Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 9 Number 4 | May 1992 | $2.50

Programming

Regena on BASIC
Trigonometric Functions

The Art of Assembly
Getting the Most from VDP RAM

MY-BASIC
More on Video XOP6

Extended BASIC Plus
Unbasher Breaks Down Programs

Reviews

LinEditor

• • •

OS/99 V3.0

• • •

Easy Data

• • •

Grades

• • •

LPJ Art 5 & 11

• • •

Extended BASIC Utilities Vol. 1 & 2

• • •

Random Number Utilities

Also inside

MICROpendium 1991 Index

• • •

A Disassembler in Extended BASIC

• • •

New Products from Horizon

• • •

An exploration of Funnelweb V4.40

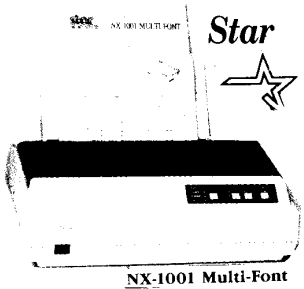
• • •

Archiving Plato disks

Super Savings

TEX-COMP
FOR TEXAS INSTRUMENTS

There may never be a better time than now to buy

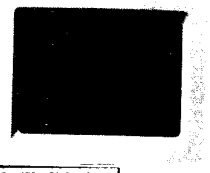


NX-1001 Multi-Font

THE NX-1001 MULTI-FONT OFFERS 5 RESIDENT FONTS WITH SPEEDS UP TO 180 CPS. FULLY TI (AND IBM) COMPATIBLE WITH ALL OF TI GRAPHICS PROGRAMS. SWITCH FROM TRACTOR TO SHEET WITHOUT REMOVING THE TRACTOR PAPER. ALL THIS PLUS GREAT LETTER QUALITY AND A FREE TEX-COMP BONUS OF 99 WRITER II WORD PROCESSOR, BETTER BANNERS WITH GRAPHICS, PRINTER CARE DISK, STAR DEMO DISK, A COPY HOLDER AND A FREE SPARE RIBBON. A $100+ BONUS VALUE.....ONLY $189.95 +s&h

TI PRINTER CABLE..$24.95 +s&h, RS232 CARD...79.95 +s&h

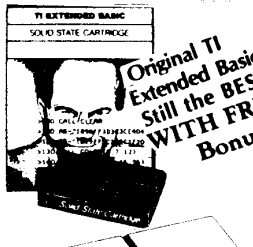
****



THE NEW MAGNAVOX 14" COMPOSITE/RGB/ANALOG COLOR MONITOR IS THE ONLY MONITOR THAT WORKS WITH EVERY TI SYSTEM INCLUDING THE GENEVE. FEATURES INCLUDE DUAL SPEAKERS AND STEREO AMP SO YOU CAN EVEN HOOK IT UP TO A STEREO VCR OR A LASER DISK PLAYER FOR FANTASTIC PICTURE & SOUND. WE INCLUDE A TI AND A IBM CABLE. WITH A TUNER OR VCR YOU ALSO HAVE A SECOND TV. COMES WITH A 2 YEAR WARRANTY FROM PHILIPS.

.....ONLY $279.95 +$10s&h

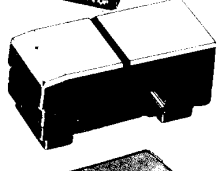
****



THE ORIGINAL TI EXTENDED BASIC IS AN ABSOLUTE MUST FOR ANY 99\4A SYSTEM. COMPLETE WITH THE ORIGINAL MANUAL. WE WERE ONLY ABLE TO GET A LIMITED QUANTITY OF THESE NEWLY PRODUCED MODULES. WITH EACH MODULE WE INCLUDE YOUR CHOICE OF 5 FREE DISKS FROM OUR FREEWARE COLLECTION, A $25 VALUE.

.....ONLY $64.95 +s&h

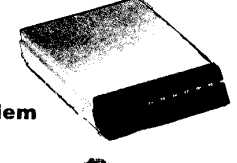
****



THE SPEECH SYNTHESIZER IS BY FAR, THE TI ACCESSORY THAT IS THE MOST DIFFICULT TO LOCATE. WE WERE ABLE TO GET HOLD OF A A VERY LIMITED QUANTITY. WITH EACH SYNTHESIZER WE INCLUDE 5 FREE DISK FROM OUR FREEWARE COLLECTION. CHOOSE FROM ONE OF THE MANY DISKS WITH SPEECH CAPABILITY.

.....ONLY $59.95 +s&h

****



Modem

OUR NEW 300/1200/2400 MODEM IS THE ONLY MODEM YOU WILL EVER NEED. WORKS WITH ANY COMPUTER..TI,MAC,IBM ETC. WE INCLUDE THE TI CABLE AND A COLLECTION OF THE BEST TI COMMUNICATIONS SOFTWARE TE-II, FAST TERM & TELCO. .....ONLY 99.95 +s&h

****

DISK DRIVE
Fully TI Compatible!

WE WERE ABLE TO GET A SPECIAL BUY ON A FULL SIZE DS/DD DISK DRIVE BY SHUGART WHO MADE THE ORIGINAL SS/SD DRIVE FOR TI. THIS DRIVE IS FACTORY NEW AND IS COMPLETELY INTERCHANGEABLE WITH THE P-BOX DRIVE IN ONLY 5 MINUTES. THIS DRIVE WORKS WITH ALL FORMAT DISKS, SS/SD, SD/DD, DS/SD AND DS/DD. WITH THE ORIGINAL TI DISK CONTROLLER YOU CAN FORMAT AND RUN BOTH SINGLE AND DOUBLE SIDED DISKS. THIS DRIVE WILL GIVE YOU TWICE THE DISK STORAGE CAPACITY WHEN USED WITH THE ORIGINAL CONTROLLER AND 4 TIMES WITH A CORCOMP OR MYARC CONTROLLER. DON'T WAIT UNTIL YOUR ORIGINAL DRIVE GOES OUT. REPLACE IT NOW AT A PRICE LOWER THAN THE COST TO REPAIR OR REBUILD THE ORIGINAL SINGLE SIDED DRIVE.....SPECIAL PRICE $39.95 +s&h

****

TEXAS INSTRUMENTS AUTHORIZED SALES

TEX-COMP™

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

AUTHORIZED DEALER

VISA and MASTERCARD HOLDERS CALL DIRECT:

(818) 366-6631

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). Tax of Mississippi 4½%. (Free shipping on all software orders over $100.00) We will ship UPS C.O.D. with 25% deposit. C.O.D. to be paid by cash or certified check. All products are sold with the original manufacturer's guarantee (sent on request). Prices and availability subject to change without notice. We reserve the right to limit quantities.

ALL PRICES REFLECT A 3% DISCOUNT FOR CASH
ADD 3% IF PAYING BY CREDIT CARD

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.

Display advertising deadlines and rates are available upon request.

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

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

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

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

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

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

Departments

***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.

Comments

Is there anything as reliable as a TI?

I've been thinking about reliability recently. I've recently had numerous pieces of equipment go out on me — a laser printer, typesetting processor, answering machine, waxer. And then, as each piece of equipment went out, one right after the other, I got to thinking about how many other problems I've had with other equipment, PCs, Macintoshes, VCRs, televisions, printers, typesetters, you name it. And the thousands of dollars I've paid over the years to have them repaired. And let's not bother getting into automobiles. A legacy of unreliabilty, is what I've found, regardless of the manufacturer.

But do you know what piece of equipment I have never had a problem with? You guessed it, my TI99/4A. I've been running one TI since 1983 at the office 6 days a week. I turn it on when I come in and turn it off when I leave for the day. Connected to it is a PEB that I got in August 1983. It keeps the same hours as the TI. Also in August 1983 I got a TI color monitor. And it, too, runs concurrently with the TI. As I figure it, I've had this equipment running for over 20,000 hours, and not once has it gone down. That's equal to more than 860 calendar days, or more than two and one-half years without a breakdown.

Oh, I've had to clean the GROM port once or twice, but that is it as far as regular maintenance is concerned. Maybe once every four years when I get ambitious I take a vacuum cleaner to it to get the dust out. But that's as obsessed about maintenance as I get.

Amazing.

I wonder if my TI is unusual? Are there others whose machines just keeping going and going? Let me know. I'm curious. But it's a nice feeling to know that when I turn on my TI, PEB and monitor that it will come up. I wish I could say that about my pickup.

—JK

BUGS & BYTES

Products debut at Lima

TI fairs are often where products make their debut, and we have a story elsewhere this issue about what Bud Mills was scheduled to introduce at the Lima fair May 15. Also scheduled to be shown at Lima, we hear, a new beta-test version of Funnelweb. Funnelweb is a menuing/operating environment program written by the talented Australian father-son team of Tony and Will Lewis.

Development conferences

Interest in the Development topic on Delphi's TI-NET appears fairly high — at least, enough for several conferences on this relatively new topic to have been held on line.

Talk back on Echo

Delphi TI-NET sysops were disappointed at the apathy on their recent poll regarding whether TI Echo, until they learned that a number of users already had access. Others were concerned at the amount of downloading time for TI Echo. TI Echo is being offered on several telecommunications services, including GEnie. TI Echo is a part of Internet, a giant, global messaging system that links computer users.

TI Echo messaging was recently introduced on Delphi and GEnie to see whether TI users who subscribe to those services would be interested in the service. Messages keyed to the TI99/4A and Geneve are automatically routed to TI Echo members regardless of where they originate on Internet. The expense to download this data is generally limited to telephone charges. Internet is used by many universities and scientific companies.

'No Comment' from Shaw

Stephen Shaw's former "Rambles" column has for the last couple of issues of the United Kingdom's TI*MES user group magazine been entitled "No Comment." In contrast to "Rambles," "No Comment" stays away from controversy, at the request of the group's powers that be reacting to complaints from some members. Shaw does invite comments from readers, however, and in the spring issue one did wrote in to suggest that other readers should be able to figure out that views expressed are those of individuals. The reader notes "as long as it is not defamatory or libelous, why not let them speak?"

Defragmenter cleans up disks

Mark Schafer is offering a program called Defragmenter for $7. The assembly language program rearranges files physically on a disk in alphabetical order. This allows the disk controller to locate programs and files more quickly. The program handles floppies from SSSD to DSDD, but not quad-density or hard drives.

We'll have a review next month, but if you are interested in ordering, mail Schafer $7 at 539 Whitaker St., Morehead, KY 40351.

ImageWise Digitizer

A number of TI users are interested in digitizing video images for use with their TI. The following information was compiled by Jim Peterson and we saw it in the newsletter of the Massachusetts Users of Nine-nine and Computer Hobbyists (MUNCH). The ImageWise Serial Video System is one way to do it. The system comes as a kit and includes a digitizer-transmitter and receiver-display. It is capable of capturing an image from video cameras, camcorders, etc. The system is supposed to work with a TI equipped with an 80-column card or a Geneve. For information and pricing, write Micromint Inc., 4 Park St., Vernon CT 06066, (203)875-2751.

Software available for TI software to be used with the ImageWise system include GRAB, SHOW and CONVERT ($9.95); ImageWise Portrait Print ($4.95); ImageWise Display Routine, requires Yamaha 9938 VDP or Geneve ($4.95); and ImageWise Pictures. All are available from R.F.W. Enterprises, 111 Oakridge St., Chicopee, MA 01020. Write first to determine system requirements of the programs (included an SASE).

Also available is a program called IWD Plus for use with the Geneve and the ImageWise Video Digitizer. This program captures digitized video data over an RS232 port. The output can be saved to disk in ImageWise or MY-Art format. Price is $14.95, plus $1.50 S&H. Order from Joseph Szydek, 99 Highland Ave., West Springfield, MA 01089-1017.

Feedback

TI in Desert Storm

Regarding Barry Traver's comment about 9900 CPUs used in missiles in Desert Storm: I work with Hughes Aircraft Co. with the software for the TOW missile, which was used with enormous success from jeep (or "Humvee") mounted launchers or Bradley fighting vehicles. It isn't the missile that has the CPU, but the launcher — two SBP9989 MPUs (current-injection logic), to be exact. This is the only use I know of where our fabulous 9900 technology was used in the gulf conflict. One 9989 executes the software for tracking the missile optically and steering it, while the other operates the infra-red video tracker and passes position data to the first, via CRU, even!

Anyway, it wasn't so much mathematical accuracy that got the 9989 picked. TI designed the guidance set! (Where have we seen that before?)

David Ormand
Tuscon, Arizona

MIDI-Master query

What's going on with MIDI-Master for our TI?

Mike Maksimik never acknowledged my registration or my subsequent letter about it or the *promised* upgrade. I believe you've already mentioned it was available.

Did anyone else get theirs? I don't mind if I have to pay a nominal fee — I need it, and the little extra out of my system that the upgrade would provide.

Randy Packahm
Nanticoke, Ontario, Canada

Our review by Bruce Harrison and Dolores Werths (October 1991) indicated an upgrade to v3.1. — Ed.

Where's the squeeze?

In your April 1992 issue, in the Comments section, a statement was made, "TI squeezed out of TICOFF."

This was not true. TI groups represented there were Harrison Software, Genial Services — Barry Traver, Mike Hense — VMC Software, Ed Machonis — MAC Labels, Colonial Software — TI shareware, Delaware Valley TI User Group, New Jersey TI User Group, Philadelphia Area TI User Group, Southern NJ TI User Group, TI Swap Shop and sales of MICROpendium and *Computer Monthly*. TI was represented by one-third of the total of 31 vendors.

The user groups were selling software and hardware as well as the commercial vendors.

If TI was *squeezed out*, it was done by the TI vendors who didn't even bother to answer our invitation to attend. Now, their TI community will have one less place to show their wares.

TICOFF 1992 was a financial success. We were able to raise $4,000 for the Roselle Park High School Scholarship Fund. In addition, $250 was raised for the March of Dimes by the 650 paid admissions. This number was down 200 from last year. The six speakers, five TI subjects plus Prodigy, had a good response.

I feel the *TI community lost* one more place to show their wares in the New York, New Jersey and Connecticut area. We lost the MANNERS show in Virginia, now TICOFF. What show next?

With the exception of *Boston* the East Coast will have no shows.

Francis J. Bubenik Jr.
Coordinator of Speakers,
TICOFF 1992
Editor, LITI 99ers Newsletter
Hicksville, New York

Still the best

I have two TI99/4As, one with an OPA-TIM 80-column card, and I am expanding this equipment slowly. I appreciate the support of the many people still working with the TI computer and I still feel it is the best unit for normal home use.

David L. Strachan
Lehighton, Pennsylvania

1992 TI FAIRS

FEBRUARY

Fest-West, Feb. 15-16, Days Inn-Phoenix/Camelback, 502 W. 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) 233-0790.

MARCH

T.I.C.O.F.F. (TI Computer Owners' Fun Faire — The IBM & Clone Owners' Fun Faire), 9 a.m.-4 p.m., March 14, Roselle Park High School, Roselle Park, New Jersey, $5. Contact Robert Guellnitz, Roselle Park Public Schools, 185 West Webster Ave., Roselle Park, NJ 07204, (908) 241-4550 (voice) or (908) 241-8902 (BBS).

APRIL

Northeast Computer Fair, April 4, Waltham High School, Waltham, Massachusetts, sponsored by TI99/4A User Group of the Boston Computer Society. Contact Ron Williams, 14 East St., Avon, MA 02322.

Dutch Annual TI-Fair, April 25, Utrecht, The Netherlands, sponsored by Dutch TI-Usergroup. Contact Drs. Erik C. van Wette, Hanninkhoek 39, 7546 AD Enschede, The Netherlands, phone: 31-53-778723.

Ottawa TI Fest, 10 a.m.-4 p.m., April 25, Merivale High School, 1755 Merivale Rd., Nepean, Ontario, Canada. Contact Ottawa Users Group c/o Bill Gard, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6; (613) 523-9396 (home); (819) 994-8856 (work); (819) 994-8873 (work, attn. DSE 2).

MAY

TI Orphan Reunion, 10 a.m.-5 p.m. May 9, Innisfail Lions' Hall, Innisfail, Alberta, Canada. Contact Fred Kessler, Box 20, Sundre, Alberta, Canada, T0M 1X0, (403) 638-3916.

TI99/4A Users Group, UK, Annual Meeting, May 16, Princess Anne Training Centre, 10 Trinity St., Derby (Derbyshire, England). Contact Stephen Shaw, 10 Alstone Rd., Stockport, Cheshire England SK4 5H.

Multi User Group Conference, May 15-16, Ohio State University Lima Campus. Contact Lima 99/4A Users Group, P.O. Box 647, Venedocia, OH 45894 or phone Dave Szippl (419) 228-7109 or Charles Good (419) 667-3131 evenings.

SEPTEMBER

State of Washington TI Convention, Sept. 19, Tacoma, Washington. Contact Jim Tomkins, (206) 756-0934.

OCTOBER

Chicago International World Faire, Oct. 30-31, Elk Grove Holiday Inn, Elk Grove, Illinois. Contact Chicago Users Group, 2515 Marcy, Evanston, IL 60201-1111.

NOVEMBER

TI-Faire, Nov. 28-29, Ashfield Boys High School Hall (next to Western Suburbs Leagues Clu), Liverpool Road, Ashfield, NSW, Australia. Contact TIsHUG (Australia) Limited, P.O. Box 1089, Strawberry Hills, NSW 2012, Australia.

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.

BASIC

Trigonometric functions

By REGENA

It is spring time, and farm work has replaced snow skiing for some members of my family. We were making plans for putting in a new pivot sprinkler system for alfalfa and had some measurements. With one known measured distance and two distances read from surveying instruments, we needed to calculate other distances. With distances we had, we could calculate the sine of an angle. Once the angle is known, the hypotenuse of the triangle times the cosine of the angle would give the distance we needed.

The TI computer came to the rescue again. Actually, I couldn't find our calculator with trigonometric functions, and a slide rule didn't give me the accuracy I wanted. (Yes, I really did dig out my slide rule to try to figure out what I wanted; but then I couldn't remember how to complete the calculations.) I found a book of mathematical tables and worked out the problem. Then, when I had to do the same thing with different measurements I realized it was time for a computer program.

This month's program actually has two parts. In the first section, you may enter an angle either in degrees or in radians. The computer will then print the sine, cosine, tangent, cotangent, cosecant and secant for the angle. The second section is the inverse function, or finding out the angle when a sine, cosine, tangent or cotangent is given.

The trigonometric functions for our computer are based on angles expressed in radians (pi radians is equal to 180 degrees). If you enter an angle in degrees, it must be converted to radians. Line 130 defines PI as 3.141592654. Lines 140-150 define functions R and D to convert an angle from degrees to radians and from radians to degrees. All screens show both degrees and radians.

Our built-in functions are SIN(A), COS(A) and TAN(A), where A is an angle expressed in radians. The other three functions are defined in Lines 170-190, each as a reciprocal of one of the built-in functions.

To define your own function use DEF, then print your function. Then, anywhere in your program you may use the function name and the computer will calculate the function. For example, Line 140 defines the function R(A) to be the variable A multiplied by PI (which has previously been defined) and divided by 180. This statement needs to be near the beginning of the program. Then, later in the program, I can use R(ALPHA) or R(30) or R(*anything*) without having to repeat the formula. I can print the function or use it in a calculation, or even use it within another formula.

Another function I defined in Line 200 is P(X) which will print a number X to four decimal places. Remember, the TI will print more decimal places if you prefer.

The only inverse function built in on our TI is the arctangent, or ATN(X), which means find the angle whose tangent is X. You may also need the arcsine or inverse sine function. In fact, this is the function I needed for our pivot calculations. This function is defined as ARCSIN(X) in Line 160 in terms of the ATN function.

Line 210 redefines character 64 so a printed infinity sign may be used for certain angles.

Lines 230-320 print the options of finding the trigonometric functions of given angles or using the inverse trig functions, or ending the program; then the computer will branch appropriately.

Lines 330-370 contain a subroutine to print the message to press the Enter key and then the computer will continue.

Lines 390-620 receive user input for the angle, either in degrees or in radians. The angle will then be printed in degrees and radians on the screen. Lines 630-800 print the sine, cosine, tangent, cotangent, secant and cosecant of the given angle.

Lines 840-870 allow you to enter a numerator and denominator of a fraction which would be a sine, cosine, tangent or cotangent of an angle. If you already know the decimal equivalent, you may enter the decimal value for the numerator and "1" for the denominator.

Lines 880-1000 print the angle whose sine, cosine, tangent or cotangent is the given fraction. Remember that the sine and cosine must be a number from 0 to 1, so if a larger number is entered, only the inverse tangent and cotangent are given.

Here is an example problem. I knew the measurements 51 feet and 795.33 feet as the short leg and hypotenuse of a triangle and wanted the distance of the other leg. RUN the program and select inverse functions. Enter 51 for the numerator and 795.33 for the denominator. The computer then prints the ARCSIN, which is the angle whose sine is 51/795.33. It turns out to be a very small angle, 3.6766 degrees. I could then select trig functions and find the COSine of that angle. Multiply the cosine by 795.33 feet, and I get the distance I need (7.93.693 feet).

If you want 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 Trig Functions for the TI and whether you want cassette or diskette.

TRIG FUNCTIONS

```
100 REM TRIG FUNCTIONS !233
110 REM BY REGENA !071
120 REM !186
130 PI=3.141592654 !096
140 DEF R(A)=A*PI/180 !012
150 DEF D(B)=B*180/PI !000
160 DEF ARCSIN(X)=ATN(X/SQR(
-X*X+1))!081
170 DEF SEC(A)=1/COS(A)!009
180 DEF CSC(A)=1/SIN(A)!012
190 DEF COT(A)=1/TAN(A)!027
200 DEF P(X)=INT(10000*(X+.0
0005))/10000 !018
210 CALL CHAR(64,"006C929292
```

(See Page 8)

REGENA ON BASIC —

(Continued from Page 7)

```
6C")!065
220 REM !186
230 CALL CLEAR !209
240 PRINT "CHOOSE:" !101
250 PRINT : :"1 TRIGONOMETRI
C FUNCTIONS" !246
260 PRINT : :"2 INVERSE TRIG
 FUNCTIONS" !130
270 PRINT : :"3 END PROGRAM"
 !060
280 CALL SOUND(100,1000,2)!1
70
290 CALL KEY(3,K,S)!190
300 IF (K<49)+(K>51)THEN 290
 !055
310 CALL CLEAR !209
320 ON K-48 GOTO 390,840,103
0 !001
330 REM !186
340 PRINT : : :"PRESS <ENTER
> TO CONTINUE.";!107
350 CALL KEY(3,K,S)!190
360 IF K<>13 THEN 350 !084
370 RETURN !136
380 REM !186
390 PRINT "TRIGONOMETRIC FUN
CTIONS" !057
400 PRINT : : :"GIVEN ANGLE
IN" !071
410 PRINT : :"D  DEGREES" !0
90
420 PRINT : :"R  RADIANS" !1
07
430 CALL SOUND(100,1000,2)!1
70
440 CALL KEY(3,K,S)!190
450 IF (K<>68)+(K<>100)+(K<>
82)+(K<>114)=-4 THEN 440 !12
3
460 UNIT=1 !248
470 IF (K=68)+(K=100)THEN 49
0 !041
480 UNIT=2 !249
490 CALL CLEAR !209
500 IF UNIT>1 THEN 570 !051
510 PRINT "ENTER ANGLE IN DE
GREES": :!190
520 INPUT ALPHA !248
530 PRINT : : :ALPHA;"DEGREE
S": :!012
540 THETA=R(ALPHA)!089
550 PRINT :THETA;"RADIANS":
:!181
560 GOTO 620 !189
570 PRINT "ENTER ANGLE IN RA
DIANS": :!193
580 INPUT THETA !008
590 PRINT : : :THETA;"RADIAN
S" !181
600 ALPHA=D(THETA)!075
610 PRINT :ALPHA;"DEGREES" !
056
620 PRINT : :!006
630 X1=P(SIN(THETA))!185
640 PRINT " SIN(A) =";X1 !16
2
650 X2=P(COS(THETA))!181
660 PRINT " COS(A) =";X2 !15
8
670 X3=P(TAN(THETA))!189
680 PRINT " TAN(A) =";X3 !15
7
690 IF X3<>0 THEN 720 !210
700 PRINT " COT(A) = @" !195
710 GOTO 730 !043
720 PRINT " COT(A) =";P(COT(
THETA))!155
730 IF X2<>0 THEN 760 !249
740 PRINT " SEC(A) = @" !184
750 GOTO 770 !084
760 PRINT " SEC(A) =";P(SEC(
THETA))!133
770 IF X1<>0 THEN 800 !033
780 PRINT " CSC(A) = @" !182
790 GOTO 810 !124
800 PRINT " CSC(A) =";P(CSC(
THETA))!129
810 GOSUB 340 !165
820 GOTO 230 !053
830 REM !186
840 PRINT "ENTER A FRACTION
N/D": :!023
850 INPUT "NUMERATOR = ":NN
!240
860 INPUT "DENOMINATOR = ":D
D !097
870 PRINT : :!006
880 F=NN/DD !236
890 IF ABS(F)>1 THEN 960 !24
8
900 ACS=ARCSIN(F)!008
910 PRINT "ARCSIN(F) =";P(AC
S);"RADIANS" !238
920 PRINT TAB(12);P(D(ACS));
"DEGREES" !172
930 ACCS=(PI/2)-ACS !054
940 PRINT :"ARCCOS(F) =";P(A
CCS);"RADIANS" !225
950 PRINT TAB(12);P(D(ACCS))
;"DEGREES" !239
960 PRINT :"ARCTAN(F) =";P(A
TN(F));"RADIANS" !068
970 PRINT TAB(12);P(D(ATN(F)
));"DEGREES" !084
980 ACCT=(PI/2)-ATN(F)!223
990 PRINT :"ARCCOT(F) =";P(A
CCT);"RADIANS" !227
1000 PRINT TAB(12);P(D(ACCT)
);"DEGREES" !240
1010 GOSUB 340 !165
1020 GOTO 230 !053
1030 CALL CLEAR !209
1040 END !139
```

(See Page 9)

THE ART OF ASSEMBLY — PART 12

Getting the most from VDP RAM

By BRUCE HARRISON

As we mentioned at the close of No. 11 in this series, there is a great and largely untapped resource of memory in the 10,198 bytes of VDP Ram (>1000 through >37D6) that are not normally used by the E/A module. We have devised a number of ways to use this resource that were never mentioned by TI. Today we'll share some of those ideas with our readers, in hopes they'll become more innovative in this regard.

Several years ago, when we started to write our own Word Processor on the TI, it very quickly became obvious that we would

(See Page 9)

ART OF ASSEMBLY—

(Continued from Page 8)

run out of memory if we tried keeping the working text in a main memory buffer space. That was of course what both the E/A editor and TI-Writer did, and doing so placed limits on what features the editor could contain. It also contributed to the need for a separate "formatter" program to do printing with TI-Writer.

Also early in the development of the WP we decided that a single document should be able to be many pages in length while still being part of one document. The answer to our dilemma lay in the VDP RAM memory. We designed the WP so that only one page would be kept in the computer at one time, and so that the text of that page would actually exist only in VDP RAM, as a series of screens. Thus the first screenful of a document page resides at >1000 in VDP RAM, the next screenful at >1400, then >1800, and so on. As the user scans through a page, we "flip" the screens by a simple VWTR operation, setting VDP Register 2 to different values so that the output to the monitor is different sections of the VDP RAM memory.

This same concept was used in our game program "SCUD BUSTERS" for getting and displaying the on-screen instructions for the user. As the file of instructions was being read from the disk, the lines of text were "displayed" into 40-character screens starting at >1000 and moving up by >400 for each new screenful. Once all the text was present in the higher VDP RAM locations, we switched VDP Register 2 so that the screen at >1000 would be visible, then switched upward by >400 for each successive screen.

Another potential use for VDP RAM that we haven't tapped yet is the concept of having multiple character sets loaded there, and thus being able to "instantly" switch in a whole new character set by writing to VDP Register 4. The Pattern Table address advances in >800 increments, so one could conceivably load one character set of 256 definitions at >800 (the default location under E/A), load another set at >1000, another at >1800, and so on. Then one could switch whole character sets by a VWTR operation. Let's just say we had an alternate character set at >1000, and wanted to instantly switch over to that set. The code to do so would look like this:

```
LI    R0,>0402
BLWP  @VWTR
```

Then to switch back to the normal character set, one could:

```
LI    R0,>0401
BLWP  @VWTR
```

Simple, eh? We mentioned that we haven't tried this, and we haven't. We have done such a switch on a one-time basis when entering an Assembly program through an Extended Basic loader, so that the dreaded "offset" would not be in effect on our characters going to the screen. The "LOAD" program for our Word Processor does just that. It uses a GPLLNK service to load the standard characters in the >800 area, then switches to text mode and changes VDP Register 4 to use the character set at >800 before putting the "LOADING MAIN PROGRAM" message on screen.

Before we go further with this treatment of VDP RAM, let's

(See Page 10)

ART OF ASSEMBLY—

(Continued from Page 9)

issue a "warning" to XB Programmers. XB has its own uses for VDP RAM space, and any program subroutine that must return to XB should not tamper with the space above >1000. In one of our first attempts to mix Assembly subroutines with an XB program, we ran afoul of this problem. We were usng an Assembly subroutine to do a sound effect by the "Sound List" method, placing our sound list data at what we thought was a safe location in VDP RAM. Before long, strange effects were happening in the Extended Basic part of the program. Weird characters were appearing on the screen, and string variables were getting lost. After trying a number of different locations for our sound list, we concluded that, while XB was in command, there was no safe place to tamper with VDP RAM.

We finally resorted to changing the subroutine so that, just before putting the sound list (132 bytes) into VDP RAM, it would do a VMBR into our own low-memory space, thus stashing the contents of that area. After the sound effect ended, we did a VMBW to put back what was in that place in VDP RAM before our sound effect. This worked.

Sound lists are another potential use for the unused VDP RAM space. Normally, we don't use the "sound list" process at all, because the limitation of working with note durations in 60ths of a second is a killer for the classical music we've programmed. In the game SCUD BUSTERS, we made an exception to our normal rule so that sound effects like the Patriot's launching sound could proceed in the "background" while the CPU was busy checking for coincidence between the sprite representing the Patriot and that representing the Scud. We used sound lists for all the "noises" in that game, but the Theme at the beginning was done by our "normal" method of sending bytes directly to the sound generator. The Theme would have been impossible in sound list format, because of its rapid pace, quick-decaying drumbeats, and so on.

Of course VDP RAM is also commonly used for "PABs and Buffers" as mentioned in the E/A manual. Some care must be taken with the choice of locations for PAB and Buffer, particularly if Memory Image files are being loaded, since these will fill large blocks at the buffer location.

In a couple of our smaller utility programs, such as the ones we use to transfer text files from and to our PC computers, we actually used the lower part of the viewable screen as the buffer. This is somewhat unorthodox, but it had the singular advantage that one could actually see the text lines being transferred right on the screen. In that case the PAB was located at an "off screen" location, since much of the PAB content is "undefined" characters.

If push really came to shove, one could even put sections of data into VDP RAM instead of in the main memory. The only drawback to this use is that access to these data items would take considerably more time than getting them from main memory takes. One could even do a kind of "dynamic allocation" in this way, where different sections of data could be brought into main memory while they're needed, then written back to VDP for storage while another section of data is brought in for another part of program operation.

The same concept could be used for program "overlays", in that several small sections of code could be brought into VDP RAM as a memory image file, then transferred to main memory for execution when needed. Again this is a concept that we haven't actually tried, but it should be able to work.

Okay, dear readers, put your thinking caps on, and see if you can actually apply some of these ideas in your programs, or perhaps come up with more ways to use VDP RAM and share them with others.

Today's is a relatively short article, with no sidebar full of source code, but we hope this "food for thought" will take hold. Next time we're going to delve into the mysteries of Random Number generation, and will provide some directly useful source code for that purpose.

EXTENDED BASIC (plus)

Unbasher breaks down programs automatically

By BARRY TRAVER

From Jerry Stern's "SUBINDEX: A Subprogram Library in XB" in the November 1988 issue of MICROpendium to his "SUBINDEX Two: The Search for Tokens" in the February 1992 issue, readers of this magazine have benefitted much in the realm of Extended BASIC programs and Extended BASIC programming. He has made a very substantial contribution during the three years or more that he served as an Extended BASIC columnist for MICROpendium.

It is thus with some hesitancy that I approach writing regularly here in the same area. Although publisher John Koloen has officially announced that I "will replace Jerry as ... Extended BASIC columnist," there is no way that I can truly "replace" Jerry in any strict sense of the word. Unlike Robert Goldsborough (who has done an able job of persuading people that he is Rex Stout, creator of fictional detective Nero Wolfe), I do not have the talent necessary that I could dare hope to per

(See Page 11)

EXTENDED BASIC (plus)—

(Continued from Page 10)

suade you that I am Jerry Stern. I'm not very good at doing "imitations," particularly of people who may have more expertise than I may have in certain areas. No, in reality Jerry and I are both "originals," and you will not be in danger of mistaking one of us for the other. My hope, however, is that you will find that each of us has distinct contributions to make.

SUBPROGRAMS

On the other hand, I do expect to be building on the foundation that Jerry has laid, at least in certain important respects. For that reason, I strongly recommend that (if you have not already done so) you send away for the disk of 110 XB subprograms that Jerry wrote previously for this column. (Just send $6 to MICROpendium, P.O. Box 1343, Round Rock, TX 78680, specifying that you want the Jerry Stern disk.) I expect to be adding new subprograms to those that Jerry has written (taking care not to overlap line numbers), but I also expect (when appropriate) to be making use of subprograms Jerry has already done. If you are serious about Extended BASIC programming, I believe you will want to work on building an increasing library of useful TI XB subprograms.

While we're on the subject of subprograms, another resource that I believe every XB programmer ought to have is the "Nuts 'n' Bolts" series of subprograms by Jim Peterson. (If you don't have this three-disk set and would like to order it, it's available for $15 from Tigercub Software, 156 Collingwood Avenue, Whitehall, OH 43213.) Jim has graciously granted permission to us to include in this column one of his subprograms from time to time, but you will benefit greatly from having the entire set.

By the way, with the "Nuts 'n' Bolts" subprograms, you may expect to run into some problems with overlapping lines, but in a future column I may offer some suggestions about resequencing Jim's routines to resolve that problem. Also, Bruce Harrison and I have hopes of converting some of Jim's subprograms into assembly language subroutines accessible from TI XB through a CALL LINK, where the increased speed or power of assembly may significantly improve the performance.

As you may guess from the topic Jerry chose for his earliest and latest articles in his column, one very powerful feature of TI Extended BASIC is the ability to have genuine subprograms accessible by name with local variables and passed parameters (rather than mere subroutines with global variables which you access by a GOSUB to an arbitrary line number). It is a feature lacking from TI BASIC (a language prepared for TI by Microsoft) and from Microsoft's BASICA or GW-BASIC for the IBM and IBM clones, but shared by TI Extended BASIC with Microsoft's QuickBASIC for the IBM. Having genuine subprograms facilitates writing structured programs (rather than the "spaghetti code" that otherwise is so easy to write in BASIC), even though many XB programmers (and that also includes myself) often do not take full advantage of this feature.

SUBPROGRAMS SAVE TIME

As Jerry commented in his last article, one of the advantages of building a collection of subprograms is that it can "save you so much time that you'll have more time to watch those movie sequels, and more time to show off your finished software." It doesn't make sense to reinvent the wheel, and even less sense to rereinvent the wheel once you've reinvented it! A lot of programming is repetitive, and it can save a lot of time if you start writing a new program and can take advantage of your having many of the subcomponents already written (which is the situation when you have a collection of useful subprograms you can draw from). For example, many programs include menu screens. Why write such sections of code tediously from scratch, when you can instead make use of an appropriate menu subprogram you have already written?

I've described my column as "Extended BASIC (plus)," because I do intend to include from time to time suggestions of ways to extend Extended BASIC. As you know, I earlier wrote a column called "BASIC Assembly" (describing how to extend Extended BASIC through adding assembly subroutines accessible from XB using a CALL LINK). You may perhaps have guessed from that column that my first love is BASIC, not assembly. Evidence:

- The column included as much Extended BASIC code as assembly code (and even where assembly code was in the picture, it was often being created by such Extended BASIC programs as FONTALS, GRAPHICOMP, TIA/LINK, VDP/SAVER, etc.).
- My purpose was obviously not one of persuading people to leave TI XB for assembly, but one of persuading people to stay with TI XB, showing ways to work around the occasional limitations of BASIC as a language.

This new column, however, is not a reincarnation of the earlier "BASIC Assembly" series. The main subject here is Extended BASIC, plus. The "plus" means that I will also be presenting from time to time some occasional thoughts on ways to add to Extended BASIC, but adding your own assembly subroutines is only one way to extend the language. Making use of alternative Extended BASICs (such as Harry Wilhelm's The Missing Link, available from Texaments) is another way. Nevertheless note that the "plus" is in lower case, because the emphasis here will be upon Extended BASIC itself (although I will be assuming that most readers will have access not only to TI XB but also to a disk drive and memory expansion).

Well, so much for introductory comments on my intended approach. I just

(See Page 12)

EXTENDED BASIC (plus)—

(Continued from Page 11)

wanted to explain what I would be doing, so that you would know what to expect. Some of my columns here may offer a specific type of "food for thought," while others may maybe remind you of visiting a cafeteria or smorgasbord, but that is intentional: it is the way I think and write, and in that way I hope to offer "meaty" fare to a variety of readers. In other words, my desire is to try to make available "something for everyone" as best I can in this column. You may find things that don't interest you, but hopefully you will also find many things that do.

UNBASHER

I do have a program for you this month that I consider to be one of the most useful programs I have written. At least it's one I use myself all the time, and could not easily get along without. It's called UNBASHER, and its specific purpose is to break an XB program down as much as possible into single-statement lines so that you have room in which to work when you are revising a program that has been "squished" using a program like XBasher, SMASH!, Compactor Plus, or the like.

The purpose of those programs is to squeeze as much as possible into each line of an Extended BASIC program. That's fine, if the program has already been perfected. But what if you want to make revisions (especially those that may involve additions to the code)? If each line already holds as much as it possibly can, it may be difficult to revise one of the statements in some of those multi-statement lines, and even more difficult to add a new statement in the middle of an already-full multi-statement line!

Well, cheer up. UNBASHER can solve the problem for you. Just save your XB program to disk in MERGE format (e.g., using SAVE DSK1.PRG1, MERGE) and then run UNBASHER. (It is often good to RESequence your program before saving it to disk in MERGE format, using, e.g., RES ,20 to "open up" the program a bit. If all goes well, UNBASHER will create a new MERGE file for you (e.g., DSK1.PRG2). The next step is to enter NEW and then MERGE the new file into memory (e.g., using MERGE DSK1.PRG2), with the result being an XB program that will give you room in which to work in your revising and reworking. (Of course, once you've perfected the revised version, you can run it again through XBasher, SMASH!, or whatever your favorite compactor program happens to be.)

UNBASHER is fairly self-explanatory (especially if you've absorbed Jerry's previous comments in this column about MERGE format and Extended BASIC tokens), which is a good thing, because I've run out of time and space for this month. I hope you find the program useful in your own XB programming, and I'll look forward to meeting you again next month. Until then, keep on compuTIn'!

UNBASHER

```
100 ! COPYRIGHT (C) 1992 by
Barry Traver, 835 Green Vall
ey Drive, Philadelphia, PA 1
9128 (phone: 215/483-1379) -
- ALL RIGHTS RESERVED! !188
110 CALL CLEAR :: CALL BLUE
!228
120 DISPLAY AT(1,1):"UNBASHE
R (vers. 2.1)": :"  Copyrigh
t (C) 1992":"  by Barry A. T
raver": : :"Smashed and Bash
ed programs" !126
130 DISPLAY AT(8,2):"are gre
at for running, but":" not g
reat for revising and":" upd
ating.  There's no room":" t
o add anything within the" !
176
140 DISPLAY AT(11,2):"to add
 anything within the":" mult
i-statement lines!": :"This
program will allow you":" to
 take apart an XB multi-" !1
81
150 DISPLAY AT(16,2):"statem
ent line program, so":" that
 (except for some com-":" pl
ex IF/THEN constructions":"
that are better left alone"
!242
160 DISPLAY AT(20,2):"becaus
e of the logic) your":" new
version is composed of":" si
ngle-statement lines." :: CA
LL PAUSE !128
170 DISPLAY AT(1,1)ERASE ALL
:"To use UNBASHER, first sav
e":" the program you want to
 be":" ""unbashed"" as a MER
GE file" !094
180 DISPLAY AT(4,2):"(e.g.,
do an OLD DSK1.PROG":" and S
AVE DSK1.PRG1,MERGE).": :"Be
fore you save the program":"
 to disk,  it may be a good"
 !135
190 DISPLAY AT(9,2):"idea to
 RESequence it  (if":" you h
ave not  already done":" so)
 to make  sure that you":" w
ill have enough line num-" !
062
200 DISPLAY AT(13,2):"bers a
vailable.": :"Here's a possi
ble sequence:": :" OLD DSK1.
PROG":" RES (or RES ,20 if n
eeded)" !022
210 DISPLAY AT(19,2):"SAVE D
SK1.PRG1,MERGE": :"If you've
 done that, you're":" ready
to proceed!" :: CALL PAUSE !
194
220 DISPLAY AT(4,1)ERASE ALL
:"UNBASHER (vers. 2.1)": :"
 Copyright (C) 1992":"  by B
arry A. Traver" !081
230 DISPLAY AT(9,1):"INPUT F
ILE?  DSK1.PRG1" :: ACCEPT A
T(9,14)SIZE(-15):A$ !181
240 ON ERROR 250 :: OPEN #1:
A$,INPUT ,DISPLAY ,VARIABLE
163 :: ON ERROR STOP :: GOTO
 270 !020
250 ON ERROR 260 :: CLOSE #1
 :: ON ERROR STOP :: RETURN
230 !058
260 RETURN 230 !055
270 DISPLAY AT(10,1):"OUTPUT
 FILE? DSK1.PRG2" :: ACCEPT
AT(10,14)SIZE(-15):B$ !074
280 ON ERROR 290 :: OPEN #2:
B$,OUTPUT,DISPLAY ,VARIABLE
163 :: ON ERROR STOP :: GOTO
 310 !204
290 ON ERROR 300 :: CLOSE #2
 :: ON ERROR STOP :: RETURN
270 !140
300 RETURN 270 !096
310 IF EOF(1)THEN 320 ELSE L
INPUT #1:C$ :: GOSUB 330 ::
GOTO 310 !210
```

(See Page 13)

EXTENDED BASIC (plus)—

(Continued from Page 12)

```
320 CLOSE #1 :: CLOSE #2 ::
DISPLAY AT(15,1):"Now enter
the following:": :"NEW": :"M
ERGE ";B$: :"RES": :"SAVE ";
B$;:: STOP !043
330 L=1 :: D$=SEG$(C$,1,2)::
 GOSUB 430 :: GOSUB 440 :: C
$=SEG$(C$,3,LEN(C$)-2)!218
340 IF SEG$(C$,1,1)=CHR$(132
)THEN 410 !000
350 B=POS(C$,CHR$(130),L)::
IF B=0 THEN 410 !020
360 IF B>2 THEN IF SEG$(C$,B
-2,1)<>CHR$(201)THEN 400 !23
4
370 M=POS(C$,CHR$(201),1)::
IF M=0 THEN 400 ELSE M=ASC(S
EG$(C$,M-1,1))!183
380 IF M<>129 AND M<>134 AND
 M<>135 AND M<>148 AND M<>17
6 THEN 400 !117
390 L=L+1 :: GOTO 350 !065
400 PRINT #2:D$&SEG$(C$,1,B-
1)&CHR$(0):: GOSUB 430 :: GO
SUB 460 :: C$=SEG$(C$,B+1,LE
N(C$)-B):: GOSUB 420 :: GOTO
 340 !135
410 PRINT #2:D$&C$ :: GOSUB
430 :: GOSUB 460 :: RETURN !
250
420 C=ASC(SEG$(D$,1,1)):: D=
ASC(SEG$(D$,2,1)):: E=256*C+
D :: E=E+1 :: C=INT(E/256)::
 D=E-256*C :: D$=CHR$(C)&CHR
$(D):: RETURN !067
430 F=ASC(SEG$(D$,1,1)):: G=
ASC(SEG$(D$,2,1)):: H=256*F+
G :: RETURN !055
440 I=H :: IF I<>65535 THEN
DISPLAY AT(12,8):"READING ";
I !029
450 IF I<>0 THEN IF I<J+1 TH
EN 480 ELSE RETURN !233
460 J=H :: IF J<>65535 THEN
DISPLAY AT(13,8):"WRITING ";
J !075
470 RETURN !136
480 DISPLAY AT(17,1):"OVERLA
PPING LINES!": :"  RESEQUENC
E AND": :"    TRY AGAIN!" ::
 STOP !076
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
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
```

M Y - B A S I C

Video XOP6 and MY-BASIC (Part 2)

By JIM UZZELL

This month's program displays in greater detail the use of SETVECTORCOLOR(>10).

If you recall, in Part 1, I applied SETVECTORCOLOR to information that had been previously displayed by another routine, and that information changed colors. Then I reversed it. I am sure some of you were able to figure out where the color came from. Those who did not should find out with this month's demo.

In order for you to understand what happens in this demo, let

(See Page 14)

COLOR CHART

COLOR	MDOS CODE	MDOS G,R,B	MYBASIC CODE	MYBASIC R,G,B	CALL LINK <10 logic op 0 HEX	DECIMAL	CODE	PC COLOR
Transparent	0	0,0,0	1	0,0,0	0000	0	7	LT. GRAY *
Black	1	0,0,0	2	1,1,1	0100	256	0	
Green	2	6,1,1	3	1,7,1	0200	512	2	
Lt. Green	3	7,3,3	4	3,8,3	0300	768	A	
Dark Blue	4	1,1,7	5	1,1,8	0400	1024	1	
Lt. Blue	5	3,2,7	6	3,4,8	0500	1280	9	
Dark Red	6	1,5,1	7	6,1,1	0600	1536	4	
Cyan	7	6,2,7	8	2,7,8	0700	1792	3	
Med. Red	8	1,7,1	9	8,1,1	0800	2048	6	BROWN *
Lt. Red	9	3,7,3	10	8,3,3	0900	2304	C	
Dk. Yellow	A	6,6,1	11	7,7,1	0A00	2560	B	LT CYAN *
Lt. Yellow	B	6,6,4	12	7,7,4	0B00	2816	E	
Dk. Green	C	4,1,1	13	1,5,1	0C00	3072	D	LT MAGENTA *
Magenta	D	2,6,5	14	7,2,6	0D00	3328	5	
Gray	E	5,5,5	15	6,6,6	0E00	3584	8	
White	F	7,7,7	16	8,8,8	0F00	3840	F	

* Arbitrary assignemtn of PC colors to MYBASIC colors
The palette mixes to emulate PC colors in MYBASIC:
1(7,7,7) 9(7,5,4) 11(3,8,8) 13(8,3,7)

MY-BASIC—

(Continued from Page 13)

me explain some things:

First, I am using the default palette with the default TCOLOR(16,6) — white on blue.

Second, the demo first displays the default colors with both hexadecimal and decimal values of the default colors.

Third, the demo then displays the default colors for logic operation 3 and 4, with logic 3 overlaying part of the display.

With the above information and what is displayed on the screen when you run the program, it should become clear as day to you. You will need the object code file from Part 1 (see April 1992, MICROpendium) to run this demo.

Also included this month is the color chart I use. I know some of you, when you look at the chart, will be asking — what do PC colors have to do with the Geneve? Well, I have ported hundreds of PC windows icons to Geneve format and, hopefully, this will be the subject of a future article.

Coming next month is Part 3, SCROLLWIN.

```
1 !XOP6-10
100 CALL GRAPHICS(2,2)
110 CALL INIT :: CALL LOAD("
DSK.XOP6DEMO.DDIXOP")
120 CLS
130 CALL TCOLOR(16,13) :: DI
SPLAY AT(2,7):"  SetVectorCo
lor  >10  "; :: CALL TCOLOR(
16,6)
140 W=1 :: C=0 :: FOR Y=1 TO
 128 STEP 8 :: FOR X=1 TO 8
150 CALL LINK("START",16,20,
21+X+Y,106,21+X+Y,(C),0,0,6)
160 IF LEN(STR$(C))<4 THEN D
ISPLAY AT(3+W,5):HEX$ (C);C;
" "; :: GOTO 180
170 DISPLAY AT(3+W,5):HEX$ (
C);C;
180 NEXT X :: C=C+256 :: W=W
+1 :: NEXT Y
190 GOTO 220
200 CALL KEY(0,K,S) :: IF S=
0 THEN 200
210 CALL RESETPLT :: CALL SC
REEN(6) :: CALL TCOLOR(16,6)
 :: END
220 W=1 :: C=0 :: FOR Y=1 TO
 128 STEP 8 :: FOR X=1 TO 8
230 CALL LINK("START",16,63,
21+X+Y,86,21+X+Y,(C),3,0,6)
240 CALL LINK("START",16,110
,21+X+Y,160,21+X+Y,(C),3,0,6
)
250 CALL LINK("START",16,163
,21+X+Y,212,21+X+Y,(C),4,0,6
)
260 NEXT X :: C=C+256 :: W=W
+1 :: NEXT Y
270 CALL TCOLOR(16,13)
280 DISPLAY AT(20,2):"|_____
__0_____|____3___|____4___|
";
290 DISPLAY AT(21,2):"
      Logic op
";
300 CALL TCOLOR(16,6)
310 DISPLAY AT(22,1):"   3=B
kgnd  4=Forgnd  of Numbers
 ":"  Default Palette W/Whit
e on Blue"
320 CALL TCOLOR(4,14)
330 DISPLAY AT(24,1):"See Ag
ain--Press any key--  Type RU
N 120"
340 GOTO 200
```

What's there?

A disassembler in Extended BASIC

By JACQUES GROSLOUIS

Over the last few years a number of excellent articles have appeared in MICROpendium about programming in assembly language. Programmers who have worked with Extended BASIC and are now trying their hand at assembly language usually miss the ability to LIST a program. This allowed them to list to the screen or print out the program which they had run, and to study and learn from the techniques of other programmers or from their own mistakes. Unless one has the source code listing from which the program was assembled, this is often not possible with many programs written in assembly language or for assembly routines which are LINKed to an Extended BASIC or BASIC program. A disassembler program can be used to convert an assembly program back to its source code.

Although a number of such programs appeared in other publications about seven or eight years ago, a disassembler program has not to my knowledge previously appeared in MICROpendium. The program which follows has been adapted from some of these previous programs, some corrections were made, some bells and whistles have been added to make it more user friendly for a novice assembly language programmer like myself. The program is written in Extended BASIC and although some persons will say it is too slow, it does it job. The main reason for its lack of speed is that the command CALL PEEK returns a decimal number for each source code and this program converts it to a hexidecimal number for output to the screen, printer or disk. The CALL BASE subroutine written by Jerry Stern was used after making changes to use negative numbers and to ensure that output was either two or four hexidecimal characters long. Since most disassembly programs cannot recreate labels which were used when the source listing was first written, an idea used by Barry Travers was used to make all line numbers behave as labels by preceeding them with the letter "L".

A disassembly programs cannot usually distinguish code and text or data. Neither does this program, but all three are presented as output so that the user can decide which is being presented. Since the program attempts to disassemble all codes, some mighty strange operand codes can be displayed. Assembly code is usually one, two or three words of source code long

(See Page 15)

DISASSEMBLER—

(Continued from Page 14)

and this may hide a code which follows. An example of this occurs if you try to disassemble line 000C and 000E in console ROM. The correct listing is 000C DATA >30AA and 000E B @@>02B2. Disassemble starting at each address to see the difference. The presentation of text and source code makes it easy to disassemble the REF/DEF table which is usually at the top of low memory say from 3FD0 to 3FFF. The last two bytes of each eight byte block section indicates where the assembly routine will be foud for the named LINK reference.

Since this program is written in Extended BASIC it cannot be used to disassemble the top of high memory from about E000 to FFFF. This section of memory contains the extended basic program lines and line tables. Except for VDP ram and GROM, most other sections of memory can be accessed. This includes high memory from A000 to FFFF , low memory from 2000 to 3FFF, console ROM from 0000 to 1FFF and ROM contained in the XB cartridge from 6000 to 7FFF. The program also identifies most of the system utility equates such as VSBW and VWTR. The main menu screen indicates some common peek contents which can be useful to guide the user where to begin looking. If low memory was initialized from XB the top second and third peeks indicate the Last Free Address in low memory and the start of the REF/DEF table. Remember memory is not lost unless the whole system is turned off. This is best illustrated when you first boot up and run the disassembler program ,most memory locations will then contain BYTE >0000. However if you break your program and enter CALL INIT from the console and rerun the program, then low memory will contain code in locations from 2000 to 24F4 and the REF/DEF will indicate that it starts at 4000 or in other words it contains no LINK references. For this reason I prefer to save this program as DSK1.LOAD and load it from the title screen. Whatever was in low or high memory (below the XB program) is still there after the program loads.

The E/A peeks are only meaningful if memory was initialized from E/A or compatible software. The LOADER from E/A organizes low memory differently than the CALL INIT command from XBasic. These differences are well explained in the series of LOADER articles by Merle Vogt starting in the March 1989 issue of MICROpendium. Output to disk may prove useful to some readers to extract small sections from existing programs. It is doubtful that a full program could be disassembled and successfully reassembled by this method. If you can do this you are past the learning stage with assembly language and have probably written your own disassembler program which will run circles around this program. Output to the printer identifies and prints out unresolved line references which occur in the section disassembled. Printer output has been coded for an EPSON printer using elite mode, to skip page perferations and to form feed at the end of the output. the string variable PR$ in lines 170 and 1410 can be changed or deleted to suit your printer.

Even with the limitations of this program I find that I can now recognize assembly routines in other programs, I can identify where work spaces for registers have been blocked off and find that my knowledge of assembly is steadily improving. Hope that you will have the same results.

WHAT'S THERE

```
100 !SAVE DSK1.DISASS OR DSK
1.LOAD !028
110 GOTO 170 !249
120 A,A$,A1,A2,B,B$,C$,CO,D,
DEC,DK,D$,H,H$,I,J,J$,K,L,L$
,N,N$,NU,O1,O1$,O2,O2$,O3,O3
$,O4,O4$,P,PR,PR$,PT$,Q$,R,S
,TD,TR,TS,WR,X,X$,X1$,HD$,Y$
,Z,Z$,[,] !074
130 UR$ !203
140 DIM Q(6),UT$(12):: CALL
LINK :: CALL LOAD :: CALL PE
EK :: CALL KEY :: CALL CLEAR
 !187
150 !@P- !064
160 DATA NUMASG,NUMREF,STRAS
G,STRREF,XMLLNK,KSCAN,VSBW,V
MBW,VSBR,VMBR,VWTR,ERR !148
170 PR$=CHR$(27)&"M"&CHR$(27
)&"0"&CHR$(27)&"C"&CHR$(0)&C
HR$(11)&CHR$(27)&"N"&CHR$(8)
:: HD$="Label Code Operands
  Utility" !168
180 RESTORE 160 :: FOR A=0 T
O 11 :: READ UT$(A):: NEXT A
 !205
190 P,]=1 :: DK,TR,PR,[=0 ::
 CALL CLEAR :: CALL BLUE ::
X1$="24F4" :: Y$="4000" :: G
OTO 230 !157
200 IF PR=] THEN GOSUB 1600
:: PRINT #2:CHR$(12)&CHR$(27
)&CHR$(64):: CLOSE #2 :: PR=
[ :: UR$="" !072
210 IF DK=] THEN CLOSE #1 ::
 DK=[ !159
220 A$,B$,H$,L$,O1$,O2$,O3$,
O4$=" " :: GOSUB 1260 !138
230 GOSUB 1280 !085
240 CALL KEY(3,K,S):: IF K=3
2 THEN 240 ELSE IF K=15 THEN
 200 !228
250 IF A>=B THEN 200 !198
260 A1=A :: IF A>32767 THEN
A1=A-65536 !007
270 CALL PEEK(A1,H,L,O1,O2,O
3,O4)!057
280 N=H*256+L :: IF N>16383
THEN 770 !240
290 IF N>14335 THEN 690 !140
300 IF N>12287 THEN 640 !094
310 IF N>11263 THEN 690 !137
320 IF N>8191 THEN 690 !094
330 IF N>4095 THEN 560 !219
340 IF N>2047 THEN 540 !194
350 IF N>1023 THEN 490 !136
360 IF N>831 THEN 860 !209
370 IF N>511 THEN 390 !244
380 GOTO 880 !194
390 IF (L AND 16)=16 THEN 88
0 !004
400 RESTORE 1140 :: J=480 ::
 Z=5 :: K=32 :: R=(L AND 15)
:: N=((H AND 3)*256)+(L AND
224):: GOSUB 1040 !094
410 IF TR<>] THEN 440 !056
420 DEC=O1*256+O2 :: CALL BA
SE(DEC,16,4,N$):: C$=">"&N$
:: A=A+4 !198
430 B$="R"&STR$(R)&"," :: GO
TO 480 !238
440 Z=2 :: GOSUB 1040 :: IF
TR<>] THEN 460 !199
450 C$="" :: A=A+2 :: B$="R"
&STR$(R):: GOTO 480 !046
```

(See Page 18)

DISASSEMBLER—

(Continued from Page 15)

```
460 Z=2 :: GOSUB 1040 :: IF
TR<>] THEN 880 !109
470 DEC=O1*256+O2 :: CALL BA
SE(DEC,16,4,N$):: B$=">"&N$
:: A=A+4 :: C$="" !051
480 GOSUB 900 :: TR=[ :: GOT
O 240 !217
490 N=(H*256)+(L AND 192)::
J=960 :: Z=14 :: K=64 :: RES
TORE 1120 :: GOSUB 1040 !045
500 GOSUB 1060 :: GOSUB 1560
 !104
510 C$="" :: IF A$="B    " AN
D B$="*R11" THEN C$=" (SAME
AS RT)" !196
520 IF O1=32 AND O2<53 THEN
X=O2/4-2 :: IF INT(X+2)<>O2/
4 THEN C$=" " ELSE C$="
"&"(" UT$(X)&")" !172
530 GOSUB 900 :: TR=[ :: GOT
O 240 !217
540 N=(H AND 11)*256 :: J=17
92 :: CO=(L AND 240)/16 :: W
R=(L AND 15):: RESTORE 1160
:: Z=4 :: K=256 :: GOSUB 104
0 !000
550 B$="R"&STR$(WR)&"," :: C
$=STR$(CO):: A=A+2 :: GOSUB
900 :: TR=[ :: GOTO 240 !237
560 RESTORE 1170 :: J=3840 :
: TR=[ :: Z=13 :: K=256 :: N
=H*256 :: GOSUB 1040 !251
570 IF TR=[ THEN 600 !022
580 IF L>127 THEN L=L-256 !0
37
590 DEC=(2*L+2)+A :: CALL BA
SE(DEC,16,4,N$):: B$="L"&N$
:: GOSUB 1560 :: GOTO 620 !2
02
600 Z=3 :: K=256 :: GOSUB 10
40 !237
610 B$=STR$(L)!184
620 C$="" :: A=A+2 :: IF A$=
"JMP " AND L=0 THEN C$=" (SA
ME AS NOP)" !037
630 GOSUB 900 :: TR=[ :: GOT
O 240 !217
640 IF (H AND 252)=48 THEN A
$="LDCR" :: GOTO 670 !128
650 IF (H AND 252)=52 THEN A
$="STCR" :: GOTO 670 !146
660 GOTO 880 !194
670 GOSUB 1060 :: GOSUB 1560
 :: NU=((H AND 3)*4)+((L AND
192)/64)!229
680 C$=","&STR$(NU):: GOSUB
900 :: TR=[ :: GOTO 240 !023
690 RESTORE 1190 :: J=7168 :
: N=((H AND 60)*256):: D=((H
 AND 3)*4)+((L AND 192)/64):
: Z=3 :: K=1024 :: GOSUB 104
0 !177
700 IF TR<>] THEN 730 !091
710 C$=","&"R"&STR$(D)!047
720 GOSUB 1060 :: GOSUB 900
:: TR=[ :: GOTO 240 !211
730 IF N<>11264 THEN 750 !13
3
740 A$="XOP " :: C$=",XOP OP
 #"&STR$(D):: GOTO 720 !136
750 C$=" " :: J=13312 :: Z=2
 :: K=1024 :: GOSUB 1040 ::
IF TR<>] THEN 880 !107
760 GOTO 720 !033
770 RESTORE 1200 :: J=12288
:: N=(H AND 240)*256 :: Z=12
 :: K=4096 :: GOSUB 1040 !20
6
780 TD=(H AND 12):: D=((H AN
D 3)*4)+((L AND 192)/64):: G
OSUB 1060 !011
790 IF TD=[ THEN C$=","&"R"&
STR$(D):: GOTO 850 !058
800 IF TD=4 THEN C$=","&"*R"
&STR$(D):: GOTO 850 !007
810 IF TD=12 THEN C$=","&"R"
&STR$(S)&"+" :: GOTO 850 !19
8
820 IF (TD=8)AND(D=0)THEN DE
C=O1*256+O2 :: CALL BASE(DEC
,16,4,N$):: C$=","&"@"&"L"&N
$ :: A=A+2 :: IF TS=32 THEN
DEC=O3*256+O4 :: CALL BASE(D
EC,16,4,N$):: C$=","&"@"&"L"
&N$ !103
830 IF (TD=8)AND(D<>[)THEN X
=O1*256+O2 ELSE 850 :: CALL
BASE(X,16,4,N$):: C$=",@L"&N
$&"(R"&STR$(S)&")" :: A=A+2
!044
840 IF TS=32 THEN X=O3*256+O
4 :: CALL BASE(X,16,4,N$)::
C$=",@L"&N$&"(R"&STR$(D)&")"
 :: A=A+2 !012
850 GOSUB 1560 :: GOSUB 900
:: TR=[ :: GOTO 240 !201
860 N=(H*256+L):: A=A+2 :: B
$="" :: C$="" :: Z=6 :: J=80
0 :: K=32 :: RESTORE 1220 !0
57
870 GOSUB 1040 :: GOSUB 900
:: TR=[ :: GOTO 240 !191
880 A$="BYTE" :: DEC=H*256+L
 :: CALL BASE(DEC,16,4,B$)::
 B$=">"&B$ :: C$=" " :: A=A+
2 :: GOSUB 900 !202
890 GOTO 240 !063
900 Q(])=H :: Q(2)=L :: Q(3)
=O1 :: Q(4)=O2 :: Q(5)=O3 ::
 Q(6)=O4 :: A2=A-A1 :: IF A1
<[ THEN A2=A-A1-65536 !002
910 IF A2<0 THEN A2=A1-A !00
5
920 IF O4>32 AND O4<128 THEN
 O4$=CHR$(O4)ELSE O4$=" " !0
02
930 IF O1>32 AND O1<128 THEN
 O1$=CHR$(O1)ELSE O1$=" " !2
43
940 IF O3>32 AND O3<128 THEN
 O3$=CHR$(O3)ELSE O3$=" " !2
53
950 IF O2>32 AND O2<128 THEN
 O2$=CHR$(O2)ELSE O2$=" " !2
48
960 IF H>32 AND H<128 THEN H
$=CHR$(H)ELSE H$=" " !219
970 IF L>32 AND L<128 THEN L
$=CHR$(L)ELSE L$=" " !239
980 Z$=SEG$(H$&L$&O1$&O2$&O3
$&O4$,],A2):: CALL BASE(A1,1
6,4,Q$)!098
990 PT$="L"&Q$&" "&A$&" "&B$
&C$ :: PRINT PT$;TAB(23);Z$
!197
1000 FOR P=] TO A2 :: X=Q(P)
:: CALL BASE(X,16,2,X$):: PR
INT ;TAB(2);">"&X$&" ";:: NE
XT P !045
1010 IF PR=] THEN GOSUB 1230
 !019
1020 IF DK=] THEN GOSUB 1440
 !211
1030 PRINT :: RETURN !166
1040 FOR I=] TO Z :: J=J+K :
: READ Q$ :: IF N=J THEN A$=
Q$ :: TR=] !080
1050 NEXT I :: RETURN !233
1060 S=(L AND 15):: TS=(L AN
D 48):: IF TS=[ THEN B$="R"&
STR$(S):: A=A+2 :: RETURN !2
13
1070 IF TS=16 THEN B$="*R"&S
TR$(S):: A=A+2 :: RETURN !01
4
```

(See Page 19)

DISASSEMBLER—

(Continued from Page 18)

```
1080 IF TS=48 THEN B$="*R"&S
TR$(S)&"+" :: A=A+2 :: RETUR
N !190
1090 IF (TS=32)AND(S=[)THEN
DEC=O1*256+O2 :: CALL BASE(D
EC,16,4,N$):: B$="@L"&N$ ::
A=A+4 :: GOSUB 1560 :: RETUR
N !215
1100 IF (TS=32)AND(S<>[)THEN
 DEC=O1*256+O2 :: CALL BASE(
DEC,16,4,N$):: B$="@L"&N$&"(
R"&STR$(S)&")" :: A=A+4 :: G
OSUB 1560 :: RETURN !143
1110 BREAK !142
1120 DATA BLWP,"B   ","X   "
,"CLR ","NEG ","INV ","INC "
,INCT,"DEC " !251
1130 DATA DECT,"BL  ",SWPB,S
ETO,"ABS " !180
1140 DATA "LI  ","AI  ",ANDI
,"ORI ","CI  " !232
1150 DATA STWP,STST,LWPI,LIM
I !223
1160 DATA "SRA ","SRL ","SLA
","SRC " !247
1170 DATA "JMP ","JLT ","JLE
 ","JEQ ","JHE ","JGT ","JNE
 " !087
1180 DATA "JNC ","JOC ","JNO
 ","JL  ","JH  ","JOP ","SBO
 ","SBZ ","TB  " !230
1190 DATA "COC ","CZC ","XOR
 ","MPY ","DIV " !125
1200 DATA "SZC ","SZCB","S
 ","SB  ","C   ","CB  ","A
 ","AB  " !102
1210 DATA "MOV ",MOVB,"SOC "
,SOCB !076
1220 DATA IDLE,RSET,RTWP,CKO
N,CKOF,LREX !012
1230 PRINT #2:TAB(10);PT$;TA
B(46);Z$;TAB(55);!225
1240 FOR P=] TO A2 :: X=Q(P)
:: CALL BASE(X,16,2,X$):: PR
INT #2:">"&X$&" ";:: NEXT P
!014
1250 PRINT #2 :: RETURN !158
1260 DISPLAY AT(24,3):"ANY K
EY OR Q TO QUIT " :: CALL KE
Y(3,K,S):: IF S=[ THEN 1260
:: IF K=81 THEN IF DK=] THEN
 CLOSE #] ELSE END !201
1270 RETURN !136
1280 DISPLAY AT(],8)ERASE AL
L:"WHAT'S THERE?" :: DISPLAY
 AT(3,1):"Some Peeks" :: GOS
UB 1470 !234
1290 DISPLAY AT(16,5):"Start
 Address ?  >"&X1$ :: DISPLA
Y AT(17,4):"To End Address ?
  >"&Y$ :: ACCEPT AT(16,23)S
IZE(-4)VALIDATE(DIGIT,"ABCDE
F")BEEP:X1$ !239
1300 CALL BASE10(A,16,X1$)::
 IF A=[ THEN 1310 :: IF A/2<
>INT(A/2)THEN 1290 !030
1310 ACCEPT AT(17,23)SIZE(-4
)VALIDATE(DIGIT,"ABCDEF")BEE
P:Y$ :: CALL BASE10(B,16,Y$)
!225
1320 DISPLAY AT(19,5):"Send
To Printer ?>N" :: ACCEPT AT
(19,23)SIZE(-])VALIDATE("YNy
n")BEEP:X$ !194
1330 IF X$="Y" OR X$="y" THE
N DISPLAY AT(20,1):"TITL >"
:: ACCEPT AT(20,7)SIZE(21):B
$ :: A$="TITL " :: C$=" " !0
24
1340 DISPLAY AT(21,5):"Save
To Disk    ?>N" :: ACCEPT AT
(21,23)SIZE(-])VALIDATE("YNy
n")BEEP:J$ !178
1350 IF J$="Y" OR J$="y" THE
N DK=] ELSE 1380 !005
1360 DISPLAY AT(22,]):"NAME
OF FILE>DSK1.TEST/S" !040
1370 ACCEPT AT(22,14)SIZE(-1
5)VALIDATE(UALPHA,DIGIT,"/")
BEEP:D$ :: OPEN #]:D$,UPDATE
,DISPLAY ,VARIABLE 80 !113
1380 DISPLAY AT(23,5):"OK To
 Continue  ?>Y" :: ACCEPT AT
(23,23)SIZE(-1)VALIDATE("YyN
n")BEEP:Q$ :: IF Q$="N" OR Q
$="n" THEN DISPLAY AT(20,5):
: GOTO 1290 !200
1390 PRINT :: PRINT " HOLD S
PACE BAR TO PAUSE": :" HOLD
FCTN 9 TO ESCAPE" :: PRINT !
166
1400 IF X$="Y" OR X$="y" THE
N PR=] ELSE 1430 !102
1410 OPEN #2:"PIO" :: PRINT
#2:PR$;TAB(10);HD$;TAB(46);"
Text";TAB(55);"Source Codes"
&CHR$(10)!045
1420 IF B$<>" " THEN PRINT T
AB(7);A$&B$ :: PRINT #2:TAB(
16);A$;B$ !213
1430 RETURN !136
1440 PRINT #]:PT$;TAB(36);Z$
;TAB(45);!249
1450 FOR P=] TO A2 :: X=Q(P)
:: CALL BASE(X,16,2,X$):: PR
INT #]:">"&X$&" ";:: NEXT P
!112
1460 PRINT #] :: RETURN !000
1470 CALL PEEK(8192,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(4,15):
"@>2000 >"&X$ !184
1480 CALL PEEK(8194,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(5,3):"
XB LINK LFA @>2002 >"&X$ !14
4
1490 CALL PEEK(8196,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(6,7):"
REF/DEF @>2004 >"&X$ !149
1500 CALL PEEK(-31866,A,B)::
 DEC=A*256+B :: CALL BASE(DE
C,16,4,X$):: DISPLAY AT(8,2)
:"START XB PGM @>8386 >"&X$
!014
1510 CALL PEEK(8228,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(10,4):
"E/A FSTHI  @>2024 >"&X$ !08
7
1520 CALL PEEK(8230,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(11,8):
"LSTHI  @>2026 >"&X$ !132
1530 CALL PEEK(8232,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(12,8):
"FSTLOW @>2028 >"&X$ !196
1540 CALL PEEK(8234,A,B):: D
EC=A*256+B :: CALL BASE(DEC,
16,4,X$):: DISPLAY AT(13,8):
"LSTLOW @>202A >"&X$ !214
1550 DISPLAY AT(15,1):"Disas
semble From" :: RETURN !146
1560 IF N$>=X1$ AND N$<=Y$ T
HEN RETURN !128
1570 P=POS(UR$,N$,1):: IF P=
0 THEN UR$=UR$&N$ !044
1580 IF LEN(UR$)>120 THEN 20
0 !240
1590 RETURN !136
1600 PRINT #2 :: PRINT #2:TA
B(10);"*UNRESOLVED REFERENCE
S" !183
```

(See Page 20)

DISASSEMBLER—

(Continued from Page 19)

```
1610 FOR I=1 TO LEN(UR$)STEP
 4 :: PRINT #2:TAB(10);"L"&S
EG$(UR$,I,4):: NEXT I :: RET
URN !000
1620 !@P+ !062
1630 SUB BLUE !149
1640 CALL SCREEN(5):: FOR L=
0 TO 14 :: CALL COLOR(L,16,1
):: NEXT L :: SUBEND !202
1650 SUB BASE(D,B,L,N$)!082
1660 T=D :: N$="" !207
1670 IF D<[ THEN T=T+65536 :
 : GOTO 1720 !030
1680 Q=INT(T/B):: R=T-B*Q !1
63
1690 N$=SEG$("0123456789ABCD
EF",R+1,1)&N$ !133
1700 IF Q<>[ THEN T=Q :: GOT
O 1680 !041
1710 GOTO 1750 !043
1720 Q=ABS(INT(T/B)):: R=T-B
*Q !219
1730 N$=SEG$("FEDCBA98765432
10",16-R,1)&N$ !189
1740 IF Q<>[ THEN T=Q :: GOT
O 1720 !081
1750 IF L=2 AND LEN(N$)>2 TH
EN N$=SEG$(N$,3,2)!167
1760 IF LEN(N$)<2 THEN N$="0
"&N$ !244
1770 N$=SEG$(RPT$("0",4),L-L
EN(N$)+1,L-LEN(N$))&N$ !058
1780 SUBEND !168
1790 SUB BASE10(D,B,N$)!182
1800 H$="0123456789ABCDEF" !
163
1810 D=[ :: FOR I=1 TO LEN(N
$):: D=POS(H$,SEG$(N$,I,1),1
)-1+B*D :: NEXT I :: SUBEND
!024
```

Newsbytes

Horizon releases mouse, Digiport

The Horizon Mouse and Digiport were scheduled to debut at the Lima, Ohio, Multi-User Group Conference May 15, according to Bud Mills.

Mills says the Mouse retails for $40, with software by Mike Maksimik. He says it is superior to any other mouse for TI.

Digiport, by Don O'Neil, also retails for $40. According to O'Neil, it is a custom digital analog convrter that plays digital sounds on a standard TI with at least 32K, SSSD and RS232 (TI or Myarc), or on a Myarc Geneve 9640 with SSSD and RS232 and M-DOS .97H or 1.14.

Cor-Comp is not supported, O'Neil says, but users who own Cor-Comp and pur-chase it will get an upgrade if and when one is available.

Digiport allows the user to play digital sound from Mac, Atari, PC, Amiga, Sun and Next soundfiles at up to 44 Khz playback speed. The user can use standard 32K expansion or 9938/58 VDP memory expansion up to 192K or RAMBO compatible such as Horizon 3000 with the RAMBO mod or 4A Memex or Geneve memory expansion.

Included with the package, according to O'Neil, are custom cable and 10 sound disks of your size choice (SSSD, DSSD or DSDD), a scripting utility for 9938/58 or Geneve and the sound file player.

For further information, or to order, con-tact Bud Mills Services, 166 Dartmouth Dr., Toledo, OH 43614-2911, or (419) 385-5946 (voice) or (419) 385-7484.

Transfer sheets allow T-shirt design with TI

Transfer sheets are available which allow the user to make press-on designs for garments with a computer, graphics software and a dot matrix printer, according to the manufacturer.

Four 8x11-inch sheets are available for $9.95; 10 sheets, $17.95; and 20 sheets, $32.95. Shipping and handling is $1.50, and New Jersey residents add 7 percent sales tax.

To order, send a check or money order to The Computer T-Shirt Shop, Ste. 157, 1050 State Hwy. 35, Shrewsbury, NJ 07702.

Australian fair date changed to Nov. 28-29

The TI-Faire sponsored by TIsHUG (Australia) Limited has been rescheduled to Nov. 28-29.

The event will be held 10 a.m. to 9 p.m. both days at the Ashfield Boys High School Hall (next to Western Suburbs Leagues Club), Liverpool Road, Ashfield, New South Wales.

According to organizers, the site is accessible by bus or train and ample parking is also available for automobiles. Restaurants and take-out food sites are near-by, and the Leagues Club offers snack bar facilities.

Activities planned include displays of the latest TI99/4A hardware and software (available for sale at the end of the fair); sponsor displays, both local and overseas; displays of rare items of TI99/4A hardware from the past; displays of owner-built peripherals and expansion systems; a computer-controlled (TI99/4A) model train layout; RAMdisk and EPROM RAMdisk demonstrations; Page Pro 99 demonstrations; greeting cards production; and computer speech demonstration.

Also, a bulletin board system (BBS) demonstration; demonstration of 80-column card and color monitor capabilities; latest TI99/4A computer games demonstrations; MIDI music demonstration; screenings of videos from past TI fairs in the United States; sales of TI99/4A software and hardware from the TIsHUG Shop; a TI99/4A computer repair clinic and technical support; and a special edition of the group's newsletter.

Several nearby motels accept all major credit cards, according to TI-Faire organizers.

For further information, write TIsHUG (Australia) Limited, P.O. Box 1089, Strawberry Hills, NSW 2012, Australia. Phone contacts are available at (02) 918-8132, (02) 871-1514 and (042) 85-5519.

Fax number is (02) 638-9757, attention Tom Marshall, available 9 a.m. to 5 p.m. weekdays, New South Wales time.

1991 MICROpendium Index

Installment covers second half of 1991

Listed below is the second installment of the 1991 MICROpendium Index by Elton Schooling. The index runs out of Extended BASIC. For information about how to use it, see the April 1992 MICROpendium. The index, covering 1984-1991 is available from MICROpendium on two disks for $6. Use the order form printed elsewhere in this issue.

1991 Index (July-Dec.)

```
10 REM INDEX 91B MICROpendiu
m INDEX for 1991, JUL to DEC
, Publisher John Koloen, edi
tor Laura Burns. !020
20 REM Compiled by Elton Sch
ooling, 4014 57th St., Sacra
mento, CA 95820 !173
30 REM Sort routine by David
 Romer and John Clulow. Ob-
tained from Boston Computer
Soc., TI994/A User Group. Fo
r use with printer or with !
254
32 REM screen display. !126
35 REM Because of many entri
es the '91 index is divided
into '91A, Jan. to June, and
 '91B, July to Dec. !104
40 REM For your printer you
may need to change line 160.
 !202
50 REM For longer dwell time
 on screen increase the DELA
Y number in line 330. !210
52 CALL INIT !157
54 CALL CLEAR !209
56 CALL LOAD("DSK1.SORT")!07
9
60 OPTION BASE 1 !137
70 CALL CLEAR !209
80 DIM N$(106)!203
90 INPUT "OUTPUT TO PRINTER?
(Y/N)":P$ !247
100 CALL CLEAR !209
110 PRINT "WORKING" !139
120 FOR I=1 TO 106 :: READ N
$(I):: NEXT I !066
130 CALL LINK("SORT",N$(),10
6)!190
140 CALL CLEAR !209
150 IF P$="Y" THEN 160 ELSE
290 !093
160 OPEN #1:"PIO" !253
170 PRINT #1:TAB(24);"MICROp
endium INDEX, 1991B, JUL to
DEC" !013
180 PRINT #1: : : :!103
190 FOR J=1 TO 106 :: IF J=1
05 THEN 200 ELSE 220 !119
200 PRINT #1: : : : :: PRINT
 #1:TAB(35);"PAGE 28" :: PRI
NT #1: : : : : : : : : : ::
GOTO 220 !201
210 PRINT #1: : : : : :: PRI
NT #1:TAB(31);"PAGE 29, INDE
X '91B" :: PRINT #1: : : : :
 : : : : :!145
220 IF J/2=INT(J/2)THEN 240
!249
230 PRINT #1:N$(J);:: GOTO 2
50 !240
240 PRINT #1:TAB(40);N$(J)!1
88
250 NEXT J !224
280 GOTO 360 !184
290 CALL CLEAR !209
300 CALL SOUND(500,110,0,131
,0,196,0)!005
310 PRINT TAB(7);"MICROpendi
um INDEX, 1991B" :: PRINT :
: :!060
320 PRINT "DATE AND PAGE NO.
 ARE LISTED TOGETHER. JAN 85
 p.16 BECOMES 1/85/16.": : :
!005
330 FOR J=1 TO 106 :: PRINT
N$(J):: FOR DELAY=1 TO 200 :
: NEXT DELAY :: NEXT J !020
340 PRINT : :!006
350 PRINT "DATE AND PAGE NO.
 ARE LISTED TOGETHER. JAN 85
 p.16 BECOMES 1/85/16." :: G
OTO 390 !062
360 PRINT #1: : :!178
370 PRINT #1:"DATE AND PAGE
NO. ARE LISTED TOGETHER. JAN
 85 p.16 BECOMES 1/85/16." !
146
375 PRINT #1: : : : : : : :
: : : :: PRINT #1:TAB(23);"M
ICROpendium Index, 1991B, Pa
ge 29" !147
380 CLOSE #1 !151
390 END !139
400 DATA BAS RISE AND FALL G
A 7/91/8,XBAS ROBOT TEACHER
7/91/11,BAS-A/L TI-ARTIST CA
LL LINK 7/91/14 !002
410 DATA ART OF A/L BOTTOM S
TART 7/91/22,TI CASINO GAME
REV 7/91/25,HARRISON MUSIC R
EV 7/91/25 !035
420 DATA ACCELERATOR SPEED G
AIN 7/91/26,SPEED GAIN ACCEL
ERATOR 7/91/26,TI-BASE USING
 SCOPE 7/91/28 !078
430 DATA MULTIPLAN 4.0 PATCH
ES USNO 7/91/29,MUSIC HARRIS
ON REV 7/91/25,HARD DRIVE HE
LP USNO 7/91/30 !171
440 DATA BAS SPATIAL RELATIO
NSHIPS 8/91/7,XBAS PHONE LIS
T 8/91/13,BAS A/L ACCESSING
A/L ROUTINES 8/91/20 !015
450 DATA ART OF A/L STARTING
 AT THE TOP 8/91/24,MARIO BR
OS GAME REV 8/91/27,TURBO 20
56 GAME REV 8/91/28 !232
460 DATA LINKAGE GAME REV 8/
91/28,MODULE TO DISK WITH F'
WEB USNO 8/91/29,BREAK KEY W
ITH XB/GENEVE USNO 8/91/29 !
104
470 DATA NEST .IF/PAPER SAVE
 TIWR USNO 8/91/29,FRENCH NE
WSLETTER ON DISK USNO 8/91/3
0,AUTO LOAD FROM TAPE USNO 8
/91/30 !241
480 DATA PHONE LIST XBAS 8/9
1/13,F'WEB MODULE TO DISK US
NO 8/91/29,TIWR NEST .IF/PAP
ER SAVE USNO 8/91/29 !232
490 DATA BAS PLAYING MUSICAL
 NOTES 9/91/8,XBAS LOTTERIES
 AND PROFITS 9/91/10,ART OF
A/L MEMORY SAVING TIPS 9/91/
14 !246
500 DATA BAS A/L RAW DISK RE
PAIRS 9/91/17,MULTIPLE COLUM
NS FOR BOOT 9/91/25,FLAGS OF
 EUROPE 9/91/27 !155
510 DATA IL PASTOR FIDO REV
9/91/32,GEN-TRI THREE IN ONE
 9640 REV 9/91/33,HEAT INDEX
 GAUGES COMFORT USNO 9/91/36
 !166
520 DATA VEHICLE FUEL STATS/
```

(See Page 22)

MICROPENDIUM INDEX 1991B—

(Continued from Page 21)

TI BASE USNO 9/91/37,DISK DRIVE QUIET USNO 9/91/38,PARALLEL PRINTER HELP USNO 9/91/38 !210
530 DATA XBAS PROGRAM TIPS USNO 9/91/38,MUSICAL NOTES BAS 9/91/8,LOTTERIES AND PROFITS 9/91/10 !216
540 DATA DISK REPAIRS BAS A/L 9/91/17,BOOT MULTIPLE COLUMNS 9/91/25,9640 GEN-TRI THREE IN ONE REV 9/91/33 !194
550 DATA PRINTER HELP USNO 9/91/38,BAS READ FLIGHT SCHEDULES 10/91/7,XBAS LOTTO AND PARI-MUTUAL 10/91/12 !012
560 DATA TEXT FILES CONVERTING PC-TI 10/91/17,PC-TI TEXT CONVERSION 10/91/17,ART OF A/L USEFUL SUBROUTINES 10/91/18 !024
570 DATA MIDI-MASTER 99 REV 10/91/28,WALLSTREET ANALYST-ADVISOR REV 10/91/30,99 COMPUTER REPAIR REV 10/91/31 !255
580 DATA 4/WHEELIN' GAME REV 10/91/32,PRINTING DOUBLE COLUMNS USNO 10/91/36,MULTIPLAN SORTS USNO 10/91/36,EQUATION CHALLENGE USNO 10/91/37 !028
590 DATA TI-BASE REPEAT CHARACTER USNO 10/91/38,XB LOCKUPS USNO 10/91/38,FLIGHT SCHEDULES BAS 10/91/7,LOTTO AND PARI-MUTUAL XBAS 10/91/12 !141
600 DATA REPAIR COMPUTER 99 REV 10/91/31,DOUBLE COLUMN PRINTING USNO 10/91/36,BAS STRING FUNCTIONS 11/91/9 !194
610 DATA XBAS PRETTY PRINTED PROGRAMS 11/91/11,BAS A/L FILE HEADERS ON DISK 11/91/14,ART OF A/L INS AND OUTS 11/91/18 !182
620 DATA CHICAGO TI FAIR REPORT 11/91/26,PC SCAN AND DIGITIZE 11/91/30,SCUD BUSTER GAME REV 11/91/32 !080
630 DATA CODE BREAKERS GAME REV 11/91/32,GENBENCH SHELL FOR 9640 REV 11/91/34,THUMBNAILS FOR 9640 REV 11/91/35 !110
640 DATA TI-BASE FOR PAYMENTS USNO 11/91/37,COMPUTER MUSIC USNO 11/91/38,STRING FUNCTIONS BAS 11/91/9 !181
650 DATA PRETTY PRINTED PROGRAMS XBAS 11/91/11,FILE HEADERS ON DISK BAS A/L 11/91/14,SCAN/DIGITIZE PC/TI 11/91/30 !248
660 DATA BAS PRINT A CALENDAR 12/91/8,XBAS GET IT ON VIDEO 12/91/10,ART OF A/L WHY A LOADER? 12/91/14 !014
670 DATA EUNICE SPOONER/KIDS 12/91/19,SCANNERS AND PROGRAMS 12/91/26,CHARACTER RECOGNITION 12/91/28 !232
680 DATA CELLULAR AUTOMATA 12/91/29,MDOS TETRIS GAME REV 12/91/32,GRAPHICS AND BONUS DISK REV 12/91/33 !075
690 DATA BARRICADE FOR ABASIC GA REV 12/91/33,ARTIST FONT MAKER REV 12/91/34,SOUND F/X REV 12/91/35 !009
700 DATA HFDC/LARGER DRIVES USNO 12/91/36,MULTIPLE INDICES TI-BASE USNO 12/91/36,TURN ON PRINTER REMINDER USNO 12/91/37 !071
710 DATA CENTER DOUBLE-WIDTH TEXT USNO 12/91/37,DISK TUTOR MODS USNO 12/91/37,CHANGING COLORS IN E/A USNO 12/91/38 !174
720 DATA PARALLEL CABLE USNO 12/91/38,CALENDAR PRINTING BAS 12/91/8,VIDEO TITLE XBAS 12/91/10 !211
730 DATA LOADERS ART OF A/L 12/91/14,KIDS/EUNICE SPOONER 12/91/19,TETRIS MDOS GAME REV 12/91/32 !042
740 DATA ABASIC BARRICADE GAME REV 12/91/33,PRINTER TURN ON REMINDER 12/91/37 !198

READER TO READER

☐ C. LaPointe, 4530 Treeline Dr., Pensacola, FL 32504, writes:

I have Console Writer from Navarone, Gemini 10X printer. Wonder if using 99/4A there is any way to use condensed print, margins and tabs with this.

☐ M.G. Poskitt, 38 Fambridge Dr., Wickford, Essex, SS12 9HR, England, writes:

I own a P-GRAM Plus card but do not possess the programming skills required to update/improve dumped cartridges. Does anyone know of any useful updates?

I continue to have problems with my Quest RAMdisk which I purchased from an Australian user group following a review in MICROpendium in May 1990. The board has been tested and appears free of fault. However, with various 32K chips purchased, the TI has trouble accessing the memory. Can anyone with a functioning Quest please let me know exactly which chips they bought and where I might purchase them.

☐ Jim W. Krych, USCG MALLET (WLIC-75304). P.O. Box 2647, Corpus Christi, TX 78403-2647, writes:

I have heard so very little about the 99/8 computer, other than what I have gotten from this magazine. I would like to know if any 99/8 owners would be able to share the information on the machine. This would include docs, hardware specs, etc. Also, does TI still maintain the schematics and code for the Armadillo (99/8) computer? And if so would they be willing to release it to the public domain? Any information would be greatly appreciated as I am interested in TI history.

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

LinEditor

• A unique utility for the TI, 9640

By BILL GASKILL

LinEditor is a 100 percent assembly language coded utility designed for TI-Writer type word processing files, plain vanilla text files and source code programming files of the type produced by Editor/Assembler, the TI-Base command file editor, PrEditor and the like. Asgard Software markets it.

The program was created by Edwin C. Hall, a name that I've not heard mentioned in the TI community before. Mr. Hall is obviously an accomplished programmer, with a pretty clear view of what 99ers do most with their computers — word process. If you use BA-Writer, Funnelweb, Form Shop, TI-Writer or any other DV/80 based text program, you will want this program in your library. This unique tool expands your text file handling capability beyond anything else currently available.

THE PACKAGE: LinEditor retails for $14.95 plus $3 S/H. The package comes in a clear plastic seal-a-meal type envelope with a SS/SD diskette and an 11-page booklet style manual. The program is an E/A option 5 program image file that can be loaded through Extended BASIC, Editor/Assembler Option 5, TI-Writer Option 3 or the Geneve's M-DOS EXEC loader. A 32K memory expansion and at least one SS/SD disk drive are required for use on the 99/4A and the disk drive for use on the Geneve. No dedicated printing facilities have been programmed into LinEditor so a printer is not required. The program *can* print files by simply using PIO or RS232 etc. as the output file name. So printing capability exists if you want to use it.

Hardware Compatibility: LinEditor is device compatible with Myarc's HFDC, the Horizon RAMdisk, the TI floppy disk controller and I suspect almost any other piece of hardware you can throw at it, although I did not test it on anything else. The manual states that it is additionally compatible with most RAMdisks, hard drives and the Geneve. It supports path names up to 40 characters and can catalog any disk using path names that long, something I dearly wish Funnelweb could do. Again, it also supports output to a printer or modem by naming the output file as PIO, RS232 etc.

Review

Report Card

Performance	**B**
Ease of Use	**B**
Documentation	**C**
Value	**A**
Final Grade	**B**

Cost: $14.95 plus $3 S&H
Manufacturer: Asgard Software, P.O. Box 10306, Rockville, MD 20849
Requirements: Either TI with Extended BASIC, Editor/Assembler or TI-Writer and 32K memory, or Geneve 9640; one disk drive; printer optional.

Overview: LinEditor is a true text editor in that it does not offer the bells and whistles of a full-featured word processor such as block operations, paragraph reformatting, font control and so on. But that is not why it was created. It was created to fill a void in the word processing arena for handling large text files. Unlike any other text editor available to the TI community, LinEditor can capably handle *massive* text files as large as your available storage media.

When loaded the program displays a three-sectioned, white on blue, 40-column screen that takes on a spreadsheet metaphor. The left side of the screen displays an under-used line pointer window, the upper right displays the active text file in a left-right and up-down scrolling text window and the lower quarter or so of the screen displays the command line and editing work area. The screen is logically laid out and easy to understand, especially to those familiar with Multiplan. But even if those unfamiliar with it will have no trouble working the screen. As I said, it uses a spreadsheet-like metaphor to interact with the user so it looks and acts like a spreadsheet. Commands are entered in a command line at the bottom of the screen and characters or lines are deleted, edited, inserted or erased in the two screen lines above the command line. The cursor never appears in the scrolling text window. Instead, you move the file around until the stationary line pointer points at the line you wish to work on. Once the desired line is active (being pointed to) you simply type in a command like D to delete, E to Edit, I to Insert et cetera and the action takes place once you press Enter. The program then updates the screen's text file window to reflect the changes, except in the case of line inserts.

Performance: LinEditor works quickly and without a major flaw, though you'll need to read the manual, such as it is, and pay close attention to the sequence of the commands required to perform any desired operation. There is nothing mystical about any LinEditor operation but if you fail to follow the steps in sequence you may end up with an output file that contains fewer lines of text than you had intended. In some cases you might even end up with an empty file if you issue the wrong command or fail to issue a required command. You *never* run any danger of damaging or destroying your source file, no matter what commands you issue or do not issue, because you are always working with a copy of your original document.

Program error trapping is superior! I could not make it lock up or fail in any way, despite my most outrageous attempts. You can't even accidentally quit the program with a careless Fctn = key press. Mr. Hall really had the novice in mind when he designed protections against disaster into the program.

You can perform input/output operations such as loading text into the buffer as quickly as your storage media will allow, meaning the text buffer fills faster when the input file is stored on a RAMdisk than when the input file is being read from a floppy disk. Internal operations such as deleting lines, inserting blank lines and the like are performed at assembly language speed similar to TI-Writer and its clones. LinEditor does have some quirks though.

First, unlike TI-Writer, which displays line numbers at the left edge of the screen for location reference in a file, LinEditor displays only a series of useless vertical

(See Page 24)

LINEDITOR—

(Continued from Page 23)

bars that appear to serve only as a border protecting nothing to the left of the border. Line numbers would indeed be useful for determining where an end-of-page marker should be placed, etc.

Second, I could find no way to insert a carriage return into a file, despite the fact that the large CHARA3 file is supposed to provide control over a 256-element character set. Form feeds, line feeds, escape sequences and such are all there, but no carriage returns. Seems odd? Normally, Enter or Control M is used to issue an ASCII 13 or Control 8 if you're using TI-Writer. But none of these work in LinEditor.

Third, when you insert a blank line the text window never shows the insertion, meaning the existing text never moves downward to make room for the blank line. When you delete a line, the text window does perform a screen refresh where the text moves up, but insertions apparently always take place on the line just above the pointer (which is at the top of the screen) because the blank line is always there when you move the text file, but you never see its impact on the rest of the file. That takes some getting used to.

Fourth, when in the Insert mode, which you use to create a file from scratch, you *never* see what you've just typed once you press Enter to accept the input. It's always off the top of the screen somewhere! This will really take some getting used to if you intend to use LinEditor regularly for file creation.

Fifth, the program offers no "bailout" feature, meaning your output file is written to disk or printed (depending upon where you sent it when you named the output device) whether you want it to or not. You get no second chances and LinEditor takes no prisoners in this area. Your only option is to turn the computer off.

Ease of Use: When ready to use LinEditor you type in O to open an input and then an output file, specifying paths and filenames for both files at the appropriate prompts. Once you press Enter after naming the output file, LinEditor will load as much as four to five pages of text from the input file into its buffer. It then lets you manipulate the text in the buffer using the simple command set C to Close the input file, D to Delete a line of text, E to Edit a line of text, I to Insert a blank line, O to Open the input/output files, Q to Quit the program, R to Read in more text from the input file, T to Terminate (close) the output file or W to Write the contents of the buffer to the output file. One of these single-letter commands accomplishes everything you need, if you forget what they are just press Fctn 7 and they're displayed.

The commands themselves are easy to remember and they have fairly logical names (except for Terminate the output file). But the sequence in which one must use commands I at first found difficult to remember. This is where the instruction manual comes in, but the docs for LinEditor leave much to be desired.

Documentation: As with too many TI-99 products, the documentation is the weakest part of the package. It contains mistakes in grammar, typing errors and a ton of superfluous verbiage that makes reading it cumbersome to say the least. The docs are organized into confusingly written narratives rather than more useful outlines listing the required steps to carry out for specific operations. The narratives are poorly thought out and they do a wonderful job of hiding the meat of the matter behind useless statements. Nowhere does the manual give the user a 1, 2, 3 step by step guide to the operations the program can perform. Instead, it leaves the reader to search for the information in a jumble of garbage under nebulous headings like Special Keys and Advanced uses.

The Special Keys section tells the user absolutely nothing useful, but does refer the reader to Appendix II and III. Appendix II is a set of four keyboard maps that even I could figure out after I read the opening paragraph to them five or six times. Appendix III is totally useless, consisting of a single paragraph of dribble that could and should have been left out of the booklet.

The so-called "Advanced Uses" are normal operations, in my opinion, ones for which most people will want the program. Why hide them behind a scary term like "advanced"? That makes the program sound like Advanced Diagnostics or something which you really need experience to use. LinEditor does not require that one be a rocket scientist use it! There is nothing "advanced" about it. The program is a useful utility that sorely needs a rewritten set of instructions with its next release (I hope there will be one).

Conclusion: The more I read the manual and the more I looked over the package as a whole the more I developed a perception that it was "rushed out the door" in order to be available for Fest-West or some other pressing event. This idea was strengthened by the fact that LinEditor's author is Edwin Hall, but the disk label for the program lists "Edward" Hall as the author. Further, the copy I tested, purchased at Fest West in Phoenix, had no XB Loader. The manual claimed that the program loads out of Extended BASIC, and it does once you have a Barry Boone type XBOPT5 loader to do it. Asgard immediately sent a replacement when I questioned the oversight, and their support response was timely and courteous, but the loader should have been there in the first place. I was told that only the first few copies released had this problem.

Despite the "rushed out the door" appearance and the "quirks" mentioned earlier, LinEditor deserves a favorable recommendation in my opinion. Its support of path names and large storage devices make it unusually flexible, while the ability to deal with large text files make it uniquely valuable to anyone who has large text files which need attention. If I had it to do over again, would I buy LinEditor? The answer is *absolutely*! In a heartbeat. It is a niche product that just happens to be in its infancy. If we are smart enough to support it now, the program just might make it to adulthood.

OS/99 Version 3

A great deal for owners of GRAM devices

By JOHN KOLOEN

OS/99 is designed to replace TI's Editor/Assembler cartridge and provides the user with direct access to E/A functions, including the editor, through a series of mnemonic commands. The program, which is freeware, was written by Guy Neubert and others. Use of OS/99 requires a GRAM device, such as GRAM Kracker, GRAMulator, P-GRAM, etc.

Performance: I'm not an assembly programmer, but I can see where OS/99 would be useful to programmers. Through the use of mnemonics, rather than menu selections, a programmer can jump from function to function without having to wade through a series of menu options. But there's more: For non-programmers, it provides the same flexibility when loading and running assembly programs.

The program loads through a GRAM device, so I won't bother going through the routine. It's a simple matter of loading the program into GRAM from a floppy disk. After that, you select it from your menu screen and almost instantly you are presented with a nearly blank screen that contains a command line at the bottom. Displayed near the bottom of the screen are the default drive and default printer name. Each may be changed easily. For example, to change the default drive from DSK1 to DSK2, simply enter $D 2. However, to take effect, the write-protect function of the GRAM device must be off. The change to the default drive or printer remains permanent until changed again.

Here is a list of the commands available through the OS/99 command line:

HE—Displays a list of mnemonic commands

H—Displays a single screen help file

ED—Editor/Assembler editor

SP—Outputs display files to a printer

AS—Editor/Assembler assembler

Review

Report Card

Performance	**A**
Ease of Use	**A**
Documentation	**B**
Value	**A+**
Final Grade	**A**

Cost: Freeware (distribution charge only)
Requirements: TI99/4A, memory expansion, disk system, GRAM device

LO—Option 3 of E/A

RU—Option 4 of E/A

EX—Option 5 of E/A

GX—Used to go to another cartridge ("X" is the GROM number)

SA—Save utility emulator

IL/CL—Initialized low memory/clear low memory

$D—Changes default drive

$P—Changes the printer's name

$B—Makes OS/99 autoboot on power-up

$U—Makes OS/99 selectable from the power-up menu

Also included with OS/99 are several useful "extensions" which will give any user a good idea of the potential of the program's command line.

CP—File copy utility

LD—Directory listing utility

RM—Remove file (delete) utility

The extensions are extremely useful and immediately made an impact on me. While in OS/99, using the three extensions that come with OS/99, I can run a directory as well as copy files and entire disks without having to load a disk manager or other program. I suspect the variety of extensions possible is limited only by the imagination and programming skills of end users.

In addition, OS/99 also allows you to load many E/A programs from the command line. If the program you want to load is on the default drive, all you need to enter is the program name, such as DM for DM1000. BA-Writer loads with the simple command BA (though you could give it any name, such as BA-WRITER). Archiver loads with ARC1.

OS/99 loads into GROM 7, G>E000 through G>FFFF, leaving the remainder of the system available for other programs. Any cartridge that doesn't use GROM 7 can be loaded into the GRAM device and reside concurrently with OS/99. OS/99 also can be entered from BASIC or Extended BASIC with the command CALL OS.

The authors of OS/99 have deliberately avoided loading OS/99 with numerous routines and functions, preferring to let other users develop it through extensions.

Ease of Use: OS/99 is easy to load. Anyone familiar with TI's Editor/Assembler will have no trouble learning how to use OS/99. What they do with it beyond the obvious is up to them.

Documentation: The docs consist of about nine pages in several D/V80 files that may be printed using TI-Writer and other word processors. The docs provide adequate detail regarding use of the various commands.

Value: This is one of few programs I've given an A+ to for value. It's very nearly free. The only charge is for the trouble Tex-Comp and others take to make and distribute copies. Unfortunately, the program is so new that you may have difficulty locating it. Tex-Comp is expected to distribute it, as are some user groups. But the program was not available from these sources as of May 12. I'm not sure whether it will be available on any bulletin boards, though I suspect it will at some point.

For anyone with a GRAM device, you can't lose with OS/99.

MICRO-REVIEWS

Easy Data, Grades, LPJ Art 5 & 11, Extended BASIC Utilities Vol. 1 & 2 and Random Number Utilities

By STAN KRAJEWSKI

This month I am happy to include a program from France. This shows that support for the TI is still going strong in other countries, and not just limited to the United States.

I was having trouble with my Dragonslayer Spell Checker and was wondering what was going on. I had used it on my CorComp Micro Expansion System before, but this time it wasn't working. When I would try to load the D/V80 file I wanted to check, the system would give me a screen with distorted colors and lines, and it just plain malfunctioned. I had been preparing my review on the Geneve with MY-Word, then saving it to disk for use with the TI and the Dragonslayer program. Now I see it does not work on the Geneve. Evidently the Dragonslayer Auto Spell-Check does not work with MY-Word either. If I reloaded the D/V 80 file, then resaved it with Funnelweb, it would then work properly. I do not have an 80-column Funnelweb program yet. If I had it, I would not have noticed the problem.

Ratings for the software reviewed in this column are based on the Star system that follows.

★ Leave it alone, back to the drawing board.

★★ Needs improvements, but workable.

★★★ A good program, worth trying.

★★★★ Send your money and buy it.

★★★ EASY DATA

This program on SS/SD disk was developed for unskilled Extended BASIC programmers. It is good to see Harrison Software responding to the needs of one of their customers, who needed to make a database to keep track of videocassettes. These programs serve more than one purpose. They give you a sort program for your database that sorts at a high speed and enables you to customize it to your needs.

System requirements are Geneve 9640 or TI99/4A, 32K memory expansion, Extended BASIC and disk drive.

You can create a program using these files without any knowledge of assembly language. The files have assembly language routines buried within Extended BASIC programs. BIGDEMO is a demo file that contains 55 records and six fields and can sort this in three seconds. You can even take this program and build a database around it. One of the nice features is that you can input information to the program by just adding DATA statements without having to change other parts of the program. This program knows that any DATA statements after line 1000 are for the purpose of sorting. You may have a primary sort or a primary and secondary. For example, primary can be by last name, and secondary can sort by state at the same time.

A program called Skeleton is provided on the disk and will run as is, but also serves as a framework on which you can build programs of your own. You may then take the menu driver and merge in your own data and other options. Two D/V 80 files are included on disk. They describe everything you need to know so you can modify it to suit your needs. Also included are other files to enhance the memory, and give you ideas on adding these features to your program.

Easy Data is available from Harrison Software, 5705 40th Place, Hyattsville, MD 20781. Price is $6 including S&H.

★★★ GRADES

My sister happens to be a schoolteacher. (Hi, Lila.) She and other teachers now have a program which can take test grades of a class, then turn them into the pupils' averages for the year. You can enter up to eight tests for each pupil.

TIs are donated by the Chicago TI Users Group through a program called "Computers to Teachers," and I'm sure they will have use for this program.

System requirements are Geneve 9640 or TI99/4A, 32K, Extended BASIC and disk drive. A printer is optional. This SS/SD disk has a LOAD program installed on it, so all you do is boot Extended BASIC.

The menu includes:

1. Enter Pupils' Names — This does what it says, plus it alphabetizes the names.

2. Enter Grades — You can enter up to eight scores. If a student misses a test, entering 0 will enter as a null value so as not to reduce the final average of the student for missing a test.

3. Print Summary — This prints the students' names, each test score, average grade and a letter grade. You have the option of printing to the screen or printer. You can print pupils' names or names and test grades. Also, all grades or just the finals can be printed.

4. Open "Names" File — This is used at the start of a new school year to enter new students' names.

I feel this program saves a lot of time over figuring averages with a calculator. A teacher, or anyone who keeps scores and averages, would appreciate this program.

The author would like this released as Fairware. A donation of $5 is needed for cost of postage and handling. Send to: Harold W. Evans, M.D., 293 Circle Hills Dr., Grand Forks, ND 58201.

★★★ LPJ ART 5 & 11

Cartoon characters plus other art never looked better! These two art programs on SS/SD disks looked as if they were scanned; instead, they were all hand drawn.

System requirements are Geneve 9640 or TI99/4A, 32K, Extended BASIC, TI-Artist, clone or a MAX/RLE program.

LPJ ART 5 has 30 Large Animal Instances. At least most of them are. There are a giraffe, camel, rhino, hippo, lion and other large animals. They start large and go down to a dog, bird, cat and a bee. They are in D/V 80 TI-Artist Instance format and can also be used with Artist Cataloger,

(See Page 27)

MICRO-REVIEWS—

(Continued from Page 26)

Giant Artist Posters and other art programs. They look realistic and are a welcome addition to anyone's files.

LPJ ART II has 12 pictures that can be viewed by MAX/RLE and printed or saved into Dis/Fix, Dis/Var, or Graphx Format. These look so good that they look like they were taken from Saturday morning cartoons! Recognizable characters are the Tazmanian couple, Woody Woodpecker, the Easter Bunny at work and more. These pictures, printed on TI-Artist Plus, print to half-page size, but can be shrunken to a smaller size for your stationary, etc.

Leland Piper has completed 19 packages of pictures and packages of instances to date. There are 12 pictures in each package and 30 to 50 instances. He has been creating these for the TI for the past eight years. For information, ordering, help or just questions he may be reached at (417) 732-7777 or write to Leland Piper, 904A N. Main, Republic, MO 65738. Each disk can be obtained for $5 plus $1 S&H.

★ ★ ★

EXTENDED BASIC UTILITIES VOLUMES 1 & 2 and RANDOM NUMBER UTILITIES

Extended BASIC Utilities Volume 1 contains six assembly language utilities, and Volume 2 contains four. System requirements for these utilities on disk are Geneve 9640 or TI99/4A, 32K, disk drive and Extended BASIC. TI-Writer or Editor/Assembler is not necessary to print the docs and the source files. Two Extended BASIC programs are included for this purpose.

These utilities come on three SS/SD disks. All three disks contain utilities for use by Extended BASIC programmers. By studying the source code, you will gain insight into the programs, as all statements have comments.

The disks contain all the utilities you need to merge the assembly routines into your Extended BASIC programs. The six utilities in volume 1 include:

DEVICE — Places device name in string variable.

NUMACC — Special ACCEPT AT for numeric variables.

STRACC — Special ACCEPT AT for string variables.

NOTEASG — Assigns note values to an array variable.

TRACK4 — replaces DSK1 in Extended BASIC programs. This is interesting since it can modify an XB program to operate at any disk it was loaded from.

TRUINT — Gives the integer part of a numeric variable.

Volume 2 contains utilities that work with DATA statements:

MENDRV — A fast menu driver that creates menus from data statements.

ASSIGN — Assigns string variable array.

STRDATA — Assigns string array from XB data.

NUMDATA — Assigns numeric array from XB data. Demo programs are on disk for every utility.

Also ALSAVE and ALLOADM are included on each disk to save time in incorporating these assembly routines into your Extended BASIC program. With each disk three pages of documentation explains the routines in conjunction with the demos.

RANDUTAL is the last disk in the package. This disk concentrates on random seeds. This set of utilities runs twice as fast as the random seeds created with RND in Extended BASIC. Again, these utilities come with demos and mostly deal with random numbers that can be used for seeds for card games. The programs are:

DEALER — can handle up to 54 cards.

MULDEK — Limits are 54 cards and 5 decks or 48 cards by 3 decks.

XBARRAY — Reports random numbers within a chosen range into any array variable.

XBRAND — Reports a random number into a numeric variable. These seeds give the capability to allow programs to run as DSK1.LOAD to seed the random number process properly, so that high quality "random" numbers sequences are generated. These routines also have annotated source code so you may understand their workings.

These disks were created by Harrison Software and are released as public domain for any TI programmer to use for any purpose he or she chooses.

This package is available from a number of sources such as Lima Users' Group, Tigercub's TI-PD service and possibly on GEnie. If you cannot get these from these sources, you can order them from Harrison Software, 5705 40th Place, Hyattsville, MD 20781 at $3 per disk including S&H.

If you would like your software or hardware reviewed in this column, you may send it to Stan Krajewski, Route 6 Box 568-15, Live Oak, FL 32060. If you would like it returned, please include postage. If you need to discuss something, for any reason, you may call me at (904) 364-7897 E.S.T.

User Notes

Bytes in memory, and substitutes for logical operators

This comes from Chuck McConnell, of Berwyn, Illinois. He writes:

When I was writing programs in TI-BASIC with the original 16K of memory, I always wanted to know how many bytes of memory I had left. One only has 12.5K of RAM. To check this, type in the following program:

```
1 A=A+8
2 GOSUB 1
```

Then run it. In a few seconds you will see MEMORY FULL in 1. Then type in PRINT A (with no line number) and press Enter. The number of bytes remaining in memory will appear. In the abov e case, 12,500. If you have your program already in memory, then follow the above procedure to find how many many bytes you have left to work with. This assumes that your program's line numbers start with a line numbered higher than two and the letter "A" is not used as a variable in your program. If "A" is used, use a different letter in this routine.

(See Page 28)

User Notes

(Continued from Page 27)

Logical operators and AND and OR can only be used in Extended BASIC. For example:

100 IF (A<2 OR B>1) AND (C<6 OR D>3 THEN 300

We could substitute an asterisk and plus sign for AND and OR, respectively to get:

100 IF (((A<2)+(B>1))*((C<6)+(D>3))) THEN 300

This expression will work for TI-BASIC and Extended BASIC.

Tips on using Funnelweb 4.40

This comes from Phil Martin of Keizer, Oregon. He writes:

Recently, I recieved a letter from a TIer who had obtained a copy of Funnelweb 4.40. He mentioned being unable to exit from Quick Directory, among other things. Since then I ordered a copy of the program and found that most of the documentation was not included. So I set it aside until I had time to explore. What follows is the result of that exploration:

- To exit from the Quick Directory feature the key combination is CTRL +/=.
- Funnelweb 4.40 no longer looks for the CHARA1/CHARA2 character filenames but rather C1/C2. If your old version has custom character files you would prefer, simply copy them to your 4.40 disk and rename them. Of course, you're using a backup copy of the Funnelweb disk to do this.
- With all versions prior to 4.40, the exit from Funnelweb was from the assembler side by pressing 8 to reset and then 8 again to quit. With 4.40 the new exit is accessed from either menu by pressing Escape (either CTRL C or FCTN 9).

If you press Enter with "N" you are given options to change. First, the character set name for the Text Edit side (1), next for the Program Edit side (2), then the printer name and, finally, the drive location for the TI-Writer and Editor/Assembler files. These are shown as 11 under the heading "From Drive TW/EA." This is read not as "eleven" but as "one-one." After this you are returned to the menu system.

- There is also a chnage in the CONFIGURE program which affects the operation of the loading procedure. This is under the LOADING heading within CONFIGURE. After loading CONFIGURE you must first load SYSCON, then select Edit from the SYSCON menu. This presents you with the menu which includes the LOADING option. Within this selection you will find the option IMMEDIATE. Pressing "I" gives you a choice of three options — FW, UL or DR. FW is, of course, Funnelweb, UL is the User List and DR is Diskreview. If you want to get to Funnelweb directly you must have this set to FW. The IMMEDIATE feature replaces the UL IMMEDIATE function found in previous versions of the CONFIGURE program.

For more information about configuring Funnelweb see Tutorial: Configuring Funnelweb by Jim Swedlow (October 1988, MICROpendium).

Diskreview is new to the system. It has some of the features offered by John Birdwell's DISKU. These include some of the disk-handling functions and a sector editor, as well as the ability to load and run many programs from the directory it generates. These include Extended BASIC programs *if Funnelweb was loaded from the XB cartridges*. This is true even if using the P-GRAM+ card. Note the key definition diagram to the left of each screen. These definitions change as you progress through the various functions of the program. As an example: From the first screen you can select a drive to scan (1-9), select a different color scheme (0) or enter the disk/RAMdisk utilities sections (D). From this same area you can return to the Editor/Assembler side of the Funnelweb system by pressing CTRL+/=. By pressing "F" you are presented with both central menus from Funnelweb, which allows you to load any of the selections from either.

I hope this helps those who received a copy of 4.40 without docs. Meanwhile, if there is somebody out there who has the documentation disk for 4.40, it might help if they wrote to MICROpendium to let the rest of us know.

Already an update of Window Maker

This comes from Arnie Stewart, of New Smyrna Beach, Florida. He writes:

The Window Maker program (March 1992, User Notes) caught my eye because of the myriad IFs, Xs, ANDs and parentheses in lines 170-210. The author, W.L. Shepard, has some pretty neat algorithms in lines 220-260, the kind I wish I could figure out when programming. However, I realized almost at once that all those IF statements could be handled by one MAX statement in a FOR-NEXT loop. From that point, I couldn't help myself. I had to go to the keyboard to test my theory.

Enclosed is a listing of the program as I have modified it. It now runs only 11 lines instead of 21, and does the same thing.

The secret starts in line 120 of the modified program, with the statement X(Z)=LEN(A$(Z)). This does the same thing as line 160 in the original program, while eliminating a program line. Line 150 finds the length of the longest string entered, and it is done in one line, eliminating old lines 170-210. The additional benefit of this approach is that we are no longer limited to five input lines, as in the original program. The only limit now is the built-in limitation in TI Extended BASIC of ten subscripts to a variable — variables A$() and X(). Of course, you could always DIMension those variables to any size desired, but if you entered more than 22 lines, you couldn't draw the box around them — and that was the purpose of the program.

```
100 ! WINDOW MAKER BY W.L. S
HEPARD - MODIFIED BY A.L. ST
EWART 3/28/92 !078
110 N=5 :: Z=1 :: CALL SCREE
N(5):: FOR I=0 TO 13 :: CALL
 COLOR(I,15,5):: NEXT I :: D
ISPLAY AT(2,7)ERASE ALL:"ENT
ER THE TEXT" !049
120 ACCEPT AT(N,1):A$(Z):: X
(Z)=LEN(A$(Z)):: DISPLAY AT(
24,4)BEEP:"ANOTHER LINE? (Y/
N)" !103
130 CALL KEY(3,K,S):: IF K=8
9 THEN Z=Z+1 :: DISPLAY AT(2
4,1):"" :: N=N+1 :: GOTO 120
 !159
140 IF K=78 THEN 150 ELSE 13
0 !161
150 A=0 :: FOR I=1 TO Z :: A
```

(See Page 29)

User Notes

(Continued from Page 28)

```
=MAX(A,X(I)):: NEXT I !066
160 CALL CLEAR :: B=(16-(A/2
)-1):: C=14-(A/2):: D=16+(A/
2):: CALL CHAR(128,"00000000
00FFFFFF"):: CALL HCHAR(3,B,
128,A+2)!003
170 CALL CHAR(129,"C0C0C0C0C
0C0C0C0"):: CALL VCHAR(4,B,1
29,2+Z):: CALL CHAR(130,"FFF
FFF0000000000"):: CALL CHAR(
131,"0303030303030303")!122
180 CALL HCHAR(6+Z,B,130,A+2
):: CALL VCHAR(4,D,131,2+Z)!
171
190 J=5 :: FOR I=1 TO Z :: C
=14-X(I)/2 :: DISPLAY AT(J,C
)SIZE(X(I)):A$(I):: J=J+1 ::
 NEXT I !072
200 INPUT "PRESS ENTER FOR A
NOTHER->":E$ :: GOTO 110 !GO
TO THE REST OF XB PROGRAM FR
OM HERE !247
```

TI-Writer Formatter and Funnelweb

This item, by John Briscoe, appeared in a number of newsletters, including the Spirit of 99 and TIsHUG News Digest.

This one is a real time saver for people who use the TI-Writer Formatter in Funnelweb. You can do a disk directory while in the Editor and mark a file so that you do not have to tyupe in the DSKn.FILENAME. This is a big help if you cannot remember the filename. If you do a disk directory while in the Formatter, however, you apparently cannot mark the file. So, if you want to mark a file for use as the Input Filename by the Formatter, you must exit the Formatter, enter the Editor, do a disk directory, mark the file, exit the Editor and reenter the Formatter. This is very clumsy and slow if you are not using a RAMdisk.

A better way is to do a disk directory (FCTN 7) while still in the Formatter. Arrow down to the file you want to output. Press the space bar — this places and invisible mark on the file. Press CTRL = to return to the Formatter, then press FCTN D to place the new filename in the Formatter Input Filename field. This saves the see-saw time for repeatedly loading the Formatter and Editor just to mark files.

To make this work, however, you must have the Funnelweb files QD and QF on your Funnelweb work disk.

Archiving Plato backups

This item, by Ken Woodcock, appeared in the newsletter of the Hampton Roads TI User Group of Norfolk, Virginia.

It is a good idea to keep backup disks of all your valued software. In fact, the old proverb "A fool and his gold are soon parted" could be rephrased to "A fool and his unbackedup software are soon parted."

Most software is easily backed-up, and by using Barry Boone's Archiver program efficient use of disk space can be achieved. However, some commercial software employs copy protection, which inhibits the making of backups. This is true of the Plato courseware disks. In fact, they use two levels of copy protection. But with a little information and time, that can be overcome.

The first copy protection method used is the Proprietary Disk flag, which caused TI's disk manager module to abort a disk copy attempt. This is no longer an obstacle since most modern disk manager programs disregard this flag. In fact, the earlier versions of DM-1000 had an option to remove disk protection. So, a duplicate copy of your Plato disks can easily be made with any sector copier.

However, if you want to reduce your backup disk storage space by utilizing Archiver and the compress mode, a little more work is required.

If you try to catalog your Plato disk you will see one file (named $$1) which is eight sectors long. You will also notice that all 360 sectors of the SSSD disk appear to be used (an obvious ploy!). The standard TI disk catalog uses sector 1 as an index to the File Descriptor Records (FDR) for each file in alphabetical order. The Plato disks use sector 359 (>167) for this purpose in an attempt to prevent accurate disk cataloging and file copying. Knowing this, it is relatively simple to copy sector 359 to sector 1. Once this is done, the disk may be cataloged in the usual manner. Most Plato disks are quite full but some use less than half the available space. Once this is done Archiver can be used to compress and combine the contents of each disk into a single file for backup purposes.

Here is the procedure for archiving a Plato disk using Disk Utilities by John Birdwell and Barry Boone's Archiver.

- Make a backup of the Plato disk using the disk utilities backup function on DISKU.
- Use the sector utilities to copy sector 167 to sector 1.
- Use Archiver to archive all files and compress.

To unarchive the file to make a working Plato disk you would reverse the above steps, with the addition of one step. Before un-arcing to a blank SSSD disk, sector 359 (>167) needs to be reserved for the use of the Plato program. Use Disk Utilities to mark that sector, then un-arc the file. Now copy sector 1 to sector 359 (>167) and you are done.

Procedure to un-arc a Plato disk.

- Mark sector >167 of a blank SSSD disk using Disk Utilities.
- Un-arc archived Plato file to disk using Archiver.
- Copy sector 1 to sector >167.

I have been able to store as many as 12 archived Plato disks on one DSDD disk, which I feel is worth the effort.

Kwikfont colors

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

Here is a tip on how to permanently change the color scheme of Wayne Stith's Kiwkfont program, which by default has black letters on a tan (what TI calls gray) background. Because Kwikfont uses a graphics display mode (the 28-column screen) you can change border, background and foreground colors like any XB program.

To do so, load Advanced Diagnostics and go to sector 49. At byte 188 you will see 1E. This is the code for a black on gray foreground and background color scheme. To change it to white on blue, replace 1E with F4. Next, more the cursor to byte 201. Here you will se a 0E, which is the color code for transparent on gray. This is the screen or border color coding, meaning the colors that appear at the extreme top and bottom edges of your monitor. Changing it to 44

(See Page 30)

User Notes

(Continued from Page 29

will make the borders the same blue that appears in the background color scheme. If you are using DISKU or other sector editor, go to sector >31, which is the hext equivalent of decimal 49 that Advanced Diagnostics uses. The byte numbers to be changed are the same, 188 and 201.

Other combinations may be put together by referring to the June 1985 issue of MCIROpendium, page 43, where Tim MacEachern shows how to modify TI-Writer colors. The same rules apply to Kwikfont, once you know where the bytes to be changed are located, and once you discover that Kwikfont offers one more color option than TI-Writer because it uses a graphics rather than text display mode.

Classified

SOFTWARE

TI-PD PUBLIC DOMAIN AND FAIRWARE

600 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 20-page catalog listing all programs and authors. Catalog also available on disk.

TIGERCUB SOFTWARE, 156 Collingwood Ave., Whitehall, OH 43213. v9/4

MIDI-Master music — Classical Potpourri — A collection of well-known classical pieces by various composers. Includes the famous pieces "Funeral March of a Marionette" by Gounod (AKA Alfred Hitchcock's Theme), Offenbach's "Barcarole," Ravel's "Pavane for a Dead Princess," and other familiar themes. All are supplied as SNF source files. Requires MIDI-Master, 32K, XB or E/A, and DS/SD disk drive. Bargain Priced at $10.00 from Harrison Software, 5705 40th Place, Hyattsville, MD 20781. 9/4

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.

MISCELLANEOUS

FOR SALE

Misc. TI equipment and books too numerous to itemize. Example of prices: TI99/4A P-Box, fully loaded, $150.00. Speech Synthesizer free with every order over $50.00. Call 813-681-8954 after 7 p.m. Eastern. v9n4

Complete TI99/4A system including: full PE-box (32K, RS232, 2 half-height drives), Amdek color monitor, 2 consoles, 35 modules including XB, E/A, Logo, Multiplan with manuals, much graphics and games software. $295 plus shipping. Call 1-503-231-0723 after 5 p.m. PDT. 9/4

ALL THIS FOR THE CC40

80 col printer	$80.00
80 col printer used	$50.00
CC40 Computers 6K	55.00
CC40 Computers 18K	80.00
Electrical Engineering package (cart)	20.00
Finance Packages (cart. & book)	10.00
Games #1	10.00
Hexbus Interface Manual	10.00
Mathematics Package (cart. & book)	20.00
Modem 300 baud	50.00
Pascal Package (cart & book)	50.00
Power Supply for CC40 and periphs.	15.00
Printer Plotter	50.00
Printer Plotter 4 color (pen set)	15.00
Program Cartridges (blanks) 8K RAM	25.00
RS232	50.00
RS232 with parallel printer cable	70.00
RS232 parallel printer cable	20.00
Statistics Package (cart & book)	20.00
Wafertape Drive User's Manual	10.00
Word Processor (cart & book)	20.00

....(also has modem program)

S&H 10% Minimum $3 214-821-9274

Jim Lesher, 722 Huntley, Dallas TX 75214

Buy and sell used computer stuff

National Used Software Club has buyers looking for TI products, as well as sellers. More than 100 TI items are listed for sale, including such hard to find items as Z80/80-column cards and GRAM Kracker. The membership fee is only $15/year and entitles you to buy or sell computer items via the NUS/HC database. You will also receive FREE our newsletter (mailed every eight weeks). **Our guarantee**: if you don't buy or sell anything during the term of your membership we will refund your annual fee.

For a free, no-obligation information packet, write to National Used Software/Hardware Club, P.O. Box 1343, Round Rock, TX 78680.

Buy and Sell On-line

NUS/HC members can access our database via our BBS (300-2400 baud). Non-members can go online and learn more about NUS/HC. Members can download/upload data. Join online if you like. Call 512-255-1557 (8N1) 6 p.m.-8 a.m. weekdays & Saturday; Saturdays 1 p.m. to midnight; Sunday all day and night.

ENORMOUS TI99/4A INVENTORY. CATALOGS $2.00. **BRAATZS COMPUTER SERVICES**, 719 E. BYRD ST., APPLETON, WI 54911. **1-414-731-3478.** 9/9

GAMES! EDUCATIONAL! HARDWARE!—TI99/4A
CALL OR WRITE FOR FREE CATALOG:
JOY ELECTRONICS, INC; P.O. BOX 542526
DALLAS, TEXAS 75354-2526
(800) 527-7438, OUTSIDE DALLAS AREA
(214) 243-5371, DALLAS AREA
9/6

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 ☐ ☐ (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