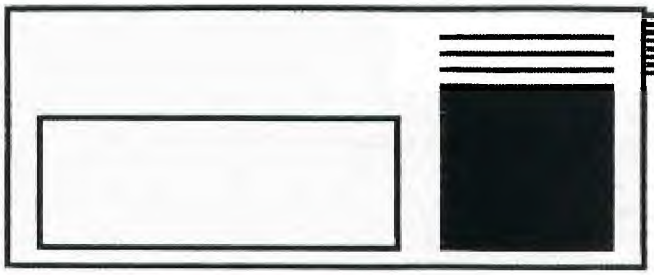
(019) 8705
Delware



THE DATA BUS

ISSUE :
VOL. 5
NO. 4
- - - -
M A Y
1 9 8 7

THE DELAWARE VALLEY USERS GROUP

DEDICATED TO THE TI AND COMPATIBLE HOME COMPUTER FAMILY

P.O. BOX 6240 STANTON BRANCH, WILMINGTON DE 19804

TIBBS
by TIBBS Committee

The BBS has been rather slow this past month. I guess with the weather turning warm everyone is going out in the yard to plant and clean. I thought this month I would print the list of programs presently available on the New Castle Tibbs to see if we could generate more activity on the board by more of our members. Many of you have been assigned safe user numbers but have not called for 6 months to a year. Do you have all the programs you want and all of your questions have been answered? Call the board, either in New Jersey, New Castle, or Dover.

LIST OF DOWNLOADS
NEW CASTLE TIBBS

NAME — DESCRIPTION

ARCHIVER2 — 19 SECTORS
(Assembly version of ARCHIVER.)
(Runs from Opt5 E/A.)

BALESHIP — 35 SECTORS
(Battleship game XB)

OUTLINE — 29 SECTORS
(Aid in making outlines XB)

2-PASS — 10 SECTORS
(2 pass copy program run from)
(Extended BASIC)

RLEXB — 42 SECTORS
(Max Rle that loads from XB.)
(Loads very fast.)

TIGER2 — 33 SECTORS
(Picture that can be used with)
(TI-Artist or with RLEXB.)

RAGLNK — 223 SECTORS
(Use ARCHIVER to restore to)
(program form.)
(Converts D/F80 files to Image)
(format. Print Doc's before using.)

MGR1 — 33 SECTORS
(Patch for version 3.5. Corrects)
(bug in the sector 0 format.)

ARCHIVER v2.11 — 32 SECTORS
(Will compress several programs into)
(one file. After transfering, file can)
(be separated into original programs.)
(Runs in XB.)

FAST/TERM — 74 SECTORS
(Use Archiver to restore. This)
(is a modified version of 1.14)
(worth looking at. It has been)
(set up to work with a ram disk)
(as well as a greatly improved)
(disk catalogue.)

DISKU v3.5 — 197 SECTORS
(Version 3.5 of J. Birdwell's sector)
(editor. Use Archiver to restore.)

CONTINUED ON PG 10

Minutes of the Delmarva Chapter - 11 May 1987

Chapter President Chuck Bower acknowledged the efforts of those responsible for the fabrication of the portable equipment unit that was discussed in detail in last month's minutes. A very neat and complete package, it will contain the clubs computer equipment, to be used at the monthly meetings. A printer and a RS-232 card are all that is needed to complete the unit. Anyone having a spare of either and would like to donate or loan it to the club, please contact one of the officers of the group. Thank You!

Jack Shattuck of the DVUG's Christiana Group was a guest at the meeting. He gave a good talk about the Group's software library. Due to other pressing business on his part, it will be necessary to identify a new "keeper" of this important asset of the Group. Mr. Shattuck went on to discuss the pitfalls surrounding the misuse of proprietary/copyright materials and the emphasis that the Group places on acknowledging the legal ramifications thereof.

It was agreed by the membership that users of the local TIBBS would contribute two dollars ($2.00) per month as a user fee. There are presently eight members who will subscribe. This will defray the expenses of the TIBBS phoneline, as was approved several months ago. The dues will begin on 1 June 1987.

CONTINUED ON PG 9

DVUG EXECUTIVE COMMITTEE MEMBERS IN 1987

PRESIDENT .....TOM AUGUST
VICE PRESIDENT .....JIM DAVIS
SECRETARY .....TIM EVERS
TREASURER .....TOM KLEIN
SGT. AT ARMS .....JIM FOLZ
DELMARVA CHAPTER CHR .....CHARLES BOWER
SO.JERSEY CHAPTER CHR .....TONY DIFEBBO
SHORE CHAPTER CHR .....HARVEY ADAMS

NORMAL MEETING SCHEDULE

CHRISTIANA GROUP	4th Thursday	6:30-9:30
DELMARVA CHAPTER	2nd Monday	7:00-9:00
SOUTH JERSEY CHAPTER	3rd Monday	6:45-9:00
SHORE CHAPTER	1st Thursday	7:30-9:00

MEETING PLACES

CHRISTIANA GROUP: Delaware's Christiana Mall on Rte. 7, at I-95 Exit 4-S. We meet in the Community Room. Enter between J. C. Penney and Liberty Travel inside the Mall.

DELMARVA CHAPTER: Kent County Courthouse, Basement Conference Rm #25, Green & State Sts, Dover, De. Use the Green St. side entrance.

SOUTH JERSEY CHAPTER: Deptford Municipal Bldg, Cooper Ave. and Delsea Drive, (Rtes. 534 & 47), in Gloucester County. Enter and park in rear of the building.

SHORE CHAPTER: Scullville Firehouse #1, County Rte. 559 (on left, between mile markers 4 and 3), in Atlantic County. Ignore Station #2 on right enroute.

DVUG BULLETIN BOARDS

(302)322-3999	Anytime
(609)429-7792	Monday-Thursday 3:00 PM-7:00 AM Friday 3:00 PM-Monday 7:00 AM
(302)674-1449	6:00 PM-6:00 AM

For general information, you may contact

TOM KLEIN	Pa.	(215)494-1372
JIM FOLZ	Del.	(302)995-6848
BUTCH FISHER	N.J.	(609)783-8276
GUS LEWIS	N.J.	(609)927-5601

Delaware Valley Users Group membership includes: library and software privileges, monthly DATABUS newsletter, plus other special benefits. Annual membership rates are: Family or Individual $15; Student $10; Newsletter only (beyond 75 mi) $10.

TRANSMIT YOUR NEWSLETTER COPY TO: The Data Bus Editor --- Jim Folz, Telephone (302)995-6848, or use the DVUG mailing address shown on Page One. PLEASE SUBMIT NEWSLETTER ARTICLES FOR AN ISSUE BEFORE THE 2ND THURSDAY OF EACH MONTH.

DVUG ADVERTISING RATES FOR THE DATA BUS:

1/4 page - $ 5/issue, or $ 45/12 issues
1/2 page - $ 8/issue, or $ 75/12 issues
Full page - $15/issue, or $125/12 issues

NOISE on The Data Bus
by Jim Folz

As I write this, the formalities of adding a third chapter are being executed. In anticipation of this addition, I have updated the general information section (Page 2). Please review this information for meeting schedules, directions, etc. At this time, the new chapter does not have a bulletin board.

Welcome, then, to:

President Harvey Adams
Vice-President Brady Moore
Secretary Maurice Trembley
Treasurer Randy Reeves

and the members of the Shore Chapter.

Now, on to the news ...

In order to promote the learning/use of XBASIC and BASIC, the Executive Board proposes a contest as follows:

* Language - XBASIC or BASIC
* Program Length - 5 numbered lines (maximum)
* The contest will run in two-month periods for as long as there is interest.
* The first winner will be announced at the July meeting of the Christiana Group.
* The contest is open to all chapters.
* A subscription to Micropendium is offered as the prize.

The Executive Board is looking for a volunteer to coordinate the contest. Please contact one of the officers if interested.

Based on the rate that the raffle tickets are selling for the Rave keyboard, it looks like the drawing will occur at the June meeting. Please have all ticket stubs and unsold tickets returned to Tom Klein before the June meeting of the Christiana Group.

The positions of Contest Coordinator, Software Librarian, Refreshment Coordinator, Equipment Coordinator, and Cashier are open. Please notify one of the officers if you can help out in one of these areas.

ADVANCE NOTICE: At the June meeting of the Christiana Group, Bill McLean will discuss Multiplan. Be sure to attend this one!

Much Thanks to Jim Peterson for the disks full of articles, programs and tips mentioned in Jack's article (Page 3). Look for these gems to appear in future issues.

CONTENTS OF THE MAY ISSUE OF THE DATA BUS:

AXIOM, MULTIPLAN and FUNNLWRITER
by Grant Nichols
(From CONNI, Spirit of 99, Sept, 1986 Issue)

This article is written for those TI users, who do not own or use an RS232 with their particular setup, but who in fact use an AXIOM or other parallel interface unit with their printer.

For those of you who use Extended Basic, Funnelwriter, or Multiplan, the AXIOM unit will perform very satisfactorily, provided you do not let the line length exceed 80 characters in length using the standard font. However, what happens when you attempt to use compressed print either in the standard form or italics?

It took me some time before I finally stumbled upon the answer. First of all, when using Multiplan, the line lengths would not exceed 61 characters which was good for about seven or eight columns, then the remaining columns on that line were printed below. The end product was one fine mess with everything jumbled together.

To end this misery, I finally found the solution, as follows. First, let's address the printer, and pay very close attention to the first line.

```
10 OPEN #1:"PIO.LL-132",VARI
ABLE 132
20 PRINT #1:CHR$(15);CHR$(27
):"G"
30 CLOSE #1
40 END
```

This little program will enable the printer to print in condensed, double strike, for a line 132 characters in length. For those users using Multiplan, a further modification must be made. When in the PRINTER MARGIN segment, there are no further modifications; however, in the PRINTER OPTIONS, where it says SETUP: insert the following: PIO.LL-132. Now you can print in Multiplan, with either normal compressed, normal compressed double strike, italics compressed, or italics compressed double strike. The only thing that needs to be changed is Line 20 above for the four different print styles.

If you're not using the AXIOM interface but rather the AXIOM GP-550A Printer, you might consult the OVUG Newsletter Library for the October, 1986, CIN-DAY NEWS, which contains a two-page program called "PRINTit", by Jim Susco for convenient set-up of that system.

HELPFUL LOADING TIP FROM AMNION SOFTWARE (Which was reprinted in the K-TOWN 99'er in June, 1986)

Always load and run your public domain disks through EXTENDED BASIC. Most BASIC programs will run in Extended Basic and will be much faster. If you get an error message, usually "BAD VALUE IN XXX", you will know that it is console Basic ONLY. It is wise to MAKE A BACKUP COPY OF THE DISK B E F O R E YOU RUN PROGRAMS, THEN L I S T THEM before running them, especially when using a printer to see if the default settings match OR to see if they require a Speech Synthesizer and console Basic. Remember - all files that show "PROGRAM" in a catalog of the disk are NOT necessarily Basic or XBasic programs. An error message #50 shows that these are memory image files loaded with E/A, Option 5 or will need a special "loader" program to load them either through XB or Assembly. Also, if a Basic or XBasic program is larger than 48 sectors, you may have to do a CALL FILES (1 or 2) before you can load it.

OLD TIGERCUBS NEVER DIE ...

TIGERCUB SOFTWARE, 156 Collingwood Ave., Columbus, OH 43213, (614) 235-3545, previously announced Nuts Bolts Disk #3, with 140 program merge routines, bringing the total for all 3 to an amazing 348 in number.

Jim Peterson has now reduced the price for any of those three to $15 EACH, pre-paid.

Four Tips from the Tigercub disks, and the 18 Tigercub Collection disks, are reduced to $10 each. His 130 individual programs are $2 each, plus $1.50 per order, disk OR TAPE (minimum $10 to an order), with tape orders available only as long as his supply lasts.

Catalogs are $1 each, deductible from your first order.

Everyone says they miss his Tips (which Jim wrote as a sample to promote his Nuts and Bolts) so he's collected a LAST batch of miscellaneous stuff to create Tips No. 42,43,44, and 45. Along with his lessons he uses to teach others program techniques in XBasic, he emerged with another 3 full SS/SD disks. Users groups editors that have been sending him newsletters can request a free set. They're available at $5 each to others.

IF YOU ORDER, please mention that you read this in THE DATA BUS (the Delaware Valley Users' Group).

User Groups trying to contact other can get a list of other groups from the Tigercub for the cost of a diskette (or send yours) and postage.

UTILITY PROGRAMS by Tom Freeman

As published in TOPICS, LA 99er Newsletter, are now available for $9. You get a disk, plus documentation, for Quad Column, Variable Column Lister, Print Sideways (also in A/L version), a CALL LOAD to A/L, A/L to CALL LOAD, Keyboard Map, Sector Checker, XBasic Checksums and XBasic Tokens.

Write to LA 99er Computer Group, P. O. Box 3547, Gardena, CA 90247-7247.

ASGARD SOFTWARE Telephone Number Changed;
FONTWRITER II Still On Hold ...

Asgard Software, P.O. Box 10306, Rockville, MD 20850, has a new telephone listing: (301)559-2429. You can reach them by voice between 9 - 5 Monday through Friday, except when personnel are at lunch from Noon to 2 p.m. Then you get their recording machine.

Asgard markets FONTWRITER by J.Peter Hoddie which allows a true merge of text and graphics, from both TI-Artist and Character Sets Graphic Designs. That version has been on sale a while. FONTWRITER II has been promised since February, with some upgrading as well as allowing use of a NEC 8032A or C.Itoh Prowriter printer.

The delay is from author Hoddie, not Asgard which refrained from advertising it until March, when Hoddie promised he'd have it for April. At the TICOFF in North Jersey, they had a special price, expecting it momentarily. Although Asgard took some orders and checks, it is their policy NOT to cash checks for any orders until they are actually sending the material.

Hoddie, who has been completely absorbed in the attempt to get Myarc's GENEVE on the street, now swears that it will be ready for shipping on Memorial Day. Asgard's Beta version ran fine.

If you call Asgard to inquire, ask for some other product information; they carry numerous interesting and current TI-99/4A products.

EXPLORE YOUR COMPUTER'S DEPTHS LINK UP TO

HOME NETWORK

In the past if you wanted a library of educational software you had but one choice - spend hundreds of dollars.

But now there's a NEW ALTERNATIVE - HOME NETWORK.

Home Network is your gateway to thousands of hours of educational courseware, as well as games, electronic mail and bulletin boards.

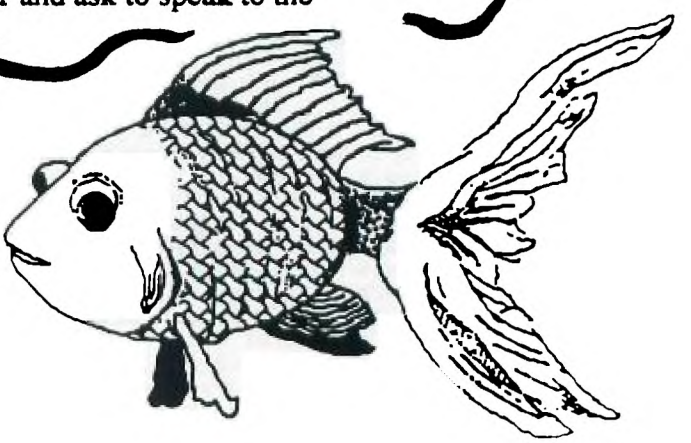
With Home Network educational courseware, you can study astronomy tonight and French tomorrow.

Via Home Network's electronic communication features you can send messages to fellow members or engage in lively bulletin board conversations on topics ranging from microcomputers to current events.

With Home Network games, you can match wits with the computer or play against other subscribers.

The Home Network is part of the University of Delaware's PLATO system which was previously available only to students. Now through the availability of microcomputers this system can be yours for a fraction of the cost of similar services.

Interested? Call (302) 451-8161 and ask to speak to the Home Network Representative.



BASIC/XBASIC PROGRAMMING TECHNIQUES
by Jack Shattuck

MUSIC, MUSIC, MUSIC - Enhancing and than Dabugging Your Programe

Adding sound affects to BASIC programs is a not-too-difficult way to enrich simple displays. The CALL SOUND (DURATION, FREQUENCY, VOLUME) - hanceforth, D,F,V - command provides information for musical notes as well as some action noises. Let's concentrate on the former for a moment.

You could intersperse repeated CALL SOUND statements with display commands (such as PRINT or DISPLAY AT text lines). That's easy to debug while creating, but is rather memory intensive:

```
100 CALL SOUND(500,392,5)::
CALL SOUND(125,330,5)::CALL
SOUND(500,262,5)::CALL SOUND
(500,330,5)::CALL SOUND(500,
392,5)

110 CALL SOUND(1000,523,5)::
PRINT "O say can you see"
```

(We'll use the multiple-statement XBasic lines to save room here; the idea is the same in BASIC.)

Another approach is equivalent to a GOSUB routine, by reading in DATA statement values for a single CALL SOUND command, e.g.,

```
190 FOR NOTE=1 TO 6

200 READ D,F,V

210 CALL SOUND(D,F,V)

220 NEXT NOTE

230 PRINT "O say can you see
"

1000 DATA 500,392,5,125,330,
5,500,262,5,500,330,5,500,39
2,5,1000,523,5
```

It may not look it now, but memory saved by the DATA read method is significant over a long musical score.

When listening for the right sounds as you check out your program, those notes play rather quickly. You could slow it down, in order to analyze it easier, by temporarily doubling the tone duration, such as

```
210 CALL SOUND(D*2,F,V)
```

Another debugging method is to leave that CALL SOUND(D,F,V) line alone, but add another,

```
215 PRINT D;F;V
```

or else

```
215 DISPLAY AT(24,1)ERASE AL
L:D;F;V
```

which always can be deleted later. While it shows the last note played, the PRINT routine also acts as a natural slow-down.

If that disrupts a graphic display during debugging, you could use these lines instead:

```
90 OPEN #1:"PIO"

215 PRINT #1:D;F;V
```

to go to a printer. Buffer delay may confuse you in comparison to the note sound, so you could add a pause,

```
216 FOR WAIT=1 TO 1000::NEXT
 WAIT
```

That's possible, but awkward and expensive (in terms of use of paper).

Some music scores have only a single melody line. You needn't be a composing genius to add a harmony or harmonic accompaniment. Your CALL SOUND command permits multiple notes (using the same duration); try this variation for the above example. Instead of

```
210 CALL SOUND(D,F,V)
```

use:

```
210 CALL SOUND(D,F,V,F*1.01,
V,F*1.02,V)
```

I'm indebted to Jim Peterson for helping me retrieve this formulation. I'd seen it briefly, once or twice, about 1 1/2 years ago, and never wrote it down. The Tigercub immediately knew what I wanted when I called him for assistance.

The plain and this expanded version of Line 210 even can be combined in one program. Reserve harmony for a Chorus, but use a single melodic verse. Just specify which note you want to use to start the chorus (for example, note #50):

```
195 N=1

200 READ D,F,V :: IF N>49 TH
EN 310

210 CALL SOUND(D,F,V)::N=N+1
::GOTO 200

310 CALL SOUND(D,F,V,F*1.01,
V,F*1.02,V)::N=N+1::GOTO 200
```

You can go back and forth as well, with an appropriate IF (Note number)/Then (Call Sound) command. It adds emphasis or breaks up monotony for what would otherwise be an unassuming series of single-beat notes.

I am requesting your assistance in finding another individual willing to undertake the task of handling the Christiana meeting's Software Library. Although I've been able to do some mailing and copying for you (easier since I have two double-sided double-density drives) and put together a few Disks-of-the-Month, I still have not had time to catalog files in an updated list - the most important ongoing need of members.

I'd also like a little time free instead of being responsible for hauling my system, meeting after meeting. My 9-year old son now has a disk drive; I'd like to have time to help him now and do some much neglected programming again myself. I'll continue to write columns for THE DATA BUS, but I need some relief on the other items. Tell your officers, call TIBBS, or show up with hands raised to volunteer, please. Thanx! - Jack

Programs That Write Programs - Part 1

by Jim Peterson

Way back in 1982, in the old 99'er Magazine, Vol. 1 Nos. 3 and 4, John Clulow wrote two articles entitled "How To Write a Basic Program That Writes Basic Programs". At that time I thought I would never understand what he was writing about!

But really, it's simple. Have you ever LISTed a program to the disk, and noticed that the resulting D/V80 file took up many more sectors than the program itself? That is because the TI saves programs in a compacted form, with each statement represented by a single token ASCII.

When a program is saved in MERGE format, by SAVE DSKX(filename),MERGE it is saved in this same compacted form, but in a D/V 163 file. And, of course, a D/V file can be created by a program - so you can write a program which will create a D/V 163 file in the form of a program, and then MERGE that file into memory and RUN it as a program, and SAVE it as a program.

You ask, why use this roundabout way of writing a program? Why not just key it in directly? Well, for one thing you can write program lines that could not possibly be keyed in directly. As for instance, the famous "line zero". Key this in, run it with a disk in drive 1, then enter MERGE DSK1.ZERO and LIST the result.

```
100 M$="BETCHA CAN'T DELETE
THIS!"
110 OPEN #1:"DSK1.ZERO",VARI
ABLE 163,OUTPUT :: PRINT #1:
CHR$(0)&CHR$(0)&CHR$(131)&CH
R$(200)&CHR$(LEN(M$))&M$&CHR
$(0)
120 PRINT #1:CHR$(255)&CHR$(
255) :: CLOSE #1 :: END
```

Actually, there is an easy way to delete that line - but no way to key it in directly.

Here's another one - the full ASCII string.

```
100 OPEN #1:"DSK1.FULLSTRING
",VARIABLE 163,OUTPUT
110 LN=100 :: GOSUB 190 :: A
$=LS&"M$"&CHR$(190)
120 FOR J=1 TO 127 :: CS=CS&
CHR$(J):: NEXT J :: AS=AS&CH
R$(199)&CHR$(127)&CS&CHR$(0)
130 PRINT #1:A$
140 GOSUB 190 :: BS=LS&"M2$"
&CHR$(190)
150 FOR J=128 TO 255 :: DS=D
$&CHR$(J):: NEXT J :: BS=BS&
CHR$(199)&CHR$(128)&DS&CHR$(
0)
160 PRINT #1:BS
170 GOSUB 190 :: FS=LS&"M$"&
CHR$(190)&"M$"&CHR$(184)&"M2
$"&CHR$(0)
180 PRINT #1:FS :: PRINT #1:
CHR$(255)&CHR$(255):: CLOSE
#1 :: END
190 LS=CHR$(INT(LN/256))&CHR
$(LN-256*INT(LN/256)):: LN=L
N+10 :: RETURN
```

Run that, then enter NEW, then MERGE DSK1.FULLSTRING. The string contains every ASCII from 0 to 255 in sequence, and there is no way to enter many of the unprintable ASCII codes from the keyboard. You can of course create that string in a program - FOR J=0 TO 255 :: M$=M$&CHR$(J) :: NEXT J - but it saves a few seconds to have it predefined.

The full ASCII string is very useful for a quick shuffle without duplication. For instance, to scramble the numbers 200-250,

```
100 M$="
                !""#$%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJK
LMNOPQRSTUVWXYZ[\] _`abcdefg
hijklmnopqrstuvwxyz{|}~"
110 M2$="




                                "
120 M$=M$&M2$
130 M$=SEG$(M$,200,50)
140 L=LEN(M$):: RANDOMIZE ::
 X=INT(L*RND+1):: N=ASC(SEG$
(M$,X,1)):: M$=SEG$(M$,1,X-1
)&SEG$(M$,X+1,255)
150 PRINT N;:: IF LEN(M$)=0
THEN STOP ELSE 140
```

One more example - can you run this program and get these results? You won't even be able to key in that last line!

```
>LIST
100 FOR J=1 TO 7 :: READ M$
:: PRINT M$ :: NEXT J
30000 DATA AAAAAAAAAAAAAAAA
AAAAAAAAAA,BBBBBBBBBBBB,BB
BBBBBBBBB,CCCCCCCCCCCC,
                DDDDDDDDDDDDDD
30010 DATA "TESTING",,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,""TEST
ING""
>RUN
AAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBB,BBBBBBBBBBBBB
CCCCCCCCCCCC
                  DDDDDDDDDDDDD
"TESTING"
,,,,,,,,,,,,,,,,,,,,,,,,,,
""TESTING""

*READY*
```

Next month - the answer to that puzzle, and a more useful program that writes a program, and then we will start learning how you too can write programs that write programs!

Plato

by John Kelley

Thanks for all the support at our demonstration at the University of Delaware last month. Seven people signed up on the System before we left that night. I hope no one is having any problems with the software or signing on the System. If you are, call Tibbs and leave a message to me, Barry, or Paul and we will try to help you out. Rae won't be able to answer questions about the software as they have never used it. I would like to hear comments from our new Plato users as to how they like the System and what they have found.

Programs That Write Programs - Part 2
by Jim Peterson

Last month I promised you something more useful, so here it is. This routine will come in very handy for formatting screen text into neat 28-column lines, and will save the text in program lines of DATA statements. When you are ready to save, type @@@ and enter as the last line, then NEW and MERGE DSK1.LINEFILE.

```
100 !LINEWRITER to aid in fo
rmatting screen text into 28
-column format and saving it
 as DATA program lines in ME
RGE format - by Jim Peterson
110 !strings containing comm
as and quotation marks will
be ACCEPTed, and converted t
o DATA statements which RUN
correctly even though they
120 !are not enclosed in quo
tation marks!
130 CALL CLEAR :: OPEN #1:"D
SK1.LINEFILE",VARIABLE 163 :
: LN=30000
140 FOR R=1 TO 24 :: DISPLAY
 AT(R,1)SIZE(1):" " :: ACCEP
T AT(R,0)SIZE(-28):A$ :: IF
A$="@@@" THEN 180 :: B$=B$&C
HR$(200)&CHR$(LEN(A$))&A$
150 X=X+1 :: IF X/4=INT(X/4)
THEN 160 ELSE B$=B$&CHR$(179
):: GOTO 170
160 GOSUB 210 :: LN=LN+10
170 NEXT R :: X=0 :: CALL CL
EAR :: GOTO 140
180 IF B$="" THEN 200 :: IF
SEG$(B$,LEN(B$),1)=CHR$(179)
THEN B$=SEG$(B$,1,LEN(B$)-1)
190 GOSUB 210
200 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1 :: END
210 PRINT #1:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&B$&CHR$(0):: B$=NU
L$ :: RETURN
```

Oh - that puzzle in last month's article? Try creating those DATA statements with this LINEWRITER program!

Now, let's get down to business and learn how to do all this. First, let's write a program that will write a program to list the token codes that you need to use to write a program that will write a program.

```
100 OPEN #1:"DSK1.TOKENLIST"
,DISPLAY ,VARIABLE 163,OUTPU
T :: FOR N=129 TO 254 :: L1=
INT(N/256):: L2=N-256*L1
110 PRINT #1:CHR$(L1)&CHR$(L
2)&CHR$(131)&CHR$(N)&CHR$(0)
:: NEXT N
120 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1 :: END
```

Key that in, RUN it, then enter NEW, then MERGE DSK1.TOKENLIST. Now LIST it and you will see a list of ASCII codes 129 through 254 and their token meanings. Delete lines 171 through 175, 185, 198, 226 through 231, and 242. Change the definition of 199 to QUOTED STRING, of 200 to UNQUOTED STRING, and 201 to LINE NUMBER, and add line 255 !END OF FILE.

You don't need all those exclamation points, so change the program to a DIS/VAR 80 file by LIST "DSK1.TOKENLIST". Then key in this little routine.

```
100 OPEN #1:"DSK1.TOKENLIST"
,INPUT :: OPEN #2:"PIO" !or
whatever
110 PRINT #2:CHR$(27);"N";CH
R$(6)
120 LINPUT #1:A$ :: PRINT #2
:TAB(10);SEG$(A$,1,4)&SEG$(A
$,6,255):: IF EOF(1)<>1 THEN
 120 ELSE CLOSE #1 :: END
```

RUN it, and print out a list of all the token codes. Keep it handy, you'll be needing it. Notice that every Extended Basic statement has its own ASCII token code - even the ones you perhaps never heard of, such as LET and GO. Notice also that every keyboard symbol which affects program execution, such as + and -, has its own ASCII token code which is NOT the same as its keyboard ASCII code. And notice that the double colon, used as a separator in Extended Basic multi-statement lines, has its own token.

Now, let's take a look at how a MERGE format program is put together. This routine will do that for you - and you will also find it very useful in debugging the MERGE programs you are going to write.

```
100 DISPLAY AT(3,5)ERASE ALL
:"D/V 163 FILE READER": :"
    by Jim Peterson": : :" T
o edit a file saved or":"cre
ated in MERGE format."
110 DISPLAY AT(12,1):"Output
 to? (S/P)S":" (S)creen":" (
P)rinter" :: ACCEPT AT(12,17
)SIZE(-1)VALIDATE("SP"):Q$
120 IF Q$="P" THEN DISPLAY A
T(14,1):"PRINTER? PIO" :: AC
CEPT AT(14,10)SIZE(-18):P$ :
: O=2 :: OPEN #2:P$
130 DATA ELSE,"::",!,IF,GO,G
OTO,GOSUB,RETURN,DEF,DIM,END
,FOR,LET,BREAK,UNBREAK,TRACE
140 DATA UNTRACE,INPUT,DATA,
RESTORE,RANDOMIZE,NEXT,READ,
STOP,DELETE,REM,ON,PRINT,CAL
L
150 DATA OPTION,OPEN,CLOSE,S
UB,DISPLAY,IMAGE,ACCEPT,ERRO
R,WARNING,SUBEXIT,SUBEND,RUN
,LINPUT
160 DATA ,,,,,,THEN,TO,STEP,"
","","";","":",),(,&,,OR,AND,XOR
,NOT,=,<,>,+,-,*,/, ,
170 DATA QUOTED STRING,UNQUO
TED STRING,LINE NUMBER,EOF,A
BS,ATN,COS,EXP,INT,LOG,SGN,S
IN
180 DATA SQR,TAN,LEN,CHR$,RN
D,SEG$,POS,VAL,STR$,ASC,PI,R
EC,MAX,MIN,RPT$,,,,,,,NUMERI
C,DIGIT
190 DATA VALPHA,SIZE,ALL,USI
NG,BEEP,ERASE,AT,BASE,,VARIA
BLE,RELATIVE,INTERNAL,SEQUEN
TIAL,OUTPUT,UPDATE,APPEND
200 DATA FIXED,PERMANENT,TAB
,#,VALIDATE
210 DIM T$(126):: FOR J=1 TO
 126 :: READ T$(J):: NEXT J
:: E$(1)="LINE NOT CLOSED WI
TH CHR$(0)"
220 DISPLAY AT(16,1):"FILENA
```

```
ME? DSK" :: ACCEPT AT(16,14)
:F$
230 ON ERROR 240 :: OPEN #1:
"DSK"&F$,VARIABLE 163,INPUT
:: GOTO 250
240 DISPLAY AT(20,1):"I/O ER
ROR" :: ON ERROR STOP :: RET
URN 220
250 ON ERROR 260 :: LINPUT #
1:A$ :: X=ASC(SEG$(A$,1,1)):
: Y=ASC(SEG$(A$,2,1)):: IF X
=255 AND Y=255 THEN 410 ELSE
270
260 PRINT #0:"FILE NOT CLOSE
D PROPERLY":"WITH CHR$(255),
CHR$(255) ?" :: STOP
270 PRINT #0:"LINE NUMBER":X
;"TIMES 256=";256*X:Y;"PLUS"
;Y;"=";256*X+Y
280 FOR J=3 TO LEN(A$)-1 ::
X=ASC(SEG$(A$,J,1))
290 IF X=201 THEN PRINT #0:X
;"LINE NUMBER" :: X=ASC(SEG$
(A$,J+1,1)):: Y=ASC(SEG$(A$,
J+2,1)):: J=J+2 :: PRINT #0:
X;"TIMES 256=";256*X:Y;"PLUS
";Y;"=";256*X+Y
300 IF X=199 THEN PRINT #0:X
;"QUOTED STRING" ELSE IF X=2
00 THEN PRINT #0:X;"UNQUOTED
 STRING" ELSE GOTO 360
310 J=J+1 :: X=ASC(SEG$(A$,J
,1)):: PRINT #0:X;"OF";X;"CH
ARACTERS"
320 ON ERROR 340 :: FOR L=1
TO X :: Y=ASC(SEG$(A$,J+L,1)
):: PRINT #0:Y;CHR$(Y):: IF
Y<32 OR Y>126 THEN PRINT #0:
"UNPRINTABLE CHAR - ERROR?"
330 NEXT L :: J=J+X :: GOTO
370
340 PRINT #0:"ERROR! INSUFFI
CIENT BYTES IN":"STRING" ::
IF ASC(SEG$(A$,LEN(A$),1))<>
0 THEN PRINT #0:ES(1)
350 ON ERROR STOP :: RETURN
250
360 IF X<129 THEN PRINT #0:X
;CHR$(X);" VARIABLE NAME" EL
SE PRINT #0:X;T$(X-128)
370 CALL KEY(0,K,S):: IF S=0
 THEN 390
380 CALL KEY(0,K2,S2):: IF S
2<1 THEN 380
390 NEXT J :: IF ASC(SEG$(A$
,J,1))=0 THEN PRINT #0:"0 EN
D OF LINE" ELSE PRINT #0:ES(
1)
400 GOTO 250
410 PRINT #0:X:X;"END OF FIL
E" :: CLOSE #1 :: STOP
```

Next month - how to do it!

FOR SALE

P-Box w/32K, BRAND NEW (never used) RS232, TI disk controller, and one SS disk drive. $350 or best offer. Contact John Kelley, (302)328-6059, 5 Holly Drive, Oak Run, New Castle, DE 19720 or TIBBS.

Sprites - Part 1

by Jim Peterson

The sprites of TI Extended Basic are mostly used in fast-action arcade-type games, but they have other uses as well.

Up to 28 sprites can be placed on the screen at one time, but there is one very serious limitation - if more than 4 of them are in a line horizontally, only the 4 lowest-numbered ones will be visible. That is why, if you have numerous sprites moving about the screen, one of them will occasionally disappear and reappear, or a horizontal slice of a magnified sprite will become transparent.

A sprite is placed on the screen by the statement:

CALL SPRITE(#N,ASC,COL,DOTROW,DOTCOL)

N is the sprite number, between 1 and 28, and it must be preceded by the # sign. ASC is the ASCII code of the character that you wish the sprite to have. It must be between 32 and 143 - the ASCII characters 33 through 126 are the keyboard characters, the others will be blank unless you redefine them. COL is the color you wish the sprite to have, using the same color codes, 1 to 16, as are used for CALL SCREEN or CALL COLOR.

DOTROW and DOTCOLUMN are the dot row and dot column at which you wish the sprite to appear. You know that the monitor screen consists of 24 rows and 32 columns. Using HCHAR or VCHAR, you can place a character on any one of those 768 spaces (PRINT and DISPLAY start at column 3 of the graphics screen). Each of those spaces consists of a grid of 8 x 8 dots, totaling 64. By turning various of those dots off (blank) or on (colored), a character is displayed on the screen. Therefore, the screen is 8 x 32 or 256 dotcolumns wide and the visible screen is 8 x 24 or 192 dotrows deep. Actually, dotrow can be anything up to 256; dotrows 193 through 256 are hidden below the bottom of the screen, and sprites can be hidden there.

The upper left hand corner of your sprite will be at whatever dotrow and dotcolumn you specify.

To convert a graphics screen (HCHAR) position into dotrow and dotcolumn, use DOTROW=8*ROW-7 and DOTCOL=8*COL-7; to convert a PRINT/DISPLAY position, you must use DOTCOL=8*(COL+2)-7.

So, CALL SPRITE(#1,42,16,89,121) will place sprite #1, in the form of the asterisk (ASCII 42), colored white (16) in the middle of the screen. If you want, you can give it motion when you create it, by giving it a row-velocity and a column-velocity. These velocities can be from -128 to 127. A positive row velocity moves the sprite down, negative moves it up; a positive column velocity moves it right, negative moves it left.

Velocity 0 is a standstill, and speed increases from 1 upwards and from -1 downwards.

So, CALL SPRITE(#1,42,16,89,121,5,5) will place that white asterisk in the middle of the screen and start it moving slowly at a 45 degree angle downward to right (since the values 5 and 5 are positive and equal). It will continue moving at that direction and speed until you tell it to do otherwise, all by itself and without program control. When it reaches the right edge of the screen, it will "wrap around" and appear at the left. When it reaches the bottom, it will disappear briefly while it

passes through those hidden dotrows, and "wrap around" to appear at the top.

If you want to change the pattern of the sprite, there are three ways to do so. You can CALL SPRITE again with the same sprite number but a different ASCII character - but if the existing sprite is not in the position of the dotrow and dotcolumn you specify, it will disappear and reappear in the new position. Or you can reidentify a character by CALL CHAR, and any sprite having that character will change accordingly, without affecting its color, position or movement. Or you can use CALL PATTERN(#N,ASC) to change the pattern of sprite #N to the pattern of the specified ASCII character, without affecting color, position or motion.

There are also two ways to change the color of a sprite. CALL SPRITE with the same sprite number and ASCII but a different color code will recreate the sprite with the new color, but in whatever position is specified. CALL COLOR(#N,COLOR) will recolor sprite #N to the specified color code without affecting its pattern, position or motion.

If you want to change the position of a sprite, CALL LOCATE(#N,DOTROW,DOTCOL) will make it disappear at its old location and appear at the new location. The pattern and color will be unchanged, and if it was in motion the same motion will continue from the new position.

To change the motion of a moving sprite, or to start a stationary sprite into motion or vice versa, use CALL MOTION(#N,RV,CV) - RV and CV being the same row velocity and column velocity optionally used in CALL SPRITE. CALL MAGNIFY will change the size of your sprite. You do not specify a sprite number with this CALL, because it affects all sprites that are on the screen or are subsequently placed on the screen. CALL MAGNIFY(2) enlarges the sprite 4 times so that it fills 4 of the graphic screen spaces, 256 dot spaces. CALL MAGNIFY(3) causes the sprite to consist of 4 characters, occupying 4 graphic screen positions. The upper left of these characters will be the ASCII specified in the CALL SPRITE or CALL PATTERN, provided that the ASCII is evenly divisible by 4 - otherwise, it will be the next smaller ASCII evenly divisible by 4. The next higher ASCII will be in lower left, the next in upper right, the next in lower right. In other words, if you use CALL MAGNIFY(3) and CALL SPRITE(#1,64,2,10,10) you will get a sprite looking like this - @B
AC
and if you CALL SPRITE(#1,65,2,10,10) you will get exactly the same thing, because the computer will substitute the next lower number, 64, which is evenly divisible by 4.

Naturally, you will not have much use for sprites consisting of four characters, unless you redefine them into a single pattern, and in that case you must remember that they will appear in that upper left/lower left/upper right/lower right sequence. Fortunately, there are sprite editor programs to take care of this for you.

CALL MAGNIFY(4) will enlarge that 4-character sprite so that it fills 16 graphic screen positions. Note that magnification options 2 and 4 actually enlarge each dot to fill 4 dot positions, so that the sprites have a more angular, blocky appearance.

And finally, CALL MAGNIFY(1) will return magnified sprites to their normal single-space size.

Programming with sprite motion is unlike any other programming, because you do not control the program execution step-by-step. When you set a sprite in motion, it continues in motion while the program goes on to do whatever it is supposed to do next. When you want to control the sprite again, you must catch up with it and find out where it is. There are three ways to do this.

CALL COINC(ALL,C) will give a value of -1 to C if any two sprites on the screen are overlapping, even slightly, or 0 if they are not. CALL COINC(#1,#2,TOL,C) will give C a value of -1 if the upper left hand corners of sprites #1 and #2 are within TOL dotrows and dotcolumns of each other. TOL may be any number you want, depending on whether you want to catch them only when they are right on top of each other, or just getting close. If not within tolerance, C will equal 0.

CALL COINC(#1,DOTROW,DOTCOL,TOL,C) will give C a value of -1 if the upper left corner of sprite #1 is within TOL dotrows and dotcolumns of the specified DOTROW and DOTCOL.

CALL COINC is not foolproof. If you give the sprites a fast motion, a coincidence may not be caught. And when you alternate your CALL COINC with other statements such as CALL JOYST, a coincidence will be missed if the program is executing some other statement at the time.

CALL POSITION(#N,DOTROW,DOTCOL) will give the dotrow and dotcolumn that the upper left corner of the sprite is occupying at the instant it is called. This one again is not foolproof because the sprite will have moved from that position before another statement can be executed to do anything with the information.

CALL DISTANCE(#1,#2,D) or CALL DISTANCE(#1,DOTROW,DOTCOL,D) will give to D a value depending on the distance between the two spritas, or between the sprite and the location. The value, as I understand it, is the square root of the total of the squares of the difference between the dotrows added to the squares of the differences between the dot columns. I'm not sure how useful all that is, and I have rarely seen this CALL used by programmers.

Finally CALL DELSPRITE(#N) will delete sprite #1 from the screen and CALL DELSPRITE(ALL) will delete them all.

Those are just the basics of sprite programming. What can be done depends solely on your ingenuity.

MINUTES... FROM PG 1

George August won the 50/50 drawing in the amount of seven dollars ($7.00).

Jim Gentry gave a brief but detailed presentation on the University of DE's "PLATO" Program. The program has exceptional possibilities. (1200 BAUD Modem is required.)

Jim England completed the meeting with another training/Q&A session on TI BASIC. An unusually high degree of member's interest was exhibited on this.

There were fourteen people in attendance at this meeting.

Respectfully submitted,
Robert Edwards, Secretary

(Thanks to the Delmarva Chapter for the update. Regretably, the April minutes were received too late for publication in last month's newsletter. Ed.)

TIBBS FROM PG 1

CHARA1 42 SECTORS
(Modified CHARA1 file for use with)
(TI-WRITER. Use ARCHIVER to restore.)

CALENDER 40 SECTORS
(Memory image file to print a calendar)
(by the year or month. Use ARCHIVER to)
(restore.)

1987CALEN 20 SECTORS
(Just what it says.)

TINY/CAL 8 SECTORS
(Very small calendar.)

CHESS 148 SECTORS
(Assembly code Chess game that can be)
(played over the modem or against the)
(computer. Use ARCHIVER to restore.)

JACKET 46 SECTORS
(A disk sleeve prgm. Use ARCHIVER to)
(restore.)

SIDE*PRIN 183 SECTORS
(Updated version of sideway print.)
(Use ARCHIVER to restore.)

READDV80 16 SECTORS
(A program to read DV80 files without)
(the TI-WRITER.)

TAXPLAN86 41 SECTORS
(This is to help you with the New Year)
(Headache. Use Before April 30th.)

XBLOAD 13 SECTORS
(Will load prgms from 2 drives.)

E/ALOADER 49 SECTORS
(Will create a menu for loading)
(E/A prgms. Must use the E/A Cart.)

SCLOADER 56 SECTORS
(Source code for the above prgm.)

REDISITV1 19 SECTORS
(Disk copy prgm for use with the)
(CC disk controller. Will copy a)
(DSDD disk in 1'10".)

TEXT128 13 SECTORS
(Will convert a DF128 file to a)
(DV80.)

DSKFRACT 18 SECTORS
(Text file on how to reconstruct)
(fractured files.)

RECVRFILE 71 SECTORS
(Text file on how to recover files.)

MONOPLY 95 SECTORS
(Runs in XBASIC. Good game. Very)
(slow loading. Use ARCHIVER to restore.)

SUPERCAT 62 SECTORS
(Use ARCHIVER to restore. Good disk)
(cataloger).

CODELDR 19 SECTORS
(Supposed to load GPL programs.)
(Loads from OPT5.)

OPT/5 v3.0 7 SECTORS
(Update of XBasic option 5 loader.)
(Very fast. Author claims will load)
(most any Opt 5 program.)

IBM-COPY 181 SECTORS
(Supposed to backup IBM disk using)
(CC disk controller and SBUG. Use)
(ARCHIVER to restore.)

ULTRACOPY 20 SECTORS
(Use Option 5. Will only work with)
(the TI-disk controller.)

RAPIDSCRL 211 SECTORS
(Use ARCHIVER to restore.)

RX-80MODE 22 SECTORS
(Sets up your Epson Printer.)

DELAWARE VALLEY USERS' GROUP
P. O. BOX 6240
STANTON BRANCH
WILMINGTON, DE 19804

Subscription
expires with
date on label

FIRST
CLASS
MAIL

TO:

0000 EDMONTON 99'ER ONLINE
P.O. BOX 11983
EDMONTON ALB. CANADA T5J 3L1