Subscription Fees

❑ 6 issues USA $35
❑ 6 issues Canada/Mexico $42.50
❑ 6 issues other countries surface mail
___ Surface mail $40
___ Air mail $52

Outside U.S., pay via postal or international money order; 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 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

$__________

No sales tax on magazine subscriptions. Texas residents add 8.25% sales tax on other items, including back issues and disk subscriptions.

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.

Micropendium
P.O. Box 1343
Round Rock, TX 78680

Covering the TI99/4A and Geneve home computers

MICROpendium

Volume 16 Number2 March/April 1999 $6

TIMUG'99

Speakers set for May 15 event
see page 18

Tim Tesch exits Geneve repair business
see Newsbytes

Saving and loading files between Funnelweb and a PC
see page 20

Assembly — reading disk sectors
see page 5

XB — utility to strip characters
see page 22

Reviews

MY-Basic v.4, DDI-ICON, TIPS Paint, Grabber, MYSIDEPRINTer, MYARK, MYFX, MYGolf, and MYWheel

CONTENTS

MICROpendium

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

No information published in the pages of MICROpendium may be used without permission of the publisher, Burns-Koloen Communications Inc. Only computer user groups

While all efforts are directed at providing factual and true information in published articles, the publisher cannot accept responsibility for errors that appear in advertising or text appearing in MICROpendium. The inclusion of brand names in text does not constitute an endorsement of any product by the publisher. Statements published by MICROpendium which reflect erroneously on individuals, products or companies will be corrected upon contacting the publisher.

Unless the author specifies, letters will be treated as unconditionally assigned for publication, copyright purposes and use in any other publication or brochure and are subject to MICROpendium's unrestricted right to edit and comment.

All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680.

Foreign subscriptions are $42.50 (Canada and Mexico); $40 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 & FAX: (512) 255-1512
Internet E-mail:
jkoloen@earthlink.net
Home page: http://
www.earthlink.net/~jkoloen/

John Koloen Publisher
Laura Burns Editor

The Art of Assembly

TIMUG '99

Funnelweb

Extended BASIC

Newsbytes

Reviews

User Notes

COMMENTS

Tim Tesch bids adieu to Geneve

The departure of Tim Tesch from the Geneve vendor ranks demonstrates a problem faced by TI users that is not likely to improve with time. Support for the TI/Geneve community is diminishing. With Tesch out of the picture, where does a Geneve user turn for repairs or upgrades? Look in a mirror. You are now your primary support person.

Tesch has done a remarkable job helping Geneve users keep their Geneve cards operating and deserves a hearty round of applause. Tim has always been fair in his dealings with users, which is more than can be said for many who provided similar services. Not only that, he's conscientious and charged too little for the work he did. I don't get the impression that money is an underlying factor to his decision to leave the Geneve marketplace. It seems to be a matter of time. But if money is involved, I'd be happy to pay an additional amount for the upgrade he did to my Geneve card last fall.

I wish Tim all the luck in the future, though it's us who are left without his services who need it the most.

Speaking of Geneve support, Charlie Good's MICROreviews is anything but "micro" this issue. He's taken on the task of covering all of Jim Uzzell's commercial and non-commercial software for the Geneve. Jim, who distributes his software under the banner of DDI Software, is another one of those developers that Geneve users ought to thank. I remember years ago when Myarc first issued its Advanced Basic, sometimes called MY-Basic. Jim would visit the MICROpendium office back then and complain mightily about how many problems the Myarc code had. How every time he tried to write a routine he'd find another problem with the basic interpreter. And then he'd finish by providing us with a demo copy of his latest Advanced Basic program that worked despite the problems of Advanced Basic itself. Finally, as you'll see from Charlie's review, Jim updated MY-Basic from version 3 to version 4, including a significantly better manual. Jim has been the one programmer who has consistently supported Geneve users with MY-Basic programs. And he continues to do so. Check out the review. There's a bunch of good stuff there.

And as long as I'm talking about development, C99 programmers should pay heed to a small article on page 41. The article mentions Oliver Arnold's CROM package. It sounds intriguing.

Continued on page 4

FEEDBACK

Last hurrah

I don't know you personally, but you have been real friends to everyone that has owned a TI99/4A. You have stuck by your printing press, even improving the quality of the magazine, long after many thought you would have given up. The integrity and "intestinal fortitude" shown by many in the TI community is something we dream of in our governmental representatives. We really can't praise you enough for what you have done to hold the "orphan group" together.

Having said that, and I really mean every word of it, I am renewing my subscription to your magazine for what I believe will be the last time. "Old habits die hard." I have had two TI systems set up, and used them, up until about three years ago. When I was unable to get repairs to some of the controller boards in the time I needed (new ones weren't available), I had to go elsewhere. Now *I'm* slowing down and just don't have the time nor the energy to use the TIs any longer. There are too many other things that I'm involved with to put the time I'd like to into the TI.

One thing I wanted to do was to learn to use assembly, but after several starts I gave up. That was before Bruce Harrison started his series of articles. He has done a *great* job, and is to be commended for it. As long as you have people willing to give the effort to do a good job like that, I believe the "Orphan" will live on forever. I'm sure it will outlive me, anyway.

So I am sending my wishes for the best, for you, your staff and the whole of the TI99/4A and Geneve community, along with my check.

GORDON H. MCCAA
LUGOFF, SOUTH CAROLINA

COMMENTS

Continued from page 3

TIMUG'99 coming up

May 15 isn't far away. Breaking from tradition, the TI Multi User Group conference isn't being hosted by Charlie Good. Nor is it being held in Lima, Ohio. This year it's being held in Brookpark, Ohio, and the sponsor is the TI-CHIPs user group. As of mid-March, three speakers were scheduled to make presentations.

Along with TIMUG'99 will be the presentation of the Jim Peterson Achievement Award. Nominees were listed in the January/February 1999 MICROpendium. Voting ends April 15 and may be done by e-mail or mail.

—JK

THE ART OF ASSEMBLY 75

Reading Disk Sectors

BY BRUCE HARRISON

This time we're into some really deep stuff, reading things directly by sectors from disks. Normally, of course, when we deal with files on disks, we let the drive controller handle all the hard stuff, like finding the contents of the file and putting records into a buffer in VDP RAM. Why, then, should we have to read a disk sector by sector? To help a friend in need is the answer.

Back in 1996, your author looked at the video tapes from the M.U.G. conference in Cleveland. On those was a lecture and demo by Mickey Cendrowski, showing her Load Master program. This program was one of those inspired by Mickey's own need for some way to make sense of the West Penn User Group's disk library.

She wanted, among other things, to have a program that would identify clearly many different file types that are available for the TI. The program was written in Extended BASIC and performed very well but slowly. The biggest problem seemed to be that files of the program (a.k.a memory image) type had to be lumped into large categories because there's no way in Extended BASIC to tell the difference between Editor/Assembler Option 5 files and those created by BASIC or Extended BASIC. Mickey had done the best she could, but for program files under 34 sectors in size, Load Master could not determine what was BASIC or XBASIC and what was E/A.

Headers the Answer

The difference between BASIC/XB programs and E/A Option 5 programs is in the content of the file header. This is the first six bytes of the file's content. When either BASIC or E/A loads a "program" file, the loader examines this header information, and can tell if the wrong type is being loaded. That idea wouldn't be useful, however, since trying to load the program file would destroy the Load Master program itself.

The only way to read the header in a controlled manner was to read by sectors from the disk. The end product, then, would have to be able to find the directory sector for each file on the disk, find from that the sector number of the file's header, read that sector, and then examine its content.

The Keys to the Disk

he keys that unlock all this are in sector 1 on the disk. In that sector are the sector numbers for the directory sectors of all the files on the disk. Sector 1 contains up to 127 words (two bytes each) that give the sector numbers for the

Continued on page 6

Continued from page 5

files on the disk in sorted order. That is, if two files are named AAA and AAB, the AAA file's directory sector number will come before that for AAB in sector 1. When new files are put onto a disk, the contents of sector 1 get sorted so that the new file is in the correct place by an ASCII sort. Unlike some other things we'll get to shortly, there's no mystery about reading the sector numbers in sector 1. Each pair of bytes, read as a word value, is the number of the sector containing the directory information for one file. If that number is zero, it means we're past the number of files stored on this disk.

Some Assembly Required

Like it says on the boxes at Toys 'R' Us, some Assembly is required to actually access the directory sectors and then the files' headers. It's impossible on the TI to read disk sectors from Extended BASIC, except of course by using CALL LINK to an Assembly routine which reads the sectors. It was obvious that Mickey's program would need such capability, so your author offered to help in that effort.

Mickey sent me the version 2.1 disk, and away we went. There were some Assembly routines in version 2.1 already, embedded by Todd Kaplan's ALSAVE method. It became obvious fairly quickly that the amount of Assembly stuff that would need to be added would exceed the space available in low memory, so "method two" became the method of choice. This means that we keep a program called LOAD on the disk, but it only places the previous Assembly routines (plus one) into low memory before RUNning another program called LOADMASTER. That second program contains the Extended BASIC stuff plus a lot of Assembly code embedded via Harry Wilhelm's High Memory Loader. This way, we get to have two sets of Assembly routines in use at the same time. The "old" routines plus Boot Tracking are kept in low memory, while the routines to read the catalog by sectors and identify various file types sits in High Memory along with the LOADMASTER program. There was still some leftover space in Low Memory, and some of that was used for temporary storage by the routines in High Memory. Thus we've made very efficient use of the whole expansion memory.

Thanks to Travis Watford

Through our friend Barry Traver, we had a disk in our collection that contained Travis Watford's T-Shell source code. Among other things, that source code contained a complete DSRLNK and the code to read sectors in an Extended BASIC environment. We had to modify Travis' DSRLNK slightly for our purposes, but the code that actually reads the sectors is largely his. In today's sidebar are portions of that code as modified. Travis' DSRLNK as modified is a very general-purpose one, which can be used in any environment and can perform linkage to just about any device service routine. Thank you, Travis!

The device service routine that we're using to read the sectors is of the "call" variety, in that the BLWP to DSRLNK is followed by DATA >A instead of the usual DATA 8. The PAB used is just two bytes in length, that being one byte of 1 (length) and a byte of >10 to call the sector service. The specifics as to the sector number, whether to read or write, etc. are placed in specific locations in RAM Pad before the DSRLNK call. The result of the call (barring error) is a dump of 256 bytes at our chosen buffer location in VDP RAM. We found it less troublesome to put both the PAB and the Buffer in the area above >37D7 in VDP. This way it doesn't get in the way of any other file accesses, nor does it interfere with the use of VDP RAM for the lookup tables and string variables that XB puts there. In other words, we found a "safe area" in VDP RAM to do our sector reads.

Before actually trying to integrate our Assembly stuff into Mickey's program, we ran a series of tests on this process, and found a real problem that we hadn't anticipated. In our system we have two "normal" floppy drives of the DS/SD variety, plus a number of Horizon RAMdisks. In our first trials, we found that our sector reading would work fine for Drives 1, 2, and 3, but for drives 4 and above it wouldn't work! Instead of a sector being read, we'd get the infamous "I Gotcha" report. This did no harm, but still it was maddening to see that on the screen. We consulted with Bud Mills, who quickly surmised that the problem, although being reported from our RAMdisk cards, was actually a result of something happening in our TI disk controller. He was right!

Through a series of carefully controlled experiments, we found that if one asks the TI controller to sector-read a disk with a number higher than 3, the TI controller reports an error in location >8350 of the RAM Pad. Through our experiments, we were able to determine that the error code reported in such a case is unique and different from the "no disk" or any other common problem.

Thus we put in a test after an error in the DSRLNK process, and if this unique error code showed up when we were accessing through CRU 1100, we could be sure it was simply the TI controller's problem. At that point, we modify the starting CRU address so that the DSRLNK will start looking at CRU address >1200, and retry the DSRLNK process. This makes for another use of the controversial self-modifying code idea, but it works as intended. Our Drive 3 RAMdisk, by the way, is at CRU address >1000, so we had no trouble reading sectors from that, since the DSRLNK found drive 3 before it got round to the >1100 TI disk controller. RAMdisks 4 and above were all at CRU addresses above >1100, and so fell victim to the error.

Having cleared that hurdle, we were ready to proceed with integration of the Assembly with Mickey's Extended BASIC program. Early in that process, we decided to put all of the process of identifying file types into the Assembly code,

Continued on page 8

Continued from page 7

so that cataloging and identification would become a single process instead of two separate ones. Doing this in Assembly made the whole process easier to manage and made execution much faster than it was in the version 2.1 of Load Master. This meant that tons of Mickey's original XB code were eliminated from the finished product, replaced by a couple of well-chosen CALL LINKs.

The Catlog/Ident Process

To do a complete cataloging job, we have to start by knowing which disk drive is desired, and have to read sector number 0 of the disk. sector 0 contains the name of the disk, the total number of sectors initialized on that disk, and the sector use map, which tells us which sectors have been allocated to files on the disk. We take the disk name and assign that to a string variable in the Extended BASIC realm. We use the total capacity and use map and create from them the numeric variables that indicate the used and available sectors. This all gets reported back to XB variables in one CALL LINK. Before exiting back to XB, this CALL LINK also reads sector 1 from the disk, and places that 256 bytes in some leftover space in low memory.

The next CALL LINK is a "biggie." It starts with the sector 1 data from the previous LINK. For each file, we take the two bytes from the sector 1 data that give us the directory sector location for that file. If this number is zero, we're past the last file. Otherwise, we go ahead and read the directory sector for one file into our VDP RAM buffer. Since we don't need all 256 bytes of that, we read only a portion into a storage location in low memory. The first ten or fewer bytes are the name of the file, so we extract that. The file's principal characteristics (e.g. type, size, protection) are contained early on in the directory sector. We have to separate that data on a bit-by-bit basis to determine what kind of file we're dealing with.

If the type indicates program, then we have to do more work before we can specify what kind of "program" file this is. Again we take some data bit-by-bit to find the number of the first sector of file content, then read that sector, and take eight bytes of it into low memory for examination. In most cases the first six bytes allow a complete identification of the program file.

For example, if the first two bytes are >FFFF, then this is an E/A Option 5 program file. If the first two bytes are zero, then this could be either an E/A Option 5 or a CHARA1 type file. If the fifth and sixth bytes are >07FA, then this is a CHARA1 file, not a program. Of course if the file is bigger than 33 sectors, it's automatically not an E/A Option 5 nor a CHARA1.

In order not to take up this whole issue with our sidebar, we've omitted large parts of the source code, including all the detailed code that identifies file types. If you're vitally interested, send me $1 and ask for the Load Master source code, and I'll send the complete source and its data files.

The Sidebar Code

Today's sidebar starts with the part that reads sector zero of the disk. When we enter this code, Register 4 already has the drive number in its high byte, and that register doesn't get changed until the whole cataloging and identifying process is finished. Error checking is done for each sector read at the end of the subroutine UDSR by moving the byte at >8350. If that byte has been cleared, then the sector read was successful. If not, then an error has occurred.

When we get that error from the TI disk controller for an attempt to read a sector from drive 4, we check to see if the CRU address in >83D0 is >1100, and if that's true whether the error code in >8350 is 7. If both conditions are true, we modify the DSRLNK code at two bytes past label DSR2A, then try again. This time the DSRLNK will start at CRU Address >1200, so it will find a ramdisk at or above that CRU address.

The code at label UDSR will look strange. Even though we're using a DRSLNK process, the PAB takes only two bytes, and other paramters for the CALL process are placed in locations in the RAM Pad before the BLWP. The word at >834C first gets set to all ones to indicate a read operation, then its left byte is set to the drive number. The sector number, as a word, is passed along to >8350. The location of the name length byte for the PAB is passed to >8356 as usual, but the PAB consists of only two bytes, one being the name length, and the other the "name", consisting only of a byte set to >10. The buffer address, instead of being part of the PAB, is placed at >834E. After all that is done, we BLWP @DSRLNK with DATA >A to perform the sector read operation.

A lot has been left out, so the sidebar is not anywhere near a complete entity. Its purpose is to supply some neat pieces of source code that you can excerpt for use in your own programs. The Travis Watford DSRLNK shown here will work for just about any "environment", even on a Geneve. Testing for the sector reading has shown that it works on any floppy disk drive with any drive controller, and also works with any RAMdisk of either the Horizon or Quest type.

The Extra Little Goodies

There are two in this sidebar that may prove useful. First is the tiny routine RSXB. To use this you'll need to include the Warren/Miller GPLLNK routine, which we've omitted. This resets all conditions to "startup" in XB without affecting the program in memory. The second little "goodie" provides a way to RUN another XB program using a string variable as the file name. Let's say your XB program has taken an input of "DSK1.MYPROG" into the string variable F$. You could then have your XB program run that by CALL LINK("RUNIT",F$). Include in your Assembly code all the stuff from label RUNIT through label TWO. This is used in Load Master to allow running a program selected from the catalog listing.

Continued on page 10

Continued from page 9

Hope you'll find some of this useful in your own programs. The topic for next time is undecided as usual. See you then.

Bruce Harrison can be reached at 5705 40th Place, Hyattsville, MD 20781 or e-mail Rottencat@aol.com.

Sidebar 75

```
* SIDEBAR 75
*
* FRAGMENTS OF SOURCE FOR LOAD MASTER V.2.2
* CODE BY BRUCE HARRISON EXCEPT AS NOTED
*
       DEF  DISKS,FILES,RUNIT,RSXB
NUMASG EQU  >2008        NUMERIC ASSIGN
NUMREF EQU  >200C        NUMERIC REF
STRASG EQU  >2010        STRING ASSIGN
STRREF EQU  >2014        STRING REF
XMLLNK EQU  >2018        XML LINKAGE
KSCAN  EQU  >201C        XB'S KEYSCAN
VSBW   EQU  >2020        XB'S VDP SB WRITE
VMBW   EQU  >2024        XB'S VDP MB WRITE
VSBR   EQU  >2028        XB'S VDP SB READ
VMBR   EQU  >202C        XB'S VDP MB READ
VWTR   EQU  >2030        XB'S VDP REG WRITE
ERR    EQU  >2034        XB'S ERROR REPORT
IOERR  EQU  >2400        CODE FOR I/O ERROR
CALPNT EQU  >832C        CALL POINTER
PAB    EQU  >3BE9        PAB VDP ADDRESS
PABUF  EQU  >3CEF        VDP BUFFER ADDR
GPLWS  EQU  >83E0        GPL WORK SPACE
GR4    EQU  GPLWS+8      GPL REG 4
GR6    EQU  GPLWS+12     GPL REG 6
FAC    EQU  >834A        F.P. ACCUMULATOR
NAMLEN EQU  >3600        LEFTOVER LOW MEM
NAMBUF EQU  NAMLEN+2       "
FSCBUF EQU  NAMBUF+32      "
DNBUF  EQU  FSCBUF+8       "
SEC1   EQU  DNBUF+12       "
TOTL   EQU  SEC1+256       "
NUMFLS EQU  TOTL+2         "
FILSIZ EQU  NUMFLS+2       "
RECSIZ EQU  FILSIZ+2       "
*
```

```
* FIRST SECTION GETS DISK NAME, CAPACITY, FREE SPACE
*
DISKS  LWPI WS            LOAD OUR WORKSPACE
       MOV  @ONES,R12     R12 NON-ZERO
       CLR  R3            SECTOR 0
       MOV  @OHEFF,@DSR2A+2 >0F00 TO START DSR
GTS0   BL   @UDSR         USE DSR
       JEQ  RDDN          IF NO ERROR, JUMP
       MOV  @>83D0,R0     CRU ADDR
       CI   R0,>1100      >1100?
       JNE  S0ERR         IF NOT, ERROR
       CB   @>8350,@SEVEN CHECK ERCODE 7
       JNE  S0ERR         IF NOT, ERROR
       MOV  R0,@DSR2A+2   >1100 TO START DSR
       JMP  GTS0          THEN TRY AGAIN
S0ERR  B    @ERROR        REPORT ERROR
RDDN   LI   R0,PABUF      POINT TO BUFFER
       LI   R1,DNBUF      AND STORAGE
       LI   R2,256        WHOLE SECTOR
       BLWP @VMBR         READ TO LOW MEM
       MOV  R1,R6         COPY ADDR TO R6
       AI   R1,9          ADD 9 TO R1
       LI   R2,10         TEN IN R2
GLLOP  CB   *R1,@H20      CHECK FOR SPACE AT END OF NAME
       JNE  LENFND        IF NOT, R2=LENGTH OF NAME
       DEC  R1            BACK UP ONE
       DEC  R2            DEC LENGTH
       JNE  GLLOP         REPEAT IF NOT 0
LENFND MOV  R2,@DNBUF-2   PUT NAME LENGTH IN PLACE
       MOV  @10(R6),R7    DISK CAPACITY
       CLR  R8            R8=0
       MOV  R7,R1         CAPACITY TO R1
       LI   R9,DNBUF+>38  START OF USE MAP
NXTWRD MOV  *R9+,R2       ONE WORD FROM MAP
       LI   R5,16         16 BITS TO EXAMINE
DEC1   DEC  R1            DEC COUNT BY 1
       JLT  ENDMAP        IF <0, JUMP
       SLA  R2,1          SHIFT R2 LEFT 1 BIT
       JNC  DEC5          JUMP IF NO CARRY
       INC  R8            ELSE INC COUNT OF USED
DEC5   DEC  R5            DECREMENT BIT COUNT
       JNE  DEC1          JUMP IF NOT 0
       JMP  NXTWRD        ELSE NEXT WORD
ENDMAP CLR  R0            NON-ARRAY
```

Continued on page 12

Continued from page 11

```
        LI   R1,1          2ND PARAM DISK NAME
        LI   R2,DNBUF-1    ADDR OF NAME STRING
        BLWP @STRASG       DISK NAME TO XB
        MOV  R7,@FAC       TOTAL SECTORS
        DECT @FAC          MINUS 2
        INC  R1            NEXT PARAMETER
        BL   @SNDINT       SEND TO XB
        MOV  R7,@FAC       TOTAL SECTORS
        S    R8,@FAC       - USED = FREE SECTORS
        INC  R1            NEXT PARAM
        BL   @SNDINT       SEND TO XB
        CLR  @SEC1         CLEAR WORD AT SEC1
        INC  R3            R3=1 - SECTOR 1
        BL   @UDSR         USE DSR LINK
        JEQ  RDSEC1        IF NO ERROR, JUMP
        B    @ERROR        ELSE REPORT ERROR
RDSEC1  LI   R0,PABUF      POINT AT BUFFER
        LI   R1,SEC1       AND LOW MEMORY ADDR FOR SECTOR 1
        LI   R2,256        WHOLE SECTOR
        BLWP @VMBR         READ TO LOW MEM
        B    @EXIT         EXIT TO XB
*
* NEXT SECTION GETS FILE DIRECTORY SECTORS
*
FILES   LWPI WS            OUR WORKSPACE
        CLR  R15           ARRAY ELEMENT 0
        LI   R9,SEC1       SECTOR 1
NXTFIL  INC  R15           NEXT ARRAY ELEMENT
        LI   R0,11*32+9
        LI   R2,13
        XOR  @ONES,R12     "WORKING" INDICATION ON-OFF
        JEQ  WRKOFF
        LI   R1,WRKSTR
        JMP  WRTWRK
WRKOFF  BL   @CLA
        JMP  NXTOK
WRTWRK  BLWP @PRSTR
NXTOK   CI   R9,TOTL       PAST END OF SEC1?
        JEQ  BEEKEY        IF SO, FINISHED
        MOV  *R9+,R3       NEXT DIRECTORY SECTOR NUMBER
        JNE  NXT1          IF NOT 0, JUMP
BEEKEY  B    @SNDNUL       DIRECTORY FINISHED
NXT1    BL   @CLRCEE       CLEAR C$
        BL   @CLREMM       CLEAR M$
```

```
        BL   @UDSR         READ DIRECTORY SECTOR
        JEQ  RDFNM         JUMP IF NO ERROR
        B    @ERROR        REPORT ERR
RDFNM   LI   R0,PABUF      VDP BUFFER
        LI   R1,NAMBUF     FILE NAME LOCATION
        LI   R2,32         ONLY 32 BYTES
        BLWP @VMBR         READ TO LOW MEM
        MOV  R1,R6         COPY ADDR TO R6
        AI   R1,9          ADD 9 TO R1
        LI   R2,10         TEN IN R2
CHSPC   CB   *R1,@H20      SPACE AT END OF NAME?
        JNE  GNLEN         IF NOT, GOT LENGTH
        DEC  R1            BACK ONE
        DEC  R2            DEC LENGTH IN R2
        JNE  CHSPC         JUMP IF NOT 0
GNLEN   MOV  R2,@NAMLEN    COPY R2 TO FILE NAME LENGTH
        MOV  @>E(R6),@>835E FILE SIZE WORD (SECTORS)
        INC  @>835E        ADD 1 FOR DIRECTORY SECTOR
        MOV  @>835E,@FILSIZ PUT AT FILE SIZE
        LI   R0,CEESTR+5   C$ PLUS 5
        BL   @SHWINT       NUMBER TO C$
        MOVB @>D(R6),R5    A FILE TYPE
        JEQ  ISPGM         IF ZERO, PROGRAM TYPE
        B    @NTPGM        ELSE NOT PROGRAM
ISPGM   LI   R1,PGSTR      "PROGRAM"
        LI   R0,CEESTR+5   INDICATE PROGRAM
        BL   @DISSTR       INTO C$
        C    *R10+,*R10+   ADD 4 TO R10
        MOVB @WS+21,@CEESTR C$ LENGTH
        CLR  @RECSIZ       CLEAR RECORD SIZE
        BL   @RHSEC        READ HEADER SECTOR
*
* FILE IDENTIFICATION STUFF OMITTED
*
SHWID   LI   R0,EMSTR+3    FILE IDENT
        BL   @DISSTR       PLACE IN M$
        INCT R10           ADD 2 TO R10
        SWPB R10           SWAP
        MOVB R10,@EMSTR    LENGTH OF M$
        MOV  R15,R0        ARRAY MEMBER TO R0
        LI   R1,1          1ST PARAM A$()
        LI   R2,NAMLEN+1   SEND A$()
        BLWP @STRASG       FILE NAME TO XB
        INC  R1            2ND PARAM
```

Continued on page 14

Continued from page 13

```
       LI   R2,CEESTR    C$
       BLWP @STRASG      SEND C$()
       INC  R1           3RD PARAM
       LI   R2,EMSTR     M$
       BLWP @STRASG      SEND M$()
IDEX   B    @NXTFIL      GO GET NEXT FILE
SNDNUL LI   R2,NAMLEN    POINT AT NAME LENGTH
       CLR  *R2          MAKE THAT 0
       MOV  R15,R0       CURRENT ARRAY MEMBER IN R0
NXTNUL CI   R0,127       IS THAT 127?
       JGT  GNUMF        JUMP IF GREATER
       LI   R1,1         FIRST PARAM (NAME)
       LI   R5,3         THREE TO SEND
SND1   BLWP @STRASG      ASSIGN NULL STRING
       INC  R1           INC PARAM NUM
       DEC  R5           DEC COUNT
       JNE  SND1         REPEAT IF NOT 0
       INC  R0           NEXT ARRAY ELEMENT
       JMP  NXTNUL       JUMP BACK
GNUMF  DEC  R15          DEC LAST ARRAY MEMBER
SNF    CLR  R0           NON-ARRAY
       LI   R1,4         4TH PARAM (N)
       MOV  R15,@FAC     NUMBER OF FILES
       MOV  R15,@NUMFLS  SAVE IN LOW MEM
       BL   @SNDINT      SEND N TO XB
       CLR  R14
       DIV  @FIFTEN,R14  DIVIDE R14-R15 BY 15
       MOV  R15,R15      ANY REMAINDER?
       JNE  DIVOK        IF REMAINDER, JUMP
       DEC  R14          ELSE DECREMENT QUOTIENT
DIVOK  MOV  R14,@FAC     PLACE AT >834A
       INC  R1           5TH PARAM (PP)
       BL   @SNDINT      SEND PARAM TO XB
       MOV  @NUMFLS,@FAC NUMBER
       C    @NUMFLS,@FIFTEN COMPARE TO 15
       JLT  SNFL         IF LESS, JUMP
       MOV  @FIFTEN,@FAC ELSE 15 TO FAC
SNFL   INC  R1           6TH PARAM (S2)
       BL   @SNDINT      SEND
       MOV  @OHEFF,@DSR2A+2 RESET DSR FOR >0F00 START
EXIT   LWPI GPLWS        GPL WORKSPACE
       B    @>6A         EXIT TO GPL INTERPRETER
ERROR  LWPI WS           OUR WS
```

```
       LI   R0,11*32+9
       LI   R2,13
       BL   @CLA         CLEAR "WORKING"
       MOV  @OHEFF,@DSR2A+2 RESET DSRLNK START
       LI   R0,IOERR     I/O ERROR CODE IN R0
       BLWP @ERR         USE XB ERROR REPORT
RSXB   BLWP @GPLLNK      USE GPLLNK (NOT SHOWN)
       DATA >6917        UNDOCUMENTED FEATURE
       LWPI >83E0        LOAD GPL WS
       B    @>6A         BACK TO GPL INTERPRETER
RUNIT  LWPI RNWS         PRELOADED REGISTERS
       MOVB @TWO,@>83C6  UNDO THE 3 KEY UNIT
       MOVB R5,*R2       MAX LEN 40
       BLWP @>2014       GET STRING VARIABLE
       MOVB *R2,@RNWS+13 ACTUAL LENGTH TO LOW BYTE R6
       MOVB R6,@LENBYT+1(R6) A ZERO AT END OF FL1
       MOV  R3,@>832C    ADDR FL1 TO >832C
       MOV  R4,@>2000    INIT VAL TO >2000 (8192)
       LWPI >83E0        LOAD GPL WS
       B    @>6A         GO TO GPL INTERPRETER
RNWS   DATA 0,1,LENBYT,FL1,>205A,>2800,0  R0 THRU R6
FL1    BYTE >82,>A9,>C7  TOKENS FOR ::,RUN,QUOTED STRING
LENBYT BSS  41           NAME GOES HERE
TWO    BYTE 2            TWO AS A BYTE
*
* SUBROUTINES
*
RHSEC  MOV  @>1C(R6),R3  FIRST SECTOR WORD TO R3
       MOV  R3,R7        COPY INTO R7
       ANDI R7,>000F     MASK ONLY LOW NYBBLE
       SWPB R7           PUT IN HIGH BYTE
       SRL  R3,8         MOVE HIGH BYTE R3 TO LOW BYTE
       MOVB R7,R3        ADD HIGH BYTE R7 TO R3
       MOV  R11,R10      SAVE R11 IN R10
       BL   @UDSR        USE DSR TO READ FIRST CONTENT SECTOR
       JEQ  RDFSC        IF NO ERROR, JUMP
       B    @ERROR       ELSE REPORT
RDFSC  LI   R0,PABUF     BUFFER
       LI   R1,FSCBUF    LOW MEM LOCATION
       LI   R2,8         FIRST 8 BYTES
       BLWP @VMBR        READ INTO LOW MEM
       B    *R10         RETURN (LOCATION IN R10)
UDSR   SETO @>834C       SET TO READ
*
```

Continued on page 16

Continued from page 15

```
* NOTE: TO WRITE A SECTOR, YOU'D USE CLR @>834C INSTEAD
* AND WOULD NEED TO PRE-LOAD PABUF WITH DESIRED CONTENT
*
       MOVB R4,@>834C    DRIVE #
       MOV  R3,@>8350    SECTOR #
       LI   R0,PAB       PAB VDP LOCATION
       LI   R2,2         TWO BYTES
       LI   R1,PABDT     PAB DATA
       BLWP @VMBW        WRITE PAB
       MOV  R0,@>8356    ADDR TO >8356
       LI   R5,PABUF     BUFFER IN VDP
       MOV  R5,@>834E    PLACE AT >834E
       BLWP @DSRLNK      USE DSR LINK
       DATA >A           "CALL" FUNCTION
       MOVB @>8350,R2    CHECK ERROR
       RT                RETURN
* T-SHELL SOURCE CODE BY TRAVIS WATFORD
* excerpt taken by B. Harrison
*-----------------------------------------------
*  DEVICE SERVICE ROUTINE
*DSRLNK
NPNTR  EQU  >8356
NLEN   EQU  >8354
CRULST EQU  >83D0
SAVADD EQU  >83D2
VDPWA  EQU  >8C02
VDPRD  EQU  >8800
DSRWS  BSS  32
DSRWS5 EQU  DSRWS+10
DSRWS0 EQU  DSRWS+1
DSRLNK DATA DSRWS,DSR
H20    BYTE >20
H2E    BYTE >2E
HAA    BYTE >AA
FNAME  BSS  7
       EVEN
DSR    MOV  *R14+,R5     GET DSR OFFSET
       SZCB @H20,R15     ZERO ALL BUT >20
       MOV  @NPNTR,R0    GET POINTER TO NAME LENGTH
       MOVB @DSRWS0,@VDPWA SET VDP READ ADDRESS
       NOP
       MOVB R0,@VDPWA
       AI   R0,-8        SET A POINTER TO ERROR RETURN BYTE
       MOVB @VDPRD,R1    GET THE NAME LENGTH
       MOVB R1,R3
       JEQ  DSR9         IF LEN=0 ABORT
       SRL  R3,8
```

```
       SETO R4
       LI   R2,FNAME     MOVE NAME TO CPU
DSR1   INC  R4
       CI . R4,7         SEE IF NAME LENGTH > 7
       JH   DSR9
       C    R4,R3        SEE IF NAME IS MAX LENGTH
       JEQ  DSR2
       MOVB @VDPRD,R1    GET CHAR
       MOVB R1,*R2+
       CB   R1,@H2E      SEE IF PERIOD
       JNE  DSR1
DSR2   CLR  @CRULST
       MOV  R4,@NLEN     SAVE NAME LENGTH
       INC  R4
       A    R4,@NPNTR    ADJUST NAME POINTER
       LWPI GPLWS
       CLR  R1
DSR2A  LI   R12,>0F00    START AT >1000
DSR3   SBZ  0
DSR3A  AI   R12,>0100    NEXT CRU BASE ADDRESS
       CLR  @CRULST
       CI   R12,>2000    QUIT AFTER CRU >1F00
       JEQ  DSR8
CRUOK  MOV  R12,@CRULST
       SBO  0
       LI   R2,>4000
       CB   *R2,@HAA     SEE IF A CARD IS PRESENT
       JNE  DSR3
       A    @DSRWS5,R2   ADD THE DSR OFFSET
       JMP  DSR5
DSR4   MOV  @SAVADD,R2
       SBO  0
DSR5   MOV  *R2,R2       SEE IF THERE ARE ANY ROUTINES
       JEQ  DSR3
       MOV  R2,@SAVADD
       INCT R2
       MOV  *R2+,R9      GET ROUTINE ADDRESS
       MOVB @NLEN+1,R5   GET NAME LENGTH
       JEQ  DSR7
       CB   R5,*R2+      SEE IF NAME LENGTH MATCHES
       JNE  DSR4
       SRL  R5,8
       LI   R6,FNAME     SEE IF NAME MATCHES
DSR6   CB   *R6+,*R2+    COMPARE CHARS
       JNE  DSR4
       DEC  R5
       JNE  DSR6         REPEAT
DSR7   INC  R1
```

Continued on page 18

THE ART OF ASSEMBLY 75

Continued from page 17

```
        BL   *R9            EXECUTE THE ROUTINE
        JMP  DSR4
        SBZ  0
        LWPI DSRWS          RESTORE REGISTERS
        MOVB @DSRWS0,@VDPWA SET VDP READ ADDRESS
        NOP
        MOVB R0,@VDPWA
        NOP
        MOVB @VDPRD,R1      GET THE NAME LENGTH
        SRL  R1,13          SEE IF ANY ERRORS
        JNE  DSR10
        RTWP
DSR8    LWPI DSRWS          RESTORE REGISTERS
DSR9    CLR  R1
DSR10   SWPB R1
        MOVB R1,*R13        PUT ERROR CODE IN R0 OF WS1
        SOCB @H20,R15       SET EQUAL BIT FOR ERROR
        RTWP
*
* DATA
*
WS      BSS  32             OUR WORKSPACE
PABDT   BYTE 1,>10          SECTOR READ/WRITE PAB DATA
*
* NOTE:  OTHER DATA HAS BEEN OMITTED
```

TIMUG'99

TIMUG'99 slates three speakers, invites others to do same

TIMUG'99 has been postponed until June 12, according to Glenn Bernasek, secretary of TI-Chips. The change notice came on April 1.

Three speakers are scheduled to
make presentations at TIMUG'99 on
May 15 in
event will b
Heights Re
Mansion in
The thre
King, and R
"The End c
showing how he gets the TI to
communicate using a 56K modem.

Ron's seminar will feature hardware
and software offered by RamCharged
led routes,
eburg
and Spang
forma-
Harry
/members.
stratos.net/harryholly/newsletter) and
also on Rich Polivka's web page. Map

TIMUG'99

and area information will be sent via e-mail or U.S. mail to those who register to attend TIMUG'99.

The conference schedule for TIMUG'99 is as follows:

Preconference get together party — 7 to 10 p.m., Friday, May 14, at Middleburg Heights Recreational Hall (Bagely Road, Middleburg Heights).

TI-99/4A and Myarc Geneve 9640 M.U.G. conference (TIMUG'99) — 7 a.m. to 7 p.m., Saturday, May 15, at the Spang Mansion (Kolthoff Road, Brookpark, Ohio).

Here is Saturday's schedule:

7 to 9 a.m. — Set-up tables and displays

9 a.m. to 5 p.m. — Seminars and demonstrations

1 to 2 p.m. — M.U.G. officers/ members conference

5 to 5:15 p.m. — Presentation of the Jim Peterson Achievement Awards

5:15 to 6 p.m. — Clean up

6 to 7 p.m. — Pizza party

If you would like to make a presentation or demonstration at the conference, contact Glenn Bernasek, 13246 Harper Road, Strongsville, Ohio 44136; Phone: (440)846-0865 (after 9 p.m. EST); E-mail: GBBasics@aol.com.

All seminars and demonstrations will be videotaped. These tapes will be available for purchase from TIers around the world at $5 per tape.

Continued on page 20

Peterson Award nominees listed

The following have been nominated for the 1999 Jim Peterson Achievement award. Deadline for voting is midnight at April 15. Votes received after the deadline will not be counted.

The awards will be presented at the TI Multi Users Group in Brookpark, Ohio, May 15.

Any TI user may vote for one recipient in each category.

Community Service: Rich Polivka, TI web page; MICROpendium, TI magazine; SouthWest 99ers Users Group, FestWest '98 - Lubbock, Texas; Tom Wills, list server.

TI99/4A Software: Bruce Harrison, Midi for the Super AMS; John Bull, Contract Bridge.

TI99/4A Hardware System 99 User Group (SNUG), SCSI board modifications; Michael Becker, High Speed GPL card; Don O'Neil, ongoing SCSI work

Myarc, Geneve 9640: Tim Tesch, enhanced the 9640 OS; Don Walden, SCSI for the Geneve

Submit votes to Glenn Bernasek, 13246 Harper Rd., Strongsville, OH 44136. E-mail: GBBasics@aol.com or dd314@Cleveland.Freenet.Edu.

TIMUG'99

Continued from page 19

There are no fees associated with attendance or participation in the conference. However, donations will be accepted to help defray TIMUG'99 conference site and support costs. Reservations for table space must be made by April 30.

FWB & PC

Saving and loading files directly between Funnelweb and a PC

BY CHARLES GOOD
LIMA OHIO USER GROUP

The following article was modified February 1999 from an article originally published in the September 1995 issue of the Lima User Group newsletter. — Ed.

Using the Funnelweb editor you can use LF and SF to LoadFiles and SaveFiles directly to and from a hard drive or floppy disk on an IBM-compatible PC. You can use the PC's hard drive to store all your important text files! Sometimes this can be done at very fast transfer rates comparable to saving and loading text using a 99/4A disk, and you don't need a modem on either the TI or the PC.

What you do need is an PC and TI computer cabled together between the TI's serial port and a PC com port. I find that a "modem cable" purchased for $8 at my local Wal-Mart works just fine with no modification, connecting my TI's RS232 to my laptop PC's COM1 port. If you want to try to make your own cable, the needed pin connections for such a cable are these:

TI	PC
1	1
2	2
3	3
6	20
7	7
20	6

To save text from Funnelweb to the PC

On the PC I use the Windows 3.1 terminal program (term.exe) on systems running either Windows 3.1 or Windows 95/98. The reason I use the older windows 3.1 program is that I can't figure out how to get Windows 95/98's Hyperterminal program to accept direct cable transfers. If you don't have this program I can e-mail it to you as an attached file or, if you send me $1 I will put it on an PC disk and snail mail it to you.

On the PC start Windows and boot the Windows 3.1 terminal program. Click on "Settings". Then click on "Terminal Preferences" and make sure CR does not generate CR/LF. You are now ready to receive text from Funnelweb.

On the 99/4A or Geneve write your document or LF a document into the Funnelweb editor. First set up the PC terminal program as described above, then enter SF from Funnelweb's command line. Use "RS232.BA=19200" (or BA=9600 on a 99/4A system) as the SaveFile file name. Yes I know the TI's RS232 isn't supposed to be able to handle a baud rate of 19200, but on my Geneve it works for me!

FWEB & PC

You need to specify the same baud rate in Window's terminal and Funnelweb's SF file name. Once you enter the SF file name your text will flow out of Funnelweb and across the serial cable into your PC. You will see the text appear on the PC screen. When the PC cursor stops displaying more text move the mouse pointer of the PC and click on "Stop." Your Funnelweb text has now been saved to a PC disk as an 80-column ASCII text file with no control characters and no tab markings.

To load text from a PC into Funnelweb

You can get text directly from a PC into a Geneve running Funnelweb, but unfortunately you can't do so with a 99/4A. I can easily get text out of a 99/4A to a PC as described above but I can't get it back in. I don't know why. I can use this technique to load text into Funnelweb on a 99/4A from my CC40 via the Hexbus RS232 and into Funnelweb running on a Geneve from my PC's com port, so I don't know why I can't load into Funnelweb on a 99/4A from a PC's com port. Getting text out of Funnelweb on a 99/4A and into a PC works well.

To load text from a PC into the Geneve load Funnelweb from GPL rather than using Exec. When using Exec v2.11 on the Geneve I can't break out of the RS232 loading process although I seem to remember that I could with an earlier version of Exec.

From Funnelweb's command line type LF and specify "RS232.BA=600" as the file name. Press enter and Funnelweb will appear to lock up as it waits for text to flow in from the RS232. You can't use baud rates faster then 600 for LoadFile even though you can use much faster baud rates to save files.

On the PC boot the Windows 3.1 terminal program. Click on "Settings" then on "Communications". Click on 600 baud, 7 bit, odd parity, the proper com port, and then click on OK. Now click on "Transfers" and then click on "Send Text File." Select the file name and drive of the text file you are loading into Funnelweb and click on OK.

At this point text will start flowing into Funnelweb and you will see line numbers increment at the right of the Funnelweb v5.x command line. When text stops flowing across the PC screen and the PC cursor returns to the terminal window and when Funnelweb's line numbers stop incrementing on Funnelweb's command line this means all the text is now in the TI's text buffer. Press FCTN-4 (the break key) and then press Enter to display this text on the TI's screen. It is this FCTN-4 keypress that fails when I load Funnelweb into my Geneve using Exec instead of GPL.

Utility strips text files

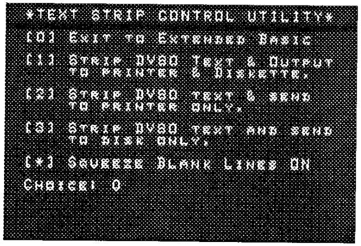
The following program is a utility that allows you to strip control codes from text documents and output the text to disk or printer. The program runs in Extended BASIC.

Simply load and run the program in Extended BASIC and you will see a menu consisting of the following options:

0 Exit to Extended BASIC
1 Strip DV80 text and output to printer and diskette
2 Strip DV80 text and send to printer only
3 Strip DV80 text and send to diskette only
* Squeeze blank lines

After selecting one of the options you are asked whether you want to use block format. Block format breaks up large text files into a size that TI-Writer can handle.

Next you are asked for the source disk and filename and then a destination disk and filename.



```
*TEXT STRIP CONTROL UTILITY*
[0] Exit to Extended Basic
[1] Strip DV80 Text & Output
    to Printer & Diskette.
[2] Strip DV80 text & send
    to Printer only.
[3] Strip DV80 text and send
    to disk only.
[*] Squeeze Blank Lines ON
Choice: 0
```

```
100 CALL CLEAR :: DISPLAY AT
(9,8):"* One Moment *":"":"
   * Loading Assembly *" ::
CALL INIT :: CALL LOAD(16376
,83,84,82,73,80,32,39,18)!18
6
110 CALL LOAD(8194,39,192,63
,248)!157
120 CALL LOAD(9460,0,0,0,0,0
,0,0,0,0,0,0,0,203,20,203,53
,203,78,203,231,204,71)!174
130 CALL LOAD(9482,204,150,2
04,228,205,29,205,75,205,96,
33,131,35,253,38,184,40,183,
41,182,42,195)!215
140 CALL LOAD(9504,43,193,44
,179,45,194,47,196,58,181,59
,180,60,191,61,190,62,192,94
,197,255,58)!167
150 CALL LOAD(9526,58,130,65
,84,240,71,79,133,73,70,132,
79,78,155,79,82,186,80,73,22
1,84,79)!215
160 CALL LOAD(9548,177,255,6
5,66,83,203,65,76,76,236,65,
78,68,187,65,83,67,220,65,84
,78,204)!238
170 CALL LOAD(9570,66,89,69,
3,67,79,78,1,67,79,83,205,68
,69,70,137,68,73,77,138,69,7
8)!208
180 CALL LOAD(9592,68,139,69
,79,70,202,69,88,80,206,70,7
9,82,140,73,78,84,207,76,69,
```

```
78,213)!184
190 CALL LOAD(9614,76,69,84,
141,76,79,71,208,77,65,88,22
3,77,73,78,224,78,69,87,0,78
,79)!047
200 CALL LOAD(9636,84,189,78
,85,77,4,79,76,68,5,80,79,83
,217,82,69,67,222,82,69,77,1
54)!000
210 CALL LOAD(9658,82,69,83,
6,82,78,68,215,82,85,78,169,
83,71,78,209,83,73,78,210,83
,81)!038
220 CALL LOAD(9680,82,211,83
,85,66,161,84,65,66,252,84,6
5,78,212,86,65,76,218,88,79,
82,188)!186
230 CALL LOAD(9702,255,66,65
,83,69,241,66,69,69,80,238,6
7,65,76,76,157,67,72,82,36,2
14,68)!135
240 CALL LOAD(9724,65,84,65,
147,128,128,128,128,128,128,
128,128,128,128,128,128,128,
128,128,128,128,128)!045
250 CALL LOAD(9746,128,128,1
28,128,128,128,128,128,128,1
28,128,128,128,229,226,228,2
28,228,228,228,228,228)!202
260 CALL LOAD(9768,228,228,2
28,228,228,228,228,228,228,2
28,228,228,228,228,228,228,2
28,228,228,228,228,228)!220
270 CALL LOAD(9790,228,228,2
28,228,228,228,228,228,228,2
27,56,188,6,160,50,218,7,66,
19,31,200,0)!135
280 CALL LOAD(9812,56,194,12
8,5,48,53,47,48,54,128,128,1
28,128,128,128,128,128,128,1
28,128,128,128)!046
290 CALL LOAD(9834,128,128,1
28,128,128,128,128,128,128,1
28,128,128,128,128,128,128,1
28,128,128,128,128,9)!092
300 CALL LOAD(9856,1,2,0,1,6
,160,32,0,0,1,0,0,6,28,255,2
55,0,0,0,0,37,110)!204
310 CALL LOAD(9878,43,124,0,
0,0,0,0,0,6,28,2,0,83,0,6,28
,0,0,6,28,0,0)!012
320 CALL LOAD(9900,3,51,43,4
4,0,0,0,0,6,79,78,76,73,78,6
9,6,72,85,78,71,85,80)!245
330 CALL LOAD(9922,2,1,0,0,2
16,1,131,116,4,32,32,28,4,19
3,208,96,131,117,152,1,38,16
0)!076
340 CALL LOAD(9944,19,8,216,
1,38,160,2,129,255,0,19,3,4,
195,208,193,16,14,4,193,2,12
)!059
350 CALL LOAD(9966,19,0,29,0
,29,7,2,44,0,64,31,21,22,4,4
,195,54,3,30,18,16,1)!147
360 CALL LOAD(9988,4,195,2,1
2,19,0,30,7,30,0,2,67,127,0,
2,224,38,242,2,0,0,0)!129
370 CALL LOAD(10010,2,1,0,1,
2,2,37,243,7,32,37,243,4,32,
32,20,2,10,37,243,216,26)!04
7
380 CALL LOAD(10032,36,244,2
,1,0,0,208,90,2,2,0,0,2,12,3
7,244,2,13,36,245,2,11)!196
```

Continued on page 24

Continued from page 23

```
390 CALL LOAD(10054,127,127,
146,220,18,1,16,4,2,11,31,31
,151,11,18,6,2,15,32,32,215,
79)!114
400 CALL LOAD(10076,5,140,5,
141,16,3,215,92,5,140,5,141,
16,0,5,130,6,194,144,66,6,19
4)!083
410 CALL LOAD(10098,22,232,1
6,0,2,2,0,0,2,15,32,32,2,13,
36,244,6,193,163,65,6,193)!1
23
420 CALL LOAD(10120,16,0,151
,79,22,8,6,193,6,1,6,193,6,1
3,144,129,22,247,2,1,1,0)!07
5
430 CALL LOAD(10142,16,0,216
,1,36,244,2,0,0,0,2,1,0,2,2,
2,36,244,4,32,32,16)!038
440 CALL LOAD(10164,2,224,13
1,224,4,224,131,124,4,96,0,1
12,32,32)!184
450 CALL CLEAR :: CALL SCREE
N(6):: FOR A=1 TO 12 :: CALL
 COLOR(A,16,6):: NEXT A :: C
ALL COLOR(13,2,6)!061
460 A$,B$,D$,H$,I$="" :: J$=
"ON" :: K$="N" :: A,B,D,G,H,
I,J,K=0 :: CALL CHAR(128,"00
0000FF00FF")!130
465 L$="1234567890ABCDEFGHIJ
KLMNOPQRSTUVWXYZ!@#$%^&*()=+
/-<,>.:;~[]_?'\`"&CHR$(34)!2
21
470 DISPLAY AT(1,1)ERASE ALL
:"*TEXT STRIP CONTROL UTILIT
Y*":"":"[0] Exit to Extended
 Basic" :: CALL HCHAR(2,1,12
8,32)!101
480 DISPLAY AT(5,1):"[1] Str
ip DV80 Text & Output    to
printer & Diskette." !173
490 DISPLAY AT(8,1):"[2] Str
ip DV80 text & send      to
printer only." !099
500 DISPLAY AT(11,1):"[3] St
rip DV80 text and send   to
 disk only.":"":"[*] Squeeze
 Blank Lines "&J$:"":"Choice
: 0" !051
510 ACCEPT AT(16,9)SIZE(-1)B
EEP VALIDATE("0123*"):A$ ::
IF LEN(A$)=0 THEN 510 !006
520 IF A$="*" AND J$="OFF" T
HEN J$="ON" :: GOTO 500 !045
530 IF A$="*".AND J$="ON" TH
EN J$="OFF" :: GOTO 500 !045
540 D=VAL(A$):: IF D=0 THEN
550 ELSE 560 !204
550 CALL CLEAR :: END !222
560 ON ERROR 570 :: GOTO 590
 !102
570 DISPLAY AT(21,1):"* Bad
Device or Filename *":"":"<P
ress any key to Continue>" !
143
580 CALL KEY(0,K,D):: IF D=0
 THEN 580 ELSE GOTO 460 !009
590 IF D<>2 THEN DISPLAY AT(
18,1):"Use block format [Y/N
] (Y)":"":"Block format will
 break up  large text files
into a sizethat TI-Writer ca
n load!" !162
600 ACCEPT AT(18,25)SIZE(-1)
```

```
VALIDATE("YN"):K$ :: DISPLAY
 AT(18,1):"":"":"":"":"" !24
8
610 DISPLAY AT(18,1):"Source
: DSK1." :: ACCEPT AT(18,9)S
IZE(-15)BEEP VALIDATE(L$):H$
 :: IF LEN(H$)<6 THEN 470 EL
SE A$=H$ !229
620 IF D=2 THEN 650 !134
630 DISPLAY AT(19,1):"Destin
ation: DSK2." :: ACCEPT AT(1
9,14)SIZE(-15)BEEP VALIDATE(
L$):D$ :: IF D$="" THEN 470
!152
640 IF H$=D$ THEN 570 !203
650 ON ERROR STOP :: ON ERRO
R 860 !194
660 OPEN #1:H$,INPUT !252
670 J,B=0 !246
680 IF D=2 THEN 690 :: OPEN
#2:D$,OUTPUT :: K=K+1 !031
690 IF D=3 THEN 700 :: OPEN
#3:"PIO" !058
700 IF EOF(1)=0 THEN LINPUT
#1:A$ :: CALL LINK("STRIP",A
$,B$)ELSE 790 !079
710 IF LEN(B$)=1 AND B$=CHR$
(32)THEN B$="" !208
720 DISPLAY AT(21,1):"Files
Created:"&STR$(K):: DISPLAY
AT(22,1):"  Total Bytes:"&ST
R$(B):: DISPLAY AT(23,1):"Li
nes Printed:"&STR$(J):: IF J
$="ON" THEN DISPLAY AT(24,1)
:"Lines Deleted:"&STR$(I)!22
3
730 IF J$="OFF" THEN 760 ELS
E DISPLAY AT(14,2)SIZE(-1):"
*" !224
740 IF B$="" THEN G=G+1 :: I
F G>1 THEN DISPLAY AT(14,2)S
IZE(-1):" " :: I=I+1 :: GOTO
 700 !134
750 IF B$<>"" THEN H=H+1 ::
IF H>2 THEN G,H=0 !155
760 J=J+1 :: IF D=1 THEN PRI
NT #2:B$ :: PRINT #3:B$ :: G
OTO 820 !168
770 IF D=2 THEN PRINT #3:B$
:: GOTO 820 !233
780 IF D=3 THEN PRINT #2:B$
:: GOTO 820 !233
790 IF D=1 THEN CLOSE #1 ::
CLOSE #2 :: CLOSE #3 !252
800 IF D=2 THEN CLOSE #1 ::
CLOSE #3 :: RUN !014
810 CLOSE #1 :: CLOSE #2 ::
DISPLAY AT(17,1):"":"":"":""
:"":"        * End Of Job *":
"":"      * Press Any Key *"
:: GOTO 580 !160
820 IF K$="N" THEN 700 !254
830 B=B+LEN(B$):: IF B<20480
 THEN 700 !180
840 IF D=1 THEN CLOSE #2 ::
J=0 :: CALL NAME(D$,D$):: GO
TO 670 !184
850 IF D=2 THEN 670 !154
860 CLOSE #2 :: CALL NAME(D$
,D$):: GOTO 670 !005
870 PRINT : :"*Disk Error! *
": :"<Press Any Key to Conti
nue>" :: GOTO 580 !008
880 GOTO 880 !194
890 SUB NAME(A$,B$):: A=ASC(
```

Continued on page 26

EXTENDED BASIC

Continued from page 25

```
SEG$(A$,LEN(A$),LEN(A$))::
A=A+1 :: B$=SEG$(A$,1,LEN(A$
)-1)&CHR$(A):: DISPLAY AT(19
,1):"  Adding File: "&B$ ::
SUBEND !035
```

READER TO READER

The Rev. George B. Salley Jr., 310 McLaws St., Savannah, GA 31405-5621, writes:

I have a complete collection of MICROpendium from February 1994 through June 1995. I will be glad to send them to anyone for the cost of shipping.

I also have six or seven metal-clad TI99/4A consoles; one Expansion box (with a 5.5-inch disk drive and a 232 card); a number of unused 5.5-inch disks plus a number with programs, data files, etc.; a great many game modules plus modules for TI-Writer, Editor/Assembler, Spreadsheet, etc.; a number of instruction manuals, many other books on the TI99/4A, etc.; a speech module; a number of TV set interfaces; a Star (Epson-type) printer which is missing a board which apparently is no longer made (and, therefore, does not work); an Axiom printer interface; a modem which uses a telephone receiver; a number of other things too numerous to name. I will also be happy to send any or all of these things to anyone for the cost of shipping.

NEWSBYTES

Tesch discontinues Myarc repairs

As of March 30, Tim Tesch of S&T Software is no longer accepting Myarc products for upgrade or repair. Tesch lists three major reasons for this decision:

1. My full-time job continues to take more and more time out of my hectic schedule. I could choose a new job, but I still like what I am doing, and must therefore commit to the responsibilites I've been given.

2. Years ago I started working on Myarc hardware to assist Don Walden. Later, that turned into helping the people who had sent Don cards months or years in the past. Finally, that turned into accepting responsibility for repairing and upgrading all Myarc equipment. This was fine for a while, but I noticed I was not programming much anymore. My Geneve is in disarray, some cards in use at my test station; my hard drive waiting for a replacement. The word "burnout" comes to mind.

3. I want everyone to get their equipment back, repaired or upgraded where possible. If not possible, my only option will be to return the card and any prepayment (if one was made) and hope someone else can remedy the trouble. I hate to send back a non-working card, but it is the only way to ensure that each person has his/her card back in their possession.

Everyone has his/her breakpoint (programmer pun?) and mine has been reached. It's time for me to step away for a bit. Once again, a big thank you to everyone for their continued support. Is is, and will always be, greatly appreciated.

For more information, Tesch can be reached at ttesch@juno.com.

NEWSBYTES

Genial TRAVelER available on the Internet

Barry Traver's Genial TRAVelER files are available from the Western Horizon web site. They can be found in a directory called "genitrav" under "pub" at ftp://ftp.whtech.com. Don O'Neill hosts the web site.

The Genial TRAVelER package was developed by CaDD Electronics and is available to any TI user. The package consists of two files: genitrav.exe (about 15 mb) and gtdisks.exe (about 3 mb). The genitrav.exe file is a self-extracting file that will create a 54 mb Adobe Acrobat pdf file. This pdf file contains an introduction to Genial TRAVelER, a list of all issues and their contents, a list of all files on the Genial TRAVelER disks and every D/V80 file on the Genial TRAVelER disks. It totals 925 pages. The gtdisks.exe file is a self-extracting file that contains all the Genial TRAVelER disks in PC99 format.

14th TI-Treffen set

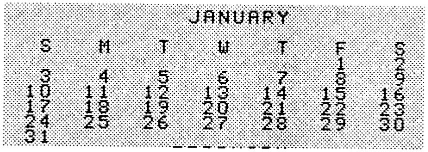
The 14th International TI-Treffen is scheduled for Oct. 1-3 in Freiberg, Germany, hosted by TI-Club Errorfree, one of the oldest computer clubs in Germany. The fair will take place at the Kleintierzuchtvereinsheim Geisingen, Austrasse 3, D-71691 Freiberg, Neckar, Germany.

Hardware installation will be Oct. 1, with the doors opening at 8 a.m. for general admission Oct. 2-3. Food and drinks will be available on site.

For further information, contact Wolfgang Bertsch Helenenburgweg 61, 74321 Bietigheim-Biss, Germany; or Oliver Arnold, Implerstrasse 8, 81371 München, Germany; or e-mail Martin Zeddies at MartinZeddies@MAUS.WOB.DE.

Looking for something to do?
Attend TIMUG'99
May 15

Everybody needs a calendar



```
              JANUARY

  S    M    T    W    T    F    S
                              1    2
  3    4    5    6    7    8    9
 10   11   12   13   14   15   16
 17   18   19   20   21   22   23
 24   25   26   27   28   29   30
 31
```

If you didn't buy a 1999 calendar by February, chances are you'll have to look hard to find one. Commercially-produced calendars have a short shelf life and businesses that sell them don't keep them around very long.

With the following program, you'll always be able to print out a calendar when you need one. The program was written by Frank C. Geitzler and updated by Phil Townsend. The program runs in Extended BASIC.

This calendar prints out a 12-month calendar on a single page with the year printed at the top in large letters. The calendar may also be output to the screen, one month at a time.

CALENDAR

```
100 REM SAVE "DSK1.CALENDAR"
 !200
110 !!131
120 !      WRITTEN BY !043
130 !   FRANK C. GEITZLER !1
08
140 !      DECEMBER, 1986 !12
6
150 !  THIS PROGRAM MAY BE !
225
160 ! FREELY COPIED, CHANGED
 !116
170 ! AND DISTRIBUTED. IT MA
Y !203
180 !   BE PUBLISHED BY ANY
!237
190 ! NON-PROFIT ORGANIZATIO
N !068
200 ! PROVIDING THIS CREDIT
!136
210 !  IS INCLUDED, AND A !0
71
220 !      COPY OF THE !020
230 ! PUBLICATION IS SENT TO
 !182
240 ! THE AUTHOR AT ADDRESS
!082
250 ! B3A 2K4, DARTMOUTH, !0
90
260 ! NOVA SCOTIA, CANADA !1
58
270 ! *********************
 !063
280 ! UPDATED..Phil Townsend
 !005
290 !   Nov. 14, 1988 !239
300 !        for !202
310 !   The KAWARTHA 99ERS !
243
320 !         box 373 !137
```

```
330 !   Peterborough, Ont. !
132
340 !    CANADA   K9J 6Z3 !1
72
350 !!131
360 !***********************
 !073
365 CALL SCREEN(15)!200
370 OPTION BASE 0 !136
380 DIM M$(12,7,7),MO$(12),M
DAYS(12),NUM$(4,24),U$(4),YD
(4)!055
390 PD$="PIO" :: D$="0123456
789" :: BYR=1901 :: BDAY=3 :
: YR4=365*4+1 :: NCPY=1 :: N
M=3 !028
400 DEF MOD4(X)=X-INT(X/4)*4
 !079
410 DEF MOD7(X)=X-INT(X/7)*7
 !088
420 DISPLAY AT(1,11)ERASE AL
L BEEP:"CALENDAR":;:;:"WHAT
YEAR DO YOU WANT? 1989" !177
430 ACCEPT AT(4,24)SIZE(-4)V
ALIDATE(D$):CY$ :: CYR=VAL(C
Y$)!117
440 FOR I=0 TO 4 :: READ U$(
I):: NEXT I !227
450 FOR I=1 TO 12 :: READ MO
$(I),MDAYS(I):: NEXT I !066
460 FOR I=1 TO 12 :: RESTORE
 1030 :: FOR J=1 TO 7 :: REA
D MN$ :: M$(I,J,1)=" "&MN$ :
: NEXT J :: NEXT I !032
470 DISPLAY AT(5,10)ERASE AL
L:"PLEASE WAIT":;:"        I
AM SETTING UP":;:;:;:;:;:"
     YOUR CALENDAR" !104
480 IF MOD4(CYR)=0 THEN MDAY
S(2)=29 ! ADJUST FOR LEAP YE
AR !059
490 AYR=CYR-BYR :: AYR=MOD7(
INT(AYR/4)*YR4+MOD4(AYR)*365
)! ADJUST FOR PREVIOUS YEARS
 !146
500 DA=BDAY+AYR :: IF DA>7 T
HEN DA=DA-7 !149
510 ROW=2 !177
520 MONTH=1 :: MCTR=1 !174
530 FOR I=1 TO 365 :: DISPLA
Y AT(10,11):"DAY";I;"OF" !15
8
540 M$(MONTH,DA,ROW)=SEG$("
 "&STR$(MCTR),LEN(STR$(MCTR)
)+1,2)!101
550 MCTR=MCTR+1 :: DA=DA+1 :
: IF MCTR>MDAYS(MONTH)THEN M
ONTH=MONTH+1 :: MCTR=1 :: RO
W=2 :: IF DA>7 THEN DA=1 !10
3
560 IF DA>7 THEN DA=1 :: ROW
=ROW+1 !161
570 NEXT I !223
580 DISPLAY AT(2,9)ERASE ALL
:"DO YOU WANT":;:"YOUR CALEN
DAR PRINTED? (Y/N)" :: ACCEP
T AT(4,25)BEEP SIZE(-1)VALID
ATE("YN"):A$ !220
590 IF A$="N" THEN GOSUB 610
 :: GOTO 580 ELSE GOSUB 670
:: GOTO 580 !192
600 REM DISPLAY CALENDAR !04
2
610 FOR I=1 TO 12 :: PRINT S
EG$(RPT$(" ",14)&MO$(I),INT(
```

Continued on page 30

Continued from page 29

```
LEN(MO$(I))/2),14+LEN(MO$(I)
)):;:;:: FOR J=1 TO 7 :: FOR
 K=1 TO 7 !042
620 PRINT USING U$(0):M$(I,K
,J);:: NEXT K :: PRINT :: NE
XT J :: GOSUB 630 :: NEXT I
:: PRINT :: RETURN !169
630 DISPLAY AT(24,1):" PRESS
 <ENTER> TO CONTINUE" :: ACC
EPT AT(24,28):X$ :: RETURN !
084
640 REM PRINT CALENDAR !161
670 ! SKIPPING WAS NOT PERFE
CTED, SO THIS IS COMMENTED O
UT !172
680 DISPLAY AT(10,1):"PRINTE
R NAME? ":PD$ :: ACCEPT AT(1
1,1)SIZE(-28):PD$ !169
690 DISPLAY AT(14,1):"NUMBER
 OF COPIES? ";NCPY :: ACCEPT
 AT(14,20)SIZE(-3):NCPY !173
700 OPEN #1:PD$,OUTPUT,DISPL
AY ,VARIABLE 80 !218
710 FOR CPYCTR=1 TO NCPY !00
4
720 DISPLAY AT(16,1):"COPY "
;CPYCTR;" OF ";NCPY;" COPIES
." !024
730 GOSUB 890 !205
740 FOR I=1 TO 12 STEP NM !1
84
750 FOR M=0 TO NM-1 !152
760 PRINT #1:SEG$(RPT$(" ",1
4)&MO$(I+M)&RPT$(" ",14),INT
(LEN(MO$(I+M))/2),26);!190
770 NEXT M :: PRINT #1:;:;:!
127
780 FOR J=1 TO 7 !PRINT WEEK
S OF MONTH !073
790 FOR M=0 TO NM-1 :: PRINT
 #1:"    ";!PRINT ONE WEEK O
F MONTH !026
800 FOR K=1 TO 7 !PRINT ONE
DAY OF WEEK !081
810 PRINT #1,USING U$(NM):M$
(I+M,K,J);!077
820 NEXT K !225
830 PRINT #1:" ";:: NEXT M !
073
840 PRINT #1 :: NEXT J !245
850 NEXT I !223
860 PRINT #1:CHR$(12)&CHR$(1
3):: NEXT CPYCTR :: CLOSE #1
 !231
870 RETURN !136
880 REM PRINT CALENDAR YEAR
!242
890 FOR CD=1 TO 4 !121
900 YD(CD)=POS(D$,SEG$(CY$,C
D,1),1)-1 !005
910 GOSUB 3440 :: NEXT CD !1
08
920 FOR CL=1 TO 24 !180
930 FOR CD=1 TO 4 !121
940 PRINT #1:NUM$(CD,CL);!07
0
950 NEXT CD !029
960 PRINT #1 !147
970 NEXT CL !037
980 PRINT #1:;:;:!026
990 RETURN !136
1000 DATA " ## ","   ##   ",
" ## ","## ",## " !090
1010 DATA JANUARY,31,FEBRUAR
Y,28,MARCH,31,APRIL,30,MAY,3
```

```
1,JUNE,30 !162
1020 DATA JULY,31,AUGUST,31,
SEPTEMBER,30,OCTOBER,31,NOVE
MBER,30,DECEMBER,31 !1511030
 DATA S,M,T,W,T,F,S !124
1040 REM NUMBER 0 !211
1050 DATA " 1
         " !197
1060 DATA " 2
         " !198
1070 DATA " 3
         " !199
1080 DATA " 4           0000
         " !008
1090 DATA " 5         000000
         " !041
1100 DATA " 6        00    0
0        " !010
1110 DATA " 7       00
00       " !011
1120 DATA " 8      00
 00      " !012
1130 DATA " 9      00
 00      " !013
1140 DATA "10     00
  00     " !021
1150 DATA "11     00
  00     " !022
1160 DATA "12     00
  00     " !023
1170 DATA "13     00
  00     " !024
1180 DATA "14     00
  00     " !025
1190 DATA "15     00
  00     " !026
1200 DATA "16      00
 00      " !027
1210 DATA "17      00
 00      " !028
1220 DATA "18       00
00       " !029
1230 DATA "19        00    0
0        " !030
1240 DATA "20         000000
         " !054
1250 DATA "21           0000
         " !023
1260 DATA "22
         " !216
1270 DATA "23
         " !217
1280 DATA "24
         " !218
1290 REM NUMBER 1 !212
1300 DATA " 1
         " !197
1310 DATA " 2
         " !198
1320 DATA " 3
         " !199
1330 DATA " 4          111
         " !251
1340 DATA " 5          111
         " !252
1350 DATA " 6           11
         " !236
1360 DATA " 7           11
         " !237
1370 DATA " 8           11
         " !238
1380 DATA " 9           11
         " !239
1390 DATA "10           11
         " !247
```

Cont. on page 32

Continued from page 31

```
1400 DATA "11            11
         " !248
1410 DATA "12            11
         " !249
1420 DATA "13            11
         " !250
1430 DATA "14            11
         " !251
1440 DATA "15            11
         " !252
1450 DATA "16            11
         " !253
1460 DATA "17            11
         " !254
1470 DATA "18            11
         " !255
1480 DATA "19            11
         " !000
1490 DATA "20        111111
         " !060
1500 DATA "21        111111
         " !061
1510 DATA "22
         " !216
1520 DATA "23
         " !217
1530 DATA "24
         " !218
1540 REM NUMBER 2 !213
1550 DATA " 1
         "," 2
          " !171
1560 DATA " 3
         "," 4        22222
2         " !027
1570 DATA " 5       22222222
22       "," 6      222
 222      " !211
1580 DATA " 7     222
 222     "," 8      22
   22     " !107
1590 DATA " 9
 222     ","10
 222      " !046
1600 DATA "11              2
22       ","12              2
22        " !057
1610 DATA "13            22
         ","14            22
          " !025
1620 DATA "15          22
         ","16          22
          " !029
1630 DATA "17        22
         ","18        22
          " !033
1640 DATA "19      22
         ","20     222222222
22222     " !244
1650 DATA "21    2222222222
2222     ","22
          " !203
1660 DATA "23
         ","24
          " !211
1670 REM NUMBER 3 !214
1680 DATA " 1
         " !197
1690 DATA " 2
         " !198
1700 DATA " 3
         " !199
1710 DATA " 4         333333
         " !058
1720 DATA " 5       33333333
```

```
33       " !135
1730 DATA " 6      333
333      " !060
1740 DATA " 7     333
 333     " !061
1750 DATA " 8     33
  33     " !024
1760 DATA " 9
 333     " !006
1770 DATA "10
333      " !014
1780 DATA "11            33
33       " !034
1790 DATA "12           3333
         " !035
1800 DATA "13           3333
         " !036
1810 DATA "14            33
33       " !037
1820 DATA "15
333      " !019
1830 DATA "16
 333     " !020
1840 DATA "17     33
  33     " !040
1850 DATA "18     333
 333     " !079
1860 DATA "19      333
333      " !080
1870 DATA "20       33333333
33       " !148
1880 DATA "21         333333
         " !073
1890 DATA "22
         " !216
1900 DATA "23
         " !217
1910 DATA "24
         " !218
1920 REM NUMBER 4 !215
1930 DATA " 1
         " !197
1940 DATA " 2
         " !198
1950 DATA " 3
         " !199
1960 DATA " 4
 4       " !220
1970 DATA " 5
44       " !241
1980 DATA " 6              4
44       " !006
1990 DATA " 7             44
44       " !027
2000 DATA " 8            444
44       " !048
2010 DATA " 9           444
44       " !049
2020 DATA "10          444
44       " !057
2030 DATA "11         444
44       " !058
2040 DATA "12        444
44       " !059
2050 DATA "13       44444444
4444     " !200
2060 DATA "14      444444444
4444     " !221
2070 DATA "15
44       " !002
2080 DATA "16
44       " !003
2090 DATA "17
44       " !004
2100 DATA "18
```

Continued on page 34

Continued from page 33

```
44       " !005
2110 DATA "19
44       " !006
2120 DATA "20
44       " !254
2130 DATA "21
44       " !255
2140 DATA "22
         " !216
2150 DATA "23
         " !217
2160 DATA "24
         " !218
2170 REM NUMBER 5 !216
2180 DATA " 1
         " !197
2190 DATA " 2
         " !198
2200 DATA " 3
         " !199
2210 DATA " 4     5555555555
5555     " !238
2220 DATA " 5     5555555555
5555     " !239
2230 DATA " 6      55
         " !244
2240 DATA " 7      55
         " !245
2250 DATA " 8      55
         " !246
2260 DATA " 9     5555555555
55       " !201
2270 DATA "10    5555555555
555      " !230
2280 DATA "11
 555     " !021
2290 DATA "12
  55     " !001
2300 DATA "13
  55     " !002
2310 DATA "14
  55     " !003
2320 DATA "15
  55     " !004
2330 DATA "16
  55     " !005
2340 DATA "17    55
  55     " !048
2350 DATA "18    555
 555     " !091
2360 DATA "19     555
555      " !092
2370 DATA "20      55555555
55       " !168
2380 DATA "21        555555
         " !085
2390 DATA "22
         " !216
2400 DATA "23
         " !217
2410 DATA "24
         " !218
2420 REM NUMBER 6 !217
2430 DATA " 1
         " !197
2440 DATA " 2
         " !198
2450 DATA " 3
         " !199
2460 DATA " 4              6
66       " !010
2470 DATA " 5             66
6        " !011
2480 DATA " 6            666
         " !012
```

```
2490 DATA " 7           666
         " !013
2500 DATA " 8          666
         " !014
2510 DATA " 9         666
         " !015
2520 DATA "10       666
         " !023
2530 DATA "11      666
         " !024
2540 DATA "12     666666666
         " !157
2550 DATA "13     666666666
66       " !202
2560 DATA "14    6666
666      " !115
2570 DATA "15    666
 666     " !094
2580 DATA "16    66
  66     " !051
2590 DATA "17    66
  66     " !052
2600 DATA "18    666
 666     " !097
2610 DATA "19     666
666      " !098
2620 DATA "20      66666666
66       " !178
2630 DATA "21        666666
         " !091
2640 DATA "22
         " !216
2650 DATA "23
         " !217
2660 DATA "24
         " !218
2670 REM NUMBER 7 !218
2680 DATA " 1
         " !197
2690 DATA " 2
         " !198
2700 DATA " 3
         " !199
2710 DATA " 4     7777777777
7777     " !010
2720 DATA " 5     7777777777
7777     " !011
2730 DATA " 6     77
  77     " !038
2740 DATA " 7
 77      " !249
2750 DATA " 8
 77      " !250
2760 DATA " 9
77       " !251
2770 DATA "10
77       " !003
2780 DATA "11             7
7        " !004
2790 DATA "12             7
7        " !005
2800 DATA "13            77
         " !006
2810 DATA "14            77
         " !007
2820 DATA "15           77
         " !008
2830 DATA "16           77
         " !009
2840 DATA "17          77
         " !010
2850 DATA "18          77
         " !011
2860 DATA "19         77
         " !012
```

Continued on page 36

Continued from page 35

```
2870 DATA "20          777
         " !027
2880 DATA "21          77
         " !005
2890 DATA "22
         " !216
2900 DATA "23
         " !217
2910 DATA "24
         " !218
2920 REM NUMBER 8 !219
2930 DATA " 1
         " !197
2940 DATA " 2
         " !198
2950 DATA " 3
         " !199
2960 DATA " 4         888888
         " !088
2970 DATA " 5       88888888
88       " !185
2980 DATA " 6      888
888      " !090
2990 DATA " 7     888
 888     " !091
3000 DATA " 8     88
  88     " !044
3010 DATA " 9     888
 888     " !093
3020 DATA "10      888
888      " !101
3030 DATA "11       8888  88
88       " !150
3040 DATA "12         888888
         " !103
3050 DATA "13         888888
         " !104
3060 DATA "14       8888  88
88       " !153
3070 DATA "15      888
888      " !106
3080 DATA "16     888
 888     " !107
3090 DATA "17     88
  88     " !060
3100 DATA "18     888
 888     " !109
3110 DATA "19      888
888      " !110
3120 DATA "20       88888888
88       " !198
3130 DATA "21         888888
         " !103
3140 DATA "22
         " !216
3150 DATA "23
         " !217
3160 DATA "24
         " !218
3170 REM NUMBER 9 !220
3180 DATA " 1
         " !197
3190 DATA " 2
         " !198
3200 DATA " 3
         " !199
3210 DATA " 4         999999
         " !094
3220 DATA " 5       99999999
99       " !195
3230 DATA " 6      999
999      " !096
3240 DATA " 7     999
 999     " !097
3250 DATA " 8     99
```

```
  99     " !048
3260 DATA " 9     99
  99     " !049
3270 DATA "10     999
 999     " !107
3280 DATA "11      999
9999     " !133
3290 DATA "12       99999999
999      " !234
3300 DATA "13         999999
999      " !185
3310 DATA "14              9
99       " !036
3320 DATA "15             99
9        " !037
3330 DATA "16            999
         " !038
3340 DATA "17           999
         " !039
3350 DATA "18          999
         " !040
3360 DATA "19         999
         " !041
3370 DATA "20        999
         " !033
3380 DATA "21       999
         " !034
3390 DATA "22
         " !216
3400 DATA "23
         " !217
3410 DATA "24
         " !218
3420 STOP !152
3430 REM READ CALENDAR CHARA
CTERS !048
3440 IF YD(CD)=0 THEN GOSUB
1050 ELSE ON YD(CD)GOSUB 347
0,3480,3490,3500,3510,3520,3
530,3540,3550 !162
3450 FOR I=1 TO 24 :: READ A
$ :: NUM$(CD,I)=SEG$(A$,5,20
):: NEXT I :: RETURN !213
3460 RESTORE 1050 :: RETURN
!ZERO !072
3470 RESTORE 1300 :: RETURN
!ONE !229
3480 RESTORE 1550 :: RETURN
!TWO !248
3490 RESTORE 1680 :: RETURN
!THREE !248
3500 RESTORE 1930 :: RETURN
!FOUR !183
3510 RESTORE 2180 :: RETURN
!FIVE !160
3520 RESTORE 2430 :: RETURN
!SIX !101
3530 RESTORE 2680 :: RETURN
!SEVEN !237
3540 RESTORE 2930 :: RETURN
!EIGHT !216
3550 RESTORE 3180 :: RETURN
!NINE !140
3560 END !139
```

CAUG on web

Harry Hoffman is publishing the Cleveland Area TI99/4A newsletter on his web site (http://members.stratos.net/harryhoffy/newsletter/). The Cleveland area TI users are following in the footsteps of the Milwaukee Area User Group. Gene Hitz started publishing the group's newsletter on its web site in January. The MAUG newsletter can be seen at http://members.tripod.com/~genehitz/maug.html.

Getting more out of Funnelweb with a Horizon RAMdisk

BY JACQUES GROSLOUIS

The following article appeared in Classic 99, the newsletter of the Hoosier User Group.—Ed

This article provides suggestions for Tlers running FunnelWeb 4.40 from a Horizon RAMdisk and an Extended BASIC module. An edit screen within the configuration program CFG allows selection of nine programs which may be CALLed.

The first of these choices is the program which will be run when the HRD is accessed or booted.The program MENU is often placed in this position probably because it comes packaged with the HRD and it will directly run most types of XB and memory image programs and directly supports direct access to the nine CALLs set up when the HRD is installed.

One requirement is that these nine CALLs must be on the first disk configured under the HRD setup and their names must not be longer than four characters. A strong case can be made that the ML option of Funnel-Web 4.40 is a better choice than MENU. ML allows a total of 48 menu items split between two screens compared to MENU's eight selections on each of three screens.

Any of the ML menu items can also access a UL menu containing eight items. This is a useful way to group similar programs. For example, a selection of game programs could be grouped on one UL menu which would be accessed from one item on the ML menu.

A major drawback of ML is its inability to run XBASIC programs. This can be overcome by using MENU as a ML menu item and configuning MENU to contain mainly XB programs. ML can also be used to run single object files or a group of such files by using the script load (SL) feature of FWB. This is not possible with MENU

With a bit of thought the nine CALLs allowed with the HRD can be very useful. These can be short XB programs and can run other CALLs or other XB or assembly programs. DELETE "Call name" will run your CALL from an XB program. You can also run an assembly or BASIC program which is on another RAMdisk by using LD.x.name, where "x" is the other drive number and "name" is the filename of your program.

Although the ML option of FWB will not directly run an XB program, the HRD will boot up to a CALL which can be an XB program In addition, resetting your TI, while depressing the shift key, will access the title screen for the XB module. From XB command mode you can access any of the nine CALLs by

entering CALL "callname." Not all of the programs accessed by these CALLs need be on the program disk because your CALL can include a line which runs a program that is on another RAMdisk.

What it can do

Many Tiers when asked to recite the features of their TI which distinguishes it from other computers will often state its ability to use sprites and speech but then will have to go searching for programs which contain these features. The following Extended BASIC program, which requires a speech synthesizer, attempts to remedy this and to provide examples of the points made in the previous paragraphs.

If set as your first CALL from your HRD using a name such as "REM," this program will show a welcome screen, speak a message, and then run a preselected program such as Remind. Thereafter, whenever the HRD recycles, it will display a screen saver type screen and will speak one of five messages chosen at random when a key is pressed before accessing the ML menu.

Holding down "M" before the sprites start will take you directly to the ML menu and holding the space bar will take you to the TI-Writer side of FWB as described in the FWB documentation. Of course, users may change the CALL SAYs to suit their preferences. The screen saver routines starting in line 420 are by Tom Jakabfy and were published in MICROpendium. The sprite patterns are in lines 640 and 650 and can be changed by the user.

REM

```
100 ! SAVE DSK5.REM !027
110 CALL CLEAR !209
120 CALL SCREEN(5)!150
130 IF RND<.7 THEN 400 !083
140 GOTO 160 :: I,B,N,R,C,SX
,SY,Q,X,Y,N=0 :: CALL SPRITE
 :: CALL CHAR :: CALL MAGNIF
Y :: CALL POSITION :: CALL D
ELSPRITE !001
150 A$,B$="" :: CALL PATTERN
 !015
160 CALL INIT !157
170 CALL KEY(3,K,S):: IF K=3
2 OR K=77 THEN 400 !001
180 CALL PEEK(2,A)!234
190 IF A=0 THEN 270 !004
200 CALL LOAD(2,0)!157
210 PRINT "  Welcome to my T
I Computer": : : : : : : : :
!028
220 FOR I=0 TO 14 :: CALL CO
LOR(I,16,1):: NEXT I !127
230 CALL SAY("I+UNDERSTAND+T
HE1+Y+TWO+K+PROBLEM.DO+YOU")
!158
240 !@P !019
250 CALL KEY(3,K,S):: IF K=3
2 OR K=77 THEN 400 !001
260 DELETE "LD.7.REMIND" !07
7
270 GOSUB 420 !245
280 RANDOMIZE Y !238
290 ON INT(RND*5)+1 GOTO 300
```

Continued on page 40

Continued from page 39

```
,320,340,360,380 !018
300 CALL SAY("WELL DONE")!01
2
310 GOTO 390 !214
320 CALL SAY("LET+GET+TO+WOR
K")!196
330 GOTO 390 !214
340 CALL SAY("ARE+YOU #READY
 TO START#")!237
350 GOTO 390 !214
360 CALL SAY("WHAT+DO+I+DO+N
EXT")!040
370 GOTO 390 !214
380 CALL SAY("DID+YOU+HEAR+T
HE1+ONE+ABOUT+THE1 #TEXASINS
TRUMENTS# HOME+COMPUTER")!11
7
390 CALL DELSPRITE(ALL)!115
400 DELETE "FW" !255
410 STOP !152
420 B=15 :: N=8 :: R=96 :: C
=104 :: SX,SY=10 !226
430 READ A$ :: READ B$ :: CA
LL CHAR(96,A$):: CALL CHAR(1
00,B$)!045
440 Q=0 :: CALL MAGNIFY(1)::
 CALL SPRITE(#1,100,16,R,C,S
X,SY)!059
450 CALL SPRITE(#2,100,7,R+B
,CB,SX,SY)!102
460 CALL SPRITE(#3,100,6,RB,
C,SX,SY)!099
470 CALL SPRITE(#4,100,14,R,
C+B,SX,SY)!085
480 CALL SPRITE(#5,100,2,RB,
CB,SX,0)!240
490 CALL SPRITE(#6,100,8,R+B
,C,0,SY)!119
500 CALL SPRITE(#7,100,4,R,C
+B,0,SY)!116
510 CALL SPRITE(#8,100,10,RB
,C+B,SX,0)!228
520 GOTO 550 !119
530 GOSUB 660 !230
540 GOTO 520 !089
550 CALL POSITION(#1,Y,X)!09
3
560 IF (Q=0)AND(Y>100)THEN G
OSUB 620 :: GOSUB 600 :: CAL
L MAGNIFY(3):: Q=1 :: GOTO 5
30 !002
570 IF Y>130 THEN CALL MAGNI
FY(4)!141
580 IF Y>210 THEN CALL DELSP
RITE(ALL):: GOTO 440 !168
590 GOTO 530 !099
600 FOR I=1 TO N :: CALL PAT
TERN(#I,96):: NEXT I !152
610 RETURN !136
620 FOR X=1 TO 20 :: NEXT X
:: RETURN !243
630 !@P+ !062
640 DATA 1C1C1C1E1F1FFCFC7F3
F07070303010000C0C000FC3E0F0
F3F3F3E1C98F0C0C0 !246
650 DATA 68607FE377151C0C !2
39
660 CALL KEY(3,K,S):: IF K>0
 THEN 280 ELSE RETURN !099
```

Other features, such as background music, graphics, or setting up your printer or some other device, could be added to this program.

FWB AND PCs

If your TI is connected to a PC by means of a serial cable from your RS232 card, you can configure your printer name under FWB to be as ENTER. In order to send your file to your printer, space out the dash after PIO. And, to send the file to your PC, delete PIO. This saves having to remember the RS232 settings.

To receive a file into Text Editor, or TI-Writer, you must save the file (SF) to RS232.BA=600. However, there is no practical way of saving this setting in advance.

Bruce Harrison's AMS Transfer program is useful and can instead by used to download a text file. It will run without an AMS card. You can also store text file names for use by a variety of other programs.

If you want to include the same feature in an XBASIC program, merge the following program into your XBASIC program. You can call it from your program by using CALL MAILBX("TARGET_FILE").

MAILBX

```
26400 SUB MAILBX(A$)!254
26405 CALL PEEK(8198,A,B)::
IF A=170 AND B=85 THEN 26410
 ELSE CALL INIT !012
26410 FOR Z=1 TO LEN(A$)!246
26415 Y=ASC(SEG$(A$,Z,1))::
CALLLOAD(24577+Z,Y)!236
26420 NEXT Z :: Y=32 !184
26425 FOR Z=LEN(A$)+1 TO 80
:: CALLLOAD(24577+Z,Y):: NEX
T Z !131
26430 SUBEND !168
```

This merge file may cause problems if you are using Brad Snyder's 40-Column Utilities because his program uses the memory area >A000 to >A007 for GPLLNK and DSRLNK routines and this area survives the loading of another program. A fix for this is to turn off the clock before using MAILBX.

Arnold demonstrates CROM

Oliver Arnold of Germany demonstrated C99 using Winfried Winkler's C compiler and Arnold's defined CROM at the TI fair in Nottingham, England, last October. The CROM is installed "resistant" in extra RAM space (>6000).

Standard libraries include CSUP80, CFIO, TCIO, FPRINTF, ACCEPT, TICK, PRINTF, SPRINTF, MOVE, FLOAT, BEEP, TEXT80, and STRING. According to Arnold, the advantage of this resistant library is that it speeds up the development process and makes use of a full 40K of CPU RAM.

Readers who are interested in learning more about the CROM package may contact Arnold at oliver.arnold@gedos.de.

Other developments reported by Arnold include a TI 16-bit logic analyzer, Risk (the strategy game for the TI), HPManager (a printer program for Hewlett Packard printers), Teletext hardware and software (Menu 80 and Teletext scripter loader), and MDC3 version 1.5 (MultiDisk Commander for SCSI and disk access).

MYBASIC Version 4, DDI-ICON, TIPS Paint, Grabber, MYSIDEPRINTer, MYARK, MYFX, MYGolf, and MYWheel

BY CHARLES GOOD

All the software I review this month is for the Geneve only. Jim Uzzell, doing business as DDI software, sent me a whole bunch of his Geneve specific software to review. Apparently Jim is the current world authority on the Geneve's Advanced Basic, because much of his software is written in this language. I have tested Jim's software using MY-Basic (also known as "Advanced Basic") v3, which is freeware and available from me for the asking. Jim also offers MY-Basic v4, described below, with several enhancements over v3.

Some DDI software I am reviewing this month is commercial, requires payment in advance, and can only be obtained directly from DDI. Other software I am reviewing this month is shareware and can be freely distributed. Payment is required if the user finds the software useful. And some of this month's reviewed software is freeware with no payment required, although donations are appreciated. All the software is easy to use and none requires a mouse. If you send me $1, I will mail you a DSSD disk with all the Geneve shareware and freeware described here in archived format. Please note that software I send you will have no hard copy documentation. If you want the hard copy docs that come with all of Jim's shareware programs, you need to order the shareware directly from Jim and send him his requested shareware fee. Commercial software reviewed here can only be obtained directly from Jim Uzzell.

MYBASIC VERSION 4

DDI software offers MY-Basic v4 with several enhancements over v3.

The hard copy manual is about 80 pages of additions and corrections to the original Advanced Basic manual that came with new Geneve computers. Pages are three-hole punched and paginated so that you can just slip the pages into your existing manual, sometimes discarding some of the original Myarc documentation in the process. New commands are described and there are numerous corrections to the original documentation. The new appendix is quite informative, listing the MY-Basic memory map, RS232 memory map when using MY-Basic, as well as floppy and MFM hard drive disk structures. The information in this manual is well worth the asking price to Geneve users.

MY-Basic v4 seems to be fully compatible with programs written for Myarc's Advanced Basic v3. Jim has already created a version of his MYMenu2+ menuing/appointment calendar/word processor/spreadsheet/hard drive backup software which takes advantage of some of the additional features built into MY-Basic v4.

MY-Basic v4 is commercial. The hard copy addendum and a TI disk with the needed files cost $15 postpaid from DDI software.

DDI-ICON

This program lets you take icons from PCs and use them in MY-Basic programs. You can paint and edit these icons from within the DDI-ICON. They are saved as six-sector ASCII D/V80 files. Lots of icons are available in the PC world, and the hard copy documentation that comes with DDI-ICON tells you how to get them into the correct format for transfer to a TI disk. It is helpful if you have either PC Transfer, a direct cable connection, or maybe PC99 to get PC icons onto a TI disk. On the PC, icons are first converted to ASCII format with a PC shareware program before transferring the PC files to a TI disk. If you don't want to bother with this conversion then you can use the hundreds of already converted icons that come free with DDI-ICON, enough to fill 5 SSSD disks.

When the software loads you are prompted for an icon file to load. If you want you can type "null" and create your own icon-sized graphic de novo. You see on screen a small version of the icon as it would appear in a Geneve program, about the same size as an icon on a PC screen. You also see a much larger version of the icon with each icon pixel greatly enlarged. This large version may be split into two windows because it is so large. It is the large version that you work on. On screen prompts tell you about all the active keys.

Arrow keys move the cursor and the "c" key paints the selected color on the current pixel. You can select from 16 colors of the displayed color palette, changing colors at will from one pixel to the next. A color chart is provided that shows you how to emulate PC colors by putting different colored pixels next to each other so that they blend. This pixel by pixel drawing and changing of colors takes a long time. The results can be excellent.

A small MY-Basic program is included that lets you view icons on screen . You are encouraged to use this code in your own MY-Basic software to display your own colored icons.

This program makes lots of small images from the PC world available to the Geneve programmer, including the several hundred converted icons that come with the software package. DDI-ICON is commercial and costs $30.

TIPS PAINT

This program allows you to

Continued on page 44

Continued from page 43

import TIPS graphics into MY-Basic programs. You can also create a new TIPS size image. Images can be edited and colored pixel by pixel to produce stunning results.

When you load TIPS Paint you are asked if you want to load an already painted TIPS Paint image or a TIPS image. You can select none by typing "null" and then you can begin drawing and coloring your own image de novo. There are hundreds of tips images available on public domain TI disks. Each TIPS graphic file contains many images. If you want to load one of these images, put the disk with an image file in a drive or select a hard drive path name. SCSI and HFDC hard drive paths are supported. The program will tell you the names of the TIPS graphic files in the selected directory and optionally will let you print the names of each graphic in the file on screen or to your printer. Finally you are asked for the name of the specific TIPS graphic within the graphic file you wish to load.

When you load a TIPS graphic to the Geneve's screen for the first time you may be amazed at the detail present. What you see is highly detailed with single Geneve pixel resolution and smoothly curved lines. These are not the blocky TIPS pictures folks using the TIPS graphic software on a 99/4A are used to seeing on a 99/4A screen and or printed by printers.

You see two images on screen — one natural size and one greatly enlarged. The natural size image is quite a bit larger than a natural size icon image, about one-third of the 512-pixel-wide screen width. The enlarged image is where you draw and color. Changes made are immediately shown in the natural size image. You move the cursor with arrow keys and color the current pixel with the "c" key. To erase select the background color as your current color.

You are given an initial color palette of 16 colors but these can be changed to any of the 256 possible Geneve colors. Each TIPS Paint image can have up to 16 different colors. The large edit image is only partially shown on screen because of its size. You have six viewing windows for the complete image and only one of these windows is visible at a time. However, you can see what you are doing because the entire image is always displayed on screen in its natural size. Images are saved to a 21-sector ASCII D/V80 file.

A small MY-Basic program is included that will display TIPS Paint images on screen, and users are encouraged to modify this code and use it in their own MY-Basic software.

A second MY-Basic program, TIPS Print, lets you print TIPS Paint images on a printer in any of eight sizes. I don't know if this works with color printers to give a color image, but it might. It works nicely on my 9-pin Epson-compatible printer. TIPS

Print first loads the color picture onto the screen. Then you are asked what size image you want and do you want it aligned left, right, or centered. Each of the 16 colors is represented by a slightly different tiny graphic shape which is downloaded into the printer's download graphic buffer. This will probably work with most 9- and 24-pin printers except the Gemini 10x.

This program makes hundreds of images from the 99/4A world available to the MY-Basic programmer,and the images look much better than they do on a 99/4A system. TIPS Paint is commercial and sells for $30. The package includes hard copy documentation, several nicely colored images, as well as the above described printing, display, and editing software.

GRABBER

This package lets you design or edit a full or partial screen of any size up to the Geneve's maximum resolution, capture the screen, and then display the screen in your own MY-Basic program. Putting the screen in a MY-Basic program is as simple as adding a couple of lines of BASIC code copied from the documentation. You can also incorporate the screen in any MDOS program if you are an MDOS programmer. Most of the menus and title screens in this as well as other DDI software were created with the full screen editor that is part of the Grabber package.

To capture a screen, first design the screen in MY-Basic. The package includes an alternative character set and the ability to define your own character set which can be used with CALL CHAR to define your custom screen. With your screen displayed, figure out the starting and ending rows and columns you want to save and make a note of the graphics mode you are in. Add this line of code to the MY-Basic code that made your screen immediately after the screen is displayed: RUN "DSK.GRABBER.GRABBER2" Then run the MY-Basic program "config" and enter the size and graphic mode of the screen you are going to save. You also enter whether you want to save the screen as assembly source code or object code. To load the screen into a MY-Basic program choose object code. Finally, run your MY-Basic program and your screen will be saved to a disk file. It is so simple! And it is just as simple to bring this screen back up anywhere you want in any MY-Basic program.

If you set up config to save the screen as assembly source code then the saved screen file can be used in an assembly MDOS program that you write and assemble. No support for doing this is provided in the Grabber package.

The package also includes a full-screen editor that you can use instead of DISPLAY AT, HCHAR, and VCHAR within MY-Basic. You can design a 24 row x 40- or 80-column screen in a manner similar to the icon

Continued on page 46

Continued from page 45

and TIPS Paint editors. You move the cursor around and leave behind blocks of any combination of 16 colors. You can load in a previously designed screen for further editing and you can type text anywhere you want. You can also load in an ASCII text file and make it part of your screen. You have available a special character set with IBM style graphics to create boxes, etc. The results of this editor can be saved as object code and later CALL LOADed and CALL LINKed from a disk or hard drive anywhere into any MY-Basic program.

The package also includes a character pattern editor that you can use to redefine most ASCII characters pixel by pixel and save as an alternate character set. Some sample redefined character sets are included. These redefined characters can be used by the full screen picture editor.

There are some limitations to the screens that can be saved. They have to be comparable to DISPLAY AT, CALL HCHAR, and CALL VCHAR. Graphic commands that are part of MY-Basic such as CALL DRAW and CALL RECTANGLE don't get saved. In spite of this limitation, the GRABBER 2 package is a marvelously useful tool for programming in the Geneve's very powerful MY-Basic.

GRABBER 2 is commercial and sells for $30.

MYSIDEPRINTer

This will do something no other TI or Geneve software can do. It takes a single D/V80 file and prints it as a pamphlet, sideways on 8.5- x 11-inch paper using both sides of the paper. The program can handle up to 96 half paper pages front and back, four "pages" per paper sheet. When the papers are stacked and then creased in the middle and optionally stapled at this crease, the resulting book reads correctly. You can even have the pages numbered. Hard copy documentation for DDI software ordered directly from Jim Uzzell is printed this way and looks really good.

There are some limitations to the D/V80 source file. It must be no more than 40 characters per line and have no control characters. The file must fit certain easily understandable mathematical criteria for a total number of lines. If there aren't enough lines you have to add some blank lines within or at the end of the file. This is easy to do with any TI word processor. If you have a file that has lines longer than 40 characters you can process the file with an included MAKE40 utility to change your original to a D/V80 file with no more than 40 characters per line.

When you run MYSIDEPRINTer it first asks for a path name. Hard drive paths are supported. Then it asks for a file name. If the indicated file does not meet length and width specifications you are told this and the program stops. If the file is OK, then sideways printing begins. When one side of the paper is printed you are asked to remove the paper and put it back in the printer for the other side to be printed. The printer's out-of-paper warning is disabled and the program determines left and right margins. You should list the program because there are some useful features noted in REM statements. For example, removing the REM from one line allows double strike printing for a better looking final product.

Separate programs are included for 9-pin and 24-pin printers.

MYSIDEPRINTer is shareware. The requested donation is $15, and well worth it in my opinion.. If you want to try before you buy, I can mail this to you. See above.

MYARK

This is an MDOS program that is functionally equivalent to and fully compatible with Barry Boone's Archiver program for the 99/4A. The main menu lists Archive Files, Extract Files, Catalog Disk, Catalog ArcFile, Copy File, List Text File, and Exit.

In all cases hard disk paths are acceptable for source and destination. When unpacking files you cannot unpack to a MFM hard drive beyond the first level subdirectory. You can unpack to WDS1.TEMP. but not to WDS1.TEMP.ARCFILES. Unfortunately, MYARK does not allow me to unpack a file to a subdirectory on my SCSI hard drive. Also you cannot include a subdirectory structure within an archive. If you archive all the files in a directory, any subdirectories of that directory will not be included in the archive.

You get a nice 80-column, 24-line display. This means that you can see lots of file names on screen at once when cataloging a disk or archived file that contains many files. These file lists can be sent to your printer.

I really like this program. Since it is an MDOS program I can have it immediately available from the custom start-up menu that appears when I turn on my Geneve. The advantages to the Geneve user of this particular archiver program are the 80-column display and the ability to use path names. Every Geneve user should have this.

MYARK is shareware. A $15 payment to the author is requested. If you want to try before you buy I can mail it to you. See above.

MYFX

This software helps you to transfer files between the Geneve and a PC that are cabled together. No modem is needed. MYFX is written in MDOS and can be run directly from the MDOS command line.

To cable the two computers together I purchased an 8 foot "modem cable" at my local Wal-Mart to connect my Geneve's RS232 to my laptop PC's COM1 port. This cable works just fine with no modifications needed. MYFX can upload and download to and from a Geneve's real floppy drives, Horizon RAMdisks, internal RAMdisk, and hard drives controlled by the HFDC card where

Continued on page 48

Continued from page 47

MYFX can directly access subdirectories. SCSI hard drives are not recognized by the current version 3 of MYFX.

MYFX is designed to allow you to upload and download D/F128 or I/F128 files with TIFILES headers to and from a PC. It will not handle text files directly. TIFILES is a standard way of storing TI files on PC hard drives, FTP sites, and computer bulletin boards. PC files with this header can be converted so that they can be used by the two common TI emulators. They can also be transferred from PC disks to TI disks using the commercial software product PC-Transfer.

D/F128 or I/F128 files are usually created with Archiver on the 99/4A or Geneve ,or with MYARK on the Geneve. Archive a bunch of files and then upload them to a PC. MYFX automatically adds a TIFILES header to the uploaded file. From there you can post these files to Internet FTP sites or computer bulletin boards for other TI users to download.

Downloading is just as easy. First download files with TIFILES headers from one of the TI CD-ROM disks available commercially, or an Internet FTP site or a computer bulletin board onto your PC's hard drive. Then move the files over to the Geneve using MYFX, which automatically strips off the TIFILES header. Then use an archiver program like MYARK to unpack the files on the Geneve. The whole process is not difficult. To check everything out I have archived files on the Geneve, sent the resulting archive file over to my PC with MYFX, downloaded the file archive back from the PC to the Geneve with MYFX, and unpacked the files onto Geneve disks.

Before you load MYFX on the Geneve you need to cable the Geneve to a PC and run a terminal program. The MYFX documentation gives specific instructions for configuring the Windows 3.1 terminal program, although I think other commercial and shareware PC terminal programs will probably also work. I have not been able to figure out how to get the Windows 95/98 Hyperterminal program to work with direct cable transfers, so for moving files from my Geneve with MYFX I use the recommended Windows 3.1 terminal program (file name term.exe) on systems running Windows 95. If you don't have the Windows 3.1 terminal program, send me $1 and I will mail it to you on a PC disk to run on your Windows 95/98 PC.

After you have your PC terminal program running you boot MYFX. To do this you need a specific autoexec configuration on the Geneve, and the easiest way to do the autoexec thing and boot MYFX at the same time is to type "&ldmyfx3" at the MDOS prompt to run the autoexec substitute file ldmyfx3 on the MYFX disk from the current drive. After you answer the prompt for baud rate on the Geneve's keyboard, all other typing is done on the PC keyboard.

MYFX makes the PC a host and the Geneve a client. You specify both PC and Geneve file names and locations from the PC keyboard. Even though you are working on the PC keyboard you need to remember that when the software says "upload" this means sending files from Geneve to the PC, and the software's "download" means sending from the PC to the Geneve.

The MYFX package includes a bunch of utilities that can be used for preprocessing D/V80 text files and for converting D/V80 to and from D/V128 format. None of this is necessary because there is a much easier way to transfer text files between a 99/4A or Geneve and a PC that does not require the use of MYFX. Just connect the computers with the above described "modem cable", load Funnelweb or other TI-Writer editor, and then load or save files using RS232 as the file name. An article describing this technique is elsewhere in this issue of MICROpendium.

MYFX is shareware with a requested $15 registration fee. I can mail it to you for evaluation.

MYGOLF

With respect to computer games, likes and dislikes are a very personal thing and can easily differ from one individual to another. Some people like to sit down and play continuously for long periods of time. Others just want to take a quick break from whatever they are doing and do a little quick gaming. Both groups of gamers will find what they crave in MYGOLF. You can play just a few holes, which takes very little time, or you can play an entire 18-hole course, which takes much longer. For me, this is one of the most entertaining TI computer games I have encountered in quite some time. A similar 99/4A program is Burkingolf which I reviewed several years ago. Burkingolf has a very bad case of the "slows" because it is written in TI BASIC. MYGOLF is much faster and more usable.

The game, written in MY-Basic, is very easy to control, which is one of the features I like. All you need to push are the four arrow keys and the "Q" key. Key response is very quick. Joystick use is optional. When you first run the program you are asked 1 or 2 players, amateur or pro players, amateur or pro course, and back 9 or all 18 holes. The pro course is more complicated than the amateur course. But let me tell you, the amateur course isn't easy. I haven't yet figured out what the difference is between an amateur and pro player.

After all the above data are entered the game begins. You see a color map of your hole on screen complete with fairway, rough, and green. The resolution of this image is comparable to bit map on a 99/4A. Obstacles include sand bunkers, ponds, streams, and trees. Your man is at the tee. Press the left/right arrows to point your man the way you want. Press the up arrow to cycle through the possible clubs. Your driver is the default club

Continued on page 50

Continued from page 49

at the tee. When you are on the green the putter is your default club. You also have various irons and woods available.

To swing press and hold "Q." The longer you hold the "Q" key down the more powerful your swing, up to a point. Your ball advances and your man is automatically positioned next to the ball in its new lie. As the ball flies you may hear it splash if it goes in the water. If the ball heads toward a group of trees it will ricochet off of several trees before coming to rest. When you finally hole your ball you hear a reassuring plunk.

The bottom of the screen usually has a message about the ball. 'Poor lie in the rough,' 'good lie in the grass,' and 'ball in fairway sand' are some of these messages. You may also be told 'grass too high for driver' if you try to use the driver on a ball in the rough. Then you have to select a different club. After each hole the bottom message tells you hole in one, double-eagle, eagle, birdie, par, bogey, double-bogey, or triple-bogey. If your score for the hole is really really bad the bottom message will use the player's name and suggest 'more lessons for xxxx,' 'another caddy for xxxx,' or 'new irons for xxxx.' I frequently earn one of these golf bimbo messages and I find them amusing.

The one feature I can complain about is lining up for a shot. The screen shows a straight line, representing a golfer, next to the ball. You point the line with the arrow keys in the direction you want to hit. When you press "Q" the line turns into the outline of a golfer. The problem is that the straight line has no arrow, so that if you don't think about what you are doing you can hit the ball backwards, 180 degrees from the intended direction. The thing to remember is that golfers are right-handed and line themselves up on the left side of the ball.

This game is freeware. I can mail it to you. See above.

MYWHEEL

This is an older 99/4A 'Wheel of Fortune' game slightly reworked for MY-Basic. One to several players are accommodated. As the ever-changing money window, equivalent to the wheel, flashes on screen a player presses Enter to stop the window. If the window says "0" then the player loses all accumulated earnings. Otherwise the player accumulates the money shown in the window. If you don't want to solve the puzzle you get the opportunity to buy a vowel or guess a letter.

Word categories include football, baseball, movies, people, phrase, presidents, states, and television. These categories change randomly each time a new puzzle appears. Although not obvious when running the game, if you examine the BASIC code you will find an entry point that lets you edit the puzzle words stored in on disk in each of these categories.

The only thing missing is a beautiful Vanna letter turning person. There is a letter turning person, but it certainly would not be described as gorgeous.

MYWHEEL is freeware. If you want me to send it to you see above.

The following DDI software games are freeware. Some are modifications of previously existing TI BASIC or Extended BASIC games. In each case the keyboard response is more precise and the game runs faster compared to the 99/4A version in part because the game is being played on the much faster Geneve. All these games are included on the shareware and fairware DDI software disk I will send to Geneve owners who send me $1.

MYAHTZEE is a modified 1984 Extended BASIC version of Yahtzee. There can be 1-4 players. It isn't as visually interesting as the TI Yahtzee command module, but it is functionally the same. The computer automatically calculates the score based on a player's choices. Messages congratulate you when you reach advance and expert status and there are other congratulatory messages.. There are also a variety of discouraging messages if you are not doing so well such as "that straight's pretty crooked!".

MYBANDIT is a slot machine game with a few extra features. You can insert as many coins as you want before pulling the handle. Of the three images that pop into the slot machine's window you can choose to (H)old onto any one or two of them when you pull the handle again after depositing more money. This feature resembles poker, where you can keep some cards and discard others prior to another draw. When you get tired of playing, or are out of money, you can end the game and display a final summary. This summary includes amount of money put in, money won, maximum cash in hand during the game, net gain, number of holds, and number of turns.

MYBINGO. This is very similar to the Bruce Harrison bingo game I reviewed in the January/February 1999 issue of MICROpendium. The main menu is: 1-automatic game mode, 2-manual game mode, 3-print bingo cards, 4-print call list, and 5-exit. Bingo card printing works on any printer and prints 1 card per sheet of paper. Automatic mode calls a new number every few seconds. Manual mode calls a new number each time the Enter key is pressed. Called numbers continue to be displayed on screen in numerical order until the current game ends.

Screen graphics are more interesting than Harrison's bingo program, but the Harrison program does something useful that MYBINGO doesn't. It talks. Harrison's program,

Continued on page 52

MICROREVIEWS

Continued from page 51

when used on a 99/4A but not on a Geneve, speaks the letter number combinations as they are being called. There is code in MYBINGO for speech, probably left over from when the program was written to operate on a 99/4A. Unfortunately there is no provision for speech in MY-Basic v3, even if you have a Rave speech card in your Geneve.

MYGAMMON is, of course, Backgammon. This is for one player against the computer. You have to know the rules in advance because there are none in the game. The computer randomly rolls the dice and automatically keeps track of the score.

MYPOKER is draw poker for one player. On-line instructions are available. You discard or hold cards by pressing the D and H keys. Unlike some versions of poker, you are allowed to discard all five of your original cards if you want. Also unlike many versions of poker, you can if you win try for double or nothing by playing a draw of high card/low card. You need at least a pair to break even. Cards are shuffled randomly and not reused in a hand. The cards are reshuffled for each new hand. I particularly enjoy this game for one of the same reasons that I like MYGOLF, namely that you can play it quickly. Load it up, play a few hands of poker, then get back to what you were doing before play. The graphics and sounds are well done.

MYSAM is a surface to air missile game. You shoot at aliens. On-line instructions are included. The aliens fly by at different speeds and heights. You have 19 missiles and you press the space bar to fire the next missile. The aliens sometimes fire ray guns to destroy some of your unused missiles. This game is very easy to play, but isn't really all that great.

MYSTRIKE is otherwise known as Video Bowling. It accommodates 1-4 players. There are on-line instructions. Rules are the same as real bowling. As the ball moves left/right along the foul line you press a key to release the ball. You can specify right hook, left hook, or straight. All players' scores are displayed on screen in a very realistic looking bowling score card. The key to high scores in this game is learning how to use hook shots.

ACCESS

Jim Uzzell (the only source for commercial DDI software); 2600 Lowndes Dr.; Valdosta GA 31602; phone 912-242-4355

Charles Good (Send me $1 and I will mail you a TI a disk of all the shareware and fairware reviewed here, or I can e-mail the disk for free in PC99 format. For another $1 I can send you the Windows 3.1 terminal program on a PC disk.); P.O. Box 647; Venedocia OH 45894; phone 419-667-3131; e-mail good.6@osu.edu

USER NOTES

Y2K Problem in Calendar Maker

This comes from Bruce Harrison.

More changes made to Calendar Maker. What bothered me was that if they'd made a mistake about the leap year, there would have to have been something else in there to compensate in years 2001 and beyond.

Sure enough, there was an IF-

Continued on page 54

USER NOTES

Continued from page 53

THEN in line 970, to wit :: IF Y>2000 THEN 990 (at the end of that program line.)

Once I'd corrected the other mistake (in line 910) this IF-THEN would cause a problem in 2001 and every year thereafter. To prove this, I printed a one month calendar for December 2000, then another one month for January 2001. As expected, with that IF-THEN still in line 970, December 2000 ends on a Sunday and January 2001 starts that same Sunday. WRONG!

Took out the :: IF Y>2000 THEN 990, saved the program and ran it again for January 2001. This time January 2001 started on Monday, as it should, so the calendars should be accurate from there on.

To summarize this far, we changed line 910 from:

```
IF  (Y=2000)+(Y=1600)<0  THEN
940
```

to read:

```
IF  (Y=1600)<0  THEN  940
```

Then in line 970, removed the last statement in that line, to wit:

```
::  IF  Y>2000  THEN  990
```

These changes take care of the years 2000 and beyond. All are in the XB program CREATE on the set's Program disk.

The other changes that I've made are to convert the program from working with 9-pin printers over to 24-pin. Two programs need changing for that:

In the program CM991, line 820, replace:

```
CHR$(27);"A";  CHR$(8);
```

with

```
CHR$(27);"  +";CHR$(40);
```

For some reason ESC "A" 8 was there twice, in that same line, so one occurrence can be eliminated. The same needs to be done in CM992, at line 570, where again there were two ESC "A" 8 strings in succession.

Have a subscription problem?

Didn't receive a disk?

You can call us at 512-255-1512
between 9 a.m. and noon Saturdays CT;
or fax us anytime at 512-255-1512;
or e-mail us at
micropendium@yahoo.com.

USER NOTES

TIC-TOC ends newsletter

The Rocky Mountain 99ers Computer Club is undergoing reorganization, according to Bob Grossart, editor of the club's newsletter, TIC-TOC. Grossart says the newsletter has suspended publication as of the November/December issue.

C99 suggestion

The following appeared in Tidbits, the newsletter of the Mid-South 99ers. It was written by Martin Zeddies.

If you are using Clint Pulley's c99/MDOS compiler, be sure that the TIMODE of MDOS is switched off before you try to compile a source code file of your own C program.

The problem begins if TIMODE is active and the compiler will find an #include command in your source. It is probable that this will happen.

In this case the C compiler will display a message that it can not find the INCLUDE-FILE if you specify the path as "DSKn." within the #include line. But when you check your system the path seems to be okay.

The problem occurs because MDOS will be "thinking" in TIMODE. If you try to restart your autoexec batch file with the ampersand (warm start of the Geneve), the system will hang. All you can do is reboot the system.

My advice is to copy your source file to a backup disk after the C compiler displays the error message and reboot your system to switch off TIMODE correctly.

CLASSIFIEDS

FOR SALE

TI99/4A hardware, software, books, articles and MICROpendium back issues. Send list and best offer to Reitman, 1 Corsa St., Dix Hills, NY 11746.

Classified ads are 10 cents per word. Payment (checks, money order, or credit card) must accompany ad. Mail to: MICROpendium Classifieds, P.O. Box 1343, Round Rock, TX 78680.

DISKS/BACK ISSUES

❑ Back Issues,$3.50 each to March 1996, later $6 each. List issues on separate sheet.

No price breaks on sets of back issues. Free shipping USA. Add $1, single issues to Canada/Mexico. Other foreign shipping 75 cents single issue surface, $2.80 airmail. Write for foreign shipping on multiple copies.

OUT OF STOCK: V1#1-2; V2#1

GENEVE PUBLIC DOMAIN DISKS
(SSSD unless specified)

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	$7	$5
❑ Series 2	$9	$7	$5
❑ Series 3	$9	$7	$5
❑ Series 4	$9	$7	$5
❑ Series 5	$9	$7	$5
❑ Series 6	$9	$7	$5