ADV-EXBASIC)
#248 STRIKE THREE BASEBALL (M)
#248 GREAT TI GAMES VOL 10
#250 BARRAGE/SPOTSHOT
#318 THREE GREAT GAMES
#319 ARCADE SPECIAL
#317 BEANSTALK ADVENTURE
****
SCOTT ADAMS ADVENTURES (USE WITH ADVENTURE MODULE)
#349 TI ADVENTURES 1-13+
#350 TI ADVENTURES 14-16+

P = Printer required
G = Graphx required
S = Speech required
M = Module Backup
MM = Mini Memory req.
E/A = Editor Assem.
Exbasic and 32k mem req. for most programs.

ORDER BY PHONE

(818) 366-6631 24 HOURS A DAY

VISA

Authorized Dealer

TEX COMP

America's Number One TI computer retailer

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

TERMS: MINIMUM ORDER FOR $2.95 PRICE IS 5 DISKS (REG. 4.95) ADD $4.00 PER ORDER FOR SHIPPING (U.S.)
ALL PRICES ARE FOR CASH/CHECK, ADD 3% FOR CREDIT CARD ORDERS. INCLUDE STREET ADDRESS FOR U.P.S.

COMMENTS

SCSI arrives, TI version due soon

It's been a long time coming, but I finally got my Western Horizon SCSI controller for the Geneve. I haven't got it running yet, I'm waiting for MDOS 2.5 and a 32K chip for my Geneve, but I've heard from several readers who have them and they are happy with them.

Mike Doane, writing in the newsletter of the Soutwest Ninety-Niners, bought his SCSI card at Fest West and reports satisfaction with the device. He also offers a number of helpful tips to new owners that we're reprinting elsewhere in this edition. He concludes, "My hat is off to Don O'Neil and Bud Mills for providing us with an excellent hard drive system."

According to Bud Mills, the SCSI runs nearly as fast as a RAMdisk, much faster than a Myarc HFDC.

According to Bud Mills, the SCSI runs nearly as fast as a RAMdisk, much faster than a Myarc HFDC. While I can't say anything about it's operation, I'll mention that the documentation that comes with it seems thorough. It covers DIP switches, disk drive recommendations, power supply requirements and tips, a parts list if you ever need repairs, installation and cabling instructions. An instruction sheet for installation of the 32K of fast RAM is also included. This involves opening up the Geneve card, removing a socketed memory chip, attaching the second chip on top of it and replacing it on the board. Soldering is also required. Those who have little or no experience soldering electronic equipment may be better paying someone else to do the work. The instructions are easy to follow for those with a steady hand. Why does the Geneve need the extra RAM to accommodate the SCSI? According to Bud, it's needed so that the Geneve can keep up with the speed of the SCSI.

Good news for TI users: Bud says that the long-awaited TI version of the SCSI card will be shown at the Multi User Group Conference in Lima, Ohio, in April.

BAD NEWS FROM OUR PRINTER

Our printer gave us some bad news early this month. Because of the skyrocketing cost of paper, the cost of printing MICROpendium is going up. The increase takes effect next month.

The amount isn't so bad that I'm pulling my hair out, but it's also only a sample of what's to come. The printer's paper supplier has already warned of additional price hikes later this year, perhaps twice more, with no end in sight. We expect to be hit with additional increases each time the price of paper rises. On top of this, as you all know, postage rates went up in January. Again, the postage rate hike isn't enough to cause me to lose sleep, but add it to the increasing cost of printing MICROpendium and the aggregate total takes on greater significance.

What does this mean for MICROpendium? I'm not sure. We haven't passed on any increases in printing costs or postage costs for several years. But I can see that the time will come when we'll have to do something.

A couple of options pop up fairly quickly. For one, we could consider reducing the number of issues we publish to compensate for the additional cost. It could be that we would produce 11 issues of each volume, instead of 12.

Also a possibility is reducing the number of pages from 32 to 24 several times a year while continuing to produce it every month. (The press we use works on increments of eight pages in our format.)

Of course, we could look at going to another press. If I thought this was a reasonable option I would do so, but the cost increases are industry-wide. Also, we've used this printer ever since our first edition and they give us good service.

I won't know for a while how bad these increases will be but I'll keep you posted.

HERE WE ARE, STARTING OUR 12TH YEAR

While the above news about paper costs doesn't make me feel like celebrating, February marked the start of our 12th year. If you'd have asked me 12 years ago how long MICROpendium would last, I'd have told you 5 years at most. Thanks for sticking with us.

—JK

FEEDBACK

Test site offered

My name is Randy McDaniel and I am writing to offer my services, and that of my entire school system, for your possible use.

My job is the director of Technology and I am challenged with school reform in a small rural district in Southeast Kansas using technology as a key part of that reform. I am not merely asking for handouts and freebies. I am willing and actually prefer to become a test site for "beta" software and/or hardware that you and your company may be in the early stages of developing. Additionally, I am working on my doctoral thesis with the emphasis being how technology can give economically disadvantaged youths a fighting chance in the information age that we are approaching.

To give you a brief snapshot of our school: 1) We are the poorest school district in the state of Kansas measured by assessed property evaluation, 2) We have the highest percentage of free and reduced lunch applications in the state of Kansas (this indicator is used to assess socioeconomic status), and 3) We are currently attempting to infuse technology into our curriculum K-12 with a five-year implementation that is currently in year two phase. We have a computer in every classroom and they are connected with a simple LAN.

As I stated earlier, my function is to attempt to train teachers and administrators on how technology can "level the playing field" in our school with limited resources. What we are not limited in is will power and effort. Although we are the poorest school, we do not use this as an excuse. On the contrary, this has actually become our rallying cry. We will simply work harder to accomplish our goals of providing a quality education for our students and attempt at the same time to show them all of the possibilities in the world.

Enough about philosophy. Again, my point in writing you is to ask if there is any way our school system could become a test site for your upcoming ventures. Be it hardware, software or courseware. We will do all of the work. All we really need is a chance to obtain the resources and I personally will be responsible for overseeing the implementation and evaluation. We currently have a decent installed base of Macintosh computers and a small supply of DOS compatibles. However, we are not locked into any platform. We want to do what is best for our kids. If I can serve your company in the above-mentioned capacities in any way please let me know. Our phone number is (316) 783-1905 and our fax number is (316) 783-1780.

Randy G. McDaniel
Director of Technology
Galena Unified School District No. 499
702 East 7th St.
Galena, KS 66739

MICROpendium received this letter and we have written Mr. McDaniel advising him that we are a magazine for the TI99/4A. We are printing his letter because among our resourceful readers may be some who can help him.

Not so

The following was emailed to MICROpendium.—Ed.

```
.LM 1;RM 132;FI;AD
  Then message on line 2
  Then save as PF, C DSK2.???
  Then run thru the formatter.
```

Only problem with the above is that it DOES NOT work. I have an Epson RX80/FT printer with 1 meg memory and Dots-Perfect chip installed. I just punch the right code and the printer will print in just about any mode that I wish. I punch in code 1 which is condensed print and then run the above with a letter starting on line 2. It prints in condensed print but in 80-column. I know that it is possible but not as stated in the MICROpendium article.

Richard C. Arthur Jr.
R.C.ARTHUR@GENIE.GEIS.COM
or FLBOY@FREENET.TLH.FL.US

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

READER TO READER

Gary Fitzgerald, 9 Merwin Ave., Milford, CT 06460, (203) 783-1776, writes:

I procured a Myarc Floppy Disk Controller and have loved having DSDD capacity. My problem is formatting disks with Funnelweb. Even though Tony McGovern uses, or used to use the same controller, all editions of Funnelweb I download or receive are set up for Cor-Comp or TI controllers. I have read the docs over and over and I understand there is a fix using a sector editor. The powers of editing through Funnelweb elude me. I'd like to talk to or write to someone who has overcome this obstacle and actually owns a Myarc FDC.

Corrections

Correction to headline

Bruce Harrison's February column, The Art of Assembly—Part 43, was not part 43 at all. Part 43 was published in the January MICROpendium. February's installment was part 44. The sidebar accompanying the column was correctly labeled as Sidebar 44.

User Note author identified

Charles Good points out that he was the original author of the User Note titled "Using TI-Writer to transfer text over the serial port" (February 1995).

RXB 1000 features unique batch file processing system

By JOHN KOLOEN

Rich Gilbertson hasn't completed work on his RXB program, probably because he'll always find something he wants to improve. Version 1000 of the program was recently completed. It's in beta, but I suspect that it won't have many bugs.

This new version introduces a new feature called "USER" that is the equivalent of a batch language for Extended BASIC. Remember, RXB got it's start as Extended BASIC. In it's current incarnation you might consider it as Extended BASIC on steroids.

There have been a lot of Extended BASIC equivalents over the years, most of them focusing on beefing up XB's graphic capabilities. RXB goes well beyond this, incorporating many new routines aimed specifically at programming. The User function goes farther by providing a batch file processing system.

According to Gilbertson, "This subprogram can be accessed from program or edit mode. What it does is bypass the normal keyscan routine in edit mode or input commands or ACCEPT commands and inserts (a specified) D/V80 file key by key. So, from a D/V80 file you can have RXB convert D/V80 files (into) an Extended BASIC program. You can also have it save the program, merge in another, run the program just created, add, delete or move, copy or edit lines. And you can now break a program, add or replace lines, then continue the program without losing variables. That means unlimited size of programs. Variables, of course, are limited by VDP size, about 12.5 kilobytes, but I'm working on that, too."

Here is an example of a batch file from the RXB manual:

```
CALL NEW
OLD DSK1.XBPROGRAM
RES 11,3
MERGE "DSK1.MERGEPGM"
SAVE "DSK1.NEWPROGRAM"
RUN
```

Other modifications include an editor that lets users enter 21 lines without pressing the insert key. The TI99/4A accepts only six lines without hitting the insert key.

RXB features rewritten ROM and GRAM from TI Extended BASIC. The ROM in RXB are fairware. The GRAM was rewritten in GPL (Graphic Programming Language).

Users of any version of RXB will probably want to upgrade to V1000, even though it is beta, just to play with the CALL USER. Of course, there are many other improvements, including program device name access. By using any true device service routine (DSR), users can access the Editor/Assembler main menu simply by typing "RUN EA" within BASIC or RXB.

RXB V1000, which is freeware, is available from various electronic services and includes extensive documentation. It requires either a GRAM device or a Geneve to run. An earlier version was described extensively in the May 1993 edition. Gilbertson can be reached at 1901 H St., Vancouver, WA 98663.

Renew your MICROpendium disk subscription now!

Reminder cards have been sent out for the new MICROpendium disk series, which starts in April and runs through next March. The cost is $40 in the U.S., $41 in Canada and $54.40 (overseas airmail) or $45 (overseas surface mail). The disks, which are mailed monthly, include all programs from each issue plus bonus shareware programs from a variety of sources.

To order, send your name, address and a check, money order or credit card information to MICROpendium Disks, P.O. Box 1343, Round Rock, TX 78680. Or to order by credit card (Mastercard/Visa), call 512-255-1512 (expect a 5% surcharge).

THE ART OF ASSEMBLY — PART 43

Would you like a little CoCo?

By BRUCE HARRISON

It is May 1994, and we are once again departing from our normal format in this column. Among other things, there's no sidebar of source code this month. This is because we've just started on another new adventure with computers, and wanted to pass along some observations that may upset or even anger the loyal TI users, but we're telling you the truth as we see it, with no apologies.

THE PURPLE HEART BARGAIN

There are a number of "thrift" stores in our area, and being on a small budget, we shop at those stores. All the goods sold are used, sold "as is," all sales final, and so on. Still, there are occasionally real gems among the rhinestones. One day while browsing in the "Purple Heart" store in nearby Bladensburg, we found a white box that contained a Radio Shack Color Computer 2. We inquired for a price, as there was no tag attached. "Twenty dollars, AS IS" was the response. This was a 64K memory model, with Radio Shack's Extended BASIC built in, so it looked like a potential bargain. We've found TI consoles (bare bones) at this same store selling for as little as $5, and they all worked. We scooped up the CC2 (or CoCo) and raced home to try it out. The CoCo is a nice package. It includes a built-in power supply, so there's no awkward transformer on its power cord. There were no cables included except RF cable to connect to a TV set. The modulator, too, is built right in, so no modulator needs to dangle at the end of a cord. The RF connected through a simple adapter box that we use with Atari video games.

It worked perfectly, just as we'd hoped. The Extended BASIC here is, of course, different in many respects from that on the TI, but we were able very quickly to get a stupid test program typed in. We used a simple FOR-NEXT loop as a quick test, like this:

```
10 FOR I=1 TO 200
20 PRINT "HARRISON'S GARBAGE REPEAT NUMBER";
I
30 NEXT I
```

Having put this simple test program in place, we typed RUN <enter>, and were amazed! This program ran so fast that the lines scrolling up the screen were a blur. It finished the 200 repeats in just a few seconds, then we could read the lines that remained on the screen. We'd never seen any version of BASIC run so fast! To see just how fast, we fired up the PC that happened to be nearby, (Tandy 1000SX) and typed the same program into its GW BASIC. Now we teamed up to start each of these two running at the same time. The CoCo finished at about the time that the PC was printing number 100. In other words, this CoCo's Extended BASIC was executing about twice as fast as the GW BASIC on the PC. We knew from earlier experiments that the PC with GW BASIC runs much faster than our trusty old TI, so we put that same program into the TI in Console BASIC. The difference is startling. TI's Console BASIC is painfully slow compared to GW BASIC, and ridiculously slow compared to the CoCo. That same program ran a bit faster in TI Extended BASIC, but was still nowhere near a match for the GW BASIC on the PC, and lagged way behind compared to the CoCo.

GRAPHICS CAPABILITIES

We called our local Radio Shack, and discovered that the manual for the CoCo is still available (about $5), so we ordered that. While waiting, we checked the shelves at a local used book store, and found a book called Color Computer Graphics. This little gem contained a very good description of the graphics capabilities provided by the CoCo, with sample programs we could type in and run. Unlike the TI, the CoCo offers excellent access to its graphics capabilities directly from its Extended BASIC. Yes, there's even a Bit-Map mode (256x192 pixels) that's accessible from the CoCo's XB. It will also permit drawing lines, boxes, filled in boxes, circles and ellipses just with simple statements from XB! Even without the regular manual, we found that we could do some pretty amazing things on this little computer. There are some twelve modes of operation for its VDP, and all of those can be used without resorting to any Assembly tricks.

LIMITATIONS

There are some limitations on the CoCo that seem strange to the TI user. There are no lowercase characters available, for example, but there are two sets of uppercase characters, one set in "normal" colors, the other in "inverted" colors. Also, in many cases the ASCII codes are not what we're accustomed to! There is no underline character, for example. Its place is taken by a left-pointing arrow symbol. Why? These character patterns are all in ROM, and there's no way to re-define them, so this weird arrangement of the ASCII codes is just a fact of life on the CoCo. This can be overcome for programs that execute within the CoCo, but could be a real problem if one's bringing in stuff from the serial port, for example, since the ASCII codes coming in might not provide the expected characters on the screen. There also is no capability for using the standard characters in the graphics modes! There are in-between modes of operation called "semigraphic" modes in which one can mix graphics with alphanumerics, but the alphanumerics are not available once you've put the CoCo into a "true graphics" mode.

One other "not-so-good" is the method of editing program lines in the CoCo's Extended BASIC. It's based on a method borrowed from the old Tandy TRS-80 BASIC, and is very cumbersome compared to the editing capability on the TI's BASICs. Neither TI nor CoCo can hold a candle to the editing capability of the GW BASIC on the PCs, but the CoCo is far below the TI in editing BASIC program lines. I'd tell you just how difficult it is, but you wouldn't believe it could be so bad, and it would take pages to describe.

TIME'S A WASTIN'

We said the CoCo is fast, and the TI, by comparison, is slow. We got out a digital watch and timed the things. First TI Console Basic, which was the slowest, then on up. Each time is expressed in minutes and seconds:

(See Page 8)

THE ART OF ASSEMBLY—

(Continued from Page 7)

Computer	Time
TI Console BASIC	1:22
TI Extended BASIC	0:38
Compiled TIXB	0:25
Tandy 1400 FD (GW Basic)	0:19
Tandy 1000SX (GW BASIC)	0:18
CoCo Extended BASIC	0:07

Going from TI Console BASIC to the CoCo Extended BASIC is nearly a 12:1 speed change! That Tandy 1400 FD in the list is the laptop PC that we use to write these columns. The 1000SX was once the top of the line Tandy PC, back when the Intel 8088 was the standard PC microprocessor. The 1400 FD also uses the Intel 8088. Both are beaten handily by the CoCo, which uses the Motorola 6809 microprocessor chip. The Extended BASIC used in the CoCo is by Microsoft, and is sort of a cross between the Microsoft BASIC used in PCs and the old Tandy TRS-80 BASIC.

Now that we've offended most of our readers by giving such a glowing account of the Radio Shack Color Computer 2, let's back off just a little for some fairness. As yet, we don't even have a cassette cable to save our CoCo programs, much less any way to use conveniences like disk drives. Also, we don't have any way to check the computer with Assembly language stuff. We don't know what internal clock speeds are used in the CoCo, but we're sure it's not in the tens of megahertz. Probably the single biggest factor in the speed of the CoCo as compared to the TI-99/4A is the lack of that GPL interpretation step. Thus there's no need for GROMs, and the BASIC interpreter is contained in ROM that's within the main address space. There's also no such thing as VDP RAM in the CoCo. The screen table is simply a piece of the regular RAM reserved for the screen images. This concept makes things happen faster, but at the expense of using memory for screen images that could otherwise be used for program space.

While we're on the subject of limitations, we should mention that there are only nine colors (including black) on the CoCo, and that in the highest resolution graphics mode (256x192 pixels) there are two sets of two colors each, so the graphics are drawn only in black on green (ugly) or black on gray (better). In lower resolution graphics modes, there are two sets of four colors to choose from, but not all the choices are useful. We're spoiled by the ability on the TI to use 16 colors even in the bit-map mode at 256x192 pixels.

We'll keep experimenting with the CoCo, and will have more to say about it later. Have no fear, we'll not give up our trusty old TI, nor our PCs. We'll continue making those little Assembly programs that some in our community find useful. We'll also keep on probing into the inner workings of our TI, and telling you what we've found lurking there.

We'll be presenting some new Assembly programs at the Lima Faire. Of course by the time you read this, it'll be almost time for the 1995 Lima Faire. Sometimes it's a real pain to be writing so far ahead of our publication dates. The only real advantage is that, whenever we want, we can take a month or two (would you believe ten?) off from writing without missing a deadline.

Next month we'll go back into the realm of TI Bit-Map mode. Among other things, we'll pass along a nifty and quick way of drawing straight lines from anywhere to anywhere on the TI Bit-Map screen. We'll show an algorithm that we "borrowed" from a book on PC Assembly, modified to work on the TI, making optimal straight lines and making them very quickly. See you then!

MYARC ADVANCED BASIC

VRAM aids work with VRAM Analyzer

By JIM UZZELL

This month is a collection of programs that provide MY-BASIC information as it relates to some of the information that the VRAM Analyzer (December 1994 MICROpendium) displays.

The first program, GRFMODES30, looks at MY-BASIC default data for each graphic mode and displays that data. This data can also be compared to Appendix K of the MY-BASIC manual. You will note that some data is displayed in Base 0. Also Screen Mode 2 (Graphics 1,1) is not recognized as a default mode. All of the Graphics(1,x) modes indicate a screen size of 672 instead of 768 because of the default margin, as stated in Appendix K, and cannot be used by DISPLAY AT or PRINT. Since HCHAR and VCHAR are not limited by margins, they can be used to fill in these areas with characters. In all cases the default Display Page is 0 and there is no routine in MY-BASIC to change this.

After the program runs, the Print Screen key is active, if you want a hard copy.

The next three programs — GRAPHICS11, GRAPHICS21, GRAPHICS31 — pokes all 256 characters to the screen repeatedly until it fills the screen. The program advises you of the screen position as it fills the screen. In each of these programs, the value of X, starts with a value of 0 and increments up to the screen size. This leads to a question — how can you poke an ASCII value of 768 (graphics11) to the screen, when the ASCII values range from 0 to 255? The "how" is explained on page 27 of the MY-BASIC manual under the definition of CHR$.

The last program, GRAPHICS31, provides information at the end of program for changes that can be made to the row value to write to rows 25-27.

I am sure some of you will run these programs in the other modes (graphic or bitmap) and, while the results may be interesting, these modes are handled in a totally different manner and are beyond the scope of this article.

(See Page 9)

MY-BASIC—

(Continued from Page 8)

GRFMODES30

```
1 !DDI SOFTWARE
2 !COPYRIGHT 1995
3 !GRAPHIC MODES INFO
4 !GRFMODES30
100 FOR X=1 TO 3 :: FOR Y=1
TO 3
110 CALL GRAPHICS(X,Y)
120 CALL PEEK(-955,A) :: CAL
L PEEK(-965,B) :: CALL PEEK(
-973,E,F,G)
125 CALL PEEK(-920,J,K)
130 C(X,Y)=A :: D(X,Y)=B ::
H(X,Y)=E :: I(X,Y)=(256*F)+G
135 L(X,Y)=(256*J)+K
140 NEXT Y :: NEXT X
141 CALL GRAPHICS(4)
142 CALL PEEK(-955,A) :: CAL
L PEEK(-965,B) :: CALL PEEK(
-973,E,F,G) :: CALL PEEK(-92
0,J,K)
143 PRINT TAB(28);"SCREEN";T
AB(35);"SCREEN";TAB(43);"MAX
 PIXEL";TAB(55);"SCREEN SIZE
":TAB(29);"MODE";TAB(36);"WI
DTH";TAB(43);"ROW   COL";TAB
(55);"DISPLAY AT"
150 FOR X=1 TO 3 :: FOR Y=1
TO 3
160 PRINT TAB(5);"GRAPHICS (
";X;",";Y;")",C(X,Y);TAB(37)
;D(X,Y);TAB(42);H(X,Y);" ";I
(X,Y);"    ";L(X,Y)
170 NEXT Y :: NEXT X
180 PRINT TAB(5);"GRAPHICS (
 4 )",A;TAB(37);B;TAB(42);E;
" ";(256*F)+G;"    ";(256*J)
+K
190 PRINT :: PRINT "    PEEK
 MEMORY LOCATIONS VALID FOR
VERSION 3.0 ONLY"
```

GRAPHICS11

```
1 !GRAPHICS11
2 CALL GRAPHICS(1,1)
100 CALL CLEAR :: X=0
110 FOR R=0 TO 23
120 FOR C=0 TO 31
130 CALL POKEV((R*32+C)+1024
,X)
136 DISPLAY AT(24,25):(R*32)
+C+1
140 X=X+1
150 NEXT C
160 NEXT R
170 CALL KEY(0,K,S) :: IF S=
0 THEN 170
180 ! 24R x 32C =768  0-767
```

GRAPHICS21

```
1 !GRAPHICS21
2 CALL GRAPHICS(2,1)
100 CALL CLEAR :: X=0
110 FOR R=0 TO 23
120 FOR C=0 TO 39
130 CALL POKEV((R*40+C),X)
136 DISPLAY AT(24,37):(R*40)
+C+1
140 X=X+1
150 NEXT C
160 NEXT R
170 CALL KEY(0,K,S) :: IF S=
0 THEN 170
180 ! 24R x 40C =960  0-959
```

GRAPHICS31

```
1 !GRAPHICS31
2 CALL GRAPHICS(3,1)
100 CALL CLEAR :: X=0
110 FOR R=0 TO 23
120 FOR C=0 TO 79
130 CALL POKEV((R*80+C),X)
136 DISPLAY AT(24,76):(R*80)
+C+1
140 X=X+1
150 NEXT C
160 NEXT R
170 CALL KEY(0,K,S) :: IF S=
0 THEN 170
180 ! 24R x 80C =1920 0-1919
   R=23
190 ! 26R x 80C =2080 0-2079
   R=25
200 ! 26.5R x 80C =2160 0-21
59 R=26
210 ! 26.5=1st 4 PIXEL ROWS
OF CHAR
```

Handy Dandies #2

17 routines linked by a menu

The following program, called Handy Dandies #2, written by Jim Peterson, actually consists of 17 short routines linked by a menu. Refer to the REM statements for information about each of the routines. In the Tigercub catalog it is listed as TC-107.

```
100 CALL CLEAR !209
101 REM Programmed by James
W. Peterson 6/83 !248
102 REM COPYRIGHT 1983 Tiger
Cub Software  156 Collingwoo
d Ave., Columbus Ohio 43213
!145
103 REM REPRODUCTION FOR RES
ALE PROHIBITED. DELETION OF
COPYRIGHT NOTICE PROHIBITED.
     !047
105 DIM A$(16),D$(52),D@(100
)!070
106 RANDOMIZE !149
107 CALL CHAR(94,"3C4299A1A1
99423C")!053
110 PRINT TAB(6);"HANDY DAND
IES #2": :"TC-107";TAB(10);"
: : : : : :!041
112 FOR D=1 TO 500 !153
114 NEXT D !218
116 CALL CLEAR !209
120 PRINT "  this program co
nsists of a":"number of shor
t routines,":"all of which w
ill run" !002
130 PRINT "separately. When
you select":"a number on the
 menu, the":"computer will t
ell you to" !065
140 PRINT "list the line num
bers for":"that item, and wi
ll then":"break. Type LIST (
the line" !086
150 PRINT "numbers) and you
can read":"the REM statement
s describ-":"ing the program
, and the" !023
```

(See Page 10)

HANDY DANDIES—

(Continued from Page 9)

```
160 PRINT "program lines whi
ch accomp-":"lish it. Then t
ype CON and":"the program wi
ll demonstrate" !136
170 PRINT "itself.":"Press F
CTN and 4 to stop it,":"then
 type RUN 210 to go back":"t
o the menu." !050
180 PRINT "  Touch any key"
!134
190 CALL KEY(0,K,ST)!015
200 IF ST=0 THEN 190 !025
205 CALL CLEAR !209
210 PRINT TAB(12);"MENU": :"
1. Take turns":"2. Two-playe
r routine":"3. Two-joystick
routine" !192
220 PRINT "4. Keyboard/joyst
ick control":"5. Read either
 joystick":"6. Read either u
pper/lower":"   case input"
!101
230 PRINT "7. Arrow key/spli
t keyboard":"8. Joystick rou
tine":"9. Joystick routine":
"10. Joystick wrap=around" !
141
240 PRINT "11. Random symmet
rical char.":"12. ASCII Code
s":"13. Long Shell Sort":"14
. Shuttle Sort" !154
250 PRINT "15. Card deck":"1
6. Quick shuffle":"17. Rando
m numbers without":"   dupl
ication" !069
260 PRINT "  Type 0 for page
 2, or type":"number of choi
ce and Enter." !117
270 INPUT TQ !055
290 IF TQ=0 THEN 300 ELSE 35
0 !047
300 CALL CLEAR !209
310 PRINT "18. Number shuffl
e":"19. String input respons
e":"20. Number input respons
e":"21. 3 Octaves of music"
!115
320 PRINT "22. 2 Octaves of
bass notes":"23. Flash print
":"24. Numbers in parenthese
s" !182
330 PRINT : :"Type 0 to go b
ack to page 1,":"or type num
ber of choice." !113
335 INPUT TQ !055
340 IF TQ=0 THEN 210 !043
350 IF (TQ<1)+(TQ>24)THEN 21
0 !093
405 CALL CLEAR !209
410 ON TQ GOTO 500,630,720,7
40,840,920,1000,1310,1380,14
90,1590,1780,1840,2070,2240,
2450,2570,2650,2770,2830,290
0,3010,3140,3210 !121
500 PRINT "LIST 500-610" !03
6
501 BREAK !142
502 REM  - By going back to
 this routine after each pla
yer's move, you can alternat
e the turns of any number of
 players. !234
510 INPUT "HOW MANY PLAYERS?
 ":P !082
520 T=T+1 !033
530 IF T<P+1 THEN 550 !067
540 T=1 !012
550 PRINT "PLAYER #";T;"'S T
URN": :!082
560 REM - the T can also kee
p track of scoring. !047
570 INPUT "TYPE NUMBER ":N !
179
580 IF N/2<>INT(N/2)THEN 520
 !220
590 S(T)=S(T)+1 !161
600 PRINT "PLAYER #";T;"'S S
CORE";S(T): :!078
610 GOTO 520 !089
630 PRINT "LIST 630-710" !04
1
632 BREAK !142
634 REM - By going back thro
ugh this routine, you can al
ternate control of split key
board between 2 players. !06
2
640 T=T+1 !033
650 IF T<3 THEN 670 !172
660 T=1 !012
670 CALL KEY(T,K,ST)!106
680 IF ST=0 THEN 670 !251
690 REM - FOR DEMONSTRATION
- press keys alternately on
left and right sides of keyb
oard !071
700 CALL SOUND(500,K*5+110,5
)!078
710 GOTO 640 !209
720 PRINT "LIST 720-735" !04
8
722 BREAK !142
724 REM - This program will
alternate joystick control b
etween 2 players. !234
726 T=T+1 !033
728 IF T<3 THEN 732 !234
730 T=1 !012
732 CALL JOYST(T,X,Y)!219
733 IF (X=0)*(Y=0)THEN 732 !
151
734 CALL SOUND(500,INT(1000*
RND+110),5)!165
735 GOTO 726 !039
740 PRINT "LIST 740-830" !04
6
741 BREAK !142
742 REM - To automatically a
ccept either keyboard or joy
stick control. !005
750 CALL KEY(0,K,ST)!015
755 REM - Type any key !154
760 IF ST=0 THEN 810 !136
770 REM - tone for demo !00
780 CALL SOUND(500,K*5,5)!
8
790 GOTO 750 !063
800 REM - Unlock Alpha Lock,
 use joystick #1 !223
810 CALL JOYST(1,X,Y)!129
820 IF (X=0)*(Y=0)THEN 750 !
169
822 K=INT(100*RND+110)!089
824 GOTO 780 !094
825 REM - Combine with items
 #5 and #6 to accept either
joystick and either upper or
 lower case keyboard input.
!063
840 PRINT "LIST 840-900" !04
5
842 BREAK !142
844 REM - To accept either j
oystick #1 or #2 input. !019
850 CALL JOYST(1,X,Y)!129
860 IF (X<>0)+(Y<>0)THEN 880
 !172
870 CALL JOYST(2,X,Y)!130
880 PRINT X,Y !000
900 GOTO 850 !164
920 PRINT "LIST 920-990" !05
3
922 BREAK !142
```

(See Page 11)

HANDY DANDIES—

(Continued fromPage 10)

```
924 REM - To obtain CALL KEY
 input of ASCII value of upp
er case letters even if Alph
a Lock is up. !146
926 REM  - Put the DEF state
ment near beginning of progr
am. Use variable name K in C
ALL KEY but refer to it as K
X thereafter. !128
930 DEF KX=K-(-32*(K>90))!19
5
940 CALL KEY(0,K,ST)!015
950 IF ST=0 THEN 940 !010
960 REM - try typing with Al
pha Lock up and down !252
970 PRINT KX;CHR$(K)!129
980 GOTO 940 !254
990 REM - To obtain ASCII va
lue of lower case letters wh
en Alpha Lock is down, use
DEF KX=K+(-32*(K<97)) !050
1000 PRINT "LIST 1000-1300"
!127
1002 BREAK !142
1004 REM - to control cursor
 in 8 directions with arrow
keys and diagonals on split
keyboard - stops at borders.
 !040
1006 CALL CLEAR !209
1010 R=1 !010
1020 SR=R !181
1030 C=3 !253
1040 SC=C !151
1045 CALL HCHAR(R,C,44)!106
1050 T=T+1 !033
1052 IF T<3 THEN 1055 !047
1054 T=1 !012
1055 CALL KEY(T,K,ST)!106
1060 REM - See item #2 for c
oding of T !236
1070 IF ST=0 THEN 1055 !126
1080 IF K=18 THEN 1295 !077
1090 IF (K=1)+((K>6)*(K<14))
THEN 1050 !153
1100 K=K-(-7*(K>13))!174
1110 K=K+(-1*(K<1))!114
1120 ON K GOSUB 1250,1280,12
20,1180,1190,1210,1240,1270
!043
1130 CALL HCHAR(SR,SC,32)!01
3
1135 CALL HCHAR(R,C,44)!106
1140 SR=R !181
1160 SC=C !151
1170 GOTO 1050 !109
1180 C=C-ABS(C>2)!060
1190 R=R-ABS(R>1)!104
1200 RETURN !136
1210 R=R-ABS(R>1)!104
1220 C=C+ABS(C<31)!109
1230 RETURN !136
1240 C=C+ABS(C<31)!109
1250 R=R+ABS(R<24)!156
1260 RETURN !136
1270 R=R+ABS(R<24)!156
1280 C=C-ABS(C>2)!060
1290 RETURN !136
1295 REM - from 1080 - for "
fire!" !163
1300 REM - old row/column is
 saved in 1020-1040 and 1140
-1150 as SR and SC so that c
ursor can be erased in 1130
!191
1305 REM - just before print
ing new location in 1135 - f
or smoother flow. !026
1310 PRINT "LIST 1310-1370"
!138
1312 BREAK !142
1314 REM - To control cursor
 in 8 directions with joysti
cks - Uses subroutines 1180-
1290 in Item #7 -  !171
1315 R=12 !061
1316 C=16 !050
1318 CALL CLEAR !209
1320 CALL JOYST(1,X,Y)!129
1330 Z=((X+3*Y)/4)+5 !163
1340 REM - This algorithm by
 David N. Lewis gives values
 for joystick positions: !14
4
1341 REM   7 8 9
           4 5 6
           1 2 3 !119
1350 ON Z GOSUB 1270,1250,12
40,1280,1372,1220,1180,1190,
1210 !023
1360 CALL HCHAR(R,C,44)!106
1370 GOTO 1320 !124
1372 REM - Joystick in cente
r position !164
1376 RETURN !136
1380 PRINT "LIST 1380-1480"
!147
1382 BREAK !142
1384 REM - joystick moves cu
rsor in 8 directions, stops
at borders !205
1390 R=16 !065
1400 C=12 !046
1410 CALL CLEAR !209
1420 CALL JOYST(1,DX,DY)!009
1430 C=C+DX/4 !098
1440 C=C+ABS(C=2)-ABS(C=32)!
099
1450 R=R-DY/4 !130
1460 R=R+ABS(R=0)-ABS(R=25)!
159
1470 CALL HCHAR(R,C,42)!104
1480 GOTO 1420 !224
1490 PRINT "LIST 1490-1580"
!150
1492 BREAK !142
1494 REM - Joystick moves cu
rsor in 8 directions, wraps
around borders !085
1500 R=16 !065
1510 C=12 !046
1515 CALL CLEAR !209
1520 CALL JOYST(1,DX,DY)!009
1530 C=C+DX/4 !098
1540 C=INT(32*((C-1)/32-INT(
(C-1)/32)))+1 !019
1550 R=R-DY/4 !130
1560 R=INT(24*((R-1)/24-INT(
(R-1)/24)))+1 !067
1570 CALL HCHAR(R,C,42)!104
1580 GOTO 1520 !068
1590 PRINT "LIST 1590-1770"
!152
1592 BREAK !142
1594 REM - Creates random sy
mmetrical characters  !185
1600 CALL CLEAR !209
1610 REM - see DIM A$(16) in
 line 105 !016
1620 DATA 18,24,3C,42,5A,66,
7E,81,99,00,A5,BD,C3,DB,E7,F
F !244
1625 RESTORE 1620 !183
1630 FOR J=1 TO 16 !112
1640 READ A$(J)!179
1650 NEXT J !224
1655 RANDOMIZE !149
1660 FOR CH=129 TO 136 !082
1670 FOR L=1 TO 4 !062
1680 X=INT(16*RND+1)!216
1690 B$=B$&A$(X)!108
1700 C$=A$(X)&C$ !110
1710 NEXT L !226
```

(See Page 12)

HANDY DANDIES—

(Continued from Page 11)

```
1720 CALL CHAR(CH,B$&C$)!183
1730 B$=NUL$ !055
1740 C$=NUL$ !056
1750 PRINT CHR$(CH): :!212
1760 NEXT CH !033
1770 GOTO 1660 !209
1780 PRINT "LIST 1780-1830"
!150
1782 BREAK !142
1784 REM - This routine give
s you the ASCII code and pri
nt character for all keys an
d key combinations !048
1790 CALL KEY(0,K,ST)!015
1800 IF ST=0 THEN 1790 !095
1810 PRINT K;CHR$(K)!041
1820 GOTO 1790 !083
1830 REM - Try 1 and Z toget
her; CTRL and 1; FCTN and 1;
 etc. !041
1840 PRINT "list 1840-2060"
!015
1842 BREAK !142
1844 REM - Long Shell Sort -
 to sort words into alphabet
ic sequence - fastest method
 for long lists !044
1846 CALL CLEAR !209
1850 PRINT "First we will ge
nerate 50   words": :!073
1855 REM  - DIM D$(52) is in
 line 105  !242
1860 N=50 !059
1861 GOSUB 1863 !158
1862 GOTO 1868 !162
1863 FOR J=1 TO N !141
1864 D$(J)=CHR$(INT(26*RND+6
5))&CHR$(INT(26*RND+65))&CHR
$(INT(26*RND+65))!016
1865 PRINT D$(J)&"  ";!048
1866 NEXT J !224
1867 RETURN !136
1868 PRINT : :"Starting to s
ort": :!126
1900 D=N !080
1910 D=INT(D/3)+1 !253
1920 FOR I=1 TO N-D !146
1930 IF D$(I)<=D$(I+D)THEN 2
010 !156
1940 T$=D$(I+D)!089
1950 J=I !081
1960 D$(J+D)=D$(J)!001
1970 J=J-D !088
1980 IF J<1 THEN 2000 !215
1990 IF T$<D$(J)THEN 1960 !0
02
2000 D$(J+D)=T$ !090
2010 NEXT I !223
2020 IF D>1 THEN 1910 !120
2030 FOR J=1 TO N !141
2040 PRINT D$(J)&"  ";!048
2050 NEXT J !224
2055 STOP !152
2060 REM - To sort numbers,
just delete all the $ !145
2070 PRINT "LIST 2070-2230"
!138
2072 BREAK !142
2074 REM - Shuttle sort - br
ief routine to sort short li
sts, too slow for long lists
     !166
2080 N=10 !055
2090 GOSUB 1863 !158
2100 REM - That generated 10
 words !087
2110 PRINT : :"Starting to s
ort": :!126
2120 FOR I=1 TO N-1 !072
2130 FOR J=I TO 1 STEP -1 !2
46
2140 IF D$(J)<=D$(J+1)THEN 2
190 !009
2150 T$=D$(J)!085
2160 D$(J)=D$(J+1)!183
2170 D$(J+1)=T$ !016
2180 NEXT J !224
2190 NEXT I !223
2200 FOR J=1 TO N !141
2210 PRINT D$(J)!187
2220 NEXT J !224
2225 STOP !152
2230 REM - To sort numbers,
delete all $$ !110
2240 PRINT "LIST 2240-2440"
!140
2242 BREAK !142
2244 REM - To set up card de
ck   !012
2245 REM  - See DIM A$(16),D
$(52) in line 105  !124
2246 RESTORE 2251 !048
2247 GOSUB 2260 !044
2250 STOP !152
2251 DATA HEARTS,DIAMONDS,CL
UBS,SPADES,ACE,2,3,4,5,6,7,8
,9,10,JACK,QUEEN,KING !035
2260 FOR J=1 TO 4 !060
2270 READ SUIT$(J)!183
2280 NEXT J !224
2290 FOR J=1 TO 13 !109
2300 READ A$(J)!179
2310 NEXT J !224
2320 FOR J=1 TO 4 !060
2330 FOR L=1 TO 13 !111
2340 N=N+1 !021
2350 D$(N)=A$(L)&" OF "&SUIT
$(J)!047
2355 PRINT D$(N)!191
2360 NEXT L !226
2370 NEXT J !224
2380 RETURN !136
2450 PRINT "LIST 2450-2560"
!146
2452 BREAK !142
2454 REM - Quick Shuffle - G
OSUB to Item #15 to get deck
     !243
2456 RESTORE 2251 !048
2460 GOSUB 2260 !044
2464 PRINT : :"Now shuffling
": :!094
2470 FOR J=52 TO 1 STEP -1 !
222
2480 K=INT(J*RND+1)!228
2490 TEMP$=D$(J)!055
2500 D$(J)=D$(K)!253
2510 D$(K)=TEMP$ !056
2520 NEXT J !224
2530 REM - To demonstrate !1
76
2540 FOR J=1 TO 52 !112
2550 PRINT D$(J)!187
2560 NEXT J !224
2562 STOP !152
2570 PRINT "LIST 2570-2640"
!148
2572 BREAK !142
2574 REM - To pick random nu
mbers without duplication !0
94
2575 N=10 !055
2580 FOR J=1 TO N !141
2590 A@(J)=INT(N*RND+1)!213
2600 FOR T=1 TO J-1 !079
2610 IF A@(J)=A@(T)THEN 2590
 !093
2620 NEXT T !234
2625 PRINT A@(J)!212
2630 NEXT J !224
2635 STOP !152
2640 REM - For many numbers,
 Item #18 is faster !103
```

(See Page 13)

HANDY DANDIES—

(Continued from Page 12)

```
2650 PRINT "LIST 2650-2760"
!150
2652 BREAK !142
2654 REM - Number shuffle !1
57
2660 N=100 !104
2670 FOR J=1 TO N !141
2680 D@(J)=J !067
2690 NEXT J !224
2700 FOR J=N TO 1 STEP -1 !2
51
2710 K=INT(J*RND+1)!228
2720 T=D@(J)!077
2730 D@(J)=D@(K)!053
2740 D@(K)=T !078
2750 NEXT J !224
2752 FOR J=1 TO N !141
2754 PRINT D@(J);!139
2756 NEXT J !224
2758 STOP !152
2760 REM  - Random numbers c
an now be pulled in sequence
 , d@(1) to d@(n), without d
uplication  !219
2770 PRINT "LIST 2770-2820"
!150
2772 BREAK !142
2774 REM - INPUT of string -
 will respond to word or ini
tial letter, upper or lower
case !187
2775 REM - Try Y,YES,y,yes !
030
2780 INPUT "PLAY AGAIN? YES
OR NO?":Q$ !092
2790 IF (SEG$(Q$,1,1)="Y")+(
SEG$(Q$,1,1)="y")THEN 2810 !
162
2800 GOTO 210 !033
2810 PRINT "Q$= ";CHR$(ASC(Q
$))!238
2820 GOTO 2780 !053
2830 PRINT "LIST 2830-2890"
!154
2832 BREAK !142
2834 REM - This routine prev
ents error tone and message
if letter is input instead o
f number !133
2835 GOTO 2840 !114
2838 PRINT : :"TRY AGAIN!!":
 :!003
2840 INPUT "SELECT SKILL LEV
EL, 1 TO 5 ":Q$ !074
2845 REM - Try inputting let
ters !139
2846 FOR J@=1 TO LEN(Q$)!054
2850 IF (ASC(SEG$(Q$,J@,1))<
48)+(ASC(SEG$(Q$,J@,1))>57)T
HEN 2838 !126
2852 NEXT J@ !032
2860 Q=ASC(Q$)!205
2870 PRINT CHR$(Q)!048
2880 GOTO 210 !033
2890 REM - alternately - IF
(Q$="1")+(Q$="2")+(Q$="3")+(
Q$="4")+(Q$="5")THEN 2860 EL
SE 2840 !076
2900 PRINT "LIST 2900-3000"
!136
2902 BREAK !142
2904 REM - This routine sets
 up the frequencies of 3 oct
aves of music in a subscript
ed variable !029
2910 DATA A,B flat,B,C,C#,D,
E flat,E,F,F#,G,A flat !137
2915 RESTORE 2910 !198
2920 F=110 !097
2930 FOR J=0 TO 35 !112
2940 D@(J)=F*1.059463094^J !
027
2950 CALL SOUND(-4000,D@(J),
5)!081
2960 READ N$ !009
2970 PRINT "N(";STR$(J);")="
;INT(D@(J));"=";N$ !014
2980 IF (J<>11)*(J<>23)*(J<>
35)THEN 3000 !117
2990 RESTORE 2910 !198
3000 NEXT J !224
3002 PRINT "D@(36)=40000=A R
EST" !193
3004 D@(36)=40000 !163
3005 STOP !152
3010 PRINT "LIST 3010-3130"
!133
3012 BREAK !142
3014 REM - This routine sets
 up the codes to play the tw
o 'secret' octaves of bass n
otes !028
3016 REM  - Note DIM D@(100)
in line 105  !210
3020 RESTORE 2910 !198
3040 F=413 !103
3050 FOR J=0 TO 24 !110
3060 D@(J)=F*1.059463094^J !
027
3070 CALL SOUND(-4000,30000,
30,30000,30,D@(J),30,-4,0)!0
68
3075 READ N$ !009
3080 PRINT "D@(";STR$(J);")=
";INT(D@(J));"=";N$ !069
3090 IF (J<>11)*(J<>23)THEN
3110 !185
3100 RESTORE 2910 !198
3110 NEXT J !224
3115 D@(25)=40000 !161
3120 PRINT "D@(25)=40000=RES
T" !092
3130 REM  - The bass note mu
st be programmed as in line
3070, as the 3rd tone with v
olume 30 and with -4 as 4th
tone at audible level !038
3135 STOP !152
3140 PRINT "LIST 3140-3200"
!135
3142 BREAK !142
3144 REM - Flash print !098
3145 CALL CLEAR !209
3150 FOR SET=5 TO 8 !230
3160 CALL COLOR(SET,1,1)!156
3165 NEXT SET !130
3170 PRINT "THIS ROUTINE FLA
SHES THE    TEXT ON THE SCRE
EN SUDDENLY": : : : : : : :
: : : : : : :!197
3180 FOR SET=8 TO 5 STEP -1
!084
3190 CALL COLOR(SET,2,1)!157
3200 NEXT SET !130
3205 GOTO 3205 !224
3210 PRINT "LIST 3210-3280"
!141
3220 BREAK !142
3230 REM - To put numbers in
 parentheses !056
3240 DEF N@$="("&STR$(N)&")
" !001
3250 FOR N=1 TO 10 !110
3270 PRINT N@$;"NUMBER";N !1
54
3280 NEXT N !228
3500 END !139
5000 PRINT "THIS PROGRAM DOE
S NOT HAVE  ENOUGH FREE MEMO
RY LEFT TO  DEMONSTRATE THAT
 ROUTINE." !094
5002 FOR D=1 TO 500 !153
5004 NEXT D !218
5010 GOTO 210 !033
```

Hardware project

Modification of the TI reset circuit

By RICHARD J. BAILEY

An annoying problem with the operation of an expanded system is that you have to turn the console on last or the initialization on power-up may not be performed in the proper sequence. If other people like myself have a switched outlet strip to turn every piece of equipment in the system on and off simultaneously then they too have probably forgotten to turn the console off then back on to assure proper initialization. You generally discover this after you try to save a program on disk only to find that the console doesn't think that the disk drive exists.

For those people who have expertise in electronics there is a relatively simple modification to the power-up reset circuit in the console that will allow one switch operation and assure proper initialization. The added circuitry will cost under $5 for two diodes, one resistor, and one capacitor. No holes have to be drilled, no runs have to be cut, and you can convert the console in under one hour.

No persons should attempt this modification unless they are competent to finish the job themselves. I have made this modification to my own 99/4A but assume no responsibility for any disaster that may befall others who may try to implement this change. Now that I've issued my disclaimer, details for the change.

Other than a two-second delay on power-up reset, the reset will act the same as before. If the console locks up when running, you can still switch it off then on to reset it. If you insert a cartridge it will still reset the console.

Power-up reset is accomplished by a simple resistor-capacitor charging circuit associated with the 9904 (74LS362) clock chip which supplies the 3 MHZ clock signal to the 9900 CPU chip. To increase the reset time you could increase the value of the timing capacitor. Unfortunately this reset line is also used when a cartridge is inserted so the circuit has to be a little more complicated than just an added capacitor.

The added circuitry is shown within the dotted lines of figure 1. When you turn the console on the original 12 K OHM resistor will charge both the original 22 MFD capacitor and the added 150 MFD capacitor through the diode connected between the two capacitors but the added 150 OHM resistor will not contribute any charging current because of the reverse-biased diode connected in series with it. When you turn the system on, it will take two seconds to charge the two capacitors. If you are running the system and need to turn the console off momentarily to reset it, the 22 MFD capacitor discharges in about one second and is isolated from the 150 MFD capacitor by the reverse-biased diode between them. If the diode were not there it would take longer for this 150 MFD capacitor to discharge and allow a reset plus the cartridge reset wouldn't work.

When you are through using the system and shut it down, the diode between the two capacitors is reverse-biased and the 150 MFD capacitor cannot discharge through this path. The function of the second diode and 150 OHM resistor that were added is to discharge the 150 MFD capacitor when you shut the system down. So the net effect is that every feature of the console remains the same except the power-up reset is delayed for about two seconds.

My console has the brushed aluminum

(See Page 15)

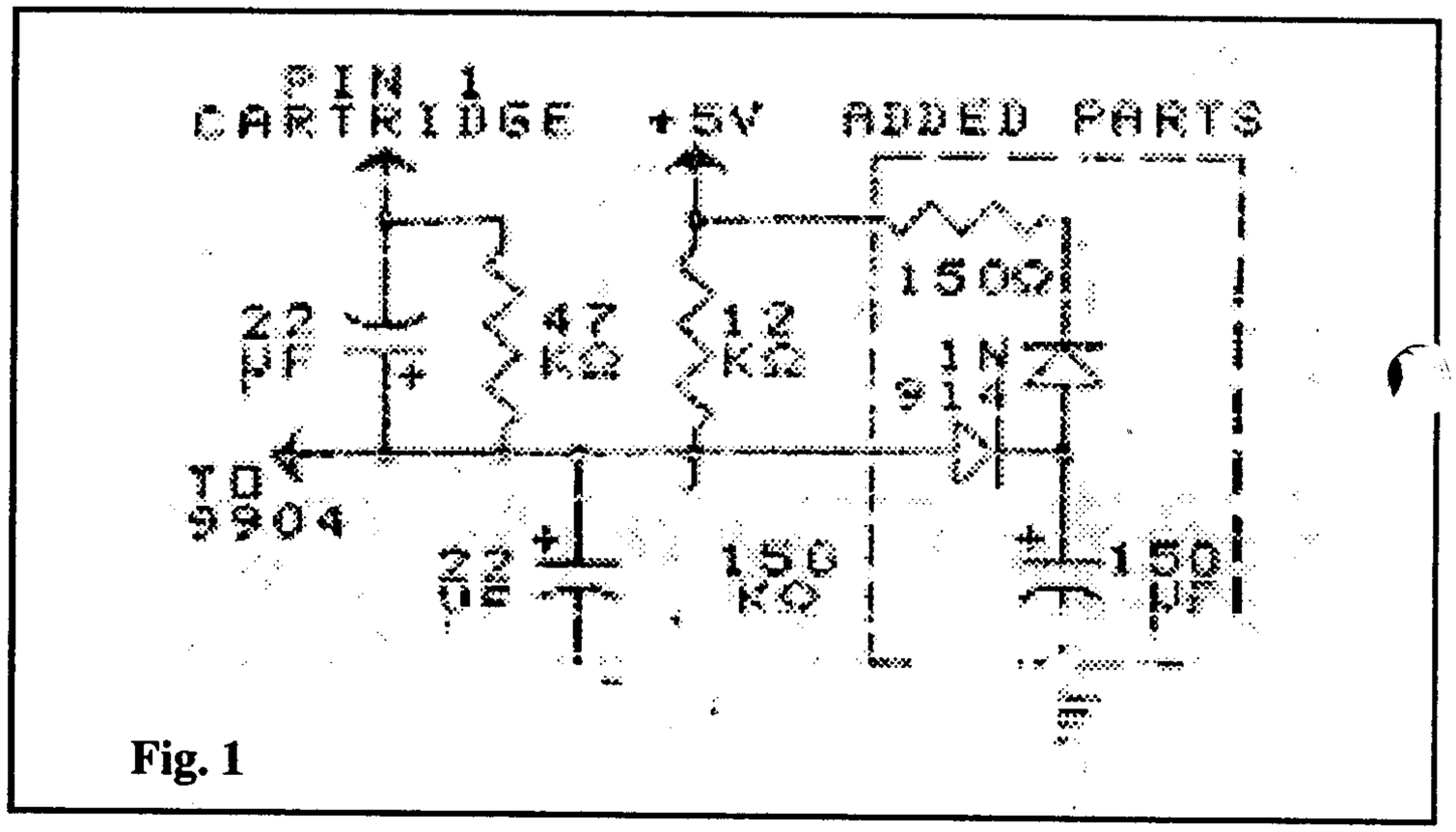


Fig. 1

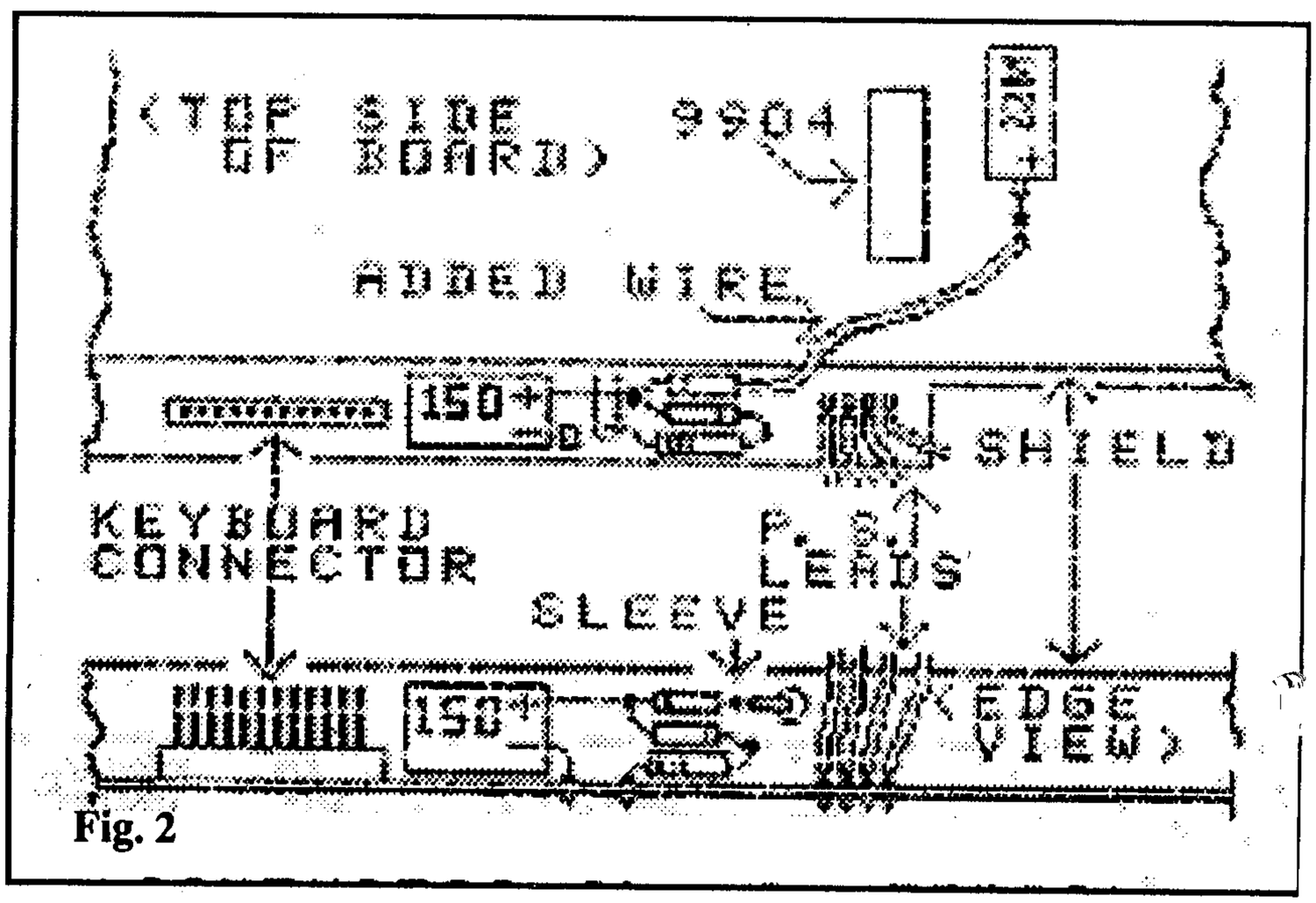


Fig. 2

MODIFICATION OF THE TI RESET CIRCUIT —

(Continued from Page 14)

trim and I had to perform the following steps to modify it.

1. Disconnect power and all cables.

2. Pull straight out (toward front) on plastic on-off switch to remove it.

3. Remove the seven screws that hold the bottom half of the case and remove it.

4. After noting the exact location of the power supply leads, remove its two screws, unplug main board, set power supply to one side.

5. Make sure there is room between the keyboard and the main board for the 150 MFD capacitor that you are adding. I used a solid tantalum capacitor for size.

6. Remove three screws holding main board(one in middle of board).

7. Lift main board about one inch and CAREFULLY unplug keyboard. (avoid excess flexing of these leads!)

8. Unplug cartridge connector from main board.

9. Remove the two clips and three screws holding the shields-note the location and shape of the two clips and length and direction of screws through the shields. IMPORTANT!!!

10. Solder one end of a four inch long piece of stranded insulated wire to the positive end of the 22 MFD capacitor (C506) nearest the 9904 (74LS362) clock chip. This chip may or may not have heatsink grease on it. Do not wipe grease off!

11. Resmear grease over tops of chips that have it and reassemble shields making sure the shield pieces near the expansion box connector slide together properly. The free end of the added wire is put through a side hole close to the right end of L115 on edge of board near the four power supply wires. Do not pinch wire between crystal and shield, etc..

12. Reassemble shield clips and screws. Plug in cartridge connector.

13. Slide one inch long piece of sleeving over free end of added wire.

14. Place components as shown in drawing using existing holes in board. The L-shaped run is +5v. and the dot-shaped pad is ground.

15. Solder all connections. Make sure to slide sleeving over diode-wire connection. Push excess wire inside shield.

16. Double-check all wiring and polarity of added parts.

17. Reassemble console by reversing steps 1-7, making sure that when you replace the power supply the switch lever fits into the slot in the slide. Check to make sure that it slides properly. Use care when reconnecting keyboard.

Reconnect the console and turn everything on except the console. After you have a blank screen on the TV, turn on the console and note the time delay until you get the title screen. This should be about two seconds. Switch console off then on to check for about one second delay before display of title screen. Plug cartridge in to check for one second delay before display of title screen.

If the console passes these tests the modification is correct and you can use one main switch to turn the system on and off.

This article originally appeared in the newsletter of the New Hampshire 99er User Group.—Ed.

K'town officer William Sheridan dies

William M. Sheridan died on February 24 at age 70 after a bout with lung cancer.

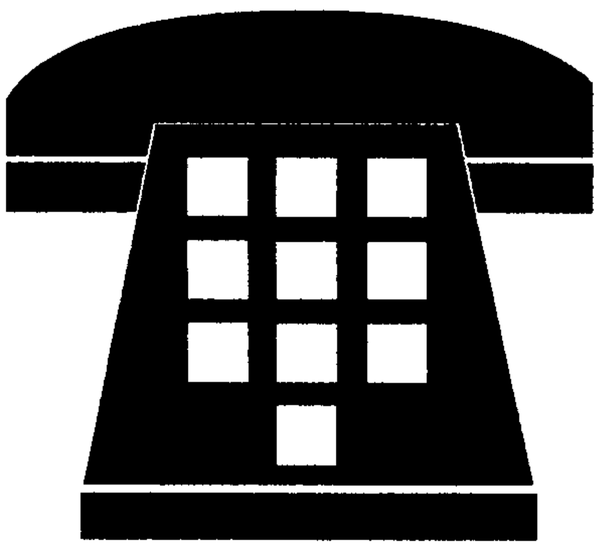
He is survived by his wife Kitty and two daughters.

He has been for several years the vice president and editor of the newsletter for the K'town 99ers.

"He will be sorely missed by his family and by the members of the K'town 99ers in Knoxville and the surrounding towns," says John Bull of the K'town 99ers.

Picture exchange programs offered

Jeffrey A. Kuhlman notes that he has written program to view monochrome TI-Artist _P files and Instances on the Sinclair QC and IBM computers (DOS program) for users who want to exchange pictures with the owners of one of these computers. Write him for information at CMR 4/6, Box D, APO, AE 09140.



Want to talk to someone at MICROpendium? You'll need to called between the hours of 9 a.m. and noon Saturdays. If you call at other times, you will probably get an answering machine. But don't let that bother you. We listen to the answering machine at least once a day and return calls as soon as possible, usually that day.

Call us at 512-255-1512

More Extended BASIC myths

You'll be surprised, mystified and wonder whether you can trust the manual

By BRUCE HARRISON

Recently, we've been playing around some more with Extended BASIC, mainly developing new Assembly routines to use with it, but in the process, we've uncovered some interesting things that many XB programmers probably don't know. We can't exactly call these things myths, because some of what we've found runs directly opposite to what the XB book says. We'll start with a couple of surprises.

THINGS THAT NEEDN'T RUN

What do we mean by "needn't run"? We mean there are statements in XB that get handled only during the pre-scan, and can be put into the program so that they don't get "run," but only pre-scanned. So far, there are three that we've proven to our own satisfaction. These are the DEF, DIM, and OPTION BASE statements. Yes, it's true that OPTION BASE has to precede any DIM, but they can be in the same program line, and that line need not execute during the running of the program. Here's an example program you can run to see that this is so:

```
10 CALL CLEAR :: RANDOMIZE :
: GOTO 20 :: OPTION BASE 1 :
: DIM A(52),B(52):: I,X :: !
@P-
20 PRINT "STARTING ASSIGNMEN
T" :: FOR I=1 TO 52 :: A(I)=
I :: NEXT I
30 FOR I=52 TO 1 STEP -1
40 X=INT(RND*I+1):: B(53-I)=
A(X):: A(X)=A(I):: NEXT I ::
 PRINT "ASSIGNMENT DONE":"CH
ECKING ";
50 FOR X=1 TO 52 :: FOR I=1
TO 52
60 IF B(I)=X THEN 80
70 NEXT I :: PRINT "ERROR';X
;"NOT FOUND" :: STOP
80 NEXT X :: PRINT "- DECK O
KAY ":"HERE'S THE SHUFFLED D
ECK"
90 FOR I=1 TO 52 :: PRINT B(
I);:: NEXT I
```

That's a short but complete program. It makes a deck of 52 "cards" in A(), then shuffles them into array B(), checks to make sure none are missing, and then prints the shuffled "deck" on the screen. Notice that in line 10, after we CALL CLEAR and RANDOMIZE, we skip over the rest of that line with GOTO 20. Thus the OPTION BASE 1 and DIM A(52),B(52) don't get executed. They don't need to, because the actions taken in response to those statements get performed during pre-scan.

What do we mean by 'needn't run'? We mean there are statements in XB that get handled only during the pre-scan, and can be put into the program so that they don't get "run," but only pre-scanned.

You'll notice also that we've put our two simple variables I and X into that line, then ended line 10 with :: !@P-. There's another myth exploded for you. !@P- can be included within a program line that does other things, provided only that it's preceded by a double colon. We always thought that this statement needed its own program line, but obviously that too is not true. To prove to yourself that !@P- works when done this way, edit line 10 in the part that says I,X. Change I to T, then run the program. It will stop with a syntax error in 20 message, because the variable I was not found in pre-scan. Change that T back to an I, and the program will run correctly.

The algorithm in lines 20-40 of this program is the most efficient and fastest way we've been able to devise so that XB would assign numbers randomly without replacement. Its design is based on emulating a routine we created in Assembly. The Assembly routine works much faster than this one, but this takes advantage of some "lessons learned" from the Assembly routine.

We haven't shown an example of the DEF statement, but have tested and can assure you that it works when in the pre-scan but not in the executed program. We never use DEFs in our own work, because using them seriously slows down execution of things. XB can be slow enough without doing things like DEF.

USING INT WITH RND

You'll notice that in the first program shown, we used the conventional scheme of INT(RND*I+1) to take our random numbers. In that case, it was necessary because we had OPTION BASE 1 in effect. When OPTION BASE 1 is not in effect, you can get away without the INT. Here's another program to illustrate that:

```
10 RANDOMIZE :: CALL CLEAR :
: GOTO 20 :: CALL KEY :: A()
,B(),I,X :: !@P-
20 PRINT "STARTING ASSIGNMEN
T" :: FOR I=0 TO 9 :: A(I)=I
+1 :: NEXT I
30 FOR I=9 TO 0 STEP -1
40 X=RND*I :: B(9-I)=A(X)::
A(X)=A(I):: NEXT I
50 PRINT "ASSIGNMENT DONE"
60 PRINT "CHECKING ";:: FOR
X=1 TO 10 :: FOR I=0 TO 9
70 IF B(I)=X THEN 90
80 NEXT I :: PRINT "ERROR";X
;"NOT FOUND" :: STOP
90 NEXT X :: PRINT "- CHECKS
 OKAY"
100 FOR I=0 TO 9 :: PRINT "
"&STR$(B(I));:: NEXT I
110 PRINT : : :: CALL KEY(0,
I,X):: IF X=0 THEN 20
```

This little program will just keep putting the numbers from 1 through 10 on the screen in random order until you stop by holding down any key, or by pressing Function-4. Here, since we are using numbers 0 through 9 of A() and B(), we don't

(See Page 17)

MORE MYTHS—

(Continued from Page 16)

need the DIM, but just need A(),B() listed within the pre-scanned part of the program. We don't need to INT the RND*I in line 40. This means that on each pass the value of X can run from 0 times I through .9999999999 times I. Since X is only used as the index for arrays, the indexing function will round off the number in X before using it to pick a member of the array. Thus on the first pass, when I is 9, the range of X will be 0 through 9 times .9999999999 etc. This latter will round to 9, so that the range in terms of indexing will be correct at 0 through 9. There's another thing we'll bet most of you didn't know, that the indexing function rounds numbers instead of truncating! Here's a quick experiment you can try in command mode, without even having a program. First, type this:

```
FOR I=0 TO 9 :: A(I)=I ::
NEXT I <ENTER>
```

Now type in the following two things:

```
PRINT A(8.51) <ENTER>
PRINT A(8.49) <ENTER>
```

The first will print 9, and the second will print 8, thus proving our point. Any entry between those parentheses that's 8.5 through 9.4999999 will take A(9). Any entry between 7.5 and 8.4999... will yield A(8), which will print 8. Try as many as you like. A(.5) will print 1, A(.499) will print 0, and so on. Q.E.D.

Thus, in this case, we didn't need to do that INT. We wouldn't bother mentioning this if it weren't that INT takes time to execute, so when we can do without it, we'd rather omit it than use it. If OPTION BASE 1 is in effect, you can still avoid use of INT, but you'll have to do something about the rounding done by the indexing of arrays. You can try this in the first program in this article. Change the first statement in line 40 from X=INT(RND*I+1) to X=RND*I+.5 and run it that way. We've done that, and it worked fine for about 50 runs, so we think it will always work. To run that many trials, we added a line to the program:

```
100 PRINT :: GOTO 20
```

That keeps the program repeating itself until we stop it with a Function-4, so we can observe runs until we're sick of doing so. Just to be sure, we let this run repeating itself for about half an hour. It worked as expected. In that time, 1 came up as the first number at least once, as did 52. Who says we have no patience? After the first 15 minutes, we stopped the program and took out the "checking" part, since the checking takes more time than the assignment of the numbers does. Okay, so we are impatient! Testing things that use random numbers can take forever waiting to see that certain outcomes are possible.

We've shown these programs with a section that checks our results so that you can be confident they do what we've said. Of course in any real application we'd not have the checking section in the final version, and neither should you. Enough with these random numbers and OPTION BASE 1, let's get on to a new topic.

DATA AND PRE-SCAN AVOIDANCE

Yes, it's true that DATA, if used in your program, must be pre-scanned in most cases, but that doesn't mean you have to put any DATA lines in the beginning of the program. Try the following small example:

```
10 GOTO 20 :: CALL KEY :: A$
,I,S :: !@P-
20 FOR I=1 TO 6 :: READ A$ :
: PRINT A$;" ";:: NEXT I
30 !@P+
40 DATA ANY,OLD,STRING,WILL,
DO,HERE
50 PRINT :: !@P-
60 RESTORE :: CALL KEY(0,I,S
):: IF S=0 THEN 20
```

Just so long as we turn on the pre-scan with !@P+ before the DATA line, this will work. Notice, however, that while we could use !@P- at the end of line 10, !@P+ won't work that way, but has to have its own line number. Neither !@P- nor !@P+ can be tacked onto the end of a DATA line. The indication DATA at the beginning of a line makes the pre-scan ignore all content beyond that. Thus we can't turn off the pre-scan again by adding !@P- at the end of the data line. Here, we've done it after the PRINT statement and double colon in line 50, and that works. You can prove it by changing the I to K in line 60, and you'll get a syntax error in 60 message when it runs.

Of course the XB manual itself was a little lacking on this subject of pre-scan avoidance. The subject was mentioned and illustrated to some degree in an addendum, but some of what was said there appears to have been misleading.

There's another way to make DATA work in your program without needing the pre-scan on. Try this small example:

```
10 GOTO 20 :: CALL KEY :: A$
,I,S :: !@P-
20 RESTORE :: FOR I=1 TO 6 :
: READ A$ :: PRINT A$;" ";:: 
NEXT I
30 DATA ANY,OLD,STRING,WILL,
DO,HERE
40 CALL KEY(0,I,S):: IF S=0
THEN 20
```

That will work, because the RESTORE is there before we try to read any DATA statements. The RESTORE itself need not be included in the pre-scan either. If need be, RESTORE (line number) can also be used, with the same effect. This may be less trouble than the other method, in that you don't have to bother with a separate program line to turn on the pre-scan, and you don't even have to know where the first line of data is. This appears to work because RESTORE will seek out the DATA token wherever that is in the program, and set the data pointer to that line, wherever it happens to be, even though pre-scan did not find any DATA in the program.

IMAGE — A STRANGE CASE

In the beginning of this article, we talked about some things that need not be executed, but only included in the pre-scanned part of the program. The IMAGE statement goes farther than that, in that one can use IMAGE in lines that are neither executed nor pre-scanned. Here's an example:

```
10 GOTO 40 :: !@P-
20 IMAGE $##.##
30 IMAGE ##.#
40 PRINT USING 20:33.5
50 PRINT USING 30:24.75
```

When this runs, you'll see two lines printed on the screen. The first will read $33.50, and the second 24.8. Yes, that's right, using the IMAGE in line 30 makes the displayed number correctly round to the digits shown. If the number at the end of line 50 were 24.74, it would be printed

(See Page 18)

MORE MYTHS—

(Continued from Page 17)

by line 50 as 24.7. You can prove that to yourself by adding to the end of line 50 so that it reads:

```
50 PRINT USING 30:24.75,24.7
4
```

Now three lines will print, and the second and third will show correct rounding of those two numbers to a single decimal place. Note that we've used a comma to separate the two numbers to be printed in the revised line 50, yet the numbers print on a separate line. The other two "normal" print separators, colon and semicolon, will not work in this case. If we want those two numbers to appear in the same print line, we can do that by changing line 30 to read:

```
30 IMAGE ##.# ##.#
```

That's a space between the two ##.# strings. With this change, the two roundings of the numbers in line 50 will show up as 24.8 24.7, with just one space between, not the two that normally would go between two positive numbers. We can put another space into line 30, and sure enough we'll get two spaces between the numbers. We can also make line 30:

```
30 IMAGE ##.# & ##.#
```

As the book indicates, the printout will be 24.8 & 24.7. Sometimes the book is actually right! It even said that IMAGE lines don't get executed, but never said they didn't need to be prescanned.

We hope that some of you have learned something you didn't know about Extended BASIC. If so, then the work of preparing this little treatise has been worthwhile.

Parsec tips

Beating the Urbites and Swoopers a matter of knowing the enemey

By BOB GAGLE

Winning at Parsec requires more than just flying through a few asteroid belts, landing in the refueling tunnel and knocking off every alien fighter or cruiser you come across. To be a true Parsec master you need good eye-hand coordination, quick reflexes and, most important, a winning strategy. The following is a careful analysis of each of the Parsec enemies their individual habits and peculiarities and a collection of tips for nailing the little nasties before they destroy you.

SWOOPERS AND LTF

First, let's take a look at the Swoopers. These enemy craft look like large-winged jets. They come in all colors and enter from the top of the screen, increasing their speed as the game progresses.

Never underestimate the power of these ships. Although they will not fire at your craft, they do have a tendency to ram into their enemies. When you encounter a Swooper try to stay in the far left hand corner of the screen. (Actually, it's a good idea to always stay as far to the left side of the screen as possible.) When battling Swoopers you want to move fast, so it is best to use lift 3. In later levels, however, it might be necessary to use lift 2. for more precise aiming capabilities.

A sleeker version of the Swooper, the LTF will emerge from the top of the screen and accelerate steadily. These multi-colored ships also resemble the Swooper in that they will not fire upon you. Their speed changes, however, are much more dynamic. They like to fly low, forcing you to crash into the plane. So stay in the middle of the screen until they come up. The best lift for this level is 3, but be ready to change to 2 in dangerous situations.

In my opinion, it is the Saucers who are the trickiest adversaries because they come from behind, seemingly out of nowhere. But never fear, they can be destroyed.

URBITES AND BYNITES

Shaped like.tiny bullets, the Urbites are armed with two cannons each. When these ships are announced, fly immediately to the extreme top of the screen. Because they follow your vertical movements only very slowly, just move and fire. Stay away from the bottom of the screen, and you will be safe.

Bynites are very similar to Urbites, and can be easily destroyed. Begin as close to the planet as possible, using lift 3. When the Bynite comes out, move all the way to the top of the screen; fire at it when it gets there and then move. It works every time!!

DRAMITES

Dramites look exactly the same as Urbites, but they track faster and have only one cannon. People say that these ships are the most deadly enemies in Parsec, but they can be easily destroyed by following these hints :

1. Always stay on lift 3 because Dramites are quite fast in tracking vertical movement.
2. Start as close to the surface of the planet as you can.
3. When the Dramite comes out, go up and down while firing occasionally, letting the ship follow you into your laser.

TRICKY SAUCERS

In my opinion, it is the Saucers who are the trickiest adversaries because they come from behind, seemingly out of nowhere. But never fear, they can be destroyed. If you have four or more ships in reserve, the Saucers will attack in random patterns. The best thing to do is stay on lift

(See Page 19)

Relative files

Extended BASIC programs focus on languages

By LUCIE DORAIS

For a long time I had wanted to introduce the concept of *Relative* files. The set of programs for this month and next are built around a database of words in five languages, INTWORDS. Once that database is resting safely on a disk, we can use it for many programs.

This month, we start with the learning module INTLEARN (all the filenames in the collection start with INT so you can find them easily on your diskettes), and next month we will play games to test our knowledge using INTSCRAM (Unscramble) and INTHANG (or Wheel of Fortune sans wheel and sans Vanna). The short program INTMENU is the menu for the three programs.

What are relative files? Normally, when you save to disk, the data is sent one line after the other. To get it back, the computer has to read the whole file from the beginning, even if you need only part of the data. This is called *Sequential* filing. With relative files, each record gets a unique RECord number (a record can be text, numbers or a mix of both), and the computer can access each record individually, very quickly. But the main advantage is that you can have as much data as your diskette will allow, and it does not clog the computer's memory.

Another advantage that we will use with my INTernational set of programs is that the same data can be used by many different programs, without having to type all the data for each one.

A third advantage is that you can easily replace data directly on the disk, one record at a time, and that can even be done in command mode (you first have to open the file). A file on disk can easily be expanded in APPEND mode, but I will leave that to you. But, because of all those advantages, a relative file cannot be kept on a cassette tape.

The data file for the INTernational programs is INTWORDS, a list of 100 words in English, with their translation in German, French, Italian and Spanish. By running INTLEARN, you'll see for yourself if it is true that German resembles English (Saxon languages) or if French, Italian and Spanish are really that close (Latin languages). All four foreign languages use accents, which English does not, and I have included some examples here.

Even if the files INTWORDS and INTNDX are already built and included on a diskette (they are included on the MICROpendium disk—Ed.), I have decided to keep the following text so you can learn how they were built. The text file WORDS, which was used to build INTWORDS, is also included so you can make changes to it, or add to it if you want.

There are two ways to build the INTWORDS database: you can type the text file WORDS into TI-Writer or the Editor/Assembler editor, copying exactly the data found at the end of this article (not included in this version of my tutorial), or use the INTDATA program given below. Whichever method you use, please note that the words with accents are preceded by a "*" flag, and that the accents are written after the letter they accent. Since the English ASCII set does not include a dieresis (two dots over a vowel) and that

(See Page 20)

PARSEC TIPS—

(Continued from Page 18)

2, and fly near the middle of the screen. If you notice a particular group is coming from the top or bottom, wait until they have been destroyed, then move your ship near their source. If too many Saucers are screen, switch to lift 3. And be careful! Sometimes when you fire your laser at Saucers on lift 3, it will go between the ship and the exhaust. It you have three or less ships in reserve, the Saucers will attack in a pattern starting at the top of the screen and moving down.

KILLER SATELLITES

Killer Satellites will appear after you complete the asteroid belt on level 4. Entering from all directions, moving erratically and firing frequently, these vicious foes are bent upon your destruction. They attack in random groups, at random speeds. The best strategy with these guys is to drop as far back as possible and use lift 3, because they are very unpredictable. There is no real sure way to destroy them.

ASTEROID BELTS

Now that you are aware of your enemies' foibles and idiosyncrasies, here are a few pointers to improve your own performance. When you are in the asteroid belt. always use lift 2, stay at the bottom of the screen, and fire continually at the lowest asteroid. That way, if you miss your target, you can dodge it and retreat to the protection of the planet. Occasionally, you may get trapped; if you get into trouble, use lift 3. And be wary of firing too much, overheating is very easy. Also, you should always use lift 1 in the refueling tunnel.

I have found that in playing Parsec, joysticks do not respond as well as the keyboard. Therefore I use the keyboard with the following finger placement: Left hand — middle finger on E key, pointer finger on X key, pinky and ring fingers control the lift. Right hand — pointer finger on the period key, middle finger controls pause (p key). For horizontal movement, I interchange the fingers on my left hand (on the E and X keys) to the S and D keys whenever needed. Always anticipate where the enemy is going, and stay calm while pressing the buttons on the keyboard. Remember that until you get accustomed to the keyboard, it will be difficult to play, because all it takes is a split second to get killed if you remove your eyes from the screen!

Good luck and happy Parsec-ing!

RELATIVE FILES—

(Continued from Page 19)

German uses it a lot, I have used the "#" character — don't worry, Tex will show the proper accent on your screen. Once your processed list is finished, you must save it with the PF option, not the SF one, to get rid of the tab line. Do PF, and when the printer name shows, write "C DSKn.WORDS", and the file will save to disk in D/V 80 format (C prevents any control codes to be saved, just in case).

Text files in TI are *Variable* format (always 80 as maximum length), and therefore cannot be used as a Relative file, which needs a *Fixed* length for each record (to allow for easy replacement of one record). They must therefore be read *sequentially* from the beginning. To create our Relative database, we use a small program, INTCOPY, that reads the WORDS file and copies every line to our database file, INTWORDS. Its record length is Fixed to 56: five words X eleven char. (with space or accent flag) plus one extra byte to hold the string length. Once a file is created with specific attributes (Variable/Fixed, Sequential/Relative, Display/Internal) you cannot open it any other way. Here below, in INTINDEX, even if INTWORDS is read sequentially, it still has to be OPENed as Relative.

INTCOPY

```
100 ! ** INTCOPY: copy recor
ds into INTWORDS **
110 DISPLAY AT(6,2)ERASE ALL
:"NOW COPYING LINE":"(total
100)"
120 OPEN #1:"DSK1.WORDS",DIS
PLAY,VARIABLE 80
130 OPEN #2:"DSK1.INTWORDS",
DISPLAY,RELATIVE,FIXED 56
  [one-
140 PRINT #2,REC 0:100

   time
150 FOR X=1 TO 100 :: DISPLA
Y AT(6,20):X :: INPUT #1:R$
   run]
160 PRINT #2,REC X:R$ :: NEX
T X
170 CLOSE #1 :: CLOSE #2 ::
END
```

As you can see in lines 120-130, each file is opened with its own attributes, WORDS is Sequential by default. To access a record by its number, we use ",REC^n" after the file number and before the colon, in line 140, we use record number zero to store the total number of data records. We do not need it with INTLEARN, but it might be useful if you build other programs around INTWORDS. I just wanted to show you how to store numeric values to a record. INTCOPY takes almost a minute to run.

If you don't have a word processor, you can still build the database by using the following program. Starting at line 1001, each line of the WORDS list is written as DATA (minus the spaces: line 170 will put them in). Type all 100 lines, until you reach line 1100. Be careful to include the accent flags "*" and the accents themselves (see lines 1004-1005).

To check the accuracy of the INTWORDS database, you can use yet another program, INTEST: you input the record number, and Tex will give you the content of that record, each word on a new line, preceded by a space or an accen flag.

INTDATA

```
100 ! ** INTDATA: use DATA t
o fill INTWORDS (if no TI-WR
ITER) **
110 DISPLAY AT(6,2)ERASE ALL
:"NOW WRITING LINE":"(total
100)"
120 OPEN #2:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56
130 PRINT #2,REC 0:100
140 FOR X=1 TO 100 :: DISPLA
Y AT(6,18):X :: R$="" :: FOR
 Y=1 TO 5
150 READ W$ :: IF Y=1 THEN L
=10 :: GOTO 170
160 L=11 :: IF ASC(W$)<>42 T
HEN W$=" "`W$
170 W$=W$`RPT$(" ",L-LEN(W$)
)
180 R$=R$`W$ :: NEXT Y :: PR
INT #2,REC X:R$
190 NEXT X :: CLOSE #2 :: EN
D
```

WORDS

```
1000 ! ** words **

  [one-
1001 DATA ALL,GANZ,TOUT,TUTT
O,TODO
   time
1002 DATA APPLE,APFEL,POMME,
MELA,MANZANA
   run]
1003 DATA ARM,ARM,BRAS,BRACC
IO,BRAZO
1004 DATA AUTUMN,HERBST,AUTO
MNE,AUTUNNO,*OTON~O
1005 DATA BACK,*RU#CKEN,DOS,
DORSO,ESPALDA(... .... ....)
1100 DATA YES,JA,OUI,SI,SI
```

To check the accuracy of the INTWORDS database, you can use yet another program, INTEST: you input the record number, and Tex will give you the content of that record, each word on a new line, preceded by a space or an accent flag. This program will work whatever manner you have used to build INTWORDS.

INTEST

```
100 ! *** INTEST: check INTW
ORDS ***
110 OPEN #2:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56
120 DISPLAY AT(5,1)ERASE ALL
:"REC #:":"(0 to end)": : :
:"words:"
130 ACCEPT AT(5,8):RN :: IF
RN>100 THEN 130
```

(See Page 21)

RELATIVE FILES—

(Continued from Page 20)

```
140 IF RN=0 THEN CLOSE #2 ::
 END ELSE LINPUT #2,REC RN:R
$
150 DISPLAY AT(12,1):SEG$(R$
,1,10):SEG$(R$,11,11):SEG$(R
$,22,11): SEG$(R$,33,11):SEG
$(R$,44,11):: GOTO 130
```

In INTLEARN, you are given a list of English words and you pick the one that you want to be translated. Tex will read the corresponding record from INTWORDS and display the English word and its four translations. So INTLEARN needs to keep only the English words in memory, instead of adding DATA lines that would repeat all the 100 English words (tedious work!). I decided to do like the big database programs do, build a separate INDEX file and then read only that file into memory. The following program, which needs to be run only once, reads INTWORDS sequentially and extracts the English words (line 140). Each time it has read 20 records, it prints the string IN$, which now contains 20 English words and their trailing spaces, to the INTNDX file (lines 150-160). Reading that file back into INTLEARN is a breeze, since it has only five records, versus 100 for INTWORDS.

INTINDEX

```
100 ! ** INTINDEX: build Eng
lish index from INTWORDS **
    [one-
110 DISPLAY AT(6,2)ERASE ALL
:"BUILDING ENGLISH INDEX:"
     time
120 OPEN #1:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56
     run]
130 OPEN #2:"DSK1.INTNDX",DI
SPLAY ,SEQUENTIAL,FIXED 201
140 FOR X=1 TO 100 :: LINPUT
 #1,REC X:R$ :: R$=SEG$(R$,1
,10)
150 IN$=IN$&R$ :: IF INT(X/2
0)<>X/20 THEN 170
160 DISPLAY AT(9,1):IN$ :: P
RINT #2:IN$ :: IN$=""
170 NEXT X :: CLOSE #1 :: CL
OSE #2 :: END
```

It is now time to attack INTLEARN. Since INTSCRAM and INTHAND were built around this file collection, the user-defined subs were typed first and saved in the MERGE format (SAVE DSKn.INTSUBS, MERGE). The SUB LANG puts sprites on the screen to label the words in each language; the sprites are just the over-sized letters contained in the string A$, read into B$ and C$. The digits in that string refer to the colors used for the sprites [K=VAL()].

The SUB SHOW extracts a word W$ from the record R$ at position P and displays it on screen. If there is an accent (IF P=1 there is never one, since the first word is English), the loop starting at line 1070 will find it, remove it from the word W$ and put it at the same position in the space string AW$. It will then display AW$ one line above W$.

The program will first read the INTNDX file and keep the total index in the IND$ array (lines 180) — this takes a second. Then the main data file INTWORDS is opened and the main screen is displayed (lines 200-230, which calls LANG to show our sprites in the right half of the screen). The 100 English words of the index are divided into five screens SCR (we start with SCR=1, line 210), each one dealing with the corresponding IND$() string. Each segment of ten characters (one word) is displayed on a separate row. The cursor row CR is set to one and put on screen, then Tex CALLs your KEY. You can press E or X (no need to press FCTN) to move the cursor up or down. Lines 310-330 deal with this move. "OCR" means old cursor row, to erase it. You can also press N or P for the Next or Previous screen (see lines 340-350. This is a cycle, the Next screen after 5 is 1, and vice-versa). And, of course, the familiar Q to quit.

You press <Enter> to tell Tex that you want to see the translation of a word (IF K=13 THEN 380, line 280). RECord number RN is derived from the screen number and the current cursor row (line 380). The proper record is read into memory from the data file INTWORDS, the right half of screen is erased, and the SUB SHOW is called five times. Please note that the words with accents takes a few nanoseconds to be displayed, since they have to be processed before showing themselves to the world. The CALL SOUND warns you

(See Page 22)

1995 TI FAIRS

APRIL

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

SEPTEMBER

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

1996 TI FAIRS

FEBRUARY

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

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

RELATIVE FILES—

(Continued from Page 21)

that the display is complete before you go back to the CALL KEY in line 280.

INTLEARN

```
100 ! ** INTLEARN ** - L.Dor
ais/Ottawa UG/Mar 92 !102
110 !!131
120 CALL CLEAR :: DIM IND$(5
):: CALL MAGNIFY(2)!168
130 A$="00000000" :: CALL CH
AR(39,A$&"10204",96,A$&"4020
1",94,A$&"205088",35,A$&"005
05",126,A$&"40A81")! accents
 !174
140 CALL CHAR(128,"181818181
8181818"):: CALL CHAR(129,"0
00000FFFF"):: CALL COLOR(13,
14,1)! lines !024
150 GOTO 160 :: A$,CR,K,OCR,
P,R$,RN,ROW,S,SCR,X :: CALL
HCHAR :: CALL VCHAR :: CALL
KEY :: CALL SOUND :: !@P- !0
69
160 DISPLAY AT(1,5):"LEARN F
OUR LANGUAGES": : : :"LOAD
ING INDEX" !178
170 OPEN #1:"DSK1.INTNDX",DI
SPLAY ,SEQUENTIAL,FIXED 201
!060
180 FOR X=1 TO 5 :: LINPUT #
1:IND$(X):: NEXT X :: CLOSE
#1 !112
190 ! ** display screen ** !
033
200 OPEN #1:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56 !
184
210 CALL CLEAR :: SCR=1 :: C
ALL HCHAR(21,1,129,32):: CAL
L VCHAR(1,15,128,20)!042
220 DISPLAY AT(22,1):"E curs
or up    N>ext screen":"X cu
rsor down  P>rev. screen":"<
enter> pick   Q>uit" !021
230 CALL LANG(5,125)!012
240 ! ** display english ind
ex ** !195
250 CALL VCHAR(1,3,32,20)::
A$=IND$(SCR):: FOR ROW=1 TO
20 !073
260 DISPLAY AT(ROW,3)SIZE(-1
0):SEG$(A$,10*ROW-9,10):: NE
```

(See Page 23)

WORDS

English	German	French	Italian	Spanish
ALL	GANZ	TOUT	TUTTO	TODO
APPLE	APFEL	POMME	MELA	MANZANA
ARM	ARM	BRAS	BRACCIO	BRAZO
AUTUMN	HERBST	AUTOMNE	AUTUNNO	*OTON~O
BACK	*RU#CKEN	DOS	DORSO	ESPALDA
BAD	SCHLECHT	MAUVAIS	CATTIVO	MAL
BEAUTIFUL	*SCHO#N	BEAU	BELLO	HERMOSO
BED	BETT	LIT	LETTO	CAMA
BEER	BIER	*BIE`RE	BIRRA	CERVEZA
BOOK	BUCH	LIVRE	LIBRO	LIBRO
BREAD	BROT	PAIN	PANE	PAN
CAR	WAGEN	VOITURE	AUTOMOBILE	COCHE
CAT	KATZ	CHAT	GATTO	GATO
CHAIR	STUHL	CHAISE	SEDIA	SILLA
CHEESE	*KA#SE	FROMAGE	FORMAGGIO	QUESO
CHILD	KIND	ENFANT	BAMBINO	*NIN~O
CHURCH	KIRCHE	*E'GLISE	CHIESA	IGLESIA
CITY	STADT	VILLE	CITTA	CIUDAD
COAT	MANTEL	MANTEAU	MANTO	ABRIGO
COFFEE	KAFFEE	*CAFE'	CAFFE	*CAFE'
COLD	KALT	FROID	FREDDO	FRIO
COUNTRY	LAND	PAYS	PAESE	PAIS
DAY	TAG	JOUR	GIORNO	DIA
DESSERT	NACHTISCH	DESSERT	FRUTTA	POSTRE
DOG	HUND	CHIEN	CANE	PERRO
DOOR	*TU#R	PORTE	PORTA	PUERTA
DRESS	KLEID	ROBE	ABITO	VESTIDO
EIGHT	ACHT	HUIT	OTTO	OCHO
EXPENSIVE	TEUER	CHER	CARO	CARO
EYE	AUGE	OEIL	OCCHIO	OJO
FAMILY	FAMILIE	FAMILLE	FAMIGLIA	FAMILIA
FIVE	*FU#NF	CINQ	CINQUE	CINCO
FLOWER	BLUME	FLEUR	FIORE	FLOR
FOOT	FUSS	PIED	PIEDE	PIE
FOUR	VIER	QUATRE	QUATTRO	CUATRO
FRUIT	FRUCHT	FRUIT	FRUTTO	FRUTA
GOOD	GUT	BON	BUONO	BUENO
HAND	HAND	MAIN	MANO	MANO
HAT	HUT	CHAPEAU	CAPPELLO	SOMBRERO
HEAD	KOPF	*TE^TE	TESTA	CABEZA
HEART	HERZ	COEUR	CUORE	CORAZON
HOT	HEISS	CHAUD	CALDO	CALIENTE
HOUSE	HAUS	MAISON	CASA	CASA
HUNDRED	HUNDERT	CENT	CENTO	CIEN
KEY	*SCHLU#SSEL	*CLE'	CHIAVE	LLAVE
LEG	BEIN	JAMBE	GAMBA	PIERNA
LEMON	ZITRONE	CITRON	LIMONE	LIMON
MAN	MANN	HOMME	UOMO	HOMBRE
MEAT	FLEISCH	VIANDE	CARNE	CARNE
MILK	MILCH	LAIT	LATTE	LECHE
MONEY	GELD	ARGENT	DANARO	DINERO
MONTH	MONAT	MOIS	MESE	MESE
MOON	MOND	LUNE	LUNA	LUNA
NAME	NAME	NOM	NOME	NOMBRE
NINE	NEUN	NEUF	NOVE	NUEVE
NO	NEIN	NON	NO	NO
NOTHING	NICHTS	RIEN	NIENTE	NADA

RELATIVE FILES —

(Continued from Page 22)

```
XT ROW !149
270 CR=1 :: CALL HCHAR(1,3,6
2)! cursor !219
280 CALL KEY(0,K,S):: IF S=0
 THEN 280 ELSE IF K=13 THEN
380 !143
290 P=POS("EXNPQ",CHR$(K),1)
:: IF P=0 THEN 280 !057
300 ON P GOTO 310,320,340,35
0,360 !191
310 OCR=CR :: CR=CR-1 :: IF
CR=0 THEN CR=1 :: GOTO 280 E
LSE 330 ! cursor up !113
320 OCR=CR :: CR=CR+1 :: IF
CR=21 THEN CR=20 :: GOTO 280
 ! cursor down !020
330 CALL HCHAR(OCR,3,32):: C
ALL HCHAR(CR,3,62):: GOTO 28
0 !132
340 SCR=SCR+1 :: IF SCR=6 TH
EN SCR=1 :: GOTO 250 ELSE 25
0 ! next screen !085
350 SCR=SCR-1 :: IF SCR=0 TH
EN SCR=5 :: GOTO 250 ELSE 25
0 ! previous screen !018
360 CLOSE #1 :: CALL CLEAR :
: RUN "DSK1.INTMENU" !073
370 ! ** display in five lan
guages ** !025
380 RN=CR+20*(SCR-1):: INPUT
 #1,REC RN:R$ !027
390 FOR ROW=2 TO 18 :: DISPL
AY AT(ROW,19):"" :: NEXT ROW
 !013
400 CALL SHOW(2,19,R$,1):: C
ALL SHOW(6,19,R$,12)! englis
h/german !141
410 CALL SHOW(10,19,R$,23)::
 CALL SHOW(14,19,R$,34)! fre
nch/italian !026
420 CALL SHOW(18,19,R$,45)::
 CALL SOUND(100,2000,0):: GO
TO 280 ! spanish !238
1000 !@P+ ! user-def sub !03
6
1010 SUB LANG(R,C)!068
1020 A$="EN14GE11FR05IT03SP0
9" :: FOR X=1 TO 5 :: P=4*X-
3 :: B$=SEG$(A$,P,1):: C$=SE
G$(A$,P+1,1):: K=VAL(SEG$(A$
,P+2,2))!197
1030 CALL SPRITE(#X,ASC(B$),
K,R,C,#X+5,ASC(C$),K,R,C+16)
!116
```

English	German	French	Italian	Spanish
ONE	EIN	UN	UNO	UNO
ORANGE	ORANGE	ORANGE	ARANCIA	NARANJA
PEPPER	PFEFFER	POIVRE	PEPE	PIMIENTA
ROAD	STRASSE	ROUTE	STRADA	CARRETERA
ROOM	ZIMMER	CHAMBRE	CAMERA	HABITACION
SALAD	SALAT	SALADE	INSALATA	ENSALADA
SALT	SALZ	SEL	SALE	SAL
SCHOOL	SCHULE	*E'COLE	SCUOLA	ESCUELA
SEVEN	SIEBEN	SEPT	SETTE	SIETE
SHIRT	HEMD	CHEMISE	CAMISIA	CAMISA
SHOE	SCHUH	SOULIER	SCARPA	ZAPATO
SIX	SECHS	SIX	SEI	SEIS
SPRING	*FRU#HLING	PRINTEMPS	PRIMAVERA	PRIMAVERA
STREET	STRASSE	RUE	VIA	CALLE
SUGAR	ZUCKER	SUCRE	ZUCCHERO	AZUCAR
SUMMER	SOMMER	*E'TE'	ESTATE	VERANO
SUN	SONNE	SOLEIL	SOLE	SOL
TABLE	TISCH	TABLE	TAVOLA	MESA
TEA	TEE	*THE'	TE	*TE'
TEN	ZEHN	DIX	DIECI	DIEZ
THOUSAND	TAUSEND	MILLE	MILLE	MIL
THREE	DREI	TROIS	TRE	TRES
TO BE	SEIN	*E^TRE	ESSERE	SER
TO BUY	KAUFEN	ACHETER	COMPRARE	COMPRAR
TO EAT	ESSEN	MANGER	MANGIARE	COMER
TO GO	GEHEN	ALLER	ANDARE	ANDAR
TO HAVE	HABEN	AVOIR	AVERE	HABER
TO RUN	LAUFEN	COURIR	CORRERE	CORRER
TO SLEEP	SCHLAFEN	DORMIR	DORMIRE	DORMIR
TO TALK	SPRECHEN	PARLER	PARLARE	HABLAR
TO WORK	ARBEITEN	TRAVAILLER	LAVORARE	TRABAJAR
TWO	ZWEI	DEUX	DUE	DOS
UGLY	*HA#SSLICH	LAID	BRUTTO	FEO
VEGETABLE	*GEMU#SE	*LE'GUME	LEGUME	VERDURA
VILLAGE	DORF	VILLAGE	VILLAGGIO	PUEBLO
WATER	WASSER	EAU	ACQUA	AGUA
WEEK	WOCHE	SEMAINE	SETTIMANA	SEMANA
WINDOW	FENSTER	*FENE^TRE	FINESTRA	VENTANA
WINE	WEIN	VIN	VINO	VINO
WINTER	WINTER	HIVER	INVERNO	INVIERNO
WOMAN	FRAU	FEMME	DONNA	MUJER
YEAR	JAHR	*ANNE'E	ANNO	*AN~O
YES	JA	OUI	SI	SI

```
1040 R=R+32 :: NEXT X :: SUB
END !236
1050 SUB SHOW(R,C,R$,P):: W$
=SEG$(R$,P,10):: AW$=RPT$("
",10):: IF P=1 THEN 1110 !21
5
1060 IF SEG$(R$,P-1,1)<>"*"
THEN 1110 ! accent flag? !24
9
1070 FOR X=1 TO 10 :: A$=SEG
$(W$,X,1):: IF POS("'`^#~",A
$,1)=0 THEN 1100 ! find acce
nt !074
1080 W$=SEG$(W$,1,X-1)&SEG$(
W$,X+1,10)! rewrite word wit
hout accent !204
1090 AW$=SEG$(AW$,1,X-2)&A$&
SEG$(AW$,X+1,10)! accent str
ing !050
1100 NEXT X !238
1110 DISPLAY AT(R,C):W$ :: D
ISPLAY AT(R-1,C):AW$ :: SUBE
ND !151
```

Using DISKREVIEW

Disk library cataloging — the easy way

By BOB CARMANY

A short while ago, I decided to undertake the Herculean task of cataloging my disk library of 600-plus disks. The idea of running them through one of the fairware or commercial programs made me realize these programs were either too slow or a bit too cumbersome to use.

At first, I thought about using Disk Utilities 4.12 by the late John Birdwell. It was a simple enough trick to change the print device from "PIO" to "DSKx.filename"; the only problem was that selecting "DISK REPORT" presented me with a D/V 80 file that contained quite a bit of superfluous information. I mean, who really cares where the FDR is or where the program resides on the disk? All I needed to know is which programs were on the disk, and some basic information like disk name and the number of free sectors. Besides, all those control characters were mixed into the file. I didn't want to have to go back into the text file and delete all of that stuff as well. Then there was the fact that the files were listed in a single-file column, which took up entirely too much space. There had to be a better way.

DISKREVIEW opens the PrnDir (print directory) in an Append mode. Besides that, you can easily change the print device when you get ready to print.

After a bit of thinking, I seemed to remember in the deep recesses of my mind something about DISKREVIEW in the

(See Page 27)

NEWSBYTES

SCSIs appear at Fest West

Working TI SCSI cards were sold at Fest West in San Diego in February, though TI EPROMs are still not finished for the products. The SCSI (small computer systems interface) can be used on the Geneve 9640 using the new MODS2.5S, released with the card.

Speakers at the event included Bruce Harrison, assembly language columnist for MICROpendium; Mary Phillips of the Ozark 99ers, who gave a demo of TIPS and related programs and also TI Artist; and Regena (Cheryl Whitelaw), well-known BASIC programmer.

The SouthWest Ninety-Niners are scheduled to host the 1996 Fest West Feb. 17 at the Quality Inn on North Oracle Street in Tuscon, Arizona.

Krych offers software, schematics for AMS

Jim Krych, developer of the Super Asgard Memory System for the TI (AMS) is offering schematics and software for those who want to build the board themselves.

The schematics and software are being offering for $20. Buyers are asked to send three double-sided, single-density disks with a self-addressed, stamped disk mailer. "I am having the rather large schematics turned into regular-sized sheets. This is to allow more copies at a cheaper cost," Krych says.

Krych notes that if there is interest in the TI community, he is willing to have the schematics turned into PCB artwork, double-sided board, for those who wish to etch their own boards. Those who are interested may contact him at ab453@cleveland.Freenet.Edu. Cost of this artwork would depend on the amount of interest, he says.

"If enough people order the PCB artwork, both the schematics and the artwork will be included. As it is now, only the schematics and the software are being offered," he says.

Dealers, seminar presenters named for Lima MUG

Here's an update about the upcoming Lima Multi User Group Conference at Ohio State University in Lima, Ohio:

According to organizer Charles Good, as of early March the following TI dealers have requested tables: Bud Mills Services, Ramcharged Computers, Competition Computers, L.L. Conner Enterprises and Cecure Electronics.

The following individuals are scheduled to give seminars : all of which will be videotaped and made available to the TI community: Bob Carmany (EPROMming with the Hunter Valley EPROMmer), Mike Wright (PC99 stage 3), Bud Mills, Don Walden, Bruce Harrison (demonstrations of his public domain software) and Jim Krych (Super AMS). Mills says he will be showing the TI version of the SCSI card at the conference. The Geneve version of the card has been available for several months.

Lists of seminars and dealers are expected to grow over the next several weeks.

Send your Newsbytes to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680; Internet jkoloen@io.com.

MICRO-REVIEWS

Video Titler Update, Horizon RAMdisk Password, Checktime, Ultimate Delay, Random Number

By CHARLES GOOD

By the time you read this it will almost be time for the next totally free all TI and Geneve Lima Multi User Group Conference. Lima, in northwest Ohio, is only a day's drive from many places in the Midwest and eastern part of the United States and Canada. I hope to meet many of you personally at this event Friday evening and Saturday, April 28 and 29. If you need more information about the MUG Conference please contact me. My address is at the end of this column.

Here are more public domain offerings from Bruce Harrison. We certainly are fortunate to have him working for us for free. Each software product described below comes on disk with demonstration XB programs, source code, very well written documentation, and something you almost never get with public domain software...free technical help! Each documentation file ends with the statement, "Should you need assistance, contact the author directly," followed by Bruce's address and home voice phone. Even if you don't need this help, phone Bruce some evening and tell him how much you enjoy his software and the neat uses you have found for his public domain products. Bruce really likes such personal expressions of appreciation and encouragement.

VIDEO TITLER UPDATE by Bruce Harrison

If you own a DOS computer you can purchase from the Damark liquidation catalog (item B-4600-379944 for $160— the cheapest price I have ever seen for this sort of thing) a device that lets you display your computer video on a TV or send the computer video to a VCR for recording ıto videotape. *Or* you can do the same thing by running your 99/4A's monitor cable to the "video in" of a VCR. You can then observe your computer's video on a TV attached to the VCR while you make a videotape recording of the 99/4A's video. For a minimum of $50 you can purchase software that will let you create custom title screens for your homemade videotapes using the above described DOS computer board, *or* you can do the same thing with your 99/4A using Bruce Harrison's recently updated free Video Titler public domain software.

I've reviewed Video Titler before in this column, but Bruce has updated the product several times since. What I have in hand is dated Dec. 28, 1994. Bruce says in his letter to me dated Dec. 17, "Here's the final update (at least for this year) of the Video Titler." He then sent me two more updates, each with added features, before the year ended. What you do is load into memory two TI Artist pictures or two pictures created with Bruce's Drawing Program (also public domain and on the same disk as Video Titler). You can then alternate back and forth between these two images. You feed this display to a VCR (through the "video in" jack on the VCR) to record onto video tape. If you want to record more than two computer pictures onto videotape you press "pause" on the VCR, load in another picture in place of one of the two pictures in the computer's memory, release the VCR's "pause" and continue to record.

The neat thing about Video Titler is the wide variety of ways Bruce provides to alternate (wipe) between the two pictures in computer memory. The updated software has a whole bunch of wipes not found in the original. To wipe, just press a specific key on the keyboard, as described in the documentation. Different buttons produce different kinds of wipes. The following wipes are available: fast from top to bottom, fast from bottom to top, slow from top to bottom, slow from bottom to top, slow from left to right, slow from right to left, slow horizontal from center to edges, slow horizontal from edges to center, slow vertical from middle to top and bottom, slow vertical from top and bottom to middle, spiral inward from edges to center, spiral outward from center to edges, venetian blind wipe top to bottom, venetian blind from bottom to top, fast or slow speed random replacement of 192 (fast) or 384 (slow) separate screen areas till other image is fully displayed, upper left corner to lower right, upper right corner to lower left, lower left corner to upper right, lower right corner to upper left, instant change to green (or black, or white) followed by a pause then the other image.

Each of these will go from whichever picture is currently displayed to the other picture in memory. You can toggle back and forth between frames at any time with a key press. The visual effects are dazzling! I've talked to two users who think Video Titler is superior to anything available for DOS (i.e., IBM-compatible) computers for making video tape title screens. Send me $1 and I will send you the most recent update of Video Titler plus Bruce's Drawing Program on one DSSD disk.

HORIZON RAMDISK PASSWORD by Bruce Harrison

Both Bruce Harrison and a member of my local user group have the same problem. They each have several indoor cats that like to jump up onto the computer table and walk on the keyboard. To deal with this difficult situation, Bruce Harrison created a password program that won't let you access your computer unless you type in the correct password. First set up Password on your Horizon RAMdisk. Then every time you turn on the computer, press the "any key" to get past the color bar screen, and you are immediately prompted for a password. As you type this password your typing is not echoed onto

(See Page 26)

MICRO REVIEWS—

(Continued from Page 25)

the screen so that nosy cats and kids can't look over your shoulder and learn your secret. If you decide to step away from the computer for a while and leave it turned on, you can reactivate the password feature. This prevents the cat from messing up your work while you are away.

Password has only been tested on a 99/4A system with Horizon RAMdisks and may not work with other types of media. Send me $1 and I'll send it to you on a SSSD disk.

CHECKTIME
by Bruce Harrison

This is fun for software users, useful for XB programmers, and seems to bring into question the true usefulness of some well-known XB programming techniques. Checktime accurately measures the speed of XB programming code.

To use this utility first CALL LOAD it into low memory and then OLD an extended basic program into memory. Insert a CALL LINK into the XB program at the point where you want to start timing and another CALL LINK into the XB program at the point where you want timing to end. Then RUN the XB program, which will execute normally. After the XB program finishes running or after you manually stop the program with a BREAK you can display on screen the duration of the timed operation rounded to the nearest second, approximately accurate to 1/60 second. Then try modifying your XB program so that timed operation runs faster, and check out your modification's execution speed.

Since the early days of our computer we have read articles in *99er Magazine*, MICROpendium, and user group newsletters about how to speed up XB programs. Two techniques that are supposed to accomplish this are using short variable names instead of long names, and using multiline statements instead of single line statements. Bruce includes a couple of demo programs that show the effect of these two techniques using Checktime. Running a for-next loop 2000 times into a variable 14 characters long ("ANYOLDVARIABLE") takes 2 minutes and 5 seconds. Renaming the variable to only one character ("A") shortens run time by only 6 seconds. Repeating a for-next loop coded with four single line basic statements 5000 times takes 5 minutes and 2 seconds. Reprogramming this loop into a single multiline statement shaves only 1 second off this running time! Bruce also includes some demos that show significant improvements in execution time by using more efficient XB code. One is a simple modification of a bubble sort that improves sorting speed from 1 minute 14 seconds to only 18 seconds.

Send me $1 and I will send you Checktime on a SSSD disk.

Press the FCTN X key 10 times while on any menu screen. You will hear a beep. If your monitor has sound, an >< will appear at the center top of the screen. Any diskette initialized as this point will be proprietary protected.

ULTIMATE DELAY
by Bruce Harrison

Ultimate Delay is for use by XB programmers. This assembly CALL LINK-able routine will provide accurately timed delays in XB programs irrespective of the speed of your computer. This means that the delay is the same length of time for a program running on a regular 99/4A, a bus modified 99/4A (32K internally on the 16K bus), and a Geneve.

You can use this routine in two ways:

1- You can insert an unrestricted delay of any time length (timed in 10ths of a second) anywhere in the program. During the delay the keyboard is ignored except for FCTN/= (Quit).

2- You can insert a delay that will time out either after a period of time or when a key is pressed. You have the option here of returning the ASCII code of the first key pressed to a variable for later use by your program. These sorts of things can be done with regular extended basic using for-next loops and CALL KEY, but the delay lengths will differ when the program is run on different types of machines. Coding for delays is much simpler with Ultimate Delay, and delay time is the same when run on different types of computers.

To experiment you can manually CALL LOAD Ultimate Delay into low memory and then play around with inserting the needed CALL LINKS into various places in your Extended BASIC program and observing the results. Once you get your Extended BASIC program working to your satisfaction you can attach Ultimate Delay to your program. You can then OLD and RUN your program without the bother of separately CALL LOADing Ultimate Delay. Software tools and directions for attaching Ultimate Delay to your Extended BASIC programs are included on the SSSD Ultimate Delay disk. Send me $1 and I will send you this disk.

RANDOM NUMBER
by Bruce Harrison

This comes on a SSSD disk. It's yours for $1. It is a set of assembly utilities for the XB programmer that generate random numbers much faster than XB's RND function. You CALL LOAD and CALL LINK to the routines, or you embed them into your finished XB program. Bruce provides several direct comparison demo programs that show how his routines assign 300 random numbers to arrays in about 1-3 seconds, while using RND to do the same thing takes 25 seconds.

One set of routines will randomly assign up to 500 integers between -32768 and 32767 to an array variable such as DIM A(500). It takes about 1 second to do

(See Page 27)

MICRO-REVIEWS—

(Continued from Page 26)

this. You can specify the maximum and minimum size of the randomly generated numbers within the above range, and you can also specify with or without replacement. Without replacement means that a randomly generated integer number will not appear more than once in the array. With replacement means that a randomly generated number will be put back into the pool of possible numbers and may appear several times, randomly of course, in the array.

Bruce provides a useful demo of random integer number generation in the form of a card shuffling XB program. How may times have you seen a TI card game say on screen "Please wait, I'm shuffling the deck," and then make you wait a while while this is done? Bruce's demo program does the shuffling really fast! Random numbers 1-52 are generated without replacement and then displayed in random order on the screen.

Another set of routines generates floating point numbers of any size. An unlimited number of floating point random numbers can be generated with these routines, limited only by the size of the array where the numbers are to be assigned. Large arrays take up lots of memory. It takes about 3 seconds to grind out and assign 300 of these floating point random numbers. You can specify the maximum and minimum size of the numbers to be generated and you can specify whether to truncate the number at the decimal point thus creating big number integers.

ACCESS

Bruce Harrison, 5705 40th Place, Hyattsville MD 20781. Phone 301-277-3467.

Charles Good, P.O. Box 647, Venedocia OH 45894. Phone 419-667-3131. Internet email cgood@lima.ohio-state.edu

DISK CATALOGING—

(Continued from Page 24)

Funnelweb package. Dutifully, I got out the docs (when in doubt ...) and found exactly what I was looking for. It seems that DISKREVIEW opens the PrnDir (print directory) in an Append mode. Besides that, you can easily change the print device when you get ready to print. Hmmm! This had some possibilities!

I chucked a disk in the drive, selected PrnDir and had at it! The device name was changed to DSK9.PART/1 (my RAMdisk). Printing to RAMdisk was almost instantaneous. Even better, as long as I kept reading the same drive number, all I had to do is slip another disk in the drive and hit FCTN-8 (Redo). The next disk was cataloged automatically.

Ever so often, I backed out with FCTN-9 and checked the size of the file on RAMdisk. As long as I stayed in DISKREVIEW, the print device stayed the same (DSK9.PART/1). When the file size approached 135 sectors, I simply incremented the device name (i.e., DSK9.PART/2) and kept on cataloging. It was a simple matter to go through my entire disk library with a minimum of effort.

The output is a centered two-column D/V 80 file that can be manipulated with the Funnelweb editor or any other conventional word processor. It can be printed, searched for file or disk names or just about anything else you can do with a text file.

It took me slightly less than 30 minutes to go through the 99 disks in the utility section of my library. Most of that time was spent shoving disks in and out of the disk drive. I would recommend DISKREVIEW as a superior disk cataloger, even though that is not what it was designed to do.

Database managers

Databases come in many shapes and sizes

(This article was originally published in the November 1994 issue of Wordplay, *the newsletter of the Portland Users of Ninety-Nines.)*

By TERRY PRIEST

Databases are part of our everyday lives. We use a few without really realizing it. One of the best examples is the phone book. It has all the parts — a collection of fields of data (last name, first name, address and phone number) which is the record of each subscriber and is bound together (a file). Information is sorted in two ways — alphabetical by users in the white pages and alphabetical categories in the yellow pages.

Another familiar one to TIers is trusty DM1000. What's that you say? — it's a disk manager. It is a disk manager, too, but all of the files, housekeeping and disk functions are dependent on the database function. The disk catalog is a fine example of a database — the fields are the columns (file name, size, type, etc.) and the rows are the records, since each row describes one file on the diskette. The whole thing is the file (database). You can even see the boilerplate. That is the graphic lines, column names and other information that is not on the disk catalog but is placed on the screen by the DM1000 program. The various actions the program can perform are the equivalent of database reports. A tabular report is a good way to visualize a database. The fields are the

(See Page 28)

DATABASES—

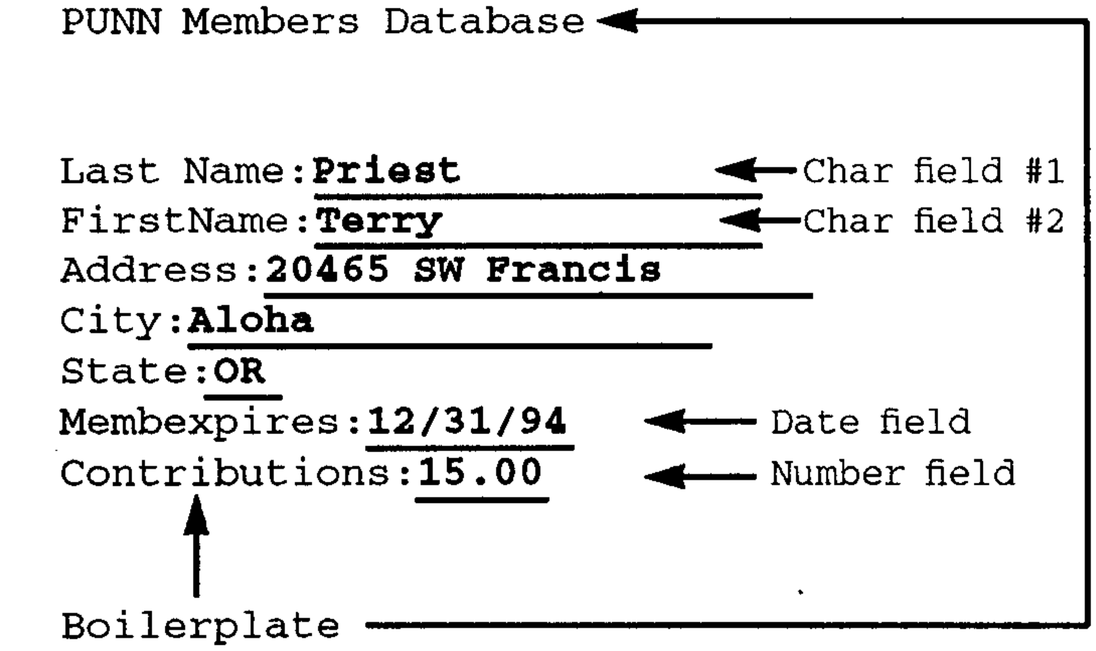
(Continued from Page 27)

columns and the records are the rows. The column names and other text is the boilerplate.

Database programs cannot contain all the data in memory and have a reasonable amount of records. What they do is maintain indexes of data that resides on the diskette. It is first located and then reassembled for use by the program or display. The only exception is Personal Record Keeping Module. It puts all data in memory and saves program and data as program format file. Unfortunately, this severely limits the number of records it can contain, especially since it does not use the 32K memory expansion..

ELEMENTS OF A DATABASE

A database consists of fields, records and files. See the illustration below:Char Field #1



Each of the pages above is a record. The collection of pages is the file or database.

Another way is the tabular report method:

LastName	FirstName	Address	City	State
Priest	Terry	20465 SW Francis	Aloha	OR←*record #1*
Zeller	John	1415 Anywhere	Portland	MA←*record #2*

TYPES OF DATABASES

There are two types of databases for the TI, relational and flat file. In a flat file format, *all* information to be stored, retrieved and manipulated must be on each record *and* in the same file. Fields that may contain data in only a few cases are on every record, occupy blank space in memory and slow processing by the computer. You cannot compare or combine data from different databases even if they have some information in common unless you manually print it out and do it yourself. In addition, you may be limited by the number of fields allowed on each record.

In a relational database you can combine, extract, compare and what have you from different sets of data. This can allow you to have many more fields if you establish a link between them. Record sizes can be smaller if data are organized such that each database's records contain fields that would be filled on most records. Data that would normally only relate to a few of the records in the main file is kept separately and linked to the origin. keeping some things separate enables the computer to work faster and is much more efficient in the use of memory and disk space. Relational databases also allow one to record a script to store the instructions or procedures for future use.

THREE DATABASE PROGRAMS FOR TIS

1) *Personal Record Keeping (PRK).* Module based, does not need P-box, can use cassette or disk. Does not recognize 32K memory expansion.

2) *PR Base.* Requires disk drive (two recommended), 32K, Extended BASIC or Editor/Assembler 5 loader. Uses the entire disk for data. Your program must be on separate diskette named PRBASE. The data disk cannot be read by any disk manager or other program.

3) *TI Base.* Requires disk drive, 32K. Loads from XB, EA, Mini-Memory, TI-Writer Option 3. Data stored on diskettes in normal way. a relational type database, you can have five separate databases communicating at once.

Feature:	PRK	PRBase	TI Base
Type		Flatfile	Flatfile
Relational			
# of fields/record	15	17	17
Max field length chars	15	255 total of all fields	255 fields
Number fields	Yes	No	Yes
Date fields	No	No	Yes
Math functions	Yes	No	Yes
Max # records	Varies, <100	350 SSSD, 710 DSSD 1,430 DSDD	16,000+
Storage	Cassette or disk	Disk	Disk
Output devices	Printer	Printer or disk file	Printer or disk file
Stored report formats	Two, fixed	User defined 5 5 report/2 labels	User defined Unlimited
Flexibility	Low	Medium	High

ORGANIZING DATA

1) Decide what data are needed and break into fields.

2) Name the fields on paper and estimate length if characters or numbers.

3) Review fields to see if greater or lesser breakdown is needed.

4) If you have too many fields, combine some or use a different program.

5) If a substantial number of records will have blank fields, use relational database. If so, use steps 1-5 to design the supporting database.

6) Enter the fields in an order that is logical to you.

7) If using PRK or PRBase be sure to leave an extra field for future expansion, as the structure cannot be changed or record transferred between databases. Not necessary with TI-Base, since records can be copied to another database, fields added or subtracted and rearranged while original data are intact.

(See Page 29)

DATABASES—

(Continued from Page 28)

OTHER CONSIDERATIONS

1) Databases have to be sorted on at least one field in order to operate. Re-sorting takes time — sort on the field most likely to be used first.

2) Use your database to select information for a TI-Writer mail merge. The formatter can write letters easier than the database.

3) If you will need a count of all or certain records, use TI-Base.

4) If you must select records based on a logical decision (if-then-else) or case, use TI-Base.

5) When setting up the fields in the database, enter them in an order that is easy to key in.

6) If you already have data on file use TI-Base. It can read text files (D/V 80) and place data in the file without you having to type it in. You must get the data in row and column format first. If the fields are longer than 80 columns, split it, read into two databases and then combine into one.

7) Dates: If you need to manipulate or calculate dates (days before, after, elapsed, etc.), use TI-Base.

8) Be aware that numbers are not rounded, they are truncated to fit the field width and decimals allowed them. Errors can accumulate during multiplication or division.

9) Searches: PRK and PRBase can do searches with only part of the desired information, but what you search for must match explicitly; e.g., PRI would find PRICE and PRIEST, while PR E would not find PRICE or PRIEST. TI-Base can "scan" the field for EST and would find ESTABLISHMENT and PRIEST in addition to being able to do the first search example.

EXAMPLES OF DATABASES

In this case, we have a roster for several different organizations that have most or many members in common. Some of the members have cellular phones in addition to a business and home number. The Computer Club has 55 members; Camera Club has 48 members; Ham Radio Club has 76 members. Of these 20 Computer Club members are also members of the Ham Radio Club and 10 are in Camera and Ham. Similar crossovers exist for the other clubs.

Herbert Schlesinger dies

Herbert I. Schlesinger of Farmington Hills, Michigan, died recently while undergoing a second heart bypass operation.

He was a founding member of the Great Lakes Computer Group and a contributer to its newsletter.

He is survived by his wife, Evelyn, and his daughters. Memorial contributions can be made to the Michigan Heart Association.

USER NOTES

Labeler3 update

The following item comes from Jim Uzzell of DDI Software. He writes:

The following is an upgrade to LABELER3 (January 1995, MICROpendium). This upgrade will allow you to have up to 127 files and 3 subdirectory names, if present. Type these changes in and save in merge format. Load LABELER3 then merge this file.

```
110 F$="|  " :: G$="|" :: C$=
"########## " :: DIM B$(132)
,Z$(32),B(32,32)
150 XT=104 :: DISPLAY AT(23,
1)BEEP :"WHICH DRIVE i.e DSK
1.":
240 CLOSE #1 :: IF I>104 THE
N XT=132
260 PRINT #1:CHR$(15);CHR$(2
7);CHR$(65);CHR$(5);E$
280 PRINT #1:CHR$(27);CHR$(8
3);CHR$(0);CHR$(27);CHR$(65)
;CHR$(5);CHR$(27);CHR$(88);C
HR$(0);CHR$(47);
305 IF XT>104 AND X=17 THEN
PRINT #1:F$;TAB(47);G$ :: PR
INT #1:SEG$(H$,1,2);TAB(46);
SEG$(H$,2,1)&G$
310 IF X=1 OR X=5 OR X=9 OR
X=13 OR X=17 OR X=21 OR X=25
 OR X=29 OR X=33 OR X=37 OR
X=41 OR X=45 OR X=49 OR X=53
 OR X=57 THEN PRINT #1:F$; :
: GOTO 330
320 IF X=61 OR X=65 OR X=69
OR X=73 OR X=77 OR X=81 OR X
=85 OR X=89 OR X=93 OR X=97
THEN PRINT #1:F$;
325 IF X=101 OR X=105 OR X=1
09 OR X=113 OR X=117 OR X=12
1 OR X=125 OR X=129 THEN PRI
NT #1:F$;
340 IF X=4 OR X=8 OR X=12 OR
 X=16 OR X=20 OR X=24 OR X=2
8 OR X=32 OR X=36 OR X=40 OR
 X=44 OR X=48 OR X=52 OR X=5
6 OR X=60 THEN PRINT #1:G$ :
: GOTO 360
350 IF X=64 OR X=68 OR X=72
OR X=76 OR X=80 OR X=84 OR X
=88 OR X=92 OR X=96 OR X=100
 THEN PRINT #1:G$
355 IF X=104 OR X=108 OR X=1
12 OR X=116 OR X=120 OR X=12
4 OR X=128 OR X=132 THEN PRI
NT #1:G$
```

Converting text files to XBASIC

The following program and text was posted on the Internet. It was written by Glenn W. Bernasek.

Just something to "spark" some interest. The following routine will read a TI-99/4A D/V80 text file to either the screen or a PIO printer. You can either type in the routine line-by-line, or you can have Paolo Bagnaresi's BASIC Builder create a running routine for you.

TEXTREADER

```
1 CALL CLEAR :: FOR I=97 TO
122 :: DISPLAY AT(12,6)SIZE(
19):"Building Lowercase" ::
READ LC$ :: CALL CHAR(I,LC$)
:: NEXT I :: CALL CLEAR :: !
(c) 1993 G.W. BERNASEK
2 DATA 00000038043C643C,0040
404078444478,0000001C2020201
C,000404043C44443C,000000384
478403C,0018242070202020,000
00038443C0438,00404040784444
44
3 DATA 0010001010101010,0004
000404042418,002020242830282
4,0010101010101010,000000685
4544444,0000005864444444,000
0003844444438,00000078447840
40
4 DATA 0000003C443C0404,0000
005864404040,0000003C4038047
8,0010103810101000,000000444
4444438,0000004444282810,000
0004444545428,00000044281028
44
5 DATA 00000044443C0438,0000
003C0408103C
100 CALL CLEAR :: PRINT "PUT
 DV80 DISK IN DRIVE.":"ENTER
 [DSKn.FILENAME]" :: INPUT "
      ":F$ :: PRINT :: INPU
T "TO:<0>-SCREEN or <1>-PRIN
TER":P :: IF P=0 THEN PRINT
"HOLD DOWN ANY KEY TO PRINT.
": : :
105 OPEN #2:F$,INPUT :: IF P
=1 THEN 110
107 CALL KEY(0,K,S):: IF S=0
 THEN 107
110 IF EOF(2)THEN END ELSE O
PEN #1:"PIO" :: LINPUT #2:T$
 :: PRINT #P:T$ :: CLOSE #1
:: IF P=1 THEN 110 ELSE 107
:: !"TEXTREAD2" (C)1989 G.W.
BERNASEK
```

If enough of us have a copy of Paolo Bagnaresi's BASIC Builder, we will be able to upload D/V80 versions of TI BASIC and XBASIC programs to this newsgroup. The text file can then be downloaded and converted to a running program quick and easy with BASIC Builder.

How do you create a text file from a BASIC or XBASIC program? Just type in LIST "DSKn.FILENAME" and press Enter.

All you have to do to convert the text file to a program file is to load BASIC Builder and enter DSKn.FILENAME at the prompt.

If you have access to a user group library, you may be able to get a copy of BASIC Builder.

Users of RXB V1000 can automatically convert text files into program files, as can Geneve owners using Myarc BASIC. —Ed.

How the SCSI card operates

The following article is condensed from an article in the March 1995 newsletter of the SouthWest Ninety-Niners by Mike Doane.

As one of the first with an installed and working SCSI (small computer systems interface) which I got at Fest West, I am providing a report on how it operates.

The SCSI card and hard drive are indeed working. The TI EPROM is *not* finished. Western Horizon Technologies and Bud Mills Services hope to be sending it out in the mail by the middle of March. Geneve users *can* use the card using the new MDOS2.5S, released with the card.

I brought my SCSI home from the Fest and installed it. I loaded up MDOS2.5S. The first thing that needs to be done is to determine the physical setting of your SCSI card. It needs to be set at address

(See Page 31)

USER NOTES

(Continued from Page 30)

>1200. Set the address using the DIP switches. Set the "physical" address of your drive. I set mine as Drive 0. This is done by either DIP switches or small jumpers on your drive. If you have only one drive then you *must* put a "termination" resistor pack on your drive. You need to install the drive on the *middle* connector and install a termination pack on the *end* connector of the cable. The packs are available from most *computer* shops.

You need to run the SCSIINIT program *first*. This initializes the SCSI bus and drive. You then need to run the SCSI editor/formatter. These programs are included in the three disks of utilities and useful programs included with the SCSI. Select the Format option from the menu. My drive formatted best at 1024 sectors. The password is "FORMAT" — capital letters, *no* quotation marks. If you read the instructions it tells you that, but who actually ever reads them!

Run the MYS program next. I am able use interleave No. 1. This is the fastest interleave. I suggest you start with 1 and if you experience any read/write errors, then increase the interleave. I suggest you skip the No. 2 setting and go straight to the No. 3 setting. The No. 3 setting is the "standard" setting and 90 percent of all SCSI drives will run at that interleave. If you have an *old* drive, you might end up all the way at the No. 7 setting. There is nothing wrong with this. Your drive may not transfer data as fast as it could on the No. 1 setting, but it will still be as fast as the older MFM-type drives you can use with the Myarc Hard and Floppy Disk Controller.

You are now ready to save programs, using Clint Pulley's Directory Manager, which comes on the disks of utilities.

There is one *major* problem with MDOS2.50S. It will not delete files from the SCSI drive. This is being resolved at this time. I suggest you change the name of any file that you no longer need by adding a "9" at the end of it. Instead of having a file named "DSKU1" you would rename it as)SKU9"; any following files, you can name in ascending order, such as, "DSKU2" would become "DSKU10" and so on. When the problem is fixed, it becomes a simple matter of deleting those files with the use of the Directory Manager program. *Do not* attempt to "Move" those discard files into a separate "Discard" directory. I speak from experience on this one!

My hat is off to Don O'Neil and Bud Mills for providing us with an excellent hard drive system.

CLASSIFIEDS

FOR SALE

BUY-SELL-TRADE

Hardware — Software. TI Buyers Guide $2. Dual 1/2 HT Kit complete with cab les, screws, drill template & instructions $59. PE Box complete $149. Corcomp DD Cont $180. Star NX-1001 Printer $175. 2400 Baud Modem $79. TI99/4A Console $45. Rare — Myarc Hard & Floppy Controller with 10 Meg Hard Disk & Case $350. 800-471-1600 (Nationwide & Canada ORDERS ONLY). 414-672-1600 Local/tech Support. Huge Genuine TI Inventory Since 1982. Competition Computer, 2219 S. Muskego, Milwaukee, WI 53215. Fax 414-672-8977. Open Daily 9-5, Sat 10-3. Bankcards, Discover, Checks & UPS/COD.

TI99/4A

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

RXB IS NOW FREEWARE

Features include: 21 lines of input per line number in edit mode. SIZE from program mode. (Assembly support space also reported) RENAME or PROTECTS disk files. (Up to 14 drives in one command) EXECUTE direct CPU address (5 times faster than LINK) CRU direct control of Input/Output chip from CALL IO CALL CATalog from program mode. EDITOR/ASSEMBLER module built into RXB. Will RUN EA5, EA3, and XB or BASIC. (Throw away your loaders) PEEKs or POKEs in VDP, GROM, and CPU. MOVE any type or size of memory to another. Or use strings. BATCH FILE SERVER is built into RXB from DV80 files. Rich Gilbertson, 1901 H St., Vancouver, WA 98663.

T.O.D. FANS!

All new Tunnels of Doom Adventure for your TI! A major revision with new graphics, spells, ploys, and surprises! It's called **Halls of Lost...MORIA!** and takes place in the dawn of the Fourth Age. Listen to these rave reviews: Justin V (age 10) Neat! It looks more like a dungeon, the fireballs are awsome! Amy V. (age 6) I think it's pretty! Diane V. (age ??) Hmm... if you cut wood, like that Dwarf chops Orcs, we wouldn't be worried about running short this winter.... Send $4.95 ($5.95 outside USA) for disk and booklet to: Michael Veprauskas, 32 Vaillancourt Dr., New Ipswich, NH 03071. v12n5

PROGRAM INNOVATORS SOFTWARE SALE

TOUCHDOWN: NFL Football Predictor $10.00 WALLSTREET: Investment Package $30.00 USVBA Power Volleyball (ML-Game) $10.00 Klingon Invaders, Destroy Klaatu, Desert Rat (Machine Language Games) $10.00 Cutthroat Cribbage, TItris, Cockroaches, Snomobile, Martian Missiles, $10.00 Multiple Orders 10% discount 4122 Glenway, Wauwatosa WI 53222-1116.

Classified ads are 10 cents per word

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

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

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$35**

❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50**

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

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

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

Address Changes

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

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

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

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

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

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

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 2.0** (req. DSSD or larger (for floppy & hard drive systems) $4.00

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS

A T EXPIRES 1/96
CHARLES GOOD
P.O. BOX 647
VENEDOCIA OH 45894