Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 8 Number 7 August 1991 $2.50

Spatial Relationships
by Regena

Keeping track of telephone lists
by Jerry Stern

Accessing built-in assembly routines
by Barry Traver

Top down programming
by Bruce Harrison

Dumping modules to disk with Funnelweb, enabling the XB break key with the Geneve and nesting .IF files with TI-Writer

Barry Boone squeezes digital sound from the TI's little sound chip

Reviews of Mario Brothers, Turbo 2056 and Linkage

Contents

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $25 per year by Burns-Koloen Communications Inc., 16606 Terrace Dr., Austin, TX 78728-1156. Second-class postage paid at Austin, Texas, and additional mailing offices. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

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

Foreign subscriptions are $30.25 (Mexico); $32.50 (Canada); $30.00, surface mail to other countries; $42 airmail to other countries.

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

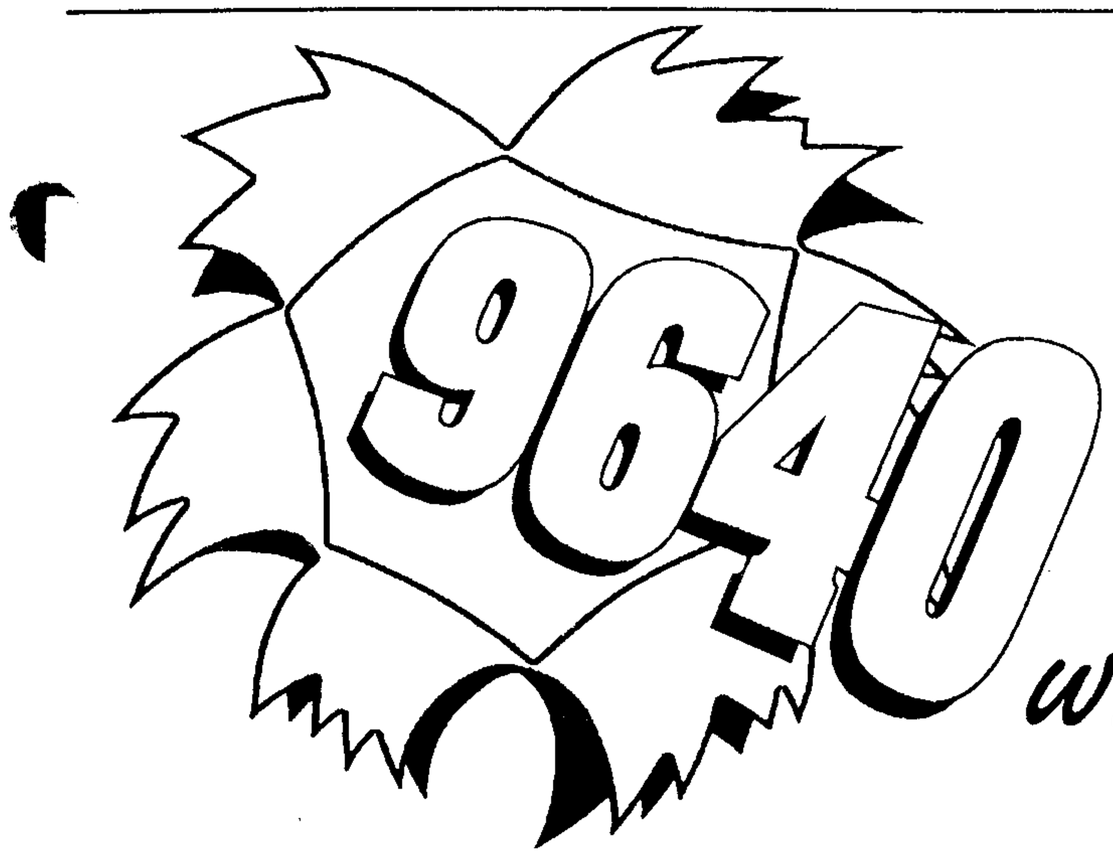
Mailing address: P.O. Box 1343, Round Rock TX 78680
Telephone: (512) 255-1512

CompuServe: 75156,3270
Delphi TI NET: MICROPENDIUM
GEnie: J.Koloen

John Koloen.....Publisher
Laura Burns.....Editor

*READ THIS

Here are some tips to help you when entering programs from MICROpendium:
1. All BASIC and Extended BASIC programs are run through Checksum, the numbers that follow exclamation points at the end of each program line. Do not enter these numbers or exclamation points. Checksum was published in the October 1987 edition.
2. Long XBASIC 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.



Let your Geneve break through with software from Asgard Software!

MDOS SPECIFIC

TRIS-9640

An addictive mind-teaser inspired by Tetris, *Tris* is a true-to-the-original version of this popular game with all of the little features and nuances that made the original a hit in the arcades and at home. After running it from the M-DOS command line, you'll be transported into a world of ever-faster falling blocks. Rotate and move them into place to form solid rows, which disappear when you do and pile up when you don't. Take advantage of time before pieces "stick", the "piece preview" function, or variable starting heights in planning your strategy. Play with keys or joysticks. By Jim Reiss.

Item E-01b - Disk - $9.95

PAGE PRO SIDEWAYS PICTURE PRINTER

With this utility, and a little help from *Page Pro*, you can quickly create landscape (sideways) calendars, certificates, flyers and more on your 9640! This utility places no limits on your imagination - use any fonts, pictures or borders in any order. Run from the command line, this program features "batch" printing, variable enlargements, 80-column support, a variety of printer resolutions and more. Requires an Epson or compatible printer. By Chris Bobbitt.

Item P04 - Disk - $12.95

TYPEWRITER 9640

Have you ever said while using a word processor, "I could do this easier with a typewriter!"? For many jobs - short notes and memos, envelopes, the occasional label - a word processor is too much. If all you need to do is pound out something quickly and easily without a lot of fuss, a typewriter fills the bill. Typewriter 99 turns your computer into a full-function modern typewriter with features like auto-justification and centering, bold and underline text, one or two line spacing, and much more. Developed with the aid of secretaries and typists, this program is the most natural way to type text on your Geneve, and is perfect for the computer-phobe or those with a lot of work and a little time. Runs directly from M-DOS. A printer required. By Jim Reiss

Item P18c - Disk - $14.95

THUMBNAILS

Page Pro and *TPA* users take note! Introducing the Ultimate tool for managing MacPaint pics for the Geneve - *Thumbnails!* This innovative program functions much like a disk manager for MacPaint pictures. Hard drive compatible, this program will allow you to catalog a disk (or directory) of MacPaint pics, display them at full size one at a time or in a high-resolution "slideshow", print them, view them in a condensed (thumbnail) format, convert them to *TI-Artist* Instance or *Page Pro* format, and finally, print them nine-to-a-page, with a border around each and the filename and pathname underneath, on your Epson or compatible printer. Perfect for generating reference sheets to keep track of your collection, or just plain viewing and converting pictures. Includes a nice collection of MacPaint pics (including some very famous karate-loving turtles). Runs from M-DOS - By Francisco Garcia.

Item G16 - Disk - $12.95

U.S. Please add$2.50/order, Canada $3.00, Airmail $5.00. Send orders to:

Asgard Software • P.O. Box 10306 • Rockville, MD 20849

Feedback

Praise for Myarc

I sent a check to Myarc April 5, 1991, ordering the Hard and Floppy Disk Controller and received the HFDC three weeks later. James Crosson of the NET 99er user group spent one afternoon setting up my recently purchased used Geneve, used hard disk drive and new HFDC. After no problems setting it up, he gave me instructions how to use the FAST computer. The only errors I have had since are when I hit the wrong key.

Thanks to Myarc, MICROpendium, James Crosson and my user group NET 99er. I have learned to use the TI99/4A as a result of being associated with them since 1983 — Myarc since April 1991.

Jo Nell Thompson
Fort Worth, Texas

Where is everybody?

Here in East Central Florida, a tiny remnant of the once-mighty Daytona 99ers, still loyal to the TI, cling desperately to MICROpendium and whatever second-hand newsletters we can get hold of, as our only link to the rest of the TI world. We who have never even seen a RAMdisk drool over such exotic goodies as hard drives, 80-column cards and all the other incredible things announced, advertised, reviewed and talked about in the pages of MICROpendium.

Yet we are beginning to wonder if all this is vaporware. At last being able to afford some of these high-powered gadgets, I resolved to catch up with the rest of TI-dom, urged on by two or three of my fellow orphans. On May 28 I wrote 17 letters to vendors noted in MICROpendium as having been at the last Chicago Faire, requesting catalogs, prices and any available information on their products. To date I have received five catalogs or price lists, all from software vendors. (Thank you to those who responded.) We don't lack for software, we need hardware! If this stuff really exists, I want to buy some of it!

In mid-June, while waiting for replies to my letters, I also resolved to get up on the bulletin boards, and to join at least one of the Big Three information services to get some information via modem. After dozens of hours and scores of phone calls, I still have not been able to obtain a working telephone number for CompuServe, GEnie or Delphi! My letters have been returned as undeliverable. Not even a phone call to MICROpendium could produce a usable phone number or a correct address!

Bulletin boards? Hah! There aren't any in Florida! The Miami User's Group does not exist, as far as Southern Bell is concerned. All other phone numbers listed anywhere for TI BBSes in Florida have led to dead ends. "We're sorry, that number is no longer in service."

Back on May 1, I had written to Barry Traver requesting his GRAPHICOMP program on disk, and enclosed the requested amount. In late June I sent him a copy of the original letter, asking if he had received it. Still no answer to either letter, and, of course, no disk.

What's going on here? Are we the victims of an elaborate, cruel hoax? Or of some sinister conspiracy to stampout all remaining TIers in Florida? If this is the kind of help and support we can expect from the TI community, it's no wonder that most of the Daytona 99ers long ago switched to clones.

I would hate to give up the nine years I've got invested in my TI to start all over again with a foreign system. But if our own people continue to ignore our cries for help, I'm about ready to defect, too.

Arnie Stewart
New Smyrna Beach, Florida

Sorry not to have been able to find these for you during your call, but you can call CompuServe at 1-800-848-8199, Delphi at 1-800-544-4005 or GEnie at 1-800-638-9636 (all voice numbers).

Where's Myarc?

I am currently in a quandary regarding what to do about my Myarc Floppy Disk Controller Card which no longer works and I wish to have repaired.

I have sent a letter for information to Myarc at Basking Ridge, New Jersey, which came back with a postal stamp, "Return to Sender, Undeliverable as Addressed, Forwarding Order Expired." This address I had dug out of the original box the card came in. Then I dug into the stuff I got with my "HFDC" and found an address in Martinsville, New Jersey — again, the prodigal letter returned!

Do you, or do any readers, have a current address or non-mechanically answered phone number where I can get some good up-to-date information on how to get my card fixed?

Patrick Graham
North Bay, Ontario, Canada

The recorded message at (908) 805-0007 says repairs go to 50 Darren Woods, Martinsville, NJ 08836, and questions to P.O. Box 140, Basking Ridge, NJ 07920. The former Alabama phone number is no longer in service. — Ed.

Help given, sought

I did the TI community a favor, now could someone please help me? On the million-plus Prodigiy membership bulletin board I posted a bulletin about the TI. In it, people can find out how to subscribe to MICROpendium, and I listed several TI suppliers. I also encouraged others to list other reputable vendors.

If you have Prodigy, you can find it by JUMPing BULLETINBOARD and choosing the Computer Club. It's listed under HARDWARE:SYSTEMS as TI99/4A STILL LIVES! I tried to list it as its own topic but Prodigy reclassified it.

My problems: I desperately need a conversion program that will change my TI word processing document to IBM and vice versa. JP Software doesn't even put on their answering machine anymore, so I can't get any help there. I work for a law firm, but am I supposed to subpoena the company for a program? I would even settle for a *copy! Help!*

Problem No. 2: My TI will no longer print anything to my printers (dot matrix and laser). I will get some stray character on the top two lines and nothing else. What is the possible problem, and who can fix it?

Frank P. DeCandia
Jersey City, New Jersey

We appreciate your mention of us on Prodigy. Now, concerning your first problem, there is no vendor that we know of who is currently supplying either Cor-

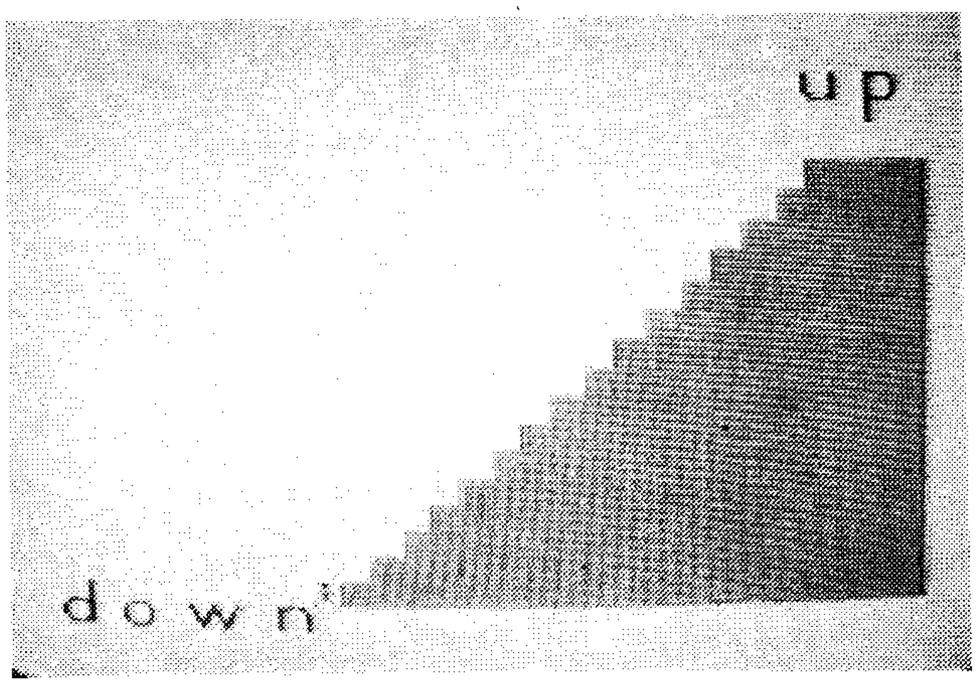
(See Page 29)

BASIC

Spatial Relationships

By REGENA

This summer has turned out to be a bit hectic. My son is getting married, then the very next day I leave for Washington, D.C., to attend the White House Conference on Library and Information Services (July 9-13) as one of the delegates from Utah. The day after I return from Washington, D.C., we will travel to another son's Senior League All-Star Baseball tournament. I just wish I had a lap-top to take with me on the trips! Computers have really become part of our daily lives!



Because of my lack of time for programming this month, I am submitting a program that I wrote in 1983 which was available on cassette only. I have revised it to be able to use with the disk system. This program is "Spatial Relationships" for preschoolers and uses the Speech Synthesizer and the Terminal Emulator 2 command module.

Put in the Terminal Emulator 2 command module and press 1 for TI BASIC. Then type CALL FILES(1) and press ENTER. Type NEW and press ENTER. Now you may type in or load the program and run it.

Spatial concepts are introduced to the young child: up/down, in/out, over/under, low/high, in front/behind, around/through and far/near. A picture is drawn for each set of concepts. The words are spelled in lowercase letters for the beginning reader. The words are spoken as the concept is illustrated with moving or blinking graphics.

After two words are introduced, the computer randomly chooses one of the concepts and illustrates it. The child must press 1 or 2 to choose which concept is shown. All instructions are given orally, so choices need to be made after the talking finishes.

With the Terminal Emulator 2 command module in the console and in TI BASIC mode, you may use the command

OPEN #1: "SPEECH",OUTPUT

Then to have the computer speak, use PRINT # 1, such as

PRINT #1:"^ YES."
PRINT #1:A$(T)
PRINT #1:"^ CHOOSE";A$(0);"__OR";A$(1)

DATA statements are used to read in the words used for A$(0) and A$(1) and the graphics character definitions and placements. Subroutines are used to present the concept for each set of words, to define characters, to draw graphics, to present the quiz and to blink or move characters.

If you wish to save typing effort, you may have a copy of this program by sending $4 to *REGENA, 918 Cedar Knolls West, Cedar City, UT 84720.* Be sure to specify that you need "Spatial Relationships" for the TI and whether you want cassette or diskette.

SPATIAL RELATIONSHIPS

```
10 REM SPATIAL RELATIONSHIPS
 !237
20 REM BY REGENA !071
30 REM REQUIRES SPEECH SYNTH
ESIZER AND TERMINAL EMULATOR
 2    !084
40 DIM A$(1)!086
50 OPEN #1:"SPEECH",OUTPUT !
122
60 CALL CLEAR !209
70 PRINT " ****************
********":" *";TAB(26);"*":"
 * SPATIAL RELATIONSHIPS *":
" *";TAB(26);"*" !142
80 PRINT " ****************
********": : : : : : : : :
 :!126
90 CALL COLOR(12,2,1)!222
100 GOSUB 3010 !029
110 DATA "",,20,97,3D4381818
181433D,98,BCC281818181C2BC,
100,0000010101010101,101,3C4
281FF8080423C,102 !158
120 DATA 060908080808083E,10
3,0101010141221C,104,0000808
08080808,105,00000008,108,08
080808080808,110 !196
130 DATA BCC2818181818181,11
1,3C4281818181423C,112,80808
080808,114,BCC281808080808,1
16,0000080808087F08,117 !129
140 DATA 8181818181433D,11
8,4141222214140808,119,04048
8885050202,123 !075
150 DATA 00000000030C30C,124
,030C30C,125,010204081020408
 !007
160 GOSUB 3280 !044
170 CALL CLEAR !209
180 CALL SCREEN(8)!153
190 CALL COLOR(1,2,1)!171
200 CALL COLOR(2,2,1)!172
210 GOSUB 3000 !019
220 DATA ^UP.,^DOWN.,2,136,1
C1C083C18142242,137,3838103C
18284442 !118
230 CALL COLOR(13,14,14)!072
240 CALL COLOR(14,5,1)!227
250 CALL CLEAR !209
260 RESTORE 280 !118
270 GOSUB 2940 !215
280 DATA 6,22,3,100,23,3,97,
```

(See Page 8)

REGENA ON BASIC—

(Continued from Page 17)

```
23,5,111,23,7,118,23,8,119,2
3,10,110 !127
290 PRINT #1:A$(1)!020
300 J=12 !053
310 FOR C=22 TO 7 STEP -1 !2
18
320 CALL HCHAR(C,J,128,31-J)
!131
330 J=J+1 !013
340 NEXT C !217
350 CALL HCHAR(4,27,117)!055
360 CALL HCHAR(4,29,98)!016
370 CALL HCHAR(5,29,112)!053
380 PRINT #1:A$(0)!019
390 GOSUB 3360 !125
400 PRINT #1:A$(1)!020
410 GOSUB 3460 !225
420 PRINT #1:A$(0)!019
430 GOSUB 3310 !074
440 ON T+1 GOSUB 3460,3360 !
052
450 GOSUB 3070 !089
460 IF F=1 THEN 250 !245
470 ON T+1 GOSUB 3460,3360 !
052
480 PRINT #1:A$(T)!110
490 GOSUB 3280 !044
500 RESTORE 520 !103
510 GOSUB 3000 !019
520 DATA ^IN.,^OUT.,9,128,01
03070F1F3F7FFF,129,0080C0E0F
0F8FCFE,130,FFFFFFFFFFFFFFFF
 !211
530 DATA 136,3C7EE7C3C3E77E3
C,137,1818FFFF18181818,138,3
C6666C3C3C3C3C3,144,3C7EE7C3
C3E77E3C !040
540 DATA 145,1818FFFF1818181
8,146,3C6666C3C3C3C3C3 !081
550 CALL COLOR(13,3,1)!224
560 CALL CLEAR !209
570 CALL COLOR(14,7,3)!231
580 CALL COLOR(15,7,1)!230
590 FOR C=21 TO 14 STEP -1 !
008
600 CALL HCHAR(C,6,130,12)!0
36
610 NEXT C !217
620 K=10 !052
630 J=6 !007
640 FOR C=13 TO 9 STEP -1 !2
20
650 CALL HCHAR(C,J,128)!150
660 CALL HCHAR(C,J+1,130,K)!
072
670 CALL HCHAR(C,J+K+1,129)!
094
680 J=J+1 !013
690 K=K-2 !017
700 NEXT C !217
710 CALL HCHAR(8,11,128)!054
720 CALL HCHAR(8,12,129)!056
730 PRINT #1:A$(0)!019
740 RESTORE 760 !087
750 GOSUB 2940 !215
760 DATA 13,22,10,105,23,10,
108,23,12,110,18,11,136,19,1
1,137,20,11,138,18,25,144 !1
78
770 DATA 19,25,145,20,25,146
,23,23,111,23,25,117,22,27,1
16,23,27,108 !013
780 PRINT #1:A$(1)!020
790 GOSUB 3280 !044
800 PRINT #1:A$(0)!019
810 GOSUB 3550 !059
820 PRINT #1:A$(1)!020
830 GOSUB 3610 !120
840 CALL COLOR(14,3,3)!227
850 CALL COLOR(15,1,1)!224
860 GOSUB 3310 !074
870 ON T+1 GOSUB 3550,3610 !
137
880 GOSUB 3070 !089
890 IF F=1 THEN 560 !045
900 ON T+1 GOSUB 3550,3610 !
137
910 GOSUB 3280 !044
920 RESTORE 940 !012
930 GOSUB 3000 !019
940 DATA ^O VER.,^UN DER.,3,
128,FFC3A59999A5C3FF,136,3C2
424FFFFFF0C0C,137,000000FFFF
FF1818 !199
950 CALL COLOR(13,2,1)!223
960 CALL COLOR(14,5,1)!227
970 CALL CLEAR !209
980 CALL HCHAR(11,6,128,22)!
021
990 CALL HCHAR(12,6,128,22)!
022
1000 CALL VCHAR(13,6,128,7)!
247
1010 CALL VCHAR(13,27,128,7)
!043
1020 CALL VCHAR(13,7,128,5)!
246
1030 CALL VCHAR(13,26,128,5)
!040
1040 RESTORE 1060 !133
1050 GOSUB 2940 !215
1060 DATA 6,10,10,136,10,9,1
37,6,13,111,6,15,118,6,17,10
1,6,19,114 !183
1070 PRINT #1:A$(0)!019
1080 GOSUB 3670 !180
1090 PRINT #1:A$(0)!019
1100 GOSUB 3280 !044
1110 GOSUB 2940 !215
1120 DATA 8,21,12,117,21,14,
110,20,16,100,21,16,97,21,18
,101,21,20,114,20,10,136,20,
9,137 !235
1130 PRINT #1:A$(1)!020
1140 GOSUB 3740 !250
1150 PRINT #1:A$(1)!020
1160 CALL HCHAR(10,23,32,2)!
218
1170 CALL HCHAR(15,19,32,2)!
228
1180 GOSUB 3310 !074
1190 ON T+1 GOSUB 3670,3740
!132
1200 GOSUB 3070 !089
1210 IF F=1 THEN 970 !200
1220 ON T+1 GOSUB 3670,3740
!132
1230 GOSUB 3280 !044
1240 RESTORE 1260 !077
1250 GOSUB 3000 !019
1260 DATA ^LO.,^HIGH.,3,128,
10525276F6FFFFFF,136,1038383
838387C92,144,020646FFFF4606
02 !224
1270 CALL COLOR(13,3,1)!224
1280 CALL COLOR(14,7,1)!229
1290 CALL COLOR(15,5,1)!228
1300 CALL CLEAR !209
1310 CALL HCHAR(24,10,128,21
)!068
1320 PRINT #1:A$(0)!019
1330 RESTORE 1350 !168
1340 GOSUB 2940 !215
1350 DATA 6,22,5,144,23,3,10
8,24,3,108,24,5,111,24,7,118
,24,8,119 !143
1360 GOSUB 3830 !084
1370 PRINT #1:A$(0)!019
1380 GOSUB 3280 !044
1390 PRINT #1:A$(1)!020
1400 GOSUB 2940 !215
1410 DATA 9,1,22,104,2,22,11
```

(See Page 8)

REGENA—

(Continued from Page 8)

```
0,1,24,105,2,24,108,2,26,97,
3,26,103,1,28,104,2,28,110,2
3,15,136 !112
1420 GOSUB 3890 !145
1430 PRINT #1:A$(1)!020
1440 CALL HCHAR(22,29,32)!05
3
1450 CALL HCHAR(1,15,32)!252
1460 GOSUB 3310 !074
1470 ON T+1 GOSUB 3830,3890
!187
1480 GOSUB 3070 !089
1490 IF F=1 THEN 1300 !020
1500 ON T+1 GOSUB 3830,3890
!187
1510 GOSUB 3280 !044
1520 RESTORE 1540 !103
1530 GOSUB 3000 !019
1540 DATA _IN ^FRONT.,_BE ^H
IND.,9,128,FFFFFFFFFFFFFFFF,
129,600C528C229146A9,130,010
2060C123146D9 !125
1550 DATA 131,610E568C32D146
A9,132,610E448830A0408,136,3
C7EE7C3C3E77E3C,137,1818FFFF
18181818 !093
1560 DATA 138,3C6666C3C3C3C3
C3,144,003C4281A9819142 !027
1570 CALL COLOR(13,12,1)!017
1580 CALL CLEAR !209
1590 CALL COLOR(14,7,12)!024
1600 CALL COLOR(15,2,1)!225
1610 PRINT #1:A$(0)!019
1620 FOR C=10 TO 15 !153
1630 CALL HCHAR(C,10,128,11)
!086
1640 NEXT C !217
1650 J=10 !051
1660 FOR C=9 TO 7 STEP -1 !1
74
1670 CALL HCHAR(C,J,130)!143
1680 CALL HCHAR(C,J+1,129,10
)!048
1690 CALL HCHAR(C,J+11,131)!
125
1700 CALL VCHAR(C+1,J+11,129
,5)!254
1710 CALL HCHAR(C+6,J+11,132
)!062
1720 J=J+1 !013
1730 NEXT C !217
1740 RESTORE 1760 !067
1750 GOSUB 2940 !215
1760 DATA 13,13,15,136,14,15
,137,15,15,138,18,5,105,19,5
,108,19,7,110 !095
1770 DATA 18,11,102,19,11,10
8,19,13,114,19,15,111,19,17,
110,18,19,116,19,19,108 !238
1780 PRINT #1:A$(0)!019
1790 GOSUB 3280 !044
1800 CALL COLOR(14,12,12)!06
9
1810 PRINT #1:A$(1)!020
1820 RESTORE 1840 !148
1830 GOSUB 2940 !215
1840 DATA 11,6,18,144,2,17,1
04,3,17,98,3,19,101,2,21,104
,3,21,110,2,23,105,3,23,108
!200
1850 DATA 3,25,110,2,27,100,
3,27,97 !168
1860 GOSUB 3280 !044
1870 PRINT #1:A$(1)!020
1880 GOSUB 3280 !044
1890 CALL COLOR(15,1,1)!224
1900 GOSUB 3310 !074
1910 ON T+1 GOSUB 3970,4020
!202
1920 GOSUB 3070 !089
1930 IF F=1 THEN 1580 !045
1940 ON T+1 GOSUB 3970,4020
!202
1950 GOSUB 3280 !044
1960 RESTORE 1980 !032
1970 GOSUB 3000 !019
1980 DATA _UH ^ROUND.,^THRU.
,2,136,0787FCFCFC4444,137,E0
E13F3F3F2222 !054
1990 CALL COLOR(13,6,1)!227
2000 CALL COLOR(14,16,1)!022
2010 CALL CLEAR !209
2020 CALL HCHAR(7,11,128,7)!
232
2030 CALL HCHAR(8,9,128,12)!
236
2040 CALL HCHAR(9,8,128,13)!
237
2050 CALL HCHAR(10,7,128,13)
!021
2060 CALL HCHAR(11,8,128,11)
!021
2070 CALL HCHAR(12,10,128,8)
!021
2080 PRINT #1:A$(0)!019
2090 RESTORE 2110 !163
2100 GOSUB 2940 !215
2110 DATA 8,17,8,97,17,10,11
4,17,12,111,17,14,117,17,16,
110,16,18,100,17,18,97,9,5,1
36 !141
2120 GOSUB 4070 !069
2130 PRINT #1:A$(0)!019
2140 GOSUB 3280 !044
2150 CALL HCHAR(16,18,32)!05
4
2160 CALL HCHAR(17,8,32,11)!
228
2170 PRINT #1:A$(1)!020
2180 RESTORE 2200 !253
2190 GOSUB 2940 !215
2200 DATA 11,16,8,116,17,8,1
08,16,10,104,17,10,110,17,12
,114,17,14,111,17,16,117 !08
6
2210 DATA 17,18,97,18,18,103
,16,20,104,17,20,110 !032
2220 GOSUB 4190 !190
2230 PRINT #1:A$(1)!020
2240 GOSUB 3310 !074
2250 CALL HCHAR(9,22,32)!002
2260 CALL HCHAR(16,8,32,13)!
229
2270 CALL HCHAR(17,8,32,13)!
230
2280 CALL HCHAR(18,18,32)!05
6
2290 ON T+1 GOSUB 4070,4190
!217
2300 GOSUB 3070 !089
2310 IF F=1 THEN 2010 !220
2320 ON T+1 GOSUB 4070,4190
!217
2330 GOSUB 3280 !044
2340 RESTORE 2360 !158
2350 GOSUB 3000 !019
2360 DATA ^FAR.,^NEAR.,11,12
8,087F08087F080808,129,FC000
00000FC,130,0000000000000101
,131,7F010101017F0101 !149
2370 DATA 132,01010101010101
01,133,00000000FF,134,080808
08FF080808,136,3838107C10282
828,144,3C7EE7C3C3E77E3C !14
3
2380 DATA 145,1818FFFF181818
18,146,3C6666C3C3C3C3C3 !081
2390 CALL CLEAR !209
2400 J=5 !006
2410 FOR C=15 TO 4 STEP -1 !
217
2420 CALL HCHAR(C,J,123)!145
```

(See Page 10)

REGENA ON BASIC—

(Continued from Page 9)

```
2430 CALL HCHAR(C,J+1,124)!0
77
2440 J=J+2 !014
2450 NEXT C !217
2460 J=13 !054
2470 FOR C=22 TO 6 STEP -1 !
217
2480 CALL HCHAR(C,J,125)!147
2490 J=J+1 !013
2500 NEXT C !217
2510 CALL COLOR(13,2,1)!223
2520 CALL COLOR(14,7,1)!229
2530 CALL COLOR(15,7,1)!230
2540 PRINT #1:A$(0)!019
2550 RESTORE 2570 !113
2560 GOSUB 2940 !215
2570 DATA 24,1,26,102,2,26,1
08,2,28,97,2,30,114,3,24,128
,4,24,108,4,17,130,5,17,131,
5,18,129 !138
2580 DATA 6,17,132,7,17,132,
7,6,133,7,7,134,7,8,133,8,6,
133,8,7,134,8,8,133,9,7,108,
10,7,108,11,7,108
2590 DATA 22,15,110,22,17,10
1,22,19,97,22,21,114 !020
2600 PRINT #1:A$(1)!020
2610 RESTORE 2630 !173
2620 GOSUB 2940 !215
2630 DATA 4,6,27,136,18,8,14
4,19,8,145,20,8,146 !080
2640 PRINT #1:A$(0)!019
2650 GOSUB 2880 !155
2660 PRINT #1:A$(1)!020
2670 GOSUB 3610 !120
2680 CALL COLOR(14,1,1)!223
2690 CALL COLOR(15,1,1)!224
2700 GOSUB 3310 !074
2710 ON T+1 GOSUB 2880,3610
!233
2720 GOSUB 3070 !089
2730 IF F=1 THEN 2390 !090
2740 ON T+1 GOSUB 2880,3610
!233
2750 GOSUB 3280 !044
2760 CALL CLEAR !209
2770 PRINT TAB(9);"G R E A T
 ": : : : :!040
2780 FOR C=1 TO 50 !103
2790 CALL SOUND(-99,INT(500*
RND+262),2)!027
2800 NEXT C !217
2810 PRINT "PRESS 1 TO TRY A
GAIN": :"      2 TO END PROG
RAM" !226
2820 PRINT #1:"^PRESS 1 2 TR
Y AGAIN. ^PRESS 2 2 END." !0
99
2830 CALL KEY(0,K,S)!187
2840 IF K=50 THEN 4270 !243
2850 IF K<>49 THEN 2830 !023
2860 RESTORE 220 !057
2870 GOTO 210 !033
2880 FOR C=1 TO 10 !099
2890 CALL SOUND(50,1497,5)!1
48
2900 CALL COLOR(14,16,1)!022
2910 CALL COLOR(14,7,1)!229
2920 NEXT C !217
2930 RETURN !136
2940 READ N !229
2950 FOR C=1 TO N !134
2960 READ A,B,G !199
2970 CALL HCHAR(A,B,G)!109
2980 NEXT C !217
2990 RETURN !136
3000 CALL CLEAR !209
3010 READ A$(0),A$(1),N !226
3020 FOR I=1 TO N !140
3030 READ C,C$ !244
3040 CALL CHAR(C,C$)!081
3050 NEXT I !223
3060 RETURN !136
3070 PRINT #1:"^PRESS 1 4";A
$(0):"^PRESS 2 4";A$(1)!191
3080 C=0 !250
3090 CALL KEY(0,K,S)!187
3100 C=C+1 !255
3110 IF C>200 THEN 3070 !102
3120 IF (K<49)+(K>50)THEN 30
90 !049
3130 IF K-49=T THEN 3190 !21
3
3140 CALL SOUND(50,330,2)!08
1
3150 CALL SOUND(50,262,2)!08
5
3160 F=1 !254
3170 PRINT #1:"^NO." !060
3180 GOTO 3250 !013
3190 CALL SOUND(50,523,2)!08
5
3200 CALL SOUND(50,660,2)!08
7
3210 CALL SOUND(50,784,2)!09
4
3220 CALL SOUND(100,1046,2)!
180
3230 F=0 !253
3240 PRINT #1:"^YES." !145
3250 PRINT #1:"^IT IS ";A$(T
)!199
3260 GOSUB 3280 !044
3270 RETURN !136
3280 CALL SOUND(1000,9999,30
)!048
3290 CALL SOUND(1,9999,30)!1
57
3300 RETURN !136
3310 GOSUB 3280 !044
3320 RANDOMIZE !149
3330 T=INT(2*RND)!227
3340 PRINT #1:"^CHOOSE";A$(0
);"_OR";A$(1)!178
3350 RETURN !136
3360 CALL HCHAR(22,11,32)!04
4
3370 CALL HCHAR(6,27,137)!05
9
3380 J=27 !059
3390 FOR C=6 TO 21 !106
3400 CALL SOUND(100,262+J*12
,4)!127
3410 CALL HCHAR(C,J,32)!095
3420 J=J-1 !014
3430 CALL HCHAR(C+1,J,137)!0
81
3440 NEXT C !217
3450 RETURN !136
3460 CALL HCHAR(6,27,32)!004
3470 J=11 !052
3480 FOR C=22 TO 7 STEP -1 !
218
3490 CALL SOUND(100,262+J*12
,4)!127
3500 CALL HCHAR(C,J,32)!095
3510 J=J+1 !013
3520 CALL HCHAR(C-1,J,136)!0
81
3530 NEXT C !217
3540 RETURN !136
3550 FOR C=1 TO 10 !099
3560 CALL SOUND(50,1497,5)!1
48
3570 CALL COLOR(14,16,3)!024
3580 CALL COLOR(14,7,3)!231
3590 NEXT C !217
3600 RETURN !136
3610 FOR C=1 TO 10 !099
3620 CALL SOUND(50,748,5)!09
7
```

(See Page 11)

REGENA ON BASIC—

(Continued from Page 10)

```
3630 CALL COLOR(15,16,1)!023
3640 CALL COLOR(15,7,1)!230
3650 NEXT C !217
3660 RETURN !136
3670 FOR C=12 TO 24 STEP 2 !
072
3680 CALL SOUND(100,-3,2)!21
9
3690 CALL HCHAR(10,C-3,32,2)
!172
3700 CALL HCHAR(10,C,136)!11
8
3710 CALL HCHAR(10,C-1,137)!
051
3720 NEXT C !217
3730 RETURN !136
3740 J=12 !053
3750 FOR C=19 TO 15 STEP -1
!016
3760 CALL SOUND(100,-3,2)!21
9
3770 CALL HCHAR(C+1,J-3,32,2
)!134
3780 CALL HCHAR(C,J,136)!149
3790 CALL HCHAR(C,J-1,137)!0
82
3800 J=J+2 !014
3810 NEXT C !217
3820 RETURN !136
3830 FOR C=5 TO 28 !112
3840 CALL SOUND(50,-2,4)!175
3850 CALL HCHAR(22,C,32)!067
3860 CALL HCHAR(22,C+1,144)!
051
3870 NEXT C !217
3880 RETURN !136
3890 J=523 !109
3900 FOR C=23 TO 2 STEP -1 !
214
3910 CALL SOUND(-100,J,4,-5,
8)!082
3920 CALL HCHAR(C,15,32)!069
3930 CALL HCHAR(C-1,15,136)!
055
3940 J=J+30 !064
3950 NEXT C !217
3960 RETURN !136
3970 FOR C=1 TO 10 !099
3980 CALL COLOR(14,12,12)!06
9
3990 CALL COLOR(14,7,12)!024
4000 NEXT C !217
4010 RETURN !136
4020 FOR C=1 TO 10 !099
4030 CALL COLOR(15,1,1)!224
4040 CALL COLOR(15,2,1)!225
4050 NEXT C !217
4060 RETURN !136
4070 RESTORE 4080 !092
4080 DATA 9,5,136,8,6,136,7,
7,136,6,8,136,5,9,136,5,11,1
36,5,13,136,5,15,136,5,17,13
6,5,19,136 !178
4090 DATA 6,21,136,7,22,136,
8,23,136,9,24,136,10,23,137,
11,22,137,12,21,137,13,20,13
7,14,19,137 !218
4100 DATA 14,17,137,14,15,13
7,14,13,137,14,11,137,14,9,1
37,13,7,137,12,6,137,11,5,13
7,10,5,137 !173
4110 FOR C=1 TO 28 !108
4120 CALL SOUND(50,523+INT(5
00*RND),4)!078
4130 READ A,B,G !199
4140 CALL HCHAR(A,B,G)!109
4150 CALL HCHAR(A,B,32)!085
4160 NEXT C !217
4170 CALL HCHAR(9,5,136)!008
4180 RETURN !136
4190 G=32 !052
4200 FOR C=5 TO 21 !105
4210 CALL SOUND(50,-5,4)!178
4220 CALL HCHAR(9,C,G)!047
4230 CALL GCHAR(9,C+1,G)!233
4240 CALL HCHAR(9,C+1,136)!0
08
4250 NEXT C !217
4260 RETURN !136
4270 CALL CLEAR !209
4280 END !139
```

EXTENDED BASIC

Where did I write that phone number?

By JERRY STERN
©1991 J.L. Stern

Here in Baltimore, the area code will change to 410 this fall, and I've been going through phone number lists, and figuring who is far enough west to keep the old 301 code, and who changes. A phone database with a search function would keep track of names, addresses, phone numbers, and area codes, and that is this month's program.

PHONELIST stores lists of names and phone numbers, and sorts, searches, and updates them as people move. It records first and last name, address, city, the postal abbreviation for the state or Canadian province, and the U.S. ZIP+4 or Canadian postal code, and of course, the area code and phone number. The sort and search routines can resequence or search based on any of these fields, and entries found can be viewed, edited, or deleted. The entire list can be saved, merged with another list, printed in column format (Use compressed print only — it's 130 columns!) or in label style. Both print routines can also send the text to a disk file in TI-Writer's Display/Variable 80 format, or you can view the entire list, three entries at a time, on screen.

The program is best suited to making multiple phone lists; make a list of business contacts, and store it away as DSK1.BUSINESS, and friends can be DSK1.FRIENDS, or DSK1.USERGROUP, and so on. The separate lists work best because PHONELIST is a memory-based database, built for speed rather than storage capacity, so the database is limited to the amount of string memory, or stack memory, in the TI 99/4A. PHONE-

(See Page 12)

EXTENDED BASIC—

(Continued from Page 11)

LIST needs the expansion memory to run, but TI Extended BASIC cannot use the entire 32K expansion. Try typing SIZE from a blank TI Extended BASIC screen.

```
>SIZE
11840 BYTES OF STACK FREE
24488 BYTES OF PROGRAM
SPACE FREE
```

The TI 99/4A uses the console memory for storing text variables, or strings, and the upper 24K of the expansion memory for the program and numeric variables. (The lower 8K is used for assembly programming, and not available to Extended BASIC without some extra steps to combine the two program types.) The phone database uses 129 characters for each record. 11480 bytes of memory are available in the stack, and at 129 characters per entry, 91 names and data sets should fit into memory. But other string variables in the program use some space, so the upper limit, where full memory errors would begin, is 76 names. I've limited the program to only 70 names because as the stack approaches those last few bytes of memory, the computer slows down dramatically. When string variables are setup with the DIM statement, XB does not set aside all the space that they will need. String, or text, variables only begin to use memory as they are filled, so the slowdown effect doesn't begin until the database is nearly full.

Here are the typing instructions for PHONELIST:

1) On line 80, change the default drive number and name to the drive you will use for your data files.

2) On line 90, change the default printer name to match your machine.

3) On line 730, PHONELIST sets the printer to condensed print with printer code 15. If your printer uses a different code, make the change here.

4) Line 120 and line 1530 use ON ERROR 1650 statements to capture any fatal errors from empty disk drives, full disks, and so on. While entering the program, leave both ON ERROR statements out until you are certain that the listing is free of typing errors, and add the ON ERROR statements. Typing errors in a program that uses Extended BASIC's fatal error trapping features can make it very difficult to find out which lines contain the typos. If you use CHECKSUM to enter the program (highly recommended—see the October 1987 MICROpendium), or purchase the monthly disk from MICROpendium, this precaution is unnecessary.

5) Those of you who have different disk numbers than 1 through 3 should change the validation string in line 1520 to prevent accidentally trying to read the wrong disk directory. Users with one drive use "01", 2 drives "012", and so on. Only numbers are allowed for the directory function. Although the other I/O functions are protected by fatal error trapping, the directory routine is in a subprogram, and the ON ERROR routines cannot be used to transfer program control to outside of the subprogram.

To get started using PHONELIST, run the program, and press any key to leave the title screen. The menu will guide you to each option. A few of the options have been combined under one choice, such as the find option, which combines searching and viewing options. Here are the options in menu order.

Add to list: PHONELIST will add a new record at the end of the list. Press Enter by itself at the first prompt (last name) to return to the menu. For personal contacts, use the last name and first name as listed, but for business contacts, use last name for the company name, and first name for the contact or department name. Letters in the postal code and state entries will automatically be entered as uppercase—there is no need to hold down the shift key.

Disk directory and erase list. These two options will help keep the files manageable. The directory function will read any of the disk drives allowed by the line 1520 validation string, and the erase function will delete any file that is entered. Use the full filename, such as DSK1.README. If you change your mind, and want to cancel out of these options, use zero for the drive number to return to the main menu from the directory option. Entering a blank string from the erase option—press Function Erase to clear the default entry—will also return you to the main menu.

Find entry: Enter a search string to find, which must be the beginning of a field, such as the first few letters of a name, and the field, such as the area code, to search. Use the same capitalization as the entry you are searching for. "Micro" and "MICRO" are not the same. PHONELIST will search the entire list, and move the matching items to the low end of the list, and then allow you to examine the list by record number.

To look at an entry by number without searching, press ENTER at the "Search for what entry?" prompt, and PHONELIST will go directly to the next screen, for finding the entries by record number. Entering zero or blanks will return you to the main menu, or the letter L will take you to the last entry in the file.

As you view each entry, you may choose to edit the entry, delete it, go on to the next entry, or return to the main menu. If you delete any entries, you should resort the list before printing, because deleted entries are refilled from the end of the list.

If you make a mistake typing a new entry in the add option, finish typing that entry, and then press enter at the last name prompt when PHONELIST takes you to the next page. From the main menu, choose Find entry, press enter for the search, and enter L for the record number, and E for edit. That's finish the screen, then enter, F, enter, L, enter, E.

Load list: Loads a file from disk, which can replace the file in memory (load) or be added onto the end of the file in use (merge). If you want to add many entries that all have the same city, state, and area code to your list, you could make a dummy file that has only those fields listed in each entry, and uses about five identical entries. Merge that dummy file with your address list, and then use the search function to view and edit those default entries. To escape this option, enter a blank for the file name — press Function Erase to clear the default file name.

Print list: Allows printing the list to the screen, three entries at a time, or printing to a printer or disk file in column format, or label style. Entering a blank file name returns you to the main menu.

Unlike many TI programs, in PHONELIST these options are NOT the same. If you were to provide a disk file

(See Page 13)

EXTENDED BASIC—

(Continued from Page 12)

name to the print option, it would save a file correctly, but the file would not be loadable by TI-Writer. Similarly, using a printer name at the save disk file option would result in an 80-column printout rather than 130 columns.

Save list: Saves the entire list to a disk file. Use a complete file name, including the "DSKx." Again, a blank file name will return you to the menu.

Quick sort: Uses a text version of the quick sort routine to sort the file by any entry. The sort time will range from several seconds for files below 40 entries, to noticeably longer for files approaching the 70-entry limit. The reduction in speed is not caused by the sort routine, but by the TI 99/4A: As the stack space used for string variables gets nearly full, the computer slows down badly. To exit the sort routine without sorting the file, press zero.

Quit: Go Home. Go away. Shut down the store. And so on. Because PHONELIST keeps its records in memory, you must remember to save before quitting, so choosing Quit will give you one chance to change your mind. In response to "PRESS SPACE BAR TO QUIT" press any other key to return to the menu.

PHONELIST

```
80 D$,DR$="DSK2." ! Default
drive or file name !197
90 PR$="RS232.DA=8.BA=4800"
! Default printer name !200
100 REM PHONE LIST !144
110 REM TIXB JLS 8/91 V. 2.0
 !063
120 ON WARNING NEXT :: CALL
CLEAR :: CALL BLUE :: ON ERR
OR 1650 !064
130 CALL TITLE2 !031
140 V$="ABCDEFGHIJKLMNOPQRST
UVWXYZ0123456789._" ! ACCEPT
 VALIDATION STRING !243
150 DIM L$(71)!153
160 A$="Add to listErase lis
t Disk dir.  Find entry Load
 list  Print list Save list
 Quick Sort " !150
170 LU(1)=1 :: LU(2)=27 :: L
U(3)=54 :: LU(4)=82 ! data p
ositions in string !043
180 LU(5)=103 :: LU(6)=107 :
: LU(7)=118 :: LU(8)=122 !05
9
190 C=1 ! current record num
ber !233
200 CALL PAUSE !232
210 CALL MENU(A$,X):: IF X=1
0 THEN X=1 ! default menu ch
oice !106
220 ON X GOSUB 250,430,1500,
1100,490,590,930,1010,1540 !
220
230 GOTO 210 !033
240 REM ADD TO LIST !162
250 DISPLAY AT(1,1)ERASE ALL
:"Adding an entry..." !033
260 IF S>69 THEN DISPLAY AT(
5,1)BEEP:"This list is FULL!
" :: CALL PAUSE :: RETURN !2
54
270 GOSUB 1550 !100
280 DISPLAY AT(21,1):"Enter
no last name to returnto the
 menu." !023
290 S=S+1 :: C=S :: GOSUB 30
0 :: IF T$="" THEN S=S-1 ::
RETURN ELSE GOTO 250 !155
300 REM ACCEPT DATA !164
310 CALL KEY(5,K,T)!193
320 ACCEPT AT(5,1)SIZE(-25):
T$ !036
330 IF T$="" THEN RETURN ELS
E L$(C)=T$ !144
340 ACCEPT AT(7,1)SIZE(-26):
Z$ :: L$(C)=L$(C)&RPT$(" ",2
6-LEN(L$(C)))&Z$ !218
350 ACCEPT AT(9,1)SIZE(-28):
Z$ :: L$(C)=L$(C)&RPT$(" ",5
3-LEN(L$(C)))&Z$ !222
360 ACCEPT AT(11,1)SIZE(-19)
:Z$ :: L$(C)=L$(C)&RPT$(" ",
81-LEN(L$(C)))&Z$ !009
370 CALL KEY(3,K,T):: ACCEPT
 AT(13,1)SIZE(-3)VALIDATE(UA
LPHA):Z$ :: L$(C)=L$(C)&RPT$
(" ",102-LEN(L$(C)))&Z$ !148
380 ACCEPT AT(15,1)SIZE(-10)
VALIDATE(UALPHA,DIGIT,"-"):Z
$ :: L$(C)=L$(C)&RPT$(" ",10
6-LEN(L$(C)))&Z$ !204
390 ACCEPT AT(17,1)SIZE(-3)V
ALIDATE(DIGIT):Z$ :: L$(C)=L
$(C)&RPT$(" ",117-LEN(L$(C))
)&Z$ !092
400 ACCEPT AT(19,1)SIZE(-8)V
ALIDATE(DIGIT,"-"):Z$ :: L$(
C)=L$(C)&RPT$(" ",121-LEN(L$
(C)))&Z$ !006
410 RETURN !136
420 REM DELETE LIST !201
430 DISPLAY AT(1,1):"Deletin
g a file..." !043
440 DISPLAY AT(4,1):"Name of
 file to delete:" !066
450 DISPLAY AT(6,4):"Press E
nter to return       to main
 menu" !157
460 ACCEPT AT(5,2)SIZE(15)VA
LIDATE(V$)BEEP:D$ !037
470 IF D$="" THEN RETURN ELS
E DELETE D$ :: RETURN !053
480 REM LOAD LIST !054
490 DISPLAY AT(1,1):"Load da
ta..." !242
500 DISPLAY AT(4,1):"Load or
 Merge?":"L / M ?" !237
510 CALL KEYAT(5,1,K,"LM"):
 IF K=76 THEN S=0 :: GOTO 53
0 !248
520 DISPLAY AT(24,7)ERASE AL
L:"  Merge file" !053
530 CALL KEY(3,L,T):: DISPLA
Y AT(4,1):"Name of file to l
oad:":D$ :: ACCEPT AT(5,1)SI
ZE(-28)VALIDATE(V$):D$ :: IF
 D$="" THEN RETURN !117
540 OPEN #1:D$,DISPLAY ,VARI
ABLE 132,INPUT !084
550 IF S>69 THEN DISPLAY AT(
22,1)BEEP:"This list is FULL
!": :" " :: CALL PAUSE :: GO
TO 570 !131
560 IF EOF(1)THEN 570 ELSE S
=S+1 :: DISPLAY AT(24,1):"Lo
ading record #";S :: LINPUT
#1:L$(S):: GOTO 550 !156
570 CLOSE #1 :: RETURN !161
580 REM PRINT LIST !163
590 IF S<1 THEN CALL EMPTY :
: RETURN !067
600 DISPLAY AT(1,1):"Print l
ist..." !162
610 DISPLAY AT(4,1):"Print t
o Printer, Screen, orD/V 80
Disk file? P" !172
620 CALL KEYAT(5,19,L,"PpSsD
d")!231
630 IF POS("SsDdPp",CHR$(L),
```

EXTENDED BASIC—

(Continued from Page 13)

```
1)<3 THEN 850 !061
640 IF POS("SsDdPp",CHR$(L),
1)>4 THEN 690 !158
650 DISPLAY AT(8,1):"Name of
 disk file?":DR$ !234
660 ACCEPT AT(9,1)SIZE(-23)V
ALIDATE(V$):P$ !007
670 IF P$="" THEN RETURN !18
1
680 OPEN #1:P$,DISPLAY ,VARI
ABLE 80,OUTPUT :: GOTO 740 !
077
690 DISPLAY AT(8,1):"Name of
 device to print on?":PR$ !0
81
700 ACCEPT AT(9,1)SIZE(-23)V
ALIDATE(V$):P$ !007
710 IF P$="" THEN RETURN !18
1
720 OPEN #1:P$,DISPLAY ,VARI
ABLE 132,OUTPUT !197
730 PRINT #1:CHR$(15)! compr
essed print !224
740 DISPLAY AT(11,1):"Column
 format or Labels?":"(Labels
 do not include phonenumbers
.) C" !172
750 CALL KEYAT(13,11,L,"CcL1
")!061
760 IF POS("CcL1",CHR$(L),1)
>2 THEN 780 !037
770 FOR L=1 TO S :: PRINT #1
:L$(L):: NEXT L :: PRINT #1:
 :: GOTO 830 !199
780 FOR L=1 TO S :: FOR L2=1
 TO 3 :: TMP$=SEG$(L$(L),LU(
L2),LU(L2+1)-LU(L2))!207790
IF ASC(TMP$)>32 THEN PRINT #
1:TMP$ !222
800 NEXT L2 !020
810 PRINT #1:SEG$(L$(L),82,3
5): :!236
820 NEXT L !226
830 CLOSE #1 !151
840 RETURN !136
850 !print to screen !083
860 FOR L=1 TO S :: FOR L2=1
 TO 4 :: PRINT SEG$(L$(L),LU
(L2),LU(L2+1)-LU(L2)):: NEXT
 L2 !047
870 PRINT SEG$(L$(L),103,26)
:RPT$("=",28)!240
880 IF L=INT(L/3)*3 THEN PRI
NT :: CALL PAUSE !075
890 NEXT L !226
900 L=L-1 :: IF L<>INT(L/3)*
3 THEN PRINT :: CALL PAUSE !
160
910 RETURN !136
920 REM SAVE LIST !069
930 IF S<1 THEN CALL EMPTY :
: RETURN !067
940 DISPLAY AT(1,1):"Save fi
le..." !007
950 DISPLAY AT(4,1):"Name to
 save file as":D$ :: ACCEPT
AT(5,1)SIZE(-15)VALIDATE(V$)
:D$ !215
960 IF D$="" THEN RETURN !16
9
970 DISPLAY AT(24,7):D$ :: O
PEN #1:D$,DISPLAY ,VARIABLE
132 !144
980 FOR L=1 TO S :: PRINT #1
:L$(L)!135
990 DISPLAY AT(24,1):"Saving
 record #";L :: NEXT L :: CL
OSE #1 :: RETURN !056
1000 REM SORT LIST !094
1010 IF S<2 THEN CALL EMPTY
:: RETURN !068
1020 DISPLAY AT(1,1):"Sort b
y characters starting at #1"
 !027
1030 GOSUB 1550 :: FOR L=1 T
O 8 :: DISPLAY AT(L*2+2,16)S
IZE(2):L :: NEXT L !011
1040 CALL KEYAT(20,16,D,"123
456780"):: IF D=48 THEN RETU
RN !041
1050 D=LU(D-48)!076
1060 DISPLAY AT(22,1):"NOW S
ORTING... BE PATIENT." !062
1070 CALL QUICK3(S,L$(),D)!0
98
1080 RETURN !136
1090 REM FIND ENTRY !141
1100 IF S<2 THEN CALL EMPTY
:: RETURN !068
1110 DISPLAY AT(1,1):"Search
 for what entry?" !090
1120 DISPLAY AT(21,1):"Enter
 no search to find     entry
 by number." !122
1130 GOSUB 1550 :: FOR L=1 T
O 8 :: DISPLAY AT(L*2+2,16)S
IZE(2):L :: NEXT L !011
1140 CALL KEY(5,K,T):: ACCEP
T AT(2,1):S$ :: IF S$="" TH
N GOTO 1230 !235
1150 DISPLAY AT(19,1):"At wh
at position?":TAB(16);1 !199
1160 CALL KEYAT(20,17,D,"123
456780"):: IF D=48 THEN RETU
RN !042
1170 D=LU(D-48)!076
1180 DISPLAY AT(22,1):"NOW S
EARCHING... BE PATIENT." !17
4
1190 C=1 :: LS=LEN(S$):: FOR
 L=1 TO S !169
1200 IF SEG$(L$(L),D,LS)=S$
THEN IF L=C THEN C=C+1 :: GO
TO 1210 ELSE T$=L$(L):: L$(L
)=L$(C):: L$(C)=T$ :: C=C+1
!241
1210 NEXT L :: D=C-1 !101
1220 DISPLAY AT(19,1):"Searc
h complete...":"matches Foun
d:";D:"Items found have been
 moved to the low end of the
 list." :: CALL PAUSE !037
1230 CALL CLEAR :: C=0 !077
1240 C=C+1 :: DISPLAY AT(24,
7):D$: :"DISPLAY RECORD #:";
C:"   Last record in file=L"
 !146
1250 CALL KEY(3,K,T):: ACCEP
T AT(2,18)SIZE(-3)VALIDATE(D
IGIT,"L "):IN$ !229
1260 IF IN$="" THEN RETURN !
252
1270 IF POS(IN$,"L",1)>0 THE
N X,C=S :: DISPLAY AT(2,18)S
IZE(3):C ELSE X,C=VAL(IN$)!2
54
1280 IF C>S THEN CALL SOUND(
100,-1,0):: GOTO 1250 ELSE I
F C=0 THEN RETURN !079
1290 DISPLAY AT(5,1):SEG$(L$
(X),1,25)!106
1300 DISPLAY AT(7,1):SEG$(L$
(X),27,26)!166
1310 DISPLAY AT(9,1):SEG$(L$
(X),54,28)!170
1320 DISPLAY AT(11,1):SEG$(L
$(X),82,19)!213
1330 DISPLAY AT(13,1):SEG$(L
$(X),103,3)!202
1340 DISPLAY AT(15,1):SEG$(L
$(X),107,10)!255
```

(Continued from Page 15)

EXTENDED BASIC—

(Continued from Page 14)

```
1350 DISPLAY AT(17,1):SEG$(L
$(X),118,3)!212
1360 DISPLAY AT(19,1):SEG$(L
$(X),122,8)!214
1370 GOSUB 1550 !100
1380 DISPLAY AT(21,1):"Edit,
 Delete, Next or Menu?":"N"
:: CALL KEYAT(22,1,K,"EDNMed
nm")!058
1390 ON POS("EDNMednm",CHR$(
K),1)GOTO 1460,1410,1240,140
0,1460,1410,1240,1400 !075
1400 RETURN !136
1410 REM delete entry !127
1420 DISPLAY AT(21,1):"Delet
e this entry? N" :: CALL KEY
AT(21,20,K,"NnYy")!128
1430 IF POS("NnYy",CHR$(K),1
)<3 THEN 1380 !174
1440 L$(C)=L$(S):: S=S-1 !17
6
1450 CALL CLEAR :: GOSUB 155
0 :: GOTO 1240 !100
1460 REM edit entry !178
1470 GOSUB 300 !125
1480 GOTO 1380 !184
1490 REM FILE MENU !047
1500 DISPLAY AT(1,1):"Disk d
irectory..." !061
1510 DISPLAY AT(4,1):"Direct
ory of disk number 1?": :"Us
e 0 to return to menu" !078
1520 ACCEPT AT(4,26)SIZE(-1)
VALIDATE("0123"):X !000
1530 IF X=0 THEN RETURN ELSE
 ON ERROR STOP :: CALL CAT(X
):: ON ERROR 1650 :: CALL PA
USE :: RETURN !034
1540 CALL ENDING :: RETURN !
042
1550 REM DISPLAY LABELS !163
1560 DISPLAY AT(4,19):"Last
name" !189
1570 DISPLAY AT(6,18):"First
 name" !051
1580 DISPLAY AT(8,21):"Addre
ss" !041
1590 DISPLAY AT(10,24):"City
" !038
1600 DISPLAY AT(12,23):"Stat
e" !144
1610 DISPLAY AT(14,23):"Zip+
4" !035
1620 DISPLAY AT(16,19):"Area
 code" !208
1630 DISPLAY AT(18,23):"Phon
e" !143
1640 RETURN !136
1650 ! ERRORS SUBROUTINE !17
6
1660 CALL SCREEN(7)!152
1670 DISPLAY AT(2,1)ERASE AL
L:"An error has been caused
by your activities." !059
1680 CALL ERR(AA,BB)!119
1690 IF AA=109 THEN CLOSE #1
 !112
1700 DISPLAY AT(6,1):"Error'
# ";AA !210
1710 IF AA=130 THEN DISPLAY
AT(8,1):"Input/Output Error"
 :: GOTO 1740 !171
1720 IF AA=109 THEN DISPLAY
AT(8,1):"File Error" :: GOTO
 1740 !089
1730 DISPLAY AT(8,1):"Unknow
n Error" !080
1740 CALL PAUSE !232
1750 ON ERROR 1650 !129
1760 CALL CLEAR :: CALL BLUE
 !228
1770 RETURN 210 !035
27000 SUB EMPTY !253
27010 DISPLAY AT(14,1):"You
can't save or print an  empt
y list, nor can you sortor s
earch a list with only  one
entry." !028
27020 CALL PAUSE !232
27030 SUBEND !168
28040 SUB KEYAT(R,C,X,V$)!21
7
28045 ! KEYAT(Row, Column, A
SCII Return variable, Valida
tion string) JLS 2/91 !033
28050 ! Combines cursor flas
h with single key entry, val
idation !111
28055 C=C+2 :: CALL GCHAR(R,
C,N(0)):: N(1)=N(0):: N(2),N
(3)=30 !163
28060 CALL HCHAR(R,C,N(Y-INT
(Y/4)*4)):: Y=Y+1 !209
28065 CALL KEY(0,X,S):: IF S
<1 THEN 28060 !092
28070 IF POS(V$,CHR$(X),1)=0
 THEN IF X=13 THEN X=N(0)ELS
E 28060 !059
28075 CALL HCHAR(R,C,X)!144
28080 SUBEND !168
29160 SUB ENDING !036
29165 !CONFIRMS PROGRAM QUIT
 JLS 9/89 !129
29170 CALL SOUND(800,130,0,1
60,0):: DISPLAY AT(24,3):"PR
ESS SPACE BAR TO QUIT" !105
29175 CALL KEY(0,K,S):: IF S
<1 THEN 29175 ELSE IF K<>32
THEN SUBEXIT !003
29180 STOP :: SUBEND !194
29505 SUB BLUE !149
29510 ! SWITCHES DISPLAY TO
WHITE ON BLUE; JLS 7/88 !230
29515 CALL SCREEN(5):: FOR L
=0 TO 14 :: CALL COLOR(L,16,
1):: NEXT L :: SUBEND !202
30595 SUB MENU(A$,X)!127
30600 ! A$ IS LIST OF OPTION
S, EACH 11 CHARACTERS LONG !
080
30605 ! X : RETURN VARIABLE
FOR NUMBER OF CHOICE !043
30610 CALL CLEAR :: FOR L=1
TO 8 !149
30615 DISPLAY AT(4+L,9):SEG$
(A$,(L-1)*11+1,11):: NEXT L
!050
30620 DISPLAY AT(16,9):"X QU
IT" :: L=L-1 !182
30625 DISPLAY AT(23,9):"CHOI
CE?" !080
30630 V$="AEDFLPSQX"&CHR$(13
):: CALL SOUND(200,-1,4)!055
30635 CALL KEY(3,X,S):: IF S
<1 THEN 30635 !120
30640 X=POS(V$,CHR$(X),1)::
IF X=0 THEN 30635 ELSE CALL
CLEAR !211
30655 SUBEND !168
30820 SUB PAUSE !236
30825 FOR D=1 TO 100 :: NEXT
 D !241
30830 DISPLAY AT(24,2):"PRES
S ANY KEY TO CONTINUE" !088
30835 CALL KEY(0,K,S):: IF S
<1 THEN 30835 !049
30840 SUBEND !168
31565 SUB TITLE2 !035
31575 DISPLAY AT(7,9):"PHONE
 LIST" :: CALL CHAR(95,"00FF
"):: CALL HCHAR(8,11,95,10)!
144
```

(See Page 20)

TEX-COMP
Celebrating Our Tenth Year

FREEWARE••

•• Public Domain and Shareware for the Texas Instruments TI-99/4A Computer.

ONLY **$4.95** Per Disk

Public Domain and Shareware Programs to Meet Your Every Computing Need.

BONUS

FREE DELUXE DISK STORAGE CASE WITH EACH ORDER OF FOUR OR MORE DISKS!!!

#48. GHOSTMAN (from England)
This Pacman/Munchman type game starts at a slow pace and slowly speeds up to a break-neck pace. A totally new experience.

#49. DEMON DESTROYER (from France)
This great assembly game starts where Invaders leaves off. Add features like descending aliens and closing walls. Hours of great arcade action.

#50. OH MUMMY (from Germany)
Move through the chambers of a Pyramid in search of hidden treasure. Fantastic graphics and great entertainment.

#51. BERLIN WALL (from Canada)
This game requires a mine field to be crossed before escaping from E. Berlin. Good graphics and a real challenge.

#52. ANIMATION 99 (from Germany)
THIS IS THE ONE!!! A demo disk filled with computer animation routines like you have never seen before on any computer. See famous cartoon figures move with more realism that on Sat. morning TV. This disk received a standing ovation when previewed at a local users group. We have even included instructions how to do it yourself on the second disk side. This one is a show stopper!!!

#53. HACKER/CRACKER
A collection of disk copying programs that copy TI disks by tracks. If one of these can't copy a protected disk nothing will. We included a collection of the very best ones including both TI and CorComp compatible. These programs require 2 disk drives and 32K of memory.

#54. ASTRONOMY
This program from Australia plots the heavens and teaches you about the solar system. A great learning and reference tool. Exbasic and 32K required. Don't confuse this one with our Astrology demo. They are not the same...ask Nancy!

#55. SCREEN DUMP
This program allows you to dump disk and even module programs to a Star/Epson compatible printer. Comes with easy to follow plans to build a load interrupt switch which is needed to dump module programs. This dump program by Danny Michael is considered the best of the bunch! Complete with documentation.

#56. SPREAD SHEET
OK, it's not Multiplan but it works great and handles many spread sheet applications. A great way to learn to use spread sheet software. Comes with full instructions and documentation.

#57. TELCO
Considered one of the best data communications programs for the TI-99/4A. Complete with documentation.

#58. PR BASE
The alltime most popular and widely used data base program for the TI-99/4A. A freeware program that is widely supported and updated.

#59. GRAPH MAKER
A collection of the best programs for producing graphs and charts from your data. Exbasic and printer.

#60. FREDDY
A fantastic game where you guide the hero through underground passages filled with danger. Nintendo quality, great graphics and fast action. One of the best we have ever seen!!!

#61. THE MINE
A fast action game from F.R.G. that will keep you going for hours. Many screens and skills required.

#62. DISK MANAGER II MODULE BACKUP
The complete TI Disk Manager II on Disk. For legal reasons it is only available to owners of the original module for backup use.

#63. ASTROBLITZ/MAZOG
A pair of great games that continue where Parsec and Munchman leave off. Imagine Parsec with enemy space craft coming from in front and in back of your ship!!!

#64. MAJOR TOM/SPACE STATION PHETA
A pair of great space games. These two are going to keep you in front of the 99/4A for hours. Great!

#65. PERFECT PUSH
An all new space game where you assemble and launch a rocket ship in outer space while avoiding a space monster. This one is professional in very way..graphics. speed and action!!!

#66. HEBREW TYPEWRITER
This program converts your TI-99/4A keyboard into a typewriter that displays Hebrew letters on the screen. Can also be printed when used in conjunction with screen dump program (included). Great for religious training or making your copy of the dead sea scrolls or ten commandments!

#67. GENEALOGY
Now you can set up your family tree and store or print out the records. Great for keeping track of family relationships and records.

#68. CHESS
The original computer chess game Sargon has been reprogrammed for the TI-99/4A. Now play chess with your computer. Documentation included. Exbasic autoload.

#69. COMPUTER PLAYER PIANO/KEY-BOARD CHORD ANALYSIS
A unique music program which displays a piano on the screen and actually plays your selections.

#70. TI RUNNER II
The very latest (and best) "runner" game based on TI Runner and Star Runner. Great action, graphics and entertainment.

#71. KIDS LEARNING II
Two more disk sides loaded with the best in educational programs. Kids improve their math, spelling and comprehension skills while having fun.

#72. CERBERUS
Fantastic space game from Germany. Pilot your ship through narrow and crooked channels in space without colliding. Great graphics and music.

#73. CRYPTO (gram)
One of the best word games we have seen for any computer. Set up like a TV game show with great screen displays.

#74. LABEL MAKER II
Make labels for holidays and special events. You compose the text and select the resident graphics for the occasion.

#75. DISK CATALOGER
Now you can organize your disk files with this great utility. Files, sorts, and prints your records. Easy to use.

#76. PROGRAMMING AIDS AND UTILITIES II
A collection of very useful material. Includes a program to convert basic to exbasic so your old basic programs will load & run in exbasic, even with graphics. Also includes two on screen diagnostic programs to test your keyboard and processor. A great merge utility is also on this disk.

#77. MICROdex 99
A database program by Bill Gaskill which files and retrieves data such as magazine articles. A sample database is included.

#78. ARTCON+ BY RAY KAZMER
ATTENTION GRAPHX AND TI ARTIST USERS!!!
This program lets you convert Exbasic graphics to TI Artist and Graphx pictures. Also contains a new MAC-RLE (2) for converting from Artist to Graphx.

#79. DM1000 V3.5
One of the most popular disk managers for the TI-99/4A. Originally a rip-off of the CorComp manager, it has been improved and refined by talented users all over the world. This version is deemed the most reliable to date and is far advanced over the TI Disk Manager II. Distributed by permission from CorComp.

#80. BIRDWELL DISK UTILITY
A must if you are iunto programming and software development. Besides being a great disk manager, it has provision for copying sectors, comparing files and is menu driven. Complete with documentation.

#81. HOME ACCOUNTING SYSTEM
A complete family & small business accounting system including a checkbook manager, budget analysis, mailing list and an inventory program. Complete with documentation. Easy to modify for specific needs.

#82. CROSSWORD PUZZLES
This program from Australia creates a different puzzle each time you run it. Self contained with definitions and vocabulary taken from a leading crossword dictionary. Great crossword fun.

#83. HOME APPLICATION PROGRAMS
A two disk side collection of useful programs for the home. Includes banking, cooking, home bar guide, utility records, and much much more. Something for everyone.

Send order and make checks payable to **TEX-COMP**

P.O. BOX 33084 — GRANADA HILLS, CA 91344

AUTHORIZED TEXAS INSTRUMENTS DEALER

TEXAS INSTRUMENTS

24 Hour Order Line

(818) 366-6631

VISA MasterCard

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Add 3% shipping and handling ($3.00 Minimum, East of Miss. $5.00 & Can. Add 7% for Credit Card orders. Prices and availability subject to change without notice. We reserve the right to limit quantities.

NOTE: Payment in full must accompany all orders. Credit card, Company check or Money order will be shipped immediately. Personal Checks require up to 4 weeks to clear. California orders add 6% Sales Tax.

TEX★COMP
Celebrating Our Tenth Year

FREEWARE••

•• Public Domain and Shareware for the Texas Instruments TI-99/4A Computer.

Public Domain and Shareware Programs to Meet Your Every Computing Need.

ONLY **$4.95** Per Disk

BONUS

FREE DELUXE DISK STORAGE CASE WITH EACH ORDER OF FOUR OR MORE DISKS!!!

#84. GALACTIC BATTLE/SPY ADVENTURE
A pair of great commercial quality games from EB Software of TI Runner fame. Galactic Battle is a space "trek" type strategy game for one or more players. Spy Adventure is an adventure game that will keep you guessing for hours.

#85. AUTOBOOT UTILITY
This utility which can be installed on a disk loads and runs or displays most files. Now you can have a disk with exbasic programs, Editor Assembler programs and TI Writer files and run or display them all from exbasic.

#86. COLUMN TEXT III V3.2
A very useful utility for printing TI Writer and 99 Writer II files in separate spaced columns. Saves hours in producing a newsletter. Complete with documentation.

#87. ARCHIVER III
This utility allows you to "pack" or combine several files into one for space utilization. A number of boards are sending files packed to save transmission costs. This utility will let you pack and/or unpack these files.

#88. AUSSIE GAMES VOL 1
A collection of games from our friends down under. Includes a great card game and board game. Hours of fun and entertainment. Includes Matchmaker & TILO.

#89. PROCALC
This is an on screen calculator for decimal/hexidecimal conversions and much more. A must for the serious programmer.

 # 90. JET CHECKBOOK MANAGER
This checkbook manager is considered the ultimate with every feature you can think of for keeping track of your checking account and keeping records of your spending for budget and tax purposes. Complete with documentation.

#91. "THE MAZE OF GROG"(St. Valentine)
Ray Kazmer has created a great maze game with fantastic graphics and the characters from his now legendary "Woodstock" disk. Fun for all!!!

#92. HOUSEHOLD INVENTORY
Written by 99/4 programming great Charles Ehninger, this prize winner originally sold for $59.95. Keeps track of household, business or personal items by category and provides automatic updating for inflation etc. A must for tax and insurance records!

#93. THE 1991 KBGB GIRLIE CALENDAR
This latest offering from programming master Ken Gilliland prints out a jumbo 12 month calendar with a knock-out centerfold pinup for each month. If you like our #14 Figure Study disk, you will flip over this one. For Adults Only!! Exbasic & d/m printer.

#94. GREAT 99/4A GAMES VOL. III
If you have seen vols. 1 & 2 of this series you know we only provide the very best. This latest volumn is also filled with a collection of great ones!

#95. WEATHER FORECASTER
The weather predictions are amazingly reliable and accurate! A great game "Lawnmower" and a mini database are also included to make this disk a fantastic value.

#96. STATISTICS & SORTING
Two great assembly utilities by John Clulow. STAT is a set of statistic routines for use in exbasic. SORT allows sorting by two separate fields and a choice of two types of sorts.

#97. MEMORY MANIPULATOR
This powerful utility lets you explore the entire memory in your 99/4A system and take apart what you find. User friendly!

#98. DAYS OF EDEN & DOORS OF EDEN
Two bible games)non-fiction) that work with the TI Adventure Module.

#99. GREAT 99/4A GAMES VOL. IV
This disk features the works of J. Peter Hoddie. All of these games are of commercial qualaity and well worth the donation requested!

#100. ASSULT THE CITY (T. of DOOM)
An exciting game for use with the Tunnels of Doom module. Several Exbasic bonus games are included.

#101. ENCHANCED DISPLAY PACKAGE
This screen enhancement utility lets you do 40 columns, windowing, reverse scrolling, clock/alarm, and a whole host of other great tricks in exbasic. Fully documented.

#102. COLOSSAL CAVES ADVENTURE
This classic adventure now available for the 99/4A is what led to the Zork series. Hours of text adventuring.

#103. SORGAN, THE 99/4A ORGAN
This program which is currently selling for big bucks on module turns your 99/4A into an electronic organ. Sound effects, different instruments and voices, chord forms, color graphics with complete control of all.

#104. C99 COMPILER AND LIBRARY
This two-sided (flippy) disk gets you into C programming with your 99/4A. Comes with a great collection of utilities such as text & graphics. (E/A)

#105. KING'S CASTLE+
A great arcade style assembly game formerly offered on module. Also includes an EB "Trek" game and a collection of sprite & graphics from Tigercub's Jim Peterson.

#106. QUEST (Dungeons & Dragons)
One of the best D&D games around! You must destroy the Dark Lord to free your homeland! Complete with documentation on disk.

#107. STAR TREK MUSIC ALBUM
Ken Gilliand's music and graphics version of the TV theme and the three motion pictures. (Exbasic)

#108. FUNLPLUS BY JACK SUGHRUE
Fantastic disk packed with Funnelweb (#42) templates, utilities and prog. to augment and configure Funnelweb. Unbeliveable collection of fantastic aids to make the best even better!

#109. TI-WRITER MINI MANUAL
This disk prints out a five page TI Writer manual with everything you need to know to use TI Writer or the many clones such as 99Writer II. Additional aids for using this powerful word processor are included.

#110. DISK + AID
A powerful disk sector editor formerly sold for $20. Menu Driven and easy to use.

#111. POP MUSIC & GRAPHICS
This exciting disk from Germany features music/graphics written in 100% assembly and what comes from the TI sound chip is sure to astound you.

#112. INVOICE PACK
An excellent invoice preparation and printing program with instructions on how to modify it for your own business.

#113. LABEL MAKER 3
A collection of label programs to create mailing and disk envelopes, disk labels and much more!

#114. PANORAMA
A drawing and illustration program that compliments Graphx and TI Artist. A must for the serious 99/4A artist!

#115. GRAPHICS DESIGN SYSTEM
A complete system for creating graphic screens in full color for your programs by J. Peter Hoddie. Fully documented.

#116. FOURTH TUTORIAL
A lesson in FORTH programming on how to create graphics.

#117. UNIVERSAL DISASSEMBLER
This powerful utility written in Forth allows disassembly of programs off disk in any format, in memory, and even off of P-Box cards. Very complete with some very unique features.

#118. FAST TERM
One of the most popular and recommended of the 99/4A terminal emulator programs. Supports TE-II, ASCII, and X-Modem transfers, print spooling and more. Loads from Exbasic or E/A.

#119. RAG LINKER
A utility for converting DIS/FIX 80 assembly object code files to PROGRAM image. This allows files to load faster and take up less space on disk. Full Doc

#120. BITMAC
The original BITMAC is now available at $4.95 with all original documentation. A powerful graphics program for the 4A which lets you print where you want..even over pre-existing text. Create great graphics in 16 colors, print text sideways, mirror image, upside down etc. etc. A **must** for anyone into 99/4A graphics. Comes with second bonus disk with utilities such as sign & banner makers. Even can computer generate your own signature!

#121. SUPER YAHTZEE & WHEEL II
If you like Yahtzee this disk is for you. A great version written in high speed assembly. Also included is another version of Wheel of Fortune which also lets you create your own puzzles with a puzzle edit program included.

#122. ADULT ADVENTURE
A trily adult adventure for use with the TI Adventure Module. Also included is a bonus adventure (not adult) "LOST GOLD" which is one of the better ones we have seen recently.

Send order and make checks payable to **TEX★COMP**

P.O. BOX 33084 — GRANADA HILLS, CA 91344

TERMS. All prices F.O.B. Los Angeles. For fastest service use cashier's check or money order. Add 3% shipping and handling ($3.00 Minimum). East of Mississippi add 5%. Add 3% for Credit Card orders. Prices and availability subject to change without notice. We reserve the right to limit quantities.

AUTHORIZED DEALER
TEXAS INSTRUMENTS

VISA MasterCard

24 Hour Order Line
(818) 366-6631

NOTE: Payment in U.S. must accompany all orders. Credit card, company check or money order for immediate shipment. Personal Checks require up to 4 weeks to clear. California orders add 6% for sales tax.

EXTENDED BASIC—

(Continued from Page 15)

```
31580 DISPLAY AT(12,2):"Name
 and Address Database" !094
31585 DISPLAY AT(18,5):"Jerr
y L. Stern 1991" !056
31595 SUBEND !168
32070 SUB QUICK3(N,X$(),D)!1
09
32075 !(NUMBER OF VALUES,STR
ING ARRAY TO BE SORTED,DIGIT
 TO SORT BY) !194
32080 ! MODIFIED FOR MIDSTRI
NG SORT JLS 4/85 !029
32085 K,I=0 :: E=133-D :: DI
M S(70)!232
32090 S(I+1)=1 :: S(I+2)=N !
207
32095 K=K+1 !015
32100 IF K=0 THEN SUBEXIT !2
21
32105 K=K-1 :: I=K+K !240
32110 A=S(I+1):: B=S(I+2)!01
0
32115 Z$=X$(A):: U=A :: L=B+
1 !197
32120 L=L-1 !018
32125 IF L=U THEN 32150 !111
32130 IF SEG$(Z$,D,E)<=SEG$(
X$(L),D,E)THEN 32120 ELSE X$
(U)=X$(L)!060
32135 U=U+1 !035
32140 IF L=U THEN 32150 !111
32145 IF SEG$(Z$,D,E)>=SEG$(
X$(U),D,E)THEN 32135 ELSE X$
(L)=X$(U):: GOTO 32120 !027
32150 X$(U)=Z$ !122
32155 IF B-U>=2 THEN I=K+K :
: S(I+1)=U+1 :: S(I+2)=B ::
K=K+1 !080
32160 IF L-A>=2 THEN I=K+K :
: S(I+1)=A :: S(I+2)=L-1 ::
K=K+1 !061
32165 GOTO 32100 !048
32170 SUBEND !168
32175 SUB CAT(X)! SUBPROGRAM
 EQUIVALENT OF DIR(DISK NUMB
ER TO READ DIRECTORY FROM) !
013
32180 DIM T$(5):: T$(1)="DIS
/FIX" :: T$(2)="DIS/VAR" ::
T$(3)="INT/FIX" :: T$(4)="IN
T/VAR" :: T$(5)="PROGRAM" !0
34
32185 OPEN #5:"DSK"&STR$(X)&
".",INPUT ,RELATIVE,INTERNAL
 :: INPUT #5:A$,J,J,K !182
32190 PRINT "DSK1  DISKNAME=
";A$:"AVAILABLE=";K;"USED=";
J-K:"FILENAME   SIZE TYPE
    P":"__________ ____ ____
______ _" !232
32195 FOR L=1 TO 127 :: INPU
T #5:A$,A,J,K !190
32200 IF LEN(A$)=0 THEN CLOS
E #5 :: SUBEXIT !086
32205 PRINT A$;TAB(12);J;TAB
(17);T$(ABS(A));!094
32210 IF ABS(A)=5 THEN 32220
 ELSE B$=" "&STR$(K)!099
32215 PRINT SEG$(B$,LEN(B$)-
2,3);!194
32220 IF A>0 THEN PRINT ELSE
 PRINT TAB(28);"Y";!013
32225 NEXT L :: CLOSE #5 ::
SUBEND !041
```

BASIC Assembly

Accessing built-in assembly routines

By BARRY A. TRAVER
©1991 B. Traver

So far we've been writing our own assembly routines for use in TI Extended BASIC (or letting utilities like GRAPHICOMP, VDP/SAVER, or TIA/LINK write them for us!), but I have some good news for you: your TI has built-in assembly routines that you can access and that can save you a lot of time and effort.

There are, in fact, three categories of such routines: DSRLNK (i.e., Device Service Routines to which you can LiNK), GPLLNK (Graphics Programming Language routines in GROM to which you can LiNK), and XMLLNK (ROM routines to which you can LiNK). (If I knew what XML stood for, I'd tell you; my best guess at the moment is that it has something to do with eXpanded Memory, but that is only a guess.)

DSRLNK links your assembly language program to built-in Device Service Routines, e.g., a routine that may access a sector on your disk drive. GPLLNK lets you use built-in miscellaneous routines written in Graphics Programming Language to do such things as load in certain character sets, give a BEEP or HONK tone, etc. Many of the built-in XMLLNK routines have to do with different types of mathematical operations and various conversions. Altogether, these "Extended Utilities" make up a very useful library, often letting you do what would otherwise be difficult or impossible in Extended BASIC.

Programs running in TI Extended BASIC do have one disadvantage over programs running in other environments: although TI Extended BASIC includes support for XMLLNK, it does not include support for GPLLNK or DSRLNK. Don't panic, however. That deficiency is rectified by the DSRLNK and GPLLNK routines included with this article! We are grateful to Doug Warren in particular for permission to make these routines (which were first published in the July 1986 issue of The Smart Programmer) more generally available once again.

In addition to his authorizing re-publication of the source code in MICROpendium, Doug says that it's okay for you to use the routines in your own personal TI Extended BASIC programs. In addition, you may share such TI XB programs (with these GPLLNK and DSRLNK routines "embedded" in the XB programs, making use of Todd Kaplan's ALSAVE, as discussed in the September 1990 issue of MICROpendium) with others, as long as you aren't charging anything for the programs.

If, on the other hand, you want to make commercial use of the routines, you do

(See Page 21)

BASIC/ASSEMBLY—

(Continued from Page 20)

need to contact Doug directly concerning obtaining permsission to do so. (See DSR&GPL/S for his address.) In my opinion (and I think many others would agree), the source code for DSRLNK and GPLLNK from Doug is the best version I know of anywhere around. Similar code is available from other sources (e.g., you can get such code directly from Texas Instruments if you ask), but I don't know any other that works from virtually any TI-99/4A environment and is so efficiently written (taking up only 186 bytes of code!).

You won't get much help from the Editor/Assembler manual on the use of DSRLNK other than a few comments on page 262. (I suspect that TI kept most of the information in a "proprietary" manual to keep people from learning the secrets of doing such things as unprotecting copy-protected software, but I may be wrong on that.) The E/A manual does give more time and space to DSRLNK (pages 251-257) and to XMLLNK (pages 257-261). Caution, however: not all of the routines discussed in the E/A manual are necessarily available in TI Extended BASIC.

USING DRSLNK, GPLLNK, XMLLNK

The purpose of this month's column is to introduce the use of DSRLNK, GPLLNK, and XMLLNK, whereas next month I hope to go into more detail on some of the built-in routines available to you in Extended BASIC using TI's "Extended Utilities."

Of all the different things I have written, it seems that "RAW" seems to have been the one used most often by other programmers. The "RAW" routines allow you to "Read And Write" individual sectors on a disk by direct access from Extended BASIC. John Johnson, Richard Mitchell, and others have made use of "RAW" in their programs. I can't boast too much about the success of "my" program, however, because I couldn't have written "RAW" without a lot of help from other TI'ers, including Michael Riccio, Todd Kaplan, Mack McCormick, Paul Charlton, and Chris Faherty! Now you understand my conviction that our TI community will continue to be strong if we continue to share with one another, learn from one another, and teach one another.

"RAW" IS REVISED

For the benefit of MICROpendium readers and others, I have thoroughly revised the source code for "RAW." The biggest change perhaps is that it now makes use of GET/SEND/S (introduced in the June 1990 issue of MICROpendium), but there have been other significant modifications as well. "RAW" now allows for drives 1-9 (instead of 1-4) and for sectors 0-2879 (instead of 0-1439, for the benefit of those who have quad-density drives). Last, not least, is this: in revising the code, I have attempted to make it simultaneously more compact and more readable.

I hope you will find "RAW" a useful utility, but I also hope that you will find it instructive as well. Note, for example, the use of error checking and responding with error messages, something we have not done before in this column. "RAW" makes use of DSRLNK, whereas BEEPHONK/S and QUIT/S make use of GPLLNK. BEEP and HONK are more illustrative than useful: all they do is make your computer beep or honk. Likewise, QUIT is the assembly equivalent of FCTN-QUIT. That's no big deal, either.

The purpose of the source code for those simple routines is to show in barest form the way in which DSRLNK, GPLLNK, and XMLLNK work: after you set things up as necessary (putting appropriate values in registers or proper memory locations as, or if, required), you do a BLWP @xxxLNK and follow that up with a DATA >xx on the next line. That tells DSRLNK, GPLLNK, or XMLLNK which built-in routine you want to access.

To call BEEP, HONK, or QUIT from Extended BASIC, simply load the assembly code and then use the relevant command: CALL LINK("BEEP"), CALL LINK("HONK"), or CALL LINK("QUIT"). To "read and write" the contents of a sector from/onto a disk, likewise load the assembly code and then use as appropriate command: CALL LINK("READ",DRIVE,SECTOR,A$,B$); or CALL LINK("WRITE",DRIVE,SECTOR,A$,B$). (The contents of the sector have to be put in two strings rather than one, because the maximum size of a string is 255 bytes, whereas the contents of a sector is 256 bytes.)

As I said, I'll tell you more next time about some specific "Extended Utilities" that are available to you (and I'll try to suggest some practical applications). Let me close this month's column, however, with a specific warning about "RAW": although the routine is tremendously powerful and useful, it is also potentially destructive if not handled properly. When experimenting with "RAW," I strongly advise that you don't have in your drives any disks that you can't afford to lose. Like a sector editor (and you could write your own XB sector-editor program using "RAW"!), "RAW" is a tool that must be handled with care. My advice for you (until you feel comfortable with "RAW"): have lots of fun playing with it — "Reading And Writing" with abandon — as long as you are doing it with a disk that you can afford to abandon (or at least reinitialize when you're finished)! Until next TIme, keep on compuTIn'!

BEEPHONK/S

```
* BEEPHONK/S
        COPY  "DSK1.GPL&DSR/S"
BASIC   EQU   >006A
        DEF   BEEP,HONK
WS      BSS   32
BEEP    LWPI  WS
        BLWP  @GPLLNK
        DATA  >34
        LIMI  2
        LIMI  0
        B     @RETURN
HONK    LWPI  WS
        BLWP  @GPLLNK
        DATA  >36
        LIMI  2
        LIMI  0
        B     @RETURN
RETURN  LWPI  GPLWS
        B     @BASIC
        END
```

GPL&DSR/S

```
*--------------------------------*
* Universal GPLLNK and DSRLNK, by  *
*  Doug Warren and Craig Miller.   *
*                                  *
* This code is reprinted in MICRO- *
*  pendium by permission (see the  *
*  July 1986 issue of THE SMART    *
*  PROGRAMMER for the original     *
*  fully commented version).       *
*                                  *
* Feel free to make personal use of *
*  these routines.  Also, you may  *
*  embed them in TI Extended BASIC *
```

(See Page 22)

BASIC ASSEMBLY—

(Continued from Page 21)

```
*  programs that you share with     *
*  others, provided that no charge  *
*  is made for such programs.       *
*                                   *
* Anyone wanting to make commercial *
*  use of the routines or republish *
*  the source code should contact   *
*  Doug Warren for permission to do *
*  so.  The address:  Doug Warren,  *
*  10349 Redwood Blvd., California  *
*  City, CA 93505.                  *
*                                   *
* Universal GPLLNK:                 *
*                                   *
* Use the same way as you would the *
*  E/A GPLLNK, e.g.,                *
*                                   *
*        BLWP @GPLLNK               *
*        DATA >34                   *
*                                   *
* Do not REF GPLLNK when using this *
*  routine in your code.            *
*                                   *
* Universal DSRLNK:                 *
*                                   *
* Use the same way as you would the *
*  E/A DSRLNK, e.g.,                *
*                                   *
*        BLWP @DSRLNK               *
*        DATA 8                     *
*                                   *
* Do not REF DSRLNK when using this *
*  routine in your code.            *
*                                   *
* 186 bytes incl. GPLLNK, DSRLNK,   *
*  and both Workspaces.             *
*-----------------------------------*

* IMPORTANT:  REMOVE THE FOLLOWING
* GPLWS EQUATE IF IT IS ALREADY
* PRESENT IN ANOTHER FILE (E.G.,
* "DSK1.GET/SEND/S).

GPLWS  EQU  >83E0

GR4    EQU  >83E8
GR6    EQU  >83EC
STKPNT EQU  >8373
LDGADD EQU  >60
XTAB27 EQU  >200E
GETSTK EQU  >166C

GPLLNK DATA GLNKWS
       DATA GLINK1

RTNAD  DATA XMLRTN
GXMLAD DATA >176C
       DATA >50

GLNKWS EQU  $->18
       BSS  >08

GLINK1 MOV  *R11,@GR4
       MOV  *R14+,@GR6
       MOV  @XTAB27,R12
       MOV  R9,@XTAB27
       LWPI >83E0
       BL   *R4
       MOV  @GXMLAD,@>8302(R4)
       INCT @STKPNT
       B    @LDGADD
XMLRTN MOV  @GETSTK,R4
       BL   *R4
       LWPI GLNKWS
       MOV  R12,@XTAB27
       RTWP

*-----------------------------------*
* IF YOU PLAN ONLY TO MAKE USE OF   *
* GPLLNK (AND NOT DSRLNK), YOU CAN  *
* OMIT THE REMAINDER OF THIS FILE.  *
* IF YOU PLAN ONLY TO MAKE USE OF   *
* DSRLNK, THE ENTIRE FILE IS STILL  *
* REQUIRED (SINCE DSRLNK MAKES USE  *
* OF GPLLNK).                       *
*-----------------------------------*

* DSRLNK

PUTSTK EQU  >50
TYPE   EQU  >836D
NAMLEN EQU  >8356
VWA    EQU  >8C02
VRD    EQU  >8800
GR4LB  EQU  >83E9
GSTAT  EQU  >837C

DSRLNK DATA DSRWS,DLINK1

DSRWS  EQU  $
DR3LB  EQU  $+7
DLINK1 MOV  R12,R12
       JNE  DLINK3
       LWPI >83E0
       MOV  @PUTSTK,R4
       BL   *R4
       LI   R4,>11
       MOVB R4,@>402(R13)
       JMP  DLINK2
       DATA 0
       DATA 0,0,0
DLINK2 MOVB @GR4LB,@>402(R13)
       MOV  @GETSTK,R5
       MOVB *R13,@DSRAD1
       INCT @DSRADD
       BL   *R5
       LWPI DSRWS
       LI   R12,>2000

DLINK3 INC  R14
       MOVB *R14+,@TYPE
       MOV  @NAMLEN,R3
       AI   R3,-8
       BLWP @GPLLNK
DSRADD BYTE >03
DSRAD1 BYTE >00

       MOVB @DR3LB,@VWA
       MOVB R3,@VWA
       SZCB R12,R15
       MOVB @VRD,R3
       SRL  R3,5
       MOVB R3,*R13
       JNE  SETEQ
       COC  @GSTAT,R12
       JNE  DSREND
SETEQ  SOCB R12,R15
DSREND RTWP
```

QUIT/S

```
* QUIT/S
       COPY "DSK1.GPL&DSR/S"
BASIC  EQU  >006A
       DEF  QUIT
WS     BSS  32
QUIT   LWPI WS
       BLWP @GPLLNK
       DATA >20
       B    @RETURN
RETURN LWPI GPLWS
       B    @BASIC
       END
```

RAW/S

```
* SINGLE-SECTOR DISK ACCESS PROGRAM BY BARRY TRAVER,
*  835 GREEN VALLEY DRIVE, PHILADELPHIA, PA 19128.
* PUT TOGETHER "WITH A LOT OF HELP FROM MY FRIENDS,"
*  MICHAEL RICCIO, TODD KAPLAN, AND MACK McCORMICK
*  (NOT TO MENTION PAUL CHARLTON AND CHRIS FAHERTY).
* THIS VERSION IS COPYRIGHT 1991 BY BARRY A. TRAVER.

       COPY "DSK1.GET/SEND/S"
       COPY "DSK1.GPL&DSR/S"

       DEF  READ,WRITE   program names

DRIVE  EQU  PARAM1       name
SECTOR EQU  PARAM2            all
A$     EQU  PARAM3                the
B$     EQU  PARAM4                    parameters

FLAG   DATA 0            set up
READF  DATA 1                  for
WRITEF DATA 2                      read/write flag
```

(See Page 23)

BASIC/ASSEMBLY—

(Continued from Page 22)

```
LB128  BYTE 128,0          left byte = 128

DSKSEC DATA >0110          sector subroutine "name"

READ   LWPI WS             load workspace registers
       BL   @GET           get parameters from XB
       MOV  @READF,@FLAG   set flag to "read"
       JMP  PART1          jump to part1

WRITE  LWPI WS             load workspace registers
       BL   @GET           get parameters from XB
       MOV  @WRITEF,@FLAG  set flag to "write"

PART1  MOV  @SECTOR,R8     check on sector number
       CI   R8,0           is sector number less than 0?
       JLT  ERROR1         if so, jump to error1 message
       CI   R8,2879        is sector number greater than 2879?
       JGT  ERROR1         if so, jump to error1 message

       C    @FLAG,@WRITEF  is flag "write"?
       JNE  PART2          if not, jump to part2

WRITE1 CB   @A$,@LB128     is first string length 128?
       JNE  ERROR2         if not, jump to error2 message

       LI   R0,>3CEF       address of vdp ram buffer
       LI   R1,A$+1        address of cpu ram buffer
       LI   R2,128         length of string to send to vdp
       BLWP @VMBW          pass first string to vdp buffer

       CB   @B$,@LB128     is second string length 128?
       JNE  ERROR2         if not, jump to error2 message

       AI   R0,128         create new vdp buffer address
       LI   R1,B$+1        address of cpu ram buffer
       LI   R2,128         length of string to send to vdp
       BLWP @VMBW          pass second string to vdp buffer

PART2  LI   R0,>03C0       address of vdp temp roll out area
       MOV  R0,@>8356      pointer to name location in vdp ram
       LI   R1,DSKSEC      address of cpu subroutine "name"
       LI   R2,2           length of string to send to vdp
       BLWP @VMBW          pass string to vdp temp roll out area
       MOV  @DRIVE,R1      check on drive number
       CI   R1,1           is drive number less than 1?
       JLT  ERROR1         if so, jump to error1 message
       CI   R1,9           is drive number greater than 4?
       JGT  ERROR1         if so, jump to error1 message

       SWPB R1             put drive number in left byte

       C    @FLAG,@READF   is flag "read"?
       JNE  PART3          if not, jump to part3

READ1  AI   R1,>0001       set for read rather than write

PART3  LI   R2,>3CEF       address of vdp buffer
       CLR  @FAC           clear Floating/point ACcumulator
       MOV  @SECTOR,@FAC+6 set sector number
       MOV  R1,@FAC+2      set drive number/read or write
       MOV  R2,@FAC+4      set address of vdp buffer
       BLWP @DSRLNK        perform device service routine
       DATA >A

       MOVB @>8350,R1      is there a read/write error?
       JNE  ERROR3         if so, jump to error3 message

       C    @FLAG,@WRITEF  is flag "write"?
       JNE  READ2          if not, continue
       B    @RETURN        if so, jump to return

READ2  MOVB @LB128,@A$     set length of first string
       LI   R0,>3CEF       address of vdp buffer
       LI   R1,A$+1        address of cpu buffer
       LI   R2,128         length of string in vdp to read
       BLWP @VMBR          pass first string to cpu buffer

       MOVB @LB128,@B$     set length of second string
       AI   R0,128         create new vdp buffer address
       LI   R1,B$+1        address of cpu buffer
       LI   R2,128         length of string in vdp to read
       BLWP @VMBR          pass second string to cpu buffer

       B    @SEND          send parameters back to XB

ERROR1 LI   R0,>1E00       "bad value" error message
       BLWP @ERR           report error
ERROR2 LI   R0,>1C00       "bad argument" error message
       BLWP @ERR           report error
ERROR3 LI   R0,>2100       "data error" error message
       BLWP @ERR           report error

       END
```

THE ART OF ASSEMBLY—PART 3

Starting at the top

By BRUCE HARRISON

In Part 2 of this series, we discussed and showed some small "primitive" subroutines and the methods for nesting them. In this article, we are going back to the "Top Down" part of writing Assembly programs. We will use for our example the Harrison Golf Score Analyzer, since its development went pretty much along the lines we're trying to encourage.

One of the first decisions you should make is how the user will interact with your program. In many games, for example, the principal means of interaction is the joystick. In a program like a Golf Score Analyzer, however, that would be a very poor interface for user input. Our preference is for simple menu interaction at the top level, so each main function of the program is readily apparent to the user, and selection of a function is just one keystroke away. In today's world of "Graphical User Interface" (GUI), where functions are represented by pictures, not words, this makes us very old-fashioned, but we do have a reason for being that way. GUIs normally require a mouse to select options, and one can't count on every customer having a mouse. Further, a mouse can't be used to input names, numbers, and other data, so with or without a mouse one still needs the keyboard. Our choice has been to require *only* the keyboard, and that makes "plain English" menus the natural choice for selecting functions.

Given that, we must make a decision as to what functions belong on the main menu. In the Golf Score Analyzer, we settled on eight functions for the Main Menu, and made each require only a single keystroke to select. The eight look like this:

1 ADD ROUNDS
2 LOAD FILE
3 DELETE DATA
4 ANALYZE DATA
5 SAVE FILE
6 ADD/EDIT COURSES
7 REVIEW COURSES
8 EXIT PROGRAM

It's important to always include an exit selection, so the user can easily get out of your program when he wants to. It's equally important to make it difficult or impossible to get out of the program by accident. In this program, selecting item 8 from the Main Menu is the *only* way to get out. We made Funtion-Quit inactive in this program. As an aside, when users are looking at subsidiary menus, Function-9 (BACK) will get back to the previous menu, but that will not get them out of the program.

In this particular program, we had a special reason for making one and only one exit point. When the user selects item 8, we perform a check to see whether the user has modified the file currently in memory. If he's not made any changes to the file, or if he's saved it since making changes, we simply return him to either XB or E/A, depending on how he entered the program. If changes have been made, we produce a prompt asking whether he'd like to save the changed file before exiting. Any answer other than N or n is taken as Yes, and he's placed in the SAVE FILE function. We take these precautions as part of our concept of "User Friendliness".

Perhaps we could illustrate the concept of "user friendliness" by an example drawn from experience. In many instances on the TI, one will encounter an error in execution of some program. Let's say we're working in XB or E/A, and try to get a nonexistent file to open for INPUT. What the TI folks will give you is a numbered "ERROR CODE," which you'll have to look up in a book. When we write our own programs, we like to provide a more definitive error indication, like "THAT FILE DOES NOT EXIST ON DRIVE x" or "THERE IS NO DISK IN DRIVE x." This way the user has a definite idea of what's wrong. Doing this, of course, eats memory, since those error messages have to be stored somewhere in the computer and printed to the screen, but we think that's a worthwhile use of memory.

But we digress. Once one has decided upon a menu, the top part of the flow chart is readily apparent. There will need to be an opening section of code that sets up such things as screen mode, color scheme, and such, then displays our copyright notice. Next is a delay loop so the user can read the copyright notice, and then we clear the screen and produce the main menu. Here we had to make a decision. Since we knew there would be more than one menu, we could have each menu produced by a separate section of code, or we could provide a "Menu Driver" section of code that would produce all the required menus simply by using different data with the same code. We chose the latter, and believe that was a wise decision, because we used less memory to do it this way. Our word processor, which we use to prepare these articles, also has a central menu driver, but the one in the Golf Score Analyzer is better, taking lessons learned from the WP program into account.

Each menu we use has a section of data associated with it, which includes the title for the top of that menu, the selections, and a "branching" lookup table, which indicates where the program will go to when it exits that menu. The legend "SELECT BY NUMBER" goes at the bottom of each menu, so the menu driver itself places that legend on each menu it displays. In our Golf Score Analyzer, by the way, we separated the code from the data into sections of memory. That is, all the executable instructions are together in a block of memory, then all the data, including text for messages and menus, is in its own block of memory. This makes a somewhat neater arrangement for the programmer, in that separate source files contain the data, and it becomes a bit easier to keep track of what one is doing while developing the program. It also makes it easier when one comes back six months later to change something in the program.

Actually, there's no reason you can't scatter data all over the place, between sections of the executable code, but our thinking on the subject has been colored by the fact that we also program in PC Assembly language, where different memory segments are

(See Page 25)

THE ART OF ASSEMBLY—

(Continued from Page 24)

(and must be) allocated for code and data. This becomes a habit that carries over to the TI.

There we are digressing again. Just for the heck of it, let's look at some of the source code. In the sidebar is the annotated source code associated with the menu driver for the Golf Score Analyzer. The first two executable lines are the required setup before branching to the driver. These lines set R9 to point to the data for the menu itself, and R13 to point to the lookup table for branching out of the Main Menu.

In the Driver itself (MENDRV), the first order of business is to clear the screen. The CLS subroutine is similar to the one shown in our last article, except that, since GSA was written to operate from Extended Basic, it adds an offset of >60 to the spaces it writes into SCRLI. As an ironic sidelight, we later added a loader so that GSA could be run from E/A, and in that loader we had to, among other things, re-arrange the tables in VDP so it would need the character offset.

Before delving further into the code, let's look at the structure of the data for the menu, at label MENDAT. It starts with a byte giving the length of the title for the menu. Next is the text of the title, then two bytes. The first of these is the number of items in this menu (8), and the second is the length of the first item description 13). After this is the text for the first item, followed by the strings for the rest of the items (a length byte, then text content). By organizing the data this way, we can make a loop in the menu driver that minimizes the memory used for the driver's code.

The business of getting the menu on-screen now proceeds by taking the length of the title line and manipulating that to position R0 so the title appears centered on Row 2 of the screen. The subroutine DISLI could also be called DISSTR, since what it does is take a string pointed to by R9 and display it at the screen location pointed to by R0. Another irony here is that, had we done this in E/A only, we could have used R1 as the pointer to the string, then DISLI would reduce to:

```
DISLI   MOVB    *R1+,R2     Get length byte into R2
        SRL     R2,8        Right justify R2
        BLWP    @VMBW       Write characters to screen
        A       R2,R1       Advance R1 beyond text
        RT                  Return
```

But we didn't do that, because we wanted GSA to be available to those who don't have the E/A module, but only the XB module. Thus we're stuck with that offset, even when the user enters the program from E/A. Live and Learn!

Another small note before we examine the rest of the source code. There is no such thing as a perfect program. As your author looks at his own sidebar, he can see several places where it could be improved. For ex-

(See Page 26)

CODE FROM GOLF SCORE ANALYZER

```
* PORTIONS OF SOURCE CODE FROM GOLF SCORE ANALYZER
*
* EQUATE FOR 32 CHARACTER SCREEN
SCRWID EQU  32
*
* SETUP FOR ENTERING MENU DRIVER TO MAKE MAIN MENU
       LI   R9,MENDAT
       LI   R13,MAINBR
       B    @MENDRV
*
*
* MENU DRIVER SOURCE CODE STARTS
MENDRV
       BL   @CLS          CLEAR THE SCREEN
       LI   R0,SCRWID     SET R0 TO SCREEN WIDTH
       MOVB *R9,R1        GET LENGTH OF TITLE IN R1
       SRL  R1,8          RIGHT JUSTIFY LENGTH
       S    R1,R0         SUBTRACT LENGTH FROM SCREEN WIDTH
       SRL  R0,1          CUT THAT NUMBER IN HALF
       AI   R0,SCRWID     ADD ONE SCREEN WIDTH
* THE ABOVE SECTION SETS R0 AT A VALUE WHICH WILL AUTO-CENTER THE TITLE
* IN ROW 2 OF THE SCREEN
       BL   @DISLI        DISPLAY THAT LINE OF TEXT
* DISLI ADVANCES R9, SO IT NOW POINTS TO BYTE BEYOND END OF TITLE'S TEXT
       MOVB *R9+,R8       GET NUMBER OF ITEMS FOR MENU
       SRL  R8,8          RIGHT JUSTIFY IN R8
       MOV  R8,@NOITEM    STASH THE NUMBER OF ITEMS AS DATA
       LI   R0,8          LOAD R0 WITH MAXIMUM NUMBER OF ITEMS IN ANY MENU
       S    R8,R0         SUBTRACT THE NUMBER OF ITEMS
       AI   R0,4          ADD FOUR
       LI   R3,SCRWID     GET R3 TO EQUAL NUMBER OF CHARACTERS IN SCREEN WIDTH
       MPY  R3,R0         MULTIPLY BY WIDTH OF SCREEN
       AI   R1,8          ADD 8 FOR COLUMN POSITIONING
       MOV  R1,R0         PLACE THIS NUMBER IN R0
* THE CODE ABOVE SETS R0 TO VERTICALLY CENTER THE NUMBER OF ITEMS IN THE MENU
* FOR A MORE CONSISTENT SCREEN APPEARANCE.
MEN1   BL   @DISLI        DISPLAY A LINE OF THE MENU
       AI   R0,SCRWID*2   MOVE DOWN-SCREEN BY TWO LINES
       DEC  R8            DECREMENT COUNTER FOR NUMBER OF ITEMS
       JNE  MEN1          IF NOT ZERO, JUMP BACK TO DISPLAY NEXT ITEM
       LI   R0,22*SCRWID+8  SET R0 FOR ROW 23, COLUMN 9
       LI   R9,SELEC      POINT TO STRING FOR "SELECT BY NUMBER"
       BL   @DISLI        DISPLAY THAT LEGEND
KYIN   BL   @KEYLOO       GET A KEYSTROKE
       CI   R8,15         WAS FUNCTION-9 STRUCK?
       JNE  ACC1          IF NOT, JUMP AHEAD
       MOV  @NOITEM,R5    ELSE PUT NUMBER OF ITEMS IN R5
       JMP  ACC2          THEN JUMP
ACC1   MOV  R8,R5         PLACE KEYSTROKE IN R5
       S    @NUMASK,R5    SUBTRACT >30 SO R5=NUMBER
       JEQ  KYIN          IF R5 ZERO, GO GET ANOTHER KEYSTROKE, IGNORE THIS ONE
       JLT  KYIN          IF R5 < ZERO, IGNORE
       C    R5,@NOITEM    ELSE COMPARE TO NUMBER OF ITEMS
       JGT  KYIN          IF GREATER, IGNORE
* AT THIS POINT, WE KNOW A NUMBER KEY WITHIN THE CORRECT RANGE HAS BEEN STRUCK
       DEC  R5            ZERO-BASE THE VALUE IN R5
ACC2   SLA  R5,1          DOUBLE THAT NUMBER, SINCE WE'RE INDEXING BY WORDS
       A    R13,R5        ADD TO R5 THE START OF THE BRANCHING TABLE
       MOV  *R5,R5        GET THE ADDRESS OF THE SELECTED CODE SECTION INTO R5
       B    *R5           AND BRANCH TO THAT ADDRESS
* END OF MENU DRIVER SOURCE CODE
* SUBROUTINE TO CLEAR SCREEN WITH OFFSET FOR XB
CLS
       LI   R4,SCRWID     SET R4 TO WIDTH OF SCREEN
       MOV  R4,R2         MAKE R2 ALSO = WIDTH OF SCREEN
       LI   R6,SCRLI      POINT R6 AT SCREEN LINE STORAGE
       MOV  R6,R1         PLACE THAT ADDRESS IN R1 ALSO
       MOVB @SPACE,R5     PUT A SPACE WITH OFFSET INTO R5
* THE BYTE AT LABEL SPACE IS >20 + >60 FOR XB'S OFFSET
LOP1   MOVB R5,*R6+       MOVE ONE SPACE WITH OFFSET, INC R6
```

THE ART OF ASSEMBLY—

(Continued from Page 25)

ample, the line just before label LOP1 in the CLS subroutine could be eliminated if the line at LOP1 said MOVB @SPACE,*R6+. Our good friend Jim Peterson (TIGERCUB), calls this kind of thinking Elegant Programming, where the programmer not only wants it to work, but wants it to be fully optimized in all respects. Maybe our next program will be better, but we're not going to re-assemble GSA just for that one small possible change.

Okay, so after the title is on the screen, we have a section of code that picks up the byte just after the title text, transfers that to R8, right justifies it, then stashes it at NOITEM. As it happens, the main menu has eight items, which is the most of any menu used in the program. The next section of code does some math with R0 and R8 to position the bulk of the menu vertically centered between the top and bottom of the screen. At label MEN1, we enter a loop which prints all the selections on the menu. Each call to DISLI leaves R9 pointed at the length byte for the next item, so the loop can proceed very quickly and efficiently.

Once all the items have been displayed (after JNE MEN1) there's another of our little tricks. We want the legend to appear at row 23, column 9. To do that, we let the assembler do the math for us. The assembler multiplies 22 by the width of the screen (this would place us at row 23, column 1), then adds eight to that number. The result is an immediate value placed in R0 which puts R0 just where we wanted it. This trick can be used in many ways, but here we've used it for positioning on the screen. One takes the number of the desired row, subtracts one, then tells the assembler to multiply by SCRWID, and then to add one less than the desired column. In addition to saving us some math, this also saves some time in program execution, because the math is performed during the assembly, and all the computer has to do at running time is load that one value into R0.

Now, once the legend is on the screen, all we need do is wait for the user to press a key. KEYLOO is the subroutine that does this for us, (see Part 2 for that subroutine) and in addition places the ASCII value of the struck key in R8. Given a keystroke, the menu driver checks it against the value 15. Fifteen happens to be the ASCII value for Function-9. In any of the menus in this program, striking Function-9 makes a branching to the last label in the lookup table for that menu. In the case of this main menu, that simply takes us back to KYIN for another keystroke. In other menus, that last label in the lookup table takes us back to a previous menu.

Having found some key value other than 15, the program must now make sure that the key struck in within the correct range for this menu. In this case that's 1 through 8. We move the keystroke to R5, subtract >30 so the number in R5 will be 1 through 8, not >31 through >38. Now we check for a result zero or less than zero. If either happens, the key struck was out of range, so we ignore it and jump back to label KYIN. Finally, we compare R5 to the data at NOITEM, which in this case contains 8. If it's greater than that, we again ignore the keystroke. While this menu is on-screen, hitting any

(See Page 27)

CODE FROM GOLF SCORE ANALYZER

```
        DEC   R4              DECREMENT COUNTER
        JNE   LOP1            IF NOT ZERO, REPEAT
        CLR   R0              SET R0 TO SCREEN ORIGIN
        LI    R4,24           24 ROWS TO CLEAR
LOP2    BLWP  @VMBW           WRITE ONE LINE OF SCRWID SPACES
        A     R2,R0           ADD SCRWID TO R0
        DEC   R4              DECREMENT ROW COUNT
        JNE   LOP2            IF NOT ZERO, REPEAT
        RT                    ELSE RETURN
* SUBROUTINE TO DISPLAY ONE STRING ON THE SCREEN
DISLI   LI    R10,SCRLI       POINT AT OUR BUFFER SCRLI
        MOV   R10,R1          MAKE R1 POINT AT THAT ADDRESS ALSO
        MOVB  *R9+,R4         MOVE THE LENGTH BYTE INTO R4
        SRL   R4,8            RIGHT JUSTIFY
        MOV   R4,R2           PLACE THAT NUMBER IN R2 FOR VMBW
        JEQ   DISLIX          IF THAT LENGTH WAS ZERO, GET OUT OF SUBROUTINE
DIS1    MOVB  *R9+,*R10       MOVE ONE BYTE OF CONTENT, INCREMENTING R9
        AB    @OFFSET,*R10+   ADD THE >60 OFFSET, AND INCREMENT R10
        DEC   R4              DECREMENT LENGTH COUNT
        JNE   DIS1            IF NOT ZERO, REPEAT
        BLWP  @VMBW           WRITE THE STRING WITH OFFSET TO SCREEN
DISLIX  RT                    RETURN

*  FOLLOWING LINES ARE FROM THE DATA SECTION OF SOURCE CODE
*  DATA FOR PRODUCING THE MAIN MENU
MENDAT  BYTE 19               LENGTH OF TITLE
        TEXT 'GOLF SCORE ANALYZER' TITLE TEXT
        BYTE 8,13             NUMBER OF ITEMS, LENGTH OF FOLLOWING TEXT
        TEXT '1  ADD ROUNDS' TEXT LINE
        BYTE 12               LENGTH OF TEXT FOLLOWING
        TEXT '2  LOAD FILE' SECOND TEXT LINE
        BYTE 14
        TEXT '3  DELETE DATA'
        BYTE 17
        TEXT '4  ANALYZE SCORES'
        BYTE 12
        TEXT '5  SAVE FILE'
        BYTE 19
        TEXT '6  ADD/EDIT COURSES'
        BYTE 17
        TEXT '7  REVIEW COURSES'
        BYTE 15               LENGTH OF LAST TEXT LINE
        TEXT '8  EXIT PROGRAM' LAST TEXT LINE
* DATA FOR PRODUCING THE LEGEND AT BOTTOM OF ANY MENU
SELEC   BYTE 16               LENGTH OF LEGEND
        TEXT 'SELECT BY NUMBER' TEXT OF LEGEND
* LOOKUP TABLE FOR BRANCHING OUT FROM MAIN MENU
* EACH DATA ITEM AT MAINBR GIVES AN ADDRESS OF A LABEL TO WHICH CODE
* BRANCHES WHEN A SELECTION IS MADE FROM THE MAIN MENU
* THE LAST ENTRY IN THE TABLE IS WHERE THE CODE BRANCHES WHEN
* FUNCTION-9 WAS STRUCK.  IN THIS CASE, WE EFFECTIVELY IGNORE THAT
* KEYSTROKE BY BRANCHING TO LABEL KYIN, WHICH SIMPLY WAITS FOR ANOTHER
* KEY TO BE STRUCK
MAINBR  DATA NRIN,FILGET,SELCRD,SELCRS
        DATA FILSAV,NCIN,CRSLST,BYE,KYIN
* MISCELLANEOUS DATA ITEMS
NUMASK  DATA >30
NOITEM  DATA 0
SCRLI   BSS   SCRWID
OFFSET  BYTE >60
SPACE   BYTE >20+>60
```

MICRO-Reviews

Mario Bros., Turbo 2056 and Linkages

By STAN KRAJEWSKI

Ratings for the software reviewed in this column are based on the good ol' star system that Harry Brashear used in his MICRO-review columns.

★ Leave it alone, back to the drawing board.

★★ Needs improvements, but workable.

★★★ A good program, worth trying.

★★★★ Send your money and buy it.

For my first installment I am reviewing games. It is the way I pass most of my time with my TI. However, I am program literate and have tried all programs that I have come across in the past seven years of being a TIer, including utility, music programs, etc, and never had much of a problem. This review will consist of programs from one distributor because, being a new columnist, I haven't had anyone send me any programs yet. These are the most recent programs I have purchased, and the more you buy the cheaper they get.

★★★

MARIO BROS.

I have seen it advertised in the classified section of MICROpendium, a Mario Bros. game for the TI? I just had to send for that one, liking the TI as much as I do, and with someone going to the trouble of re-creating a very popular game so we can continue to use our computer and still get satisfaction out of it.

It starts out with a TI Extended BASIC loader going into assembly routines. Yes, it takes 1-2 minutes to load, but it does look interesting once you get there. At the start-up screen (level 1) in the top left hand corner it displays the Lives and Level. Across the top are displays for the Timer, Coins and Score. Immediately you must start running your Mario to the right. You can jump hit the bricks with your head to reveal coins and mushrooms. You can get big by landing on a mushroom and collect all the coins you can. Once you are big it is easier to hit the bricks with the top of your head and, if you hit another mushroom, you turn into a fiery Mario. You can then pull down on your joystick and shoot fireballs, depending which way you are facing. There are surprises, because you might jump for no reason or miss a brick and a hidden brick comes into view. You have to be careful of the bad guys, as not all of them die by jumping on them.

There are 14 levels to keep you busy, and you must make it to the flagpole to reach another level. I have wanted to keep playing again so it's not easy to lose interest in it.

Some of the bad parts are: The timer does run awfully quick. You barely have time to grab coins and mushrooms before reaching the flagpole without the timer running out. You can move both foward and backward but the screen is a little confusing as everything moves in both directions instead of just the Mario character. I have used it on both my CorComp 9900 Micro Expansion system and the P-box and it runs flawlesly most of the time. I say most of the time because I have experienced screen malfunctions on both systems, causing me to reboot.

All in all I do enjoy the program and it does have entertaining music while you play. I have not tried it on my Geneve 9640 yet, but would assume the timer would be faster than it is now.

The program is available from Baker Software 8301 Stevenson Ave. Sacramento, CA 95828 for $9.50 + 1.50 S+H.

★★★★

TURBO 2056

This game brings back memories of Car Wars by T.I. but with more twisting and turning and the ability to compete. It has more screens than you can count and uses

(See Page 28)

ART OF ASSEMBLY—

(Continued from Page 27)

key other than the numbers 1 through 8 will have no effect whatsoever.

Immediately after the operation JGT KYIN, we know that the number in R5 is a number in the range 1 through 8, so we can proceed to branch out from the menu. First we must DEC R5, so that the range is actually 0 through 7. (If Function-9 had been struck, we'd jump to label ACC2 with 8 in R5.) Now, since we're going to index a table of words, not bytes, we must double the number in R5. The easiest way to double a simple integer like this is to Shift it left by one bit, and that's what we do at label ACC2. Now the number in R5 has a range of 0 through 14 (by twos), or 16 if Function-9 has been pressed.

R5 now has the index value for the member of the lookup table we want. We add R13, which contains the address of the start of the lookup table. The next operation, MOV *R5,R5, takes the number at that address in the lookup table and places it in R5. Finally, we branch to the address contained in R5, and that takes us into the selected function. In effect, we have performed an ON-GOTO function based on the key struck.

In this article we looked at some overall principles for Top-Down program design, then we presented one alternative for user interaction through a Menu Selection, and showed the source code for a reasonably effective menu driver. There are many other ways to implement a menu system, and we can be sure that some of our readers will come up with better ones than ours. Our purpose in these articles is mainly to teach principles of using Assembly, so the reader can use his own creativity in this language.

In the next installment, we'll try to concentrate on ways to make code as efficient in memory use as possible, with some "wrong way" and "right way" examples.

MICRO-REVIEWS—

(Continued from Page 27)

joysticks to control each player's car. It's a two-player game (which comes in handy in my family) and lets you compete against each other for a victory in each screen.

Both players start together in the same corner of each screen trying to battle their way around obstacles and walls to reach the finish line. The name Turbo really applies here, for when you press the fire button you really move. You have to master when you can use it and really gives you an edge when you just have to catch up to the other guy. But watch out, you don't know when you use all your turbo until it's to late. But thank goodness you get it all back at the next screen. At the top right of the screen the computer records how many wins each player gets.

The program autoloads out of Extended BASIC and you can pick how many rounds you and your opponent want to go. You have three choices, game ends after either player gets 10 wins, 25 wins or, if you really want to spend some time at the races, 50 wins.

At this price I cannot find a thing wrong with this one. Also from Baker Software at $4. One or more programs $1.50 S&H.

★ ★ ★

LINKAGE

If you like shooting games with a joystick, this could be for you. I like games that use the joystick so with this it is a plus for me.

In Linkage, you are flying through a trench with a barrage of alien aircraft coming at you. You can fire missiles at the enemy until you die or win the game. But winning is not easy. Along with shooting and dodging the aliens you must push up on the joystick and try to make it to the end of the trench. As you continue up the trench the distance meter lets you know how close you are getting to the end. You want to keep looking how far you got but hate to take your eye off the oncoming aliens. If you do make it through all those aliens, you must face another one that you must blast many times before proceding to the next level. Along with the distance, the screen displays the score your lives and your level.

This game offers good graphics and has sound and a bargain price. Once again it is $4, plus $1.50 S&H per order.

Call or write Baker Software for more than one game and see about their discount. 916-689-6946 or same address as before.

In my next review I will be able to run programs from the Geneve. I didn't have time this month to get it up again since my last minor problem. Happy computing.

I hope to be able to review your program in the future. Send programs to: Stan Krajewski, Route 6 Box 568-15, Live Oak, FL 32060.

Comments

Digitized sound using the TI/9640

Barry Boone has a pair of programs that produce incredible sound on the TI99/4A and Geneve. One program is used to convert digitized sound from PC format to a TI/Geneve format. The second program plays the sound on the TI and Geneve. The Geneve will handle files up to 2 megabytes long with a Memex card. Using high-resolution playback, a file this long will last about five minutes.

The sounds that come through the TI are incredibly realistic, whether the sound be music or voice. Barry said the sounds are better on the TI and Geneve than on PCs. When Barry gave me a demo over the phone, the only thing I could say was "where can I get it." Texaments will be handling the program, which is tentatively priced at $14.95. It is expected to be available in mid-October.

How does it work? The program loads the sound chip in the TI and Geneve with one frequency and then modulates the volume (this is not what TI recommended when using its sound chip, but it works). The results are realistic and compelling. He used digitizing equipment from a PC to generate raw digitized files and then translated the files for use with the TI and Geneve. The program, called Sound F/X, will come with the playback software and a collection of sounds.

Also coming this fall, for the Geneve, will be some games ported over from other machines, including one educational game. No word yet on titles, but my source says you'll recognize them when you see the names.

CALLING MYARC

We've been getting a lot of calls from readers wondering how to get in contact with Myarc. I wish we could help, but Myarc disconnected its phone at its Alabama office, and that is the only voice contact that was available. I've been hearing from readers who say that they shipped their Geneves and HFDCs to Myarc as far back as February and still haven't gotten them back. By the way, these repairs were prepaid, as per Myarc's repair policy. (Myarc's had my HFDC since May, with no word about its condition.)

Please, Myarc, why don't you at least send a postcard to your loyal users and let them know that you've got their equipment. It costs only 19 cents, and MICROpendium won't have to continue to deal with the calls that are meant for you.

The most recent information I have, from a reliable source, is that Myarc had to replace it's repair technician in late spring and that the replacement took awhile to get up to speed. I was told this in early July, so I would assume there is another reason as to why repairs aren't being made in a timely fashion.

BACK TO 40 PAGES NEXT MONTH

We will be back to 40 pages next month.

—JK

User Notes

Dumping modules to disk with F'web

This comes from Sam Carey, of Portland, Oregon. He writes:

If you have some sort of assembly language cartridge dump program on disk, and you want to dump a game module to disk with it — and don't know how — read on.

First, load Funnelweb. Then load the FW editor, E/A or TIW, it doesn't matter which. Carefully remove the XB/EA/MM cartridge from the cartridge port and replace it with the game cartridge you'd like to dump. Exit the editor with the following keystrokes: Function 9, Q, Enter, E, and Enter.

Next, load the dump program. You're all set. You could use this method of switching cartridges for any reason. It's that easy to dump a game cartridge.

Using the break key with XB and Geneve

Geneve users who use TI Extended BASIC know that the break key doesn't work, so programs can't be stopped once they've started running. Barry Boone says that users can change this by making a a modification to the first two files of TI Extended BASIC. You'll need a sector editor to do.

First, load the first of the XB files into the sector editor and search for the following hex string:

83D6 020C 0024 30E0

Replace it with the following string:

83D6 06A0 0020 1010

Do this for the second XB file as well.

You now have a functional break key.

Nesting .IF files and saving paper with TI-Writer

This comes from Harold "Pete" Sarasin, of Goleta, California. He writes:

Page 109 of the TI-Writer Reference Guide says that the .IF (Include File) command does not permit nesting of files — calling a second file from the first and a third from the second. However, it can be done by using Alternate Input (*n*). The "n" is a number from 1 to 99. Alternate Input can be used with or without Define Prompt (.DP).

Take a blank disk and create the following six files with filenames FN1 to FN6. These files will demonstrate how .IF (Include File) can be accomplished by entering all or part of a .IF DSKn.FILENAME as prompted.

Filename FN1
0001 TEST 1
0002 .DP 2:ENTER "n" of DSK1. FN"n"
0003 .IF DSK1.FN*2*
Filename FN2
0001 TEST 2
0002 .DP 3:ENTER "FNn" of DSK1. "FNn"
0003 .IF DSK1.*3*
Filename FN3
0001 TEST 3
0002 .DP 4:ENTER "DSK1. FNn"
0003 .IF *4*
Filename FN4
0001 TEST 4
0002 .DP 5:ENTER "n" of DSK"n".FN5
0003 .IF DSK*5*.FN5
Filename FN5
0001 TEST 5
0002 .DP 6:ENTER ".IF DSKn.FNn"
0003 .IF *6*
Filename FN6
0001 TEST 6
0002 .DP 7:PRESS "FCTN 9" TO EXIT
0003 *7*

Enter the Formatter with disk in drive 1. Enter DSK1.FN1 in the initial Formatter prompts.

TEST 1 printed

(See Page 30)

Feedback

(Continued from Page 6)

Comp's TI-IBM Connection or Mike Dodd's PC-Transfer. If you like, you may submit an item for our Reader-to-Reader column in hopes that another reader may have what you want. Or, you can send for information about joining National Used Software/Hardware Club (see the ad on the classified page). NUS/HC has a listing for CorComp's TI-IBM Connection cartridge, which will allow you to transfer text files from a TI to a PC.

As for your second question, the problem may be in your RS232 card. If both printers started to exhibit problems at the same time, or both printers work fine with another computer, suspect the card. TI will repair it for a modest charge. Call 800-TI-CARES. For technical assisance, call TI at 806-741-2663.

READER TO READER

☐ Bob Zink, 4217 Molokai Dr., Naples, FL 33962, writes:

In the last eight years I have accumulated quite a few disks. Lately, I have begun to experience more and more disk failures. While I was visiting in Detroit last month, I brought along a rather important data disk and its backup, both of which garbaged the catalog screen.

Herb Schlesinger of the Great Lakes 99ers somehow salvaged the disk. I have not been able to duplicate his feat, either by following his directions, replacing the disk header or using Recover. I cannot access any of the files to do anything with.

Also, how often are files supposed to be transferred to new disks? Even though I have three DSDD exterior drives, replacing all the disks would be a formidable task.

☐ Sam Carey, 5820 S.E. Westfork St., Portland, OR 97206, writes:

Does anyone know of, or can anyone write, a routine in assembly language that will scan Joystick No. 1 and move sprites 1 through 9 the opposite direction of the joystick and hold the sprites still if the joystick is not used, and will move the sprites to the other side of the designated area if they go too far in one direction?

☐ Geographically alert readers no doubt sent answers to the query of Michael G. Mickelson last month to Des Plaines IL rather than Des Plaines IA as we printed.

Reader to Reader is a column to put TI and Geneve users in contact with other users. Be sure to address your questions to Reader to Reader, c/o MICROpendium, P.O. Box 1343, Round Rock, TX 78680.

User Notes

(Continued from Page 29)

at prompt enter "2"
TEST 2 printed
at prompt enter "FN3"
TEST 3 printed
at prompt enter "DSK1.FN4"
TEST 4 printed
at prompt enter "1"
TEST 5 printed
at prompt enter ".IF DSK1.FN6"
TEST 6 printed
prompt will be "PRESS FCTN 9 to QUIT"

Pressing FCTN 9 returns to TI-Writer menu. This last sample can also be used (with or without a prompt) to save paper, by making the "n" Alternate Input the last line of the file.

Incidentally the Alternate Input with Define Prompt can also be used with the Header or Footer commands when printing documents that are not in conscecutive page number order.

French users offer newsletter on disk

A group of 20 TI users in France has prepared its fifth newsletter on disk.

According to Pierre Delfort of the group, the newsletter contains announcements, idea exchanges and programs in Extended BASIC or Editor/Assembler.

"We have a fabulous assembler programmer," Delfort comments.

He says the group works with English, Belgian and German users groups.

To receive the disk news, send one double-sided or two single-sided disks with postage (buy international reply coupons at the post office) to Delfort at 7 allee de la pinede, 30200 Bagnols/ceze, France.

Hunter Valley UG disbands

The Hunter Valley 99ers User Group, New South Wales, disbanded June 25, 1991. Reasons given for disbanding include lack of members willing to participate on committees, absence of contributions to the group's newsletter and the lack of programs available locally for the TI.

Auto-loading from tape

This item, by D.N. Harris, appeared in the newsletter of TIsHUG, Sydney, Australia.

The command RUN "CS1" can be used instead of END so that a tape of Extended BASIC programs will load each oth Each program must be END changed to RUN "CS1". The effect is like RUN "LOAD" in a disk collection. Suppose the program line is:

1000 END

Make it:

1000 RUN "CS1"

It starts a cassette load as soon as the program ends.

Remember to put the quotation marks around CS1, otherwise it is not a valid device, although for SAVE CS1 you leave off the quote marks. The same for OLD CS1. For RUN "CS1" you must use the quotes both as a systems command and as a statement in a program.

This will also work for TI-BASIC programs running out of the Extended BASIC environment provided that the allowable range of character sets is employed and that the program does not ask for "SPEECH", OUTPUT.

MICROpendium pays $10 for items submitted by readers for publication i *User Notes*. If you have a tip or idea, routine or other information that may be of interest to other readers, send it to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

1991 TI FAIRS

MARCH

Family Computer Exposition and Ham Radio Festival, (formerly TICOFF), March 6, Roselle Park High School, 185 West Webster Ave., Roselle Park NJ 07204. Sponsored by students of the high school and the Old Bridge Ham Radio Club. For information write the high school or call (201) 241-4550 or call the 24-hour informational BBS at (201) 241-8902.

APRIL

Northeast TI99/4A Home Computer Fair, April 6, Central Middle School, Waltham, Massachusetts. Contact Justin Dowling, The Boston Computer Society, 1 Kendall Square, Boston, MA 02139.

Canadian TI-Fest, April 27, Merivale High School, Nepean, Ontario, Canada. Contact Bill Gard, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6 or (613) 523-9396 or Fax (819) 997-2194 Attn: DMES 2.

MAY

TI Orphan Reunion, May 11, Innisfail Lions Hall, Innisfail, Alberta, Canada. Contact Fred Kessler, Box 20, Sundre, Alberta, Canada T0M 1X0 or (403) 638-3916.

TI99/4A Users Group, UK, Annual Meet, May 11, The Music Hall, The Square, Shrewsbury, England. Contact Stephen Shaw, 10 Alstone Rd., Stockport, Cheshire, England, SK4 5AH.

Multi User Group Conference, May 18, Reed Hall, Ohio State University Lima Campus. Contact the Lima User Group, P.O. Box 647, Venedocia, OH 45894, or phone Dave Szippl evenings, (419) 228-7109.

SEPTEMBER

6th International TI User Treffen, Sept. 13-15, Berlin. Contact Henry Hillsberg, Uhlandstr. 70, (W) 1000 Berlin 31, Germany.

Convention, Sept. 21, South End Pool Center, 402 E. 56th St. Tacoma, Washington. Contact Barb Wiederhold, (206) 546-1865 (BBS) or (206) 546-1205.

NOVEMBER

Chicago International World Faire, Nov. 1-2, Elk Grove Holiday Inn, Elk Grove Village, Illinois. Contact Chicago TI Users Group, P.O. Box 578341, Chicago, IL 60657.

1992 TI FAIRS

FEBRUARY

Fest-West, Feb. 15-16, Days Inn-Phoenix/Camelback, 502 West Camelback,Phoenix, Arizona. Contact VAST Users Group, c/o Tom Pfeffer, 116 S. Stellar Parkway, Chandler, AZ 85226; H. Knight (602) 938-5446; R. Rees, (602) 869-8145; or the VAST BBS, (602) 869-8145.

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.

Classified

SOFTWARE

TI-PD PUBLIC DOMAIN AND FAIRWARE

550 DISKS just $1.50 EACH! And orders for 8 or more disks are postpaid.

Thousands of programs selected from the best from the U.S., Canada, Australia, England, Germany, Holland and Belgium. FAIRWARE IS OFFERED BY AUTHOR'S WRITTEN PERMISSION ONLY. Disks as full as possible, arranged by exact category, BASIC programs converted to XBASIC, assembly programs with XBASIC loader, disks with autoloader by full program name.

Send $1.00 (deductible from first order) for 13-page catalog listing all programs and authors. Catalog also available on disk.

TIGERCUB SOFTWARE, 156 Collingwood Ave., Whitehall, OH 43213. v8/10

USVBA Volleyball $10.00; Cutthroat Cribbage with 5 other games, $10.00; NFL Football Forecaster, $10.00; Wall Street Advisor, $30.00. Program Innovators, 4122 Glenway, Wauwatosa, WI 222. v78,10,12,v82,4,6

FOR SALE

ENORMOUS TI99/4A INVENTORY

Catalogs $2.00. Braatzs Computer Services, 719 E. Byrd St. Appleton, WI 54911. 1-414-731-3478. 8/8

FOR SALE

Geneve 9640, PE Box, TI controller, two 5 1/4 drives, RS232, TI color monitor. Also TI99/4A keyboard, 32K card, P-Code card. Software on disks and car-

Buy or sell used software and hardware

National Used Software/Hardware Club brings computer buyers and sellers together. Whether you want to buy or sell, we can help you. Annual dues are $15, and include newsletter. For free, no-obligation information package, call 1-800-777-6632, or send #10 SASE to NUS/HC, Dept. M, P.O. Box 1343, Round Rock, TX 78680.

Policy

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

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

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

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

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

Send classified advertising to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680.

FOR SALE

tridges. $475 plus shipping. Call 512-343-2130 after 6 PM CDT. v7n8

HARDWARE FOR SALE

NEW: Bare 5.25" ss/sd disk drive.
USED: 2 external 5.25" ss/sd disk drives, printer A/B/C switch, Star Gemini 10X printer, TI99/4A console, misc. cables, 3 5.25" diskette storage boxes, and ten cartridges (incl. XB, TEII, Disk Manager 2)
USED AND AS IS (locks up after several minutes): PEB with 2.25" ss/sd disk drive, drive controller card, 32K card, RS232 card, and Triple Tech card.
Manual will be provided with most items. I will pay shipping. MAKE OFFER on any or all items. Write to: Kenneth Dirks, H.C. 68 Box 18, Chappell, NE 69129. 8/7

HARDWARE

FOR SALE

Two 99/4A consoles, PEB and DS disk, 32K, RS232, controller, printer, XBASIC, Logo II, TI-Writer, synthesizer, Multiplan, Plato, Volksmodem, joystick, manuals, software & more. $575. List available. Call 205-956-1476. v8/7

99/4A & Full PE-Box,$200.00; 384K HRD, $140.00; P-Gram, $140.00; Rave Speech Card, $35.00; TI Recorder, $20.00; 99/4 Impact Printer, 40.00; P-Code w/Manuals, $45.00; P-Code, $30.00; US Robotics 1200 modem, $25.00. Shipping Extra, UPS Ground

HARDWARE

(COD). SASE for Software List. Ernie Pergrem, 2288 W. Nichols Rd., Arlington Hts., IL 60004. (708)255-5075 after 5PM CST.

MISCELLANEOUS

NINTENDO FOR SALE

Nintendo game player, pistols, joysticks, including rapid fire joystick. Game cartridges include R.C. Pro Am, Ice Hockey, Major League Baseball, Bases Loaded Baseball, Mario Bros/Duck Hunt, Fast Break, NFL Football, Stealth, Racket Attack, Metal Gear, Sky Shark and Ninja Gaiden. $150 takes all. Call 512-255-1512. v8/7

PASCAL-COMPLETE SYSTEM	$150
TI RS232 CARD used	$95
FULL TI PE/BOX-RS232-32K-DRIVE	$299
EMPTY TI PE/BOX (used)	$100
18" P-BOX EXTENSION CABLE	$25
EDITOR ASSEMBLER PKG. (new)	$12
SPEECH SYNTHESIZER used	$35
PARALLEL PRINTER CABLE 6'	$20
PE-BOX TECH TRAINING MANUAL	$20
TI-ORIGINAL COLOR MONITOR	$150
SERVICE MANUAL (CONSOLE/P-BOX)	$25
4A FACTORY REPAIR MANUAL	$20
DISK CONTROLLER & 32K CARDS ea	$95
P-BOX "CARD" REPAIR MANUALS (ea)	$10
TI COLOR MONITOR SERVICE MANUAL	$15
SCHEMATICS: CARDS/CONSOL/PBOX ea	$5
CC40 (TI Compact Computer)	$55
CC40 peripheral user manuals	$10

USED TI99/4A, Hardware, Software, Books and Parts. Call or write for complete free list. S/H, $3 min.
Jim Lesher, 722 Huntley, Dallas, Texas 75214; 214-821-9274.

The ONLY monthly devoted to the TI99/4A

Subscription Fees

$25 for 12 issues via domestic second class mail
$30.25 (U.S. funds) Mexican delivery
$32.50 (U.S. funds) Canadian delivery
$30.00 (U.S. funds) for 12 issues other foreign delivery via surface mail
$42.00 (U.S. funds) for 12 issues other foreign delivery via air mail

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.

Back Issues

Back issues of MICROpendium are available to subcribers only. Those wishing back issues may notify us of the issue(s) desired and include $2.50 per issue desired in a check or money order or by credit card. (Minimum credit card order is $9.) No shipping charge in U.S. and Mexico; Texas residents add 7.75% sales tax. Shipping charge of 30 cents per issue to Canada. For other foreign delivery, add 50 cents per issue surface mail, $2 per issue air mail. No discounts on orders of sets. All prices U.S. funds.
OUT OF STOCK: Vol 1, nos. 1-2, Vol 2, no. 1

Miscellany

TI-Forth Disks (2 disks, program and demo disks, no manual).....**$6.00**
MICROpendium Index (2-SSSD disks, XBASIC required).....**$6.00**
Disk of programs from one issue of MICROpendium (must be a subscriber to order).....**$4.00**
12 monthly disks (April 1991-March 1992) of programs appearing in each edition of MICROpendium (must be a subscriber to order).....**$40.00**
Magazine holders (12/set-add $1 shipping/order).....**$3.00**

Send name, address, product(s) ordered, check, money order or Visa/MasterCard number and expiration date ($9 minimum on credit card orders $9) to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680. (Foreign orders write for postage fees. Texans add 7.75% sales tax.)

Tell us about it

Please let us know what columns or features you like the most about MICROpendium. Rank your selections in order of preference using this form. Return it to us when you renew your subscription.

1. _____
2. _____
3. _____
4. _____

Other suggestions:_____

_____

_____

Send me the next 12 issues of MICROpendium. I am enclosing $ in a check or money order in U.S. funds. Or bill my ☐ MC ☐ VISA (Check one)

Exp. Date _____

Card No. _____

Minimum credit card order is $9

Signature _____

(required on credit card orders)

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

Name_____

Address_____

City_____

State____**ZIP**_____

The numbers on the left of your mailing label indicates the cover date of the last issue on your subscription.

SECOND CLASS

A T EXPIRES 1/92
CHARLES GOOD
P.O. BOX 647
VENEDOCIA OH 45894