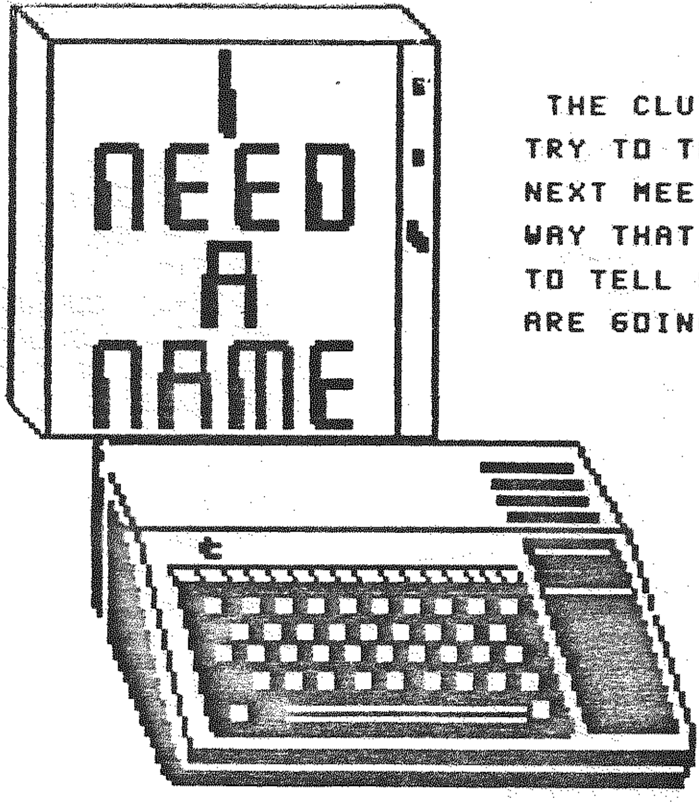

THE CLUB NEEDS A NAME !!! TRY TO THINK OF ONE FOR THE NEXT MEETING. IT'S THE ONLY WAY THAT YOU WILL BE ABLE TO TELL SOMEONE WHERE YOU ARE GOING. THINK ABOUT IT !

OCTOBER 1985 NO. 1

I think that the T.I.99 is alive and well in Westmoreland county !!! The meeting held at the Greengate Mall on September 23rd, although an organizational meeting, was well attended. Nearly 30 people were in attendance through the course of the evening. We discussed many items, from where to have the meetings, to what's comming out for the T.I.(including its possible successor).

I have been impressed with the people here who have expressed both enthusiasm and willingness. There is a very wide variety of talent and interests within the group, and this should present both a challenge and a feeling of fullfillment for those who have volunteered to teach and those who will participate in the Special Interest Groups.

Some of you have been hard at work, seeking a larger facility, with more appropriate room for the S.I.G.'s to meet. Some are already writing articles for this, and the next newsletter, some have offered suggestions (these are very important), and others have indicated their confidence in the success of this club. We NEED IT ALL!!!

The next meeting will be held on OCTOBER 14,1985, at the GREENGATE MALL COMMUNITY room, at 7:00P.M. PROMPTLY !! Since the meetings are on a week night, time must be used to the best advantage. We are beginning two S.I.G's , they are "BASIC" and "EXTENDED BASIC", both will start immediately after the club meeting (approximately 8:00 to 8:30 P.M.. Please attend and bring anyone who is interested. CHARLES STRINK will be teaching BASIC, and JOHN ANDRASKO will be teaching the EXTENDED BASIC class. We cannot begin any others until November, due to space and time considerations. T.I.WRITER, and HARDWARE S.I.G.'s will probably be the next to begin in November.

Please excuse the size of this print and the format of this article but this is my first attempt at using T.I.WRITER, and the printer I'm using is on that I borrowed to print out some graphics. I changed some switches on it, and now I can't find the documentation to set the switches back. This should show you that T.I.WRITER is pretty good, because even a novice can put out a letter .

We are looking for articles of concern to the club to print in this newsletter. The articles can deal with hardware, software, gotchya's, or even ongoing tutorials. I beleive that we can even produce a USER'S REFERENCE GUIDE that would put anything currently in existance to shame, if we put the knowledge we have together!

Some things you should give consideration to before the next meeting are: what DAYS/EVENINGS you can attend the club meetings and/or S.I.G.'s, the TIMES that suit, and of course a dues amount you could afford (there's got to be something bad about everything). Keep in mind that dues for clubs of this kind are generally between $10.- $20. per year.

Tom Hare, has gone to Texas, you know where our machine was built, to work, but before he left, he gave me around 150 to 200 programs on ta and disk to begin our library with. Ken Rambler also has a large library which can be merged with these and any other programs that are NOT copyrighted to form our new library. We must hold off on providing these in any form until LIBRARIANS can be chosen and given time to provide copies and indexes. You can see if we started to lend them out now, we would not have much of a library in two months.

By the way, for any one reading this, Ken Rambler has a Bulletin Boar set up on one of his systems, and for any one with the equipment, the phone number is 412-925-8059 (New Stanton, PA.). Perhaps some meeting soon we will demonstrate what equipment is necessary, and how it works.

Here is my address and phone number:

JOHN F. WILLFORTH
R.D.#1 BOX 73A
JEANNETTE, PA.
15644

oops! 527-6656

BASIC Basics
by Charles Strink

Well, here we go with the first newsletter of our new TI Users club.

We will have a lot of busy and exciting days ahead of us. New club members to meet. New classes to attend and new programs to use.

To make a group operate well, each member must not only learn from the group but must also be willing to share any knowledge you might have. Even if you are a novice to computing you still have knowledge that will benifit the group.

The following program is for you that are new to computing. It is written in Basic so all you need is your keyboard hooked up to a TV set. (some of you oldtimers may like this also)

```
 10 CALL CLEAR
 20 DEF R=INT(16*RND+1)
 30 RANDOMIZE
 40 FOR I=1 TO 14
 50 CALL SOUND(-50,R*110,4)
 60 CALL COLOR(I,R,R)
 70 CALL SCREEN(R)
 80 CALL HCHAR(R+4,R*2,R*R/2,R*R)
 90 CALL VCHAR(R+4,R*2,R*R/2,R*R)
100 NEXT I
110 GOTO 110
120 END
```

See you at the next meeting. Until then HAPPY COMPUTING !!!!

RANDOM ACCESS MEMORY

Did you ever play with balloons? One game I remember you could either play by yourself or with others. I doubt if it had a name, but the object was to keep the balloon up in the air by periodically hiting it as it started to fall toward the ground. Once it hit the ground you lost and were out of the game. We would try to let the balloon get as close to the ground as possible before we gave it another hit to keep it in the air.

Our 99/4A has 16K bytes of <u>R</u>andom <u>A</u>ccess <u>M</u>emory to which we can read or write. RAM memories in computers are usually one of two types: Static RAM or Dynamic RAM. Static RAM is essentially a switch (flip-flop/on-off). Once you turn it on or off by way of your programming, it will stay that way until either your program changes it or you turn the power off to your computer.

Dynamic RAM functions the same as Static RAM, however it constantly looses its ability to keep the on/off switch in the position you want it in unless it is periodically reminded (refreshed) as to where it should be. Back to the balloon--It "forgot" it was to be up in the air so it fell, you hit it again to remind (refresh) its memory to stay up. (By way of contrast, the Static balloon was glued to the ceiling/tree limb, etc. to keep it up.) Dynamic RAM needs to be "hit" again and again by the microprocessor to refresh its memory.

TI uses the TMS4116 Dynamic RAM chip in our computer (16,384 locations by 1 bit per location)--8 chips are used to give us 16K RAM on a 8 bit address bus. Why did TI use Dynamic RAM instead of Static RAM?

1. Dynamic takes only 1 to 3 transistors per switch, while Static takes 4 or more.
2. Dynamic takes less physical room on a circuit board vs. Static.
3. Dynamic consumes much less power and generates less heat than Static.
4. Dynamic is cheaper ($) than Static.
   (Apple=Static: more chips, more heat, more $, etc. TI=Dynamic: less of everything.)

The ONLY advantage the Static RAM has over Dynamic RAM is that Static RAM does not have to be "refreshed" periodically by the microprocessor. This refreshing does come at a cost--it takes up 2.2% of our processor's activity. This means that our TMS9900 processor runs 2.2% slower with Dynamic RAM than it would with Static RAM. Given the reasons listed above, TI thought the trade off well worth their effort.

So when you tire of playing with TI's Dynamic RAM, go find a balloon. Keep it up in the air while reading a book. You will then be like a processor refreshing Dynamic RAM while you are involved in another project. Remember, if the balloon hits the ground, all of your RAM programs are lost.

Rev. Keith G. F

PROGRAMMING TIPS

Faster Keyboard Response from the User Group of Orange County, Calif., newsletter:

The following routine will give much faster response to keyboard entries (especially when deep in a program):

```
820 FOR I=0 TO -1 STEP -1
830 CALL KEY(0,K,S)
840 I=S=1
850 NEXT I
```

Line 840 means that if S=1 then set I=-1 (the value for TRUE), if S<>1, then set I=0 (the value for FALSE). Line 850 says that if I=-1 then fall through. If I=0, then set I=-1 and go to 830 using a pointer (instead of the normal GOTO 830, which would have to start searching for a line number 830 at the beginning of the program). This method also can be used to speed up response to the joystick, with the CALL JOYST statement.

* * *

Another CALL KEY routine from the Dallas newsletter, written by Richard Lamson and Wes Irby:

```
100 CALL KEY(0,K,S)
110 IF S=0 THEN 100
120 ON POS("RZ1A4",CHR$(K),1)+1 GOTO 200,250,300,350,400,450
```

The program goes to line 200 if K is not R,Z,1,A or 4; to 250 if K=R; to 300 if K=Z; to 350 is K=1; to 400 is K=A; to 450 if K=4.

* * *

An Extended BASIC routine from the Cin-Day Users' Group Newsletter written by Jim Schwaller:

This program will allow you to move a sprite in any direction with the direction controlled by the joystick control and speed by the fire button. This program is designed to that two people can move the sprites with the two joysticks:

```
100 CALL CLEAR :: CALL CHAR(96,RPT$("FFFFFFFF",8)) :: CALL SPRITE(#1,
    96,5,92,124) :: CALL CHAR(104,RPT$("FFFFFFFF",8)) :: CALL SPRITE(#2,
    104,7,92,124) :: CALL MAGNIFY(4)
110 CALL JOYST(1,A,B) :: CALL JOYST(2,C,D)  :: CALL KEY(1,F,G) :: CALL
    KEY(2,H,I) :: J=J+F/9-1 :: K=K+H/9-1
120 IF J<0 THEN J=0 :: IF K<0 THEN K=0 :: IF J>31 THEN J=31 :: IF
    K>31 THEN K=31
130 CALL MOTION(#1,-(B*J),A*J) :: CALL MOTION(#2,-(D*K),C*K) :: GOTO 110
```

UNPROTECTING XBASIC PROTECTION

There was a short immediate mode listing that was supposed to break the protection on a program. I tried it and it works. I now have all the programs on my working disk and my purchased one is the backup. Here is the procedure.

Have your program on disk as DSK1.XXXXXX (for example), then type:

```
CALL FILES(1)          (enter)
NEW                    (enter)
CALL INIT              (enter)
OLD DSK1.XXXXXX        (enter)
CALL LOAD(-31931,0)    (enter)
```

You should now be able to SAVE or LIST.

JOHN JENSEN

OR

LOAD PROGRAM FROM CASSETTE OR DISK. PRESS "FUNC 4" AND:

```
> CALL INIT <ENTER>
> CALL LCAD(-31931,Ø) <ENTER>
```

NOW YOU CAN LIST THE PROGRAM AND SAVE IT TO CASSETTE OR DISK.

John Dow has sent in the following demonstration program. His explanation comes first and program notes follow:

"The TI99/4(A) allows you to use an ordinary cassette recorder as an inexpensive storage device for programs or data. However, each time data are written to a tape, a lengthy tone precedes the record. This makes tapes very slow for data storage unless the data are packed into records. This is called blocking.

This demonstration program allows you to type in strings (into the variable E$) to be written to tape and then lets you read them back in again. It shows how to do this as efficiently as possible by packing the strings into 192-character records.

```
100 REM  DATA STORAGE ON CASSETT
E DEMONSTRATION PROGRAM
110 CALL CLEAR
120 PRINT "DATA STORAGE ON CASSE
TTE":"DEMONSTRATION PROGRAM.":::
130 PRINT ::"1:WRITE TO CASSETTE
":"2:READ FROM CASSETTE":"3:STOP
"::
140 INPUT "CHOOSE 1-3:":A
150 PRINT ::
160 IF A=2 THEN 330
170 IF A=3 THEN 430
180 REM  WRITE TO CASSETTE
190 PRINT "ENTER TEXT TO BE WRIT
TEN":"TO TAPE.":"TYPE 'END' TO S
TOP."
200 OPEN #1:"CS1",OUTPUT,FIXED 1
92,INTERNAL
210 INPUT E$
220 IF SOFAR+1+LEN(E$)<190 THEN
250
230 PRINT #1:"."
240 SOFAR=0
250 PRINT #1:E$,
260 SOFAR=SOFAR+1+LEN(E$)
270 IF E$="END" THEN 300
280 GOTO 210
290 PRINT #1:
300 CLOSE #1
310 GOTO 130
320 REM  READ FROM CASSETTE
330 OPEN #1:"CS1",INPUT ,FIXED 1
92,INTERNAL
340 INPUT #1:E$,
350 IF E$<>"." THEN 380
360 INPUT #1:E$
370 GOTO 340
380 IF E$="END" THEN 410
390 PRINT E$
400 GOTO 340
410 CLOSE #1
420 GOTO 130
430 END
```

Notes on the program:

1) Use "pending print" to append strings to build the record. (See statements 250 and 340)

2) On output, count how much data are already in the record. Count the length of the string plus another byte for the length to be stored. (See statement 260)

3) Put a special one-character string at the end of each record to mark the end. This is "." is the demo program. (See statements 230 and 350)

4) You may need a special string to mark the end of all the data, depending on whether your program knows exactly how many strings are on the tape. In the demo program "END" is typed by the user and is then written to the tape. (See statements 270 and 380)

5) If you store numerical data, each value requires 8 bytes plus another to indicate its type, so at statement 260, you would add 9 instead of 1+LEN(E$).

3-D IMAGING ON THE 99/4 -- This program, written by Bill van Kerkoerle of the Netherlands and reprinted in the MSP 99 Newsletter in Minnesota by editor Richard Clemetson, provides an interesting demonstration of the highest display resolution available on the 99/4. It simulates a 3-D image using small linear segments and produces a symmetrical picture often seen in computer graphics advertisements. Although this version has certain features to speed up execution, Clemetson warns that it takes about 12 minutes and 48 seconds to draw the entire image. The first half takes about three-quarters of the time; the second half is simply a mirror image of the first and, therefore, takes less time to draw. This program was originally presented in TI Home Tidings, an English 99/4 computer newsletter.

```
100 DIM C$(128)
110 GOTO 330
120 Y=INT(R/8+.875)
130 X=INT(C/8+.875)
140 CALL GCHAR(Y,X,H)
150 IF H>31 THEN 220
160 IF S=95 THEN 320
170 S=S+1
180 C$(S-31)=Z$&CHR$(Y)&CHR$(X)
190 CALL CHAR(S,Z$)
200 CALL HCHAR(Y,X,S)
210 H=S
220 H=H-31
230 B=C-X*8+8
240 P=2*R-16*Y+16+(B<5)
250 IF B<5 THEN 270
260 B=B-4
270 I$=SEG$(B$,POS(H$,SEG$(C$(H),P,1),1),4)
280 I$=SEG$(I$,1,B-1)&"1"&SEG$(I$,B+1,4-B)
290 I=POS(B$,I$,1)
300 C$(H)=SEG$(C$(H),1,P-1)&SEG$(H$,I,1)&
    SEG$(C$(H),P+1,18-P)
310 CALL CHAR(H+31,C$(H))
320 RETURN
330 CALL SCREEN(8)
340 S=31
350 CALL HCHAR(1,1,S,768)
360 B$="0000.0001.0010.0011.0100.0101.0110.0111.
    1000.1001.1010.1011.1100.1101.1110.1111"
370 H$="0....1....2....3....4....5....6....7....
    8....9....A....B....C....D....E....F"
380 HR$="0....8....4....C....2....A....
    6....E....1....9....5....D....3....
    B....7....F"
390 Z$="0000000000000000"
400 H1=192
410 V=104
420 X1=H1/2
430 X2=X1*X1
440 Y1=V/2
450 Y2=V/4
460 FOR X5=0 to X1 STEP 2
470 X4=X5*X5
480 M=-Y1
490 A=SQR(X2-X4)
500 FOR I1=-A TO A STEP V/10
510 R1=SQR(X4+I1*I1)/X1
520 F=(R1-1)*SIN(R1*12)
530 R=INT(I1/5+F*Y2)
540 IF R<=M THEN 590
550 M=R
560 R=Y1-R
570 C=X1-X5+32
580 GOSUB 120
590 NEXT I1
600 NEXT X5
610 H=S-31
620 FOR K=1 TO H
630 O$=C$(K)
640 Y=ASC(SEG$(O$,17,1))
650 X=33-ASC(SEG$(O$,18,1))
660 FOR L=2 TO 16 STEP 2
670 I$=SEG$(O$,L,1)
680 J$=SEG$(O$,L-1,1)
690 I$=SEG$(HR$,POS(H$,I$,1),1)
700 J$=SEG$(HR$,POS(H$,J$,1),1)
710 C$(K+H)=C$(K+H)&I$&J$
720 NEXT L
730 CALL CHAR(K+S,C$(K+H))
740 CALL HCHAR(Y,X,K+S)
750 NEXT K
760 GOTO 760
```