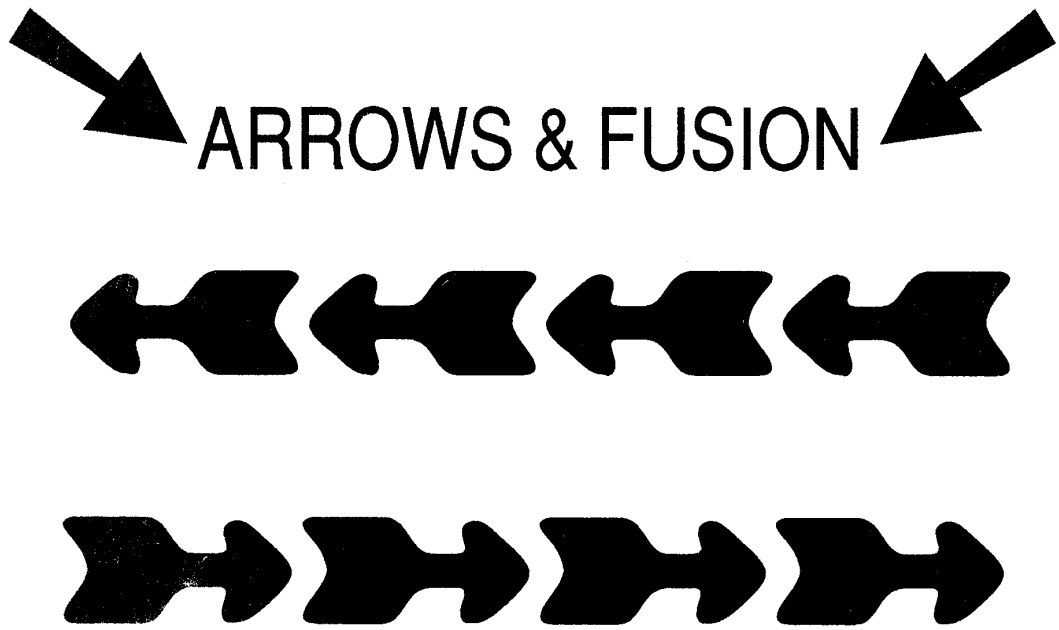
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 11 Number 9 October 1994 $3.50



See page 6

Preview of the TI International World's Faire • X-raying the Geneve • A color tutor • All about hard drives • And much more!

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.

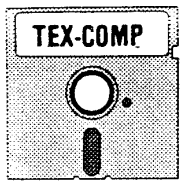
Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone: (512) 255-1512
CompuServe: 75156,3270
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@io.com
John Koloen.....Publisher
Laura Burns.....Editor

*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 GRAFHS (M)
#52 ANIMATION 99' (2)
#69 PLAYER PIANO/KEYBOARD ANALYSIS
#93 RBGB GIRLIE CALENDAR (+)(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 ARTCON GRAPHIC CONVERSION
#79 DISK MANAGER 1000
#80 BIRDWELL DISK UTILITY (2)
#85 AUTOBOOT UTILITY
#86 COLUMN TEXT TTI (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 MIRBLER/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 YOURSELF 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 LAFD 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 WH. 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
#42 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)
#50 FREDDY (HIT from GERMANY)
#51 THE MINE (from GERMANY)
#52 ASTROBLITZ & MAZOG
#54 MAJOR TOM & SPACE STATION FHETA
#55 PERFECT PUSH (HIT)
#68 CHESS (SARGON)
#70 TI RUNNER II (HIT)
#72 CEBERUS (HIT SPACE GAME)
#73 CRYIO (GRAM)
#82 CROSSWORD (PUZZLES)
#84 GALACTIC BATTLE & SPY ADVEN.
#88 AUSSIE GAME COLLECTION VOL 1
#91 THE MAZE OF GROG (MOODSTOCK 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 (ID)
#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.

Authorized Dealer

TEX COMP

America's Number One TI computer retailer

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

ORDER BY PHONE

(818) 366-6631 24 HOURS A DAY

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

Looking forward to Chicago

The Chicago TI fair is nearly here. This event has been a highlight of the year for me since 1985. This and other fairs are a great opportunity to meet the people who make the TI work. This includes programmers, vendors and users. It's difficult to visit one of these events and not have a good time. Not only are you immersed in things TI, but there are new friends to be made, old friendships to be revived and plenty of things to see and do. Just poring over the hundreds of items displayed by vendors can absorb hours of time. Couple this with seminars, giveaways and good conversation, this and similar events are a motherlode for anyone interested in the TI. They are absolutely indispensible for anyone looking for a hard-to-find piece of software or hardware. If you've got a shopping list, these fairs are your mall.

The downside to this and other fairs, of course, is that they are diminishing in size and number. Several hundred users is now considered a good draw when five years ago the numbers topped a thousand. How long they continue depends entirely on users like you and vendors. I suspect the fairs will last as long as MICROpendium. Once people get tired of the TI, both of us will disappear. But until then, I'm going to enjoy the opportunity to have a good time with other TIers. I hope you do, too.

OASIS STILL PRIORITIZING

I don't know how long it takes to get one's act together, but Oasis Pensive Abucators certainly has to be taking it to new heights. Now we hear that OPA has moved into a three-story building, with two floors dedicated to manufacture and one as a sales floor. Whoa! That's a lot of space, much of which, no doubt, will be dedicated to something other than the TI. At this point the company is promising to rebuild it's credibility in the TI community by putting the manufacture of its TIM product on a fast track. Frankly, people have been waiting for this for years. I don't think it's going to generate a lot of interest, certainly not among those who have already been burned by the company's slipshod approach to customer service.

I hope this means that OPA is turning over a new leaf. I just find it hard to believe.

—JK

FEEDBACK

Getting MDOS going

Recently I received MDOS V.2.00 from you. First I had problems getting it running correctly. MDOS booted from hard disk, but then failed to execute the AUTOEXEC file present in the ROOT directory of the hard disk, and showed the system prompt A>. Changing the prompt to E> and starting AUTOEXEC by hand resulted in the execution of AUTOEXEC (what else?).

Now I realized that I had to change the defaults in MDOS telling the system where to look for AUTOEXEC. Using John Birdwell's Disk Utilities 4.2 I found the string DSK1. several times (about four or five). After changing them all into HDS1. everything worked fine.

Alfred Slovak
Vienna, Austria

Gap filled in by alert reader

This is to extend thanks to Tony Knerr, of Downingtown, Pennsylvania. Tony's letter, as published in the August issue, fills in a gap that I didn't realize was there. Since I have only a "simple" P-Gram in my machine, there was no need to set up for paging the contents of the card.

My apologies go to any of my readers who have P-Gram Plus and have run into this particular problem. The only change I would make to Tony's suggestion is that I'd include the R in those two lines of source code that he recommended adding, so they'd read:

```
LI    R1,>9804 (or >9808 or >980C)
MOV   R1,@>80FA
```

That would keep the added lines consistent with the rest of my source code, which always uses the R for registers, just to keep *me* from getting confused. I don't know whether the assembler would tolerate having part of the source file use the R and part not use it, as I've always used the R and, of course, included the R option when assembling.

It's really great to see that there are readers like Tony out there, paying close attention to our work. His comments are most welcome any time.

Bruce Harrison
Hyattsville, Maryland

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

Chicago-Milwaukee fair scheduled for Nov. 12

Several speakers have been confirmed for the TI International World's Faire in Gurnee, Illinois, Nov. 12, sponsored by the Chicago and Milwaukee users groups, according to Don Walden of the Chicago Users Group.

Lee Bendick of Newark Ohio, collector of TI hardware, who owns a TI99/2 and TI99/8, will speak on the CC40

Tim Tesch will do a presentation on his program CYA. He is releasing extra enhancements for the program at the fair and will be selling CYA at a special fair price of $13. He has added some test programs for joysticks, mouse port and to check HFDC for the version of the EPROM and whether it has a 32K or 8K, DIP switch locations of the floppies and the HFDC's CRU address

Jim Schroeder and Walden will make a presentation on PFM Plus and other topics.

David Nieters and Brad Snyder are scheduled to give presentations on products from Western Horizon Technology, including its keyboard interfaces.

Vendors confirmed as of press time include L.L. Conner Enterprises, Competition Computer, Cecure Electronics, Western Horizon Technology, S&T Software, RAMcharged, Bud Mills Services and RDB Enterprises

User groups to be represented include the Mid-South Users Group, Hoosier Users Group, Sheboygan 99ers, TI User Group of Will County and the Lima User Group.

Dan Eicher, Jeff White and Mike Maksimik, who are developing SCSI DSR for MDOS will attend, Walden says. Brad Snyder, also scheduled to attend, has replaced Maksimik as developer of the TI EPROM DSR. The SCSI (small computer systems interface) has been under development by Bud Mills Services for some time.

International visitors are expected from Germany and The Netherlands, Walden says, and some will probably attend from Canada as well.

Special room rates are available at the Faire site, the Holiday Inn in Gurnee. Persons desiring these rates must mention the Faire and must call the Holiday Inn in Gurnee directly at (708) 336-6300. For coupons and discount tickets for area attractions call the Lake County Visitors Center at 1-800-525-3669.

Extended BASIC

Arrows and fusion

By LUCIE DORAIS

Michel Montmigny from Sherbrooke, Quebec, sent me a whole collection of programs he wrote, many in Pascal, but some in Extended BASIC and even more in TI BASIC. Here are two of his XB programs, short but very well coded.

The first one, ARROWS, is a simple game, but very hard to win. You must shoot an arrow. When launched, it must cross the whole screen to touch the target at the right. Easy? Not when six long moving shapes continually cross your field! Even more so if they move in opposite directions.

The game uses the Speech Synthesizer, something I have never discussed in my tutorials because I don't use mine. The only changes I have done to Michel's program is to translate the text from French to English, modify the CHARDEF for the arrow to make it easier to type, adding a yellow target at right and some sound to hear when the arrow hits it if you don't have a Speech Synthesizer. I also added one more arrow to shoot for game (you now have six).

Line 120 is a new trick: the border character, ASCII=31, is CALL VCHARed at the left and the right of the screen to create a nice border, which will be colored blue in line 130. The character sets with letters and digits are then made white on black (space will therefore be black, too) and more sets are colored, for the yellow arrows and the grey target. The introduction screen is then displayed. When you answer the question, more CHARDEF is done (char. 128 will be the long moving shapes, 136 the target, 132 the arrows, both the arrow sprite and the small "remaining ones"). Then CALL sub CLS to erase the screen with the space character drawn vertically from the upper left (24x28=672, line 350), reset the High-Score to zero, and make the sprites big.

In line 200, we initialize some strings: D will be used to modify the vertical motion of the long moving shapes (by negating it, then making it positive again, in line 210: D=-D). PT is the total points, AR the beginning number of arrows to shoot. The moving sprites are CALLed by line 210: the complicated formula is just to make sure they are evenly spread, and that their motion is set at random. Then the big arrow that you must shoot is put on screen, and Tex speaks to you if you have a synthesizer.

Finally, the top of the screen is refreshed: the remaining arrows (char. 134) are displayed, as well as your total points for the current game (zero when you start, of course). Note that if, when Tex waits for a key to press, you see the big arrow blink, don't worry: it is because there are sometimes more than four sprites on the same row and the arrow, having the highest number, disappears.

To launch the big arrow, you simply press the famous "any key." If you still wonder which key it is, I have one attached to my key chain. As soon as you press a key, sprite No. 7 is set into MOTION (you can make it slower of faster by making the second value, 50, higher or lower). CALL COINC looks for a coincidence between all sprites; if there is one (C), Tex beeps and CALLs STOP, which immediately stops the arrow and reLOCATEs it at the left of the screen (line 240). The remaining ARrow number is decremented; if no arrow left, you end the game in line 290, otherwise you are taken back to line 230 to display the remaining arrows and reset your total points.

If there is no coincidence, Tex goes back to line 260 until the big arrow reaches pixel column 225 (there are 256 such columns, but the position is calculated from the upper left corner of the sprite). If your arrow successfully reached the target, everything STOPs, you earn 150 points, and Tex may speak to you if you have a Speech Synthesizer. If not, line 280 will sound something that is close (?) to the sound of an arrow reaching a target. In both cases you go back to line 240: this winning arrow is still yours to shoot, since the variable AR is not modified.

Each game ends when you have no more arrows left, in line 290: the big arrow SPRITE (No. 7) is DELeted, and the high score displayed. If it is higher than the one from a previous game, it will flash with sounds (line 300). You are then asked if you want to play another game. If so, ALL the moving SPRITES are DELeted, the screen cleared, and you go back to line 200 to reinitialize Tex.

The second program, FUSION, is a utility that can be fun to play with. You enter two characters from the keyboard, and the program combines them into one and gives you its hexadecimal code.

It could be easily modified to use characters that had been redefined beforehand (you could enter two hex strings instead of the characters, or leave out the ACCEPTS and enter their definitions as program lines; in both cases, the character definition is held by C$ and D$, here CALL CHARPATed in line 180).

The program is very simple, except for the fusion routine in lines 190-260, which I will not even start to try to explain. The only change from Michel's program is the translation of the text strings.

ARROWS

```
100 ! *** ARROWS *** by Mich
el Montmigny, Sher-TI, 1986
!216
110 ! transl. by L. Dorais,
Ottawa UG, Dec. 1992 !111
120 CALL CLEAR :: CALL VCHAR
(1,1,31,48):: CALL VCHAR(1,3
1,31,48)!129
130 CALL SCREEN(5):: FOR T=1
 TO 8 :: CALL COLOR(T,16,2):
: NEXT T :: CALL COLOR(13,11
,2,14,11,1)!141
140 DISPLAY AT(7,9)BEEP:"A R
 R O W S": : : : :"   (USE A
NY KEY TO SHOOT)" !047
150 DISPLAY AT(22,2):"DO YOU
 HAVE A SPEECH":" SYNTHESIZE
R (Y/N)? N" :: ACCEPT AT(23,
21)VALIDATE("YN")SIZE(-1):A$
 :: IF A$="Y" THEN SP=1 !150
160 CALL CHAR(128,RPT$("03",
16)&RPT$("C0",16),136,RPT$("
1E",8))!157
170 Z$="00000000" :: CALL CH
AR(132,Z$&"FFFF0000"&Z$&Z$&"
00000406FFFF0604"&Z$)!143
```

(See Page 7)

ARROWS AND FUSION—

(Continued from Page 6)

```
180 CALL CLS :: HS=0 :: CALL
 MAGNIFY(4):: CALL VCHAR(10,
31,136,8)!038
190 ! === game === !235
200 RANDOMIZE :: D=1 :: PT=0
 :: AR=5 !193
210 FOR T=7 TO 27 STEP 4 ::
CALL SPRITE(#(T-3)/4,128,(T+
5)/4,INT(RND*256)+1,T*8-7,D*
INT(RND*41+30),0):: D=-D ::
NEXT T !052
220 CALL SPRITE(#7,132,10,96
,17):: IF SP THEN CALL SAY("
#READY TO START")!105
230 DISPLAY AT(2,18):RPT$("
"&CHR$(134),AR)!111
240 DISPLAY AT(2,2)SIZE(12):
"POINTS";PT !228
250 CALL KEY(0,K,S):: IF S=0
 THEN 250 ELSE CALL MOTION(#
7,0,50)!248
260 CALL COINC(ALL,C):: IF C
 THEN CALL SOUND(100,110,0,2
20,0,330,0,-8,0):: CALL STOP
: AR=AR-1 :: IF AR=-1 THEN
 290 ELSE GOTO 230 !238
270 CALL POSITION(#7,X,Y)::
IF Y<=225 THEN 260 ELSE CALL
 STOP :: PT=PT+150 :: IF SP
THEN CALL SAY("#GOOD WORK"):
: GOTO 240 !087
280 FOR T=1 TO 5 :: CALL SOU
ND(100,500,3*T,-6,3*T):: NEX
T T :: GOTO 240 ! boing!!! !
218
290 CALL DELSPRITE(#7):: DIS
PLAY AT(2,15):"HISCORE" :: I
F PT<=HS THEN DISPLAY AT(2,2
2):HS :: GOTO 310 !024
300 HS=PT :: FOR T=1 TO 8 ::
 DISPLAY AT(2,22):"" :: CALL
 SOUND(-99,T*110,0):: DISPLA
Y AT(2,22):HS :: NEXT T !110
310 DISPLAY AT(22,3):"DO YOU
 WISH TO PLAY":TAB(4);"ANOTH
ER GAME (Y/N) ?" !042
320 CALL KEY(0,K,S):: IF S=0
 THEN 320 :: IF K=89 THEN CA
LL DELSPRITE(ALL):: CALL CLS
 :: GOTO 200 !087
330 IF K<>78 THEN 320 ELSE C
ALL CLEAR :: END !160
340 SUB STOP :: CALL MOTION(
#7,0,0):: CALL LOCATE(#7,96,
17):: SUBEND !066
350 SUB CLS :: CALL VCHAR(1,
3,32,672):: SUBEND !237
```

FUSION

```
100 ! *** FUSION *** by Mich
el Montmigny, Sher-TI, 1986
!206
110 ! transl. by L. Dorais,
Ottawa UG, Dec. 1992 !111
120 CALL CLEAR :: CALL VCHAR
(1,1,31,48):: CALL VCHAR(1,3
1,31,48)!129
130 CALL SCREEN(5):: FOR C=1
 TO 14 :: CALL COLOR(C,2,16)
:: NEXT C :: H$="0123456789A
BCDEF" !172
140 DISPLAY AT(3,9):"F U S I
 O N" !171
150 DISPLAY AT(6,3):"ENTER A
 CHARACTER :" :: ACCEPT AT(6
,23)SIZE(-1):A$ :: IF A$=""
THEN 150 !179
160 DISPLAY AT(8,4):"AND NOW
 A SECOND :" :: ACCEPT AT(8,
23)SIZE(-1):B$ :: IF B$="" T
HEN 160 !091
170 DISPLAY AT(11,6):"one mo
ment please" !246
180 E$="" :: CALL CHARPAT(AS
C(A$),C$):: CALL CHARPAT(ASC
(B$),D$)!060
190 FOR I=1 TO 16 !111
200 E=0 :: C=POS(H$,SEG$(C$,
I,1),1)-1 :: D=POS(H$,SEG$(D
$,I,1),1)-1 !074
210 FOR N=3 TO 0 STEP -1 !17
2
220 IF C-2^N>=0 THEN CC$="1"
 :: C=C-2^N ELSE CC$="0" !15
0
230 IF D-2^N>=0 THEN DD$="1"
 :: D=D-2^N ELSE DD$="0" !15
7
240 IF CC$="1" OR DD$="1" TH
EN E=E+2^N !201
250 NEXT N !228
260 E$=E$&SEG$(H$,E+1,1):: N
EXT I !186
270 CALL CHAR(128,E$):: DISP
LAY AT(11,5)BEEP:"result: ";
CHR$(128): : :"    hexadecim
al code": :"    "&E$ !028
280 DISPLAY AT(19,4):"PRESS
[Q] to quit": :TAB(10);"ANOT
HER KEY":TAB(10);"to continu
e" !225
290 CALL KEY(0,K,S):: IF S=0
 THEN 290 !103
300 IF K<>81 AND K<>113 THEN
 CALL VCHAR(1,3,32,672):: GO
TO 140 ELSE CALL CLEAR :: EN
D !197
```

GEnie makes offer for TI, Geneve users

The GEnie TI and Orphans RoundTable is offering a special promotion for TI99/4A and Myarc Geneve 9640 users. GEnie is offering a $50 usage credit, to be used during the first month on-line with GEnie, valid until Dec. 31.

The monthly charge on GEnie is $8.95 U.S., $10.95 Canadian, which includes four free non-prime time hours.

To sign up, set your communications software for half-duplex (local echo) at 300, 1200 or 2400 baud; recommended parameters are 8 data bits, no parity, 1 stop bit. Dial 1-800-638-8369 (in Canada, 1-800-387-8330. Upon connection, type HHH. At the U#= prompt, type JOINGENIE and press return. At the offer code prompt, enter DGB225. This is a special code for the TI/Geneve offer. You will then be prompted for a signup with a major credit card.

For further information by voice, call 1-800-638-9636 or write GEnie, c/o GE Information Services, P.O. Box 6403, Rockville, MD 20850-1785.

THE ART OF ASSEMBLY — PART 40

It pays to be ...

By BRUCE HARRISON•

Back in your author's early childhood, there was no such thing as television. Network shows existed on the radio, and these provided much of the entertainment in our house. There were some shows with truly ridiculous ideas behind them. (Television didn't change that!)

As we were getting ready to write this, an old radio show came to mind. It was titled "It Pays to Be Ignorant". As we recall, it had a theme song that went "It pays to be ignorant, to be dumb, to be stupid, to be ignorant...." It was a quiz show on which the contestant who gave the dumbest answer to the question was the winner.

That came to mind because the main topic for this month's column came from a fellow who asked me what most TI programmers would call a dumb question. It happened one Sunday evening, while I was pecking away at my TI, trying to solve some knotty problem in the Compiler. The phone rang. The caller was a gentleman from Massachusetts, and he had several questions, all wrapped around the idea of taking an existing Option-3 assembly program and converting it to Option-5. He was trying to see if our Part 14 (Crossing the Bridge) would be useful, but not having any luck understanding the process. We explained that to use what's in Part 14, one needed to start with the source code for the program in question. Since he had only the object file, we told him that conversion would be impossible. Anybody who knows anything about TI assembly knows that!

We sent the gentleman a disk with some things to solve a couple of other problems he'd encountered, but his question about conversion without the source code kept nagging at us. The mind keeps working on something like this, even without much conscious thought. We kept hearing "What if we.... No! That won't work because...." Maybe 15 or 20 ideas came and went in this manner, but then finally there was an idea that just might work.

THE BRIGHT IDEA

If we took our "sandwich" from Part 14, and separated it into two source files, inserted REFs and DEFs in various places, then assembled the two parts of the sandwich into separate object files, we could do this impossible task. We would use the "linking" feature of the E/A Object loader (as we seldom do) to give these modules information about each other and the original object file, and the "can't be done" would become a "here's how."

Several conditions must be met for this method to work. First, the object file must be in relocatable code. Many are, since in most cases for an Option-3 file there's no need to AORG the program. The other requirement is that one must know the entry point label for the program. That seemed a likely thing, since one can't run an Option-3 from the Editor/Assembler without knowing the entry point. (Except for Auto-Start.)

(See Page 9)

Sidebar 40

```
* SIDEBAR 40
* FIRST PART, TWO SOURCE FILES
* FOR USE IN "CONVERTING" FROM
* OPTION-3 TO OPTION-5 WITHOUT
* HAVING ORIGINAL PROGRAM'S
* SOURCE CODE.


* SFIRST/S  FIRST SOURCE FILE
* IN THE TWO PLACES WHERE DEBUG APPEARS,
* IT SHOULD BE REPLACED BY THE ENTRY
* LABEL FROM THE OPTION-3 PROGRAM
* (SEE TEXT FOR DETAILS)
*
      DEF  SFIRST,SLOAD DEFINITIONS REQUIRED BY TI SAVE
UTILITY
      REF  DEBUG,EAUT,WST REFS TO OPTION-3 AND SLAST/O
SFIRST
SLOAD
      LWPI WST          LOAD TEMPORARY WORKSPACE
      LI   R9,EAUT      POINT R9 AT STORED UTILITIES
      LI   R10,>2000    POINT R10 AT >2094 IN LOW MEMORY
      LI   R4,>2676->2000  LOAD R4 WITH NUMBER OF BYTES TO
MOVE
PUTUT MOV  *R9+,*R10+   MOVE ONE WORD, INCREMENT POINTERS
BY TWO
      DECT R4           DECREMENT COUNT BY TWO
      JNE  PUTUT        IF NOT ZERO, REPEAT OPERATION
      B    @DEBUG       BRANCH TO OPTION-3'S ENTRY POINT
      END
*
* END OF SFIRST/S
*
* SLAST/S SECOND SOURCE FILE
* NO CHANGES REQUIRED
* FOR USE WITH ANY OPTION-3
*
      DEF  SLAST,EAUT,WST,SAVIT
      EVEN              INSURE THAT EAUT IS AT AN EVEN MEMORY
LOCATION
EAUT  BSS >2676->2000   LENGTH OF BSS IS >676 BYTES
      REF  SAVE         REFERENCE TI'S SAVE UTILITY
SLAST
* SLAST MARKS THE END OF WHAT THE SAVE UTILITY WILL PUT IN
MEM-IM FILE
SAVIT
      MOV  R11,@>8300   STASH R11
      LWPI WST          LOAD OUR TEMPORARY WORKSAPCE
      LI   R9,>2000     POINT R9 AT BEGINNING OF AREA TO BE
SAVED
      LI   R10,EAUT     POINT R10 AT MEMORY LOCATION ABOVE
      LI   R4,>2676->2000  LOAD R4 WITH NUMBER OF BYTES TO
MOVE
GETLP MOV  *R9+,*R10+   MOVE ONE WORD AND INCREMENT POINT-
ERS BY TWO
      DECT R4           DECREMENT COUNT BY TWO
      JNE  GETLP        IF NOT ZERO, REPEAT AT LABEL GETLP
      B    @SAVE        BRANCH DIRECTLY TO TI'S SAVE UTILITY
WST   BSS  32           OUR TEMPORARY WORKSPACE
      END
*
* END OF SLAST/S
```

THE ART OF ASSEMBLY—

(Continued from Page 8)

We decided to try a little experiment, using a handy Option-3 file called DEBUG, which was supplied with the E/A package. As shown in the sidebar, we split the sandwich into a part called SFIRST/S, which includes a ref to the label DEBUG, and SLAST/S, which didn't need any tailoring except to have a couple of DEFs added.

Here's how it worked. We assembled SFIRST/S into SFIRST/O, and SLAST/S into SLAST/O. Now we went into E/A, selected Option-3, and started loading object files. First, we loaded DSK4.SFIRST/O. Next, DSK5.DEBUG, then DSK4.SLAST/O, and finally DSK5.SAVE. (This last is the TI SAVE utility.)

Now we got to the Program Name prompt, and typed in the word SAVIT. This meant we'd run the part of the SLAST file that captures the E/A utilities and places them within the saved Option-5 file, then branches to TI's SAVE to actually create the Option-5 program files. We answered the SAVE prompt with DSK5.DEBUGOP5, and sure enough that worked.

Before the mail comes in, we'll admit openly that making an Option-5 out of the debugger doesn't, in itself, make any sense. The idea was to prove a concept, not to do something useful. DEBUGOP5 worked perfectly, whether loaded from E/A, or from our RAMdisk menu loader. Q.E.D.

STEP BY STEP

Okay, let's say you have an Option-3 file, and that you know the entry point, and you know that it's relocatable. (We'll tell you how to find out in a couple of paragraphs.)

Follow these steps in order:

1. Assemble SLAST/S into SLAST/O.

2. Edit SFIRST/S. Replace the two occurrences of DEBUG with the entry point name for your Option-3 program. Save that to disk and assemble it into SFIRST/O.

3. Find a disk that has TI's SAVE utility on it. Copy that onto the same disk with SFIRST/O, SLAST/O, and the original Option-3 object file. (From here on, we'll assume that disk is in drive 1, but, of course, any drive will do.)

4. Get into E/A Option-3. At the File Name prompt, type in DSK1.SFIRST/O Enter.

5. When that finishes loading, type in the name of the original object file, then Enter.

6. When that finishes loading, type in DSK1.SLAST/O Enter, and follow that loading with DSK1.SAVE Enter.

7. After DSK1.SAVE has loaded, leave the entry field blank and press Enter. Answer the Program Name prompt with SAVIT Enter.

8. You'll now have a prompt from TI's SAVE utility on the screen. Type in the name you are going to use for your Option-5 program file. (e.g., DSK1.MYOPT5) Press Enter. Drive 1's activity light will come on while the Option-5 file(s) are being saved.

9. Now you'll have on disk a program file or a series of them, depending on length. This should run from any method you choose to use for loading Option-5 programs.

(See Page 10)

```
* NEXT PART IS THE EXPERIMENT IN
* ROUNDING NUMBERS USING XMLLNK
* THIS MUST BE USED WITH THE XB
* PROGRAM SHOWN BELOW FOR TESTING



* TEST/S - SOURCE CODE FOR
* DEMONSTRATION OF ROUNDING
* THROUGH USE OF XMLLNK
* ENTRY LABEL TEST
*
* REQUIRED EQUATES
*
NUMREF EQU  >200C       NUMERIC REFERENCE
NUMASG EQU  >2008       NUMERIC ASSIGNMENT
XMLLNK EQU  >2018       XML LINKAGE VECTOR
CIF    EQU  >20         CONVERT INT TO F.P.
CFI    EQU  >12B8       CONVERT F.P. TO INT
FAC    EQU  >834A       F.P. ACCUMULATOR
*
* CODE SECTION
*
       DEF  TEST        DEFINE ENTRY LABEL
TEST   LWPI WS          LOAD OUR WORKSPACE
       CLR  R0          CLEAR R0, NOT ARRAY
       LI   R1,1        FIRST PARAMETER
       BLWP @NUMREF     GET PARAMETER VALUE
       BLWP @XMLLNK     USE XML LINKAGE
       DATA CFI         CONVERT TO INTEGER (ROUNDS)
```

THE ART OF ASSEMBLY—

(Continue d from Page 9)

HOW TO TELL

We promised to tell you how to tell whether an object file is absolute or relocatable. The answer depends on whether the object file is in compressed or uncompressed format, but you can tell in either case. Get into E/A Option 1, (Edit), and load the object file for editing. Several things can happen here that may alarm you, but just bear with us and proceed as instructed. You may get a Memory Full warning when loading the object file. Just ignore that, press enter, and then 2 to edit. For compressed object files, you'll get a Control Character Removed warning. Ignore that too.

In edit, you'll just look first at the first record in the file (top line of the screen). The first thing in that line, for an uncompressed file, will be a number in hex notation. If that number is 00000, this file is absolute, and can't be converted by our method. Anything other than 00000 means the conversion will probably work.

For compressed files, there won't be a readable number there in the first line. Instead, look down the leftmost column on the screen, where the tags are. If there's a series of Bs and/or 9s down the left column, then this is absolute code and conversion won't work. If the left column contains As and/or Cs, then this is relocatable code, and conversion will probably work.

In any case, after you've examined the object file with the editor, just get out of there, preferably with Function-=, so you won't be tempted to save this file back to disk. *Don't save* after this "edit"! If your answer was "yes" from this check, then use the step-by-step procedure above to do your conversion. There may be cases where this method won't succeed, as for example if the Option-3 file was close to filling the memory without adding our "sandwich." Should you need help, please give us a call at (301) 277-3467, any time from 9 a.m. through midnight Eastern time, seven days a week. This column is living proof that *no question is too dumb!* We may not immediately have the solution, but don't get discouraged, we'll keep trying.

ANOTHER WHAT IF

There's another potential hazard to this process for converting Option-3 to Option-5, and that's the case where the original Option-3 was set up with an auto-start label. If that's the case, the Option-3 file will start executing as soon as it loads from E/A, and you'll never get the chance to load either the SLAST/O or SAVE files. Yes, this too can be overcome, thanks to the genius of Tony McGovern. Funnelweb has an Option-3 loader which will simply ignore the auto-start, and allow you to continue with loading until you're ready to run something. Select Loaders from Funnelweb's menu, then the L/R Auto Off choice from the menu that appears. The only trick required here is that, after loading the SAVE utility, you'll have to press Function-3 to clear the entry field. Now press Enter, and use the left-right arrow keys to select SAVIT from the list of DEFs then on the screen. Press Function-6 with SAVIT selected, and you're in business. Here's another thanks to Tony McGovern, for a great feature in Funnelweb.

This loader that's provided by Funnelweb can also be used to find the entry point for a "mystery" Option-3 file. Simply select the Load/Run or the L/R Auto Off choice, then enter the object file's name and let it load. When it finishes, press Function-3 to delete all of the file name from the entry field, and press Enter. Any and all defined labels in that file will be shown on the screen. If there's only one, then that's the entry label you'd use at two places in SFIRST/S. If there are two or more, apply some educated guess logic to figure out which is the main entry, or else just select one of them and run the program from Funnelweb. If it seems to work as you'd expect, then you've guessed correctly. If not, try this process again and make a different selection.

```
      BLWP @XMLLNK      USE XML AGAIN
      DATA CIF          CONVERT BACK TO F.P.
      BLWP @NUMASG      RE-ASSIGN TO XB VARIABLE
      LWPI >83E0        LOAD GPL WORKSPACE
      B    @>6A         BRANCH TO GPL INTERPRETER
*
* DATA SECTION
*
WS    BSS  32           OUR OWN WORKSPACE
      END
*
* END OF SOURCE FILE TEST/S
*
* BELOW IS LISTING OF THE XB
* PROGRAM TO BE USED WITH
* THE OBJECT FILE TEST/O, MADE
* FROM THE SOURCE ABOVE
*
1 ! TEST PROGRAM - ROUNDING
2 ! USING ASSEMBLED FILE TEST/O
3 ! PUBLIC DOMAIN
4 ! BY BRUCE HARRISON
5 ! EXTENDED BASIC ONLY!
10 CALL INIT
20 CALL LOAD("DSK1.TEST/O")
30 INPUT "A NUMBER ":A
40 CALL LINK("TEST",A)
50 PRINT A:"ANOTHER? (Y/N)"
60 CALL KEY(0,K,S)
70 IF S<1 THEN 60
80 IF K=89 OR K=121 THEN 30
```

SERENDIPITY RESULTS

As so often happens in this business, we find out purely by accident that certain things on our TI behave in ways we didn't expect. That was the case with XB's Print Tab function, as we explained last month. Now our work on the Compiler has yielded another unexpected but pleasant surprise.

In the Compiler's own subroutines, we used XMLLNK to invoke the Convert Floating Point to Integer (CFI) routine in the console. In the way we are using it, it does exactly what we needed, taking an eight-byte floating point number at FAC and converting that to a two-byte integer at FAC. This routine produces weird results for numbers outside the range of -32768 through 32767, but that was of no concern to us, since such numbers were not involved in the situation we were working with.

In one of our Demo XB programs for the compiler, we used an INPUT statement so the user could decide how many times to execute a FOR-NEXT loop. INPUT places that number into a floating point variable, as you'd expect. In the process of using that number to control the FOR-NEXT loop, the compiled program takes that floating point number into FAC and uses CFI through

(See Page 11)

THE ART OF ASSEMBLY —

(Continued from Page 10)

XMLLNK to convert it to an integer as the loop's limit value. Just to see what would happen, we typed in a number that included a decimal part, as in 10.25. The program just ignored the decimal, and ran 10 repeats of the FOR-NEXT loop. Seeing that this worked, we tried entering 10.5, and were surprised to see the loop execute 11 times, not 10. Other quick checks followed, and, in all cases, if the decimal part was .499999 or less, the CFI routine rounded it off to the lower integer value, while if the decimal part was .5 or greater, the CFI routine correctly rounded to the next integer.

Today's sidebar contains a little routine you can use from Extended BASIC to see how this works, plus a short XB program to allow testing of this "rounding" process.

In this case, we've reconverted the integer to floating point so that we could use NUMASG to report the result back into XB and print it. We have run tests, and the only anomaly we found was that, for negative numbers, the rounding does not go to the next integer until the decimal part is >.5. Thus -10.5000 will round to -10, while -10.50000001 will round to -11.

Finally, we apologize if we seemed to poke fun at our friend in Massachusetts, whose phone call inspired the first part of our column. Like many of our readers, and like your author, he was struggling with trying to make his TI do something new, and had gotten frustrated enough to call for help. Many columns ago, we promised not to scold our readers by name, and we've kept that promise. Our friend in Massachusetts will no doubt recognize his case as we've described it. We hope he'll be happy that thanks to his having the courage to ask his question, an answer has been created that may help dozens of other users in our community.

Next month's topic is undecided. Perhaps we'll get another phone call that will lead to yet another discovery in our quest to know everything there is to know about this wonderful but strange machine. The only thing sure is that we will write a column for next month's issue, as this "old windbag" always has something to say.

Geneve 9640

'X-raying' the Geneve floppy

By JIM UZZELL

This article is intended to explain what and where information is stored on the floppy disk. The important sectors are 0,1 and the File Descriptor Record. While the previous statement is basically true, after reading through the information you will see that sectors 0,1 are the only constant sectors. Both sector 1 "SubDir" and FDR's can be anywhere on the disk.

So, get out your favorite sector editor and do some exploring.

SECTOR 0 - Volume Information Block VIB

ADDRESS	CONTENTS
0000-0009	Disk name-up to 10 characters
000A-000B	Total number of sectors on disk
	>0168=360 SS/SD 9(>09) sec/trk 40 trks 92160 bytes
	>0280=640 SS/DD 16(>10) 40 163840
	>02D0=720 SS/DD 18(>12) 40 184320
	>02D0=720 DS/SD 9(>09) 40 184320
	>0500=1280 DS/DD 16(>10) 40 327680
	>05A0=1440 DS/DD 18(>12) 40 368640
	>05A0=1440 SS/DD 18(>12) 80 368640
	>0A00=2560 DS/QD 16(>10) 80 655360
	>0B40=2880 DS/QD 18(>12) 80 737280
	>1680=5760 HiDen 36(>24) 80 1474560
000C	Number of sectors/track (see above)
000D-000F	DSK (>44534B)
0010	>50 = Disk protected >20 = Not protected
0011	Number of tracks >28=40 >50=80
0012-0013	Number of sides/density
	>0101 SS/SD >0202 DS/DD
	>0102 SS/DD >0202 DS/QD
	>0201 DS/SD >0203 DS/HD
0014-001D	1st Sub Directory Filename
001E-001F	Directory link for FDR of 1st SubDir See
0020-0029	2nd Sub Directory Filename NOTE 1
002A-002B	Directory link for FDR of 2nd SubDir below
002C-0035	3rd Sub Directory Filename
0036-0037	Directory link for FDR of 3rd Sub Dir
0038-00EB	Sector allocation bit map (AU)

This is a sector by sector bit map of sector use, 1=used 0=available. The first byte at >38 is for sectors 0 through 7(a fresh formatted DD or less with no subdirectories will have >03 which equals 0000 0011 or 2 sectors used—read right to left—sector 0 and sector 1), next byte is for sectors 8 through 15, and so on. For QD each bit equals 2 sectors, HiDen equals 4 sectors.

NOTE 1

It is highly recommended that you create sub directory prior to placing files on a disk because MDOS uses the next available sector to create the directory link to the file descriptor records, which would place the sub directories directory link at sector 2, 3, 4 and may make it possible for recovery of files easier in case the disk crashes.

SECTOR 1 DIRECTORY LINK

Each 16-bit word lists the sector number of the File Descriptor Record for an allocated file, in alphabetical order of the filenames. Each Subdirectory will have a sector identified as its directory link and will be structured the same as sector 1.

SECTOR 2 FILE DESCRIPTOR RECORDS FDR

ADDRESS	CONTENTS			
0000-0009	File name-up to 10 characters			
000A-000B	Extended Record Length (if =>256)			
000C	Filetype			
	NOT PROTECTED	PROTECTED	FILE CHANGE FLAG	
	>00=DIS/FIX	>08	>10	
	>01=Program(memory-image)	>09	>11	See
	>02=INT/FIX	>0A	>12	NOTE 2
	>80=DIS/VAR	>88	>90	below

(See Page 12)

THE ART OF ASSEMBLY—

(Continued from Page 11)

```
          >82=INT/VAR                >8A          >92
000D      Number of (MAXRECSIZE) records/sector or records/AU
000E-000F  Number of sectors allocated to the file
0010      For memory-image program files and variable-length
data files this contains the number of bytes used in the last
disk sector of file. This is used to determine end-of-file.
0011      MAXRECSIZE of data file (logical record length if
<256 else 0)
0012-0013  File record count, but with the second byte being
the
         high-order byte of the value. (i.e. >2301=>0123)
0014-0015  Date of creation     bits:  YYYY YYYM MMMd dddd
0016-0017  Time of creation            hhhh hmmm mmms ssss
0018-0019  Date of last change   secs are /2 remainder dis-
carded
001A-001B  Time of last change
001C-001E  Block link
```

For a file which is "not fractured", these three bytes point to the sectors on which the file is stored. If we let the 6 nybbles of these bytes be represented by >UVWXYZ then the word formed from >0XUV will be the sector number of the first sector of the file and >0YZW will be the logical offset of the last sector of the file. That is, the number of sectors in the file will be >0YZW + >0001(File Descriptor Record is not included in the sector count). If the file is "fractured", then this three-byte block refers to the first segment of the fractured file and will be followed by as many additional three-byte blocks as there are additional file segments. In each block the word >0XUV is the starting sector of the segment and the word >0YZW is one less than the total number of sectors used by the file through the current segment.

FILE STORAGE

Files are placed on the disk in first-come/first served manner. The first file written will start at sector >0042, and each subsequent file will be placed after it. Sectors >2 through >41 are reserved for File Descriptor Records. File data will be stored in these sectors if no other sectors are available. If more than 64 files are stored on a disk, additional File Descriptor Records will be allocated as needed, one sector at a time, from the next available pool of sectors unused. A Subdirectory Directory Link map will be allocated the same as a FDR as described in this section.

NOTE 2

You should never see these codes on a floppy only system. These codes are used as part of the hard drive structure. For hard drive, this byte in bit form of 76543210, will have bit 4 set if file has changed since last backup, i.e. I/V which equals >82 and in bit form would be 1000 0010 and would be 1001 0010 or >92 if file has changed. Also bit 5 will be set if file is a "DSK1" type file.

HOW TO DISTINGUISH A PROGRAM FROM A PROGRAM IMAGE FILE

Using a sector editor, find the first sector of the program, this can be done by using DSKU edit feature or looking at the file descriptor record of program at >1C->1E. Look at the *first three words* of that first sector. Now XOR word two with word three. If word one is equal to the result then it is either a MYBASIC, Basic, or X-Basic program. If word one is equal to the *two's* complement of the result then it is a protected program. If the result is neither then it is a program image file.

EXAMPLE

```
C305 A243 9EB8   Protected MYBASIC program
3CFB A243 9EB8   Not Protected

A243  1010 0010 0100 0011 word two
9EB8  1001 1110 1011 1000 word three
      ---------
3CFB  0011 1100 1111 1011 word one Not Protected
      ====================
       1100 0011 0000 0100
                       + 1
      ---------
C305  1100 0011 0000 0101 word one Protected
      ====================
XOR and TWO'S complement are explained in E/A manu-
al.
```

Hardware, software brokerage announced

Raymond Frantz of Phoenix, Arizona, has announced that he is setting up as a broker of used TI99/4A hardware, software and books.

Frantz says he is doing this as a service to users, with charges intended to cover his costs only. He says he does not buy anything, but rather lists and arranges sales of materials.

For further information, send a No. 10 self-addressed envelope with 52 cents of postage attached, to Raymond Frantz, "Broker of 99/4A Stuff" Worldwide, 502 N. 51st St., Ste. 2, Phoenix, AZ 85008-6649.

Free booklet focuses on cumulative trauma disorders

Data entry workers and frequent computer users are at high risk for injuries caused by high levels of repetitive finger motions or workspaces that don't fit them well.

The injuries, called cumulative trauma disorders, can lead to severe back and neck pain, eyestrain and tendinitis, as well as carpal tunnel syndrome, according to Nancy Osterman, director of the Workplace Ergonomics Institute at Quill Corporation.

A booklet called "The Easy 8-Step Ergonomics Guide" details techniques that help reduce the risk of such disorders.

For a free copy of the booklet, contact Quill Corporation, Workplace Ergonomics Institute, Public Relations Dept., 100 Schelter Rd., Lincolnshire, IL 60069.

Dallas Tiers plan reunion

The Dallas TI Home Computer Group invited all its present and former members to a reunion Oct. 29 at Informart. The event included a vendor area, refreshments and souvenirs.

Extended BASIC

Color Programming Tutor is a good place to start

The following program, called "Color Programming Tutor," was written by Jim Peterson. It's a tutorial on the use of color in Extended BASIC. It requires a memory expansion and works best with a color monitor. Peterson, who owned Tigercub Software, died early this year.

COLORTUTOR

```
100 CALL CLEAR :: MB$="187E6
600DBA53C2442187E55E76618182
45AA5C35A8199420000667EA5C3D
B24BD5A66BD7E817E7E5A3CC3A51
80066429900" ! USED IN LINE
378 !139
110 CALL SCREEN(16):: CALL C
HARPAT(64,@$):: CALL CHAR(64
,"0",37,"3C4299A1A199423C"):
 DIM C$(16),A$(16),K@$(14),
L$(12):: RANDOMIZE :: GOSUB
2150 !145
120 B,C,D,F,G,J,K,L,R,S,T,X,
Y,Z,CH,COL,K2,SE,SET,TM,TW,T
X,X@,Z@,CN(1),R@(1),L2,P=0 !
038
130 C$(1),@$,B$,CC$,A$(1),C@
$,CH$,C44$,C52$,X$,BB$,M@$(1
),L1$,L2$,L$(1)="" !166
140 !@P- !064
150 GOSUB 2790 ! SAVE LOWER
CASE HEX CODES !138
160 DISPLAY AT(3,10):"TIGERC
UB": :"  COLOR PROGRAMMING
TUTOR": :" TCX-1115 %Tigercu
b Software": :!157
165 ! programmed by Jim Pete
rson. Copyright 1984 Tigercu
b Software, 156 Collingwood
Ave., Columbus OHIO 43213 !0
46
170 DISPLAY AT(10,1):" The T
I-99/4A Home Computer":" can
 put 15 vivid colors on":" t
he screen at one time, and"
!222
180 DISPLAY AT(13,1):" can c
reate an endless vari-":" et
y of colorful effects." :: G
OSUB 1980 ! STOP,WAIT,WIPE !
068
190 GOSUB 2150 ! GET BORDER
!031
200 DISPLAY AT(3,1):" The co
lors and their codes":" are:
" !002
210 RESTORE 2090 !143
220 FOR J=1 TO 16 :: READ C$
(J):: DISPLAY AT(J+6,3):" ";
C$(J):: NEXT J :: GOSUB 1980
 ! LIST CODES AND COLORS !15
3
230 DISPLAY AT(6,1):" These
are the colors --" :: FOR D=
1 TO 400 :: NEXT D :: CALL C
LEAR :: GOSUB 2350 ! TO DISP
LAY AND LABEL COLORS !240
240 CH=24 :: CALL CLEAR :: F
OR R=4 TO 18 :: CALL COLOR(R
-4,R-2,R-2):: CALL HCHAR(R,1
,CH,32):: CH=CH+8 :: NEXT R
! HORIZONTAL RAINBOW !119
250 CH=40 :: FOR C=7 TO 21 :
: CALL VCHAR(1,C,CH,24):: CH
=CH+8 :: NEXT C :: FOR D=1 T
O 500 :: NEXT D :: GOSUB 213
0 ! VERTICAL RAINBOW !135
260 CALL VCHAR(1,31,1,96)::
GOSUB 2150 ! EDGES AND BORDE
R !219
270 DISPLAY AT(6,1):" There
are three different":" comma
nds to put color on":" the s
creen." !054
280 DISPLAY AT(10,1):" CALL
SCREEN will color the":" ent
ire screen, or any part":" o
f it that has not been":" co
lored by another command." :
: GOSUB 1980 !152
290 !@P+ !062
300 CALL CHARSET :: CALL CHA
R(64,"0"):: FOR SET=2 TO 8 :
: CALL COLOR(SET,2,16):: NEX
T SET ! SUB @ FOR SPACE !010
310 !@P- !064
320 FOR S=2 TO 16 :: CALL SC
REEN(S):: DISPLAY AT(12,5):C
$(S):: DISPLAY AT(15,5):"CAL
L@SCREEN(";STR$(S);")" :: GO
SUB 2330 ! DISPLAY, LABEL SC
REENS !209
330 NEXT S :: GOSUB 2330 ! P
AUSE !255
340 CALL CLEAR :: GOSUB 2150
 !017
350 DISPLAY AT(3,1):" The se
cond command is CALL":" COLO
R." !163
360 DISPLAY AT(6,1):" The pr
intable characters on":" the
 keyboard are divided":" int
o 12 sets of 8":" characters
 each." !029
370 DISPLAY AT(12,1):" In Ex
tended Basic, set 0":" and s
ets 13 and 14 are also" !205
380 DISPLAY AT(14,1):" avail
able for color, but":" sets
15 and 16 cannot be":" used.
" :: CALL CHAR(64,@$):: GOSU
B 1980 ! RESTORE @, GET BORD
ER !212
390 GOSUB 2260 ! PRINT ALL C
HAR SETS !242
400 CALL CHARSET :: DISPLAY
AT(1,1):" Each set can be gi
ven a":" different foregroun
d and":" background color. T
he" !188
410 DISPLAY AT(4,1):" foregr
ound color is the":" color o
f the character, the":" back
ground is the color of" !035
420 DISPLAY AT(7,1):" the re
st of the print":" space." !
253
430 DISPLAY AT(9,1):" Some c
ombinations of colors":" con
trast much better than":" ot
hers." !144
440 DISPLAY AT(12,1):" The s
hades of color can":" vary g
reatly depending on" !039
```

(See Page 14)

COLOR TUTOR—

(Continued from Page 13)

```
450 DISPLAY AT(14,1):" your
TV set and its color":" adju
stment. They can also":" var
y greatly in depth and" !165
460 DISPLAY AT(17,1):" brigh
tness depending on the":" co
mbinations and propor-":" ti
ons of colors on the":" scre
en." :: GOSUB 1980 !196
470 DISPLAY AT(9,1):" The fo
llowing program will":" show
 you the 3375 possible":" co
mbinations of screen," !171
480 DISPLAY AT(12,1):" foreg
round and background":" colo
rs - but it takes 11":" minu
tes to run, so you can" !068
490 DISPLAY AT(15,1):" stop
it any time with the":" Q ke
y." :: GOSUB 1980 !233
500 FOR J=1 TO 6 :: PRINT "
THIS IS A TEST": : :: NEXT J
 !211
510 PRINT " FOREGROUND": :"
BACKGROUND": :" SCREEN": : :
 : : : :: CALL CHAR(64,"FFFF
FFFFFFFFFFFF")!175
520 FOR G=3 TO 8 :: CALL HCH
AR(G,22,64,5):: NEXT G :: K2
=1 !212
530 FOR J=2 TO 16 :: FOR L=3
 TO 7 :: CALL COLOR(L,K2-ABS
(K2=17),J):: NEXT L ! GO THR
U FOREGROUND COLORS !251
540 DISPLAY AT(16,14)SIZE(4)
:" "&STR$(J)&" " !238
550 FOR K2=2 TO 16 :: IF K2=
J THEN 620 !022
560 DISPLAY AT(14,14)SIZE(4)
:" "&STR$(K2)&" " !031
570 FOR L=3 TO 7 :: CALL COL
OR(L,K2,J):: NEXT L ! GO THR
U BACKGROUND COLORS !134580
FOR S=2 TO 16 !122
590 DISPLAY AT(18,14)SIZE(4)
:" "&STR$(S)&" " :: CALL SCR
EEN(S)! GO THRU SCREEN COLOR
S !212
600 CALL KEY(3,K,ST):: IF K=
81 THEN 640 !079
610 NEXT S !233
620 NEXT K2 !019
630 NEXT J !224
640 CALL CLEAR :: CALL SCREE
N(16):: FOR SET=1 TO 8 :: CA
LL COLOR(SET,2,1):: NEXT SET
 :: CALL CHAR(64,"0")! RESTO
RE !191
650 DISPLAY AT(2,1):" WANT T
O TRY YOUR OWN COLOR":" COMB
INATIONS?": :!117
660 DISPLAY AT(4,1):"SELECT:
":" SCREEN COLOR CODE #":" F
OREGROUND COLOR CODE #":" BA
CKGROUND COLOR CODE #" !048
670 DISPLAY AT(10,1):" THE R
ESULT MAY NOT BE":" LEGIBLE,
 SO READ THIS NOW!": :" SELE
CT AND ENTER SCREEN,":" FORE
GROUND AND BACKGROUND" !245
680 DISPLAY AT(15,1):" COLOR
S (1-16)": :" THEN ENTER:":"
 S TO CHANGE SCREEN COLOR":"
 F TO CHANGE FOREGROUND":" B
 TO CHANGE BACKGROUND" !029
690 DISPLAY AT(22,1):" OR Q
TO QUIT" !187
700 ACCEPT AT(5,22)SIZE(2)VA
LIDATE(DIGIT)BEEP:S :: IF S<
1 OR S>16 THEN 700 !196
710 ACCEPT AT(6,26)SIZE(2)VA
LIDATE(DIGIT)BEEP:F :: IF F<
1 OR F>16 THEN 710 !172
720 ACCEPT AT(7,26)SIZE(2)VA
LIDATE(DIGIT)BEEP:B :: IF B<
1 OR B>16 THEN 720 !171
730 CALL CLEAR :: CALL SCREE
N(S):: GOSUB 750 !000
740 GOTO 760 !073
750 FOR SET=2 TO 8 :: CALL C
OLOR(SET,F,B):: NEXT SET ::
RETURN !163
760 DISPLAY AT(6,1):" SCREEN
 COLOR ";C$(S):: DISPLAY AT(
12,1):" FOREGROUND ";C$(F)::
 DISPLAY AT(18,1):" BACKGROU
ND ";C$(B)!110
770 DISPLAY AT(20,1):" TO CH
ANGE":TAB(6);"SCREEN COLOR T
YPE S":TAB(6);"FOREGROUND TY
PE F":TAB(6);"BACKGROUND TYP
E B" !127
780 DISPLAY AT(24,5):"TO QUI
T, TYPE Q" !112
790 CALL KEY(3,K,ST):: IF ST
<1 THEN 790 ELSE IF K=81 THE
N 850 ELSE IF K<>83 THEN 810
 !195
800 S=S+1-(15*ABS(S=16)):: C
ALL SCREEN(S):: DISPLAY AT(
,1):"SCREEN COLOR ";C$(S)::
GOTO 790 !034
810 IF K<>70 THEN 830 :: F=F
+1-(15*ABS(F=16)):: GOSUB 75
0 !017
820 DISPLAY AT(12,1):" FOREG
ROUND ";C$(F):: GOTO 790 !08
4
830 IF K<>66 THEN 790 :: B=B
+1-(15*ABS(B=16)):: GOSUB 75
0 !226
840 DISPLAY AT(18,1):" BACKG
ROUND ";C$(B):: GOTO 790 !05
9
850 CALL CLEAR :: CALL SCREE
N(16):: FOR SET=1 TO 8 :: CA
LL COLOR(SET,5,16):: NEXT SE
T !221
860 CALL CHARSET :: DISPLAY
AT(3,1):" The format of the
CALL":" COLOR statement is":
" CALL COLOR(set number,":"
foreground color number," !1
37
870 DISPLAY AT(7,1):" backg
ound color number)":"  For i
nstance:":"  CALL COLOR(2,5
,16)" !086
880 DISPLAY AT(11,1):" If th
e foreground and back-":" gr
ound colors are the same,":"
 for instance CALL COLOR" !0
27
890 DISPLAY AT(14,1):" (2,5,
5), any character in":" that
 set will print as a":" soli
d block of color." !244
900 DISPLAY AT(18,1):" CALL
COLOR does not put any":" co
lor on your screen. You":" m
ust use PRINT, CALL HCHAR" !
227
910 DISPLAY AT(21,1):" or CA
LL VCHAR to display":" chara
cters that you have":" color
ed.":"  Press any key" ::
GOSUB 2000 !142
920 DISPLAY AT(6,3):"NOW" ::
 DISPLAY AT(8,6):"THIS" :: D
ISPLAY AT(10,9):"IS" :: DISP
LAY AT(12,12):"IMPORTANT!!!"
 :: DISPLAY AT(20,15):"PRESS
 ANY KEY" !117
```

(See Page 15)

COLOR TUTOR—

(Continued from Page 14)

```
930 DISPLAY AT(12,12):"impor
tant" :: DISPLAY AT(12,12):"
IMPORTANT!!!" :: CALL KEY(0,
K,ST):: IF ST<>0 THEN 940 EL
SE 930 ! JIGGLE WORD !036
940 CALL CLEAR :: CALL SCREE
N(8)!236
950 DISPLAY AT(2,1):" When y
ou turn on the compu-":" ter
, the screen color is":" #8
cyan (pale blue)" !063
960 DISPLAY AT(6,1):" The fo
reground color of all":" the
 character sets is":" #2 bla
ck" !030
970 DISPLAY AT(10,1):" And t
heir background color":" is
#1 transparent" !044
980 DISPLAY AT(13,1):" Since
 the background color":" is
TRANSPARENT, the cyan":" col
or SHOWS THROUGH!!!" !232
990 DISPLAY AT(22,1):"Press
any key" !065
1000 CALL KEY(0,K,S):: IF S<
>0 THEN 1020 !004
1010 DISPLAY AT(14,1):" is
            the cyan" :: FOR
 D=1 TO 10 :: NEXT D :: DISP
LAY AT(14,1):" is TRANSPAREN
T  the cyan" :: GOTO 1000 !
FLASH WORD !079
1020 CALL CLEAR :: DISPLAY A
T(2,1):" When you turn on th
e compu-":" ter, or after yo
u CALL":" CLEAR, the screen
is filled" !103
1030 DISPLAY AT(5,1):" (exce
pt for a strip at top":" and
 bottom) with 24 rows":" and
 32 columns of the blank" !2
14
1040 DISPLAY AT(8,1):" space
-bar character, ASCII":" cod
e 32." !071
1050 DISPLAY AT(11,1):" Sinc
e the space character":" has
 no foreground pattern,":" a
nd the background color is"
!054
1060 DISPLAY AT(14,1):" tran
sparent, you can't see":" it
 of course." !146
1070 DISPLAY AT(16,1):" But
suppose we change the":" bac
kground color of Set 1,":" w
hich contains the space" !12
0
1080 DISPLAY AT(19,1):" char
acter, to light yellow:":" C
ALL COLOR(1,2,12)" :: DISPLA
Y AT(24,5):"Press any key" !
068
1090 CALL KEY(0,K,S):: IF S<
1 THEN 1090 !140
1100 CALL COLOR(1,2,12):: GO
SUB 1980 ! COLOR SET 1 !242
1110 DISPLAY AT(2,1):" The s
pace characters are":"all ye
llow and the punctua-":"tion
 characters in Set 1 are":"b
lack on yellow, but all" !13
4
1120 DISPLAY AT(6,1):"the ot
her character sets are":"sti
ll black on transparent,":"s
o the cyan screen color":"sh
ows through." !047
1130 DISPLAY AT(10,1):" This
 can be very useful,":" but
is very confusing if":" you
don't understand it!" !100
1140 DISPLAY AT(15,1):" Don'
t change the background":" c
olor of Set 1 unless you":"
know what you're doing!" !04
1
1150 DISPLAY AT(20,1):" Of c
ourse, we can also":"change
the background color":"of th
e other sets.": :"Press any
key" !070
1160 CALL KEY(0,K,ST):: IF S
T=0 THEN 1160 !120
1170 FOR SET=2 TO 12 :: CALL
 COLOR(SET,2,12):: NEXT SET
:: GOSUB 1980 ! CHANGE ALL S
ETS BACKGR COLOR !004
1180 DISPLAY AT(2,1):" The e
dge character 1 is al-":" wa
ys transparent. After you":"
 have changed the space" !06
6
1190 DISPLAY AT(5,1):" chara
cter to a color, you":" can
make the screen color":" sho
w on the left and right" !14
5
1200 DISPLAY AT(8,1):" borde
rs by":" CALL VCHAR(1,31,1,9
6)." !219
1210 DISPLAY AT(11,1):" This
 can give an entirely":" dif
ferent contrast to your":" c
olor combinations." !207
1220 DISPLAY AT(15,1):" This
 `window' is not":" affected
 by scrolling, but":" is era
sed by CALL CLEAR," !072
1230 DISPLAY AT(18,1):" so c
lear the screen by":" CALL V
CHAR(1,3,32,672)" !055
1240 CALL VCHAR(1,31,1,96)!
SCREEN WINDOW !026
1250 DISPLAY AT(23,3):"Press
 any key" :: CALL KEY(0,K,ST
):: IF ST<1 THEN 1250 !154
1260 CALL VCHAR(1,3,32,672):
: CALL COLOR(1,2,16):: CALL
SCREEN(2):: CALL VCHAR(1,31,
1,96):: GOSUB 1630 ! WINDOW
WIPE !179
1270 DISPLAY AT(2,1):" CALL
COLOR(1,2,16)":" CALL SCREEN
(2)":" CALL VCHAR(1,31,1,96)
":" FOR SET=2 TO 12" !110
1280 DISPLAY AT(6,1):"CALL C
OLOR(SET,2,16)":" NEXT SET":
 : : :" PRESS:":"  S TO CHAN
GE SCREEN COLOR" !206
1290 DISPLAY AT(13,1):"  X T
O CHANGE SET 1 BACK-":"  GRO
UND":"  B TO CHANGE OTHER BA
CK-":"  GROUNDS" !101
1300 DISPLAY AT(17,1):"  F T
O CHANGE FOREGROUNDS":"  Q T
O QUIT" !046
1310 B=16 :: C=3 :: F=2 !049
1320 CALL KEY(3,K,ST):: IF S
T<1 THEN 1320 !030
1330 ON POS("SXBFQ",CHR$(K),
1)+1 GOTO 1320,1340,1350,136
0,1390,1400 !147
1340 S=S+1-ABS(S=16)*15 :: C
ALL SCREEN(S):: GOTO 1320 !
GO THRU SCREEN COLORS !039
1350 B=B+1-ABS(B=16)*16 :: C
ALL COLOR(1,F,B):: GOTO 1320
 ! GO THRU BACK COLORS !153
1360 B=B+1-ABS(B=16)*15 ! GO
 THRU SET 1 BACK COLORS !055
1370 GOSUB 750 !064
1380 GOTO 1320 !124
```

(See Page 16)

COLOR TUTOR—

(Continued from Page 15)

```
1390 F=F+1-ABS(F=16)*15 :: G
OTO 1370 ! GO THRU FORE COLO
RS !049
1400 CALL SCREEN(4):: GOSUB
2130 !193
1410 DISPLAY AT(3,1):" Small
 areas of color on a":" blac
k screen appear washed":" ou
t and almost colorless," !24
1
1420 DISPLAY AT(6,1):" but a
re very bright when":" the b
order is white." !086
1430 DISPLAY AT(10,1):" Smal
l areas of color on a":" whi
te screen appear too":" dark
, but are brighter when" !11
8
1440 DISPLAY AT(13,1):" the
border is black." :: GOSUB 1
980 !070
1450 !@P+ !062
1460 DATA 00,18,24,3C,42,5A,
66,7E,81,99,A5,BD,C3,DB,E7,F
F !244
1470 !@P- !064
1480 RESTORE 1460 !022
1490 FOR J=1 TO 16 :: READ A
$(J):: NEXT J :: CALL COLOR(
1,2,2):: CALL SCREEN(2):: CA
LL VCHAR(1,31,1,96)!139
1500 GOSUB 1520 !069
1510 GOTO 1550 !099
1520 FOR SET=2 TO 8 :: X=INT
(14*RND+2)! RANDOMLY COLOR S
ETS !067
1530 Y=INT(14*RND+2):: IF Y=
X THEN 1530 !197
1540 CALL COLOR(SET,X,Y):: N
EXT SET :: RETURN !103
1550 FOR CH=40 TO 88 STEP 8
:: GOSUB 1580 ! RANDOM SYMME
TRICAL REDEFINED CHARS !170
1560 GOTO 1600 ! JUMP OVER I
N-LINE GOSUB !124
1570 !@P+ !062
1580 FOR L=1 TO 4 :: X=INT(1
6*RND+1):: B$=B$&A$(X):: CC$
=A$(X)&CC$ :: NEXT L :: CALL
 CHAR(CH,RPT$(B$&CC$,4)):: B
$,CC$=NUL$ :: RETURN !250
1590 !@P- !064
1600 FOR J=1 TO 3 :: CALL HC
HAR(INT(20*RND+3),INT(25*RND
+5),CH):: NEXT J :: NEXT CH
!117
1610 IF TM=1 THEN 1640 ELSE
CALL SCREEN(16):: GOSUB 1630
 !068
1620 GOTO 1650 !199
1630 FOR SET=2 TO 12 :: CALL
 COLOR(SET,2,16):: NEXT SET
:: RETURN !115
1640 CALL SCREEN(2)! CHANGE
TO WHITE W/BLACK BORDER !033
1650 FOR J=1 TO 5 :: GOSUB 1
520 ! SCATTER CHARS !078
1660 NEXT J :: TM=TM+1 :: GO
SUB 1980 !178
1670 IF TM=2 THEN 1680 ELSE
CALL CLEAR :: CALL COLOR(1,1
6,16):: CALL SCREEN(16):: CA
LL VCHAR(1,31,1,96):: GOTO 1
500 !023
1680 FOR SET=1 TO 12 :: CALL
 COLOR(SET,4,16):: NEXT SET
:: CALL CLEAR :: CALL CHARSE
T :: GOSUB 2800 !129
1690 CALL SCREEN(16):: DISPL
AY AT(6,1):" If the foregrou
nd and back-":" ground color
s are the same":" the charac
ter will print as" !160
1700 DISPLAY AT(9,1):" a sol
id block of that color": : :
 :" FOR SET=2 TO 12":" CALL
COLOR(SET,SET,SET)":" NEXT S
ET" :: GOSUB 1980 !006
1710 CALL CLEAR :: FOR SET=2
 TO 13 :: CALL COLOR(SET,SET
,SET):: NEXT SET :: GOSUB 22
00 ! DISPLAY CONCENTRIC SQUA
RES !224
1720 FOR D=1 TO 300 :: NEXT
D :: GOSUB 2130 !031
1730 DISPLAY AT(6,1):" If th
e foreground and back-":" gr
ound colors are different":"
 the results are even more"
!073
1740 DISPLAY AT(9,1):" color
ful.": : :" FOR SET=2 TO 13"
:" CALL COLOR(SET,SET,SET+1)
":" NEXT SET" :: GOSUB 1980
!102
1750 CALL CLEAR :: FOR SET=2
 TO 13 :: CALL COLOR(SET,SET
,SET+1):: NEXT SET :: GOSUB
2200 ! DISPLAY CONCENTRIC SQ
UARES !155
1760 FOR D=1 TO 400 :: NEXT
D :: GOSUB 2130 !032
1770 DISPLAY AT(3,1):" As yo
u have already seen in":" th
e screen borders and":" wipe
s, redefined characters" !03
5
1780 DISPLAY AT(6,1):" in tw
o colors can give very":" co
lorful effects. You can":" l
earn one way to create" !146
1790 DISPLAY AT(9,1):" these
 by listing lines":" 3780-38
80, which produce":" the scr
een border." :: GOSUB 1980 !
179
1800 GOSUB 2450 ! TO FLASH T
EXT ON AND OFF !207
1810 GOSUB 2530 ! FOR BLENDE
D COLORS !197
1820 GOSUB 2810 ! FOR SPRITE
 ROUTINES !110
1830 CALL CHARSET !118
1840 CALL DELSPRITE(ALL)!115
1850 DISPLAY AT(12,1)ERASE
LL:" THIS WILL BE THE GRAND"
:" FINALE OF THIS LESSON." !
188
1860 FOR CH=40 TO 136 STEP 8
 :: GOSUB 1580 ! REDEFINE SY
MMETRIC CHARS !035
1870 NEXT CH :: GOSUB 3260 !
187
1880 CALL COLOR(1,1,1):: GOS
UB 2200 ! FOR CONCENTRIC SQU
ARES !242
1890 FOR J=1 TO 5 :: GOSUB 3
260 !215
1900 NEXT J :: CALL CLEAR ::
 CALL SCREEN(5):: FOR J=1 TO
 30 :: X=INT(10*RND):: IF X<
>0 THEN 1920 ! PICK RANDOM B
ICOLORS !208
1910 CALL CLEAR ! RANDOM BAR
S ROUTINE !195
1920 IF X<>1 THEN 1930 :: CA
LL SCREEN(INT(14*RND+2))!058
1930 CALL HCHAR(INT(22*RND+2
),INT(26*RND+4),INT(13*RND+5
)*8,INT(20*RND+2))!033
1940 CALL VCHAR(INT(22*RND+2
),INT(26*RND+4),INT(13*RND+5
)*8,INT(10*RND+2))!046
```

(See Page 17)

COLOR TUTOR—

(Continued from Page 16)

```
1950 IF J>10 THEN 1970 ! RAN
DOM SYMMETRIC GIANT SPRITES
ERROR: Procedure File Not
Found!
!009
1960 CALL SPRITE(#J,INT(12*R
ND+5)*8,INT(15*RND+2),100,12
0,INT(10*RND)-INT(10*RND),IN
T(10*RND)-INT(10*RND))!178
1970 NEXT J :: GOTO 3120 !23
7
1980 DISPLAY AT(23,15):"pres
s any key" ! GOSUB TO HOLD,
WAIT FOR KEY, RANDOM WIPE !1
84
1990 !@P+ !062
2000 CALL KEY(0,K,ST):: IF S
T<1 THEN 2000 !197
2010 !@P- !064
2020 TW=TW+1-2*(ABS(TW=2)):
 X=INT(15*RND+2)!178
2030 Y=INT(15*RND+2):: IF Y=
X THEN 2030 !188
2040 CALL COLOR(14,X,Y):: ON
 TW GOTO 2050,2060 !077
2050 CALL VCHAR(1,1,143,768)
:: GOTO 2070 !026
2060 CALL HCHAR(1,1,143,768)
!029
2070 CALL CLEAR :: RETURN !2
19
2080 !@P+ !062
2090 DATA 1  TRANSPARENT,2
BLACK,3  MEDIUM@GREEN,4  LIG
HT@GREEN,5  DARK@BLUE,6  LIG
HT@BLUE,7  DARK@RED,8  CYAN
!214
2100 DATA 9  MEDIUM@RED,10 L
IGHT@RED,11 DARK@YELLOW,12 L
IGHT@YELLOW,13 DARK@GREEN,14
 MAGENTA,15 GREY,16 WHITE,DU
MMY !254
2110 !@P- !064
2120 CALL SCREEN(4)!149
2130 CALL CLEAR :: FOR SET=1
 TO 12 :: CALL COLOR(SET,2,3
):: NEXT SET :: RETURN ! GOS
UB TO RESTORE COLORS !172
2140 FOR SET=1 TO 12 :: CALL
 COLOR(SET,2,3):: NEXT SET :
: RETURN !061
2150 X=INT(42*RND+1)*2-1 ::
CALL CHAR(143,SEG$(MB$,X,16)
):: Y=INT(12*RND+5)! GOSUB F
OR RANDOM COLORED BORDER DES
IGN !010
2160 Z=INT(12*RND+5):: IF Z=
Y THEN 2160 !066
2170 !@P+ !062
2180 CALL COLOR(14,Z,Y):: CA
LL HCHAR(1,2,143,30):: CALL
HCHAR(24,2,143,30):: CALL VC
HAR(1,2,143,24):: CALL VCHAR
(1,31,143,24):: RETURN !167
2190 !@P- !064
2200 FOR X=1 TO 12 ! DISPLAY
 CONCENTRIC SQUARES !127
2210 CALL HCHAR(X,1+X,8*X+32
,30-X-TX)!130
2220 CALL HCHAR(25-X,1+X,8*X
+32,30-X-TX)!117
2230 CALL VCHAR(X,1+X,8*X+32
,25-X-TX)!148
2240 CALL VCHAR(X,31-X,8*X+3
2,25-X-TX):: TX=TX+1 :: NEXT
 X :: TX=0 :: RETURN !123
2250 ! TO PRINT ALL SETS AND
 CHARS !207
2260 FOR CH=128 TO 143 :: CA
LL CHAR(CH,"00FF81818181FF")
:: NEXT CH :: CALL COLOR(13,
2,16,14,2,16)!163
2270 FOR SE=0 TO 14 :: PRINT
 " SET";SE;TAB(9);"";:: FOR
L=24+SE*8 TO 30+SE*8 :: PRIN
T CHR$(L);" ";:: NEXT L :: P
RINT CHR$(L):: NEXT SE ! PRI
NT ALL !145
2280 PRINT : : : : :"PRESS A
NY KEY" :: GOSUB 2150 ! GET
BORDER !056
2290 CALL KEY(0,K,ST):: IF S
T<1 THEN 2290 !232
2300 FOR SET=0 TO 14 :: CALL
 COLOR(SET,SET+2,1):: NEXT S
ET :: CALL COLOR(1,2,16):: C
ALL SCREEN(2):: CALL VCHAR(1
,31,1,96):: GOSUB 1980 ! COL
OR EACH SET DIFFERENTLY !233
2310 CALL COLOR(1,2,1):: GOS
UB 2120 !205
2320 RETURN !136
2330 FOR D=1 TO 200 :: NEXT
D :: RETURN ! DELAY !014
2340 ! ROUTINE TO DISPLAY AN
D LABEL COLORS !039
2350 FOR SET=12 TO 14 :: CAL
L COLOR(SET,1,1):: NEXT SET
:: CALL HCHAR(1,1,120,256)::
 CALL HCHAR(9,1,128,256):: C
ALL HCHAR(17,1,136,256)!106
2360 !@P+ !062
2370 DATA 4,3,13,8,6,5,10,9,
7,12,11,14,2,15,16 !147
2380 !@P- !064
2390 RESTORE 2370 :: R@(1)=3
 :: R@(2)=11 :: R@(3)=19 !17
8
2400 FOR J=1 TO 15 STEP 3 ::
 FOR COL=1 TO 3 :: READ CN(C
OL):: CALL COLOR(11+COL,CN(C
OL),CN(COL)):: DISPLAY AT(R@
(COL),5)SIZE(16):C$(CN(COL))
!033
2410 NEXT COL :: DISPLAY AT(
24,5)SIZE(15):" TOUCH ANY KE
Y" !156
2420 CALL KEY(0,K,ST):: IF S
T<1 THEN 2420 !107
2430 NEXT J :: RETURN !234
2440 ! ROUTINE TO FLASH TEXT
 ON AND OFF !042
2450 FOR SET=1 TO 12 :: CALL
 COLOR(SET,1,1):: NEXT SET !
048
2460 PRINT " As you have jus
t seen, the":" character set
s can be given":" the transp
arent foreground" !172
2470 PRINT " color before te
xt is placed":" on the scree
n, and then":" changed back
to another" !222
2480 PRINT " color to sudden
ly become":" visible. You ca
n do the":" same with graphi
cs, of":" course.": : : : :
: : : :!043
2490 FOR T=1 TO 3 :: FOR SET
=1 TO 12 :: CALL COLOR(SET,1
,1):: NEXT SET !247
2500 FOR SET=12 TO 1 STEP -1
 :: CALL COLOR(SET,2,1):: NE
XT SET !159
2510 FOR D=1 TO 100 :: NEXT
D :: NEXT T :: GOSUB 1980 !2
42
2520 RETURN !136
2530 CALL CLEAR :: CH$=RPT$(
"AA55",4)! ROUTINE TO SHOW B
LENDED COLORS !098
2540 DISPLAY AT(8,1):" The 1
5 basic colors can be":"blen
```

(See Page 18)

COLOR TUTOR—

(Continued from Page 17)

```
ded into hundreds of":"shade
s by combining two of":"them
 in a character which" !207
2550 CALL CHARPAT(34,C@$)::
CALL CHAR(37,C@$)!107
2560 DISPLAY AT(12,1):"is re
defined by CALL CHAR":"(CH,R
PT$(%AA55%,4))" :: DISPLAY A
T(24,1):"Press any key" !226
2570 CALL KEY(0,K,S):: IF S<
1 THEN 2570 !090
2580 GOSUB 2600 !130
2590 FOR SET=2 TO 12 :: CALL
 COLOR(SET,2,1):: NEXT SET :
: GOTO 2720 !173
2600 CALL CLEAR :: FOR CH=40
 TO 136 STEP 8 :: CALL CHAR(
CH,CH$):: NEXT CH !037
2610 CALL VCHAR(1,1,40,768):
: DISPLAY AT(24,1):"    ANY
KEY TO GO ON " !172
2620 FOR B=2 TO 16 :: FOR F=
2 TO 16 :: IF F=B THEN 2640
!119
2630 CALL COLOR(2,F,B)!063
2640 CALL KEY(0,K,S):: IF S<
>0 THEN 2660 !115
2650 FOR D=1 TO 100 :: NEXT
D :: NEXT F :: NEXT B !169
2660 CH=40 :: FOR C=3 TO 28
STEP 2 :: CALL VCHAR(1,C,CH,
48):: CH=CH+8 :: NEXT C :: D
ISPLAY AT(24,1):"      ANY K
EY TO GO ON" !066
2670 F=2 :: B=2 !124
2680 FOR S=2 TO 14 :: CALL C
OLOR(S,F,B):: F=F+1 :: IF F=
17 THEN F=2 :: B=B+1 !130
2690 IF B=17 THEN 2670 !167
2700 CALL KEY(0,K,ST):: IF S
T<>0 THEN RETURN !108
2710 NEXT S :: GOTO 2680 !06
0
2720 CALL CLEAR :: GOSUB 280
0 ! ROUTINE TO SHOW BLEEDING
 !228
2730 DISPLAY AT(6,1):" The h
orizontal scan of your":"TV
set may also produce":"multi
color effects by" !094
2740 DISPLAY AT(9,1):"bleedi
ng colors into each":"other,
 especially when the":"patte
rn is vertical.":" However,
this effect may" !219
2750 DISPLAY AT(13,1):"not b
e the same on all TVs." :: D
ISPLAY AT(24,1):"Press any k
ey" !086
2760 CALL KEY(0,K,ST):: IF S
T<1 THEN 2760 !192
2770 CALL CLEAR :: CH$=RPT$(
"AA",8):: GOSUB 2600 !017
2780 FOR SET=2 TO 12 :: CALL
 COLOR(SET,2,1):: NEXT SET :
: RETURN ! RESTORE ALL TO BL
ACK ON TRANSP !241
2790 X=0 :: FOR CH=40 TO 136
 STEP 8 :: X=X+1 :: CALL CHA
RPAT(CH,K@$(X)):: NEXT CH ::
 RETURN ! SAVE HEX CODES OF
LOWER CASE !196
2800 X@=0 :: FOR CH=40 TO 13
6 STEP 8 :: X@=X@+1 :: CALL
CHAR(CH,K@$(X@)):: NEXT CH :
: RETURN ! RESTORE LOWER CAS
E CHARS !026
2810 CALL CLEAR :: GOSUB 280
0 ! SPRITE ROUTINES !176
2820 DISPLAY AT(2,1):" The t
hird way to place":"color on
 your screen is by":"the use
 of sprites." !017
2830 DISPLAY AT(6,1):" You c
an have up to 28 of":"them o
n the screen at one":"time.
The only problem is":"that i
f you put more than" !061
2840 DISPLAY AT(10,1):"4 of
them in a horizontal":"row,
only the 4 lowest":"numbered
 ones will be":"visible." !1
99
2850 DISPLAY AT(15,1):" Each
 sprite can be of the":"shap
e of any character, or":"can
 be any redefined char-":"ac
ter." :: GOSUB 1980 !099
2860 DISPLAY AT(3,1):" By us
ing MAGNIFY(4), and":"redefi
ning an ASCII which is":"div
isible by 4 with":"CALL CHAR
(CH,RPT$(% F%,64))," !051
2870 DISPLAY AT(7,1):"you ca
n color an area of":"4 x 4 p
rint spaces." !148
2880 !@P+ !062
2890 CALL MAGNIFY(4):: FOR C
H=128 TO 140 STEP 4 :: CALL
CHAR(CH,RPT$("F",64)):: Z@=Z
@+1 :: CALL SPRITE(#Z@,CH,Z@
+4,100,40+Z@*32):: NEXT CH :
: GOSUB 1980 !DISPLAY 4 BLOC
KS !213
2900 !@P- !064
2910 !@P+ !062
2920 CALL DELSPRITE(ALL):: D
ISPLAY AT(3,1):" A sprite ca
n be given any":"color, rega
rdless of the":"color assign
ed to the set":"from which i
ts character is":"taken." !0
18
2930 !@P- !064
2940 DISPLAY AT(12,1):RPT$("
A",28):: DISPLAY AT(14,1):RP
T$("Z",28)!PRINT ROWS OF A,
Z !108
2950 CALL MAGNIFY(2):: CALL
SPRITE(#1,65,11,90,8,0,10,#2
,90,5,110,8,0,10):: GOSUB 19
80 ! SPRITES A & Z !005
2960 CALL DELSPRITE(ALL):: C
ALL CHARPAT(44,C44$):: CALL
CHARPAT(52,C52$):: CALL CHAR
(123,C44$,96,C52$)!REDEFINE
COMMA & 4 FOR NEXT TEXT !018
2970 CALL SCREEN(8):: DISPLA
Y AT(5,1):" A sprite can hav
e only one":"color{ but up t
o ` sprites":"can overlay ea
ch other{ and" !199
2980 DISPLAY AT(8,1):"sprite
s can overlay the":"colors p
roduced by the other":"metho
ds" !SPRITES OF OVERLAID SQU
ARES !240
2990 M@$(1)="FFFFC0C0C0C0C0C
0C0C0C0C0C0C0FFFFFFFF0303030
303030303030303FFFF" !069
3000 M@$(2)="00003F3F3030303
0303030303F3F00000000FCFC0C0
C0C0C0C0C0C0CFCFC" !210
3010 M@$(3)="000000000F0F0C0
C0C0C0F0F0000000000000000F0F
030303030F0F0" !095
3020 M@$(4)=RPT$("0",13)&"30
30303"&RPT$("0",24)&"C0C0C0C
" !229
3030 CALL CHAR(124,M@$(1),12
8,M@$(2),132,M@$(3),136,M@$(
4))!205
3040 CALL MAGNIFY(4):: CALL
```

(See Page 19)

COLOR TUTOR—

(Continued from Page 18)

```
SPRITE(#1,124,2,80,112,#2,12
8,16,80,112,#3,132,11,80,112
,#4,136,16,80,112)!152
3050 FOR CH=40 TO 52 STEP 4
:: FOR L=1 TO 4 :: X$=SEG$("
0018243C425A667E8199A5BDC3DB
E7FF",INT(16*RND+1)*2-1,2)::
 BB$=BB$&X$ :: CC$=X$&CC$ ::
 NEXT L !OVERLAID SYMMETRIC
SPRITES !013
3060 CALL CHAR(CH,RPT$(BB$&C
C$,4)):: BB$,CC$="" :: NEXT
CH !233
3070 CALL SPRITE(#5,40,1,120
,112,#6,44,1,120,112,#7,48,1
,120,112,#8,52,1,120,112)!14
1
3080 DISPLAY AT(24,1):"Press
 any key to continue" !215
3090 FOR J=1 TO 8 :: CALL CO
LOR(#J,INT(15*RND+2)):: CALL
 KEY(0,K,ST):: IF ST<>0 THEN
 RETURN ! CHANGE SPRITE COLO
RS !244
3100 NEXT J :: GOTO 3090 !20
7
3110 END !139
3120 CALL CLEAR :: FOR S=2 T
O 14 :: CALL COLOR(S,1,1)::
NEXT S !187
3130 FOR L=1 TO 12 :: FOR L2
=1 TO 12 :: X$=CHR$(INT(13*R
ND+5)*8):: L1$=L1$&X$ :: L2$
=X$&L2$ :: NEXT L2 :: L$(L)=
L1$&L2$ :: PRINT TAB(3);L$(L
):: L1$,L2$=NUL$ :: NEXT L !
139
3140 FOR P=12 TO 2 STEP -1 :
: PRINT TAB(3);L$(P):: NEXT
P :: PRINT TAB(3);L$(1);!111
3150 CALL DELSPRITE(ALL)!115
3160 GOSUB 3250 !014
3170 Z=INT(7*RND+1):: ON Z G
OSUB 3180,3220,3190,3220,320
0,3220,3210 :: GOTO 3160 !13
7
3180 FOR C=2 TO 14 :: CALL C
OLOR(C,1,1):: GOSUB 3250 ::
NEXT C :: RETURN !210
3190 CALL SCREEN(INT(15*RND+
2)):: RETURN !100
3200 X=INT(15*RND+2):: FOR C
=2 TO 14 :: CALL COLOR(C,X,X
):: NEXT C :: GOSUB 3250 ::
RETURN !232
3210 FOR C=2 TO 14 :: X=INT(
15*RND+2):: CALL COLOR(C,X,X
):: GOSUB 3250 :: NEXT C ::
RETURN !232
3220 FOR C=2 TO 14 :: X=INT(
13*RND+2)!192
3230 Y=INT(13*RND+2):: IF Y=
X THEN 3230 !111
3240 CALL COLOR(C,X,Y):: GOS
UB 3250 :: NEXT C :: RETURN
!165
3250 CALL KEY(0,K,ST):: IF S
T<>0 THEN STOP ELSE RETURN !
133
3260 FOR S=2 TO 14 :: X=INT(
15*RND+2)!210
3270 Y=INT(15*RND+2):: IF Y=
X THEN 3270 ELSE CALL COLOR(
S,X,Y)!218
3280 NEXT S :: RETURN !243
```

1995 Lima MUG Conference slated

The Lima Multi User Group Conference has been scheduled for April 29 at Reed Hall on the campus of the Ohio State University at Lima, Ohio, according to Charles Good of the Lima Users Group.

The event is free of charge for those attending. Vendors may also set up displays free of charge. Tables may be set up after 4 p.m. April 28, according to Good.

For further information, contact the Lima Users Group, P.O. Box 647, Venedocia, OH 45894, or Charles Good at (419) 667-3131 or E-mail cgood@lima.ohio-state.edu.

San Diego TI SIG folds

The TI SIG (special interest group) of the San Diego Computer Society has ceased meeting because of a lack of interest, according to David Sichak, newsletter editor of the society.

San Diego is also the home of the Southern California Computer Group, which is scheduled to host the TI Fest West in February

1994 TI FAIRS

OCTOBER

9th International TI-Meeting, Oct. 14-16, Kirch l. Gemeindehaus Roshorf, German, sponsored by TI-Club Goettingen. For information, contact Jörg Kirstan, Mengershäuser Weg 5, D-37124 Rosdorf, Germany, tel. 01551/781153; Reinhard Obuch, Keplerstr. 5, D-37085 Göttingen, Germany, tel. 0551/46405; or Hans-Hartmut Kortry, Grüner Weg 10, D-37181 Hardegsen, Germany, tel. 05505/1470.

NOVEMBER

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

1995 TI FAIRS

FEBRUARY

Fest West '95, Feb. 18, Fabulous Inn, San Diego, California. Contact Southern California Computer Group, P.O. Box 152535, San Diego,CA 92195, or call the SCCG BBS, (619) 263-9135, User No. 25, password FEST

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.

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.

Hard drives

Different types and characteristics of drives described

By GARY COX

(Reprinted from the June 1994 newsletter of the Mid-South 99ers)

This article was originally written for an IBM compatible, but I have modified the article to make it generic enough to apply to any computer, I hope.

Where did the term "hard disk" come from? Obviously it came from the way the disk was constructed, as, unlike floppy disks that can bend, hard drive platters are solid and thus they cannot bend. Another name for a hard drive is "Fixed Disk" or "Winchester." The meaning of the term "Fixed Disk" is obvious as the disk is fixed in place and cannot be removed (although removable hard drives exist but they are actually fixed in a case that can be removed from the housing).

The term "Winchester" is also used to describe a hard drive. The term "Winchester" goes back to the 1960s where IBM developed a high-speed hard disk that had 30 megabytes of fixed platter storage and 30 megabytes of removable platter storage, thus calling the disk a 30-30 drive. However, instead of the disk being called a 30-30 drive, it gained the nickname "Winchester" referencing the "Winchester 30-30 caliber rifle." Nowadays the term "hard drive" is usually the only term that is used, but you may see references to older hard drives and controllers as "Winchester" drives or "Winchester" controllers.

BASIC COMPONENTS

The basic components of all hard drives are the same; each hard drive contains at least one disk platter, rewrite heads, a head actuator or arm, a platter motor or spindle actor, a circuit board, connectors and a case. The platters are stored one on top of another with spacing in between for the read/writer heads to magnetically store data on the media. All the inner workings are sealed inside a metal housing that protects the platter and heads from dirt and dust. In fact, the tolerance between the platter and head is so close that even one particle of dust can cause a scratch on the platter surface, thus damaging the drive as one particle of dust is larger than the gap between the head and the platter. Therefore, a hard drive should *never* be opened! Hard drives are manufactured in dust-free "Class 100 clean rooms" which are cleaner than operating rooms! CMI (Computer Memories Incorporated) had trouble with a contaminated clean room that caused some of the early hard drives manufactured by them to be contaminated, thus causing a variety of problems, including a total disk crash.

Each hard drive usually contains more than one platter with a head attached to an arm on each of the platters creating a surface for data (kinda like a record player).

A variety of types of hard drives exists, with each type using a different method of coding the information on the disk. The different coding methods provide for the different sizes, reliability and speed of the hard drive. The different coding methods are more or less a reflection of the changes in technology over the years. The physical size of the hard drive is no indication of the amount of storage space, as a very large (full height) drive might be only 10 megabytes where a hard drive smaller than the size of your hand might be able to store 300 megabytes! The encoding method as well as the way the drive is build determines the drive's size in megabytes.

The tolerance between the platter and head is so close that even one particle of dust can cause a scratch on the platter surface, thus damaging the drive as one particle of dust is larger than the gap between the head and the platter.

MFM DRIVES

MFM-type drives are among the oldest and most common found in older computers. MFM stands for Modified Frequency Modulation and originated out of FM Encoding or Frequency Modulation Encoding. FM Encoding is an outdated coding scheme that used half the disk up just for timing signals and thus was replaced with MFM coding.

Another type of drive coding is RLL, which stands for Run Length Limited. RLL is more complicated than MFM, but allows for the placement of more data on the disk for the given amount of space. RLL may often be referred to in technical publications as RLL 2,7, referencing that the run length is limited to 7.

Also used is RLL 3,9 (also called ARLL) which increases disk space even more. Drives for use as MFM and RLL come new from the factory completely blank, as the encoding for MFM and RLL drives is created during the low-level format. This low-level format is done by the user by invoking a built-in program in the ROM on the controller or by using a program to low-level format it. The low-level format creates the tracks that the drive uses to store data. As well, the low-level format locates any bad areas and locks them off. Thus, the encoding method is determined by the controller that you use, but with an MFM drive you should use *only* an MFM controller and with an RLL drive use only an RLL controller, as doing otherwise can cause intermittent problems and/or loss of data!

Another type of encoding used on hard drives is ESDI, which stands for Enhanced Small Device Interface, which was created to allow for larger data capacity and speed in hard drives, as MFM and RLL drives are limited to under 140 megabytes due to their design. ESDI is really more of a standard of control rather than an encoding method, as ESDI was designed as an "intelligent" controller that could potentially handle not only hard drives, but floppy drives, tape backups, etc. However, I have never seen ESDI used for anything other than hard drives.

Another type of encoding used on hard drives is SCSI, which stands for Small Computer Systems Interface. SCSI is also really more of a standard of control rather

(See Page 21)

HARD DRIVES—

(Continued from Page 20)

than an encoding method, as SCSI is an "intelligent" control method used not only to control hard drives but other devices such as tape drives, floppy drives, CD-ROMs, printers, etc. Up to seven devices can be attached to a SCSI controller as long as each device has an individual ID number. SCSI drives usually come in large capacities and are often low-level formatted at the factory or come with low-level formatting software.

Nowadays the most common type hard drive is IDE, which stands for Imbedded Drive Electronics. On IDE drives the controller is actually built onto the drive itself; thus, only a sort of simple interface card is used to interface the drive into the computer databus. IDE drives are low-level formatted at the factory and should *never* be formatted by the user. Attempting to low-level format an IDE drive will probably destroy it! IDE drives only need to be partitioned and high-level formatted.

Some of the terms that you might run across with hard drives are as follows:

INTERLEAVE — Since the drive continuously spins at 3600 RPM and the drive head reads the data off a platter at that speed it is possible that your computer or controller card cannot accept data from the hard drive as fast as the hard disk is reading it. Since the platter cannot be slowed down, the sectors on the hard disk are spread out differently, so that the head will physically pass over a specified number of sectors before it reads more data where that sector is what the drive perceives as actually being the next sector, but in actuality it could be physically several sectors down the platter. In skipping several sectors, this method gives the computer time to catch up before more data is thrown into it. So a 1 or 3 interleave would be sequential sectors set up every 3 physical sectors. A 1-to-1 interleave would be the best case, where each sector is actually set up the same as what is physically there.

Only on those hard drives that can be low-level formatted can an interleave be set up, thus IDE drives cannot have an interleave set as they are factory set at 1 to 1. If an interleave is set wrong, it will only slow down your drive as, if the next sector comes around too soon and the computer is not ready for it, it will skip that sector and wait until that sector comes around again, which takes time.

TRACK — A track is concentric circles that hold data on a hard disk platter with a track being composed of, not physical tracks, but magnetically allocated areas on the hard disk.

SECTOR — A section of one track is called a sector; usually 512K bytes of data are contained per sector.

ZONE BIT RECORDING — Since the tracks toward the outside of the disk are longer than those toward the center of the disk, due to the disk circular design, a lot of space toward the outside is wasted. Thus, a recording method called Zone Bit Recording (ZBR) was created to which tracks toward the outside of the platter have more sectors per track than the inside of the platter, but each sector still contains only 512K bytes of data.

CYLINDER — A cylinder is basically the combination of identical track numbers on each platter of the disk of which each platter of a disk is stacked on top of each other; thus, just imagine a cylinder as being a rod vertically through each track on the platters. The usefulness of a cylinder is that it is one of the components used to find an address on the hard disk, with the other components being the head number and sector number.

CLUSTER — A cluster is basically the number of sectors that the operating system uses each time space on the hard disk is needed.

TRACK 0 — Track 0 on a hard drive is very important, as when the drive is first powered up it goes to track 0 to recalibrate the position of the heads, as that is the only position where the drive can physically tell where the head is located. Also at track 0 the file allocation table (FAT) resides. The FAT contains information about the location of files and bad tracks.

In closing, the encoding on the hard drive determines the drive's size in megabytes as well as its speed and efficiency. The cylinders, heads and sectors make up the scheme in locating addresses for data on that drive. A combination of all of the above and more is what is actually involved in the workings of a hard drive.

I hope that this gives you some insight into the interworkings of a hard drive.

Western Horizon announces product availability

Several new products are available from Western Horizon Technologies, according to Don O'Neil of the company.

An AT-Keyboard and ROM Upgrade for the TI99/4A is available for $65. The upgrade does not include the keyboard.

According to O'Neil, the upgrade allows the use of any standard AT keyboard on a 99/4A in conjunction with an existing console keyboard. It does not replace the old keyboard like other upgrades, he emphasizes. Both manual and auto-switch keyboards can be used.

O'Neil says it provides an expanded and partially optimized new version of the TI operating system to scan both keyboards in the system simultaneously. The new ROMs are expandable to 64K for future enhancements to the operating system.

Also included are "bug fixes" for Advanced Video Processor Cards owners with RS232 problems, and "bug fixes" for other 80-column cards. According to the company, the product is compatible with all existing 80-column cards (TIM, Mech

(See Page 22)

WESTERN HORIZON—

(Continued from Page 21)

AVPC) and 98 percent of all current software (list of incompatible software is provided), and the user has the option to use the existing console keyboard if a program will not run with the new one.

The unit also provides an upgrade path for 64K of 0-wait state RAM for future enhancements or programs. (This RAM does not function as standard 32K memory.)

The interface installs inside the console, is "clipped" on to the 9900 and requires only one solder joint for installation. For another $20 plus shipping, WHT will install the interface.

O'Neil notes that the company stocks many kinds of AT 101 keyboards starting at $20.

Scheduled for release in late 1994 (pending completion and 20 pre-sale orders) is the latest version of Rich Gilbertson's RXB, Version 6.0.

"Rich has painstakingly hand entered all the original TI Extended BASIC source code from printouts he acquired," O'Neil says, "then carefully proceeding, debugged the entered code to generate a complete set of commented source codes for TI's XB." According to O'Neil, Gilbertson has cleaned up and sped up much of TI's XB code, and because of the larger space available in the WHT module and GRAM emulator devices (32K GROM as compared to XB's original 24K) was able to add many new features found in other versions of Extended BASIC, only this time, seamlessly. Besides speed, O'Neil says, Gilbertson hopes to include full 80-column support, a new editor and other features. RXB 6.0 also features 100 percent TI XB compatibility.

RXB 5.58 is also available in diskette format for GRAM emulator owners (GRAM Kracker, P-GRAM, Gramulator, Geneve, etc.) from CaDD Electronics,.81 Prescott Rd., Raymond, NH 03077, (603) 895-0119.

The company is now carrying XBIII by Winfried Winkler through special arrangement with the author. This new XB from Germany is the same XB that Asgard once carried. According to O'Neil, this version has many new features to allow a user more control over hardware, new commands, 25-50 percent faster program execution, and 100 percent TI XB and BASIC compatibility.

Shipment of XB-III is also pending 20 pre-sale orders.

TURBO Video at $25.00 is an upgrade for the Myarc Geneve 9640 said to speed video display up to 25 percent. It is a drop-in replacement for Myarc's original design.

The company plans to release 4a Memex Jr, an offshoot product from the original planned 4a Memex card, which it says will works with a Geneve 9640 or 99/4a to provide up to 4 megabytes of RAM available for programs. (MDOS 2.0 currently only supports 2 megabytes.)

Upgradeable in 512K increments for $25 per 512K, this card is designed to replace any other memory card on a 99/4a, or add to Geneve memory. MDOS-like access on the 99/4a is documented for C and FORTRAN programmers to access the memory instantly, according to the company.

Version 1.0 of the DSR provides a power-up for automatic configuration of Geneve or 99/4a usage, and provides a set of DSR links for banking in and out memory, according to the company. Future revisions of the DSR may contain RAMBO compatibility, an advanced memory manager, and other features (pending developer input).

Estimated ship date is late 1994.

Western Horizon Technology is also developing a Super VGA Video Interface designed to replace old VDP processors on both 99/4A and Myarc Geneve systems through hardware modifications as a P-Box peripheral said to all the existing software for 80-column cards, as well as providing a true VGA-compatible video output

for VGA and SVGA monitors and providing video resolutions of up to 1280x1024 pixels (with 1 megabyte or 2 megabytes memory) and up to 16.8 million colors (in 640x480 mode with 1 megabyte memory). This new video card will work with existing RGB monitors (composite output is not available from this card; an external VGA to video adapter must be used) in resolutions up to 640x480 in 16.8 million colors, according to the manufacturer, including all the standard Geneve and 80-column video modes. (Modes above 640x480 require an SVGA monitor.) The SVGA Video Interface comes standard with 704k dedicated video memory, and is expandable to 1216K and 2240K memory. Estimated ship date is late 1994.

For further information, contact Western Horizon Technologies, 3297 Woody Lane, San Jose, CA 95132 (408)-934-0352, or E-Mail doneil@delphi.com

Hardware project

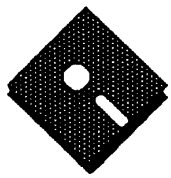
Have an AT keyboard? Here's a project to consider

By FRANK FRANKENBERGER

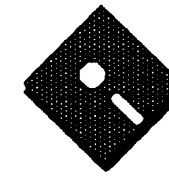
This article originally appeared in the Computer Voice, the newsletter of the Southern California Computer Group.—Ed.

This simple project is neither new nor original. However, with the advent of Don O'Neil's AT keyboard interface and ROM upgrade, it may come in handy for those who use the AT 101 keyboard and want to store the TI console back out of the way. With this project you can either set the Widget within easy reach, or you can change your current single cartridge with

(See Page 24)



MICROpendium DISK SALE



If you've been waiting for a sale on MICROpendium program disks, this is it! For a limited time (through Nov.1, 1994) Series 1-7 disks are available for a special price. (Series 7 disks are mailed monthly starting with the October 1994 edition, programs from April 1994 through October 1994 will be mailed as soon as the order is placed.)

MICROpendium disks

SERIES #	REGULAR PRICE	SALE PRICE	YOU SAVE	DISCOUNT
Series 1 (Apr. '88-Mar. '89)	$25.00	**$15.00**	*$10.00*	*40%*
Series 2 (Apr. '89-Mar. '90)	$25.00	**$15.00**	*$10.00*	*40%*
Series 3 (Apr. '90-Mar. '91)	$25.00	**$15.00**	*$10.00*	*40%*
Series 4 (Apr. '91-Mar. '92)	$25.00	**$15.00**	*$10.00*	*40%*
Series 5 (Apr. 92-Mar. 93)	$25.00	**$15.00**	*$10.00*	*40%*
Series 6 (Apr. 93-Mar. 94)	$25.00	**$15.00**	*$10.00*	40%
Series 7 (Apr. 94-Mar. 95)	$40.00	**$25.00**	*$15.00*	38%

(Circle the items you want to order)

ACT NOW — THIS SALE ENDS NOV. 30, 1994

Customer information

Name _____

Address _____

City _____

State _____

ZIP _____

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

Total Remitted _____

(U.S. Funds only; Texas residents add 7.75% sales tax)

Disk shipping information

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

FOR CREDIT CARD ORDERS

Credit Card No. _____

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

Signature _____

(credit card orders only)

EXTENSION—

(Continued from Page 23)
out the need of digging out the TI console. This idea was taken from an article originally written by Arthur Hazboun of Harbor City, California, and published in the February 1988 issue of MICROpendium. However, the 1988 project was not designed with the AT keyboard in mind.

MATERIALS

1 — old double-sided contact cartridge (Munchman, Microsurgeon, etc.)

1 — Old cartridge case

1 — 36-pin PC Mount connector

3 to 6 feet of 36-wire ribbon cable

Low-wattage soldering iron, flux and solder

You can substitute a 36-pin card edge connector (female) in place of the 36-pin PC Mount connector, if you can find one.

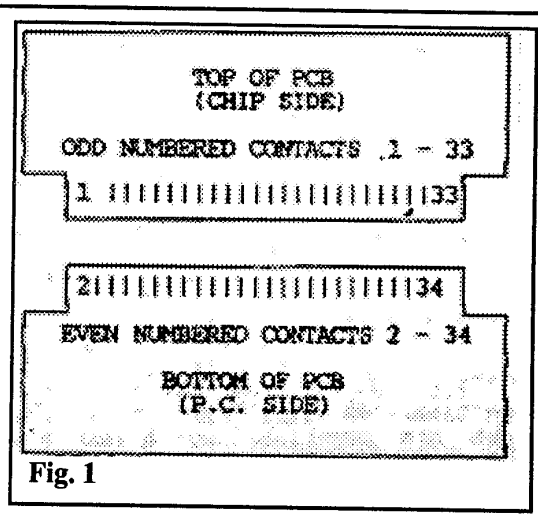


Fig. 1

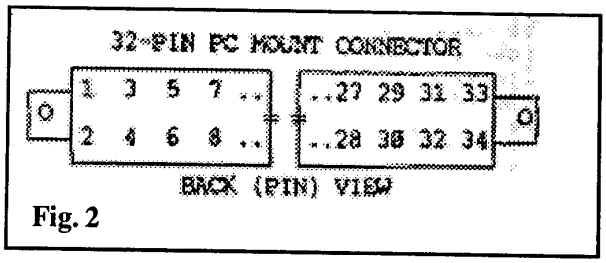


Fig. 2

Open the cartridge case with the dual contacts and remove the printed circuit board (PCB). Cut across all the traces at the back of the contact area so that the rest of the components on the board are separated from the card edge contacts. A small Dremel-type grinder will work perfectly for this. However, you can use an Xacto knife or single-edge razor blade to cut the traces. Removal of the chips is not necessary, unless you want to make sure you have plenty of room to run the cable through the hinged area of the cartridge case.

Cut a slot in the middle of the hinged section of the cartridge case wide and high enough for the 36-wire ribbon cable to pass through.

Separate the wires of the cable at one end back far enough so that half of the wires will fit over the PCB with enough room to solder the ends to the card edge contacts (2¼ inches). Strip back about one-quarter inch of insulation from each wire and solder them to the card edge contacts, away from where it plugs into the GROM port.

The contact order for the PCB is shown in Fig. 1

On most ribbon cables, wire No. 1 is a different color from the remaining wires to make it easily identifiable. The rest of the wires are numbered in order, 2-3-4-5-6, etc. Carefully solder the wires of the ribbon cable to the PCB board contacts. Make sure to solder them in their correct order: 1-1, 2-2, 3-3, etc. When you've finished soldering, check your work to ensure that no solder has flowed onto any adjacent contact.

Now you can place the PCB back into the cartridge case, with the ribbon cable running out the back of the box.

It is now time to separate the wires at the other end of the ribbon cable, about 1½ inches and strip back about one-quarter inch so you can solder the other end of the cable to the 36-pin PC Mount connector. The pins on the connector should be numbered. However, if yours are not, the diagram in Fig. 2 should be followed.

On the old cartridge case, remove and discard the unusable PC board, dust cover and spring. Cut a slot in the back of the cartridge for the 36-wire ribbon cable to pass through.

TEST THE CABLE

If you have checked your work and everything is done properly, then you should test the cable in your system *before* you install the PC Mount connector into the empty cartridge case. If the extension cable is working properly, you can install the PC Mount connector into the empty cartridge case with hot glue.

The connector should extend out of the front of the empty shell as far as possible. You may want to plug in a cartridge to see how far out it needs to extend. After you have it properly positioned, you need to use the hot glue all around the inside — even over the wires. After the hot glue has set thoroughly, you can assemble the top of the cartridge and hot-glue in the holes. When all is set and hard you have a very strong and handy GROM extension.

One last detail is to label your cartridge cases. Do this so that if someone else spots it they will know what it is (maybe). GROM Extension is one label suggestion.

If you don't plan to use the TI console at all, or you can't find a 36-pin connector, you may consider removing the L-shaped 36-pin connector from the motherboard and use it instead. If you do this you will need to cut the PCB down so it is about half its original size. This is to ensure that it will fit inside the console.

MICRO-REVIEWS

The TI Educational World of Tony Falco and Adventure Compendium II

By CHARLES GOOD

In the case of the two products reviewed this month I am deviating from my usual practice of offering to directly send my readers public domain or shareware software I review. The following two collections of public domain and shareware software were compiled by and are sold by the M.U.N.C.H. user group as a fundraiser project. As librarian of the enormous Lima User Group software library I am familiar with lots of the public domain 99/4A software that has been around for many years. Although much of what I describe below is older software, some dating from 1982, most of it is new to me, meaning I haven't seen it before. This means that the software is probably *not* in your personal or user group library, and old software that hasn't been seen before is usually just as interesting to a user as new software. Send your money for what I describe here to M.U.N.C.H. c/o James Cox, 905 Edgebrook Dr., Boylston MA 01505. If you want to talk to me or send me software to review my evening phone is 419-667-3131, my internet email address is cgood@lima.ohio-state.edu and my post office address is P.O. Box 647 Venedocia OH.

The TI Educational World of Tony Falco

Tony Falco is a high school teacher who has been with the TI community for a long time. He had a couple of his Extended BASIC programs published in *99er Magazine* way back when and still sometimes attends meetings of M.U.N.C.H. user group, although these days he writes software for other computers besides the 99/4A. For the benefit of his students and his children he has written lots of XB software, mostly of an educational nature. His programs are full of color graphics sprites speech and music, making full use of all the TI's bells and whistles. The technical quality and interesting design of his XB programs are comparable to what Jim Peterson used to produce. I am amazed that we TIers aren't more familiar with this talented individual.

M.U.N.C.H.'s entire Tony Falco collection consists of 5 SSSD disks and 94 files. They are asking $8.95 plus $3 postage for the complete set. Topics include basic math, algebra, foreign language drill, writing music, spelling, health, basic computer, geography, drawing, and just for fun games. Each disk comes with an XB LOAD that allows you to run everything on the disk from a menu. It would take too much space to describe all the programs individually, so I will highlight some that I find particularly interesting.

CRAYON-BOX. This is a drawing program that doesn't require joysticks. By moving the cursor with keyboard keys you leave behind blocks of color 8x8 pixels. These color blocks can each be any of 11 colors and any of 8 shapes (squares, rectangles, triangles, solid or alternating pixels, etc.). Kids can make some really colorful interesting designs, or you can play with the 11 pictures that come on the disk with Crayon-Box. Probably the most surprising part of this program is its screen dump. Press "P" and you are in for a surprise if you don't know what to expect. Your printer grinds on and on and on as your picture is automatically printed six different ways before the printer stops. Some are large, some are small. Some have a white background and some have a black background. Some are vertical and some are horizontal on the printer paper. Each of the screen colors is represented by a different dot density.

WORD-WORLD, ARMY-WORLD, POLICE-WORLD. These are all similar spelling games for early elementary school kids and remind me of TI's Story Machine cartridge. You type a word from the game's list and a sprite appears on screen. Sometimes you can then type another word like LEFT or FLY and specify movement of the sprite. If you type something the game doesn't recognize you are given unlimited additional opportunities to get things right. Eventually the screen fills with all these sprites walking, driving, flying, sailing, etc. Sound effects used with each sprite are well done. For example, for the different kinds of mobile pictures such as cars and planes you hear their motors rev up as they start moving, and each motor sounds different. Word-World was originally published in *99er*.

USA-MAP gives you a rather blocky map of the United States and asks you to identify the state indicated. If you guess the wrong state or type nonsense the correct answer is given. If you guess the correct state, but misspell the name, the computer tells you to try again and spell the state's name correctly this time. The program recognizes the difference between "correct" but misspelled answers and incorrect answers.

HEALTHEXAM. On the basis of on a 1982 Blue Cross/Blue Shield questionnaire, this program asks you a long list of multiple choice questions. The usual overweight, family history, and smoking questions are included as well as a bunch of questions designed to gauge your emotional health. Do you live a stressful life? At the end you are given a numerical score and short written evaluation. I like this program because it says I am in "excellent health." Computers, being very exact machines, always tell the truth.

Adventure Compendium II

(See Page 26)

MICRO-REVIEWS—

(Continued from Page 25)

This comes archived on two DSSD disks (not DSDD as stated in the July 94 issue of MICROpendium). The Geneve and 99/4A archiver programs are included so you can unpack the disks. Cost is $6.95 plus $2 postage. Included are adventures that require the XB, Tunnels of Doom, and Adventure modules. Everything is in English, but some of these adventures were written by German TI users and are probably new to most of us in the states.

DUNE for the TOD module. *Dune* was a popular science fiction book. It is now a very popular game for IBM and maybe also MAC PCs. As evidence of its popularity, there is a DUNE newsgroup on the internet. My kids spend hours and hours playing Dune on our home PC and the soundblaster sound effects drive me nuts. Now there is a version for the 99/4A, and at least there is no sound. Dune is a planet with a desert environment that contains the most valuable substance in the universe, a particular kind of spice. Two groups are each trying to harvest the maximum amount of this limited resource. You are the Atreides and the enemy are the Harkonnens, just like the book. You meet all kinds of creatures and weapons including sand spinners, sand splitters, sand throwers, sand jumpers, sand ghosts, sand dragons, etc.

GARFIELD for the TOD module. Garfield, comic strip cat, has lost his toys and has to hunt through the basements in the neighborhood to find them. The neighborhood pets and pests want to keep the toys for themselves, but Garfield has a group of friends to help get them back.

NINJA for the TOD module. A party of adventurers must enter the Ninja temple and retrieve gems within. Also, they need to find the Ninja's golden throwing star. Getting out of the temple alive before the Ninja destroys his gems is not easy.

The following eight adventures each require the Adventure module: Escape from Alcatraz, Bigfoot, Escape from Cannibal Island, The Mystery of Cap' Kidd, The Great White North Adventure (save the world from contaminated beer), Moon Adventure, Nessy (Loch Ness of course), and Travelling (you are a passenger on a jet with no pilot). Each of these adventures is new, to me. None of them are my user group's software library.

Among the adventures that run from XB is ALADDIN. This is a very complex game that runs similarly to an Adventure module game, with no graphics. Lots of files load into memory at the appropriate time. The story pretty much follows the fairy tale. You have to rescue the beautiful princess from the Sultan's palace. It helps if you find and figure out how to use the magic lamp.

You also get Carfax Abbey, an old favorite of mine that has been enhanced so it loads faster. The game hasn't changed, but it now takes tens of seconds rather than minutes to load Carfax. There is also "Visitor From Outer Space" and "Nasty." Nasty isn't really an adventure so I don't know why it is included in the package. Nasty seems to take over your computer and delivers humorous on-screen messages and altered color bar and first menu screens. The adventure I guess is trying to figure out how to get out of Nasty and back into regular old Console or Extended BASIC without shutting down your system.

Last but not least, Adventure Compendium II includes a group of fast Infocom game loaders. They are all much faster than the original and are all generic loaders. This means that the same loader can be used to load all the Infocom games. Just put your GAME1 and GAME2 files for the particular game you want to play in DSK1. Your game will start in 10-20 seconds. The original Infocom loaders took more than a minute to start a game. There is an XB loader and an EA5 loader for 40 column systems. These both have a nice lowercase character set. You also get two different 80-column loaders. One will work with any 80-column TI system and the other will run directly out of MDOS on a Geneve.

BUGS & BYTES

OPA said prioritizing TIMs

According to BUG-Bytes, a publication of the Brisbane Users Group in Australia, OPA (Oasis Pensive Abacutors) of Canada has stated that filling outstanding orders will be part of its main short-term goal in the TI arena, to rebuild its credibility and reputation. Manufacture of TIMs will be a priority, according to this report.

OPA has recently moved to a new building, according to the report, with two floors for manufacture and research and development and another for retailing.

Ink jet, bubble jet printers worth a look for TI users

By JAN PERKEN

The following article appeared in the newsletter of the Cleveland Area User Group, and others.—Ed.

Is ink jet is a new technology?

No! It is approximately 40 years old. I personally worked on a small conveyor for Eveready, the battery company, to date-code watch batteries. The technology we used was ink jet. The dispensing unit was made by the AB Dick Company. It cost big bucks and was messy.

That was about 15 years ago. IBM commercially produced jet-style printers 25 years ago. These printers needed special paper. They were slow and prone to clogging. New ink jets use replaceable print heads with a new supply of ink each time the old supply runs out. A new print head for the HP Desk Jet is about $22. Ink refills are available that can be used to refill the head about twice, giving about 2,000 copies with each refill. The refill costs $10.

How does it work? A small droplet of ink is forced out of a very small tube. This takes place many times a second and the printers have some 50 to 60 jets.

Ink jets in the future will not replace

(See Page 27)

INK JET—

(Continued from Page 26)

laser printers. The Canon bubble jet has a resolution of 300 dots per inch. The Hewlett-Packard Desk Jet 500 has a resolution of 300 dots per inch. Both use ink jet technology. On the other hand, new laser printers can support 1,200 dots per inch resolution. For graphics, laser printers give superior gray shading. However, ink jets excel at printing text. Their graphic output is also good but not up to the standards of a good laser printer. The ink jet handles paper differently than a laser printer. The laser hardware is also much more expensive. Color ink jet printers are also available at affordable prices while color laser printers are very expensive.

I was able to borrow an HP Desk Jet 500 printer to play with. This printer weighs 14 pounds. It has a tray to hold paper and stacks copies on a top of the supply tray.

Santa, and the finance company, plus some discount coupons and a company discount, helped purchase a Canon Bubble Jet 230, the wide version of the BJ-200. It was purchased for Computer Aided Design. Even though the BJ-230 prints on 11x17-inch paper, it is small. My wife observed, "It's so small, and so expensive!" It is also light, weighing about seven pounds. The paper tray holds 100 sheets and is angled into the printer. The paper is bent less by the Canon than by the HP. So, you ask, who cares? Remember, labels tend to peel when bent around a small diameter, though I have yet to figure out how to print labels on the ink jet printer without wasting a sheet. The answer will come in time.

The HP came with a good manual. Newer ones do not, and the manual needs to be purchased separately. You can't keep cutting costs without cutting out something. The HP Desk Jet 500 sells for about $250. The manual is important if you plan to use control codes to print. It also explains PCL (Printer Control Language). This language is standard to laser copiers.

Cartridges can be added to the HP to emulate Epson FX printers. However, I didn't get a chance to try this with TI graphics, though I don't see why it wouldn't work. The cartridge costs about $60. Also, font and memory cartridges are available. A driver disk for users of Microsoft Windows is included.

The bubble jet cam with a manual, which provides much information but would be better with more examples. The Canon BJ emulates an IBM X24E and an Epson LQ 510. However, few examples are given. An Epson LQ manual would be handy. The BJ doesn't use PCL, but it comes with a drivers for Windows users that address the printer as a Canon bubble jet.

I printed some TIPS graphics and they came out higher on the page than they should have. That's a problem caused by using a 24-pin printer with 9-pin graphics. What I need to do is to get it to emulate a 9-pin printer. (After finishing this article I found the line spacing to be the problem when printing graphics. A 24-pin printer defines the 9-pin increment as a different value. For example, ESC A is n/72 in 9-pin and ESC A is n/60 in 24-pin.)

You can get the BJ-200 for about $250. Both HP and Canon offer telephone help lines and automated fax help lines. Canon also offers a BBS. I've seen some reviews in which the Canon BJ printer outscored the HP Desk Jet. Canon's newer design may have given it an edge. Both printers offer useful features and both work with TI or IBM computers. However, without the emulation cartridge, the HP won't work with Epson codes when using TI-Writer. But a transliteration file can fix that. BASIC code can be written without much trouble. PCL codes are another matter. The Canon in Epson emulation takes very few changes for text when using a TI. Graphics are another matter about which I'm not sure. Perhaps someone with more expertise in graphics could offer some advice. Both support standard Centronics cabling, so there's no problem plugging them into a TI system.

Either printer would be a good addition to a TI or PC system.

USER NOTES

Saving time with System III conversions

This comes from Bill Gaskill of Grand Junction, Colorado. He writes:

If you have ever used Irv Crowley's System III Checkbook and Budget Manager you probably know that it comes in a variety of versions depending upon the capacity of your floppy disk drives. Recently I decided to take my DSSD version and convert it from DSK1 and DSK2 to DSK3 and DSK4 for use on my new Horizon RAMdisk.

The task of doing so proved to be quite time-consuming because of the number of programs that make up System III. Because of this, I decided to write down the line numbers in each program where DSK access code is located, in case I ever wanted to change things back, or alter the native code in any other way. After doing so, the idea that others might be able to save some time by knowing where the DSK access code is came to mind, so here it is.

PROGRAM	LINE
PART1	230
	390
	520
	900
PART2	230
	430
	3380
PART3	230
	420
	520
	2030
PART4	230
	420
	1320
	1500
	1510
	1520
PART5	240
	400
	690
PART6	230
	380

(See Page 28)

USER NOTES

(Continued from Page 27)

	810		860
	820		870
	1080		890
	1090		900
	1100		910
	1110		920
PARTA	230	PARTE	230
	240		400
	440	PARTF	230
	700		390
PARTB	230		660
	380		710
	730	PARTG	230
	750		1470
	1010	PARTH	230
	1200		400
	1220	PARTI	240
PARTC	230		390
	390		400
	850		410
		PARTJ	280
			410
			480
		PARTK	240
			410
			690
			790
			1030
			1300
		PARTL	250
			390
			880
			1020
		PARTM	240
			250
			270
			460
			730
		PARTN	240
			480
		PARTO	250
			420
			990
			1020
		PARTP	240
			360
		PARTQ	250
			410
			740
			750
		PARTR	280
			360
		PARTS	240
			650
		PARTT	210

Correction to XSNDTBL

Sometimes the left hand doesn't know what the right hand is doing. We had the following update to the XSNDTBL, a part of last month's article entitled Experiment with Sound on page 6. We had the correction in-house but it never crossed paths with the article. The correction goes at the beginning of the Sound Tables sidebar on page 7. Here it is. We apologize for the inconvenience.

```
--------------------------------
* TONE-GENERATOR: (2 bytes) ***
Binary      Hex        Dec      Fcode
--------------------------------
GT1; (Byte 1)                   (Lb)
 10000000   >80         128      0
 1000xxxx   >81-F     129-142   1-14     (change 81-F
 10001111   >8F         143      15      to 81-E)
GT1,2,3     (Byte 2)            (Hb)
 00000000   >00          0       0
 00xxxxxx   >1-3E      1-62     1-62
 00111111   >3F         63       63
--------------------------------
GT2;        (Byte 1)            (Lb)
 10100000   >A0         160     same
 1010xxxx   >A1-F     161-174    as      (change A1-F
 10101111   >AF         175     GT1     to A1-E)
                      (Byte 2)  (Hb)
--------------------------------
GT3;        (Byte 1)            (Lb)
 11000000   >C0         192     same
 1100xxxx   >C1-F     193-206    as      (change C1-F
 11001111   >CF         207     GT1     to C1-E)
                      (Byte 2)  (Hb)
--------------------------------
```

Curing corrupted RAMdisk files

The following was written by Dick Warburton and has appeared in several user group newsletters.

Here is a tip for those with a RAMdisk which may be helpful at some time.

I have often accidentally corrupted files on the RAMdisk, usually by using DM1000 or switching the RAMdisk off. When the files get corrupted, they often overwrite the size of the file. One suddenly is faced with gibberish on the screen with a file size of 8844 kbytes.

I know from bitter experience that if I simply delete the file, it usually destroys the contents of the whole RAMdisk. This

(See Page 29)

USER NOTES

(Continued from Page 28)

is because the apparent file size is greater than the capacity of the RAMdisk.

I recently found a simple solution. Like Archimedes in the bath, I suddenly saw the light. If I simply rename the file as %%%, it automatically becomes the last file where the headers are located on sector 1. Using Disk Patch in Funnelweb, I simply delete the last header. Problem solved. No file, no corruption. It's a pity we couldn't apply the same logic to politicians.

Another way to line up decimals

The following is from Leonard Taffs, of Tucson, Arizona. He writes:

My eye caught the User Note "Lining up decimals" in the August 1994 edition. It was timely to come across the article because my work as new treasurer for a local non-profit organization has required my going over all the group's records. For this I needed a printout of the entries. Having a program that aligns the amounts properly saves a lot of wear and tear on the eyes.

Tabbing can be achieved to line up decimals properly with the statement: TAB(C-POS(A$,".",1)), where C is the desired column. There is a problem with the decimal program program — it does not align all amounts, even if the &MISSING 5 data items are added. I solved the dilemma inadvertently by choosing to have entries first entered as strings (which later are converted to numeric values).

I am incluing two versions of a short routine that line up decimals and also add zeroes when TI drops them. One, called POS:NUMER, is for numerical input and the other, very similar but more versatile for development of options, is called POS:STR$.

I have written a longer, crash-proof version which adds a printer routine, a forwarding balance option (for use with check registers or bank statements (deposits are entered with a minus sign). It will probably be in Southwest99ers October newsletter — it will definitely be on our Disk of the Month for that month as DECJUSTIFY. It is a considerable improvement of my earlier attempt to modify Earl Raguse's miniature version printed in a long-ago LA99ers newsletter, which did not always align properly and was prone to crashes if the user forgot to avoid illegal inputs. This earlier version appeared on the SW99ers DOM for July 1994 as JSTFY/DEC.

For kicks I have added a little program (GOSUBTNSHN) that users can add to their programs for any kind of waiting portion of their program, or use as an attention getter, etc.

POS:NUMER

```
1 REM [POS:NUMER] 8-30-94
     SUMMARY BY W.L.TAFFS
  SW99ers TUCSON,AZ.85711 !0
32
100 CALL CLEAR !209
110 C=15 !049
120 INPUT "(NUMR)
          AMT: ":A !255
130 A$=STR$(A)!172
140 L=LEN(A$)!177
150 P=POS(A$,".",1)!015
160 IF P THEN 170 ELSE A$=A$
&".00" :: GOTO 210 !179
170 A2$=SEG$(A$,P,L)!001
180 A2=LEN(A2$)!010
190 IF A2=2 THEN A$=A$&"0" !
152
200 IF A2$="." THEN A$=A$&"0
0" !232
210 PRINT TAB(C-POS(A$,".",1
));A$ !036
220 GOTO 120 !199
```

POS:STR$

```
1 REM [POS:STR$ ] 8-30-94
  STRING VERS. OF POS:NUMER
    SUMMARY BY W.L.TAFFS
  SW99ers TUCSON,AZ.85711 !0
35
2 !!131
3 REM ENTERING NUMBER AS A
      STRING ENABLES USE OF
      ALPHA CHARACTERS TO
      DIRECT MENU OPTIONS !0
76
4 !!131
5 REM SUCH AS ADDING PRINTER
      ROUTINES, ETC., OR
      INCORPORATING THIS
      ROUTINE IN A LARGER
      PROGRAM. !096
6 !!131
100 CALL CLEAR !209
110 C=15 !049
120 INPUT "(STR$)
          AMT: ":A$ !254
130 !!131
140 REM PROGRAM OPTIONS CAN
       BE INSERTED HERE !12
1
150 !!131
200 L=LEN(A$)!177
210 P=POS(A$,".",1)!015
220 IF P THEN 230 ELSE A$=A$
&".00" :: GOTO 270 !044
230 A2$=SEG$(A$,P,L)!001
```

(See Page 30)

READER TO READER

❑ Merle Vogt, 14350 IH-35 S., Von Ormy, TX 78073, writes:

Back in the late '80s there was a disk producer named QS, Quality Software. They put out a number of interesting items, such as QS-RAMdisk (for Foundation card), QS-Banner Maker, QS Disk Labeler, QS Writer, QS Sideways, QS Convertor, Disk Manager IV, Quick Copier II and others. All these disks had some sort of extremely obscure protection scheme and could not be copied, or fixed when they did not work.

Could you find anyone out in the TI 4A world who could tell us how to crack this stuff? Some of the programs our user group has either don't run or only run sometimes.

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

USER NOTES

(Continued from Page 29)

```
240 A2=LEN(A2$)!010
250 IF A2=2 THEN A$=A$&"0" !
152
260 IF A2$="." THEN A$=A$&"0
0" !232
270 PRINT TAB(C-POS(A$,".",1
));A$ !036
280 GOTO 120 !199
```

GOSUBTNSHN

```
1 REM [GOSUBTNSHN] 8-30-94 G
OSUB VERSION OF EXPERIMENT A
 WAY TO MAKE A WINDOW (IN PR
OGRESS) !036
10 GOSUB 30000 ! This line i
n program's place you need !
255
100 INPUT "READY FOR PROGRAM
!":K$ !100
30000 REM [WINDOWSHIN] 8-30-
94 EXPERIMENT A WAY TO MAKE
A WINDOW (IN PROGRESS) !136
30030 A$=RPT$(".",1)!020
30040 CALL CLEAR !209
30050 X2=28 !124
30060 A1$=CHR$(46)!203
30070 X=X+1 !041
30080 CALL HCHAR(1,3,ASC("."
),27)!190
30100 FOR B=2 TO 24 !104
30110 DISPLAY AT(B,1):A$ !00
8
30120 NEXT B !216
30130 CALL HCHAR(24,3,ASC(".
"),27)!244
30140 FOR C=1 TO 24 !104
30150 DISPLAY AT(X2-1,28):A$
 !070
30160 X2=X2-1 !142
30170 NEXT C !217
30180 DISPLAY AT(12,3):" YOU
R ATTENTION PLEASE!  ." !031
30190 FOR A=1 TO 150 :: NEXT
 A !240
30200 CALL KEY(0,K,S):: IF S
<>1 THEN 30240 !156
30210 IF K=13 THEN CALL CLEA
R :: GOTO 100 !113
30220 CALL KEY(0,K,S):: IF S
<>1 THEN 30220 !136
30240 GOTO 30040 !028
30250 RETURN !136
```

Statement of Ownership, Management and Circulation
(Required by 39 U.S.C. 3685)

1A. Title of Publication	1B. PUBLICATION NO.	2. Date of Filing
MICROpendium	1 0 4 3 2 2 9 9	10/15/94
3. Frequency of Issue: monthly	3A. No. of Issues Published Annually: 12	3B. Annual Subscription Price: $35/yr

4. Complete Mailing Address of Known Office of Publication (*Street, City, County, State and ZIP+4 Code*) (*Not printers*)
502 Windsor Rd., Round Rock, Willlamson, TX 78664-7639

5. Complete Mailing Address of the Headquarters of General Business Offices of the Publisher (*Not printer*)
P.O. Box 1343, Round Rock, TX 78680-1343

6. Full Names and Complete Mailing Address of Publisher, Editor, and Managing Editor (*This item MUST NOT be blank*)

Publisher (*Name and Complete Mailing Address*)
John Koloen, P.O. Box 1343, Round Rock, TX 78680-1343

Editor (*Name and Complete Mailing Address*)
Laura Burns, P.O. Box 1343, Round Rock, TX 78680-1343

Managing Editor (*Name and Complete Mailing Address*)

7. Owner (*If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.*) (*Item must be completed.*)

Full Name	Complete Mailing Address
Burns-Koloen Communications, Inc.	P.O. Box 1343, Round Rock, TX 78680-1343
John Koloen	P.O. Box 1343, Round Rock, TX 78680-1343
Laura Burns	P.O. Box 1343, Round Rock, TX 78680-1343

8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities (*If there are none, so state*)

Full Name	Complete Mailing Address

9. For Completion by Nonprofit Organizations Authorized To Mail at Special Rates (*DMM Section 424.12 only*)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (*Check one*)

(1) ☐ Has Not Changed During Preceding 12 Months
(2) ☐ Has Changed During Preceding 12 Months
(*If changed, publisher must submit explanation of change with this statement.*)

10. Extent and Nature of Circulation (*See instructions on reverse side*)	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies (*Net Press Run*)	1908	1700
B. Paid and/or Requested Circulation 1. Sales through dealers and carriers, street vendors and counter sales	277	260
2. Mail Subscription (*Paid and/or requested*)	1099	962
C. Total Paid and/or Requested Circulation (*Sum of 10B1 and 10B2*)	1376	1222
D. Free Distribution by Mail, Carrier or Other Means Samples, Complimentary, and Other Free Copies	63	53
E. Total Distribution (*Sum of C and D*)	1447	1275
F. Copies Not Distributed 1. Office use, left over, unaccounted, spoiled after printing	469	425
2. Return from News Agents	0	0
G. TOTAL (*Sum of E, F1 and 2 - should equal net press run shown in A*)	1908	1700

11. I certify that the statements made by me above are correct and complete

Signature and Title of Editor, Publisher, Business Manager, or Owner
Laura Burns 10/15/94

PS Form 3526, January 1991 (See instructions on reverse)

DSK.DISK.FILENAME

The following was written by Jan Alexandersson of the Swedish User Group Programbiten. It has appeared in user group newsletters.

It is possible to call a file by using DSK.DISKNAME.FILENAME instead of DSK1.FILENAME. The program will start the search on DSK1 for a disk named DISKNAME. If this disk is not found on DSK1., the search will continue on DSK2, and so on, until the disk is found. When it is found, the file called FILENAME will be loaded from the disk.

I find this search from drive to drive to be slow, so the only use is for disks that should be put in DSK1. This is useful if you have only one drive, a hard disk with DSK emulation or a RAMdisk on CRU >1000.

If you have a TI controller on CRU >1100, then the search from BASIC will begin with any RAMdisk on CRU >1000, if one is present, and then in sequence before an error message is given: DSK1 to DSK2 to DSK3 then "I/O ERROR 57." A RAMdisk DSK9 on CRU >1200 or higl will not be reached.

I have a Myarc HFDC (DSK5-8) on CRU >1000 and a TI controller (DSK1-3)

(See Page 31)

USER NOTES

(Continued from Page 30)

on CRU >1100. This gives the following search order: WDS1 to DSK5 to DSK6 to DSK7 to DSK8 to DSK1 to DSK2 to DSK 3 before generating the "I/O ERROR 57" message. A RAMdisk on a high CRU address will not be reached because the TI card will prevent this. A copy of Multiplan that is stored on a subdirectory such as WDS1.DSK.TIMP thus will load nice and quickly.

If I remove the TI card but keep the Myarc HFDC on CRU >1100 together with a Horizon RAMdisk (DSK9) on a high CRU address, then the search will be: WDS1 to DSK1 to DSK2 to DSK3 to DSK4 to DSK9 before generating the "I/O ERROR 50" message. As you can see, a high CRU can now be reached.

If I repeat the same search from Extended BASIC, the search is repeated twice before generating an error message. This applies to all three of the above examples. With only the TI card in place I get DSK1 to DSK2 to DSK3 to DSK1 to DSK2 to DSK3 before generating the "I/O ERROR 07" message. The Myarc card gives error "I/O ERROR 00." I don't know if this is caused by my Dijit AVPC or if it is the same on a normal TI99/4A.

This trick is handy to make a program drive independent. However, one drawback is that if the user relocates a program to a RAMdisk from a floppy or hard disk, it won't work unless the RAMdisk is renamed or the diskname reference is changed in the program.

MICROpendium pays $10 for items submitted by readers for use in the User Notes column. Send them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

CLASSIFIEDS

FOR SALE

ALL THIS FOR THE CC40		
8K Ram cartridge (for 6K computer)		35.00
16K Ram cartridge (adds 16K)		35.00
CC40 Computers	6K	50.00
CC40 Computers	18K	70.00
Electrical Engineering package (cart)		20.00
Finance Packages (cart. & book)		10.00
Games #1 (1 used $5)		10.00
Hexbus Interface Manual		10.00
Mathematics Package (cart. & book)		20.00
Memo Processor (word processor)		20.00
Modem	300 baud	50.00
Modem	300 baud (USED)	25.00
RS232 User's manual		5.00
Power Supply for CC40 and periphs.		15.00
Printer Plotter	4 color (pen set)	15.00
RS232		50.00
RS232 with parallel printer cable		75.00
RS232 parallel printer cable		20.00
Statistics Package (cart & book)		20.00
Wafertape Drive User's Manual		10.00
Wafertapes		10.00
Word Processor (cart & book) used (also has a modem program)		10.00

S&H 10% Minimum $3 214 821 9274
Jim Lesher, 722 Huntley, Dallas Tx 75214

GRAM KRACKER FOR SALE

80K GRAM Kracker, with docs and disks — $80. Call 512-255-1512.

v10n9

"THE WATCHAMACALIT" From DBM TECHNOLOGIES

ATTENTION MYARC HFDC USERS! LIMITED SUPPLY $29.95 Each. GET YOUR HFDC UPGRADED NOW! From DBM TECHNOLOGIES 17725 22nd Street Ogden, Utah 84401-2112 Phone: (801) 782-1004 or (801) 394-6815. "THE WATCHAMACALIT" is designed to Back-Up the clock on the Myarc HFDC. Reviewed in MICROpendium in August 1993. Comes complete with software.

v11n10

FOR SALE

TI99/4A complete system, all or part, including 3 consoles, PEB with 32K card, RS232 card, disk controller, 2 DSDD disk drives, original TI color monitor. System comes with 50+ disks of programs, including TI-Artist, Writer, Multiplan, & Editor/Assembler Extended BASIC. All comes with instruction books & a complete set of MICROpendium from December '83 to December '92. Make offer. Call Tim Blakley, (916) 788-7468. v11n10

FOR SALE — TI99/4A

HARDWARE — 3 '99/4A consoles and expansion boxes with cards. (2 exp. boxes also contain P-code cards.) CC-40 computer with modules. Many more items such as speech, joysticks, keyboards, drives, cables, much else.

FIRMWARE/SOFTWARE — Huge assortment. 75+ modules & many dozens of disks. Includes Ex. BASIC modules, TI-Writer, Multiplan, Plato, Pascal, Miller Graphics & Traver programs. Lots of educational games, utilities, etc.

BOOKS/MAGAZINES — Several shelves of TI material. Books, newsletters, many publications, including all back issues of MICROpendium.

Only $325.00 for all. Jerry B. 612-439-9675

v11n9

What a deal!

Prices slashed on MICROpendium Classifieds!

Classified ads are now available for 10 cents per word, a reduction of more than 50 percent.

If you've got something you want to sell or buy, advertise it in MICROpendium Classifieds.
Simply write your ad on a separate sheet of paper, count the words (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.

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