NEW HAMPSHIRE 99'ERS
AUGUST 1984
NEWSLETTER

NHUG NEWSLETTER - PO BOX 7199 HEIGHTS STATION - CONCORD, NH 03301

TI-FORTH

TI-FORTH is a language that combines the speed of assembly language with the ease of use of and power of TI-LOGO and the versatility of TI-EXTENDED BASIC.

TEXAS INSTRUMENTS has made TI-FORTH available to a selected number of TI user groups across the country, to sell as Public Domain software. And your local NH 99'ER USER GROUP is one of the chosen ones!

The required hardware to run TI-FORTH is a 32K Memory Expansion, a Disk Drive and Controller, and the Editor Assembler Module. An RS-232 Peripheral and Printer are optional. Also recommended is the book STARTING FORTH by Leo Brodie of FORTH, Inc. This is almost a necessity for most people because the TI-FORTH manual does not explain all of the basics of FORTH Programming, but instead refers to Brodie's book for many needed examples. Like many manuals, the TI-FORTH manual is an excellent reference for all commands, but assumes a basic knowledge of the FORTH language.

The NH 99'ER group is offering the TI-FORTH Language Package (manual + disk) to members for $17 and to non-members for $32 (add $3 for shipping and handling). The manual alone contains over 100 double-sided pages! Please allow 4 weeks for processing your order.

TIdBIT: TI-FORTH

Screen 72, line 5, change "PAB_ADDR" to "PAB-ADDR".

FORTH manual, chapter 6, page 3, "SCREEN" must follow 2 color values, foreground and background colors, or screen writing becomes transparent.

WE NEED ARTICLES AND IDEAS

We don't mind doing the work because we get a great deal of satisfaction in the finished product..like an artist stepping back to admire his latest creation. But we do need your ARTICLES AND IDEAS.

In order to fill empty spaces we sometimes use articles from other User's clubs. Why not articles from our own club members?

REDO REVISITED

Richard J. Bailey
68A Church Street
Gonic, N.H. 03867

One use of the REDO key that wasn't mentioned in the July newsletter is that it allows you to enter lines up to 178 characters long. To see how to do this, enter the following line;

```
310 IF I=41 THEN DISPLAY AT(
15,3)ERASE ALL:"*WARNING  >4
0 PROGRAMS":"":" ON DISK-40
IS MAX ALLOWED.":"":"  *PROG
RAM TERMINATED *":: CALL SOU
ND(800,220,0,444,4,900,8) ::
END
```

After you type in the 140th character, the computer won't let you enter more. Press ENTER and REDO. If you haven't any error messages for unmatched quotes, etc., the line will reappear. Use right arrow (FCTN D) to move the cursor to the end of the line and continue typing. You can get up to 178 characters on that line. This number seems to vary with the line content but should be between 166-178.

If you enter too many characters you'll get an error message "LINE TOO LONG" and will have to REDO and reenter from the end of the original line.

Also if you try to REDO a line that is over 140 characters long, only 140 characters will be displayed of that line. The original line remains intact unless you press ENTER without changing the line number on the REDO line in which case the truncated line replaces the original long line. Once the line is over 140 characters you can use FCTN X to edit the line.

This use of the REDO key can be useful in lines that would not be easy to divide, like 270 below.

```
270 DEF DI$(R)=CHR$(162)&CHR
$(240)&CHR$(183)&CHR$(200)&C
HR$(LEN(STR$(R)))&STR$(R)&CH
RS(179)&CHR$(200)&CHR$(LEN(S
TRS(COL)))&STR$(COL)&CHR$(18
2)&CHR$(181)
```

Personalized Math Program

The following program (from TI USER'S NEWSLETTER - 1/82, No.1) is an example of how easy it is for you to use a TI99/4A in developing an educational program for your child. This program uses TI's innovative SOLID STATE SPEECH technology feature and requires a Terminal Emulator II Module and the Speech Synthesizer.

Once you have it keyed in, RUN it and simply follow the instuctions given. To find out your score during an exercise, type SCORE when asked for an answer. To end the program, simply type END.

The following is a brief description of what each section of the program does.

```
100-200 initialization of speech, graphics, and variables
210-300 input name and number
310-450 problem display and input
460-540 correct response routine
550-630 incorrect response routine
640-670 initialization of error counters
680-750 problem input
760-960 display of score and reward messages
```

1. Turn the computer ON and wait for the master title screen to appear. Then slide the TE II module into the slot on the console.
2. Press any key to make the master selection list appear. Select [1] TI BASIC.
3. Key in the following program (with [ALPHA LOCK] DOWN!):

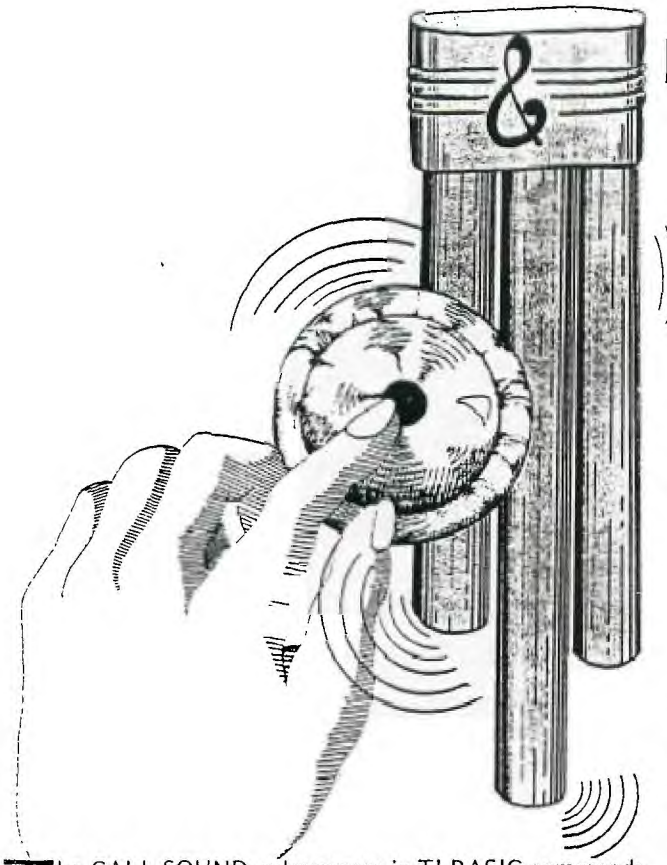
```
100 REM  MATH DRILL AND PRAC
TICE
110 CALL CLEAR
120 OPEN #1:"SPEECH",OUTPUT
130 CALL CHAR(96,"00")
140 CALL COLOR(9,10,10)
150 CALL SCREEN(16)
160 CALL CHAR(104,"00")
170 RANDOMIZE
180 CALL COLOR(10,13,13)
190 TTRY=0
200 TRIGHT=0
210 PRINT #1:"WHAT _IS ^YOUR
  NAME? "
220 INPUT "WHAT IS YOUR NAME
 ":N$
280 PRINT #1:"WELCUM TO MY C
LASS FOR^SMART^ CHILDREN _"&
N$
290 PRINT #1:"WHAT IS ^YOUR
FAVORITE ^NUMBER _ _"&N$
300 INPUT I
310 N=5
320 CALL CLEAR
330 GOSUB 640
340 REM  I=INT(RND*N)
350 J=INT(RND*N)
360 NTRY=NTRY+1
370 TTRY=TTRY+1
380 IF RND*10<5 THEN 410
390 IF TTRY<3 THEN 410
400 GOSUB 760
410 PRINT " "
420 PRINT I;TAB(6);"+";TAB(9
);J;TAB(15);"=";TAB(18);
430 GOSUB 680
440 IF FLAG=1 THEN 410
450 IF K<>I+J THEN 550
460 TRIGHT=TRIGHT+1
470 CALL HCHAR(23,26,104)
480 PRINT #1:"YOU _ARE^RIGHT
_"&N$
490 IF NTRY<3 THEN340
500 SCORE=(NTRY-NERR)/NTRY
510 IF SCORE<.74 THEN 340
520 N=N+5
530 GOTO 330
540 GOTO 340
550 REM WRONG ANSWER
560 CALL HCHAR(23,26,96)
570 CALL SOUND(500,-3,0)
580 FOR LOOP=1 TO 300
590 NEXT LOOP
600 PRINT #1:"_NO ^"&N$&" TH
AT IS _ _ _ _WRONG"
610 NERR=NERR+1
620 TTRY=TTRY+1
630 GOTO420
640 REM  REINT ERROR COUNT
650 NERR=0
660 NTRY=0
670 RETURN
680 INPUT A$
690 JFLAG=1
700 IF A$="SCORE" THEN 760
710 IF A$="END" THEN 880
720 K=VAL(A$)
730 FLAG=0
740 JFLAG=0
750 RETURN
760 REM PRINT OUT SCORES
770 YSCORE=TRIGHT/(TTRY-1)*1
00
780 IF YSCORE>90 THEN 900
790 IF YSCORE>79.9 THEN 920
800 IF YSCORE<80 THEN 940
810 IF JFLAG<>1 THEN 850
820 PRINT "YOUR SCORE IS ";T
AB(16);YSCORE
830 STRY=TTRY-1
840 PRINT "YOU HAVE TRIED ";
TAB(16);STRY;TAB(20);" PROBL
EMS"
850 FLAG=1
860 JFLAG=0
870 RETURN
880 GOSUB 760
890 END
900 PRINT #1:"_YOU ARE DOING
^ VERY ^WELL _"&N$
910 GOTO 810
920 PRINT #1:"_YOU ARE ^DOIN
G ^O _K BUT ^NEED ^PRACTICE
_"&N$
930 GOTO 810
940 PRINT #1:"^OH..._DEAR. _
"&N$&" ^YOU ^NEED LOTS OF ^P
RACT_ESS"
950 PRINT #1:"I HOPE THAT _Y
OU  WILL ^LET _ME _HELP ^YOU
 _"&N$
960 GOTO 810
```

COMPUTE! INDEX

New member Ellen Rule of Concord has submitted the following index of TI99/4A related articles that have appeared in COMPUTE! from 3/83 to 6/84. They were compiled with the aid of the PERSONAL RECORD KEEPING module. She writes: "I have found that module to be quite useful, once having figured it out, and will share information on it if anyone is interested" (Editor's note: This is exactly the reason for the formation of the NH TI 99'er User Group. To promote a better understanding and appreciation of the versatility and usefulness of this exceptional tool. By such exchange and sharing of knowledge in this newsletter we hope to realize the full potential of your investment. JOIN US!)

Ellen has owned her CPU since 1982, has upgraded her system to include the REX Box with a single disk drive, 32K and AXIOM printer. She recently tractored a mile of paper to perfect a screen dump program for the AXIOM and has currently been working on learning the idiosyncrasies of TI WRITER.

NAME	DATE(M/Y)	PROG/TYPE	PG #'S
C/A/D -- ART	9/83	ART	236
TYPING TEACHER	4/83	EDU/GAME	76
FIRST MATH	8/83	EDU/GAME	92
CLUES	8/83	EDU/GAME	109
MYSTERY SPELL	9/83	EDU/GAME	112
MATCH-EM	4/83	EDU/GAME	123
MEMORY TRAINER	6/83	EDU/GAME	112
STATISTICS	12/83	EDUCATION	196
AQUARIUM	3/84	ENTERTAIN	138
BOGGLER	3/83	GAME	78
ASTROSTORM	6/83	GAME	72
ALPHA BLAST	11/83	GAME	94
TI TOWERS	9/83	GAME	142
GOLD MINER	8/83	GAME	113
SPACE STATION 1	8/83	GAME	132
SNERTLE	5/84	GAME	88
GET THE GOLD	12/83	GAME	132
ROBOT RUNNER	1/84	GAME	146
ROADER	3/84	GAME	66
DIAMOND DROP	9/83	GAME	76
GOBLIN	7/83	GAME	64
MOSAIC PUZZLE	10/83	GAME	90
JUMPING JACK	5/83	GAME	3[illegible]
GOODBY CHARLIE	11/83	GAME	68
QUATRAINMENT	2/84	GAME	76
DEVASTATOR	8/84	GAME	58
JACKPOT	8/84	GAME	83
CIRCUS	2/84	GAME	62
SPRITE EDITOR	9/83	GRAPHICS	258
RUNWAY 180	10/83	GRAPHICS	208
PROGRMNG GRAFCS	5/83	GRAPHICS	218
GRAFCS MADE E-Z	3/83	GRAPHICS	205
TI GRAPHICS	6/84	GRAPHICS	110
CALORIE COP	12/83	HOUSEHOLD	52
MAILING LIST	7/83	HOUSEHOLD	242
PHONE DIR/DIALR	11/83	HOUSEHOLD	212
MICRO MECHANIC	1/84	HOUSEHOLD	52
PAYCHCKANALYSIS	12/83	HOUSEHOLD	66
UTIL.BILL AUDIT	12/83	HOUSEHOLD	72
RETIREMNT PLNNR	4/83	HOUSEHOLD	71
WORD PROCESSOR	12/83	HOUSEHOLD	314
COUPON FILE	10/83	HOUSEHOLD	52
INTRO DATA STOR	3/83	INSTRUCTN	46
FILE PROCESSING	3/84	INSTRUCTN	156
ANSWERS TO ?'S	11/83	INSTRUCTN	228
TROUBLE-SHOOTNG	8/83	INSTRUCTN	15[illegible]
PLNNING COLOR	7/83	INSTRUCTN	19[illegible]
PROGRAM EVOLUTN	5/84	INSTRUCTN	142
PROGRAMMNG TIPS	1/84	INSTRUCTN	152
ESTIMATNG MEMRY	4/83	INSTRUCTN	21[illegible]
TRICKS AND TIPS	4/84	INSTRUCTN	165
RIGHT WORDPROC	4/83	INSTRUCTN	24
EASY EDITING	3/83	INSTRUCTN	201
DATA/READ/RSTOR	8/83	INSTRUCTN	192
DATA FILING SYS	10/83	INSTRUCTN	213
SECONDRY EDUCTN	4/83	INSTRUCTN	180
DATA FILING SYS	7/83	INSTRUCTN	156
TRANSLATNG TIPS	6/83	INSTRUCTN	182
DATA FILING SYS	9/83	INSTRUCTN	210
DATA FILING SYS	8/83	INSTRUCTN	158
BASIC 1-LINERS	5/83	INSTRUCTN	250
STRUCTURED BASC	6/83	INSTRUCTN	204
SUBSCRPTD VAR'S	9/83	INSTRUCTN	221
DATA BASE	5/83	INSTRUCTN	226
FILE PROCESSING	4/84	INSTRUCTN	135
COMPUTER FUN	12/83	INSTRUCTN	252
FILE PROCESSING	5/84	INSTRUCTN	159
USING A PRINTER	6/83	INSTRUCTN	61
CHARACTER SET	11/83	INSTRUCTN	273
SOUND SHAPER	3/84	MUSIC	149
MOZART MACHINE	1/84	MUSIC	160
PLAYING MUSIC	10/83	MUSIC	224
PENTOMINOS	5/84	PUZZLE	106
CROSSWORDS	5/83	PUZZLE	76
MATH ACTN GAMES	3/83	REVIEW	108
GAME MODULES	9/83	REVIEW	181
COMPUTER WAR	2/84	REVIEW	134
FOREIGN LANG	2/84	SPEECH	168
SINGNG COMPUTR	8/84	SPEECH	115



LIVENING UP YOUR CALL SOUNDs

By: Al Kanda
Box 3494
Scottsdale, AZ 85257

The CALL SOUND subprogram in TI BASIC commands an amazing integrated circuit in your TI-99/4A, called the SN76489 Sound Generation Controller. On a single chip, TI has squeezed in three programmable frequency dividers, a programmable noise generator, four programmable attenuators (volume controls), and eight registers to hold the data that control the tones, noise, and their volume levels. In effect, the tones and noise are synthesized to your specifications from a frequency of 3.58 megahertz; this is also the frequency that carries the color information from your computer to your color monitor or video modulator.

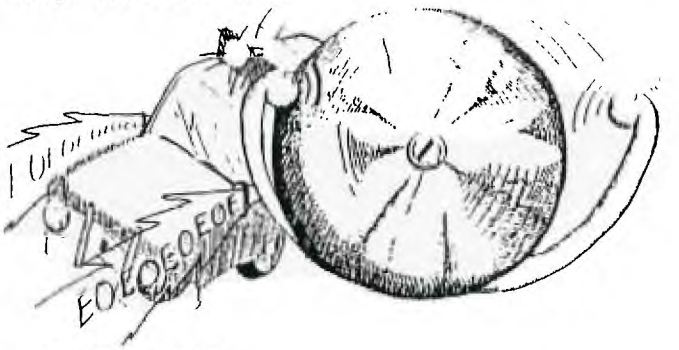
If the only use you have made of CALL SOUND has been to produce miscellaneous beeps, noise, and music, read on. I'm going to give you some "mini programs" that demonstrate the variety of other sounds your 99/4A is capable of producing.

For the first example, let us try to recreate the sound of a door bell of the type associated with the once popular "Avon Calling" commercial. This is an example of an object that is struck with a sharp blow and allowed to vibrate at its resonant frequencies. The following characteristics are needed to recreate this sound: 1) the fundamental frequencies of the two tones, 2) the overtone frequencies, and 3) a gradually decaying volume. Those of you with a sense of absolute pitch would immediately recognize the two fundamental frequencies, but in my case, I actually measured the dimensions of the sounding bars, their points of support, and determined with a magnet that the bars were probably steel. From a textbook, *Acoustical Engineering* by Harry F. Olson, I obtained the formula and values of the constants needed to calculate the resonant frequencies of the bars. The calculated frequencies came out to be very close to 698 and 554 cycles per second (F and C# above high C). The book also told me that the two closest overtones were 2.756 and 5.404 times the fundamental frequency. The bars were supported on rubber mounts close to the theoretical nodes (points of minimum vibration) for the fundamental and the first overtones, but were located near points of maximum vibration for the second overtone. I therefore assumed that the second overtone would be dampened out, so I omitted it from the CALL SOUND specification for each tone. The decaying volumes for the tones were obtained by including each CALL SOUND in a FOR–NEXT loop as follows:

```
100 REM DOOR CHIMES
110 FOR A=0 TO 30 STEP 5
120 CALL SOUND(-99,698,A,1924,A)
130 NEXT A
140 FOR A=0 TO 30 STEP 5
150 CALL SOUND(-99,554,A,1527,A)
160 NEXT A
```

If you are wondering about the significance of the 99 for the durations (other than this being a 99'er article), it is simply an easily keyed number larger than the 50 milliseconds needed to make the steps sound continuous. The minus sign indicates that the sound generator will be updated as soon as the new value for A is determined; the duration specified need only be long enough to cover the time between updates.

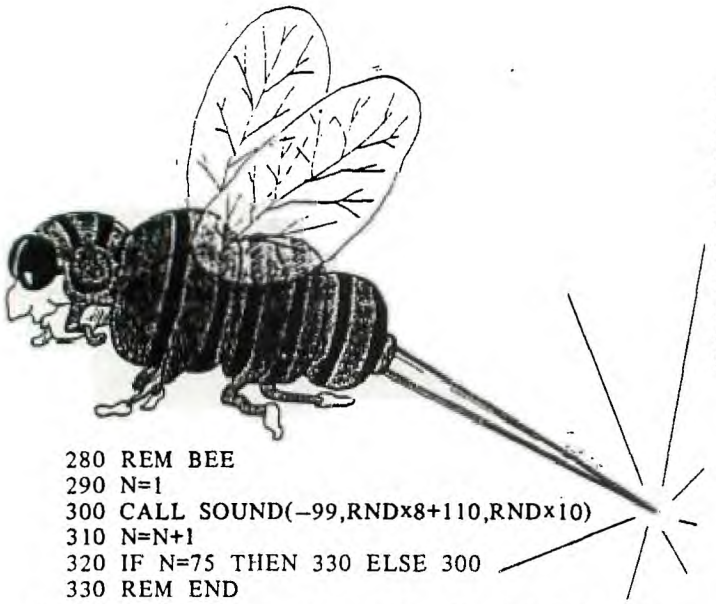
Next, let us try a sound in which the frequency varies with time. A siren is an example which can be characterized by a slowly rising and falling frequency. Apparently, this is a sufficient clue to the brain for us to recognize it as a siren. Try varying the frequency range and step in the following program, and see how far they can be varied and still have it recognizable as a siren.



```
170 REM SIREN
180 N=1
190 FOR F=700 TO 900 STEP 5
200 CALL SOUND(-99,F,0)
210 NEXT F
220 FOR F=900 TO 700 STEP -8
230 CALL SOUND(-99,F,0)
240 NEXT F
250 N=N+1
260 IF N=4 THEN 270 ELSE 190
270 REM END
```

N=4 on line 260 limits the siren to 3 up-down frequency sweeps.

In the next example, let us vary both the frequency and the volume as a function of time. Imagine a large "killer" bee buzzing around you, with the frequency of the buzz proportional to the rate of the beating wings, and the volume proportional to the closeness of the bee.



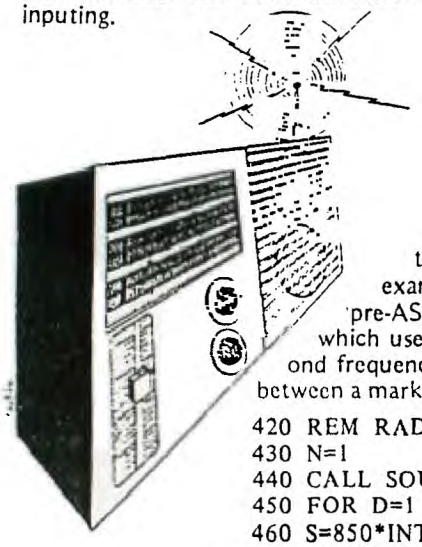
```
280 REM BEE
290 N=1
300 CALL SOUND(-99,RNDx8+110,RNDx10)
310 N=N+1
320 IF N=75 THEN 330 ELSE 300
330 REM END
```

Unlike the previous examples, where the variations in frequency and volume were obtained by using a FOR–NEXT loop, the variations in this case were obtained by using the RND statement. It is interesting to note that this routine will not sound the same in TI Extended BASIC–with the bee sounding very sluggish. This is one case in which TI BASIC runs faster than the Extended version.

For the next sound, imagine that you are tuning a short-wave radio receiver. The background static is simulated with noise type –8 and the random signal is simulated with frequency #3. The random volume on frequency #3 simulates varying signal levels with the noise volume formulated to be high when the signal level is low and vice versa.

```
340 REM SHORTWAVE RECEIVER
350 N=1
360 F=RND*15000+110
370 A=RND*30
380 CALL SOUND(-99,111,30,111,30,F,A,-8,30-A)
390 N=N+1
400 IF N=100 THEN 410 ELSE 360
410 REM END
```

Frequencies #1 and #2 are "do nothing frequencies," since their volumes are set to the minimum, and are inserted so the program will recognize frequency #3 from which noise type –8 is derived. The 111's therefore were picked for ease of inputing.



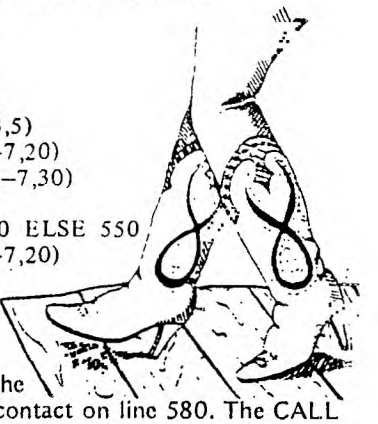
Next, imagine that the radio of the previous example is now tuned to a pre-ASCII teleprinter signal which uses an 850 cycle-per-second frequency shift to differentiate between a mark and space.

```
420 REM RADIO TELEPRINTER
430 N=1
440 CALL SOUND(22,2975,0)
450 FOR D=1 TO 5
460 S=850*INT(RND*2)
470 CALL SOUND(22,2125+S,0)
480 NEXT D
490 CALL SOUND(31,2125,0)
500 N=N+1
510 IF N=30 THEN 520 ELSE 440
520 REM END
```

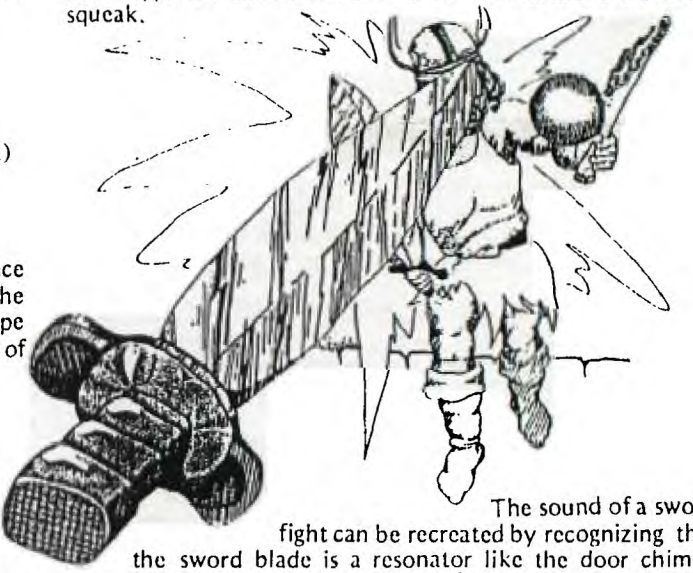
One character consists of a 22 millisecond (ms) start pulse, followed by a five bit code for the character, with each bit 22 ms long, and a 31 ms stop pulse. Line 440 generates the start pulse, which is always a space. The FOR–NEXT loop in lines 450-480 randomly generates a mark or space pulse for the five data bits, and line 490 generates the stop pulse, which is always a mark. Line 510 limits the number of characters generated to 29. Like the "bee" sound, this will not come out well in Extended BASIC. In general, data communications signals are easy to imitate because they are well defined by standards.

For a change of pace, try the following sound:

```
530 REM FOOTSTEPS
540 N=1
550 X=INT(RND*5)
560 IF X=2 THEN 620
570 CALL SOUND(5,-3,5)
580 CALL SOUND(30,-7,20)
590 CALL SOUND(500,-7,30)
600 N=N+1
610 IF N=30 THEN 640 ELSE 550
620 CALL SOUND(60,-7,20)
630 GOTO 590
640 REM END
```



The CALL SOUND on line 570 is the heel contacting the floor, followed by the sole contact on line 580. The CALL SOUND on line 590 is the delay between steps. Lines 550, 560, and 620 add a shuffle about once in every 4 steps to make the footsteps sound a little more natural. Changing the noise type on line 580 from –7 to –5 will make the shoes squeak.



The sound of a sword fight can be recreated by recognizing that the sword blade is a resonator like the door chimes, except that instead of being essentially free, it is clamped at the handle–thus creating overtones at different ratios than for the chime bars. Also, the amplitude decays faster, since the collision of the two blades would have a dampening effect.

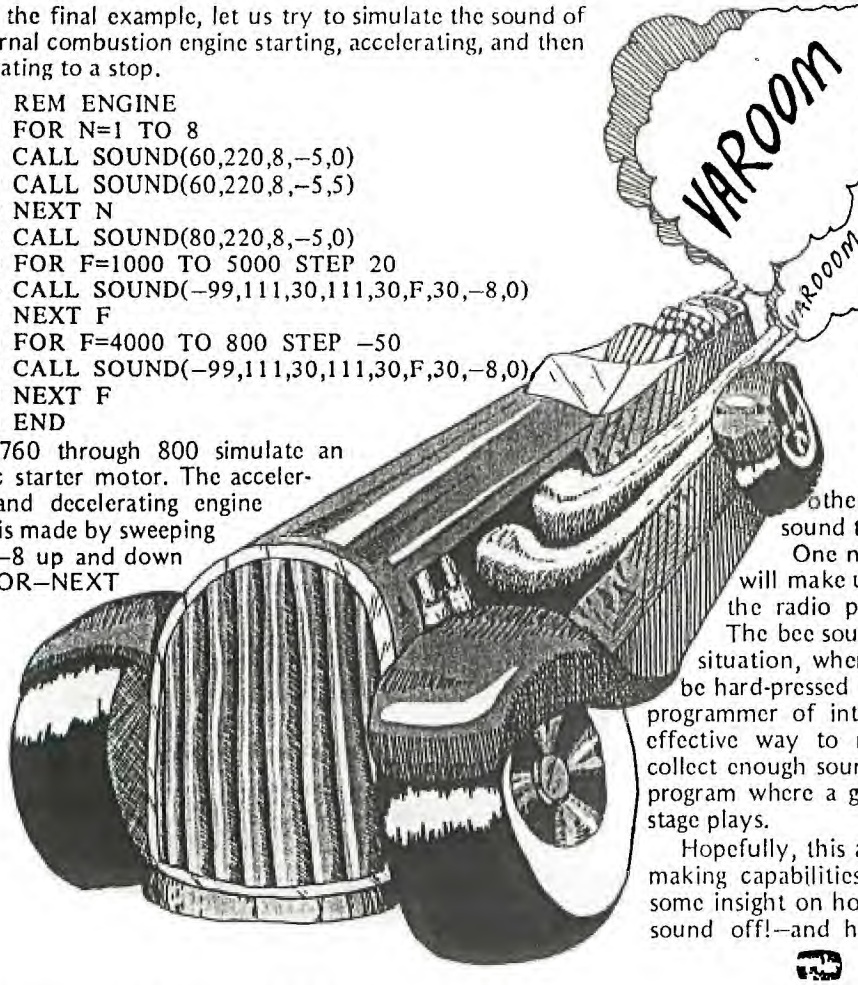
```
650 REM SWORD FIGHT
660 N=1
670 FOR A=0 TO 30 STEP 15
680 CALL SOUND(-99,1000,A,3250,A,6750,A)
690 NEXT A
700 FOR D=1 TO RNDx200
710 NEXT D
720 N=N+1
730 IF N=30 THEN 740 ELSE 670
740 REM END
```

Lines 700 and 710 add a random delay between sword clashes.

For the final example, let us try to simulate the sound of an internal combustion engine starting, accelerating, and then decelerating to a stop.

```
750 REM ENGINE
760 FOR N=1 TO 8
770 CALL SOUND(60,220,8,-5,0)
780 CALL SOUND(60,220,8,-5,5)
790 NEXT N
800 CALL SOUND(80,220,8,-5,0)
810 FOR F=1000 TO 5000 STEP 20
820 CALL SOUND(-99,111,30,111,30,F,30,-8,0)
830 NEXT F
840 FOR F=4000 TO 800 STEP -50
850 CALL SOUND(-99,111,30,111,30,F,30,-8,0)
860 NEXT F
870 END
```

Lines 760 through 800 simulate an electric starter motor. The accelerating and decelerating engine sound is made by sweeping noise -8 up and down in FOR-NEXT loops.



Now that you're convinced that your computer can produce a wide variety of sounds, you are probably wondering how one uses these sounds. If you are an adventure game programmer, suppose that the player is confronted with a door with a knocker and a bell button. Wouldn't it be more interesting if the player *heard* the bell upon pressing the bell button—before getting the usual textual message? Or if you are dynamically simulating a race car, you could use line 820 in the engine sound example in a CALL KEY loop where the F parameter would depend on the accelerator pedal setting. The duration in the CALL SOUND would have to be increased if you are updating other parameters in the loop—i.e., for the sound to be continuous.

One nice thing about sounds is that the listener will make up the visual image that fits, which is why the radio programs of years past were so effective. The bee sound, for instance, immediately conveys the situation, whereas a screenful of color graphics would be hard-pressed to evoke the same feeling. Thus, for the programmer of interactive fiction, sound should be a very effective way to make a story come alive. If you could collect enough sounds, you could even write a sound effects program where a given sound could be accessed on cue for stage plays.

Hopefully, this article has opened your ears to the sound-making capabilities of your TI-99/4A, and has given you some insight on how to create and use your own sounds. So sound off!—and have fun doing it.

NHUG SALES

C10 DATA CASSETTES	$1.00 ea
C20 DATA CASSETTES	$1.25 ea
5 1/4 SS/SD DISKS	$1.75 ea
FORTH Language Package	
members	$17.00 pk
non-members	$32.00 pk

These are CASH & CARRY prices. For mail orders, please add $3.00 for shipping and handling. Write for quantity discounts on cassettes.

PUBLISHER'S NOTES

The club would like to thank member Jerome Belair for submitting the "Enjoy your 'TI'" logo which has appeared in the last two newsletters.

NHUG members may place up to 1/4 page (classified type) ad per issue during their entire membership. Display ads at 25% off. All ads must be computer related.

Deadline for submissions is the 10th of each month.

I have over 600 non-copyrighted programs in my library. I will process your choices onto cassette, disk, or hardcopy for $1.00 each and copy unit. Comparable programs may be exchanged. For more info, see me at the next meeting or contact me at:

JEROME G. BELAIR
672 HUSE ROAD #38
MANCHESTER, NH 03103
603-669-9498
Please enclose a SASE for written info

NEXT MEETING
===========

WEDNESDAY

AUGUST 29, 1984

7:30 P.M.

PABLO'S RESTAURANT

D.W. HIGHWAY NO.

U.S. ROUTE 3

HOOKSETT

ENJOY YOUR 'TI' COMPUTER

N.H. 99'ERS

NEW HAMPSHIRE 99'ERS USER GROUP, INC.
P.O. BOX 7199, HEIGHTS STATION
CONCORD, NEW HAMPSHIRE 03301

MANCHESTER, NH 031
P.M.
AUG 19
1984

USA 20c

EDMONTON USER'S GROUP
PO BOX 11983
EDMONTON, ALBERTA
CANADA T5J 3L1