Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 11 Number 4 | May 1994 | $3.50

Utility Bill Audit in Extended Basic

See Page 12

More undocumented features of Extended BASIC

Living with Spiders

Programming with Funnelweb, second in a series

A hardware project to bring your Myarc RS232 card into the 1990s

Page 19

Geneve 9640

MY-BASIC and the internal RAMdisk

Geneve and MDOS

CONTENTS

MICROpendium

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

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

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

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

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

*READ THIS

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

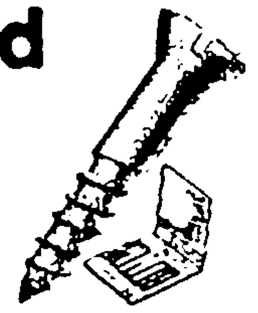
TEX+COMP

DOLLAR DAYS

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

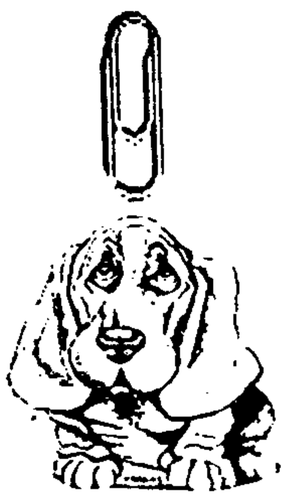
OUR LOWEST PRICES EVER ON THIS ENTIRE PAGE!

Big Price Reduction for Our Full-Featured COMPLETE PRINT SHOP PACKAGE FOR THE TI-99/4A special price $4.95*



featuring

GRAPHX + with Flip & Rotate & Quick Load (REG $19.95)



THE ALL TIME BEST GRAPHICS PROGRAM FOR THE 99/4A HAS JUST GOT EVEN BETTER WITH AN ALL NEW PACKAGE AT A GREAT LOW PRICE. ONLY $19.95 COMPLETE WITH GRAPHX+ AND TWO DISK SIDES FILLED WITH NEW CLIPART AND FONTS. GRAPHX+ OFFERS EVERY FEATURE YOU NEED TO CREATE THE BEST 4A GRAPHICS YOU HAVE EVER SEEN. GRAPHX+ OFFERS FEATURES SUCH AS FREE HAND DRAWING AND ERASING, ZOOM FOR DETAILED WORK, AUTO FILL, COLORS, TEXT AND TITLES, AUTO CIRCLE DRAWING, CLIP STORAGE AND ANIMATION. 32K, EXBASIC, AND A STAR/EPSON COMPATIBLE DOT MATRIX PRINTER IS REQUIRED.

NEW FONTS INCREDIBLE

***BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF $4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE FOR ONLY $1.00.**

GRAPHX DINOSAURS!

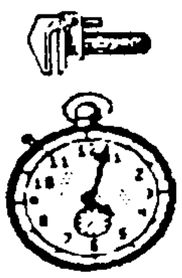
$4.95

Asgard Software

DISK OF DINOSAURS is a fantastic collection of real and whimsical dinosaurs from that master 99/4A artist Jon Guthrie. This format utility for Graphx also contains a bonus item "Dust of the Dinosaurs" for creating great scenes along with all the other great clip art.

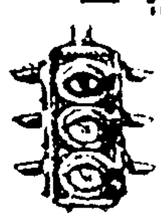
Graphx with printout resolution far superior to TI-Artist along with it's nearly instantaneous "zoom", built-in move/copy function, multi speed cursor, and always available clipboard is, in the words of Chris Bobbitt of Asgard Software, "the best drawing tool available for the 99/4A".

Now you can "dinosaurize" anything from calendars to greeting cards to reports. Create dinosaur stickers, logos, and pictures faster than you can say "Brontosaurus". Can also be used with TI Artist with the conversion routine supplied. GRAPHX and a 32K disk drive system required.



MAC-FLICK

HIGH RESOLUTION CLIP ART FOR GRAPHX AND TI ARTIST

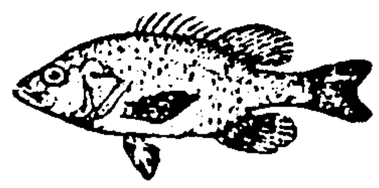


TEX-COMP HAS TAKEN THE BEST MAC® CLIPART AVAILABLE AND CONVERTED IT TO GRAPHX CLIPART FILES. NOW YOU CAN PRODUCE GRAPHICS WITH YOUR 4A THAT RIVALS THE BEST ANYWHERE. WE FILLED SIX DISK SIDES WITH OVER 140 PIECES OF CLIPART THAT ARE READY TO BE PRINTED OUT WITH GRAPHX OR CAN EVEN BE CONVERTED TO OTHER GRAPHIC SYSTEMS. THE SUPERIOR PRINTING CAPABILITY OF GRAPHX WILL CREATE ART THAT WILL BE OUTSTANDING!!

$4.95

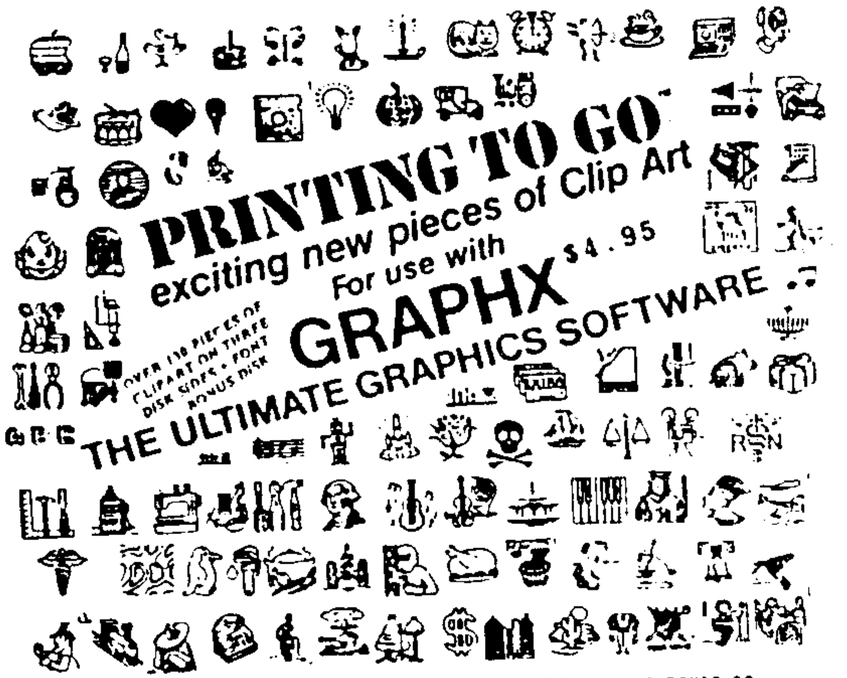
SIX DISK SIDES GRAPHX, 32K, AND STAR/EPSON DOT MATRIX PRINTER REQ.

SIX DISK SIDES



ASGARD SOFTWARE HAS PUBLISHED FOUR SETS OF PROFESSIONAL CLIPART AND FONTS FOR USE WITH THE GRAPHX+ PUBLISHING SYSTEM SO YOU CAN CREATE FANTASTIC ARTWORK WITHOUT EVEN HAVING ARTISTIC TALENT. EACH SET IS ONLY $4.95 (VOL I-IV)

GRAPHX SLIDESHOW BY ASGARD IS A CLIPART COLLECTION AND A SPECIAL PROGRAM FOR DISPLAYING YOUR GRAPHX CREATIONS IN AUTOMATIC SEQUENCE. GREAT FOR DISPLAYS, TRADESHOWS AND PRESENTATIONS.....ONLY $4.95



PRINTING TO GO IS A GREAT COLLECTION OF CLIP ART AND FONTS TO ADD TO YOUR GRAPHX PUBLISHING SYSTEM FOR THE TI-99/4A. THIS GREAT PACKAGE OF FOUR DISK SIDES AT A FANTASTIC PRICE IS ADDED PROOF THAT GRAPHX IS THE BEST BUY IN 4A GRAPHIC PACKAGES AND BY FAR, THE BEST SUPPORTED. THIS COLLECTION CAN BE USED FOR CREATING NEWLETTERS, ANNOUNCEMENTS OR ANY OTHER APPLICATION WHERE YOU NEED PROFESSIONAL LOOKING ART. THESE CLIPS CAN ALSO BE USED FOR CREATING PERSONALIZED STATIONARY OR EVEN FOR MAKING T-SHIRTS WITH THE TEX-COMP T-SHIRT KIT. OVER 130 PIECES OF QUALITY CLIPART + FONTS.

GRAPHX+ SYSTEM (reg. 19.95) special sale price	$4.95*
ASGARD GRAPHX COMPANION I	$4.95
ASGARD GRAPHX COMPANION II	$4.95
ASGARD GRAPHX COMPANION III	$4.95
ASGARD GRAPHX COMPANION IV	$4.95
ASGARD GRAPHX SLIDESHOW	$4.95
PRINTING TO GO (Clipart for GRAPHX reg 9.95)	$4.95
GRAPHX DINOSAURS (Dinosaur clipart reg. 12.95)	$4.95
MAC-FLICK (Hi-res. clipart for GRAPHX reg. 12.95	$4.95

*BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF $4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE (REG $19.95) FOR ONLY $1.00.

TEX+COMP

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

VISA and MASTERCARD HOLDERS CALL DIRECT:

(818) 366-6631

ORDER BY PHONE 24 HOURS A DAY (818) 366-6631

7 Days a Week!

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.

Send $2 for our complete catalog

COMMENTS

A parallel IDE interface?

Don Walden of Cecure Electronics is working on a way to get PC IDE hard drives to work with a Geneve. Sounds improbable, doesn't it?

Nonetheless, that's what he is doing. The trick, he says, is to build a parallel port that would allow the IDE drive to work off the port. Work is under way, he says, The device, which would use a standard PC cable, would offer parallel output, but not input. Input would come later, he says.

Software to operate the drive would be included in the interface cable itself. Jim Schroeder is in charge of the code. Assuming the device could do input and output, Walden thinks it would also be possible to use it on a PC.

"I'm looking at people with Geneves," he says. "Most of them have IBMs. It's a simple matter of them plugging it into their IBM."

Walden also was touting PFM+, which he describes as a "sibling" of RAMdisk "except that it doesn't go away." He describes PFM+ as being "like an internal ROMdisk" on which Geneve users can place GPL, MY-Word or other programs, up to 500 sectors worth. It requires PFM (Programmable Flash Memory) to work.

Users with PFM, PFM+ and a relatively large RAMdisk can now just about get along without a floppy drive.

DISK FORMATTING IN ABASIC

Jim Uzzell of DDI Software is developing a program called MYFORMAT for the Geneve. It's written in ABASIC and is used to format floppy disks. The program supports fo mats from single-sided single-density to double-sided do ble-density. It also lets the user select the skew and interlac The copy I've seen is a beta version that seems to work ju fine, as long as you don't go around hitting keys randomly.

SCSI DSRs DUE SOON

Bud Mills had hoped to have the DSRs for the SCSI card time for the Lima fair but reports that it didn't happen. Mil Maksimik told Mills that he'd have the DSRs done "ve soon." Because of the missing DSRs, Mills declined to se any of the cards at Lima.

Mills did sell some of his other products, however, inclu ing an AT-style keyboard for the TI, RAMdisks, Meme cards and P-GRAM cards.

Mills notes that he had planned to use a modified version MDM5 as a SCSI disk manager but there were too man problems. Maksimik is working on a built-in CALL FOR MAT and CALL MD (make directory) and CALL RD (re move directory). These will work from BASIC and Extend BASIC. The remaining disk manager functions, such as fil copying, can be done through Funnelweb's Disk Review util ity. "This will give us an introductory functioning system, Mills says. "Mike is still going forward with producing thorough disk manager that will cover all systems to be re leased in the future."

—JI

BUGS & BYTES

TI a threat?

The board of directors of the 500-member IBM-oriented Tucson Computer Society removed the announcement of Fest West, sponsored by the SouthWest 99ers, from their BBS. Though the sysops of each board had decided to post announcements about each other's group's fairs, the Tucson Computer Society's board felt announcement of the TI fair would detract from their fair Feb. 26.

Faire rerun set

The TI International World's Faire, the oldest of all TI fairs, is in the works, according to Don Walden, fair chairman. A tentative date of Nov. 12 is being looked at for a combined fair of the Chicago and Milwaukee Users Groups in Gurnee, Illinois. Walden says vendor tables will be available at a low price and vendors can charge their table rental on Visa or MasterCard.

Kids' day planned

The Dallas TI Home Computer Group plans a "Kids'Day" to include TI99/4A games with other activities and refreshments in July. They are opening it to elementary throughigh school aged children and grandchildren of members. If successful, they hope it will be an annual event.

Tigercub program re-release?

The Central Ohio Ninety-Niners Inc. (CONNI) have been given access to the entire library of the late Jim Peterson, who operated Tigercub Software, by his son, Alan Peterson. A committee from CONNI is now involved in copying the programs, and the group is studying the question of access to the programs by other TI users groups.

FEEDBACK

MDOS and GPL with joysticks

A note about MDOS 2.00 and GPL 2.00. Joysticks still require a TI Y-adaptor but the diodes are not needed, so simpler Y-adapter can be made. Both joysticks will now work at the same time.

Jack C. Mathis
Tuscon, Arizona

Praise for a classic

I've had my TI system for about a year and a half now. At the place I purchased the whole system, I was able to acquire back issues from 1987 to 1991; it took me about a year to finally read everything. The quality of the magazines is superb; clearly, there has never been a finer publication regarding the TI than yours. By going through those back issues, there is a true"history" of events such as the various TI fairs, new hardware and software offerings and insight into the many people who have stood out in the TI community. Also, I have never seen such quality customer support than that of TI vendors whom I have done business with.

Many people would of course suggest that I "upgrade" by getting something that is technically "state of the art." To me, ease of use is, by far, more important than a graphics resolution in excess of 256x192 pixels. A quality magazine, quality vendors and quality products have kept the TI going strong farther than many people would expect.

John Keating
Green Bay, Wisconsin

First step to fame

If you were to purchase a book entitled *Internet Basics — Your Online Access to the Global Electronic Superhighway*, you might notice and recognize the authors. Do the names Steven Lambert and Walt Howe ring bells? Mr. Howe states in the preface"I bought my personal computer in 1981, a TI99/4A .... " The book is targeted to users on Delphi accessing the Internet. Just a little gossip. While the TI community is shrinking, the influence of TIers may be expanding, particularly with J.P. Hoddie working for Bill Gates.

Gary Fitzgerald
Milford, Connecticut

Howe and Hoddie wrote a number of third-party programs for the TI, wrote for MICROpendium and were sysops of Compuserve's TI Forum at one time. Anyone want to contribute information on where any other former movers and shakers in the TI community are now?— Ed.

***Feedback is a reader forum. The editor may condense excessively lengthy submissions if necessary. We ask that writers limit themselves to one subject per submission. Our only requirement is that submissions be of interest to those using the TI99/4A, the Geneve 9640 or compatibles. Send items to* MICROpendium Feedback, P.O. Box 1343, Round Rock, TX 78680.**

READER TO READER

❑ Michael Scheller, 1379 W. Iris Place, Casa Grande, AZ 85222-3701, writes:

I am currently a member of VAST (Valley of the Sun TI) Users Group, Phoenix, Arizona. I want to correspond to other users groups, but more than that I want to what I consider the foremost element in maintaining this "orphan," the individual TI99/4A user.

Persons willing to correspond should write Scheller at the above address.

❑ Jerry Clasby, 612 Meandering Rd., Frederick, OK 73542, writes:

I've just come upon a printer buffer I hope will work with the TI99/4A to Gemini connection. It is a buffer from Global, model C4684. The jacks are marked with"parallel output" and "parallel input." However, there is an 8 pole dip switch on the bottom of the box which I believe is baud rate for a serial buffer, right? I tried the buffer's self test (which also says that it is a parallel buffer) and the printer responded correctly. Would a serial buffer do that?

Also, I seem to remember reading somewhere that Texas Instruments swapped some wires to make their system incompatible with other systems of the day. If so, can someone enlighten me so that if I do come across a parallel buffer, I could make the changes?

❑ Ron Warfield, of the B.C. 99er User Group in New Westminster, British Columbia, Canada, has shared his replies to two readers who had letters in the March MICROpendium. To Vern Jensen, he writes:

You asked about a compiler for the TI. The only compiler we have ever had is a program called SST BASIC Compiler System by SST Software (1983). It was meant to operate from the Mini-Memory card and a cassette recorder. It would convert a BASIC program to machine language. It could save the compiled program to a disk. If there are other compilers, we have not heard of them yet.

To Olden Warren, he writes:

You asked about getting a pair of disk drives to work. It has been our experience that most hardware from Digital Computers just is not compatible with the TI. We have tried drives and printers with no luck.

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

THE ART OF ASSEMBLY — PART 35

More undocumented features of Extended BASIC

By BRUCE HARRISON

Our production of this series has been stopped for a short while as we wrote the "beginner" mini-series. This month, we take up where Part 30 left off, with more on the use of features built into the TI, many of which are undocumented. We'll be overlapping into our friend Barry Traver's territory somewhat, in that much of what's shown this month is designed for operation in the Extended BASIC "environment." That is, with a 32-character screen, the offset needed for characters written to the screen, and so on. Today's sidebar is, in fact, the source code for a couple of routines we made for use under Extended BASIC. The key "what's theres" in this case came from Harry Wilhelm, to whom we owe yet another debt of gratitude.

CLEARING THE DECKS

Let's start with something really simple, which will get the computer into a "BASIC" environment very quickly. One little BLWP to GPLLNK will do all of the following:

1. Clear the screen.
2. Restore the normal BASIC character sets, both upper and lowercase, including the cursor and edge characters.
3. Set the random number seed.
4. Delete any and all sprites.
5. Set the screen color to cyan.
6. Set the colors for all character sets to black on cyan.
7. Set all VDP registers to their BASIC values.

Pretty good for just one call to GPLLNK. The magic number is >27E3. Or, as it shows up in source code:

```
BLWP @GPLLNK
DATA >27E3
```

That's it! All seven of those things happen in a trice, and you're set up with a cleared cyan screen and essentially in a BASIC environment. The singular drawback to all this (you knew there had to be one) is the ever-popular offset for the screen characters. Adding >60 to each character sent to the screen and subtracting >60 from any character read from the screen is a real pain, but that's part of the price we pay for the "efficient" arrangement of the character definitions in BASIC. The other part of the price we pay is that characters below the cursor (>1E, or >7E with the offset) cannot be used, since the space that would be used for their definitions is part of the screen.

We've seen cases where a programmer got around that limitation by using >3800 as his screen image table, but that's not a practice we recommend. We won't condemn you for doing that, but don't complain to us if that screws up something else you were trying to do.

Included in today's sidebar is another gift from Harry Wil-

(See Page 7)

```
* SIDEBAR 35
*
* FIRST PART IS "RSXB" - RESETS MANY
* PARAMETERS TO DEFAULT CONDITIONS
* AS EXTENDED BASIC CALL LINK("RSXB")
* PUBLIC DOMAIN
*
STATUS EQU  >837C
GPLWS  EQU  >83E0
GR4    EQU  GPLWS+8
GR6    EQU  GPLWS+12
STKPNT EQU  >8373
LDGADD EQU  >60
XTAB27 EQU  >200E
GETSTK EQU  >166C
       DEF  RSXB
RSXB
       LWPI WS           LOAD OUR WORKSPACE
       CLR  @STATUS      CLEAR GPL STATUS
       BLWP @GPLLNK      USE GPL LINKAGE
       DATA >27E3        CLEAR OUT TO XB DEFAULTS
       LWPI GPLWS        LOAD GPL WORKSPACE
       B    @>6A         BRANCH TO GPL INTERPRETER
WS     BSS  32           OUR WORKSPACE
*
* FOLLOWING IS GPLLNK BY DON WARREN/CRAIG MILLER
*
GPLLNK DATA GLNKWS
       DATA GLINK1
RTNAD  DATA XMLRTN
GXMLAD DATA >176C
       DATA >50
GLNKWS EQU  $->18
       BSS  >08
GLINK1 MOV  *R11,@GR4
       MOV  *R14+,@GR6
       MOV  @XTAB27,R12
       MOV  R9,@XTAB27
       LWPI GPLWS
       BL   *R4
       MOV  @GXMLAD,@>8302(R4)
       INCT @STKPNT
       B    @LDGADD
XMLRTN MOV  @GETSTK,R4
       BL   *R4
       LWPI GLNKWS
       MOV  R12,@XTAB27
       RTWP
       END
*
* END OF THE "RSXB" ROUTINE
*
* SECOND PART IS SOURCE CODE FOR THE
* ULTIMATE ACCEPT AT ROUTINE
*
* PUBLIC DOMAIN
* USE WITH EXTENDED BASIC CALL LINK
* USE BY CALL
LINK("ULTACC",R,C,CLR,"PROMPT",NOCHRS,VAR[,"BEEP"])
* WHERE: R,C ARE ROW AND COLUMN
* CLR MAY BE 0, 1 OR 2 (0 DOES NOT CLEAR SCREEN, 1 DOES, 2 RE-
SETS XB)
* PROMPT MAY BE ANY STRING, INSERT "" FOR NO PROMPT
* NOCHRS IS NUMBER OF CHARACTERS TO ACCEPT.
```

THE ART OF ASSEMBLY—

(Continued from Page 6)

helm, in the form of two utility vectors (PRSTR and VMBW60) that will help you over the problem of writing things to the screen with the offset. These will help so long as you keep in mind that the characters from number 29 downward are verboten. (German for Forbidden.)

Pardon a digression, but that word verboten has such a neat ring to it. It always reminds us of the opening scene of the TV show Hogan's Heroes. As the prisoners were dashing out into the snow from their quarters, we noticed a sign beside the door. The sign was in German, so don't ask us what it said, except that the word in large letters at the very top was VERBOTEN, and the list of items in very small letters stretched down to the ground. That kind of thing always has special meaning for those who've worked for the Government. The list of what's verboten always fills many more volumes than the list of what's permitted. Now I've been reminded of another story, concerning the use of Festive Decorative Materials, but I'd better save that story and get on with the business at hand, or this column will be verboten. (That's five times we've used that word. Enough!)

Where were we? Oh yes, the lead-in to the sidebar should come next. In that sidebar, the first part is source code for what's called the RSXB routine. As you can see, this is a very simple but complete sub-program for use with CALL LINK from Extended BASIC. (Presumably it could also be used from console BASIC the E/A cartridge is available.) What makes it complicated is the need to include one's own GPLLNK, since TI in its infinite wisdom decided not to provide that for XB users. (Thanks again, TI!)

THE LINE EDITOR

Harry Wlhelm has provided us with another undocumented GPL routine, the Line Editor. So far as we can tell, this is the routine that's used by the Command mode in BASIC and Extended BASIC to allow you to type in commands or program lines, and of course to edit existing program lines. As soon as we'd tinkered around with this a bit, an old idea came flashing back into our consciousness. That idea was something we call the "Ultimate Accept At". We always liked TI Extended BASIC's ACCEPT AT, for the neat way it allowed us to take user inputs from anywhere on the screen. Our main gripe with it was that no more than 28 characters could be input. String Variables can hold as many as 255 characters in XB, but not if entered through ACCEPT AT.

Having the TI Line Editor available, with its many allowed parameters, gave us the freedom to invent that desired routine which permits an effective Accept At with room for 255 characters if desired. That's real Power, there in that Line Editor which TI didn't bother to mention to us. One can place the start and stop locations for input anywhere on the screen. (The stop location can even be off the bottom of the screen, if we'd like.)

The second part of today's sidebar is the source code for the Ultimate Accept At. This uses TI's Line Editor by the simple source statements:

```
BLWP @GPLLNK
DATA >285A
```

(See Page 8)

```
* MAKING NOCHRS NEGATIVE WILL PUT EXISTING VALUE OF VAR ON
SCREEN AS DEFAULT
* THE VARIABLE MAY BE ANY NUMERIC OR STRING VARIABLE
* THE "BEEP" IS AN OPTIONAL PARAMETER - PLACING ANYTHING
* AFTER THE VAR PARAMETER WILL MAKE A BEEP OCCUR
* IF THE WORD BEEP IS USED, IT MUST BE SURROUNDED BY QUOTES
*
VMBW   EQU  >2024       VDP MULTI-BYTE WRITE
VSBW   EQU  >2020       VDP SINGLE-BYTE WRITE
VMBR   EQU  >202C       VDP MULTI-BYTE READ
VSBR   EQU  >2028       VDP SINGLE-BYTE READ
KSCAN  EQU  >201C       KEYBOARD SCAN
CFI    EQU  >12B8       CONVERT FLOATING POINT TO INTEGER
CIF    EQU  >0020       CONVERT INTEGER TO FLOATING POINT
CNS    EQU  >0014       CONVERT NUMBER TO STRING
CSN    EQU  >11AE       CONVERT STRING TO NUMBER
ARG5ID EQU  >8305       LOCATION FOR TYPE OF ARGUMENT 5
ARGNUM EQU  >8312       LOCATION FOR NUMBER OF ARGUMENTS
SCRWID EQU  32          SCREEN WIDTH
STRASG EQU  >2010       ASSIGN STRING VARIABLE
STRREF EQU  >2014       GET STRING VARIABLE
NUMREF EQU  >200C       GET NUMERIC
NUMASG EQU  >2008       ASSIGN NUMERIC VARIABLE
XMLLNK EQU  >2018       XML LINKAGE VECTOR
FAC    EQU  >834A       FLOATING POINT ACCUMULATOR
FAC11  EQU  FAC+11      PLUS ELEVEN BYTES
FAC12  EQU  FAC+12      PLUS TWELVE BYTES
SCROLL EQU  >0026       XB SCROLL ROUTINE (W/XMLLNK)
GPLWS  EQU  >83E0       GPL WORKSPACE
GR4    EQU  GPLWS+8     GPL REGISTER 4
GR6    EQU  GPLWS+12    GPL REGISTER 6
STKPNT EQU  >8373       STACK POINTER
LDGADD EQU  >60
XTAB27 EQU  >200E
GETSTK EQU  >166C
STATUS EQU  >837C       GPL STATUS BYTE
       DEF  ULTACC,ULTCLR
ULTCLR
       LWPI WS          LOAD OUR WORKSPACE
       BL   @CLRXB      USE CLRXB SUBROUTINE
       B    @EXIT       EXIT THIS ROUTINE
ULTACC
       LWPI WS          LOAD OUR WORKSPACE
       CLR  @NUMFLG     CLEAR NUMBER FLAG
       CLR  @DEFFLG     CLEAR DEFAULT FLAG
       CLR  R0          NOT ARRAY
       LI   R1,1        FIRST PARAMETER (ROW)
       BLWP @NUMREF     GET NUMBER
       BLWP @XMLLNK     USE XML VECTOR
       DATA CFI         TO CONVERT TO INTEGER
       MOV  @FAC,R4     PUT ROW IN R4
       JLT  ROWERR      IF NEG, ERR
       JEQ  ROWERR      IF ZERO, ERROR
       CI   R4,25       COMPARE TO 25
       JLT  GETCOL      IF <25, OKAY
ROWERR B    @ROWNG      ELSE REPORT ERROR
COLERR B    @COLNG      REPORT COLUMN ERROR
GETCOL
       MOV  R4,@ROW     PLACE ROW # AT LOCATION ROW
       INC  R1          NEXT PARAM (COLUMN)
       BLWP @NUMREF     GET NUMBER
       BLWP @XMLLNK     USE XML
       DATA CFI         CONVERT TO INTEGER
       MOV  @FAC,R4     GET INTO R4
       JLT  COLERR      IF NEG, ERROR
       JEQ  COLERR      IF ZERO, ERROR
       CI   R4,28       COMPARE TO 28
       JGT  COLERR      IF >28, ERROR
       MOV  R4,@COL     MOVE TO COL
GETCLR INC  R1          NEXT PARAM
       BLWP @NUMREF     CLEAR SCREEN?
       BLWP @XMLLNK     USE XML
       DATA CFI         CONVERT TO INTEGER
```

THE ART OF ASSEMBLY—

(Continued from Page 7)

Of course this little routine does a lot of work getting prepared for that call, including the reading of parameters that XB passes along in the Call Link, clearing the necessary screen area for our input field if desired, and placing a prompt on the screen if needed. The routine also checks to see whether there's enough room on the screen for the desired length of input field, and moves its ACCEPT up by enough rows to insure screen space.

This routine, designed for the Extended BASIC programmer, is available as Public Domain software from a number of sources, including the Lima and Chicago User's Groups. The disk includes the complete source code, object files, instructions on how to use the routines, and demo XB programs to show how they work. If you can't get it anywhere else, we'll provide it directly for a paltry $1.50 to cover media and mailing. (5705 40th Place, Hyattsville MD 20781 USA)

OTHER UNDOCUMENTEDS

No, you can't hire these to look after your children, or to serve as domestic workers. There are two more for today's column that are worth mentioning. Back in Part 30, we showed a GPL routine that would take an integer number and convert it into a string, which we then displayed on the screen. Our friend Harry has an even better deal for us. Once again, this uses the offset for screen characters, but we don't need to add the offset, or even do the displaying. All we do is take the one-word integer, move it into location >835E, put our desired screen location at >8320, then call the routine through GPLLNK. Looks like this:

```
LI   R0,32*11+2  (row 12, col 3, for example)
MOV  R0,@>8320   put that at >8320
MOV  @NUMBER,@>835E  place the integer at ARG
BLWP @GPLLNK     use GPL linkage
DATA >2842       with this DATA number.
```

Voila! The number is on the screen, with offset, in decimal notation, at row 12, column 3. Neat, eh?

That's another small miracle. No doubt there are hundreds of these that we don't yet know about. Last on today's list is the matter of scrolling the screen. Some time ago, we showed the undocumented way to do that using GPLLNK:

```
BLWP @GPLLNK
DATA >4D00
```

That one will work from E/A, or BASIC, or Extended BASIC. The newest addition to our growing list is actually a documented one (Page 416 of the E/A Manual). This is a scroll that can be used only in the XB environment, and through XMLLNK, not GPLLNK. It looks like this:

```
BLWP @XMLLNK
DATA >0026
```

This is, as we said, documented on page 416 of the E/A manual, as part of a list of unexplained equates, all of which are to be used with XMLLNK, except that TI forgot to mention that this is how they are to be used. That list includes:

```
CNS    EQU  >06  convert number to string
VPUSH  EQU  >0E  PUSH a value?
VPOP   EQU  >10  POP a value?
ASSGNV EQU  >18  does what?
```

(See Page 9)

```
       MOV  @FAC,R4      MOVE TO R4
       MOV  R4,@CLRFLG   PLACE AT FLAG LOCATION
       JEQ  NOCLR        IF ZERO, DONT
       CI   R4,1         IS VALUE 1?
       JGT  CLRALL       IF GREATER, JUMP
       CLR  R0           ELSE CLEAR R0
       LI   R1,BLNKLN    POINT AT BLANK LINE
       LI   R2,32        32 CHARACTERS
       LI   R4,24        24 ROWS
CLSLP  BLWP @VMBW60      WRITE 32 SPACES TO SCREEN
       A    R2,R0        MOVE DOWN 1 ROW
       DEC  R4           DEC ROW COUNT
       JNE  CLSLP        IF NOT ZERO, REPEAT
       CLR  R0           RE-CLEAR R0
       JMP  NOCLR        JUMP AHEAD
CLRALL
       BL   @CLRXB       USE SUBROUTINE
NOCLR
       LI   R1,4         PROMPT PARAMETER
       LI   R2,PRMSTR    POINT AT PROMPT STRING
       MOVB @MAXLEN,*R2  PLACE MAX LENGTH NUMBER THERE
       BLWP @STRREF      GET THE STRING
       MOVB @PRMSTR,R4   GET ACTUAL LENGTH IN R4
       SRL  R4,8         RIGHT JUSTIFY
       CI   R4,29        COMPARE TO 29
       JLT  LENOK        IF LESS, LENGTH IS OKAY
       B    @PRMNG       ELSE BRANCH TO ERROR REPORT
LENOK  MOV  R4,@PRMLEN   STASH LENGTH
       INC  R1           NEXT PARAM
       BLWP @NUMREF      ALLOWED INPUT LENGTH
       BLWP @XMLLNK      USE XML
       DATA CFI          CONVERT TO INTEGER
       MOV  @FAC,R4      MOV TO R4
       JGT  C256         IF POSITIVE, JUMP
       JEQ  CHRERR       IF ZERO, ERROR
       INC  @DEFFLG      SET DEFAULT FLAG
       NEG  R4           NEGATE R4 (MAKES IT POSITIVE NUMBER)
C256   CI   R4,256       COMPARE TO 256
       JLT  ARG4OK       IF LESS, OKAY
CHRERR B    @CHRNG       ELSE ERROR
ARG4OK
       MOV  R4,@MAXCHR   STASH ALLOWED
       INC  R1           NEXT PARAMETER
GARG5  MOVB @ARG5ID,R4   GET PARAMETER TYPE IN R4
       SRL  R4,8         RIGHT JUSTIFY
       CI   R4,4         COMPARE TO 4
       JLT  FNDTY        IF LESS, PARAMETER IS OKAY
       B    @PTNG        ELSE ISSUE ERROR
FNDTY
       ANDI R4,1         MASK ALL BUT LAST BIT
       JNE  GETSTR       IF NOT ZERO, JUMP
       INC  @NUMFLG      ELSE THIS IS A NUMERIC PARAMETER
       MOV  @THIR2,@MAXCHR SET MAX CHARACTERS AT 32
       BLWP @NUMREF      GET NUMBER FROM XB
       MOVB @ONE,@FAC11  SET FAC+11 TO ZERO
       CLR  @STATUS      CLEAR GPL STATUS BYTE
       BLWP @GPLLNK      USE GPL LINK
       DATA CNS          TO CONVERT NUMBER TO STRING
       MOVB @FAC12,R4    GET STRING LENGTH
       LI   R10,TEMSTR   POINT AT TEMPORARY STORAGE
       MOVB R4,*R10+     MOVE LENGTH TO THAT LOCATION
       SRL  R4,8         RIGHT JUSTIFY LENGTH
       MOVB @FAC11,R9    GET LOW BYTE OF ADDRESS
       SRL  R9,8         RIGHT JUSTIFY
       AI   R9,>8300     ADD >8300 HIGH BYTE
       CB   *R9,@MINUS   SEE IF A NEGATIVE NUMBER
       JEQ  MOVB         IF SO, JUMP
       INC  R9           ELSE POINT AHEAD BY ONE BYTE
       DEC  R4           DECREMENT LENGTH COUNT
       MOVB @WS+9,@TEMSTR  MOVE LOW BYTE OF R4 TO LENGTH BYTE
TEMSTR
MOVB   MOVB *R9+,*R10+   TAKE ONE BYTE OF STRING
       DEC  R4           DECREMENT LENGTH COUNT
```

THE ART OF ASSEMBLY—

(Continued from Page 8)

```
CIF    EQU  >20  convert integer to floating point
SCROLL EQU  >26  Scroll screen
VGWITE EQU  >34  does what?
GVWITE EQU  >36  does what?
```

The meanings of the "does whats" in this list is unclear. If any of our readers has used these equates for anything, please let us know, and we'll pass the information along. We suspect that the VPUSH and VPOP have something to do with using a value stack in VDP ram for floating point math operations, but can't at this time confirm or deny that suspicion. The SCROLL in this list reportedly works faster than the >4D00 scroll, but this one apparently works only from the Extended BASIC environment.

Here it is, getting on toward ten years since the end of production of the TI-99/4A, and still there are mysteries buried in the "system" software needing to be unraveled. Our friend Harry Wilhelm uses the old book *The TI Intern* to dig out some information, but having looked at that, we still don't see how he comes up with useful routines from it.

That's enough for today. There's a rather big sidebar for today's column, with all that annotated source code to chew on, so this will be it for now. Next month's topic is still undecided, so we'll once again surprise you (and ourselves) next month.


```
       JNE  MOVB          IF NOT ZERO, REPEAT
       JMP  PRMOK         ELSE JUMP AHEAD
GETSTR LI   R2,TEMSTR     POINT AT TEM STRING SPACE
       MOVB @MAXLEN,*R2   PUT MAX LENGTH THERE
       BLWP @STRREF       GET THE STRING FROM XB
PRMOK
       MOVB @TEMSTR,R4    GET BACK STRING LENGTH
       SRL  R4,8          RIGHT JUSTIFY
       C    R4,@MAXCHR    COMPARE TO MAX NUMBER CHARACTERS
       JEQ  SLOK          IF EQUAL, OKAY
       JLT  SLOK          IF LESS THAN, OKAY
       MOV  @MAXCHR,R4    ELSE TRUNCATE STRING
SLOK   MOV  R4,@DEFLEN    MOVE R4 TO DEFAULT STRING LENGTH
       SWPB R4            SWAP BYTES
       MOVB R4,@TEMSTR    PLACE AT TEMP STRING
       MOV  @ROW,R0       GET ROW VALUE INTO R0
       MOV  @PRMLEN,R1    MOVE PROMPT LENGTH TO R1
       MOV  @MAXCHR,R2    MOVE MAX CHARACTERS TO R2
       MOV  @DEFLEN,R3    GET DEFAULT LENGTH IN R3
       C    R2,R3         COMPARE
       JGT  R2OK          IF GREATER, JUMP
       MOV  R3,R2         ELSE REPLACE R2 WITH R3
R2OK
       A    R1,R2         ADD PROMPT LENGTH
       A    @COL,R2       ADD COLUMN
       INC  R2            THEN ADD ONE
* R2 HAS SUM OF COLUMN, AND LENGTHS
       MOV  R2,@TOTLEN    MOVE TO TOTAL LENGTH WORD
       CLR  R1            CLEAR R1
       DIV  @TWEN8,R1     DIVIDE R1-R2 BY 28
       MOV  R2,R2         CHECK REMAINDER
       JEQ  R1OK          IF ZERO, OKAY
       INC  R1            ELSE INCREMENT R1
R1OK
* R1 HAS NUMBER OF ROWS REQUIRED FOR
* PROMPT AND INPUT
       MOV  R1,@TOTROW    MOVE TO TOTAL ROWS LOCATION
       A    R0,R1         ADD START ROW
       CI   R1,25         WILL THIS GO PAST 24?
       JLT  ROWOKY        IF LESS, OKAY
       AI   R1,-24        ELSE SUBTRACT 24
       S    R1,@ROW       THEN SUBTRACT THAT FROM ROW
ROWOKY
       MOV  @ROW,R9       GET ROW INTO R9
       DEC  R9            DECREMENT
       MPY  @THIR2,R9     MULTIPLY BY 32
       MOV  R10,R14       SAVE PRODUCT IN R14
       MOV  @COL,R9       GET COLUMN
       INC  R9            INCREMENT
       A    R9,R10        ADD TO PRODUCT OF ROW-1 X 32
       MOV  R10,R0        MOVE TO R0 - STARTING SCREEN POSITION
       MOV  R10,@STPRM    STASH AT LOCATION STPRM
       MOV  @TOTLEN,R2    GET TOTAL LENGTH INTO R2
       MOV  @TOTROW,R4    MOVE TOTAL ROWS TO R4
       LI   R1,4          LOAD 4 INTO R1
       MPY  R1,R4         MULTIPLY R4 BY 4
       A    R5,R2         ADD PRODUCT TO R2
       MOVB @SPACE,R1     LOAD R1 WITH SPACE BYTE
       MOV  R0,R13        STASH R0 IN R13
CLRLP  BLWP @VSBW         WRITE ONE SPACE
       INC  R0            INCREMENT SCREEN LOCATION
       DEC  R2            DECREMENT COUNT
       JNE  CLRLP         IF NOT ZERO, REPEAT
       DEC  R0            DECREMENT LOCATION
       MOV  R0,@ENDOC     STASH AT ENDOC
       AI   R14,SCRWID    ADD SCREEN WIDTH TO R14
       DECT R14           THEN SUBTRACT 2
       MOV  R14,R0        MOVE THIS TO R0
       MOV  @TOTROW,R4    MOVE TOTAL ROWS TO R4
       LI   R1,EDGE       POINT AT EDGE CHARACTERS
       LI   R2,4          FOUR OF THEM
EDGELP
       BLWP @VMBW         WRITE FOUR EDGE CHARS
```

THE ART OF ASSEMBLY—

```
      AI   R0,32        MOVE DOWN ONE ROW
      CI   R0,23*32     COMPARE TO ROW 24
      JLT  DEC4         IF LESS, JUMP
      DECT R2           ELSE SUBTRACT 2 FROM R2
      JMP  LASEDG       THEN JUMP
DEC4   DEC  R4          DECREMENT ROW COUNT
      JNE  EDGELP       IF NOT ZERO, JUMP
LASEDG BLWP @VMBW        WRITE LAST TWO EDGE CHARACTERS
      DEC  R0           DECREMENT POSITION
      MOV  R0,@LASPOS   STASH AT LASPOS
* NEXT SECTION PUTS PROMPT ON SCREEN
      MOV  @STPRM,R0    GET PROMPT START POSITION
      LI   R9,PRMSTR    POINT AT PROMPT STRING
      MOVB *R9+,R2      GET ITS LENGTH
      SRL  R2,8         RIGHT-JUSTIFY
      JNE  PRDLP        IF NOT ZERO, OKAY
      MOV  @STPRM,@STDOC ELSE MOVE START OF PROMPT TO STDOC
      JMP  PUTDEF       THEN JUMP AHEAD
PRDLP
      BLWP @VSBR        READ THE CHARACTER AT CURRENT SCREEN PO-
SITION
      CB   R1,@EDGE     IS THAT EDGE CHARACTER?
      JEQ  SKIPW        IF SO, SKIP IT
      MOVB *R9+,R1      ELSE GET PROMPT CHARACTER
      AB   @OFFSET,R1   ADD OFFSET
      BLWP @VSBW        THEN WRITE
      DEC  R2           DEC CHARACTER COUNT
      JEQ  PRDX         IF ZERO, EXIT
SKIPW
      INC  R0           INC SCREEN LOCATION
      JMP  PRDLP        THEN REPEAT ABOVE
PRDX
      INC  R0           INC SCREEN LOCATION
      MOV  R0,@STDOC    MOVE TO STDOC
PUTDEF
      MOV  @STDOC,R0    GET STDOC BACK INTO R0
CKED   BLWP @VSBR        READ CHARACTER
      CB   R1,@EDGE     IS THAT EDGE?
      JNE  PDEF1        IF NOT, GO AHEAD
      INC  R0           ELSE POINT AT NEXT SCREEN LOCATION
      JMP  CKED         THEN REPEAT
PDEF1
      MOV  R0,@STDOC    STASH R0 AT STDOC
      MOV  @DEFFLG,R4   GET DEFAULT FLAG IN R4
      JNE  PDEF0        IF NON-ZERO, JUMP
      MOV  R0,@STDOC    AND STASH
      JMP  GETIN        THEN JUMP AHEAD
PDEF0  LI   R9,TEMSTR    POINT AT DEFAULT STRING
      MOVB *R9+,R2      GET LENGTH INTO R2
      SRL  R2,8         RIGHT JUSTIFY
      JEQ  GETIN        IF ZERO, JUMP AHEAD
DEFLP
      BLWP @VSBR        READ CHARACTER FROM SCREEN
      CB   R1,@EDGE     IS THAT EDGE?
      JEQ  SKIPD        IF SO, SKIP
      MOVB *R9+,R1      ELSE GET STRING CHAR INTO R1
      AB   @OFFSET,R1   ADD OFFSET
      BLWP @VSBW        WRITE THAT
      DEC  R2           DEC COUNT
      JEQ  GETIN        IF ZERO, JUMP AHEAD
SKIPD  INC  R0          INCREMENT SCREEN POS
      JMP  DEFLP        THEN JUMP BACK

GETIN
      MOV  @STDOC,R0    SET FOR START OF INPUT FIELD
      MOV  @MAXCHR,R2   MOVE MAX CHARACTERS TO R2
GETINL BLWP @VSBR        READ CHARACTER
      INC  R0           INCREMENT POINTER
      CB   R1,@EDGE     IS THAT EDGE?
      JEQ  SKIP3        IF SO, SKIP
      DEC  R2           ELSE DEC COUNT
      JEQ  GETIN0       IF ZERO, JUMP
SKIP3
      JMP  GETINL       ELSE REPEAT
GETIN0
      MOVB @ARGNUM,R4   GET NUMBER OF ARGUMENTS INTO R4
      SRL  R4,8         RIGHT JUSTIFY
      CI   R4,7         COMPARE TO 7
      JLT  NOBEEP       IF LESS, NO BEEP ISSUED
      CLR  @STATUS      CLEAR GPL STAT
      BLWP @GPLLNK      USE GPLLNK
      DATA >34          FOR BEEP SOUND
NOBEEP
      DEC  R0           DECREMENT SCRN LOCATION
      MOV  R0,@ENDOC    MOVE TO ENDOC
      MOV  R0,@>832A    MOVE TO HEX 832A
      MOV  R0,@>835E    AND HEX 835E
      MOV  @STDOC,R0    GET START OF INPUT POSITION
      MOV  R0,@>8320    MOVE TO HEX 8320
      MOVB R0,@>8361    MOVE HIGH BYTE TO HEX 8361
      MOVB @WS+1,@>8362 LOW BYTE TO HEX 8362
      CLR  @STATUS      CLEAR GPL STATUS
      BLWP @GPLLNK      USE GPL LINK
      DATA >285A        LINE EDITOR FUNCTION
      MOV  @ENDOC,R0    GET END OF INPUT FIELD INTO R0
FLCLP
      C    R0,@STDOC    COMPARE TO START OF FIELD
      JLT  NULIN        IF LESS, WE HAVE A NULL INPUT
      BLWP @VSBR        READ THE CHARACTER
      CB   R1,@EDGE     IS THAT EDGE?
      JEQ  DECZER       IF SO, JUMP
      CB   R1,@SPACE    IS IT A SPACE
      JNE  SETEND       IF NOT, JUMP AHEAD
DECZER DEC  R0          ELSE MOVE R0 BACK ONE BYTE
      JMP  FLCLP        THEN JUMP TO REPEAT
SETEND
      MOV  R0,@ENDOC    R0 MARKS END OF INPUT CONTENT
      MOV  @STDOC,R0    GET START OF INPUT FIELD
      LI   R9,TEMSTR+1  POINT AT TEMP STRING PLUS ONE
      CLR  R2           CLEAR REG 2
RDLP
      BLWP @VSBR        READ ONE CHARACTER
      CB   R1,@EDGE     COMPARE TO EDGE
      JEQ  SKPRD        IF EQUAL, SKIP IT
      SB   @OFFSET,R1   SUBTRACT OFFSET
      MOVB R1,*R9+      MOVE TO TEMSTR LOCATION POINTED BY R9
      INC  R2           INC COUNT OF CHARACTERS
SKPRD  INC  R0          POINT AT NEXT SCREEN LOCATION
      C    R0,@ENDOC    COMPARE TO END OF INPUT
      JGT  STROUT       IF GREATER, JUMP
      JMP  RDLP         ELSE READ MORE
NULIN  CLR  R2          SET R2 TO 0
STROUT
      SWPB R2           SWAP BYTES
      MOVB R2,@TEMSTR   MOVE TO LENGTH BYTE AT TEMSTR
      MOV  @NUMFLG,R4   GET NUMERIC FLAG
      JNE  NUMOUT       IF NOT ZERO, JUMP
      CLR  R0           ELSE CLEAR R0
      LI   R1,6         PARAMETER 6
      LI   R2,TEMSTR    POINT AT TEMSTRING
      BLWP @STRASG      ASSIGN TO VARIABLE IN XB
      JMP  EXIT         THEN EXIT ROUTINE
NUMOUT SWPB R2          SWAP SO R2=LENGTH
      MOV  R2,R2        MOVE R2 TO ITSELF
      JNE  GDNUM        IF NON-ZERO, GOOD NUMBER
      CLR  @FAC         ELSE CLEAR FLOATING POINT ACCUMULATOR
      CLR  @FAC+2       AND THE SECOND WORD
      JMP  ASG          THEN JUMP AHEAD
GDNUM  LI   R0,>1000     POINT R0 AT >1000
      MOV  R2,R9        MOVE R2 INTO R9
      LI   R1,SV1000    POINT AT TEMPORARY STORAGE
      LI   R2,33        33 CHARACTERS
      BLWP @VMBR        READ 33 CHARACTERS
      LI   R1,TEMSTR+1  POINT AT CONTENT OF STRING
      MOV  R9,R2        GET LENGTH BACK INTO R2
      BLWP @VMBW        WRITE STRING TO VDP
```

THE ART OF ASSEMBLY—

```
      MOV  R0,@FAC12    PLACE ADDRESS >1000 AT FAC+12
      A    R2,R0        ADD LENGTH TO R0
      MOVB @ANYKEY,R1   PUT A SPACE IN R1
      BLWP @VSBW        WRITE A SPACE AFTER STRING IN VDP
      BLWP @XMLLNK      USE XML LINK
      DATA CSN          TO CONVERT TO NUMBER
ASG   CLR  R0           CLEAR R0
      LI   R1,6         6TH PARAMETER
      BLWP @NUMASG      ASSIGN NUMBER AT FAC TO XB VARIABLE
      LI   R0,>1000     POINT AT >1000
      LI   R1,SV1000    STASHED CONTENT
      LI   R2,33        33 BYTES
      BLWP @VMBW        WRITE BACK
EXIT  LWPI GPLWS        LOAD GPL WORKSPACE
      B    @>006A       BRANCH TO GPL INTERPRETER
ERREX
      BLWP @PRSTR       PRINT ERROR MESSAGE
      LI   R0,23*32+3   SET ROW 24, COL 3
      LI   R1,PAK       POINT TO "PRESS ANY KEY"
      BLWP @PRSTR       DISPLAY THAT
KEY   BLWP @KSCAN       SCAN KEYBOARD
      CB   @ANYKEY,@STATUS HAS KEY BEEN PRESSED?
      JNE  KEY          IF NOT, SCAN AGAIN
      LI   R0,22*32     ELSE SET TO ROW 23
      LI   R2,64        TWO ROWS TO WRITE
      LI   R1,SAVBOT    FROM SAVBOT
      BLWP @VMBW        WRITE BACK
      CLR  R0           CLEAR R0
      MOVB @ARGNUM,R1   GET NUMBER ARGS
      SRL  R1,8         RIGHT JUSTIFY
      INC  R1           INC - NON EXISTENT PARAMETER
      BLWP @NUMASG      ASSIGN NUMBER
      JMP  EXIT         THEN JUMP TO EXIT
VNG
      BL   @CLRBOT      CLEAR BOTTOM OF SCREEN
      LI   R0,22*32+5   POINT AT ROW 23, COL 6
      LI   R1,ROWSTR    AT ROW MESSAGE
      JMP  ERREX        JUMP BACK
COLNG
      BL   @CLRBOT
      LI   R0,22*32+3   ROW 23, COL 4
      LI   R1,COLSTR    COLUMN ERROR
      JMP  ERREX        JUMP
PRMNG
      BL   @CLRBOT
      LI   R0,22*32+3   ROW 23, COL 4
      LI   R1,PTLSTR    PROMPT ERROR
      JMP  ERREX
CHRNG
      BL   @CLRBOT
      LI   R0,22*32+3
      LI   R1,TMCSTR    TOO MANY CHARACTERS (OVER 255)
      JMP  ERREX
PTNG
      BL   @CLRBOT
      LI   R0,22*32+4
      LI   R1,PARSTR    WRONG PARAMETER TYPE
      JMP  ERREX
*
* FOLLOWING ARE SUBROUTINES USED BY THE
* MAIN CODE SECTION ABOVE
*
CLRBOT
      LI   R0,22*32     POINT AT ROW 23, COL 1
      LI   R1,SAVBOT    AND AT MEM LOCATION
      LI   R2,64        TWO ROWS
      BLWP @VMBR        READ WHAT'S THERE
      LI   R1,BLNKLN    POINT AT BLANK LINE
      SRL  R2,1         CUT R2 IN HALF
      BLWP @VMBW60      WRITE ONE BLANK ROW
      A    R2,R0        ADD ONE ROW
      BLWP @VMBW60      WRITE ANOTHER
      RT
CLRXB CLR  @STATUS      SUBROUTINE
      BLWP @GPLLNK      CLEAR
      DATA >27E3        TO XB DEFAULTS
      RT                RETURN
*
* PRSTR UTILITY - COURTESY HARRY WILHELM
*
PRSTR  DATA >2038,PRSTR1 UTILITY BLWP VECTOR
VMBW60 DATA >2038,VMBW61 UTILITY BLWP VECTOR
*
PRSTR1 BL   @>24CA      USE A SUBROUTINE TO PASS PARAMETERS
FROM CALLING WS
      MOVB *R1+,R2      GET STRING LENGTH BYTE INTO R2
      SRL  R2,8         RIGHT JUSTIFY
      JEQ  VMBW6X       IF ZERO, SKIP THE STRING, IT HAS NULL LENGTH
      JMP  VMBW62       ELSE JUMP AHEAD
VMBW61 BL   @>24CA      USE SUBROUTINE TO PASS PARAMETERS
VMBW62 MOVB *R1+,R3     MOVE A BYTE INTO R3
      AI   R3,>6000     ADD THE OFFSET FOR XB
      MOVB R3,@>8C00    PLACE AT VDPWD LOCATION
      DEC  R2           DECREMENT CHARACTER COUNT
      JNE  VMBW62       IF NOT ZERO, SEND ANOTHER CHARACTER
VMBW6X RTWP             ELSE RETURN TO CALLERS WS AND CODE
*  GENERAL PURPOSE GPL LINK
*  FOR USE UNDER EXTENDED BASIC
*
* (INCLUDE HERE THE GPLLNK SHOWN ABOVE)
* (BE SURE NOT TO COPY THE END DIRSCTIVE)
*
* FOLLOWING IS THE DATA SECTION
* FOR THE ABOVE CODE
*
EDGE   DATA >7F7F,>7F7F  EDGE CHARS WITH OFFSET
WS     BSS  32          OUR WORKSPACE
ENDOC  DATA 0           STORAGE FOR END OF INPUT FIELD
STDOC  DATA 0           STORAGE FOR START OF INPUT
STPRM  DATA 0           START OF PROMPT
ROW    DATA 0           ROW OF SCREEN
COL    DATA 0           COLUMN OF SCREEN
ONE    DATA 1           THE NUMBER ONE
NUMFLG DATA 0           FLAG FOR NUMERIC VARIABLE
THIR2  DATA 32          THIRTY TWO
DEFFLG DATA 0           FLAG FOR DISPLAY OF DEFAULT
LASPOS DATA 0
DEFLEN DATA 0           LENGTH OF DEFAULT STRING
PRMLEN DATA 0           LENGTH OF PROMPT
TWEN8  DATA 28          TWENTY-EIGHT
MAXCHR DATA 0           MAX CHARACTERS TO ACCEPT
TOTROW DATA 0           TOTAL ROWS TO BE USED
TOTLEN DATA 0           TOTAL LENGTH
CLRFLG DATA 0           FLAG FOR SCREEN CLEARING
BLNKLN TEXT '                              ' 32 SPACES
PRMSTR BSS  30          STORAGE FOR PROMPT STRING
TEMSTR BSS  256         STORAGE FOR IN-OUT STRING
MINUS  TEXT '-'         MINUS SIGN
SPACE  BYTE >80         SPACE CHARACTER WITH OFFSET
OFFSET BYTE >60         OFFSET FOR BASIC
MAXLEN BYTE 255         MAXIMUM STRING LENGTH
ANYKEY BYTE 32          KEY STROKE COMPARISON
TMCSTR BYTE 26          LENGTH OF MESSAGE
      TEXT 'CHARACTERS 1 THRU 255 ONLY'  MESSAGE
PAK    BYTE 25          LENGTH
      TEXT 'PRESS ANY KEY TO CONTINUE' MESSAGE
PTLSTR BYTE 25          LENGTH
      TEXT 'PROMPT STRING IS TOO LONG' MESSAGE
ROWSTR BYTE 21
      TEXT 'ROW MUST BE 1 THRU 24'
COLSTR BYTE 24
      TEXT 'COLUMN MUST BE 1 THRU 28'
PARSTR BYTE 23
      TEXT 'PARAMETER IS WRONG TYPE'
SAVBOT BSS  31          MEMORY TO SAVE SCREEN BOTTOM TWO ROWS
SV1000 BSS  33          SAVING >1000 BYTES
      END
```

Utility Bill Audit

XB program puts you in charge of your utility bills

By HARLEY RYAN JR.

Utility Bill Audit is a versatile Extended BASIC program that lets you check your electric,gas, water, and phone bills for accuracy, or split the costs of these bills among the people living in your household. Also,if you are interested in energy savings, you can monitor your daily electric and gas consumption with this program.

PERSONALIZING THE PROGRAM

Before you run this program, it is necessary to have a thorough understanding of how each bill is calculated in the program.

First,a particular bill is split up according to the values (2, 1, 2, 4) given in the DATA statement in line 1640. These values are assigned to the variable N(X) and represent the number of individuals who must pay for each bill. In its present form, the program assumes that the electric and water bills will be paid for by two individuals,the phone bill by four, and the gas bill by one individual. However, it's unlikely that these numbers will correspond to the financial arrangements in your household. So,be sure to substitute the appropriate values in this line before you continue. If the bills are paid for by one individual, simply replace the numbers in line 1640 with 1, 1, 1, 1.

Since the program works on the actual costs of your utility bills based on local rates,certain information about these rates must be provided before you can run the program. This information is READ in lines 380 and 400 from the DATA statements in lines 1650-1680.

Notice that the first three DATA statements in this sequence have nine entries and apply to the electric, gas and water bills, respectively. Let's consider line 1650 as an example. The first entry in this line is the name of the utility (ELECTRIC) for which the rates that follow apply. The second entry is the unit of measurement for that particular utility (KWH for kilowatt hours). The next entry is the minimum service charge for the utility ($5.40 for electricity). The fourth entry is the tax rate based on the sum of the service charge and the rate charge (zero percent for electric use). These first four DATA entries are READ in as A$(1), B$(1), M(1) and Z(1), respectively.

At this point,the numbers begin to get a little confusing,so read carefully.

The next two numbers are cutoff limits for each electric rate and are represented in the program by L1(1),and L2(1). The last three numbers are the actual rates charged per kKW use for each level of usage R1(1), R2(1), and R3(1) in the program. Thus, the program is presently set up so that the rate charged for electricity is $.0495 for the first 350 KWH, $.0565 for the next 950 KWH (i.e., 1300 minus 350), and $.0541 for any usage exceeding 1300 KWH.

The DATA statement in line 1680 is easier to follow: the utility (PHONE), the minimum service charge ($13.50), and the tax rate on the service charge and long distance calls (3 percent).

So, get out your most recent bills and read off the various rate (per KWH for electric, per CCF or hundred cubic feet for gas and water). If the rates are not given on your bill, contact the utility company to get a schedule of the latest rates. Then just substitute your local rates for those in the DATA statements in lines 1650 to 1680.

PROGRAM OPERATION

After inserting the correct rates, run the program. You will then be asked which utility bill you wish to check. The first three menu choices are Electric, Gas and Water. Bills for these three utilities are all calculated in the routine beginning at line 560.Let's look at an electric bill as an example.

When the routine at line 560 is executed, you will be required to INPUT the present and previous meter readings. These values can be read directly from your latest electric bill. Next, you must INPUT the number of days in the billing period. Then you will be asked to INPUT any adjustments to the bill, either positive (for example, connection fees, previous balances) or negative (credits).

The program will next calculate the amount of electricity consumed for the given period (defined as U in line 720). Then, depending on the value of U relative to the two rate limits, L1(1)

(See Page 13)

UTILITY BILL AUDIT—

(Continued from Page 12)

and L2(X) (lines 740 and 750), an amount owed (T) before tax and adjustments will be calculated (lines 760, 780 and 800). Next, the tax on this amount will be determined (T1). And finally a total electric bill - the sum of the minimum charge,usage cost,tax, and adjustments - will be calculated (T3) in line 830.

The results are then PRINTed on the screen with provisions for formatting the output to two places past the decimal. Any numbers in the third place past the decimal are simply dropped. If you prefer rounded numbers, you could easily modify the program to achieve that.

The routine beginning at line 560, as mentioned, also calculates the gas and water bills. These are based on the rates READ from the DATA statements in lines 1660 and 1670. Notice the sets of large numbers (precisely, 99999) in line 1660. The rates for gas where I live are the same, regardless of the amount used. By using large numbers here for the cutoff limits, L1(2) and L2(2), for this utility, it's unlikely that the actual usage will exceed these amounts (see lines 740 and 750). Thus, the charge for this commodity will always be based on the first rate, or R1.

The rates for water, as READ from the DATA statement in line 1670, are based on a single cutoff limit (L1(3)) of 1000 CCF. For less than this, a usage rate (R1(3)) of $0.144 per CCF is charged. If water usage exceeds 1000 CCF, a second rate (R2(3)) of $0.160 is charged. Again, using a very large number (99999) for the second cutoff limit (L2(3)) assures that the overall usage cost is based only on two rates.

ANALYZING THE PHONE BILL

Phone bills are checked in a separate routine in the program beginning at line 1110. In this routine, adjustments to the bill are initially INPUT in the same manner as they were with the electric, gas and water bills. Next, the person responsible for each long distance charge is required to INPUT the amount of each long distance call. A separate routine (lines 1260-1360) allows the individual to correct any typing mistakes. Finally, the amount owed by one individual is displayed.

The portion of the phone bill that each person must pay is the sum of their long distance tolls, a proportional amount of both the service charge and the billing adjustments, and a proportional amount of the tax levied on the service and long distance calls. Again, if only one person in the household foots the bills, the last number in line 1640 should be 1.

In addition to enabling you to catch billing errors and helping you to easily divide up household bills, this program can help you monitor your costs. If you add an energy-saving device that is supposed to save, say, 10 percent of your total electric bill, take a meter reading when it is installed and verify the savings with a later reading. You can also project weekly, monthly and yearly savings for any utility in this manner.

UTILITY BILL AUDIT

```
100 ! Utility bill program,o
riginal source unknown,name
Larry I.Bihlmeyr on article,
modified by H.Ryan of CONNI,
 April 7,1994 !097
110 CALL CLEAR :: CALL SCREE
N(5):: FOR C=0 TO 14 :: CALL
 COLOR(C,16,1):: NEXT C !216
120 DIM A$(4),B$(4),L1(3),L2
(3),M(4),R1(3),R2(3),R3(3),W
(50),Z(4)!118
130 ON WARNING NEXT :: GOTO
300 !213
140 A1=0 !041
150 DISPLAY AT(8,3):"ENTER A
DJUSTMENTS TO BILL": :"   (+
 or -, '0' WHEN DONE)" !017
160 ACCEPT AT(R,14)BEEP VALI
DATE(NUMERIC):E :: R=R+1 !01
5
170 A1=A1+E !168
180 IF E=0 THEN 200 !193
190 GOTO 160 !239
200 GOSUB 260 !085
210 RETURN !136
220 CALL CLEAR !209
230 RETURN !136
240 DISPLAY AT(8,CC):A$(X);"
 BILL (CONT')" !175
250 RETURN !136
260 DISPLAY AT(24,3):"PRESS
'C' TO CONTINUE C" !090
270 ACCEPT AT(24,26)BEEP VAL
IDATE("C")SIZE(1):C$ !232
280 GOSUB 220 !044
290 RETURN !136
300 FOR I=1 TO 4 :: READ N(I
):: NEXT I !185
310 FOR I=1 TO 3 !058
320 READ A$(I),B$(I),M(I),Z(
I),L1(I),L2(I),R1(I),R2(I),R
3(I)!142
330 NEXT I !223
340 READ A$(4),M(4),Z(4)!071
350 GOSUB 220 !044
360 DISPLAY AT(10,5):"UTILIT
Y BILL PROGRAM" !026
370 DISPLAY AT(14,8):"1. ELE
CTRIC BILL":TAB(8);"2. GAS B
ILL":TAB(8);"3. WATER BILL"
!172
380 DISPLAY AT(17,8):"4. PHO
NE BILL":TAB(8);"5. ALL OF A
BOVE":TAB(8);"6. EXIT": :TAB
(9);"<-CHOICE" !191
390 ACCEPT AT(21,8)BEEP VALI
DATE(DIGIT)SIZE(1):P !123
400 IF (P<1)+(P>6)THEN 390 !
055
410 ON P GOTO 760,780,800,82
0,1220,1460 !138
420 GOSUB 220 !044
430 L=LEN(A$(X))+5 :: C=(28-
L)/2 :: LL=LEN(A$(X))+12 ::
CC=(28-LL)/2 !216
440 DISPLAY AT(8,C):A$(X);"
BILL" !152
450 DISPLAY AT(10,1):"PREVIO
US MTR READING" !017
460 ACCEPT AT(10,22)BEEP VAL
IDATE(DIGIT)SIZE(7):E1 !210
470 DISPLAY AT(11,1):"PRESEN
T  MTR READING" !214
480 ACCEPT AT(11,22)BEEP VAL
IDATE(DIGIT)SIZE(7):E2 !212
490 DISPLAY AT(12,1):"DAYS I
N THIS BILL ->" !119
500 ACCEPT AT(12,22)BEEP VAL
IDATE(DIGIT)SIZE(2):D !157
510 GOSUB 260 :: GOSUB 220 :
: GOSUB 240 :: GOSUB 140 !03
5
520 U=E2-E1 !194
530 Y=U/D !116
540 IF U>L2(X)THEN 600 !175
550 IF U>L1(X)THEN 580 !154
560 T=M(X)+R1(X)*U !069
```

(See Page 14)

UTILITY BILL AUDIT—

(Continued from Page 13)

```
570 GOTO 610 !179
580 T=M(X)+R1(X)*L1(X)+R2(X)
*(U-L1(X))!197
590 GOTO 610 !179
600 T=M(X)+R1(X)*L1(X)*(L2(X
)-L1(X))+R3(X)*(U-L2(X))!062
610 T1=T*Z(X)!121
620 T2=T+T1 !222
630 T3=T2+A1 !254
640 GOSUB 220 :: GOSUB 240 !
238
650 DISPLAY AT(10,1):USING 1
320:INT(U*100)/100,B$(X)!210
660 DISPLAY AT(11,1):USING 1
330:INT(Y*100)/100 !003
670 DISPLAY AT(12,1):USING 1
340:INT(T2/D*100)/100 !067
680 DISPLAY AT(13,1):USING 1
350:INT(T*100)/100 !020
690 DISPLAY AT(14,1):USING 1
360:INT(T1*100)/100 !080
700 DISPLAY AT(15,1):USING 1
370:A1 !205
710 DISPLAY AT(16,1):USING 1
380:INT(T3*100)/100 !104
720 IF N(X)=1 THEN 740 !174
730 DISPLAY AT(17,1):USING 1
390:N(X),INT(T3/N(X)*100)/10
0 !016
740 PRINT !156
750 GOSUB 260 :: RETURN !095
760 X=1 :: R=13 !208
770 GOSUB 420 :: GOTO 360 !0
47
780 X=2 :: R=13 !209
790 GOSUB 420 :: GOTO 360 !0
47
800 X=3 :: R=13 !210
810 GOSUB 420 :: GOTO 360 !0
47
820 GOSUB 220 !044
830 X=4 :: R=13 !211
840 DISPLAY AT(7,9):A$(X);"
BILL" !086
850 PRINT !156
860 GOSUB 140 !220
870 FOR K=1 TO N(X)!083
880 I=1 !001
890 IF N(X)=1 THEN 910 !089
900 DISPLAY AT(5,8):"FOR PER
SON #";K !089
910 R=13 :: DISPLAY AT(7,4):
"ENTER CHARGE FOR EACH":TAB(
5);"LONG DISTANCE CALL": :TA
B(4);"(ENTER '0' WHEN DONE)"
 !101
920 ACCEPT AT(R,12)BEEP VALI
DATE(NUMERIC):W(I):: R=R+1 !
213
930 IF W(I)=0 THEN 960 !132
940 I=I+1 !011
950 GOTO 920 !234
960 GOSUB 220 !044
970 DISPLAY AT(19,8):"PERSON
 #";K !132
980 FOR J=1 TO I-1 !068
990 PRINT USING 1400:J,W(J)!
143
1000 PRINT !156
1010 DISPLAY AT(24,1):"IS TH
IS CORRECT (Y/N)" :: ACCEPT
AT(24,22)BEEP VALIDATE("YN")
SIZE(-1):C$ !103
1020 IF C$="Y" OR C$="" THEN
 1050 !007
1030 PRINT "TYPE IN CORRECTI
ON" !134
1040 INPUT W(J)!160
1050 PRINT !156
1060 NEXT J !224
1070 GOSUB 220 !044
1080 L=LEN(A$(X)):: CC=L !13
6
1090 GOSUB 240 !064
1100 T=0 !011
1110 FOR J=1 TO I-1 :: T=T+W
(J):: NEXT J !093
1120 DISPLAY AT(10,4):USING
1410:INT(M(X)/N(X)*100)/100
!229
1130 DISPLAY AT(11,4):USING
1420:T !224
1140 DISPLAY AT(12,4):USING
1430:INT(A1/N(X)*100)/100 !0
91
1150 T1=T+INT(M(X)/N(X)*100)
/100 !188
1160 T2=INT(T1*Z(X)*100)/100
 !038
1170 DISPLAY AT(13,4):USING
1440:T2 !040
1180 DISPLAY AT(14,4):USING
1450:T1+T2+INT(A1/N(X)*100)/
100 !254
1190 PRINT :: GOSUB 260 !115
1200 NEXT K !225
1210 GOTO 360 !184
1220 FOR F=1 TO 3 !055
1230 X=F :: R=13 !028
1240 GOSUB 420 !245
1250 NEXT F !220
1260 GOTO 820 !134
1270 DATA 2,1,2,4 !153
1280 DATA ELECTRIC,KWH,5.40,
0,350,1300,.0495,.0565,.0541
 !215
1290 DATA GAS,CCF,4.05,0,999
99,99999,.49541,0,0 !189
1300 DATA WATER,CCF,3.26,0,5
00,99999,.144,.160,0 !007
1310 DATA PHONE,13.50,.03 !0
96
1320 IMAGE "USE THIS PERIOD
 ->:#### ###" !168
1330 IMAGE "COST THIS PERIOD
 ->:$###.##" !191
1340 IMAGE "COST PER DAY IS
 ->:$###.##" !101
1350 IMAGE "W/OUT TAX
 ->:$###.##" !182
1360 IMAGE "TAX IS
 ->:$###.##" !052
1370 IMAGE "ADJUSTMENT
 ->:$###.##" !010
1380 IMAGE "TOTALS
 ->:$###.##" !098
1390 IMAGE "SPLIT # WAYS
 ->:$###.##" !254
1400 IMAGE "CALL No #>$##.##
" !210
1410 IMAGE "SERVICE    ->:$#
#.##" !146
1420 IMAGE "LD CALLS   ->:$#
#.##" !128
1430 IMAGE "ADJUSTMENT ->:$#
#.##" !032
1440 IMAGE "TOTAL TAX  ->:$#
#.##" !210
1450 IMAGE "TOTAL BILL ->:$#
#.##" !232
1460 END !139
```

Living with spiders — Part II

Programming with Funnelweb

By TONY MCGOVERN

In this episode we will look at some more aspects of writing programs to co-exist with the Funnelweb system. This time it will be on programs which make extensive enough use of memory that any other code is obliterated. A typical example is a disk manager program, where every spare byte is needed for buffers, let alone the program code. The idea of a dual-mode program is less compelling here, but experience with fully adapted programs shows that it is worthwhile. The code excerpts used to illustrate will be drawn from the modifications made to the Ottawa UG's DM-1000 fairware program to make it Funnelweb-aware.

The idea of a dual-mode program is less compelling here, but experience with fully adapted programs shows that it is worthwhile.

So the necessary steps are

1. Extract load-time details
2. Avoid treading on Atrax R.
3. Arrange FWB re-entry

The reasons for (1) are much the same as they were in Part I of this saga. The only thing really left under (2) is to avoid trashing the Mailbox unnecessarily. Item #3 is almost more psychological than real. FWB is so easy to reload that going through the title screen is hardly any more work for the user than setting up a direct return. There are benefits in direct reload though such as color continuity and retaining of the character set.

```
*  FUNNELWEB System block equates

FWENTR EQU  >E006
SVGPRT EQU  >FF14
RDISK  EQU  >FF18
BTFLAG EQU  >FF1A
INCOL  EQU  >FF26
MODFLG EQU  >FF5A
CMSRET EQU  >FF5C
GRMAD  EQU  >FF5E
NAMBUF EQU  >FF62
LDR11  EQU  >FF9C

*  DM-1000 equates

PAB    EQU  >0FE0
VBUF   EQU  >1000

FWFLAG DATA >0
*
INDSK  DATA >0        Keep
FCOLRS DATA >0        in
SVGPTN DATA >0        just
SVGRAD DATA >0        this
SVMODF DATA >0        very
RFDISK DATA >3131     order.

NULL   BYTE >00
FIVE   BYTE >05
HFF    BYTE >FF
HEXAA  BYTE >AA
COLBUF BYTE >F4

       EVEN

*  Initial entry point

FWSTAT LIMI 0
       C    R11,@LDR11
       LWPI MYREG1
       JNE  NOTFWL

*  FWB load path

       SETO @FWFLAG

*  Retrieve color info

       LI   R0,>380
       BLWP @VSBR
       MOVB R1,@COLBUF
       LI   R1,FCOLRS
       MOV  @INCOL,*R1+

*  Save system details

       MOV  @SVGPRT,*R1+
       MOV  @GRMAD,*R1+
       MOV  @MODFL,*R1+

*  Save current load paths

       MOV  @RDISK,*R1
       MOVB @NAMBUF+4,@DEFDRV
       JMP  MGRST

*  On to DM-1000 regular entry

NOTFWL EQU  $
       ..
       ..
```

(See Page 16)

LIVING WITH SPIDERS—

(Continued from Page 15)

This code illustrates a few more features than would be necessary in a minimal reloader. All that is strictly necessary is to reload UTIL1 or FW as a program file and branch to it normally, but here we are also going to cater for XB/SSSD users who might wish to have only LOAD on their working disk, and not UTIL1 as well. The FWB code is buried in LOAD so that when XB loads LOAD the assembly code is in its correct position in hi-mem. So the strategy adopted with LOAD is to load it into VDP as if it were any old program file, and then to search for the start of the assembly code, before VMBR'ing this to where it belongs. This means that the normal entry code in LOAD which finds return, GROM, and XML addresses is bypassed so these are saved from the previous time FWB was in control. If you were to do this for UTIL1/FW as well, you would also have to clear the QDFLAG at >FF52 because one of the functions of the lead-in code of FW/UTIL1 is to put the imbedded code for QD in its correct place.

As in Part I the first task is to see if it was loaded from FWB. If not just go to the normal entry code. If so we set the flag and read the information from FWB. The minimum necessary is FWB's internal color pointer at INCOL (>FF26) so that when FWB is reloaded this can be reset to what it was when FWB was previously exited. If it is not reset FWB will revert to its configured first value which may not be the one you were last using. The next 3 items are only absolutely needed for re-entry by way of LOAD, and contain the GPL return, GROM address to be reset, and the XML address corresponding to that GPL XML instruction in GROM (see FWDOC/REPT) which is also used as an implicit flag for the module in use.

Location RDISK (>FF18) in FWB contains the FWB primary and secondary disk drive numbers in ASCII form, and are saved so that preloaded re-entry prompts may be made. The next item fetches the drive number from the FWB loader name buffer at NAMBUF (>FF62). This is used by DM-1000 for saving its configuration information back to disk, and since it is available in definite form without needing a search we might as well use it.

```
*  FUNNELWEB VN 4.1 Reloader
*  As used in DM-1000 for FWB

ILOAD  LWPI MYREG1
       BLWP @CLRSCN

*  Screen messages

FWBLOD BLWP @DSPTXT
       DATA 6
       DATA 47,TXTCA6,26
       DATA 206,TXTI1,20
       DATA 486,TXTI2,15
       DATA 566,TXTI3,15
       DATA 502,TXTDRV,9
       DATA 582,TXTDRV,9

*  Get drive numbers

IDR    LI   R0,512
       MOVB @RFDISK,R1
       BLWP @GTSKEY
       MOVB R1,@INDSK
       MOVB R1,@FLDISK

       LI   R0,592
       MOVB @RFDISK+1,R1
       BLWP @GTSKEY
       MOVB R1,@INDSK+1
       MOVB R1,@FWDISK
       MOVB R1,@UTDISK

*  Load FWB from nominated disk
*  Try FW first

LDFWR  EQU  $
       LI   R1,FWPDAT
       BL   @FINDFW
       JMP  FWRITR

*  UTIL1 next

       LI   R1,UTPDAT
       BL   @FINDFW
       JMP  FWRITR

*  Try XB LOAD then

       LI   R1,FWLDAT
       BL   @FINDFW
       JMP  LODCHK

*  Error return

LDFAIM MOV  @FWFLAG,R0
       JNE  LDFAIN

       BLWP *R0
LDFAIN B    @ILOAD
```

This next block of code handles the reloading of UTIL1/FW or LOAD. First there are housekeeping details, and then the screen is written up. GTSKEY is a standard DM-1000 routine which accepts a single key with default shown on screen. The nominated drives are stored in INDSK and written into the PAB data for reloading. The E/A side has to be written into PAB data for both FW and UTIL1. It was thought simpler just to repeat the whole PAB. A common load routine FINDFW, code given further on, is used for all versions and success goes to the immediately following JMP, with failure dropping through. In turn it looks for FW on the E/A side drive, and UTIL1 there also. This specific order of sear[illegible] allows the filename UTIL1 to be used for other purposes as it is a name in much demand, and it is the policy of FWB to be as as much an invisible hand in the background as it can.

(See Page 17)

LIVING WITH SPIDERS—

(Continued from Page 16)

If both of these fail then it tries for LOAD on the TI-Wr side drive. If this also fails the error return is also taken. If FWFLAG is set it is all tried again, and if not it returns to the title screen. Writing it as shown saves a word of code as R0 of necessity contains >0000.

```
*  Load FW/UTIL1 into place

FWRITR LI   R0,VBUF+6
       LI   R1,FWENTR
       LI   R2,>FFD8-FWENTR
       BLWP @VMBR
       MOV  @INDSK,@RDISK
       MOV  @FCOLRS,@INCOL
       CLR  @BTFLAG
       B    @FWENTR
```

When UTIL1 or FW is loaded a simple VMBR of the code into place is followed by rewriting the drive numbers and color pointer, turning off the boot tracking flag at BTFLAG (>FF1A), and a direct branch to FWENTR. This is a extra safety precaution in case you have a copy of the FWB system with boot tracking enabled, but temporarily residing on a deviant device such as a Myarc RAMdisk, not supported by the boot tracking code. We have seen a flawed attempt to do this in which the interrupt hook was loaded and pointing into the FW entry code area with interrupts still on, even though the programmer had been careful not to have the code with the VMBR destroyed by the incoming block from VDP. There is no way any program can survive this even if its first instruction disables interrupts.

```
*  Search for start of FWB in LOAD

LODCHK EQU  $
       LI   R7,>1A00
LDCK20 MOV  R7,R0
       INC  R7
       CI   R7,>2400
       JHE  LDFAIM

       BLWP @VSBR
       CB   R1,@HEXAA
       JNE  LDCK20

*  Set up transfer addresses

       STWP R1
       AI   R1,8
       LI   R2,4
       BLWP @VMBR
       CI   R4,>AAAA
       JNE  LDCK20

       CI   R5,>000A
       JNE  LDCK20

*  Fetch relevant part of LOAD
*  Start CPU pointer to R3

       AI   R7,-3
       MOV  R7,R0
       STWP R1
       AI   R1,6
       LI   R2,2
       BLWP @VMBR

*  Calculate length to transfer

       LI   R2,>FFD8
       S    R3,R2
       INCT R7
       MOV  R7,R0
       MOV  R3,R1
       BLWP @VMBR
```

The code after LODCHK looks for where the FWB code is located in LOAD. When this code was first written the LOAD program still contained substantial and variable amounts of XB code, so a wide search range has been left. FWB 4.1 hardly needs this range since the form of the program is now largely fixed, but it is a good idea to leave it flexible. The search uses the knowledge that the start of the main block of FW code has form IDPTR,>AAAA,>000A,.... an ident word >AAAA at IDPTR (=$+2) immediately followed by permanent program data. These are not necessarily on word boundaries in the LOAD file, as XB is byte oriented.

The program searches VDP until it finds byte >AA and then reads that and the next 3 bytes into R4,R5 and checks them there. If the search fails the error exit is taken. If successful the VDP pointer in R7 is backed off to IDPTR and these bytes read into R3 as temporary. The length is then figured from the known endpoint and the transfer to CPU made.

```
*  Restore FW to match previous

FWEXIT EQU  $
       LI   R1,INDSK
       MOV  *R1+,@RDISK
       MOV  *R1+,@INCOL
       MOV  *R1+,@SVGPRT
       MOV  *R1+,@GRMAD
       MOV  *R1+,@MODFL

*  Hand over to FWEB

       LWPI FWREGS
       CLR  R4
       SETO R13
       MOV  @CMSRET,R11
       RT
```

The values that would be established by the normal FWB entry

(See Page 18)

LIVING WITH SPIDERS—

(Continued from Page 17)

code are now refreshed and an exit made to the TI-Wr side of the central menu screen, as discussed in Part I of this minor opus. The loader BL subroutine and various PAB data follow.

```
*  Setup DSR and do it

FINDFW EQU  $
       LI   R0,PAB
       LI   R2,>20
       BLWP @VMBW
       LI   R0,PAB+9
       MOV  R0,@>8356
       BLWP @DSRLNK
       DATA 8
       JNE  FWSUCC
       INCT R11
FWSUCC RT

*  PAB data for reloading FWB

FWPDAT DATA >0500,VBUF,>0,>2200
       DATA >07
       TEXT 'DSK'
FWDISK TEXT '1.FW'
       EVEN

UTPDAT DATA >0500,VBUF,>0,>2200
       DATA >0A
       TEXT 'DSK'
UTDISK TEXT '1.UTIL1'
       EVEN

FWLDAT DATA >0500,VBUF,>0,>27D0
       DATA >09
       TEXT 'DSK'
FLDISK TEXT '1.LOAD'
```

A possibility implicitly not considered in the previous discussion is that the program being loaded might destroy part or all of the FWB system block at the top of hi-mem. In that case the simplest way out is just to return to the title screen, as there is then no immediate way to check if the program was loaded from FWB. Otherwise a specific option to return to FWB may be included which reloads UTIL1 or FW and branches to its normal entry at FWENTR (>E006).

Geneve and MDOS

MY-BASIC and the internal RAMdisk

By JIM UZZELL

This article is intended for Geneve users who have a standard Geneve and no "goodies" in the PE box.

Regardless of how many disk drives you currently have you can add one more — the internal RAMdisk.

First you will have to reread some articles I wrote. The first one was published in May 1990. In that article I indicated that MY-BASIC had a maximum 200K memory allocation. That is no longer true on a non-modified Geneve. The maximum is now 192K.

The second article was published in June 1990, a program (MYMENU) that cataloged a disk then allowed you to run programs from the catalog. The memory allocation becomes important if you intend to set up an internal RAMdisk. MY-MENU is being used as an example of what follows.

We can start to set up our RAMdisk using a batch file. From the MDOS prompt, type ASSIGN and press enter. You should see a list saying disk A-D and HDOS E-G. This is the default configuration. Rather than change any of these we are going to add "H" to the list and call it disk 5.

Sample Batch File

```
REMAP 5E
LASTDRIVE = H
ASSIGN H=DSK5:
RAMDISK 127
COPY E:MB.LOAD H:
E:MB.ABASIC1 H:LOAD
```

First, read the section on remap in the file update that comes with MDOS 2.0. In that file, letter "E" is the RAMdisk, so the first item in the batch file will be remap 5E which says remap the internal drive to disk 5. In order to use drive H we must tell MDOS that it is there, so the next item for the batch file would be LASTDRIVE = H (the spaces are required). Now let's tell MDOS that H=DISK 5 by entering this item to batch file — ASSIGN H=DSK5: (the colon is required). Now let's set up the RAMdisk size. This is where memory allocation becomes important, because the size of the RAMdisk reduces the amount of total memory allocation you can use.

The maximum RAMdisk size is 127, which leaves 64K memory allocation for MY-BASIC, which is the default. Let's put the RAMdisk in the batch file: RAMDISK 127. Let's also add MY-MENU to the batch file (you did type in this program back in June 1990, didn't you?): COPY A:LOAD H: (I changed the name, you can use the name you save it

(See Page 19)

MY-BASIC AND RAMDISK—

(Continued from Page 18)

to). Now load MY-BASIC and our load program by adding this to our batch file — A:ABASIC1 H:LOAD.

The following is a summary of what we have done:

REMAP 5E
LASTDRIVE = 5
ASSIGN H=DSK5:
RAMDISK 127
COPY A:LOAD H:
A:ABASIC1 H:LOAD

I have used drive A in this example, but you can use any drive you have. Now save this batch file by typing COPY CON A:RAM then type in the items from the above summary. Then type CRTL Z to save.

Before we run this batch file we need to make one change to MYMENU. In line 140, change D$="12" TO D$="125". This adds the RAMdisk to the drive options. Run it by typing A:&RAM.

To see the size of RAMdisk change the A> prompt to H> then do a DIR for the number of sectors. Then do a CHKDSK for the number of bytes.

Warning: Any batch file that is run after setting up the internal RAMdisk where TIMODE is activated on an un-modified Geneve will destroy the RAMdisk.

One final note: those who have a modem, can download MYMENU+, an updated shareware version, from the 9640 News BBS (901-368-0112).

Have fun with your new drive.

Hardware project

Bring your Myarc RS232 into the nineties!

By JEFFREY H. WHITE and DAN H. EICHER

When the Myarc RS232 card first came out, people were impressed by the fact that this card could be opened at 19,200 bps. TI and CorComp had only listed OPEN statements up to 9,600 bps in their manuals. It was only later (after we had become much more sophisticated, right?) that we came to realize that the RS232s manufactured by the big three (TI, CorComp and Myarc) were all capable of very fast speeds. In the case of CorComp and TI cards, 19.2 bps, as it is commonly called, was not coded into the Device Service Routine bit rate tables, but they are actually capable of these speeds and more, if you are working in assembly.

What follows are two modifications that can be made to the Myarc RS232 card. The first is a simple fix that will make the second serial port function. The second is an upgrade/enhancement to increase the capability of your card.

Some users, myself included, have had difficulties using the second RS232 port on Myarc cards. This is because of a design screw-up. The original board layout did not properly implement the RS232 port. The DTR line from pins 18 and 19 was hooked up to an output of the quad line receiver (1489), and the corresponding input was then connected to the !CTS (pin 6) and !DSR (pin 7) inputs of the 2nd UART (9902A). It takes only three jumpers to fix this, but sometimes these jumpers were installed wrong or not at all by Myarc when shipping RS232 cards.

Here is what you need to do to ensure your card is modified to function properly with its second port:

The trace from pin 4 of the 1489 must be cut so there is no continuity with pins 6 and 7 of the closer 9902A. Still on the front of the card is a trace from pin 6 of the 1489 to a solder pad between pins 4 and 5 of the quad line driver (1488) that must be cut. On the back of the card, the trace from pin 6 of the 1489 to the leg of a 2.2k resistor must be cut. On the back, run a jumper from the solder pad between pins 4 and 5 of the 1488 (trace from pin 19 of the D-sub) to pin 4 of the 1489, then from that pin 4 to the leg of the 2.2k resistor. Jumper from pin 6 of the 1489 to either pin 6 or 7 of the closer 9902A.

If this is done correctly, your second RS232 port should work, given no bad solder joints or chips. The quickest and easiest way to check this modification out is to use a TI built Y-splitter cable for the RS232 and hook a modem with status lights up to the second port. If everything seems to work, you are set. Alternatively, if you have a Y-splitter cable and a breakout box, you can now use the breakout box to test functionality of the second port.

On to the enhancement:

As high speed modems have become commonplace, a few transitions in the typical use of RS232 have occurred — the first being in the area of speed. When the TI RS232 card was originally designed, one of its primary purposes was to interface to a printer — in fact, TI's impact printer (an Epson with the TI logo) was designed for serial usage. The maximum transfer rate for the typical printer of that time was around 120 characters per second, with each character requiring about nine bits. It was no real challenge for the RS232 port to keep up with the printer.

Very quickly, with the rise of the TIBBS BBS by Ralph Fowler, hooking up a modem became the thing to do. We have all seen the increase in modem speeds. First 300 baud, then 1200, followed shortly by 2400. But, only within the past year have 9600 and 14.4 baud modems become affordable to us common users.

While 300/1200/2400 bps modems present computers and serial ports with little challenge, the same cannot be said for speeds over 9600. Actually, a 9600 baud modem runs even faster, if you consider the modems' implementation of MNP5 and 42 bis `on the fly' data compression.

The first change that must occur to ac-

(See Page 20)

MYARC RS232 FIX—

(Continued from Page 19)

commodate faster communication speeds is your communication software. There are two primary methods of handling flow control. Flow control refers to the agreed upon way of deciding if the computer or the device it is talking to, a modem perhaps, is ready to send or receive. At low transmission speeds — under 9600 baud — the most common form of flow control is called XON/XOFF. This can be accomplished using only two wires: Transmit and Receive. This works based upon the practice that both devices have agreed upon characters they will both utilize for "OK to send" (XON) and "Whoa, hold on a sec" (XOFF).

This method does not work very well at high speeds. The problem is, that by the time one system actually "Whoas" after the other system sends a "Whoa, hold on a sec" message, a lot of data can pass under the bridge.

So, what do you do? You use some more of the pre-defined control wires in the RS232C definition. The wires most commonly used for this function are DTR and RTS. However, the current RS232 cards are wired like a modem, so the wires that we will use are properly called DSR and CTS. You need not understand this peculiarity of the RS232 cards.

To fully explain the modification we are about to make to your card, a little hardware detail will be presented. In the TI RS232, TI used certain Communication Register Bits to control the functionality of the card. See the chart for the TI definitions as compared to current and added definitions on the Myarc.

Terminal emulators and BBS software packages that connect the secondary CTS line to the modem DTR line do not work with Myarc cards. This is because the CTS line is inoperative as supplied by Myarc. We will fix that.

ENHANCEMENT WORKING DIRECTIONS

The first thing you will need to do is run down to your local Radio Shack and pick up part number 276-2520. It's a MC1488. FYI - its mate, the MC1489, is also available as part number 276-2521. They are both listed as costing $1.29 each in the 1994 catalog.

While you are there you may want to pickup some wire-wrap wire (278-501,502 or 503) and a wire-wrap wrapper/stripper (276-1570) — the stripper part is hidden in the handle. In fact, you can pull the wrapper portion out of the handle and stick it on cordless screw driver and make an auto-wrap tool.

Bit #	Function	Myarc
0	DSR ROM page Bit, 1=active	Same
1	PIO Port Mode Control, 1=Input	STROBE
2	PIO Output "STROBE" bit	Mode Control
3	Spare PIO output strobe	LED, 0=On
4	Flag Bit, 1=Set	DSR (*)
5	CTS - control, Primary RS232 port, 0=Active	CTS 1 (+)
6	CTS - control, Secondary RS232 port	CTS 2 (+)
7	LED Control, 1=ON	spare

(*) — This pin's usage is new to all three cards, and it can be used to hang up a modem by toggling the modem DTR line. Some terminal emulators and some BBS software packages connect the secondary CTS line to the modem DTR line to implement this function.

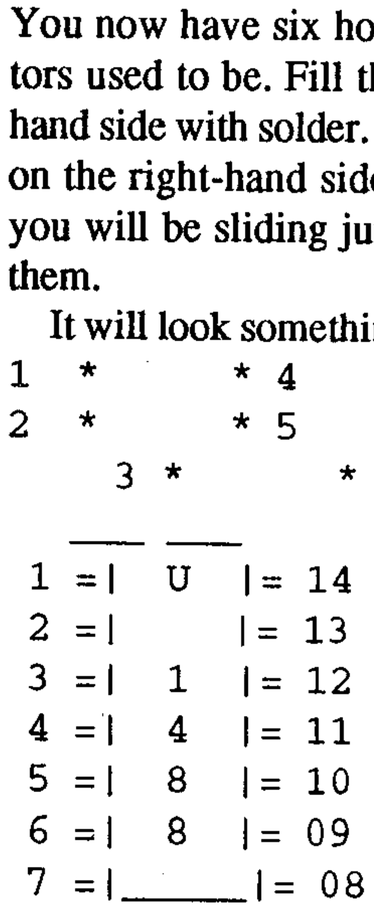
(+) - These are the flow control lines new to the Myarc card.

If you have a soldering iron and solder we are good to go.

Step 1 — Take the cover off.

Step 2 — Orient the board so that you are looking down at the chips and the edge connector is facing you. Now remove the three resistors in the top right-hand corner. You now have six holes where the resistors used to be. Fill the three on the left-hand side with solder. Make sure the holes on the right-hand side are clear, because you will be sliding jumper wires through them.

It will look something like this:

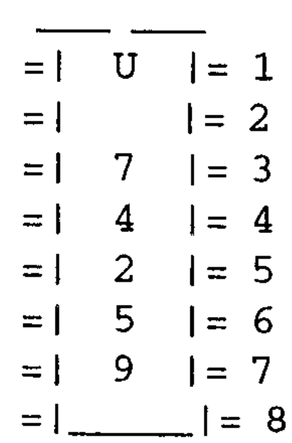


Step 3 — Take out your new 1488 and piggy-back it to the existing 1488 by soldering pins 1, 7 and 14 together. If this chip is socketed, remove it from the socket before piggy backing it. Next, bend up the unsoldered pins.

Step 4 — You will need three short and three long jumper wires. Solder a short jumper wire from solder hole 4 to pin 11 of the piggy-backed 1488. Solder another short jumper wire from solder hole 5 to pin 6 of the piggy-backed 1488. Solder the final short jumper wire from solder hole 6 to pin 8 of the piggy-backed 1488.

Now for the long jumpers.

Solder pins 4 and 5 of the piggy backed 1488 together. Run a jumper wire from that connection through solder hole 2. On the back side connect this to pin 11 of the 74LS259, which is the farthest chip away from the 1488 on the top row. Note: Since you are now working on the backside of the board, pin numbering is reversed, so the 259 we are talking about is numbered like this:



Solder pins 9 and 10 of the piggy-backed 1488 together. Run a jumper wire from that connection through solder hole 3. On the back side connect this to 09 of the 74259.

Solder pins 12 and 13 of the piggy-backed 1488 together. Run a jumper wire from that connection through solder hole 1, now the only one remaining. On the backside connect this to pin 10 of the 74259.

That's it for the hardware mods. Double-check everything. Tape down the three

(See Page 21)

MYARC RS232 FIX—

(Continued from Page 20

long jumper wires on the back and put the case on. You have now added three more control lines to the RS232 port that you can use to implement hardware flow control in your software.

Here are the assembler instructions to manipulate these three lines:

```
LI    R12,>1300    CRU base of RS232 card 1
SBZ                activate DSR
SBZ   5            activate primary CTS
SBZ   6            activate secondary CTS
SBO   4            de-activate DSR
SBO   5            de-activate primary CTS
SBO   6            de-activate secondary CTS
```

CRU bit 7 is left as a spare. There is also a spare line driver on the MC1488. We could attach pin 12 of the 74LS259 to pin 2 of the MC1488, then run a wire from pin 3 of the 1488 to an otherwise unused pin on the 25-pin RS232 connector. Another option would be to use CRU bit 7 for another control signal out the parallel port (without using the 1488). Right now we are happy to have "fixed" the Myarc RS232.

About the only thing we have not discussed in this article is how to change the Myarc card from RS232/1&2 to RS232/3&4. This will be explained at a later date.

WHERE TO BUY 99XX CHIPS

Many people have asked us where they can buy hard to come by 99XX chips, such as the 9902A's used in the RS232. There are two routes to go. One is not so sure, but very cheap, the other is guaranteed to have what you need, but it's a little on the pricey side.

The cheap route: Look in your local phone book for computer scrap or salvage people. They usually have tons of 99XX chips that they will be glad to part with at a fair price.

The more expensive route is to call Newark Electronics at 312-784-5100. This company should be able to tell you the number for your local Newark office, or just look in the phone book under electronics. They have 9901s, 9902s and a few others.

Transferring files from TI-Base to a PC

The following article was published in the newsletter of the West Penn 99ers.—Ed.

By DICK OHI

This article describes one method of transferring TI-Base data files from a TI99/4A to an IBM-compatible. It assumes that you are familiar with TI-Base and with the programs that you are using on the PC.

Most database programs will import data in an ASCII table format. This format is basically a text file that has one record per line. Each field contains the same number of characters in every record. An ASCII table file would look like the following example:

```
REC    LN      FN      PHONE
0001   JONES   SAM     311-555-1234
0002   SMITH   ANN     613-555-4321
0003   GREEN   JOHN    555-555-9876
```

This method of file transfer was done using a TI99/4A and a Gateway 2000 computer connected by an RS232 cable. This requires that the two computers are set up near each other. I used a ribbon cable with the appropriate DB25 connectors wired to the same pin numbers. Radio Shack part number 260-1408 should work.

Load TI-Base on the TI and change the set up with the following commands at the dot prompt:

```
SET PRINTER RS232.BA=4800.DA=8.PA=N.LF
SET SPACES 1
SET PAGE 200
```

Note that the page parameter should be set to a value greater than the number of records in the database file. The printer parameter assumes the cable is attached to RS232 port 1. If you are connected to RS232 port 2, then the command should be:

```
SET PRINTER RS232/2.BA=4800.DA=8.PA=N.LF
```

The SET SPACES 1 command will place a space between each field. This will cause each field length in the new database on the PC to be one character longer than in the TI database.

Load the database to be transferred into TI-Base.

Load a communications program on the PC that supports a "log" or capture feature. A log will capture to disk anything that is received by the program. In this example, I used Procomm Plus. Set up the PC to the same parameters (4800, 1, N) on the port the cable is connected to (COM1 or COM2). Open the log file.

On the TI, type the command PRINT ALL at the dot prompt in TI-Base. What happens here is that the TI "thinks" it is printing the database to the RS232 port and the PC thinks it is receiving data on the COM port. Procomm saves the data to disk as it is received. When TI-Base has finished printing, close the log on the PC.

Load your database program on the PC (I use Alpha 4), open a new database, select the import feature, type in the path to the log file and follow the procedures to set up the new database. If the log file is clean, that is each line of text is the same length and the data in each field begins at the same column and there are no extra control characters (page breaks, etc.) in the log file, then you should have all your data transferred properly.

However, if the new database has some of the data split among different fields, exit the program and load the log file into a text editor and look for extra control characters or fields that do not have the correct number of characters according to the original structure of the original TI-Base file.

Remember that we added one space character to each field in the printing process. Try to use an editor that will display control characters. Notepads in PC Tools, Word for Windows and WordPerfect have this feature.

In the above example, the data will ap-

(See Page 22)

PROGRAMMING IN FUNNELWEB—

(Continued from Page 21)

pear in the editor in columns or in table form if it had been printed on paper. If the records are longer than the number of characters displayed between the margins of the text editor, the lines of text will wrap to the next displayed line and not look like a table.

Another possible reason for the text not appearing to line up in table fashion is that the font used for displaying the text is a proportional font. Characters of proportional fonts are of varying widths so that, even if there are the same number of characters in each line, the lines appear to be of differing lengths on the screen. Try to use a font that will display each character at the same width, such as Courier. Using your text editor, make sure to eliminate any control characters, and check to be sure that each field in each record is of the correct length.

Other database programs for the PC have import features that will accept an ASCII table as input, though the terminology used to describe such things as fixed record length or system data format may differ.

Horizon SCSI card puts TI/Geneve users in the middle of an evolving technology

The following article was published in the newsletter of the Southern California Computer Group. It has been edited.—Ed.

By JAMES D. LANMAN

This article isn't quite a primer, but rather an introduction for those who are interested about or planning to purchase a Western Horizon Technologies (WHT) SCSI card. What follows should be useful and informative. A glossary is included at the end of this article to help explain some of the terms used. But, first, let's start with my definition of SCSI.

SCSI (Small Computer System Interface) is an evolving standard for connecting various types of peripherals to a host (CPU or internal bus). It is a bus specification and command set. The command set (CCS) optimizes use of the SCSI bus.

There are three versions of SCSI: SCSI-1, SCSI-2, and the upcoming SCSI-3 standard. Extensions to this standard are: SCSI Wide (utilizing either a 16-bit or 32-bit wide data path) and SCSI Fast (provides for 6-10 MHz data transfer rates between a peripheral and host). A SCSI host adapter may be single-ended or differential in electrical configuration. Data transfers may be asynchronous or synchronous. The WHT SCSI card is SCSI-2 compatible, which means it works only with SCSI-2 devices.

Having briefly defined SCSI, it should be known that this versatility in the SCSI standard allows it to be implemented on systems ranging from the TI 99/4A to supercomputers.

Many types of peripheral devices can be linked to a host by the SCSI interface. A device driver (software) will be needed to use a specific device. Some of the SCSI devices are: CD-ROM, floppy, hard, magneto-optical, tape back-up, and WORM drives — as well as — scanners and laser printers. Each type or make of device will require a separate driver. While all the necessary drivers to run SCSI floppy and hard drives will be included with the WHT card, others will have to be written. Other drivers will be made available later. Finally, having covered many of the SCSI device types, we come to the SCSI cable and the features of the WHT SCSI card.

50-PIN CABLE

The cable is a 50-conductor twisted pair/stranded-type ribbon cable. A 50-pin female dual-row ribbon connector will be needed for connecting to the WHT SCSI card. This is similar to the IDC type connector. For each peripheral, you will need a 50-conductor female Centronics-type ribbon cable connector. (See Table 1 for SCSI pin-out information.) A cable six feet or less in length is desirable.

In any case, a cable shouldn't exceed six meters. Generally, cables are more susceptible to interference as their length increases. In addition, SCSI peripherals use a terminating resistor pack just like older TI floppy drives. These terminating resistors will need to be removed from all of the devices except for the last device connected to the cable

(See Page 23)

Table 1
SINGLE-ENDED SCSI INTERFACE

Pin #	Descriptor	Pin #	Descriptor
1	GND	26	TRMPWR
2	D0-	27	GND
3	GND	28	GND
4	D1-	29	GND
5	GND	30	GND
6	D2-	31	GND
7	GND	32	ATN-
8	D3-	33	GND
9	GND	34	GND
10	D4-	35	GND
11	GND	36	BSY-
12	D5-	37	GND
13	GND	38	ACK-
14	D6-	39	GND
15	GND	40	RST-
16	D7-	41	GND
17	GND	42	MSG-
18	DPAR-	43	GND
19	GND	44	SEL-
20	GND	45	GND
21	GND	46	C/D-
22	GND	47	GND
23	GND	48	REQ-
24	GND	49	GND
25	OPEN	50	I/O-

SCSI CARD—

(Continued from Page 22)

(which is farthest on the cable from the SCSI controller). Save the terminating resistor packs and store them in a safe place. Should you need to change your configuration, you will have prevented any problems arising from a lost or discarded terminating resistor pack.

The WHT SCSI card can use up to seven devices (actually eight, if bus arbitration is turned off). This includes SCSI floppy drives, 5.25-inch or 3.5-inch, with capacities up to four megabytes, SCSI hard drives with a capacity up to 2.7 gigabytes, or any combination of storage devices up to a total of four gigabytes.

OTHER DISK CONTROLLERS

WHT SCSI users can also use all the standard floppy drives, from 360K to 2.88 megabytes, with the addition of the FC-1 card. This card requires a SCSI card and is available for $100 from Horizon and Bud Mills. The SCSI card coexists with other hard or floppy disk controllers.

The SCSI card also supports some CD-ROM drives for access to pictures and sound. Drivers for hard drives are available, as well as a SCSI version of Unimanager by Mike Maksimik.

Any SCSI device running at rates from 1 MHz to 5 MHz may be used with a proper driver. These devices are connected in a daisy chain fashion with a ribbon cable. The real power of the WHT SCSI card will become apparent when combined with the 4/A Memex card.

This article has barely covered some of the basics of SCSI. It would take a series of articles to really explain this new standard for the TI. I would like to thank Don O'Neil of WHT and the ANCOT Corporation for their invaluable assistance in writing this article.

The SCSI card is available from Western Horizon Technologies and Bud Mills Services for $170. Write or call them at: Western Horizon Technologies, Don O'Neil, 10225 Jean Ellen Drive, Gilroy, CA 95020; (408) 848-5947; Bud Mills Services, 166 Dartmouth Drive, Toledo, OH 43614-2911; (419) 385-5946.

A SCSI glossary

Arbitration — The process of selecting one respondent from a collection of several candidates that request use of the SCSI bus concurrently.

Asynchronous Transmission — A transmission in which each byte of information is synchronized individually, through the use of interlocking the REQ and ACK signals.

BIT — BInary digiT, which can have a value of 0 or 1. It is the smallest unit of data that a computer can process. Bits are arranged into groups of eight called bytes. A byte is the equivalent of one character.

Bus — A signal path or line shared by many devices. Information is often sent to all devices throughout the bus; only the device to which it is addressed will accept it.

CCS — Common Command Set. CCS is a collection of 18 commands, which is a subset of SCSI-1. SCSI-1 specification allowed too many vendor specific features. CCS was designed to improve compatibility between SCSI devices from different vendors. CCS is included in SCSI-2.

CPU — Central Processing Unit is the nerve center or brain of the computer. It interprets programs and tells the computer how to execute them.

Differential Interface — An electrical signal configuration using a pair of lines for transfer. On a SCSI bus `TRUE' (logical `1') is defined as -SIGNAL (about 1 Volt) higher than +SIGNAL line, and opposite polarity for `FALSE'. The advantage of differential configuration (as compared to single-ended) is in relatively higher tolerance for common-mode noise, and little crosstalk when used with twisted pair cables. It allows for connections up to 20 meters (about 22 yards). Its disadvantage is higher (component) cost.

Host — A processor, usually consisting of a CPU and memory. Typically, a host communicates with other devices, such as peripherals and other hosts.

Host Adapter — Circuitry that translates between a processor's own internal bus and a different bus, such as SCSI. On the SCSI bus, a host acts as an initiator and a peripheral acts as a target.

MHz — MegaHertZ(MHz) is a million cycles per second. Used as a measurement of data transfer rate.

Peripheral — A device that can be attached to a host computer, using a SCSI bus for example. Typical types of peripherals are floppy disk drives, hard disk drives, etc.

SCSI — Small Computer System Interface. Pronounced "Skuzzy". An industry standard for connecting peripheral devices and their controllers to a microprocessor. SCSI defines both hardware and software standards for communication between a host computer and a peripheral.

Single-ended Interface — An electrical signal configuration using a single line for each signal, referenced to a common ground path. The advantage of a single-ended configuration (as compared to differential) is in using half the number of pins, chips, and PCBoard area. Its disadvantage is higher susceptibility to common mode noise, and a limited cable distance.

Synchronous Transmission — A transmission in which the sending and receiving devices operate continuously at the same frequency, and are held in a desired phase relationship by a correction device. For buses, synchronous transmission is a timing protocol that uses a master clock and has a clock period and allowable offset.

WORM — Write Once Read Many. An optical storage device on which data is permanently recorded. Data can be erased, but not altered or additional data added.

MICRO-REVIEWS

The Complete Sherlock Holmes Stories, The Castle, Constitution Reader

By CHARLES GOOD

THE COMPLETE SHERLOCK HOLMES STORIES by Arthur Conan Doyle

Sherlock Holmes is probably the most famous detective in English literature. By just glancing at someone he could usually tell the person's occupation, where he had been recently, and other details of the person's personal life. He was a "consulting detective." Other detectives (public police and private) as well as the general public went to him for advice when a case seemed particularly confusing. It was the Sherlock Holmes stories that made author Arthur Conan Doyle very wealthy in spite of Doyle's efforts to direct his writing talents elsewhere. On two separate occasions Doyle tried to tell his public that there would be no more Holmes stories forthcoming. Each time public pressure and the offer of vast sums from publishers forced Doyle to write additional Holmes adventures.

The Sherlock Holmes stories were originally published between 1887 and 1926. The copyrights have expired and everything is now in the public domain. I have downloaded all these short stories and book length manuscripts from an information system, converted them into D/V80 files on double-sided single-density disks, and am making them available to the TI community.

These text files are nicely formatted in 80 columns, displayed double-spaced, and contain no control codes or strange printer format codes. Each file is about 200-300 sectors in length and contains either a complete short story or 2-3 chapters of a book length manuscript. Book length stories are split up into several of these files. The text files are too large to load into most versions of TI-Writer, but that's okay because you don't want to manipulate this sort of text.

You want to view the text on screen or print the text with your printer and this can be done with either Funnelweb, DM1000 or DSKU. With each of these software packages you bring up a disk directory and put the cursor next to the file name you want to view or print. At this point using Funnelweb's Disk Review or using Birdwell's DSKU you press "V" to view and then a second key to print. Using DM1000 you press "T" to type the text onto the screen, or "P" to print the whole file. 40 column users will probably prefer using DM1000 for viewing and printing. Funnelweb is the choice for those with 80-column systems. Double-spacing makes it easy to view these 80-column files on a 40-column screen. Each text line wraps around to the next 40-column line during such a viewing, resulting in double lines of text on screen with a blank line separating the next double text line.

The sound, speech, graphics, and interest-holding ability of The Castle are first rate. I give it a thumbs up.

The following Sherlock Holmes stories make up this collection:

- "A Study in Scarlet"-the book that introduced Holmes and Watson to the world in 1887;
- "The Sign of Four"-a book;
- "The Valley of Fear"-a book that has within it another book;
- "Hound of the Baskervilles"-in my opinion the greatest detective story ever written;
- and 56 short stories originally published separately and later gathered together by Doyle into books titled "The Adventures of Sherlock Holmes," "The Memoirs of Sherlock Holmes," "The Return of Sherlock Holmes," "His Last Bow," and "The Casebook of Sherlock Holmes." Several of these short story compilations contain very interesting prefaces by Doyle which were not included when the stories were first published separately.

I am not asking for any copy fee. You can have the complete collection, archived, by sending me 14 DSSD disks or 7 DSDD (18 sectors/track) disks and a paid return mailer. Please make sure that disks are initialized and that the return postage is correct. I will include the Archiver program (E/A5 version) so you can unpack the archives onto DSSD disks and I will also include my SPEAK D/V80 program. You can run SPEAK D/V80 from Extended BASIC and have your 99/4A read the Sherlock Holmes stories to you using the speech synthesizer. SPEAK D/V80 will "speak" any D/V80 file.

THE CASTLE by Vern Jensen

This is a really good game, somewhat similar to TI-Runner in that it has multiple levels (screens) that you reach as you progress. My at-home testing panel (ages 8, 13, and 15) each spent at least two hours playing The Castle. The 13-year-old spent about six hours spread over several days before he finally tired of the game. These are really good times! The attention span of these kids, who have access to a 386 PC and a Sega Genesis, usually isn't very long when it comes to TI games. I spent several hours myself getting lost in The Castle in preparation for this review.

The game is written in Extended BASIC with CALL LOADed assembly routines. The graphics are great and the game play is fast enough to suggest that it was totally written in assembly, which it isn't.

(See Page 25)

MICRO-REVIEWS—

(Continued from Page 24)

You start outside in a hail storm and duck inside a handy castle to get away from the precipitation. There are treasures in the castle which you are supposed to find as you travel through all the rooms in an attempt to get to the last room where the exit is located. Each room (screen) has lots of doors and various brick barriers. One door leads to the next room, one door goes to the previous room, and most of the doors take you to another location within the same room. It is very frustrating trying to find the "next room" door, and part of the frustration is physically getting to all the doors you can see in order to try them out. Those brick barriers get in your way. However some of the barriers are fake. You can walk or fall right through what appear to be solid walls or floors.

There is no instruction manual, but there is a very well done on-screen, interactive tutorial on how to play the game. You actually play an abbreviated version of the game. You see the graphics and you get to experiment with the various types of on-screen movement. Everything is nicely explained. My 8-year-old had no trouble running the game, going through the tutorial, and then playing the game, all with no help from me.

The Castle comes with a bunch of predefined rooms. When you have them all memorized, you can make a custom game by creating your own rooms. You put doors, true and false barriers, and so forth where you want them. A documentation text file explains how, since the tutorial does not cover custom room creation.

The sound, speech, graphics, and interest-holding ability of The Castle are first rate. I give it a thumbs up. You can obtain The Castle for $6 plus $2 postage directly from the author Vern Jensen, 817 Kingsway Dr. East, Gretna LA 70056.

CONSTITUTION READER
by Sam Carey

When was the last time you sat down and actually read the United States Constitution, our country's basic law? I did it a few years ago when there were special celebrations commemorating the 200th anniversary of this document. Before that, the last time I did so was when I was in high school. This is a really important document with which we should all be familiar. It appears in the news almost daily. Gun control (why can't laws prohibit the ownership of guns), capital punishment (is it cruel and unusual punishment), abortion (do the unborn have "constitutional rights"), separation of church and state (is it legal to give tax breaks or "vouchers" to defray the cost of educating children in church sponsored schools) are among many current issues that relate directly to our Constitution.

Now we can read the Constitution on our TIs. Constitution Reader includes the entire text with all the amendments in 40-column format, and software for on-screen reading and searching of this text. The reader is "Load and Run" assembly software that loads from the E/A module, the Mini-Memory Module, or from Funnelweb. You see 40-column text at the top of the screen and a command line at the bottom. From the command line you can move back and forth through the text one line or one screen at a time. You can also search for text strings, and you can print the part of the text displayed on the screen. For example, you can search for the string "AMENDMENT XIV" to go directly to the 14th amendment. A second copy of the Constitution nicely formatted in D/V80 80-column format is also included so you can also use a word processor to print the entire document to your printer.

Constitution Reader v2.1 comes on a DSSD disk and costs $10, plus $1 for postage and handling. A version that works from one SSSD is also available by special request, but the 80-column text is archived and has to be unpacked to another SSSD disk. Send your money to Sam Carey, 5820 SE Westfork St., Portland OR 97206-0742.

If you'd like your software reviewed in MICROreviews, send it to Charles Good, P.O. Box 647, Vendocia, OH 45894. His e-mail address is cgood@magnus.acs.ohio-state.edu. Or call him by voice at 419-667-3131.

New version of DM1000 shows improvement

By MARY PHILLIPS

REVIEW

Let me tell you, this version of DM1000 is the most beautiful disk and file manager I've ever seen. And the documentation is very clear and easy to follow.

What makes v6.1 different from 5.0? Major modifications, by Jack Mathis, include:

• Consolidation of the disk and file utility menus into one main menu.

• T)ype or (P)rint in the CMD column of the File Utilities catalog displays a DV/DF80 file to the screen, or print it out.

• Disk initialization (formatting) and copying are speeded up.

• Defaults for disk formatting, printer configuration (device and codes), and foreground and background colors may be saved into the program.

• Choice of drives for saving configure defaults.

• Works with Myarc HFDC on the TI and on Geneve with Ben Hatheway's ROMPAGE loaded.

Print out the documentation with TI-Writer or the Print File Option of BOOT!, MENU, or DM1000 itself. A Quick Reference Guide is included in the documentation.

The following key presses are active in File Utilities:

FCTN 1: Delete a character

FCTN 2: Insert a character

FCTN 3: Configure List Device (printer or

(See Page 26)

DM1000—

(Continued from Page 25)

DSKn.filename)
FCTN 4: Halt disk drive I/O operation
FCTN 5: Return to DM1000 main menu.
FCTN 6: Request "EXECUTE COMMANDS Y/N" prompt
FCTN 7: Print Catalog to List Device
FCTN 8: Re-enter Drive #
FCTN 9: Return to DM1000 main menu
FCTN =: Exit Disk Manager 1000
FCTN E: Move cursor up one field
FCTN X: Move cursor down one field
FCTN S: Move cursor left one character or back one field
FCTN D: Move cursor right one character or ahead one field
CTRL E: Move cursor back one page
CTRL X: Move cursor ahead one page
CTRL C: Copy all files
CTRL D: Delete all files
CTRL N: Perform No Action on any files
CTRL P: Protect All Files
CTRL U: Unprotect All Files

Individual files may be marked for Copy, Delete, Move, Protect, or Unprotect and then press FCTN 6 to proceed. T or P must be done by themselves.

When DM1000 copies a disk you have a choice of bitmap (copying only the sectors that are used) or sector copying all the disk sectors.

If you (D)elete a file then wish you hadn't, Undelete will ask you for the disk drive number and the filename and it will reconstruct the link between the directory and the file so you have your file back.

Unprotect is only for Extended BASIC programs. If it is used on other files, they may be unusable.

Error messages are in friendly English, no code numbers. This program is user friendly.

To put DM1000 on your BOOT! or Horizon Ramdisk you need only MGR1 and MGR2. For the DOM 1/89 BOOT!, just copy these two files over the ones you have and delete MGR3. In MENU, delete files MG and MH and copy MGR1 and MGR2, then rename them MG and MH.

1994 TI FAIRS

MAY

Lima Multi User Group Conference, May 13-14, Ohio State University Lima Campus, Lima, Ohio. Contact Lima Ohio Users Group, P.O. Box 647, Venedocia, OH 45894.

NOVEMBER

The TI International World's Faire, tentatively scheduled for Nov. 12 in Gurnee, Illinois. Don Walden says vendor tables will be available at a low price and vendors can charge their table rental on Visa or MasterCard.

1995 TI FAIRS

FEBRUARY

Fest West'95, Feb. 18, Fabulous Inn, San Diego, California. Contact Southern California Computer Group, P.O. Box 152535, San Diego,CA 92108, or call the SCCG BBS, (619).263-9125, User No. 23, password FEST.

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.

NEWSBYTES

Fest West '95 set for San Diego

Fest West '95 will be held Feb. 18 at the Fabulous Inn in San Diego, California, according to John Chatfield, president of the Southern California Computer Group, the host group for the fair.

Hours for the fair are scheduled from 9 a.m. to 6 p.m. The Fabulous Inn offers special rates to fairgoers Feb. 17-20, according to the Fest West committee. The rates are $44 per day single occupancy (one bed) or $49 per day double occupancy, (two beds). Add $5 per day for deluxe room upgrade.

For hotel information and registration, contact the Fabulous Inn, 2485 Hotel Circle Place, San Diego, CA 92108, California toll-free 1-800-647-1903, U.S. toll-free 1-800-824-0950.

The hotel features free covered parking, a large heated pool and Jacuzzi and adjacent golf and tennis. It is 5 to 10 minutes from the airport, San Diego's Jack Murphy Stadium, Sea World, the San Diego Zoo, Old Town, Seaport Village and the beach.

Vendors will receive two tables and two admission tickets for $20, with a $10 fee for each additional table. However, vendors sending in fees before Dec. 31 will receive two tables and two admission tickets for $10, with $10 for each additional table.

For further information, contact the SCCG BBS, (619) 263-9135, user number 25, password FEST. Or write the SCCG, P.O. Box 152535, San Diego, CA 92195.

Constitution Reader V.2.1 released

Machicolation Systems has released V.2.1 of its Constitution Reader. The new version offers an improved search routine, print-out option and the text of the Constitution in Display/Variable 80 form, according to Sam Carey of the company.

The program is available for $10 plus $1 shipping and handling, on either a single-sided or double -sided disk. Text on the single-sided disk is archived using Barry Boone's Archiver.

For information or to order, write Sam Carey, 5820 SE Westfork St., Portland, OR 97206-0742. Make all checks payable to Sam Carey. Carey can also be contacted at his Internet address, Sam.Carey@f34.n105.z1.fidonet.org.

TM Direct produces last 4A catalog

TM Direct Marketing scheduled release of its last TI99/4A-related catalog in mid-

(See Page 27)

NEWSBYTES

(Continued from Page 26)

May, according to Terry Miller of the company.

He says the catalog reflects close-out prices, but the company's 30-day guarantee on products will be honored, and Texas Instruments still honors its warranties on TI products.

Persons not on the company's mailing list who wish to receive a catalog or find out the price and availability of specific items can call 1-800-336-9966 or write 1757 E. Bayshore Rd., Unit 12, Redwood City, CA 94063. Miller says he will probably continue selling TI products through the summer.

Chicago group has new mailing address

The Chicago Users Group has a new mailing address, according to Hal Shanafield of the group. It is P.O. Box 7009, Evanston, IL 60204-7009.\

Lewis offers rights to math coprocessor

Tony Lewis has announced that he is offering the engineering data, prototype, software and other rights to his Motorola 68881 math coprocessor card for the TI99/4A to any individual who would like to develop the card for sale. Lewis notes that the product is not commercially complete and work on the software must be completed.

Contact Lewis at CompuServe ID 73357,1730.

Voltage suppressor goes on market

New Stabiline Power Quality Interfaces are being manufactured by Superior Electric.

These plug-in voltage suppressors have a UL 1449 transient level suppression performance rating of 330 volts and allow bidirectional protection from source or load originated power disturbances, according to the company.

The PQI unit can be plugged into any convenient wall outlet and the power cord of the equipment to be protected then plugged into the unit.

Model PQI-1115W sells for $35. Model PQI-1115WD sells for $39. Both are packaged in fire-rated ABS plastic housings with NEMA 5-15P plug and plastic outlet locator pin assemblies. Model PQI-1115WD additionally has two RJ11 modular jacks with transient suppression and is supplied with a six-foot telephone line cordset for use with fax machines, computers with modems and similar applications.

For full information, write, fax or phone for PQI-1115W(D) Data Sheet, Superior Electric, 383 Middle St., Bristol, CT 06010, (203) 585-4500 (voice), (203) 582-3784.

Right-to-die BBS goes online

Choice in Dying, a national non-profit organization, has introduced an electronic bulleting board service devoted solely to right-to-die issues and end-of-life decision-making. Files include files about living wills and durable powers of attorney for health care, which are legal documents allowing an individual to participate in medical treatment decisions at the end of his or her life. It contains updates on legal issues and other information as well.

The 24-hour BBS can be accessed at (212) 727-8219, N-8-1, at baud rates up to 14.4 kbps. There are no online registration fees or hourly charges, according to Karen Orloff Kaplan, executive director of Choice in Dying.

Send your information about products and services for the TI/Geneve community to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680.

USER NOTES

RAMID modification bypasses password

Bruce Harrison of Hyattsville, Maryland, offers the following modification of William H. Berendts' RAMID program, which appeared in our February 1994 issue. Harrison writes that Berendts had come up with an interesting little concept for "securing" one's TI system by having the RAMdisk run an Extended BASIC program which would require entering a password before allowing the RAMdisk's Menu to come up.

His program would, of course, work, but it would keep coming back to the password program any time the system cycled back to title screen. Here's a short XB program that overcomes that problem. Once this has run, and a correct password has been entered, it clears a byte of memory in the Load Interrupt area. CALL LOAD(-2,0), which contains>FF on cold startup. Thus when next we cycle through the title screen, the program finds 0 at location -2, and so skips right ahead to Menu. So long as XB is available (as is is on systems with P-GRAM as well as RAMdisk), this will work correctly. It won't make you reenter the password unless the P-Box has been shut off or some program has changed the memory byte at >FFFE. (Almost all programs leave that part of memory alone. It's of no use to anyone without a Load Interrupt switch.)

To use this, simply type it in under Extended BASIC, change the password in line 10 (e.g. PW$="ANGRY CAT"), then save it to the same RAMdisk drive as your Menu program is on. Give it a short file name (e.g. DSK5.PW). Now go into Horizon's Configure utility and edit your ROS so that 4 MENU is the second item on the calls list, and make the first item 2 PW. Shut your system down and fire it up again. Instead of the menu, you'll get the prompt for a password. Type in yours, and the normal menu should appear. So long as you don't shut off the P-Box, you can

(See Page 28)

USER NOTES

(Continued from Page 27)

do anything you want without seeing the password request again. That's it. Enjoy! Should you forget your password, there's a back door method of getting into your computer. Call me at (301) 277-3467 to find out.

The two drawbacks to this are that you need XB present to make it work, and anyone can see the password while you're typing it in. I am making an assembly version that will overcome both problems. That assembly disk will be made available through Lima and other user groups, probably before this appears in User Notes.

PASSWORD

```
10 ON BREAK NEXT :: PW$="PEN
TAGON" :: CALL INIT !020
20 CALL PEEK(-2,A):: IF A TH
EN 30 ELSE DELETE "MENU" !16
4
30 DISPLAY AT(12,1)ERASE ALL
:"TYPE YOUR PASSWORD" :: ACC
EPT AT(14,1)BEEP:ENT$ :: IF
ENT$<>PW$ THEN 30 !133
40 CALL LOAD(-2,0):: DELETE
"MENU" !122
```

'Hydro' is puzzling

The following was written by Lucie Dorais of the Ottawa TI User Group. It is excerpted from a longer article.

I looked into my collection and found a little game that I had typed on day and decided it was too short. Five years later, it is finally getting its 15 minutes worth of celebrity. The code is not mine, only the graphics are, I think. I don't have the original book anymore. When you start, 10 light bulbs are on (they are really yellow squares). You have to switch the lights off and on until all the lights are off.

Because of the very complicated routine in line 260, funny things happen when you switch a light — other lights are switched off as well, but still other ones are switched on. I was never able to finish the game, so I added my favorite "give up" key. Line 260 uses the random statement RND a few times, all by itself, which gives values >0 and >1, with 10 decimals: try FOR X=1 to 100 :: PRINT RND :: NEXT X to check. If that was not enough, the author is also using the TAN and SIN statements, which are mathematic formulae. The Extended BASIC manual does not explain what they really are, and my trigonometry lessons are too far away. I have completely forgotten what they do, but the result is completely maddening.

HYDRO

```
100 REM ** HYDRO ** J.Deconc
hat / Adapted by L.Dorais /
Ottawa U.G. / March 1989 (Fe
bruary 1994) !086
110 CALL CLEAR :: CALL SCREE
N(4):: RANDOMIZE :: DIM A(10
)!228
120 A$=RPT$("F",16):: CALL C
HAR(104,A$,112,A$):: CALL CO
LOR(10,14,1,11,11,1):: L$="
  "&RPT$("h",23)!106
130 GOTO 140 :: I,K,M,N,R,S,
T :: CALL HCHAR :: CALL SOUN
D :: CALL KEY :: !@P- !019
140 DISPLAY AT(6,1):L$:L$:L$
:L$:L$:L$:"   hh0h1h2h3h4h5h6h7
h8h9hh":L$ !212
150 DISPLAY AT(17,1):" RANDO
M THINGS HAPPEN WHEN  YOU S
WITCH THE LIGHTS..." !080
160 DISPLAY AT(20,1):" PRESS
 A SWITCH NUMBER TO":" PUT L
IGHT OFF OR ON, UNTIL":" ALL
 LIGHTS ARE OFF..." !165
170 DISPLAY AT(24,4):"(PRESS
 ""G"" TO GIVE UP)" :: M=0 !
083
180 T=0 :: FOR I=1 TO 10 ::
T=T+A(I):: CALL HCHAR(8,2*I+
S+5,104)!246
190 IF A(I)THEN 200 ELSE CAL
L HCHAR(8,2*I+6,112)!007
200 NEXT I :: IF T=10 THEN 2
90 !190
210 CALL SOUND(100,-1,0)!215
220 CALL KEY(5,K,S):: IF S=0
 THEN 220 :: IF K=71 THEN A$
="  YOU GAVE UP..." :: GOTO
300 !221
230 IF K<48 OR K>57 THEN 220
 ELSE CALL SOUND(40,800,0)::
 DISPLAY AT(13,4):"" !192
240 N=K-47 :: M=M+1 :: CALL
HCHAR(13,2*(N+3),94)!163
250 IF A(N)=1 THEN A(N)=0 ::
 GOTO 260 ELSE A(N)=1 !166
260 R=TAN(RND+N/RND-N)-SIN(R
ND/N)+336*SIN(8*N):: N=INT(1
0*(R-INT(R)))!207
270 IF N=0 THEN N=INT(RND*10
+1)!001
280 IF A(N)=1 THEN A(N)=0 ::
 GOTO 260 ELSE A(N)=1 :: GOT
O 180 !043
290 A$="FOUND IN";M;"TRIALS"
 !118
300 DISPLAY AT(19,1):"":"":"
":TAB(5);A$:"":TAB(7);"PLAY
AGAIN?  Y" !058
310 ACCEPT AT(24,20)SIZE(-1)
VALIDATE("YN")BEEP:A$ :: IF
A$="N" THEN END !129
320 DISPLAY AT(13,4):"" :: F
OR I=1 TO 10 :: A(I)=0 :: NE
XT I :: GOTO 160 !070
```

Better error handling in Extended BASIC

The following was written by Mark Schafer of the Bluegrass 99 Computer Society of Lexington, Kentucky. It appeared in Bytemonger, the group's newsletter.

This tiny tip was inspired by a tip I saw in another newsletter, but I intend to take it a step further.

It begins with the supposition that you have a line like this in your Extended BASIC program:

```
100 ACCEPT AT(10,16)VALIDATE
(DIGIT):H
```

This line prevents the user from typing non-numeric characters, so you might think you have prevented the String-Number Mismatch error from occurring. Well, you have, but you have not prevented a Warning, which would not only shake up your user but also will mess up your screen. This will happen if the user simply hits return without entering a number.

There are three ways to combat this. The first one is ineffective but, if you are the only person who will ever use your program, you don't have to sweat it. Let's say you want zero to be the default. You could do this:

```
100 DISPLAY AT(10,16):"0"::
```

(See Page 29)

USER NOTES

(Continued from Page 28)

```
ACCEPT AT(10,16)VALIDATE(DIG
IT)SIZE(-4):H
```

This would put a zero in the input field. So, if the user hits return, H will be zero instead of a warning being issued. However, if the user hits FCTN-Erase before hitting Enter, the warning will still be forthcoming. That isn't likely if the user doesn't have malicious intent. Still, this method also forces you to set a limit on the number of digits to be typed, in this case 4. And it is some work. It could be less work if the default value could be added to the end of a previous DISPLAY AT statement where the question was put on the screen.

This leads us to the second way which was given in the article I read. It suggests simply adding the following line to your program:

```
90 WARNING NEXT
```

This will prevent the warning by telling the computer to ignore it. If the user gives a null response, the computer simply reprompts him in the same place. No harm done.

But suppose you want the best of both worlds. That is, you want a default, like the first method, and you want it to be idiot-proof and flexible, like the second method. This is where my suggestion comes in. Here is how I do it:

```
100 ACCEPT AT(10,16)VALIDATE
(DIGIT):H$:: IF H$="" THEN H
=0 ELSE H=VAL(H$)
```

Now if the user hits only the Enter key, zero will be returned. This method can also be combined with the second method in those programs for which you want the user to be re-prompted sometimes and a default to be used at other times.

If you like to be cryptic in your programs, or you need to save memory, the line line above can also be written as follows:

```
100 ACCEPT AT(10,16)VALIDATE
(DIGIT):H$:: H=VAL("0"&H$)
```

Of course, this works only if you want the default to be zero.

You can also fix the problem of the first method by simply adding the line of the second method. The first and third methods can be combined if you want a null response to be interpreted as the default *and* you want the user to see what the default is. Lots of ways to go here. Pick one.

Cataloging disks to text files

The following two programs and text were written by Bob August. They appeared in Bug News, the newsletter of the Brea 99ers User Group in California.

The first program will catalog your disks, listing the files to a D/V80 file with the file name, size and disk name, along with a comment line of up to 28 characters.

(See Page 30)

USER NOTES

(Continued from Page 29)

The second program will print the listing with margin and skip-over so you can put it in a loose-leaf binder.

The data file is opened in Append mode so you can do some disks one day and then add to it on another day, or start a new file. Both programs require Extended BASIC.

DISK FILE CATALOG

```
100 ! DISK FILE CATALOG !041
110 ! IN TI EXTENDED BASIC !
234
120 ! BY R.W. AUGUST !092
130 DIM FT$(5)!179
140 FT$(1)="DIS/FIX" :: FT$(
2)="DIS/VAR" :: FT$(3)="INT/
FIX" :: FT$(4)="INT/VAR" ::
FT$(5)="PROGRAM" !145
150 DISPLAY AT(2,3)ERASE ALL
:"<< DISK FILE CATALOG >>" !
209
160 DISPLAY AT(5,1):"ENTER T
HE NUMBER OF THE":"DRIVE TO
CATALOG:": :"ENTER THE DRIVE
 NUMBER OF":"THE DATA DISK:"
 !165
170 ACCEPT AT(6,19):CD$ :: A
CCEPT AT(9,19):DD$ :: DISPLA
Y AT(11,1):"CREATE NEW DATA
FILE Y/N Y" :: ACCEPT AT(11,
26)SIZE(-1)VALIDATE("YyNn"):
CNF$ !199
180 DISPLAY AT(13,1):"PLACE
DISK TO BE CATALOGED": :"IN
DRIVE NUMBER: ";CD$ !067
190 DISPLAY AT(17,1):"PLACE
DISK TO SAVE DATA ON": :"IN
DRIVE NUMBER: ";DD$: : :"PRE
SS ENTER WHEN READY." !128
200 CALL KEY(0,K,S):: IF K<>
13 THEN 200 :: CALL CLEAR ::
 DISK=0 !177
210 DISPLAY AT(10,1):"ENTER
FILE NAME TO SAVE YOUR": :"D
ATA TO." :: ACCEPT AT(14,1)S
IZE(10):DF$ :: CALL CLEAR ::
 FN$="DSK"&DD$&"."&DF$ !231
220 ON ERROR 420 :: OPEN #2:
FN$,APPEND,VARIABLE 80 :: IF
 CNF$="N" OR CNF$="n" THEN 2
50 ELSE 230 !217
230 PRINT #2:TAB(6);"FILE NA
```

Disk File Catalog

FILE NAME	SIZE	TYPE	DISK NAME	FILE COMMENTS
*README	6	DIS/VAR	MAYSTUFF	
*UTIL/BILL	37	DIS/VAR	MAYSTUFF	XB PROGRAM
-README	8	DIS/VAR	MAYSTUFF	
AL	2	DIS/VAR	MAYSTUFF	
BUYFOOD	29	PROGRAM	MAYSTUFF	XBASIC PROGRAM
HEADER_DOC	10	DIS/VAR	MAYSTUFF	
LOG-SCSI	45	DIS/VAR	MAYSTUFF	
MYPROG_C	17	DIS/VAR	MAYSTUFF	
NSTDIO_H	10	DIS/VAR	MAYSTUFF	
QL5	6	DIS/VAR	MAYSTUFF	
RAM	2	DIS/VAR	MAYSTUFF	
RAMARTCLE	16	DIS/VAR	MAYSTUFF	
SIDEBAR35	96	DIS/VAR	MAYSTUFF	E/A SOURCE CODE

```
ME SIZE TYPE DISK NAME
FILE COMMENTS" !108
240 PRINT #2:TAB(6);"-------
--- ---- ------- ----------
----------------------------
" !074
250 DISK=DISK+1 :: CD=0 :: O
N ERROR 450 :: OPEN #1:"DSK"
&CD$&".",INPUT ,RELATIVE,INT
ERNAL !251
260 INPUT #1:X$,W,X,Y :: U=(
X-Y)+2 !012
270 DISPLAY AT(6,1):"Disknam
e is ";X$;TAB(24);"#";STR$(D
ISK): :"Available =";Y;TAB(1
8);"Used =";U :: CD=1 !110
280 DISPLAY AT(11,1):"Catalo
g this disk Yes/No Y":"or En
ter S to stop." !154
290 ACCEPT AT(11,26)VALIDATE
("NnSsYy")SIZE(-1):YN$ :: IF
 YN$="N" OR YN$="n" THEN DIS
K=DISK-1 :: GOTO 370 !020
300 IF YN$="S" OR YN$="s" TH
EN CLOSE #1 :: GOTO 410 !089
310 FOR F=1 TO 127 :: INPUT
#1:A$,A,J,K :: IF LEN(A$)=0
THEN 370 ELSE DISPLAY AT(11,
1):"FILE NAME SIZE TYPE":"-
--------- ---- -------" !175
320 DISPLAY AT(13,1):A$;TAB(
12);J;TAB(17);FT$(ABS(A));"E
nter Comment Yes/No Y" :: AC
CEPT AT(16,22)VALIDATE("NnYy
")SIZE(-1)BEEP:YN$ !246
330 DISPLAY AT(16,1):"" :: I
F YN$="N" OR YN$="n" THEN FI
LECOM$=" " :: GOTO 350 !032
340 DISPLAY AT(16,1):"Enter
File Comment" :: ACCEPT AT(1
8,1):FILECOM$ !055
350 PRINT #2:TAB(6);A$;TAB(1
7);:: PRINT #2,USING "####":
J;:: PRINT #2:TAB(22);FT$(AB
S(A));TAB(30);X$;TAB(41);FIL
ECOM$ !228
360 DISPLAY AT(18,1):" " ::
NEXT F !218
370 CLOSE #1 :: DISPLAY AT(6
,1):"":"PLACE THE NEXT DISK
TO BE":"":"CATALOGED IN DRIV
E: #";CD$:"":"LEAVE THE DATA
 DISK IN": :"DRIVE #";DD$ !0
26
380 DISPLAY AT(14,1):"":"":"
PRESS ENTER WHEN READY OR":
:"PRESS Q TO QUIT" !159
390 CALL KEY(0,K,S):: IF K=8
1 OR K=113 THEN 410 !052
400 IF K<>13 THEN 390 :: CAL
L CLEAR :: GOTO 250 !154
410 CLOSE #2 :: CALL CLEAR :
: STOP !005
420 DISPLAY AT(12,3)ERASE AL
L:"<< ERROR IN DATA DISK >>"
 !068
430 FOR D=1 TO 1000 :: NEXT
D :: IF CD=1 THEN CLOSE #1 !
174
440 STOP !152
450 DISPLAY AT(12,1)ERASE AL
```

(See Page 31)

USER NOTES CLASSIFIEDS

(Continued from Page 30)

```
L:"<< ERROR IN CATALOG DISK
>>" :: IF CD=1 THEN CLOSE #1
 :: CLOSE #2 !204
460 IF CD=0 THEN CLOSE #2 !0
10
470 FOR D=1 TO 1000 :: NEXT
D :: END !047
```

PRINT D/V80

```
100 ! < PRINT D/V80 > !059
110 ! < FROM EX-BASIC > !221
120 ! < BY R.W. AUGUST > !02
2
130 CALL SCREEN(5):: CALL CL
EAR !233
140 FOR I=0 TO 12 :: CALL CO
LOR(I,16,1):: NEXT I !125
150 DISPLAY AT(6,8):"< PRINT
 D/V80 >": : : :"DISK DRIVE
[1-4]:[1]" !214
160 ACCEPT AT(10,19)VALIDATE
(NUMERIC,"1234")SIZE(-1)BEEP
:N !179
170 N$=STR$(N):: DK$="DSK"&N
$&"." !061
180 DISPLAY AT(12,1):"ENTER
`filename'" !203
190 DISPLAY AT(14,1):DK$ !06
7
200 ACCEPT AT(14,6)SIZE(10)B
EEP:F$ :: FILE$=DK$&F$ !203
210 OPEN #1:FILE$,INPUT ,DIS
PLAY ,VARIABLE 80 !001
220 OPEN #2:"PIO",OUTPUT,DIS
PLAY :: DISPLAY AT(18,1):"PR
INTING FILE: ";F$ !012
230 PRINT #2:CHR$(27);"N";CH
R$(3);!048
240 LINPUT #1:A$ !187
250 IF EOF(1)THEN 280 !071
260 PRINT #2:A$ !174
270 GOTO 240 !063
280 CLOSE #2 !152
290 DISPLAY AT(22,1):"FILE "
;F$;" PRINTED" !114
300 CLOSE #1 :: END !164
```

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

Policy

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

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

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

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

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

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

FOR SALE FOR SALE

Try a MICROpendium classified to sell or buy TI/Geneve products

Buy/Sell Used Hardware and Software

National Used Software/Hardware 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 our members want to buy:

Myarc HFDC .....Offer	Triple Tech Card.....75.00
GRAMKracker (80K) 95.00	Rave 105 Keyboard Interface....85.00
TI composite monitor 75.00	80-column card.....Offer

Call our BBS for periodic 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 ❑ MasterCard ❑ VISA ❑ (check one)

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

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

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

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

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

GENEVE DISKS (SSSD unless specified)

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

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS

[illegible]
[illegible]
P.O. BOX 647
SEBECIA OH 45894