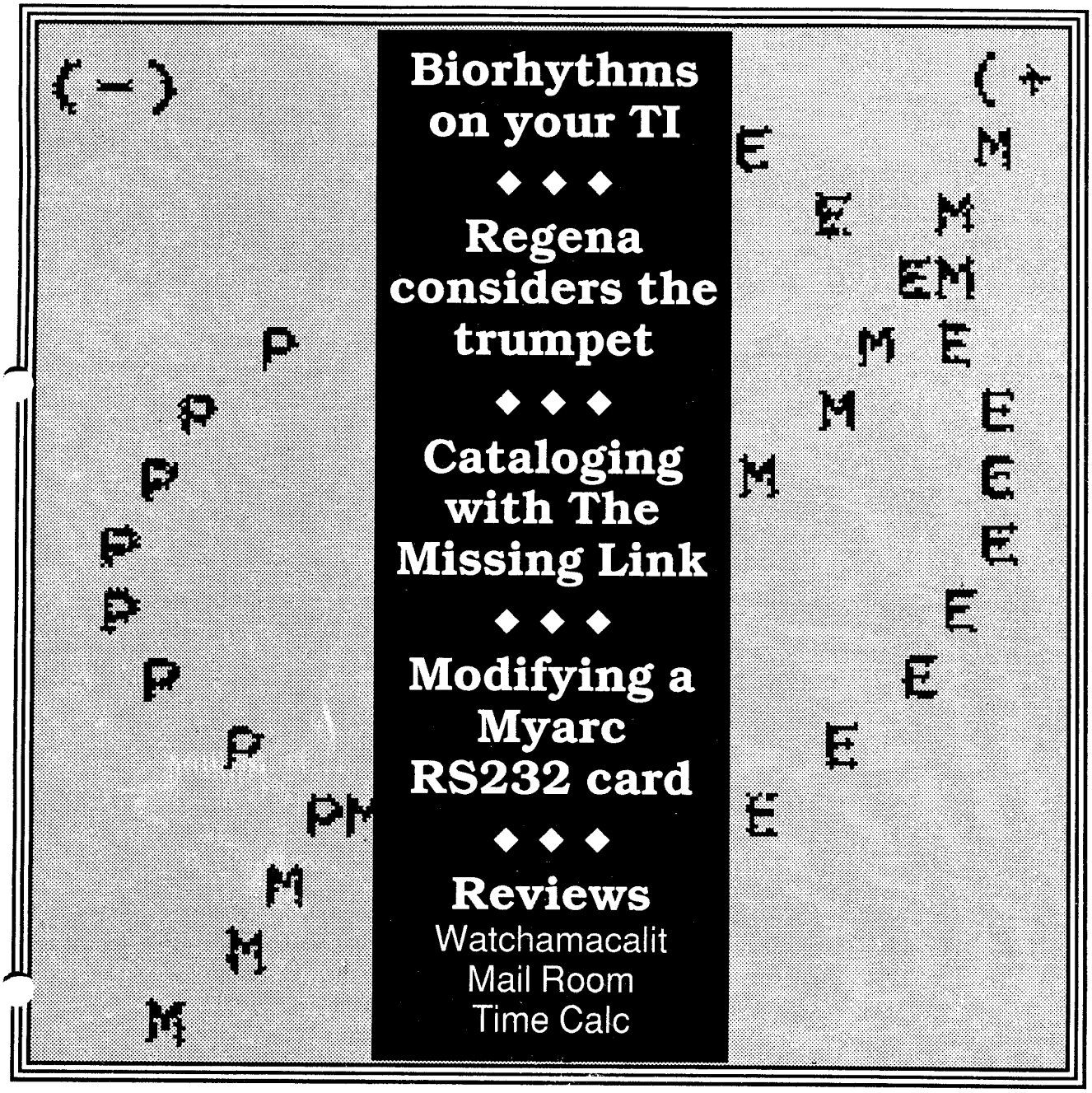
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 10 Number 7 August 1993 $3.50



Biorhythms on your TI

◆ ◆ ◆

Regena considers the trumpet

◆ ◆ ◆

Cataloging with The Missing Link

◆ ◆ ◆

Modifying a Myarc RS232 card

◆ ◆ ◆

Reviews

Watchamacalit
Mail Room
Time Calc

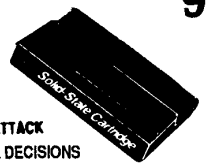
TEX-COMP

$$ DOLLAR DAYS

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

SOFTWARE SUPER SALE!

PHM 3027 ADDITION & SUBTRACTION I
PHM 3028 ADDITION & SUBTRACTION II
PHM 3093 MILLIKEN DIVISION
PHM 3051 NUMERATION II
PHM 3094 MILLIKEN INTEGERS
PHM 3118 MINUS MISSION
PHM 3029 MULTIPLICATION I
PHM 3002 EARLY LEARNING FUN
PHM 3003 BEGINNING GRAMMAR
PHM 3112 PARSEC
PHM 3007 HOUSEHOLD BUDGET MANAGEMENT
PHM 3044 PERS. REPORT GENERATOR**
PHM 3053 TI INVADERS
PHM 3194 JAWBREAKERS II
PHM 3016 TAX INVESTMENT RECORD KEEPING***
PHM 3022 PERSONAL REAL ESTATE
PHM 3023 HUNT THE WUMPUS
PHM 3031 THE ATTACK
PHM 3220 MICROSURGEON
PHM 3067 MUNCHMAN
PHM 3110 CHISHOLM TRAIL
PHM 3219 SUPER DEMON ATTACK
PHM 3006 HOME FINANCIAL DECISIONS
PHM 3035 TERMINAL EMULATOR II
PHM 3052 TOMBSTONE CITY
PHM 3109 LOGO II* → Plus $4 additional shipping & handling
PHM 3065 EDITOR ASSEMBLER
PHM 3010 PHYSICAL FITNESS
PHM 3146 MUNCHMOBILE



Special Purchase

DONKEY KONG featuring Mario $4.95 (regular $9.95)

ALSO INCLUDED ARE THE FOLLOWING ARARISOFT MODULES:
DEFENDER, PROTECTOR II, PICNIC PARANOIA

*ADD AN ADDITIONAL $4.00 EACH ON THESE THREE TITLES DUE TO WEIGHT
** FOR USE WITH PERSONAL RECORD KEEPING, *** DISK DRIVE REQUIRED
PERSONAL RECORD KEEPING AVAILABLE AT $14.95

AS LOW AS $1.95 EACH!!

WHEN BUYING TEN OR MORE
OR BUY ANY AMOUNT AT ONLY $2.95 EACH.

99¢ BOOK BLOWOUT!!!

PUBLISHER'S CLEARANCE ON BOOKS FOR THE 99/4A
BUY FIVE OR MORE BOOKS FOR ONLY 99 CENTS EACH + s&h.
BUY ANY AMOUNT LESS THAN 5 FOR ONLY $1.49 EACH + s&h.

GAMES TIs PLAY
THE ELEMENTARY TI
COMPUTER PLAYGROUND (TI)
INTRO TO ASSEMBLY LANGUAGE
PROGRAMS FOR THE TI COMPUTER
TI-99/4A GAME PROGRAMS
STARTER PACK I
STARTER PACK II
GAME WRITERS PACK I
GAME WRITERS PACK II

SPECIAL PACKAGE AND PRICING ON Adventure

THE ADVENTURE SERIES...

SPECIAL PACKAGE AND PRICING ON THE ADVENTURE SERIES... ON DISK OR CASSETTE (specify)

THE COMPLETE SCOTT ADAMS SERIES I (ADVENTURELAND, PIRATE ISLAND, THE COUNT MISSION IMPOSSIBLE, VOODOO CASTLE, STRANGE ODYSSEY, MYSTERY FUN HOUSE, PYRAMID OF DOOM, GHOST TOWN, SAVAGE ISLAND I, SAVAGE ISLAND II, AND GOLDEN VOYAGE PLUS KNIGHT IRONHEART BONUS ADVENTURE

ONLY $12.95 FOR ALL 13.

THE SCOTT ADAMS ADVENTURE SERIES II (THE HULK, SPIDERMAN, BUCKAROO BANZAI, AND THE SORCERER OF CLAYMORGUE CASTLE PLUS 2 BONUS ADVENTURES

ONLY $9.95 FOR ALL 8

COMMAND MODULE FOR ABOVE ADVENTURES (reg $9.95) ONLY 95 CENTS WITH EITHER OF THE ABOVE SERIES

HINT BOOK FOR ALL OF THE ABOVE SCOTT ADAMS ADVENTURES ONLY 95 CENTS WITH EITHER OF THE ABOVE SERIES.

NEW First time on disk — Return to Pirate's Island with Graphics. XBASIC or Editor/Assembler Load $4.95

ADVENTURE EDITOR

NOW WRITE AND EDIT YOUR OWN GAMES FOR THE TI ADVENTURE MODULE!

SUPER BUY!

TEX-COMP is proud to announce the 99/4A Adventure Editor — an all new program which allows you to write your own adventure programs — Scott Adams move over! This disk or cassette based program allows:

1. Any existing adventure for the Adventure command module to be edited, altered, listed, and copied from any storage medium to any other storage medium (i.e. tape to disk or vice versa).
2. New adventure games can be created freely using a template game as a start up step. These games can be conceived using the full power and capabilities of the TI Adventure command module.

The program is written for Mini Memory or Editor Assembler command modules (cassette or disk) and offers a full screen editor and a special easy to use mnemonic language (called A.P.L. (Adventure Program ming Language) and comes with a complete and detailed 75 page manual.

TEX-COMP is also organizing a new TI Adventuring Users Group which will allow low cost exchange of user written adventures between members.

$4.95 Available NOW From TEX-COMP NEW LOW PRICE
Specify Mini Memory (Cassette) or Editor Assembler (Disk) Versions

$4.95 Per Disk NEW BONUS SAVE AS MUCH AS 33%
BUY FIVE DISKS GET ONE FREE
BUY TEN DISKS GET THREE FREE
BUY FIFTEEN DISKS GET FIVE FREE

EDUCATION

DISKETTE PROGRAMS

PHD 5009 Music Skills Trainer 4.95
PHD 5011 Computer Music Box 4.95
PHD 5018 Market Simulation 4.95
PHD 5030 Speak & Spell II (Ex Basic req.) 9.95
PHD 5031 Speak & Math (TE-II req.) 4.95
PHD 5042 Spell Writer (TE-II req.) 4.95
PHD 5026 Bridge Bidding I 4.95
PHD 5039 Bridge Bidding II 4.95
PHD 5041 Bridge Bidding III 4.95
SPECIAL!! BRIDGE BIDDING I, II & III 9.95
PHD 5070 Music Maker Demo (use with module) 4.95
PHD 6010 Mystery Melody (stop the music quiz) 4.95

CASSETTE PROGRAMS*
*see disk versions for requirements i.e. TE-II

PHT 6009 Music Skills Trainer 4.95
PHT 6010 Mystery Melody 4.95
PHT 6011 Computer Music Box 4.95
PHT 6018 Market Simulation 4.95
PHT 6031 Speak & Math 4.95
PHT 6042 Spell Writer 4.95
PHT 6026 Bridge Bidding I 4.95
PHT 6039 Bridge Bidding II 4.95
PHT 6041 Bridge Bidding III 4.95
SPECIAL!! BRIDGE BIDDING I, II & III 9.95
PHT 6070 Music Maker Demo (use with module) 4.95

MANAGEMENT AND BUSINESS

DISKETTE PROGRAMS

PHD 5001 Mailing List (upgraded version) 4.95
PHD 5003 Personal Financial Aids 4.95
PHD 5021 Checkbook Manager 4.95
PHD 5027 Finance Manager 4.95
PHD 5024 Inventory Management 4.95
PHD 5027 Invoice Management 4.95
PHD 5029 Cash Management 4.95
PHD 5038 Lease/Purchase Decisions 4.95
PHD 5075 TI/Multiplan upgrade disk 4.95

CASSETTE PROGRAMS

PHT 6003 Personal Financial Aids 4.95
PHT 6038 Lease/Purchase Decisions 4.95

HOME ENTERTAINMENT

DISKETTE PROGRAMS

PHD 5002 TI-TREK (TE-II req. for speech) 4.95
PHD 5010 Mystery Melody 4.95
PHD 5015 Oldies But Goodies I 4.95
PHD 5017 Oldies But Goodies II 4.95
****SPECIAL Oldies But Goodies I & II 7.95
PHD 5023 Sat. Night Bingo (Ex-Basic & Speech) 4.95
PHD 5037 Draw Poker (Ex-Basic) 4.95

CASSETTE PROGRAMS

PHT 6002 TI-TREK (TEII req. for speech) 4.95
PHT 6010 Mystery Melody 4.95
PHT 6013 Oldies But Goodies I 4.95
PHT 6017 Oldies But Goodies II 4.95
****SPECIAL Oldies But Goodies I & II 7.95
PHT 6026 Sat. Night Bingo (Ex-Basic & Speech) 4.95
PHT 6037 Draw Poker (Ex-Basic) 4.95

COMPUTER PROGRAMMING AIDS

DISKETTE PROGRAMS

PHD 5007 Teach Yourself 99/4A Basic 4.95
PHD 5019 Teach Yourself Extended Basic 4.95
PHD 5004 Programming Aids I 4.95
PHD 5005 Programming Aids II 9.95
PHD 5077 Programming Aids I,II,III 19.95
PHD 5067 Beginning Basic Tutor 4.95
PHD 5076 Text to Speech (Ex-Basic Speech) 4.95
PHD 5098 TI Forth & manual (Ed/Assem req.) 19.95
PHD 5078 TI Forth Demo Disk (Ed/Assem) 4.95
PHD 5079 TI Forth Source Code (2 disks) 4.95

CASSETTE PROGRAMS

PHT 6006 Programming Aids I 4.95
PHT 6007 Teach Yourself 99/4A Basic 4.95
PHT 6019 Teach Yourself Extended Basic 4.95
PHT 6067 Beginning Basic Tutor 4.95

MATH AND ENGINEERING

specify disk or tape with order

Math Routine Library 4.95
Electrical Engineering Library 4.95
Graphing Package 4.95
Structural Engineering Library 4.95
AC Circuit Analysis 4.95
***SPECIAL--ALL 5 OF THE ABOVE ON DISK OR TAPE---17.95

TEX-COMP
P.O. Box 33084, Granada Hills, CA 91344

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 ten days to clear. Add 3% shipping & handling (with a $3 minimum). Add 4.5% east of Mississippi. All prices shown include a 3% cash discount. Add 3% if paying by Visa or Mastercharge. Prices and availability subject to change. Include street address for UPS delivery. California orders add 8.25% sales tax. Canada, Hawaii, Alaska and overseas shipping extra.

AUTHORIZED DEALER

CHARGE IT WE ACCEPT VISA or MASTERCARD

ORDER BY PHONE 24 HOURS A DAY (818) 366-6631
7 Days a Week!

Send $2 for our new 1993 catalog

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $25 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664. 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.

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

Regena on BASIC
Trumpet playing.....Page 8

The Art of Assembly
Odds and ends.....Page 11

Double Column Output
Using TML to catalog disks.....Page 14

Getting more out of your printer
How to use control codes with TI-Writer to handle your style and formatting.....Page 16

Biorhythms
An Extended BASIC program to let your computer compile a chart for you.....Page 19

Reviews
Watchamacalit.....Page 22
MICRO-Reviews: Mail Room, Time Calc.....Page 23

Hardware Project
Modifying a Myarc RS232 card.....Page 25

User Notes
TI-Writer to TI-Base, software control of program records, and even more on circle segments for TML.....Page 26

Classified.....Page 31

Departments
Bugs and Bytes.....Page 7
Comments.....Page 5
Fairs.....Page 18
Feedback.....Page 7
Reader to Reader.....Page 5

*READ THIS

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

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

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

COMMENTS

If you can't afford $35, pay $25

We've received several letters from readers who suggest that they won't be able to renew their MICROpendium subscriptions at the new $35 rate. I thought I had made it clear that those who are on limited incomes may renew at the old rate if they inform us of their situation. We will not turn readers away because they cannot afford the extra $10, whether they live in the U.S. or overseas. It is just that the additional revenue will help us to continue publishing MICROpendium without further reductions in size. There is even the possibility that we will be able to publish some editions of 40 pages, assuming that there is enough money to pay the printer, the post office and writers.

Please, if the extra $10 is a hardship, let us know. You may renew at the old rate.

Of course, the price of back issues will also increase as a result of the subscription increase. However, until the end of September we're selling back issues at a special price. See the ad on page 6 for more information.

DREAMING

Not much seems to be going on in the TI world this month. The summer slump has hit and it's too hot to think. But dreaming comes easy, and what I'm dreaming about is Horizon's SCSI project. Wouldn't it be nice to have a SCSI interface that lets TI and Geneve users connect SCSI hard drives, and potentially other devices, to their computers. Those with the Myarc HFDC are probably very happy with what they've got, but HFDC cards are hard to find these days. And the MFM hard drives that they require are no longer being manufactured, having been replaced by RLL, IDE and EISA drives on the PC. However, I doubt that anyone who wants an MFM drive will have a difficult time finding a used one. Think of all those millions of MFM drives that were sold in the 1980s that won't work with the newer PCs.

Similar shortages are being reported for 360K floppy drives. I've seen new ones selling for $10 each at the warehouse distribution centers of manufacturers such as CompuAdd. But their supplies won't last forever. Even some companies that sell to TI/Geneve users are stocking 720K drives rather than 360K. There's nothing wrong with using a 720K drive instead of a 360K drive. You can still format and use a 360K disk on a 720K drive.

But the reason I'd like a SCSI card, in addition to running a large SCSI hard drive, is the possibility of connecting scanners, tape drives and other SCSI devices to the TI. Of course, once you've got a SCSI card, you still have to write drivers for the hardware. And everyone knows that developing software is often much more difficult than developing hardware. But there's no point letting reality cloud such dreamy potential. A fellow can dream, can't he?

—JK

Reader to Reader

■ Jim McLaren, Site 1, Box 7, Whitefish, Ontario, Canada P0M 3E0, would like to contact people who own the Rave MX01 memory card for problem solving and general interest.

■ Vandsteene Carlo, Elsrakenstraat 52, B-3500 Hasselt, Belgium, writes:

I'm about to write an article about the things (computers and peripherals) Texas Instruments was going to release, but never did because of the withdrawal from the home computer market, now 10 years ago. I've already found some (general) information about the 99/8, the GROM Box .... But over here in Belgium, it's really hard to find. I'm still looking for more information, photographs, photocopies, printouts, articles, etc. in order to get a complete (as possible) survey of what TI intended to produce or had produced but didn't release.

That's why I address myself to you; can anyone send me some information concerning the 99/4, (99/4A), 99/4B, 99/8, 99/2, CC/40, hard drive controller, four-part RS232 card, 128K Super RAM and 374K Ultra RAM card, GROM Box, smart modem, video controller? Anything would help; a photograph you took from your 99/8 if you own one (yes, they have been sold in Texas!) or a photocopy of it out of a magazine or a brochure. I don't expect you to send me your owner's manual, but especially descriptions, test reports, documentation, advertising, pictures and so on (may be photocopies too). So if you have anything, could you please send it to the address above? However little it may be, it will be well appreciated. But, there is a but, be sure that I receive it as soon as possible, because it will take a while to get the article ready in time. Thanks beforehand.

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 76860. We encourage those who answer the questions to forward us a copy of the reply to share with readers.

MICROpendium Back Issue Sale

Because of the subscription price increase that went into effect in August, this is your last chance to get back issues at the current rate of $2.50 each. In fact, depending on how many you buy, the price can be as little as $2.00 per back issue and we pay the postage on U.S. orders. Residents of Canada and Mexico add 30 cents for each issue, other countries 50 cents each issue (surface mail) or $1.50 each issue (air mail). All payments must be in U.S. funds.

Here's how to order:
List the month and year of each back issue (or the volume and number) below. If you order 10 or fewer issues, the price is $2.50 each. If you order more than 10, the price is $2.00 each.

NOTICE: These issues are not available:
Volume 1, Nos. 1 & 2 (February and March, 1984); Volume 2, No. 2 (February 1985); Volume 4, No. 9 (October 1987)

Enter the month and year or the volume and number for each back issue.

_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____		
_____	_____		

Mail completed order forms to:
MICROpendium Back Issues
P.O. Box 1343
Round Rock, TX 78680

Name _____

Address _____

City _____

State _____ **ZIP** _____

Circle payment method

Check/MO **Visa** **M/C**

FOR CREDIT CARD ORDERS ONLY

Credit Card # _____

Exp. Date _____

Signature _____

FEEDBACK

Public thanks

In your May 1993 "Reader to Reader" column you printed my plea for help with the TRACE program that had been printed in the December 1992 issue. Almost by return mail I received two programs from Jim Peterson (Tigercub). One prints the TRACE line numbers to the printer; the other displays the line number on the screen each time Enter is pressed. This is my public thanks to him.

Harold W. Evans, M.D.
Grand Forks, North Dakota

The Feedback column is a reader forum for TI99/4A and Geneve 9640 users to communicate with other readers. The editor will condense excessively lengthy submissions where necessary. We ask that readers restrict themselves to one subject for the sake of simplicity. Mail Feedback items to *MICROpendium Feedback, P.O. Box 1343, Round Rock, TX 78680.*

BUGS & BYTES

Delphi increase?

Some members of the TI-Net on Delphi have been speculating about the meaning of Delphi's slow service lately, commenting that an influx of customers and a slowdown of service happened on GEnie before a price rise on that service.

Hardware explanations have been offered for the slowdown, but one speculation was a possible surcharge for 9600 baud, still not officially supported by Delphi.

Other commercial services have gone through price changes in recent months. Prodigy, after several years of flat-rate fees, started charging by the hour in August. GEnie, on the otherhand, reduced its rates to about $3 per hour in the U.S. GEnie, which instituted the new rate in July, has always run a bit on the slow side while Prodigy runs even slower. CompuServe is probably the fastest service in terms of responding to user commands, but it is by far the most expensive.

Faire siting

Don Walden of the Chicago TI Users Group says the site in Gurnee, Illinois, for the group's TI International Faire Oct. 30 has gotten favorable comments and is attracting some attendees who didn't come previously, particularly some whose non-computing families plan to hang out at the Gurnee Mills mall. Also, the $30-per-table vendor rates have been appreciated.

But you can't please everyone. The fact that the new site is "almost in Wisconsin" was bemoaned in a recent newsletter from the TI Users Group of Will County, Illinois.

BASIC

Playing the trumpet

By REGENA

Summertime is a time for summer band. A couple of my boys have learned to play the trumpet (and cornet). The program this month is designed for a beginning trumpet student who is learning the fingering for the keyboard.

In the first section of this program, the scale of trumpet notes is played. The note is drawn on a staff as the fingering is shown in standard notation as filled-in holes, and the valves are shown on a trumpet. As the scale is being played, the student can press M at any time to return to the main menu screen or press any other key to continue the scale.

In the second section the student may choose a note by using the arrow keys to move the note up or down on the staff, then pressing Enter to hear the note and see it played on the trumpet.

The third section of the program plays a sample tune showing the trumpet and the movement of the valves.

Some of the notes are enharmonic notes — for example, F sharp is the same as G flat. Only one of the notations is used, so the student may need to translate a flat to a sharp or a sharp to a flat.

For each note a frequency F is read in from data along with a code V$. V$ is a combination of three characters which may be a V or an O. V is the key pressed down and the hole filled, and O is open with the key (valve) up.

Line 120 DIMensions variables for the 23 possible notes for the trumpet. F is the frequency, V$ gives the fingering, P is the row position for the corresponding note, and A is the style of note drawn (needing one or two characters).

Lines 150-210 read in from data the 23 values for the dimensioned variables.

Lines 220-520 define graphics characters using FOR-NEXT loops and DATA statements. L$, T3$ and T4$ are strings of graphic characters used in drawing the staff and the trumpet. Colors are also defined. Character sets 9-13 use light and dark yellow and include characters used to draw the trumpet. Set 14 uses red characters and draws the valves, the fingering holes and the notes on the staff. The redefined character sets that stay black are used to draw the musical staff and treble clef.

Lines 530-660 print the main menu screen, wait for a response and branch appropriately.

Lines 670-770 are part of a subroutine. They draw the fingering holes and the trumpet. Lines 780-850 are the rest of the subroutine and draw the musical staff.

Lines 860-970 are the subroutine that plays the scale. Lines 890-960 start with note No. 1 and end with note No. 23. The note is drawn, the tone is sounded and the fingering is printed.

Lines 980-1150 are the subroutine that draws the note N on the staff. If the note is Middle C or lower, ledger lines must be drawn. Some of the notes need a sharp or a flat. Some of the notes (notes on spaces) take two characters (140 and 141) to draw. Notes on the lines need only one character (142).

Lines 1160-1250 sound the tone, then draw the fingering using the information in V$. The three characters in V$ are analyzed. If the character is V, the fingering hole is solid and the valve is down. If the character is V, the fingering hole is open and the valve is up.

Lines 1260-1340 erase a note that has previously been drawn. The sharp or flat and any ledger lines must also be erased, and the staff lines are redrawn.

Lines 1350-1580 are the subroutine to play a note in the second main section of the program. The student may press the up arrow or down arrow to move the note up or down on the staff. The note is drawn as the arrow keys are pressed. When the Enter key is pressed, the fingering is shown.

Lines 1590-1760 are the subroutine to play a sample tune. The DATA statements have timing and note numbers. To put in a different tune, list two numbers for each note in DATA statements replacing Lines 1710-1740. Keep Line 1740 to indicate the end of the tune.

The first number is a timing factor, such as 1 for a quarter note and 2 for a half note. The second number is the note number. Note 1 is the note A below Middle C. Note 23 is G above the staff. Notes increment by 1 from the lowest note. Line 1650 may need to be adjusted for timing. I used a factor of 500 to multiply by the factor in the data statement for the length of the note. Lines 1660 and 1680 are REM statements because they slow down the tune. If you would like to add them, remove the REM. These subroutines called will erase and draw the notes on the musical staff.

Line 1770 ends the program.

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

TRUMPET

```
100 REM TRUMPET !011
110 REM BY REGENA !071
120 DIM F(23),V$(23),P(23),A
(23)!100
130 CALL CLEAR !209
140 PRINT TAB(8);"TRUMPET" !
185
150 FOR J=1 TO 23 !110
160 READ F(J),V$(J),P(J),A(J
)!221
170 NEXT J !224
180 DATA 196,VVO,24,1,208,VO
O,23,2,220,OVO,23,2,233,OOO,
23,1,247,VVV,23,1,262,VOV,22
,2,277,OVV,22,1 !180
190 DATA 294,VVO,22,1,311,VO
O,21,2,330,OVO,21,2,349,OOO,
21,1,370,OVV,20,2,392,VVO,20
,2 !129
200 DATA 415,VOO,20,1,440,OV
O,20,1,466,OOO,19,2,494,VVO,
19,2,523,VOO,19,1,554,OVO,18
,2,587,OOO,18,2 !182
```

(See Page 9)

REGENA ON BASIC—

(Continued from Page 8)

```
210 DATA 622,VOO,18,1,659,OV
O,18,1,698,OOO,17,2 !071
220 PRINT : : :"... DEFINING
 GRAPHICS ...": : :!195
230 FOR J=1 TO 28 !115
240 READ C,C$ !244
250 CALL CHAR(C,C$)!081
260 NEXT J !224
270 DATA 33,182444444448485,
45,000000FF,36,505060FF60404
0C,37,010202FF0808102,38,404
040FF2020202 !026
280 DATA 39,404080FF80808182
,40,101010FF3F50901,41,00000
0FFE0101008,42,828282FF82424
12,43,080808FF08080808 !110
290 DATA 44,080404FF04040408
,47,20100FFF,58,0404FFFF0404
0404,59,0810F0FF,61,04040444
242418 !036
300 DATA 60,10387E101010101,
62,10101010107C381,64,202020
2C3428302 !038
310 DATA 128,FFFFFF7E7E7E7E7
E,129,7E7E7E7E7E7E7E,130,7
E7E7E7E !158
320 DATA 136,000000000000FFF
F,137,FFFF181818181818,138,F
FFFFFFFFFFFFFFF,139,FF818181
818181FF !070
330 DATA 140,000000003C7EFFF
F,141,FF7E3C,142,3C7EFFFFFF7
E3C !065
340 L$="————————" !141
350 FOR J=96 TO 122 !222
360 READ C$ !254
370 CALL CHAR(J,C$)!088
380 NEXT J !224
390 DATA FFFFFFFFFFFFFFFF,80
C0E0F0F8FCFEFF,FFFEFCF8F0E0C
08,FFFFFCFBF7EFDFDF,C73FFFFF
FFFFFFFF !014
400 DATA E3FCFFFFFFFFFFFF,FF
FF3FDFEFF7FBFB,3F3F7F7F7FFFF
FFF,FFFCF0E0C0C0808,FF3F0F07
03030101 !015
410 DATA FCFCFEFEFEFFFFFF,FF
FFFF7F7F7F3F3F,8080C0C0E0F0F
CFF,01010303070F3FFF,FFFFFFF
EFEFEFCFC !114
420 DATA 1F1F0F0703,FFFFFFFF
FFFF3F07,FFFFFFFFFFFFFCE,F8F
8F0E0C,0000000000000FFF,0000
00000FFFFFFF !134
430 DATA 00000FFFFFFFFFFF,0F
FFFFFFFFFFFFFF,FF0F,FFFFFF0F
,FFFFFFFFFF0F,FFFFFFFFFFFFFF
0F !222
440 VV$=CHR$(138)!056
450 VO$=CHR$(139)!050
460 FOR J=9 TO 14 !118
470 READ C !218
480 CALL COLOR(J,C,1)!067
490 NEXT J !224
500 DATA 12,12,12,12,11,7 !2
42
510 T3$="  "&CHR$(129)&" "&C
HR$(129)&" "&CHR$(129)&"  "
!153
520 T4$="``"&CHR$(129)&"`"&C
HR$(129)&"`"&CHR$(129)&"``"
!026
530 CALL CLEAR !209
540 CALL SCREEN(8)!153
550 PRINT TAB(8);"TRUMPET" !
185
560 PRINT : :"CHOOSE": :!254
570 PRINT "1  PLAY SCALE" !1
59
580 PRINT "2  CHOOSE NOTES"
!078
590 PRINT "3  SAMPLE TUNE" !
002
600 PRINT "4  END PROGRAM":
: :!019
610 CALL KEY(3,K,S)!190
620 IF S<1 THEN 610 !109
630 IF (K<49)+(K>52)THEN 610
 !121
640 CALL CLEAR !209
650 ON K-48 GOSUB 860,1350,1
590,1770 !120
660 GOTO 530 !099
670 REM DRAW SCREEN !232
680 PRINT TAB(11);VO$;" ";VO
$;" ";VO$ !224
690 PRINT TAB(11);"1 2 3" !1
35
700 PRINT :TAB(11);CHR$(137)
;" ";CHR$(137);" ";CHR$(137)
;TAB(25);"st" !231
710 PRINT "  a       ";CHR$(
128);" ";CHR$(128);" ";CHR$(
128);"    stuv``" !245
720 PRINT "  ````cd";T4$;"ef
````````" !065
730 PRINT " b   gh";T3$;"ij
 wxyz``" !180
740 PRINT TAB(7);"` ";T3$;"
`   wx" !244
750 PRINT TAB(7);"kl";T3$;"m
n" !016
760 PRINT TAB(7);"op";T4$;"q
r" !033
770 PRINT TAB(11);CHR$(130);
" ";CHR$(130);" ";CHR$(130)!
107
780 PRINT : :" !" !049
790 PRINT "—$-";L$!054
800 PRINT "-%&-";L$!048
810 PRINT "-'()";L$!048
820 PRINT "-*+,";L$!057
830 PRINT "-/:;";L$!092
840 PRINT " =" !227
850 RETURN !136
860 PRINT "PRESS M TO GO=TO
MAIN MENU" !242
870 PRINT "PRESS ANY KEY TO
CONTINUE": :!204
880 GOSUB 680 !250
890 FOR N=1 TO 23 !114
900 GOSUB 980 !039
910 GOSUB 1160 !220
920 CALL KEY(3,K,S)!190
930 IF S<1 THEN 920 !164
940 IF (K=77)+(K=109)THEN 97
0 !020
950 GOSUB 1260 !064
960 NEXT N !228
970 RETURN !136
980 IF N>1 THEN 1020 !004
990 CALL HCHAR(23,14,45,3)!2
27
1000 CALL HCHAR(24,14,45,3)!
228
1010 GOTO 1040 !099
1020 IF N>5 THEN 1040 !029
1030 CALL HCHAR(23,14,45,3)!
227
1040 ON A(N)GOSUB 1110,1130
!043
```

(See Page 10)
```

REGENA ON BASIC—

(Continued from Page 9)

```
1050 IF (N=2)+(N=7)+(N=12)+(
N=14)+(N=19)THEN 1060 ELSE 1
080 !151
1060 CALL HCHAR(P(N),14,64)!
017
1070 GOTO 1100 !159
1080 IF (N=5)+(N=10)+(N=17)+
(N=22)THEN 1090 ELSE 1100 !1
41
1090 CALL HCHAR(P(N),14,35)!
015
1100 RETURN !136
1110 CALL HCHAR(P(N),15,142)
!064
1120 RETURN !136
1130 CALL HCHAR(P(N),15,140)
!062
1140 CALL HCHAR(P(N)+1,15,14
1)!250
1150 RETURN !136
1160 CALL SOUND(600,F(N),2)!
035
1170 FOR J=1 TO 3 !059
1180 IF SEG$(V$(N),J,1)="V"
THEN 1220 !197
1190 CALL HCHAR(4,11+2*J,139
)!253
1200 CALL HCHAR(7,11+2*J,137
)!254
1210 GOTO 1240 !043
1220 CALL HCHAR(4,11+2*J,138
)!252
1230 CALL HCHAR(7,11+2*J,136
)!253
1240 NEXT J !224
1250 RETURN !136
1260 CALL HCHAR(P(N),14,32,2
)!186
1270 IF N<3 THEN 1290 !021
1280 CALL HCHAR(P(N)+1,15,32
)!200
1290 IF N>5 THEN 1320 !054
1300 CALL HCHAR(23,14,32,3)!
223
1310 CALL HCHAR(24,14,32,3)!
224
1320 CALL VCHAR(18,15,45,5)!
248
1330 CALL VCHAR(18,14,45,5)!
247
1340 RETURN !136
1350 PRINT "USE < > TO MOVE
THEN ENTER" !043
1360 PRINT "PRESS M TO GO TO
 MAIN MENU": :!183
1370 GOSUB 680 !250
1380 N=16 !061
1390 GOSUB 980 !039
1400 CALL KEY(3,K,S)!190
1410 IF S<1 THEN 1400 !134
1420 IF (K=77)+(K=109)THEN 1
580 !121
1430 IF (K<>69)+(K<>101)+(K<
>11)=-3 THEN 1490 !057
1440 GOSUB 1260 !064
1450 N=N+1 !021
1460 IF N<23 THEN 1390 !172
1470 N=23 !059
1480 GOTO 1390 !194
1490 IF (K<>88)+(K<>120)+(K<
>10)=-3 THEN 1550 !119
1500 GOSUB 1260 !064
1510 N=N-1 !022
1520 IF N>0 THEN 1390 !119
1530 N=1 !006
1540 GOTO 1390 !194
1550 IF K<>13 THEN 1400 !114
1560 GOSUB 1160 !220
1570 GOTO 1400 !204
1580 RETURN !136
1590 PRINT : : :!187
1600 N=5 !010
1610 GOSUB 680 !250
1620 RESTORE 1710 !017
1630 READ TM,NN !135
1640 IF TM=9999 THEN 1750 !2
32
1650 CALL SOUND(TM*500,F(NN)
,2)!212
1660 REM GOSUB 1260 !067
1670 N=NN !168
1680 REM GOSUB 980 !027
1690 GOSUB 1170 !230
1700 GOTO 1630 !179
1710 DATA 1,4,4,9,1,11,1,13,
4,16,1,18,1,16,4,14,1,13,1
1,4,16,1,18,1,20,3,21,3,16
245
1720 DATA 3,14,3,13,3,11,1,1
3,1,9,1,6,5,11,1,4,3,14,3,13
,4,11 !123
1730 DATA 1,13,1,14,3,16,3,1
4,4,13,1,14,1,16,3,18,3,6,3,
16,1,14,1,13,1,11,3,9,2,11,1
,13,6,9 !234
1740 DATA 9999,1 !240
1750 CALL SOUND(1,9999,30)!1
57
1760 RETURN !136
1770 END !139
```

Vendors sought for Chicago TI Faire

TI vendors looking for price breaks will get one at this year's International Chicago TI Faire. Rates for tables have been set at $30 per table, less than half of the price vendors paid for tables at least year's faire. Electricity will be provided free of charge to vendor tables, according to Jim Deards, Faire chairman.

Deards notes that a limited number of systems will be available for vendor use. Vendors need to let him know if they need a system, and what kind of setup is needed.

The Faire begins at 9 a.m. Oct. 30 at the Holiday Inn at 6161 West Grand Ave., Gurnee, Illinois.

A limited number of rooms is reserved for the Faire at the Holiday Inn. Persons wishing to stay at the Holiday Inn need to fill out a reservation card.

Reservation cards and vendor information are available from Deards at 1536 Amarillo Dr., Carpentersville, IL 60110, telephone (708) 426-6301.

THE ART OF ASSEMBLY — PART 26

Odds and Ends

By BRUCE HARRISON

Today we're picking up some loose ends left behind from previous parts in this series, and passing along some insights. We'd like to start with a couple of sincere "Thank You" notes.

First goes to Harley Ryan, Jr. of Whitehall, OH. He got us started on the path to solving the "strange case" of TI's GPLLNK, which we talked about in the July 1992 issue. Mr. Ryan found a solution for the Auto-Run Option 3 program, which led us down quite a path. Thanks also to Miller's Graphics, for publishing the solution passed along to us by Mr. Ryan.

Second thanks goes to Merle Vogt, of Von Ormy, TX. He passed along a different way of solving the utilities problem for Option 5 programs, plus another suggestion for the opening and closing sections of an Assembly program, for Option 5 or Option 3.

THE STRANGE CASE...

Back in Part 14, which appeared in July 1992, we reported that we'd never been able to get TI's GPLLNK to work correctly under either an Auto-Start Option-3 or an Option-5 situation. Shortly after that article appeared, Mr. Ryan sent a letter which gave a solution to the problem for Auto-Start Option-3 programs. He found his information in the manual for the Miller's Graphics Explorer. It seems the TI GPLLNK depends on the state of the GROM address when it's called, and that is set differently by Option-3 depending on whether the program loaded has auto-start or not.

Miller's suggestion was to read the GROM address in the manner given by the E/A manual, then add >63 if your program is to auto-start and write that back through GRMWA. This works. In today's sidebar is the short test program we put together immediately after getting Mr. Ryan's letter. This will work for Auto-start Option-3 programs with both the E/A module and the Mini-Memory. Needless to say, we were delighted. Our delight was, however, short-lived, because Option-5 turned out to be quite a different can of worms.

If one tries adding Hex63 to the GROM address for an Option-5 program, the GPLLNK does not work! After doing some detective work, inserting a "HEXDIS" routine into our source code, we found that we could make an Option-5 program work with GPLLNK under certain conditions by adding >2E4 to the GROM address. That was not all. We also had to stash and restore all of the low memory area from >2000 through >2676 instead of the shorter utilities portion from >2094 through >23BA, as we'd suggested in Part 14.

That worked. It worked as long as we entered our program from E/A Option-5 itself. Part two of today's sidebar shows the source code for that solution. Of course we couldn't stop trying things, so, since we have a RAMdisk on our system which allows direct running of Option-5 programs from its menu screen, we tried that. We also have a P-Gram, which allows us to select either Extended BA-

(See Page 12)

SIDEBAR 26

```
* PART ONE - AUTO-START
* OPTION-3 FROM E/A MENU
* TEST PROGRAM - PUBLIC DOMAIN
* BY B. HARRISON
*
      REF  GPLLNK,GRMRA,GRMWA,KSCAN
      REF  VMBW
      DEF  START
START
      LWPI >20BA        LOAD USER WORKSPACE
      MOVB @GRMRA,@GRMSAV READ HIGH BYTE GROM ADDRESS
      NOP               KILL TIME
      MOVB @GRMRA,@GRMSAV+1 READ LOW BYTE GROM ADDRESS
      NOP               KILL TIME
      DEC  @GRMSAV      DECREMENT THE ADDRESS
      A    @HEX63,@GRMSAV ADD HEX 63
      MOVB @GRMSAV,@GRMWA WRITE HIGH BYTE BACK
      NOP               KILL TIME
      MOVB @GRMSAV+1,@GRMWA WRITE LOW BYTE BACK
      NOP               KILL TIME
      LI   R0,22*32+8   POINT TO ROW 23, COL 9
      LI   R1,TEST      MESSAGE
      LI   R2,15        LENGTH OF MESSAGE
      BLWP @VMBW        WRITE TO SCREEN
      BLWP @GPLLNK      CALL GPLLNK
      DATA >34          FOR BEEP SOUND
KEYLOO BLWP @KSCAN       SCAN KEYBOARD
      LIMI 2            ALLOW INTERRUPTS
      LIMI 0            THEN SHUT OFF INTERRUPTS
      CB   @ANYKEY,@>837C KEY STRUCK?
      JNE  KEYLOO       IF NOT, KEEP SCANNING
      LWPI >83E0        LOAD GPL WORKSPACE
      B    @>6A         RETURN TO GPL INTERPRETER
GRMSAV DATA 0           WORD TO STASH GROM ADDRESS
HEX63  DATA >63         VALUE HEX 63
ANYKEY BYTE >20         KEYBOARD TEST BYTE
TEST   TEXT 'THIS IS A TEST ' MESSAGE
      END  START



* PART TWO - FOR E/A OPTION 5
* THIS DOES NOT WORK ALL CASES
*
      REF  GPLLNK,GRMRA,GRMWA,KSCAN
      REF  VMBW
      DEF  SFIRST,SLOAD,SLAST
SFIRST
SLOAD
      LWPI WS           LOAD OUR WORKSPACE
      LI   R9,DATALD    POINT AT SAVED UTILITIES
      LI   R10,>2000    POINT AT LOW MEMORY
      LI   R4,>2676->2000 BYTES TO MOVE
PUTUT  MOV  *R9+,*R10+   MOVE ONE WORD
      DECT R4           DECREMENT COUNTER BY TWO
      JNE  PUTUT        IF NOT ZERO, REPEAT
      MOVB @GRMRA,@GRMSAV READ HIGH BYTE GROM ADDRESS
      NOP               KILL TIME
      MOVB @GRMRA,@GRMSAV+1 READ LOW BYTE GROM ADDRESS
      NOP               KILL TIME
      DEC  @GRMSAV      DECREMENT THE ADDRESS
      A    @HEX2E4,@GRMSAV ADD HEX 2E4
      MOVB @GRMSAV,@GRMWA WRITE HIGH BYTE BACK
      NOP               KILL TIME
      MOVB @GRMSAV+1,@GRMWA WRITE LOW BYTE BACK
```

THE ART OF ASSEMBLY—

(Continued from Page 11)

SIC or Editor/Assembler with just the press of a key. This opened yet another can of worms for us. If, for example, the menu selection at the bottom of the RAMdisk menu said Extended BASIC when we direct ran our Option-5 program, the call to GPLLNK placed us in XB without the normal character set. A mess. Typing in BYE <Enter> got us back to the RAMdisk menu, but taught a valuable lesson. Adjusting the GROM address in this way couldn't be done under those circumstances.

THE "FINAL SOLUTION"

Finally we knew why Doug Warren and Craig Miller developed their own PLLNK, rather than try using TI's version. To make our Option-5 test program work with the P-GRAM and RAMdisk situation we just described, we had to devise a means of first finding the right GROM in the P-GRAM. This we did by a trial and error method of comparing certain key bytes in each GROM address from >6000 up, in >2000 steps. Once we'd found the correct GROM, (in our case at >E000) we set up the correct GROM address as >E892, wrote that through GRMWA, and everything worked. This also works when run from Option-5 of E/A, either using P-Gram or using an E/A module. As you can see in Part 3 of the sidebar, this takes a whale of a lot of work. It's also just possible that no E/A is actually in the P-Gram, so we put in an escape exit to the vector at >0000 in case the E/A GROM was not found.

The why, then, for Warren and Miller's GPLLNK is probably that using the TI GPLLNK can be a lot more trouble than it's worth. We've shown that it can be done, but all the code that was necessary to make it work adds up to more memory usage than just putting the Warren/Miller GPLLNK into our source file to begin with. Any of our readers can excerpt the source code shown in part 3 of the sidebar and prove for himself that it works, but we really don't recommend this approach for general use. (See Part 14 in the July '92 issue for how to use this source code, should you wish to try it out.)

OTHER INS AND OUTS

Merle Vogt is a devoted reader of this column, and has more than once come up with suggestions for ways of doing things that are different from our ways. We enjoy this, and will pass along two of his suggested methods today. First, a different way of handling the entry and exit problem for any Assembly program. At the beginning, try this:

```
START   STWP R12       STASH THE INCOMING WORKSPACE
        MOV  R12,@E1+2 MOVE THAT TO LOCATION BELOW
        MOV  R11,@E2+2 MOVE THE RETURN ADDRESS
        LWPI MYWS      LOAD YOUR WORKSPACE
```

(program continues)

Then at the exit:

```
EXIT    CLR  @>837C   CLEAR GPL STATUS BYTE
E1      LWPI 0        DUMMY ADDRESS FOR LWPI
E2      B    @0       DUMMY RETURN ADDRESS
MYWS    BSS  32
```

This works as follows: The STWP at the very beginning places the workspace pointer for whatever workspace was in use before your program started into R12 of that workspace. Next, the con-

(See Page 13)

```
        NOP             KILL TIME
        LI   R0,22*32+8   POINT TO ROW 23, COL 9
        LI   R1,TEST      MESSAGE
        LI   R2,15        LENGTH OF MESSAGE
        BLWP @VMBW        WRITE TO SCREEN
        BLWP @GPLLNK      CALL GPLLNK
        DATA >34          FOR BEEP SOUND
KEYLOO BLWP @KSCAN        SCAN KEYBOARD
        LIMI 2            ALLOW INTERRUPTS
        LIMI 0            THEN SHUT OFF INTERRUPTS
        CB   @ANYKEY,@>837C KEY STRUCK?
        JNE  KEYLOO       IF NOT, KEEP SCANNING
        LWPI >83E0        LOAD GPL WORKSPACE
        B    @>6A         RETURN TO GPL INTERPRETER
WS      BSS  32           OUR WORKSPACE
GRMSAV DATA 0             WORD TO STASH GROM ADDRESS
HEX2E4 DATA >2E4          VALUE HEX 2E4
ANYKEY BYTE >20           KEYBOARD TEST BYTE
TEST   TEXT 'THIS IS A TEST ' MESSAGE
       EVEN
DATALD BSS  >2676->2000  STORAGE AREA FOR UTILITIES
SLAST
       DEF  SAVIT         DEFINE SAVIT ENTRY POINT
       REF  SAVE          REF TI SAVE UTILITY
SAVIT
       LWPI WS            LOAD OUR WORKSPACE
       LI   R9,>2000      POINT AT START OF LOW MEMORY
       LI   R10,DATALD    POINT AT DATA SPACE FOR UTILITIES
       LI   R4,>2676->2000 BYTES TO MOVE
GETUT  MOV  *R9+,*R10+    MOVE ONE WORD
       DECT R4            DECREMENT COUNTER BY TWO
       JNE  GETUT         IF NOT ZERO, REPEAT
       B    @SAVE         BRANCH TO SAVE UTILITY
       END

* PART 3 - GENERAL CASE OPTION-5
* WILL ALSO WORK FROM RAMDISK MENU RUN OPTION
* AND P-GRAM WITH E/A INCLUDED


       REF  GPLLNK,GRMRA,GRMWA,KSCAN
       REF  VMBW,GRMRD
       DEF  SFIRST,SLOAD,SLAST
SFIRST
SLOAD
       LWPI WS            LOAD OUR WORKSPACE
       LI   R9,DATALD     POINT AT SAVED DATA
       LI   R10,>2000     AND AT START OF LOW MEMORY
       LI   R4,>2676->2000 BYTES TO MOVE
PUTUT  MOV  *R9+,*R10+    MOVE A WORD
       DECT R4            DEC R4 BY TWO
       JNE  PUTUT         NOT ZERO, REPEAT
       LI   R3,>6000      SET R3 TO >6000
TGROM  BL   @WGA          WRITE ADDRESS IN R3 TO GROM ADDRESS
       BL   @RGD          READ ONE BYTE
       CB   R5,@HEXAA     IS THAT HEX AA?
       JNE  NXTGRM        IF NOT, SKIP AHEAD
       INC  R3            ELSE INCREMENT ADDRESS IN R3
       BL   @WGA          SET FOR THAT ADDRESS
       BL   @RGD          READ BYTE
       CB   R5,@HEX01     COMPARE TO HEX01
       JNE  NXTGRM        IF NOT EQUAL, JUMP AHEAD
       INC  R3            ELSE LOOK AT NEXT ADDRESS
       BL   @WGA          SET THAT
       BL   @RGD          READ FROM GROM
       CB   R5,@HEX01     COMPARE
       JNE  NXTGRM        IF NOT EQUAL, JUMP
       AI   R3,5          ADD 5 TO R3 ADDRESS
       BL   @WGA          SET THAT ADDRESS
       BL   @RGD          READ A BYTE
       CB   R5,@HEX10     COMPARE TO HEX 10
       JEQ  GRMOK         IF EQUAL, WE'VE FOUND E/A IN GROM
NXTGRM ANDI R3,>F000      MASK ALL BUT FIRST NYBBLE
```

THE ART OF ASSEMBLY—

(Continued from Page 12)

tents of that R12 are moved into the spot two bytes beyond label E1, replacing the zero that was there in the source code. The return address from R11 of the original workspace will then be placed two bytes past label E2, replacing the zero that was there. If, for example you entered with the GPL workspace, and a return address of >0070, the code at E1 and E2 will be self-modified to:

```
E1      LWPI >83E0     LOAD GPL WORKSPACE
E2      B    @>0070    BRANCH TO HEX 70 ADDRESS
```

On some occasions we have tried this kind of approach with success, so we're sure Merle is right when he says this will work for just about any way of running your Assembly program.

THE UTILITIES PROBLEM

We said more than once in this column that there are nearly as many ways of doing anything in Assembly as there are people trying to do it. Vogt has once again proved that to be true. He came up with a way of saving and restoring the E/A utilities for Option-5 programs that we'd never have thought of, but it's an effective way of doing the deed. With another thanks to Merle, here's that way.

Start by making this very simple source file:

```
        DEF  SFIRST,SLOAD,SLAST
SFIRST  EQU  >2000
SLOAD   EQU  >2000
SLAST   EQU  >2676
        END
```

Assemble this and name it MODULE2/O or MODULE2OBJ, or whatever you like. Now load this under Option 3, then load TI's SAVE utility and run program name SAVE. Call the memory image file MODULE2.

Now place the labels SFIRST, SLOAD and SLAST in the appropriate places in your main program's source file. Assemble that, then load it under Option 3, load TI's SAVE utility, and SAVE that as MODULE1.

Here's where it gets a tad tricky. Using a sector editor, find the first sector of the MODULE1 file and change its first two bytes from >0000 to >FFFF. Write the modified sector back to the disk. Make sure that MODULE1 and the MODULE2 file you made previously are on the same disk.

Choose Option 5 and load MODULE1 from whatever drive the disk happens to be in. Because the header of MODULE1 has that >FFFF in it, the Option 5 loader will look on the same drive for a file called MODULE2, and will load that into memory as well. Voila! The utilities area from the E/A will be restored into place in low memory just where it belongs and your program can use those utilities by simple REFs.

If your main program itself is large enough to become more than one file when saved by TI's SAVE utility, then you must rename the MODULE2 to be the next number in line, (e.g. MODULE3 or MODULE4) and make that sector edit in whatever was the last file in your main program's sequence.

We haven't actually tried this yet, but have no doubt that it will work. Normally we try to use and recommend methods that don't assume our readers have such programs as sector editors handy,

(See Page 15)

```
        AI   R3,>2000     ADD >2000
        CI   R3,>0000     IS THAT NOW ZERO?
        JEQ  ALTEX        IF SO, NO E/A FOUND
        JMP  TGROM        ELSE GO LOOK AT NEXT POSSIBLE ADDRESS
GRMOK   ANDI R3,>F000     MASK OFF ALL BUT LEFT NYBBLE
        AI   R3,>0892     ADD >892
        BL   @WGA         WRITE THAT ADDRESS TO GRMWA
        LI   R0,22*32+8   POINT TO ROW 23, COL 9
        LI   R1,TEST      MESSAGE
        LI   R2,15        LENGTH OF MESSAGE
        BLWP @VMBW        WRITE TO SCREEN
        BLWP @GPLLNK      CALL GPLLNK
        DATA >34          FOR BEEP SOUND
KEYLOO  BLWP @KSCAN       SCAN KEYBOARD
        LIMI 2            ALLOW INTERRUPTS
        LIMI 0            THEN SHUT OFF INTERRUPTS
        CB   @ANYKEY,@>837C KEY STRUCK?
        JNE  KEYLOO       IF NOT, KEEP SCANNING
        LWPI >83E0        LOAD GPL WORKSPACE
        B    @>6A         RETURN TO GPL INTERPRETER
ALTEX   LIMI 2            ESCAPE EXIT POINT
        LWPI >83E0        LOAD GPL WORKSPACE
        BLWP @0           RETURN TO VECTOR AT 0

WGA     MOVB R3,@GRMWA WRITE HIGH BYTE OF ADDRESS
        SWPB R3           SWAP BYTES R3 AND KILL TIME
        MOVB R3,@GRMWA WRITE LOW BYTE OF ADDRESS
        SWPB R3           SWAP BYTES R3 AND KILL TIME
        RT                RETURN
RGD
        MOVB @GRMRD,R5    READ A GROM BYTE INTO R5
        NOP               KILL TIME
        RT                RETURN
```

THE ART OF ASSEMBLY—

(Continued from Page 14)

but if you have one, as many people do, then Merle's approach may be just the thing for building your bridge to Option-5.

NOBODY'S PERFECT

Some time back, we offered some business advice in this column, and boldly asserted that in the years since 1988, when we first started selling software to the TI community, we'd never had a check bounce. It's August 1992 now, and we can't say that any more. An overseas customer wrote us a check on a U.S. bank, and that came bouncing back to us as "ACCOUNT CLOSED". Sooner or later it had to happen, but that's only one check out of hundreds that we've received, so the TI community still will get prompt shipments from us. Of course if it happens again...

Today's column has been a real mixed bag of stuff to chew on. Next month we'll try to stick to one topic, and beat that one to death instead of skipping around the map.

```
WS     BSS  32           OUR WORKSPACE
GRMSAV DATA 0            WORD TO STASH GROM ADDRESS
HEXAA  BYTE >AA          HEX VALUE AA
HEX01  BYTE >01          VALUE 1
HEX10  BYTE >10          VALUE HEX 10
ANYKEY BYTE >20          KEYBOARD TEST BYTE
TEST   TEXT 'THIS IS A TEST ' MESSAGE
       EVEN
       DEF  SAVIT        DEFINE SAVIT ENTRY
       REF  SAVE         REF THE SAVE UTILITY
DATALD BSS >2676->2000   UTILITY STORAGE AREA
SLAST
SAVIT
       LWPI WS           LOAD OUR WORKSPACE
       LI   R9,>2000     POINT AT START OF LOW MEM
       LI   R10,DATALD   AND AT STORAGE AREA
       LI   R4,>2676->2000 NUMBER OF BYTES
GETUT  MOV  *R9+,*R10+   MOVE A WORD
       DECT R4           DEC COUNT BY TWO
       JNE  GETUT        IF NOT ZERO, REPEAT
       B    @SAVE        BRANCH TO TI SAVE UTILITY
       END
```

Double column output

Using TML to catalog disks

By OLLIE HEBERT

This disk catalog was written for use in The Missing Link environment. Screens created in Extended BASIC programs aren't usable with TML, so your usual XB disk cataloger is of no use here.

There are three output options: screen only, screen and printer, or printer only. Each screen contains up to 46 files, and you may use TML's screen dump (Ctrl-Fctn) if you need a more compact printout. TML was written by Harry Wilhelm, is available from Texaments, and requires XB, 32k, and a disk system.

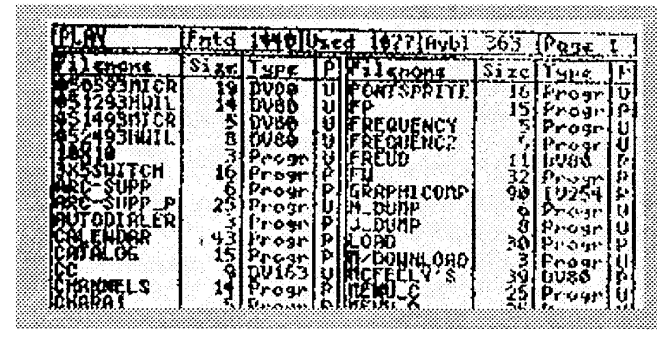
Read the REMarks in statements 110 and 120 and make changes as appropriate. 57FONT as well as your default font file must be available to this program.

In statement 160, KEY (3 causes all keypresses to be returned as upper case. Prescan is turned off here also.

Statements 170-310 are the subroutines: 170-180 loads a font, 190 clears the screen while 200 erases a portion of the screen. 210 or 220 gets a keypress, 230 and 240 are HONK and BEEP sounds. 250-260 replace diskname and filename characters outside the TML range of 32-127 with either a space (for printer options), or character 127 (for screen only option). 270 draws lines for the files, 280 ends the program, and 290-310 traps for errors when files are opened.

Statement 320 loads 57FONT, changes character size to 5x7, and redefines CHR$(127) to a small rectangle.

Statements 330-390 print the title screen and then gets responses for F$ (default font), P (option 1-4), D$ (disk), P$ (printer), and then waits for a keypress. There is space for 19 characters to describe your drive, maximum sector count and file size is 9999, and maximum files is 999.



Statements 400-410 open the printer and disk files and get the disk data. Y=Y+2 accounts for sectors 0 and 1 which are used by the formatting process and are therefore not available.

Statement 420 gets the data for the first file. This is done here to get "one file ahead" which prevents going to a new screen and then not having any data for it.

Statements 430-460 print the disk infor-

(See Page 15)

TML-CATALOG—

(Continued from Page 14)

mation to the screen, and 470-490 send it to the printer.

Statements 500-590 use a loop to read and display/print data for up to 46 files. For fields that would have contained spaces, periods are added to the printout for easier reading.

Statement 510 sets variables R (rows 23 to 177 step 7 twice) and C (columns 2 or 123) for screen printing. In 520, V is either 0 or -2.

Statements 540-550 print a file to the screen, and statements 560-570 send file data to the printer. In TML, using variables for both row and column in CALL LINK("PRINT") gives automatic updating for the next print. This is an excellent feature, but isn't wanted here because R and C are also used in sub 270. To turn this feature off, you must enclose both row and column variables within parentheses.

Statements 530 and 560 illustrate an alternate way of determining filetype as opposed to the usual five element array.

Statement 580 reads data for the next file, and Statement 590 ends the loop. In 600, if J$="" (end of file), the file is closed, "Page=" is erased and then the same spot is re-used for "Files=".

Statements 610-640 are end of page and end of disk procedures.

Correspondence and phone calls are welcome. Write Hebert at Rt. 4, Box 23; Brewton, AL 36426; 205 867-7193.

TML_CATALG

```
100 ! TML_CATALG  (TML) v1.2
 Ollie Hebert        May '93
 Rt. 4, Box 23
 Brewton, AL 36426  USA
 Ph: 205 867-7193 !034
110 ! Catalog: to screen (46
 files per screen), printer,
 or both.  Change the three
 defaults in statement 120
 to your usual responses. !2
02
120 D$="DSK1." :: F$="68FONT
 :: P$="PIO" ! If your TML
 fonts are stored on other
 than "DSK1.", change
 statement 170 to suit you.
!055
130 DATA 1,38,DISK CATALOG f
or THE MISSING LINK,12,63,Pr
ogram by Ollie Hebert,36,58,
Your Default Font?,59,46,0 O
utput to Screen Only,66,46
140 DATA 1 Output to Screen
and Printer,73,46,2 Output t
o Printer Only,80,46,3 Quit,
94,86,Your Choice?,117,73,Ca
talog Drive,140,63 !142
150 DATA Printer Parameters?
,========== ==== ==========
=,"Filename   Size Type  " !
095
160 CALL KEY(3,M,R):: GOTO 3
20 :: CALL ERR :: CALL LINK
:: CALL SOUND :: A,B,C,C$,F,
I,I$,J,J$,K,K$,L,L$,P,Q,S,T$
,V,W,X,Y,Z :: !@P- !160
170 GOSUB 190 :: CALL LINK("
CHSIZE",8,8):: CALL LINK("PR
INT",92,65,"Loading "&I$)::
OPEN #1:"DSK1."&I$,INTERNAL,
VARIABLE 241,INPUT !183
180 FOR I=33 TO 123 STEP 15
:: INPUT #1:I$ :: CALL LINK(
"CHAR",I,I$):: NEXT I :: ON
ERROR STOP :: CLOSE #1 :: GO
SUB 190 :: RETURN !221
190 CALL LINK("CLEAR"):: RET
URN !065
200 CALL LINK("PE"):: CALL L
INK("FILL",161,1,192,240)::
CALL LINK("PD"):: RETURN !18
4
210 R=186 :: K$="T"&C$&"M"&C
$&"L"&C$&"."&C$ !008
220 A=A+1+8*(A>7):: CALL LIN
K("PRINT",R,(C),SEG$(K$,A,1)
):: CALL KEY(3,M,B):: IF M<0
 THEN 220 ELSE CALL LINK("PR
INT",R,(C),CHR$(M)):: RETURN
 !097
230 CALL SOUND(-2E2,22E1,0):
: RETURN !165
240 CALL SOUND(-2E2,14E2,0):
: CALL SOUND(1,2E4,30):: RET
URN !140
250 M=LEN(J$):: FOR A=1 TO M
 :: B=ASC(SEG$(J$,A,1)):: IF
 B<32 OR B>127 THEN J$=SEG$(
J$,1,A-1)&CHR$(127+95*(P>0))
&SEG$(J$,A+1,9)!131
260 NEXT A :: RETURN !225
270 CALL LINK("BOX",21,C-1,R
+V,C+117):: CALL LINK("BOX",
12,C+77,R+V,C+107):: CALL LI
NK("LINE",12,C+52,R+V,C+52):
: RETURN !013
280 IF F$="57FONT" THEN END
ELSE I$=F$ :: ON ERROR 290 :
: GOSUB 170 :: END !164
290 CALL ERR(B,B,B,B):: GOSU
B 230 :: W=0 :: ON ERROR 300
 :: CLOSE #1 !248
300 ON ERROR 310 :: CLOSE #2
 :: ON ERROR STOP !180
310 GOSUB 200 :: IF B>170 TH
EN RETURN 330 ELSE PRINT "Fo
nt file ";I$;" error.": :"LI
ST 120 and 170 to find": :"a
nd fix the problem." :: END
!226
320 I$="57FONT" :: ON ERROR
290 :: GOSUB 170 :: CALL LIN
K("CHSIZE",5,7):: CALL LINK(
"CHAR",127,"003030303"):: C$
=CHR$(127)!087
330 RESTORE :: FOR I=1 TO 10
 :: READ R,C,I$ :: CALL LINK
("PRINT",R,C,I$):: NEXT I ::
 READ L$,T$ :: CALL LINK("PR
INT",36,153,F$)!008
340 CALL LINK("PRINT",117,14
3,D$):: CALL LINK("PRINT",14
0,163,P$):: IF L=0 THEN L=1
:: GOSUB 240 :: CALL LINK("I
NPUT",36,153,F$,10,F$)!076
350 R=94 :: C=151 :: K$="0"&
C$&"1"&C$&"2"&C$&"3"&C$ :: G
OSUB 240 !039
360 GOSUB 220 :: IF M=51 THE
N 280 ELSE IF M<48 OR M>50 T
HEN GOSUB 230 :: GOTO 360 EL
SE GOSUB 240 :: CALL LINK("I
NPUT",117,143,D$,19,D$)!073
370 P=M-48 :: Q=3-160*(P=2):
: IF P AND W=0 THEN W=1 :: G
OSUB 240 :: CALL LINK("INPUT
",140,163,P$,15,P$)!027
380 GOSUB 240 :: CALL LINK("
PRINT",186,6,"<FCTN-E> or In
sert disk, then press any ke
y."):: C=231 :: GOSUB 210 !2
19
390 IF M=11 THEN GOSUB 200 :
: GOTO 350 !179
400 IF P=0 THEN GOSUB 190 ::
 GOTO 410 ELSE ON ERROR 290
```

(See Page 16)

TML-CATALOG—

(Continued from Page 15)

```
:: OPEN #2:P$ :: ON ERROR ST
OP :: ON P GOSUB 190,200 !18
9
410 ON ERROR 290 :: OPEN #1:
D$,RELATIVE,INTERNAL,INPUT :
: INPUT #1:I$,S,Y,Z :: ON ER
ROR STOP :: F,S,X=0 :: Y=Y+2
 :: J$=I$ :: GOSUB 250 !192
420 I$=J$ :: INPUT #1:J$,I,J
,K :: IF J$>"" THEN GOSUB 25
0 !179
430 CALL LINK("BOX",Q-2,1,Q+
7,240):: CALL LINK("BOX",Q-2
,54,Q+7,103):: CALL LINK("BO
X",Q-2,152,Q+7,201):: CALL L
INK("PRINT",Q,3,I$)!077
440 CALL LINK("PRINT",Q,56,"
Fmtd "&STR$(Y)):: CALL LINK(
"PRINT",Q,105,"Used "&STR$(Y
-Z)):: CALL LINK("PRINT",Q,1
54,"Avbl "&STR$(Z))!225
450 X=X+1 :: CALL LINK("PRIN
T",Q,203,SEG$("FlsPage "&STR
$(X),1-3*(P<2),3-4*(P<2))):
 IF P=2 THEN 470 !172
460 IF J$="" THEN ON P+1 GOT
O 600,470 ELSE CALL LINK("BO
X",12,1,21,119):: CALL LINK(
"PRINT",14,2,T$&"P"):: IF P=
0 THEN 500 !108
470 IF S THEN 500 ELSE PRINT
 #2:TAB(26);CHR$(14);"TML DI
SK CATALOG":CHR$(27);"~1";CH
R$(27);"1";CHR$(27):TAB(10-L
EN(I$)/2);"Diskname: ";I$: :
!080
480 PRINT #2:"Fmtd:";STR$(Y)
;",";TAB(11);"Usd:";STR$(Y-Z
);",";TAB(20);"Avbl:";STR$(Z
):: S=1 :: IF J$="" THEN 600
 !085
490 PRINT #2: :T$;"     P":L
$ !012
500 FOR L=1 TO 46 :: IF L=24
 AND P<2 AND J$>"" THEN CALL
 LINK("BOX",12,122,21,240)::
 CALL LINK("PRINT",14,123,T$
&"P")!072
510 R=L*7+16+161*(L>23):: C=
2-121*(L>23):: IF J$>"" THEN
 A=LEN(STR$(K)):: B=ABS(I)::
 F=F+1 :: M=LEN(STR$(J)):: O
N P+1 GOTO 530,530,560 !221
520 IF P<2 AND L<>24 THEN L=
46 :: V=2*(R=23):: GOSUB 270
 :: GOTO 590 ELSE L=46 :: GO
TO 590 !020
530 K$=J$&RPT$(" ",15-LEN(J$
)-M)&STR$(J)&" "&SEG$("DFDVI
FIVPr",B*2-1,2)&SEG$("ogr"&S
TR$(K)&"  ",-3*(B<5)+1,3)&SE
G$(" P U",SGN(I)+2,2)!076
540 CALL LINK("PRINT",(R),(C
),K$):: IF L<>23 AND L<>46 T
HEN ON P+1 GOTO 580,560 !167
550 V=7 :: GOSUB 270 :: IF P
=0 THEN 580 !193
560 PRINT #2:J$;RPT$(".",10-
LEN(J$));" ";RPT$(".",4-M);S
TR$(J);" ";SEG$("Dis/FixDis/
VarInt/FixInt/VarProgram",B*
7-6,7);!193
570 K$=SEG$(" P U",SGN(I)+2,
2):: IF B<5 THEN PRINT #2:ST
R$(K);RPT$(".",3-A);K$ ELSE
PRINT #2:"...";K$ !101
580 INPUT #1:J$,I,J,K :: IF
J$>"" THEN GOSUB 250 !152
590 NEXT L !226
600 IF J$="" THEN CLOSE #1 :
: CALL LINK("PE"):: CALL LIN
K("FILL",Q,204,Q+6,238):: CA
LL LINK("PD"):: CALL LINK("P
RINT",Q,204,"Fls "&STR$(F))!
244
610 IF P AND J$="" THEN PRIN
T #2: :TAB(10);F;"Files":CHR
$(27);"~0":CHR$(27);"1";CHR$
(0): : : : :: CLOSE #2 ELSE
IF P=2 AND J$>"" THEN 500 !1
64
620 GOSUB 240 :: IF J$>"" TH
EN CALL LINK("PRINT",186,46,
"Press any key for next page
"):: C=191 :: GOSUB 210 :: G
OSUB 190 :: GOTO 430 !095
630 CALL LINK("PRINT",186,48
,"Catalog another disk? (Y/N
)"):: K$=RPT$("Y"&C$&"N"&C$,
2):: R=186 :: C=188 !216
640 GOSUB 220 :: IF M=89 THE
N IF P=2 THEN GOSUB 200 :: G
OTO 340 ELSE GOSUB 190 :: GO
TO 330 ELSE IF M<>78 THEN GO
SUB 230 :: GOTO 640 ELSE 280
 !244
```

Getting more out of your printer

Using control codes with TI-Writer to handle styles and formatting

By COL CHRISTENSEN

The following article appeared as part of a series about word processing with TI-Writer in TIsHUG News Digest of Sydney, New South Wales.—Ed.

One aspect of word processing that needs more coverage is the use of printer control code sequences to manipulate the various printer functions. These code sequences range in length from just one character to a great number. The most common sequences, however, are from 1 to 3 characters.

When the printer receives a valid single control character, the character is removed from the text string and the text string is acted upon according to the function that it controls. In a sequence, the first control character will usually be the character 27 and it will be followed by one or more other characters. The control character 27, usually referred to as ESCape, will be removed as well as a predetermined number of other characters. The number depends on what the second character is.

As a matter of interest, the control characters from zero to 31 in value were named in the days of early systems of electronic text communication and became more evident in the days of teletype transmissions. Some of the mnemonics that generally depict the function of the characters are:

ASCII	MNE	Meaning
2	STX	Start of text
3	ETX	End of text

(See Page 17)

CONTROL CODES—

(Continued from Page 16)

6	ACK	Acknowledge
7	BEL	Ding-a-ling
8	BS	Back space
12	FF	Form feed
14	SO	Shift out
20	DC4	Device control 4
27	ESC	Escape

PRINTER CONTROL CODES

Printer control sequences may be grouped into the following categories:

1. Text mode settings
2. Print position
3. Page formatting
4. Graphic bit imaging
5. Downloading characters
6. Printer status

For each of these groups the printer functions and control code sequences are given as both mnemonics and their equivalent characters.

TEXT MODE SETTINGS

Function	Mnemonic	Characters
Expanded printing for one line	SO	14
Cancel expanded	DC4	20
Begin condensed mode	SI	15
Cancel condensed	DC2	18
Begin pica size	DC2	18
Begin elite size	ESC :	27 58
or	ESC M	27 77
Begin italics	ESC 4	27 52
Roman print	ESC 5	27 53
Begin emphasized	ESC E	27 70
Cancel emphasized	ESC F	27 71
Begin underlining	ESC - 1	27 45 1
Stop underlining	ESC - 0	27 45 0
Begin double strike or near letter quality	ESC G	27 72
Cancel above	ESC H	27 73
Begin subscript	ESC S 1	27 83 1
Begin superscript	ESC S 0	27 83 0
Cancel either	ESC T	27 84
Begin expanded print	ESC W 1	27 87 1
Cancel expanded	ESC W 0	27 87 0
Begin overlining	ESC _ 1	27 95 1
Cancel overlining	ESC _ 0	27 95 0

Some of the different text modes can be combined, as you probably have noticed in Fig. 1. For convenience, those that will combine are more easily seen if the codes are put into sub-groups. You may use any code from one of the sub-groups with one from any or each of the others. But don't be disappointed if some discrepancies occur. Your printer manual may list some of the restrictions and which ones have priority over others.

1. Expanded
2. Pica, elite condensed
3. Subscript, superscript
4. Italics, Roman
5. Emphasized or not
6. Underlined
7. Overlined
8. Double strike (NLQ) or not

PRINT POSITIONING

Function	Mnemonic Characters	
Line feed	LF	10
Form feed	FF	12
Carriage return	CR	13
Set perforation skip	ESC N n	27 78 n
Cancel perf skip	ESC O	27 79
Set margins	ESC X m n	27 88 m n

The perforation skip refers to the space left unprinted at the perforation of continuous (fan fold) paper. It sets the bottom of the page margin to "n" lines. That is, the distance from the last print line on one page and the first on the next page.

The margins code allows setting of the printer's left margin (m) and the right margin (n). If you want to list a BASIC program in 28 columns just like it appears on the screen, set these two values to 1 and 28, respectively. Or, if you want to print farther over on the page set them to a higher pair of numbers, such as 10 and 38.

PAGE FORMATTING

Function	Mnemonic Characters	
Set 1/8" line spacing	ESC O	27 48
Set 7/72" spacing	ESC 1	27 49
Set 1/6" line spacing	ESC 2	27 50
Set n/216" spacing	ESC 3 n	27 51 n
Set page length lines	ESC C n	27 67 n
Set page length inches	ESC C 0 n	27 67 0 n

The standard line spacing as set on the printer DIP switches is one-sixth of an inch. Double-spaced print may be set with the code ESC 3 H, the H with ASCII 72 making 72/216 of an inch, or twice one-sixth of an inch. One-eighth of an inch line spacing may be used for listings if you want to conserve paper.

GRAPHIC BIT IMAGING

These codes are used by such programs as TI-Artist, MY-Art and Page Pro in their output of graphic designs to paper. The complications of their use is beyond the scope of this article. Briefly put, the picture data is sent out, row by row, in streams of bytes each representing eight vertical dots of the picture. The eight dots are converted to data numbers in much the same way as in redefining a character in a CALL CHAR statement in BASIC.

Most printers already have a set of graphic characters restricted to horizontals and verticals that could be suitable for doing line drawing. Their ASCII values are usually greater than 128, which means they are not readily available for word processing. They may be printed, however, by using transliteration like those below that suit my printer. Change the graphics character numbers to suit your own printer and remember to place a CR symbol immediately after the actual TL.

```
.CO Use CTRL/U & SHIFT
.TL 1:218 A TL CORNER
```

(See Page 18)

CONTROL CODES—

(Continued from Page 17)

.TL 4 :195 D L INTERSECTION
.TL 5:197 E CROSS
.TL 6:180 F R INTERSECTION
.TL 7:192 G BL CORNER
.TL 8:193 H B INTERSECTION
.TL 9:217 I BR CORNER
.TL 16:196 P HORIZ LINE
.TL 17:179 Q VERT LINE

DOWNLOADING

Redefined characters can be downloaded to many printers using code from this group. Another code allows selection of the downloaded character set or the standard set for printing. This data transfer sends bytes which each represent eight vertical dots to be printed. The 8- or 9-pin type of printer would take about 13 bytes to define a character. The more pins your printer has, the more bytes that have to be sent to define a character. For my printer, which has a character definition 48 dots deep and 36 wide in the high quality mode, it takes 224 bytes to redefine just 1 character and 96 disk sectors to house a full set from ASCII 32 to 126.

Modern printers have a range of selectable fonts, such as Courier, Sanserif, Orator and others that give a good variety in output. These may be selected either by using control codes or by push buttons on the printer.

PRINTER STATUS

Function	Mnemonic Characters	
Reset printer	ESC @	27 64
Unidirectional printing	ESC U 1	27 85 1
Bidirectional printing	ESC U 0	27 85 0

Resetting the printer cancels all previously set codes. Unidirectional print is necessary for printing graphics, as variations in the printer's horizontal registration otherwise tend to produce wavy shapes in vertical lines.

THE HEXADECIMAL NUMBER SYSTEM

Let's look at numbers up to 31 in decimal and hexadecimal, the latter usually being prefixed with a > or an "H" or suffixed with an "H." You can apply the principles of decimal notation to hexadecimal. In decimal, the two "houses" are tens and units while in hexadecimal they are 16s and units. So, >1B is equivalent to one 16+B (11 decimal) units totaling 27 decimal.

Decimal	Hex	Decimal	Hex
0	>0	16	>10
1	>1	17	>11
2	>2	18	>12
3	>3	19	>13
4	>4	20	>14
5	>5	21	>15
6	>6	22	>16
7	>7	23	>17
8	>8	24	>18
9	>9	25	>19
10	>A	26	>1A
11	>B	27	>1B
12	>C	28	>1C
13	>D	29	>1D
14	>E	30	>1E
15	>F	31	>1F

(See Page 19)

1993 TI FAIRS

APRIL

Northeast TI Fair, April 17, Waltham High School, Waltham, Massachusetts. Contact Ron Williams, 14 East St., Avon, MA 02322.

Canadian TI Fest, April 24, Merivale High School, Nepean, Ontario, Canada. Contact Bill Gard, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6 or (613) 523-9396 or Fax (819) 997-2194 Attn: DMES 2.

MAY

Lima Multi User Group Conference, May 14-15, Ohio State University Lima Campus, Lima, Ohio. Contact Dave Szippl, 4191 Patterson Haplin, Sidney, OH 45365; phone (513) 498-9713 (evenings).

Fourth Annual TI Orphans Reunion, May 15, Zurich Insurance Claims Centre, 9715 Ottewell Rd., Edmonton, Alberta, Canada. Contact Ron Hohman, (403) 456-0862.

SEPTEMBER

Western Washington TI Fair, Sept 18, Tacoma Waterworks, 3506 S. 35th, Tacoma, Washington. Jim Tompkins, (206) 756-0934.

OCTOBER

Annual International TI-Faire, Oct. 8-10, Evangelisches Ferienwaldheim Weidachtal, 7000 Stuttgart 80 (Mörhingen), Weidach Gewann 8, Germany. Contact Hans Huben, Berberitzenweg 6, 7033 Herrenberg, Germany; Wolfgang Bertsch, Helenenburgweg 61, 7120 Bietiigheim-Biss, German; or Dierk Warburg, Lilienweg 12, 7141 Benningen, Germany.

Chicago International World Faire, Oct. 30, Holiday Inn, Gurnee, Illinois. Contact Cecure Electronics, P.O. Box 132, Muskego, WI 53150, or Don Walden, (414) 679-4343.

Milwaukee TI Fair, Oct. 31, Quality Inn, 5311 Howell Ave, Milwaukee, Wisconsin. Contact Gene Hitz, 4121 North Glenway, Wauwatosa, WI 53222.

1994 FAIRS

FEBRUARY

Fest-West, Feb. 19-20,Santa Rita Park Inn, Tucson, Arizona. Contact Tom Wills, Fest-West '94 Committee, Southwest 99ers Users Group, P.O. Box 17831, Tucson , AZ 85731 or (602) 886-2460; BJ Mathis, (602) 747-5046; or the Cactus Patch BBS, (602) 290-6277.

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

CONTROL CODES—

(Continued from Page 18)

In the CTRL-U mode, type the uppercase alphabet from A to Z while watching each resultant character on the screen. Which keypress produced a tiny "1" on the screen? Of course, the "A," the first letter of the alphabet. And which key put a tiny "9" on the screen? Right again, the ninth letter of the alphabet, "I". But when we get to the "Z," that's only the 26th. How do we get the others up to 31? Look up an ASCII table to find the next characters. The one after "Z" is the 27th, a "[".What keys do you press to type "["? The magical FCTN-R, of course, which gives us the ESC character, 27, as we already knew but didn't know why.

One other number is missing, the zero. It surely comes before 1, which is obtained with Shift-A. The ASCII table shows the "A" character is preceded by "@", so that's it. Play with them for a while if it's still a little hazy. Notice that you may type your own CR, LF and FF symbols on the screen.

USING THE CODES

You may use printer control codes in your text whether you intend to process it through the formatter or just print it straight out from the editor using the P(rint)F(ile) command. As I may have said in an earlier article, it is preferable to use transliterates when using the formatter as these tend to keep the adjusted right margin more even.

Now comes the process of typing control code sequences into your text. You will have to refer to both the mnemonic and the characters listed for that particular function.

First, we'll take the code to set the printer to subscript, i.e. ESC S 1 with the characters 27 83 1. As you know, we need the CTRL-U mode to type characters with an ASCII value of less than 32. Starting at ESC, which is character 27 and should be displayed on the screen as a small "B" with a little vertical dash before it, press CTRL-U and FCTN-R and the character will appear then CTRL-U again to get back to a normal cursor. Now, besides that, we need an "S" typed in the normal way. Just press Shift-S. Next comes the character "1". We need CTRL-U again for this one, and Shift-comma, and CTRL-U again.

Your code on the screen should read dash-B S tiny-1 to represent ESC S 1. Anything typed after that will print out in subscript style. You do not want all of your text to the be in subscript, so you must cancel that style somewhere. Looking up the codes table, we find the code to effect that is ESC T. To get it, press CTRL-U, FCTN-R, CTRL-U, Shift-T. Got it? Now you should be able to handle the 27 keystrokes and 15 typed characters needed to be able to print the chemical formula for battery acid, which is H_2SO_4.

If you can manage that exercise, you can class yourself capable of mastering control of your printer. Try to use other printer codes to bring your printer's hidden talents to the light of day. I know, you will have to refer to the code lists above or your printer manual from time to time. Who doesn't? What I find helpful are little lists stuck here and there over my console and, guess what, they all refer to printer codes. Without them as ready references, I might not bother to use printer codes. I would get by without making the printer do what I would like it to and never be really satisfied with the result.

I hope by now that some of the fog surrounding the use of printer codes is beginning to clear and a ray or two of golden sunshine is beginning to peep through.

Biorhythms

Let your computer do your chart

The following program was written by John Simpkins. It includes the main biorhythm program, a second LOAD program and a th ird documentation file. The doc file should be input using a word processor and saved in D/V80 format. The program requires Extended BASIC, a disk system if the LOAD and doc files are used, and a memory expansion. A printer is also required. For more information about running this program, see the BIO/DOC listing at the end of the programs.

BIORHYTHMS

```
100 REM TI-O-RHYTHM BY J SIM
PKINS VER.1.5 9/15/91 XBASIC
 !191
110 CALL INIT :: CALL LOAD(-
31878,0)!091
120 OPEN #1:"PIO" !253
130 CALL SCREEN(15)!200
140 CALL CLEAR !209
150 PRINT "         TI-O-RHY
THM": : : :!141
160 PRINT "    BIORHYTHM CAL
CULATOR": : : : : : : :!099
170 MA$="DAY" !024
180 FLAG=0 !209
190 X9=0 !072
200 DIM X$(31)!161
210 DIM M$(12)!149
220 GOSUB 2870 !145
230 P9=6.283185 !182
240 P1=23 !110
250 P2=28 !116
260 P3=33 !113
270 D1=P9/P1 !001
280 D2=P9/P2 !003
290 D3=P9/P3 !005
300 DATA 31,28,31,30,31,30,3
1,31,30,31,30,31 !110
310 CALL SCREEN(12)!197
320 INPUT "Enter Name: ":NAM
E$ !088
330 PRINT :"Enter M for MONT
H or":"      Y FOR YEAR";!08
6
340 INPUT CHART$ !040
350 IF (CHART$<>"M")THEN IF
(CHART$<>"Y")THEN 340 !192
360 N1=0 !054
370 PRINT :"ENTER BIRTH DATE
 MO,DA,YR - example (2,1,194
7) :" !149
380 INPUT MO,DA,YR !196
390 GOTO 2120 !159
```

(See Page 20)

BIORHYTHMS—

(Continued from Page 19)

```
400 IF YR<1900 THEN 2660 !10
7
410 IF MO>2 THEN 460 !033
420 IF MO<>2 THEN 440 !204
430 IF MO=2 AND DA=29 THEN 4
60 !082
440 IF INT((YR-1900)/4)<>(YR
-1900)/4 THEN 460 !231
450 N1=1 !055
460 PRINT : :"ENTER START MO
NTH, YEAR     example (7,199
1):" !183
470 INPUT C1,C3 !047
480 COL=8 !157
490 ROW=13 !228
500 CALL CLEAR !209
510 MSG$="WORKING...." !116
520 FOR MSG=0 TO LEN(MSG$)-1
 :: CALL HCHAR(ROW,COL+MSG,A
SC(SEG$(MSG$,MSG+1,1))):: NE
XT MSG !033
530 IF YR>=C3 THEN 2680 !030
540 RESTORE 300 !138
550 FOR J=1 TO MO :: READ X
:: NEXT J !174
560 N1=N1+X-DA !028
570 IF MO=12 THEN 600 !222
580 RESTORE 300 !138
590 FOR J=MO+1 TO 12 :: READ
 X :: N1=N1+X :: NEXT J !243
600 IF C3-YR<2 THEN 650 !038
610 FOR J=YR-1899 TO C3-1901
 !040
620 IF INT(J/4)=J/4 THEN 270
0 !163
630 N1=N1+365 !230
640 NEXT J !224
650 RESTORE !148
660 IF C1=1 THEN 680 !211
670 FOR J=1 TO C1-1 :: READ
X :: N1=N1+X :: NEXT J !153
680 IF INT((C3-1900)/4)<>(C3
/4)THEN 700 !022
690 IF C1>2 THEN N1=N1+1 !21
8
700 I1=N1 !183
710 I2=N1 !184
720 I3=N1 !185
730 READ X !239
740 GOSUB 1380 !185
750 PRINT : : : : :!037
760 PRINT #1:TAB(27);"*** B
I O R H Y T H M ***" !106
770 PRINT #1: : : : : :!209
780 PRINT #1:"BIOCHART FOR -
- ";NAME$ !005
790 PRINT #1:"BIRTHDATE -- "
;MO;"";DA;"";YR !043
800 PRINT #1:"You were born
on a ";!085
810 GOSUB 1420 !225
820 PRINT #1:"Julian day =";
Z9: :!234
830 GOSUB 2170 !210
840 PRINT #1:"Your BIRTHSIGN
 is ";Q$ !223
850 PRINT #1: : :!178
860 PRINT #1:" P = Physical"
 !070
870 PRINT #1:" E = Emotional
" !167
880 PRINT #1:" M = Intellect
ual" !240
890 PRINT #1:" * = Intersect
ion of two or more cycles.":
 : :!040
900 PRINT "press 'ENTER' to
continue" !029
910 INPUT DMY$ !160
920 CALL CLEAR !209
930 L=0 !003
940 GOSUB 1290 !095
950 D=0 !251
960 L=L+1 !017
970 FOR I=1 TO 31 :: X$(I)="
 " :: NEXT I !039
980 X$(12)="|" !024
990 Y1=INT(11*SIN((L+I1)*D1)
+12.5)!049
1000 Y2=INT(11*SIN((L+I2)*D2
)+12.5)!052
1010 Y3=INT(11*SIN((L+I3)*D3
)+12.5)!055
1020 X$(Y1)="P" !073
1030 X$(Y2)="E" !063
1040 X$(Y3)="M" !072
1050 IF Y1=Y2 THEN 2740 !142
1060 IF Y1=Y3 THEN 2760 !163
1070 IF Y2=Y3 THEN 2780 !184
1080 D=D+1 !001
1090 IF D<X+1 THEN 1230 !229
1100 S1=S1+1 !129
1110 IF S1=12 THEN 2800 !102
1120 C1=C1+1 !097
1130 IF C1>12 THEN 1190 !008
1140 READ X !239
1150 GOSUB 1380 !185
1160 PRINT #1:CHR$(12)!184
1170 GOSUB 1290 !095
1180 GOTO 1230 !033
1190 RESTORE !148
1200 C1=1 !044
1210 C3=C3+1 !101
1220 GOTO 1140 !199
1230 PRINT #1:D;TAB(20);!137
1240 FOR J=1 TO 31 !109
1250 PRINT #1:X$(J);!047
1260 NEXT J !224
1270 PRINT #1:!072
1280 GOTO 960 !018
1290 IF X9=1 THEN 2800 !064
1300 IF CHART$="M" THEN 2720
 !016
1310 CALL SCREEN(6)!151
1320 PRINT "CALCULATING CHAR
T & SENDING TO PRINTER...":
:"PLEASE WAIT.....": : : : :
 : : : : :!140
1330 PRINT #1:"^^ BIOCHART F
OR ";M$(C1);" ";C3 !058
1340 PRINT #1:TAB(5);NAME$ !
168
1350 PRINT #1:TAB(19);"(-)
     (0)        (+)":!202
1360 D=1 !252
1370 RETURN !136
1380 IF X<>28 THEN 1410 !143
1390 IF INT(C3/4)<>C3/4 THEN
 1410 !173
1400 X=X+1 !041
1410 RETURN !136
1420 Z1=MO !229
1430 Z2=DA !207
1440 Z3=YR !246
1450 Z9=0 !074
1460 IF Z1=1 THEN 1680 !214
1470 Z9=Z9+31 !211
1480 IF Z1=2 THEN 1680 !215
1490 Z9=Z9+28 !217
1500 IF Z1=3 THEN 1680 !216
1510 Z9=Z9+31 !211
1520 IF Z1=4 THEN 1680 !217
1530 Z9=Z9+30 !210
1540 IF Z1=5 THEN 1680 !218
1550 Z9=Z9+31 !211
1560 IF Z1=6 THEN 1680 !219
1570 Z9=Z9+30 !210
1580 IF Z1=7 THEN 1680 !220
1590 Z9=Z9+31 !211
1600 IF Z1=8 THEN 1680 !221
1610 Z9=Z9+31 !211
1620 IF Z1=9 THEN 1680 !222
1630 Z9=Z9+30 !210
```

(See Page 21)

BIORHYTHMS—

(Continued from Page 20)

```
1640 IF Z1=10 THEN 1680 !007
1650 Z9=Z9+31 !211
1660 IF Z1=11 THEN 1680 !008
1670 Z9=Z9+30 !210
1680 Z9=Z9+Z2 !049
1690 IF INT(Z3/4)=Z3/4 THEN
1900 !006
1700 Z4=Z3-1900 !049
1710 Z4=Z4*365.25 !158
1720 Z4=Z4+Z9 !046
1730 IF INT(Z4)=Z4 THEN 1940
 !174
1740 Z5=Z4/7 !159
1750 Z6=INT(Z5)!025
1760 Z7=Z5-Z6 !048
1770 IF Z7=0 THEN 1980 !008
1780 Z7=(Z7*7)+.11 !044
1790 Z8=INT(Z7)!029
1800 IF Z8=1 THEN 2000 !030
1810 IF Z8=2 THEN 2020 !051
1820 IF Z8=3 THEN 2040 !072
1830 IF Z8=4 THEN 2060 !094
1840 IF Z8=5 THEN 2080 !115
1850 IF Z8=6 THEN 2100 !136
1860 IF Z8=0 THEN 1980 !009
1870 IF Z8>6 THEN 1960 !253
1880 PRINT #1: : :!178
1890 RETURN !136
1900 IF Z1=1 THEN 1700 !234
1910 IF Z1=2 THEN 1700 !235
1920 Z9=Z9+1 !159
1930 GOTO 1700 !249
1940 Z4=Z4-1 !150
1950 GOTO 1740 !033
1960 PRINT "(CALCULATION ERR
OR!)" !194
1970 GOTO 1880 !174
1980 PRINT #1:" SUNDAY" !010
1990 GOTO 1880 !174
2000 PRINT #1:" MONDAY" !254
2010 GOTO 1880 !174
2020 PRINT #1:" TUESDAY" !08
6
2030 GOTO 1880 !174
2040 PRINT #1:" WEDNESDAY" !
221
2050 GOTO 1880 !174
2060 PRINT #1:" THURSDAY" !1
72
2070 GOTO 1880 !174
2080 PRINT #1:" FRIDAY" !245
2090 GOTO 1880 !174
2100 PRINT #1:" SATURDAY" !1
65
2110 GOTO 1880 !174
2120 IF MO>12 THEN 2150 !244
2130 IF DA>31 THEN 2150 !222
2140 GOTO 400 !224
2150 PRINT "Check Your Dates
...":"They do not compute...
":!170
2160 GOTO 370 !194
2170 IF MO>1 THEN 2230 !017
2180 IF DA<21 THEN 2210 !024
2190 Q$=" AQUARIUS" !142
2200 RETURN !136
2210 Q$=" CAPRICORN" !197
2220 RETURN.!136
2230 IF MO>2 THEN 2270 !058
2240 IF DA<20 THEN 2190 !003
2250 Q$=" PISCES" !232
2260 RETURN !136
2270 IF MO>3 THEN 2310 !100
2280 IF DA<21 THEN 2250 !064
2290 Q$=" ARIES" !148
2300 RETURN !136
2310 IF MO>4 THEN 2350 !141
2320 IF DA<20 THEN 2290 !103
2330 Q$=" TAURUS" !005
2340 RETURN !136
2350 IF MO>5 THEN 2390 !182
2360 IF DA<21 THEN 2330 !145
2370 Q$=" GEMINI" !218
2380 RETURN !136
2390 IF MO>6 THEN 2430 !223
2400 IF DA<22 THEN 2370 !186
2410 Q$=" CANCER" !205
2420 RETURN !136
2430 IF MO>7 THEN 2470 !008
2440 IF DA<22 THEN 2410 !226
2450 Q$=" LEO" !254
2460 RETURN !136
2470 IF MO>8 THEN 2510 !049
2480 IF DA<22 THEN 2450 !010
2490 Q$=" VIRGO" !167
2500 RETURN !136
2510 IF MO>9 THEN 2550 !090
2520 IF DA<23 THEN 2490 !051
2530 Q$=" LIBRA" !138
2540 RETURN !136
2550 IF MO>10 THEN 2590 !172
2560 IF DA<23 THEN 2530 !091
2570 Q$=" SCORPIO" !065
2580 RETURN !136
2590 IF MO>11 THEN 2630 !213
2600 IF DA<22 THEN 2570 !131
2610 Q$=" SAGITTARIUS" !118
2620 RETURN !136
2630 IF DA<22 THEN 2610 !171
2640 Q$=" CAPRICORN" !197
2650 RETURN !136
2660 PRINT :"YEAR MUST BE 19
00 OR LATER" !182
2670 GOTO 370 !194
2680 PRINT :"START YEAR MUST
 BE GREATER":"THAN BIRTH YEA
R!!" !047
2690 GOTO 460 !028
2700 N1=N1+1 !119
2710 GOTO 630 !199
2720 X9=1 !073
2730 GOTO 1310 !114
2740 X$(Y1)="*" !035
2750 GOTO 1060 !119
2760 X$(Y1)="*" !035
2770 GOTO 1070 !129
2780 X$(Y3)="*" !037
2790 GOTO 1080 !139
2800 PRINT "DO ANOTHER (Y/N)
?" !190
2810 CALL KEY(0,K,S):: IF S=
0 THEN 2810 !072
2820 IF (K=78)+(K=110)THEN 2
850 !109
2830 IF (K=89)+(K=121)THEN C
ALL CLEAR !076
2840 GOTO 170 !249
2850 CLOSE #1 !151
2860 END !139
2870 RESTORE 2900 !188
2880 FOR J=1 TO 12 :: READ M
$(J):: NEXT J !015
2890 RETURN !136
2900 DATA JANUARY,FEBRUARY,M
ARCH,APRIL,MAY,JUNE,JULY,AUG
UST,SEPTEMBER,OCTOBER,NOVEMB
ER,DECEMBER !203
```

LOAD

```
100 CALL CLEAR :: CALL SCREE
N(14)!026
110 DISPLAY AT(4,5):"CHOOSE:
" :: DISPLAY AT(7,7):"1 - RU
N BIORHYTHM PROGRAM" :: DISP
LAY AT(10,7):"2 - REVIEW DOC
UMENTATION" !243
120 DISPLAY AT(13,7):"3 - EX
IT TO EXTENDED BASIC" !006
130 ACCEPT AT(24,25)VALIDATE
("123")SIZE(-1)BEEP:A !216
140 ON A GOTO 150,180,240 !0
93
```

(See Page 22)

BIORHYTHMS—

(Continued from Page 21)

```
150 CALL CLEAR :: CALL SCREE
N(5)!233
160 DISPLAY AT(15,1):"LOADIN
G BIORHYTHM PROGRAM..." !010
170 RUN "DSK1.BIORHYTHM" !11
7
180 CALL CLEAR :: CALL SCREE
N(12)!024
190 DISPLAY AT(8,1):"THE DOC
UMENTATION FOR THIS":"PROGRA
M CAN BE VIEWED AND":"EDITED
 WITH TI-WRITER, ETC.":"JUST
 LOAD THE FILE NAMED:" !116
200 DISPLAY AT(17,9):"'BIO/D
OCS'" !236
210 DISPLAY AT(24,1):"PRESS
ANY KEY TO CONTINUE" !087
220 CALL KEY(3,K,S):: IF S=0
 THEN 220 !035
230 GOTO 100 !179
240 CALL CLEAR :: CALL SCREE
N(13)!025
250 DISPLAY AT(10,1):"EXITIN
G TO EXTENDED BASIC..." !199
260 FOR D=1 TO 750 :: NEXT D
 !252
270 END !139
```

BIO/DOCS

Biorhythms are the rhythmic cycles of the life process of an individual. There are three elements which make up a biorhythm chart. They are a 23 day physical cycle, a 28 day emotional cycle, and a 33 day mental or intellectual cycle. According to those who accept the theory, biorhythms begin at birth and continue with regularity until death. On the graph, the median line (0) is the critical point and is when most accidents occur. The high periods (+) are when you have the most energy, and the low periods (-) are regarded as recuperative periods.

This program will chart your daily biorhythmic patterns a month at a time, or you may choose to have an entire year done at once. As a bonus, it will also tell you what day of the week you were born on, and give your astrological sign.

Although most scientists feel there is no actual proof that the theory really works, give it a try for yourself and see what you find. You may be surprised!

Dutch user group changes address

The Dutch TI Users have changed their address, according to Erik van Wette of the group.

New address is Texas Instruments Gebruikers Groep, H.B. Blijdensteinlaan 24, 7514 CB Enschede, The Netherlands. Telephone is (31) 53-339887.

Put a mouse on your lap

A new product has been developed which may be of interest to Geneve 9640 users or to TI99/4A users ith Rave 99 keyboards.

The MousLounge Keyboard Caddy from Armchair General is a keyboard-mounted side-tray for keeping a mouse, trackball or joystick at hand. The adjustable device can clamp on to keyboards for right or lefthand use, or it can be placed on the user's lap. It lists from $19.95 from Armchair General, P.O. Box 2211, Twin Falls, ID 83303, (203) 733-0700 (voice) or (208) 733-9316 (fax).

Catch a TI fair in September or October
see the listing on page 18

Watchamacalit

An accurate clock for the HFDC

By DAN H. EICHER

The Watchamacalit (WCMCI) is a hardware upgrade device for use with the Myarc Hard Floppy Disk Controller (HFDC). The WCMCI was designed by David DeHeer and produced in cooperation with the Ogden TI Users Group.

As anyone who owns an HFDC has discovered, the clock on the HFDC needs to be set every time the TI-99/4A is powered on. Even then its time keeping accuracy (while varying from unit to unit) tends to be very poor. This was very surprising, considering that earlier Myarc had produced a clock based upon the same chip for use in the Geneve computer. That implementation was fairly accurate and battery backed. Why Myarc took a step back when designing the HFDC is a mystery to all.

(See Page 23)

REVIEW

WATCHAMACALIT—

(Continued from Page 22)

What does the WCMCI do? The WCMCI battery backs up the clock on the HFDC. It also gives the user a way of adjusting the speed with which the clock runs to insure ongoing accuracy.

The WCMCI is a daughter board that plugs into the socket which currently holds the chip labeled MM58274. (A "daughter board" is a board that plugs in to an existing board to augment the main board's functionality.) The installation consists of desoldering (or cutting out, whichever you prefer) one component, then removing the MM58274 from its socket on the HFDC, plugging it into the socket provided on the WCMCI board, and finally plugging the WCMCI into the just vacated socket. That's it! When you install this card, you will notice something really interesting — surface mounted components on a card for the TI.

Everything you need is included, except for the battery. The documentation for this device is excellent. The documentation discuss the history of this project, how to install the device and theory of operation of the WCMCI board. The manual consists of about ten pages of single spaced documentation in all.

The software presents somewhat of a problem. The software provided works fine in the 99/4A environment, but does not run properly, as is, with the Geneve. The problem is that the Geneve does not use the DSR provided by the HFDC, instead it uses a device driver built into MDOS. The driver built into MDOS does not appear to support OPEN "TIME" as it is supposed to. This problem was overcome by running a utility called ROMPage that "pages in" the HFDC's built in DSR. After running ROMPage, everything appeared to run smoothly.

Upon contacting the author about this problem, he said that the device was not really intended for use with the Geneve since it had its own working, built-in time clock. Fair enough.

Another feature that would have been handy to have included in the software is the ability to time the running of your HFDC clock against a known standard so you could immediately adjust the running speed of the HFDC's time clock. This could have been accomplished by timing the clock chip against either the 9901 count down timer or the VDP interrupt timer.

My only hesitation about recommending this product is the fact that after installation, you cannot put your clamshell back on the card (it sticks out too far). In fact, I was not able to install the HFDC in a Peripheral Expansion Box slot next to another card in a clamshell. To be fair, with a daughter board on a PEB card, that was rather unavoidable.

All in all, I would recommend without reservations this upgrade. The hardware design and documentation are among the best I have seen.

The price is approximately $35.

For further information contact. David DeHeer via the Salt Lake Flats BBS (8N1), 24 hours, 300/1200/2400, 317-392-2312; or Ricky Bottoms Enterprise, 625 N. Pike St., Shelbyville, IN 46176, 801-394-0064 from 6 to 9 p.m.

MICRO-REVIEWS

Mail Room and Time Calc

By CHARLES GOOD

Some of the software I review is shareware or public domain. In order that my readers can have ready access to this software I will sometimes offer to directly distribute some of these products (such as TIME CALC reviewed in this column) if you send me $1. Your money pays for a disk and postage, and the rest goes in my Florida vacation fund. This is done strictly as a service to the TI community. I suspect it will be a long time before my "Florida" fund will be large enough to get me to the edge of town. Loan me your products for MicroReviews by sending them to Charles Good, P.O. Box 647, Venedocia OH 45894. My phone is 419-667-3131. All products will eventually be returned

REVIEW

MAIL ROOM

This is Asgard's top of the line "name/address/phone" data base. There are lots of public domain name-and-address programs for the 99/4A written in TI BASIC or Extended BASIC. I have been using one for years to handle the mailing list of my user group's newsletter. All these PD programs suffer from a bad case of the "slows." Most will handle only a limited number of names (usually about 100) because all the names and addresses have to be loaded into memory at the beginning of program execution. Many have problems with foreign addresses because there is no place to put "country" in the name and address data. Although written in Extended BASIC, MAIL ROOM deals effectively with all these problems.

NO FILE SIZE PROBLEM

MAIL ROOM, which was written by Larry Tippett, saves name, address, city, state, zip code, and phone on a disk file whose data is only loaded into memory as needed. Thus, large files are possible. A DSSD disk can hold the program itself and a file of 1,000 addresses, although 500 is the limit if you want MAIL ROOM to sort the file for you. A SSSD disk can hold the software and a file of 388 addresses, which is more than the membership of the largest TI user group.

You can print labels, envelopes (complete with your return address), or just a

(See Page 24)

MICRO-REVIEWS—

(Continued from Page 23)

hard copy list of your data. You can print these things for all the addresses in your list, for just one address, or you can mark specific addresses in the file and print only those addresses. The data file is in INT/FIX 128 format, but MAIL ROOM allows you to convert the file to DIS/VAR 80 for use with the mail merge option of TI-Writer.

NO FOREIGN ADDRESS PROBLEM

There is plenty of room in the "zip code" field for non-standard foreign postal codes and the full name of a country. When printed, the zip code or country name is the last line on the address label, and because it is indented several spaces it stands out very strongly from the rest of the address.

GOOD SPEED FOR XBASIC

Although MAIL ROOM is largely written in XB (and is thus easily modifiable by the user), there is lots of embedded assembly code to speed things up. Sorting in particular is done with a nice fast assembly routine. Prior to printing you can sort your file by any field. If you sort by name the result is alphabetical by last name, except for addresses you specified as "corporation" when the data was entered. Corporations will end up in the sorted list alphabetically by first name. For example, ASGARD SOFTWARE will be with the A's. Speed is noticeably enhanced by using a RAMdisk.

MAIL ROOM is disk intensive, loading different parts of itself into memory from disk and then going to disk to look up parts of its data file. Assembly code can speed up things when already in memory, but can't do much to speed disk access.

OTHER NICE FEATURES

If you have a modem you can display an address, press <D>ial, and MAIL ROOM will dial the phone number for you, assuming you entered the phone number in the telephone field.

The program package comes with two disks containing a 40-column and an 80-column version. The two versions are functionally identical, but the 80-column version really looks great on my AVPC equipped 99/4A system! There are lots of overlapping windows. The 80-column version is written in XB using Alexander Hulpke's X80. In my opinion, everyone should try to upgrade to 80 columns because there is now a nice variety of useful TI/Geneve 80-column applications. These include MAIL ROOM, Funnelweb, Spell It, First Draft, Y.A.P.P., 80-column Multiplan and Telco.

CONCLUSIONS AND COMPLAINTS

I like MAIL ROOM enough that I now use it to handle the newsletter mailing list and the membership list of the Lima User Group. The only other TI software comparable in terms of ease of use, speed, and list size is the RAMBO-specific mail list program (I can't remember its title) written by Gary Bowser.

I do have a couple of complaints about MAIL ROOM, however. The printed documentation does not explain how to set the program up to run off of some drive other than DSK1, and the Configure option from within the software won't do this either. With no hints in the docs, it took me a long time to get MAIL ROOM to run off of my RAMdisk as DSK5. I finally discovered that there are 4 or 5 separate parts of MAIL ROOM that you need to OLD into memory, change the XB code, and then SAVE back to disk. There are REM statements near the beginning (but confusingly not exactly at the beginning) of each program segment that tell you which XB program lines to modify. This procedure should be spelled out in detail in the printed documentation, or at least in a README file on 40- and 80-column program disks. My other gripe is that removing an entire address record (deleting a name) is difficult, though not impossible.

MAIL ROOM requires only a minimal SSSD disk system with 32K. An 80-column device (or Geneve), printer, and modem are supported. It is available for $14.95 plus shipping from most TI dealers and from Asgard Software, 1423 Flagship Dr., Woodbridge VA 22192, phone 703-491-1267.

TIME CALC

The TI community has been greatly enriched by the efforts of Bruce Harrison. Assembly programmers have told me that his assembly column in MICROpendium is first class. Also, he has contributed lots of useful assembly language and assembly/XB hybrid software to the public domain for us to enjoy for free. His latest public domain offering is TIME CALC.

Have you ever tried to do simple math with time? For example, let's say you want to fill up one side of a C60 cassette tape with specific music tracks from your CD or record collection so you can play the music in your car. Each track of music has a time listed in minutes and seconds; 7:25, 2:15, etc. How many tracks can you fit on a 30-minute C60 tape side? Simple: just add up the individual times on a pocket calculator.

Well, it isn't very simple. Just try it some time. The problem is that pocket calculators do math using a base of 10 whereas time has a base of 12, or 24, or 60. TIME CALC does this work for you with an accuracy of one second. TIME CALC does the following using either 12- or 24-hour time:

- **Elapsed Time**: Input start time, input stop time, and you get "elapsed time."
- **Cumulative Sum**: Input a time and either add or subtract it from the running total. This is the option you use to fill up those C60 music tapes. You can keep adding and/or subtracting from this running total indefinitely.
- **Time Multiply or Divide**: Input a time and then divide it by a number such as 3.5 or just 3. For example, 5 hours and 35 minutes divided by 3 equals 1 hour 51 minutes and 40 seconds. Just try that on a calculator!

Inputting time is easy because you can ignore minutes and seconds if you want. 12 and Enter is the same as typing 12/00/00 and Enter. As a comparison, the TIME command in MS-DOS usually requires hours, minutes, and seconds even if the minutes and seconds are zero.

I like TIME CALC enough to keep it on my RAMdisk and listed on my Funnelweb XB menu. You can get a copy by sending me $1 at the address above carefully wrapped in a piece of paper upon which is written your name, address, and a request for TIME CALC.

TIRUG pulls out

After 10 years of existence, the TI Riverside Users Group in Riverside, California, has disbanded. according to Ed Butcher, founder and president.

Hardware project

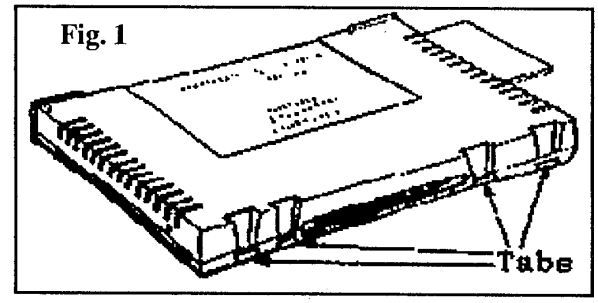
Modifying a Myarc RS232 card

By BRUCE FORBES

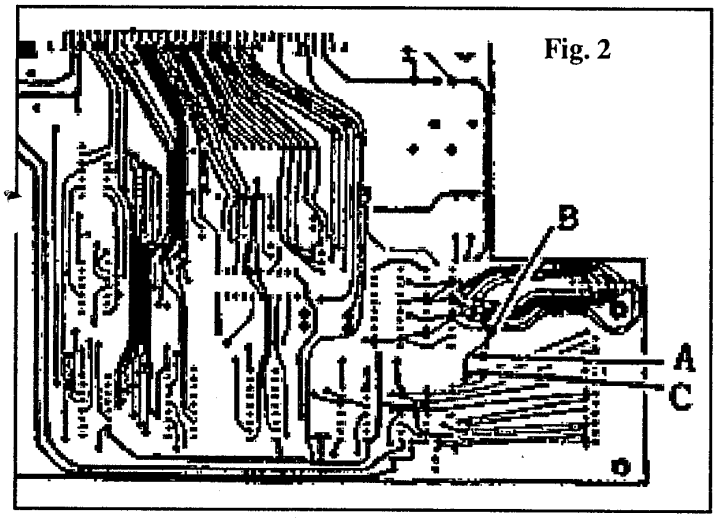
This article originally appeared in several user group newsletters. The reader is entirely liable for the outcome if this project is undertaken.—Ed.

Anyone who has ever written an article for publication knows how hard it can be to put your thoughts on paper in order for them to be informative to the reader. All those members that have taken the time to make contributions to the newsletter should be congratulated for a job well done. There are several of us who have had experiences that would be helpful to other members if we would just take the time to put them in writing. I realize that we do share our experiences verbally, but somehow most of the things we hear seem to get filed away in some deep, dark corner of our brains, seldom to ever surface again. One example comes to my mind; — at one of our 1985 meetings someone told us that a few of the printers on the market had a tendency to put out high spikes on their Acknowledge, Strobe and Busy signals. These spikes could cause some type of problems. However, there was a remedy for it, but I couldn't remember what it was.

During the course of last year I burned out two Myarc RS232 cards. The problem leading up to the eventual burn out of the cards was frustrating and led me to believe that Myarc might have been manufacturing inferior equipment. I even told them so when I sent the card back for repairs the second time. Their silence and the lack of knowledge on behalf of my local supplier left me bewildered as to what was causing my problems.



When I finally located my problem. I was talking with a technician about modifying a second RS232 card so I could run two of them in my PE box (more on this at a later date). During the course of our conversation he described, in painful details, the problems I had been having. It seems that the Panasonic line of printers happens to be one of those printers that puts out high spikes. Eventually these spikes will burn out the RS232 card. Your printer might skew a line, drop a letter or word and even print garbage (sometimes). Eventually the printer will shut down. The next time you use it everything may/may not work fine. These problems might span a week or months before blowing the RS232 interface. It seems there is a simple modification to the RS232 card that will protect it from these spikes and eliminate the printer problems. I will attempt to describe this modification in order for you to protect your Myarc RS232 interface card.

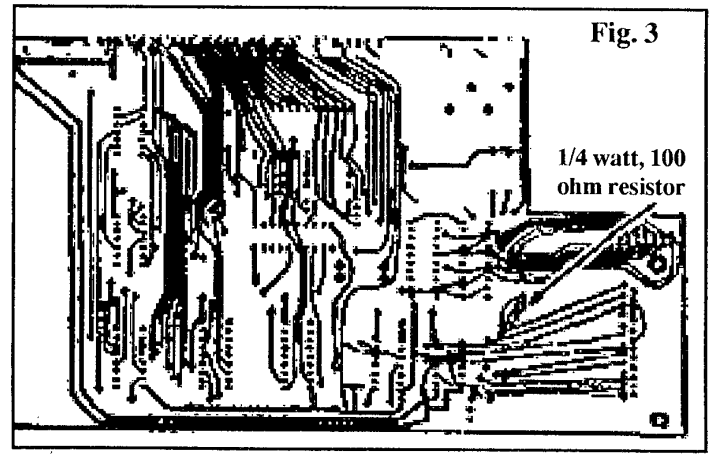


You will need the following:

- needle nose pliers
- wire cutters
- small screwdriver
- solder
- low wattage soldering iron
- a sharp object such as an X-Acto knife or a razor blade
- one 100-ohm, one-quarter watt resistor

Proceed by pulling the RS232 interface card out of the PE box, follow all the safety precautions as described with your equip-

(See Page 26)



USER NOTES

TI-Writer to TI-Base

This comes from Jerry Keisler, of the Paris (Texas) TI99/4A User Group. He writes:

I wrote an article on converting TI-Writer to TI Base files, which appeared in Micropendium, when TI Base was in version 1. I am still receiving phone calls about this
conversion process. TI Base is now in version 3.02 and can easily do the conversion. Version 2.0 has the command ".CONVERT FROMFILE TOFILE GO", which is found on page 3-27 of the version 2 manual. This could have the form of ".CONVERT DSK1.filename DSK2.filename GO". If you only have one drive, leave "GO" off and TI Base will prompt you for the correct disk.

TI-WRITER FILE

The TI-Writer file can take several forms, but it can not exceed 80 columns. Print the file to printer and identify the fields you want to convert to TI Base.

One of my TI-Writer files called BUDGET, has a date field, money field, initial field, and description field with sizes of 8, 9, 3 and 60 characters.

```
DATE      MONEY  INI DESCRIPTION
-------- -------- -- ----------
01/06/93    45.39 FP ----
02/23/93   184.33 FP ----
03/04/93   -23.45 JP ----
```

This file is a list of operating expense and who paid what.

TI BASE CONVERSION

I know the field names and their size so to convert to TI Base, load TI Base and enter the command ".CONVERT DSK1.BUDGET DSK2.BUDGET2 GO".

TI Base wants to know the record information and shows the structure input screen: I completed the structure input screen as follows:

```
NAME     TYPE   SIZE   DEC
DATE       D      8
MONEY      N      9      2
INI        C      3
DESC       C      60
```

Press Function 8 and TI Base will read the BUDGET file and create a BUDGET2/D file using the structure information provided. Each record in BUDGET2/D contains 80 characters.

".USE DSK2.BUDGET2" will open the new TI Base file.

".RECOVER" will build the structure file, BUDGET2/S. The TI-Writer file is now in TI Base format.

RULES OF CONVERSION

1) You can not change the order of the input file (TI-Writer file).

2) You can convert less than the full record. All conversions start from the first character in a record (or a TI-Writer line). The end of a record or TI-Writer line will be truncated. If only DATE, MONEY AND INI in the structure then DESC would be truncated. The TI Base records would be only 20 characters wide.[

3) You can create a TI Base file that is larger than the TI-Writer file. The extra fields will be blank. If DATE, MONEY, INI, DESC, BUDITEM and ACCOUNT are used, in the structure with sizes of 8, 9, 3, 60, 30 and 40, the new TI Base structure will have 150 characters.

This allows you to manually fill in extra data. If you use size 8, 9, 3, 30 and 40, you would have part of DISC in the BUDITEM and ACCOUNT fields. 4) Conversion is not limited to TI-Writer files. Any ordered file can be converted to a TI Base file.

HOW TO SUM

Using TI Base version 2 and the above TI-Writer file, which contains 100 to 300 records, I can SUM all of the MONEY field. I can also do a conditional sum on the MONEY field using the INI field.[

This is done as follows:

Load TI Base.

```
".CONVERT DSK1.BUDGET DSK2.B
UD GO"
NAME   TYPE   SIZE   DEC
DATE     D      8
MONEY    N      9      2
INI      C      3
```

Press Function and 8.

".USE DSK2.BUD"

".RECOVER"

".SUM MONEY" or

".SUM MONEY ;FOR (INI = " FP")" and

".SUM MONEY ;FOR (INI = " JP")"

The total amount of MONEY or the amounts of MONEY for the conditions shown will be displayed to screen. This saves a lot of time on the calculator.

MYARC RS232—

(See Page 26)

(Continued from Page 25)

ment. Before you proceed any further remember that static electricity can damage some semiconductor devices. Drain off any electrostatic charge from your body by touching a known earth ground. Make sure that your work area is free from any static electricity. As an added protection wear a discharge wrist strap device. Now locate the four plastic tabs at the bottom (card edge side) of your RS232 (see Figure 1). Gently press in each of the four tabs while carefully prying the two halves apart. Once you have the card open remove the circuit board and place it on your work surface, component side down. The card edge should be away from you with the plug side to your right (see Figure 2). Locate the boomerang shaped foil track in the lower right corner of the card (see "A" in Figure 2). With your sharp object scrape a path through the foil track to cut the continuity between points B and C (see Figure 2). Your last step is to solder the one-quarter watt, 100 ohm resistor between points B and C (see Figure 3), turn the card over and cut the excess wire from the resistor. Now reassemble your card and the modification is complete.

Please be aware that this modification is tailored to the Myarc RS232 card *only*. If you have a card by any other manufacturer you should contact a *reliable* technician who is familiar with that brand of card. Of course the modification to your card probably will void any warranty left on your card. This is a proven, workable modification and will not interfere with the interface or any peripheral that you might want to

Software control of program recorders

This item appeared in the TIsHUG

(See Page 27)

USER NOTES

(See Page 27)
(Continued from Page 26)

News Digest of Sydney, New South Wales. The program was written by Ed Hall.

The following program will turn your cassette recorder or other device on and off under software control.

```
10 ! program by Ed Hall !000
11 ! to control the !008
12 ! "CS1" remote control !1
23
13 ! put in a music tape !12
7
14 ! plug in the remote !095
15 ! and press keys P and S
!053
16 !!131
17 ! reverse polarity of !04
0
18 ! remote if it will not o
perate your recorder !217
19 !!131
20 !!131
100 CALL INIT !157
110 CALL LOAD(16368,79,70,70
,32,32,32,36,252)!035
120 CALL LOAD(16376,79,78,32
,32,32,32,36,244)!041
130 CALL LOAD(8194,37,4,63,2
40)!041
140 CALL LOAD(9460,2,12,0,45
,29,0,4,91,2,12,0,45,30,0,4,
91,203,78)!034
150 PRINT "PRESS:":" P Play"
:" S Stop" !149
160 CALL KEY(3,A,B)!163
170 IF B<1 THEN 160 !152
180 ON POS("PS",CHR$(A),1)+1
 GOTO 160,190,200 !089
190 CALL LINK("ON"):: GOTO 1
60 !219
200 CALL LINK("OFF"):: GOTO
160 !026
210 END !139
```

This program requires Extended BASIC and a memory expansion. It can operate the cassette player to provide audiovisual tuition, with cassette parts triggered at the appropriate stage by the program.

The remote control is an electronic switch. (For the technically minded, most consoles use a Darlington driver controlled by an optically isolated device.) The polarity of its connection to the remote device matters. If it does not work, try reversing the wires to the 2.5mm plug.

As an electronic device, it uses up some of the voltage available to the remote device, a minimum of one volt. Some devices may be unhappy to have a whole volt removed — use main supply for your recorders if possible an d note that Ni-Cad rechargeable batteries only start with a lower open circuit than other cells. Not too much lower, but if the one volt drop counts, then you should be aware of this.

You can use this program together with a clock, such as that found in the Super Extended BASIC module or Enhanced Display Package, etc., to turn the cassette on and off at specific times. With this program you can turn this device on and off several times a minute all year if you wish! A simple FOR-NEXT loop delay program can be used to time your on-off periods.

Of course, you are not limited to turning just a cassette recorder on and off. You can turn anything on and off, provided you observe the correct polarity and do not try to switch too great a load.

In simple terms, the absolute maximum you can switch is 40V DC at 400mA, but you can exceed that by using a relay, ensuring that you use a diode to protect the computer circuitry. Most consoles use a TIL119 isolator and add to it a TIS92NPN transistor, usually with the collector connected to the tip of the plug.

Corrections to Page Form

This comes from Raymond Frantz, of Phoenix, Arizona. Frantz is president of the VAST 99ers User Group. He writes:

The PAGEFORM program (June 1993) is wonderful. But there are a few errors in the listing. Here are the four lines that I modified:

```
510 CALL KEY(3,K,S):: GOSUB 1
150 :: DISPLAY AT(24,1):"Save
:        (Enter=Exit)" :: A
CCEPT AT(24,6)SIZE(10):P$ ::
IF P$="" THEN 720
640 CALL KEY(3,Z,Y):: GOSUB 1
150 :: DISPLAY AT(24,1):"Load
:        (Enter=Exit)" :: D
ISPLAY AT(24,6)SIZE(-10):E$
960 GOSUB 1140 :: DISPLAY AT(
23,1):" * Device Error! *" :
: GOTO 140
1110 DISPLAY AT(23,1):"Overw
rite enabled." :: CALL HCHAR(
21,31,32,1):: OP=0 :: GOTO 14
0
```

The only remaining problem that I can't find is that FCTN-E does not move the cursor up but it does change the number at the bottom right of the screen.

More on circle segments for TML

This comes from Jim Lesher, of Dallas, Texas. He writes:

Again, these programs can only be used with The Missing Link program. This time, at the third screen prompt, select 2. We will be learning how to manipulate the segments to get the desired effect.

As mentioned in previous User Notes, the numbers in the 8 circle segments are used to blank out that segment and, of course, if we want to blank out 2 segments we simply add the numbers in those 2 segments and plug it into our program. Referring to page 18 of the TML manual, if you want to blank segments 1 and 8, you would type in the number 9.

If this is a bit confusing, maybe the following program will help you determine which numbers to use to achieve the desired pattern. These patterns are most of the basic ones available. There are 255 actual combinations, but most of them are duplications of what you will see in this program but at a different position on the circle. In the upper left hand corner you will see a number. It corresponds to the segments being suppressed.

```
100 ! REDATCIR! !018
110 CALL CLEAR !209
120 CALL SCREEN(11)!196
130 READ S !234
140 CALL LINK("PRINT",22,16,
S)!040
150 IF S=999 THEN END !070
160 FOR A=1 TO 96 STEP 6 !03
2
170 CALL LINK("CIRCLE",96,12
0,A,S)!122
180 NEXT A !215
```

USER NOTES

(Continued from Page 27)

```
190 CALL LINK("CLEAR")!055
200 GOTO 130 !209
210 IF K>6 THEN 320 !072
220 GOTO 110 !189
230 DATA 1,4,64,128,8,2,32,1
6 !101
240 DATA 1,5,69,197,205,207,
239 !145
250 DATA 254,250,186,58,50,4
8,16 !196
260 DATA 105,131 !007
270 DATA 254,190,126,252,222
 !083
280 DATA 999 !009
```

Musical subprograms are real bell-ringer

The following was written by Earl Raguse and appeared in the LA99ers Toplcs newsletter.

The following subprograms are handy for programming music. The variables F, A, D, N stand for Frequency, Attenuation (loudness), Duration and Number of times to ring (as in bell). If you know what you are doing, you may modify these slightly to get different effects. Multiplying or dividing a frequency by 1.2 has the effect of raising or lowering key by one note. The duration of the quaver routine is not exact, and you may wish to fiddle with the divisor for D. You must supply all variables in the CALL statement.

```
9000 SUB BELL(F,N)
9010 FOR I=1 TO N
9020 FOR V=0 TO 20 STEP 4 ::
CALL SOUND(-99,F*1.783,V+3,
F*1.335,V+3,F,V):: NEXT V
9030 FOR V=20 TO 27 :: CALL
SOUND(-99,F*1.783,V+3,F*1.33
5,V+3,F,V):: NEXT V
9040 NEXT I
9050 SUBEND

9100 SUB WOW(F,D)
9110 FOR R=1 TO D
9120 X=F :: Y=X/.99 :: Z=F/1
9130 FOR V=20 TO 0 STEP -2
9140 CALL SOUND(-99,X,V,Y,V+
3,Z,V+3)
9150 NEXT V
9160 FOR V=0 TO 20 STEP 2
9170 CALL SOUND(-99,X,V+3,Y,
V,Z,V+3)
9180 NEXT V :: NEXT R
9190 SUBEND

9200 SUB MAJCHORD(F,A,D)
9210 X=F :: Y=F/1.26 :: Z=F/
1.5 :: XV=A
9220 YV,ZV=XV+5
9230 CALL SOUND(D,X,XV,Y,YV,
Z,ZV)
9240 SUBEND

9300 SUB MINCHORD(F,A,D)
9310 X=F :: Y=F/1.19 :: Z=F/
1.5 :: XV=A
9320 YV,ZV=XV+5
9330 CALL SOUND(D,X,XV,Y,YV,
Z,ZV)
9340 SUBEND

9400 SUB QUAVER(F,A,D)
9410 X=F :: Y=F*.99/2 :: Z=F
*1.01/2 :: XV=A:;YV=XV+9 ::
ZV=20
9420 FOR W=1 TO D/100
9430 CALL SOUND(-99,X,XV,Y,Y
V,Z,ZV)
9440 CALL SOUND(-99,X/1.01,X
V+2,Y,YV+4,Z,ZV+4)
9450 NEXT W
9460 CALL SOUND(-1,X/2,XV)
9470 SUBEND
```

Printing on both sides of the paper with TIW

This article was written by Frank W. Aylstock and appeared in several user group newsletters.

Since I won the laser printer at the 1993 Fest West North, I have been playing with TI-Writer and have tried some of the forgotten uses that our benefactor (Texas Instruments) had incorporated in the program. This is a tale of one of these facets of the program which I feel many of us may have overlooked.

I was printing out another copy of a disk I had received from the Chicago user group which contained a collection of their newsletter articles that had been written by Krome Dome about the Geneve.

This is a good tutorial and a story of how a Geneve user suffered and worked out the problems. It also is a good example of how to create an autoexec file with a menu. I first printed out the articles one at a time and ended up with a collection of 150 pages. This caused me to look at the problem of reducing the amount of paperwork.

I figured that the only way to conserve paper would be to print on both sides of each page.

I created an Include File to set up the FI,AD and all of the other printer commands I wanted to use. This would allow the pages to look alike, but I had to go to each file and strip them of all the existing printer commands. I then renamed the files and added their names to the Include File. Here's an example:

```
(.FI;AD;LM8;RM70;PL60)
(.FO   PAGE % )
.IF FILE1
.IF FILE2
```

I did this until all of the files had been renamed. I believe most users know how to use Include File so I won't bother going into the details.

Next, I placed a page break (BP) at the end of each file.

Now, when you get into the formatter, and you are prompted for a page number, you would respond with 1,3,5,7,9 etc. instead of accepting the default of *all*. You can do 5-10 pages with little trouble, but the buffer can hold only so much so I did only a few pages at a time.

The big item here for those with tractor feed is, after you have printed the odd pages, turn the paper over and feed it in to be printed with the even numbered pages.

When you get up to the higher pages it takes time for the computer to load the programs, figure where each page ends and which pages to print.

This all can be done with tractor feed paper or single sheet feed. With the single sheet you can also put the numbers of the pages in consecutive order and state that you wish to *pause* at the end of each page. Then feed each page in and turn it over to print the next side.

The reasoning in doing both sides of the page is to reduce the number of pages you have to have in your doc files.

I appreciate the extensive docs that have

(See Page 29)

USER NOTES

(Continued from Page 28)

come with many of the latest program such as DM-1000 Ver. 6, Funnelweb Ver. 4.4 or 5. I like the extensive documentation because it explains every detail to help make these programs more user friendly as well as to describe in detail the many functions available.

Batch file for 9640 shows text files

This item is by Bob Sherburne and has appeared in several user group newsletters. It requires a Geneve.

Did you ever say to yourself: "I wish those idiots would quit using slashes in filenames." MDOS despises the "/" character since it is a reserved symbol, and refuses to play ball with any file that has a slash in its name.

On occasion I like to read DV80 files from MDOS, so I wrote a little batch file to display them in TYPE/M (page) format. I call the batch file "SHOW" (see Fig. 1).

Lines 3 through 9 can be omitted if you can remember the syntax. "SHOW A DOG" would type out the file DOG on drive A:. "SHOW DOG/BONE" would ignore the slash in the filename and type out the file it it were found on the current working directory. I should mention that I keep this batch file in my PATH so that it is available at all times.

Fig. 1

```
ECHO OFF
CLS
IF NOT "%1"=="" GOTO WORK
ECHO FORMAT IS: SHOW [drive] [filename]
ECHO
ECHO [drive] IS OPTIONAL IF [filename] IS ON DEFAULT DRIVE
ECHO
GOTO END
:WORK
IF "%2"=="" GOTO CURRENT
ECHO SHOWING [ %1:%2 ]
ECHO
TYPE %1:"%2"/M
ECHO
GOTO END
:CURRENT
ECHO SHOWING [ %1 ]
TYPE "%1"/M
ECHO
:END
```

Perfect typist

The following program was written by Stephen Shaw, of Great Britain. No further explanation is needed.

```
1 REM SILLY PROG BY S SHAW
  MARCH 1991
2 ! did you see Computer War
s-the film? It is said that
the star, who was required t
o type fast into a computer
3 ! could not type, so a pro
gram just like this one was
used to give a good effect!
4 ! now adjust it how you wi
sh and show your friends how
 fast you can type
5 ! at end of text string pr
ogram will just stop with th
is listing but can be modifi
ed to do anything you wish!
6 !
100 A$="This is how a non-ty
pist canproduce information
on      screen quickly, with
out      "
110 A$=A$&"having to look at
 what keys are being bashed!
 Just bash keys and watch ho
w perfect  text appears no m
atter what you press."
120 CALL CLEAR :: PRINT A$:
: : : : :
130 CALL KEY(5,A,B):: IF B<1
 THEN 130
140 C=C+1 :: PRINT SEG$(A$,C
,1);:: IF C=LEN(A$)THEN 160
150 GOTO 130
160 GOTO 160
```

Converting TI and PC text files

The following was written by Jim Swedlow and originally appeared in ROM, the newsletter of the User Group of Orange County, California.

In our TI world, most text is saved in Display Variable 80 (D/V80) files. This is what the TI-Writer editor uses.

What is D/V80? Display means that the file is saved using ASCII characters. If you use a disk sector editor to look at the D/V80 file, you will see that the text looks just about the same as it was written. Internal files are not always as easy to read.

A file is made up of records. In a D/V80 file each record contains one line of text as it appears in the editor. Variable means that each record is only as long as the text line.

Consider these two lines of text:

TI
99/4A

When this is saved on disk, there will be two records in the file. The first record is "TI" and the second is "99/4A." Each record is preceded by the number of characters in the record in hexadecimal. Hex FF is used to mark the end of the file. The file would look like this:

```
HEX 02 54 49 05 39 39 2F 34 41 FF
ASC    T  I    9  9  /  4  A
```

In a Display Fixed 80 (D/F80) file, each record still contains one text line but is exactly 80 characters long. Your 4A pads each record by adding the required number of spaces to the end of each text line.

As a loyal TI user you may not think that you need to know how others (MS-DOS, C/PM, etc.) save text files. If you do any work with modems, however, you do.

The reason is that you may download a text file and find that it is Display Fixed 128 (D/F128). Why? There is a standard protocol in the TI world for transferring files using XMODEM. It was designed by Paul Charlton. The first record is *not* the first line of text. Instead, it is the disk header sector, which describes the file in a manner than can be read by the disk controller.

(See Page 30)

USER NOTES

(Continued from Page 28)

However, if the first record is not the header your terminal program (Telco, Fast Term, Mass Transfer, etc.) assumes that you are talking to a non-TI system and will save the file as D/F128.

Unlike a D/F80 file, there is no padding in a D/F128 file. Instead, all of the text lines are run together. The end of each text line is marked with a carriage return — CR or CHR$(13) — and a line feed — LF or CHR$(10).

One record may have 1, 2 or more text lines, each ending with a CR and LF. If there is not enough room left in a record for the next text line, enough of the line is added to the record to bring it to 128 characters. The rest of the text line starts the next record, followed by a CR and LF. The end of the file is marked with CHR$(26), which in the IBM and C/PM words is CTRL-Z.

Remember our sample text?

TI
99/4A

Since it is well under 128 characters, the fill will contain only one record:

```
HEX 54 49 0D 0A 39 39 2F 34 41 0D 0A 1A
ASC  T  I        9  9  /  4  A
```

Hex 0D 0A is a CR and LF. Hex 1A is the end of file marker, CHR$(26).

There are a number of programs that convert files from D/F128 to D/V80 or vice versa. Some of the assembly ones are quite fast. The program listed below does this, as well as two other conversions.

A little background. Sometimes you may look at a file and notice that there are no carriage returns (CRs). If you reformat such a document, everything will be jumbled into one big paragraph. TI-Writer stops reformatting when it hits a CR. Funnelweb stops when it hits a CR or a blank line. Either way, the document is a mess.

CONVERT, when converting a D/F128 file to D/V80, can add a CR to blank lines, to the end of paragraphs and to lines that start with a period (formatter commands). This takes a little long but it makes the file much easier to edit. Also, CONVERT can ad CRs to D/V80 files that lack them.

```
100 ! CONVERT !196
110 ! Version 1.0 !056
120 ! 09 Aug 88 !217
130 ! By Jim Swedlow !163
140 ! Based on XPREP by Carl
 Walters !205
150 !!131
160 DISPLAY ERASE ALL :: CAL
L SCREEN(5):: FOR A=0 TO 14
:: CALL COLOR(A,16,1):: NEXT
 A !126
170 FOR A=1 TO 4 :: READ T$(
A):: NEXT A !203
180 N$=CHR$(13)&CHR$(10):: Z
$=CHR$(26):: C$=CHR$(13):: G
OTO 300 !016
190 DATA DF128 -> DV80 add C
Rs,DF128 -> DV80 no CRs,DV80
  -> add CRs,DV80  -> DF128
!093
200 CALL KEY :: Q$,S,P,K,I$,
W$ :: !@P- !095
210 !!131
220 ! STRING CHECK SUB !002
230 !!131
240 P=1 :: IF I$=" " OR I$="
" THEN I$="" :: RETURN ELSE
IF ASC(I$)=46 THEN RETURN EL
SE P=0 :: RETURN !240
250 !!131
260 ! CLOSE FILES AND END !1
50
270 !!131
280 CLOSE #1 :: CLOSE #2 ::
DISPLAY AT(19,1)BEEP:"DONE"
!167
290 FOR P=1 TO 100 :: NEXT P
 !009
300 !!131
310 ! TITLE SCREEN !005
320 !!131
330 DISPLAY AT(5,5):"CONVERT
 Version 1.0": : : : : : : :
"Press For" !223
340 FOR S=1 TO 4 :: DISPLAY
AT(14+S,1):STR$(S);" ";T$(S)
:: NEXT S :: DISPLAY AT(19,1
)BEEP:"5 End Program" !054
350 !!131
360 ! PICK FUNCTION !080
370 !!131
380 CALL KEY(0,K,S):: IF K<4
9 OR K>53 THEN 380 ELSE K=K-
48 :: IF K=5 THEN DISPLAY ER
ASE ALL :: STOP !016
390 DISPLAY AT(13,1):T$(K):
:"Input File: DSK": :"Output
 File: DSK": : : :!109
400 ACCEPT AT(15,18)BEEP:I$
!039
410 ACCEPT AT(17,18)BEEP:W$
!055
420 !!131
430 ! OPEN FILES & INIT !002
440 !!131
450 DISPLAY AT(19,1):"Workin
g . . . ." !132
460 IF K>2 THEN OPEN #1:"DSK
"&I$,INPUT ELSE OPEN #1:"DSK
"&I$,INPUT ,FIXED 128 !144
470 IF K=4 THEN OPEN #2:"DSK
"&W$,OUTPUT,FIXED 128 ELSE O
PEN #2:"DSK"&W$,OUTPUT !120
480 A=1 :: W$="" :: ON K GOT
O 720,570,490,650 !043
490 !!131
500 ! DV80 -> DV80 ADD CRs !
099
510 !!131
520 LINPUT #1:I$ :: GOSUB 21
0 :: IF EOF(1)THEN 550 !063
530 IF A THEN IF P THEN PRIN
T #2:I$;C$ :: GOTO 520 ELSE
Q$=I$ :: A=0 :: GOTO 520 !02
7
540 IF P THEN PRINT #2:Q$;C$
:I$;C$ :: A=1 :: GOTO 520 EL
SE PRINT #2:Q$ :: Q$=I$ :: G
OTO 520 !044
550 IF A=0 THEN IF P THEN PR
INT #2:Q$;C$ ELSE PRINT #2:Q
$ !200
560 PRINT #2:I$;C$:C$ :: GOT
O 250 !184
570 !!131
580 ! DF128 -> NO CRs !056
590 !!131
600 LINPUT #1:I$ :: W$=W$&I$
 :: K=1 :: S=LEN(W$)!243
610 IF SEG$(W$,K,1)=Z$ THEN
250 ELSE IF K>S THEN IF EOF(
1)THEN 250 ELSE W$="" :: GOT
O 600 !133
620 P=POS(W$,N$,K):: IF P TH
EN PRINT #2:SEG$(W$,K,P-K)::
 K=P+2 :: GOTO 610 !219
630 P=POS(W$,Z$,K):: IF P TH
EN PRINT #2:SEG$(W$,K,P-K)::
 GOTO 250 !230
640 W$=SEG$(W$,K,255):: IF
OF(1)THEN PRINT #2:W$ :: GOT
O 250 ELSE 600 !043
650 !!131
```

(See Page 31)

USER NOTES

(Continued from Page 30)

```
660 ! DV80 -> DF128 !117
670 !!131
680 LINPUT #1:I$ !:: IF ASC(
I$)=128 THEN I$=" " !115
690 W$=W$&I$&N$ :: P=LEN(W$)
!080
700 IF P>128 THEN PRINT #2:S
EG$(W$,1,128):: W$=SEG$(W$,1
29,255)!040
710 IF EOF(1)THEN PRINT #2:W
$&Z$ :: GOTO 250 ELSE 680 !0
30
720 !!131
730 ! DF128 -> DV80 ADD CRs
!134
740 !!131
750 LINPUT #1:I$ :: W$=W$&I$
 :: K=1 :: S=LEN(W$)!243
760 IF SEG$(W$,K,1)=Z$ THEN
820 ELSE IF K>S THEN IF EOF(
1)THEN 820 ELSE W$"" :: GOTO
 750 !215
770 P=POS(W$,N$,K):: IF P TH
EN I$=SEG$(W$,K,P-K):: K=P+2
ELSE 800 !245
780 GOSUB 210 :: IF A THEN I
F P THEN PRINT #2:I$;C$ :: G
OTO 760 ELSE Q$=I$ :: A=0 ::
 GOTO 760 !159
790 IF P THEN PRINT #2:Q$;C$
:I$;C$ :: A=1 :: GOTO 760 EL
SE PRINT #2:Q$=I$ :: GOTO 76
0 !021
800 P=POS(W$,Z$,K):: IF P TH
EN I$=SEG$(W$,K,P-K):: GOTO
820 !005
810 W$=SEG$(W$,K,255):: IF E
OF(1)THEN I$=W$ ELSE 750 !21
6
820 IF A=0 THEN GOSUB 210 ::
 IF P THEN PRINT #2:Q$;C$ EL
SE PRINT #2:Q$ !108
830 PRINT #2:I$;C$:C$ :: GOT
O 250 !184
```

MICROpendium pays $10 for items sent in by readers and used in User Notes. Send your programs, tips and hints to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

ATTEND A TI FAIR IN 1993

CLASSIFIEDS

Policy

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

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

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

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

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

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

FOR SALE

TI99/4A

PHP1100	TI, 99/4A JOYSTICKS (Pair)	14.95
PHP1260	32K MEMORY CARD	99.95
PHP1500	SPEECH SYNTHESIZER	29.95
PHM3109	TI LOGO II (32K Mem. Req.)	1.95
PHP2500	TI IMPACT PRINTER	249.95
PHM3012	SECURITIES ANALYSIS	12.95
PHM3058	MINI-MEM (W/Assembler)	39.95
PHM3122	PLATO INTERPRETER	49.95

JOY ELECTRONICS INC.—P.O. BOX 542526 DALLAS, TX 75354-2526
(800) 527-7438, OUTSIDE DALLAS AREA
(214) 243-5371, DALLAS AREA
ADD $5.00 S&H—TEX RES ADD 8 ¼% TAX
10/8 **CALL OR WRITE FOR FREE CATALOG**

FOR SALE

ENORMOUS TI99/4A INVENTORY. CATALOGS $2.00. **BRAATZS COMPUTER SERVICES**, 719 E. BYRD ST., APPLETON, WI 54911. **1-414-731-3478.** 10/12

PRINTER FOR SALE

Panasonic KX-P1124, 24-pin dot matrix printer. Includes several ribbons, manual, TI cable. $125 + shipping. In very good shape, some discoloration from sunlight. Call 512-244-7330.

Buy/Sell Used Hardware and Software

National Used Software Club has buyers looking for TI products, as well as sellers. More than 170 TI items are listed for sale. 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 8-12 weeks). 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 NUS/HC, P.O. Box 1343, Round Rock, TX 78680; or call 512-255-1512.

Here are some of the items offered for sale:

Horizon RAMdisk (192K) $125.00
Advanced Diagnostics $25.00
Seagate 251 HD (40 meg) $75.00
TI XBASIC.....$20.00

Call our BBS for weekly price and product updates — 512-255-1557. Hours are from 5:30 p.m. to 8 a.m. weekdays and from noon Saturday through 8 a.m. Monday weekends. Supports 300-2400 baud.

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$35** ❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50** ❑ 12 issues, other countries surface mail, **$40.00**

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

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

Address Changes

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

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

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

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

_____

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

OUT OF STOCK: Vols. 1, No. 1-2; Vol. 2, No. 1; Vol. 4, No. 9

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 1.23F** (req. SSDD or larger, for MBASIC).....$4.00

❑ **MDOS 1.50H** (req. for MBASIC).....$4.00

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS

A T EXPIRES 1/94
CHARLES GOOD
P.O. BOX 647
VERMILION OH 44394