



nibble tunes

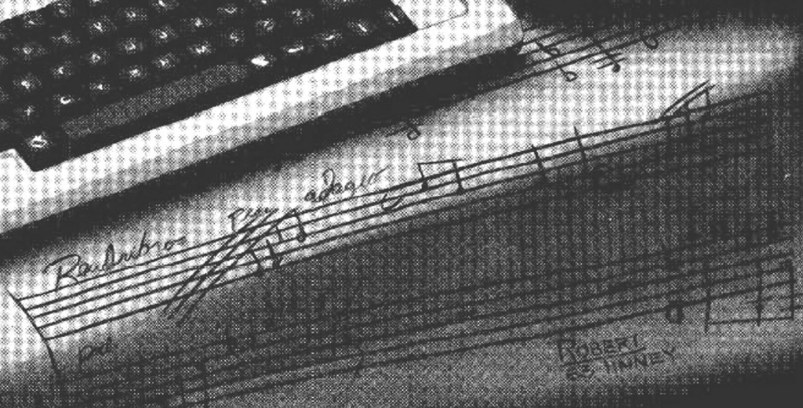
Tone Quality and Note
Encoding System



apple II



BEETHOVEN



ROBERT
E. HONEY

T.U.N.E.S. TONE UTILITY AND NOTE ENCODING SYSTEM

T.U.N.E.S. Tone Utility and Note Encoding System

by Dr. S. Scott Zimmerman
1129 East 470 North
Orem, UT 84057

INTRODUCTION

Music and sounds within Applesoft programs improve user friendliness and promote user interest. For example, in a "States & Capitals" game I wrote for my children, the program introduction contains a few bars of the **Star Spangled Banner**, a "tick-tock" sound during the questions reminds the child that the response is being timed; and correct answers are rewarded with the "charge" fanfare. In a graphics program I wrote for typing shapes from the keyboard, an audible "click" is heard after each key is pressed to simulate a typewriter; and control commands are accompanied with a characteristic two-note sequence to provide feedback that the command took effect. Thus, the user not only sees but also **hears** what is happening in the programs.

The problem, of course, is getting the music into the program. You can spend hours fiddling with POKEs and CALLs and never come out with the musical notes you want. When confronted with music having staccatos, fermatas, ritardandos and rests, you give up. The machine-language tone routine you borrowed from the local computer users group just doesn't carry the tune.

Fanfare, maestro! Here's just what you ordered! I call it **T.U.N.E.S., Tone Utility and Note Encoding System**. It is a powerful yet easy-to-use program utility designed for quick entry of any tune into your Applesoft program. It contains the flexibility to play any note within a four octave range, with a duration as short as a quick click or as long as six continuous minutes! With a little extra effort, T.U.N.E.S. also allows you to make some snazzy sound effects to jazz up your computer games.

T.U.N.E.S. TUTORIAL

Once the binary file TUNES is created (how to do this is explained later on), just BRUN TUNES, and you're ready to go. Now type an ampersand (&) followed by a <RETURN> and you will hear a short, clear tone. The ampersand calls the TUNES program, and sounds the note. Try it again: & <RETURN>. The note you hear is **middle C**, which is in **octave 3** within the T.U.N.E.S. note range. The tone is also a quarter note at a tempo of 160 beats (quarter notes) per minute. Now type & : & : & <RETURN>. (I won't mention the <RETURN> again, but of course it's always needed.) You hear four middle C quarter notes, played legato. Legato (for you nonmusicians means "smooth" — there is no break or pause between the notes. Thus, the **default** conditions for TUNES are pitch C, octave 3, quarter note, legato, 160 beats per minute.

PITCH AND OCTAVE

Now let's try other notes. Type & : &D : &E : &F : &G : &A4 : &B : &C. You have just played the C scale. The first (default tone) was mid-

dle C, the second note was D, the third E, and so forth. The different pitches are specified with the pitch parameter, in the range A-G. All of these notes are played as legato quarter notes, 160 beats/minute. Since the musical alphabet spans only A to G, we had to start over with A after G above middle C. But we wanted an A of the **next higher octave**; therefore, we had to specify a new octave, in this case 4. The final two notes, B and C, took the **new default octave 4**. To play the scale again, you will have to respecify octave 3 at the first: &C3 : &D : &E : &F : &G : &A4 : &B : &C.

The lowest note available in T.U.N.E.S. is D-sharp. Notes specified lower than this, e.g., &A1, &B1, &C1, and &D1, will still make a sound, but the sound will be a D-sharp. The highest **accurate** pitch available is &F5. Higher notes (&G5, &A6, etc.) may be specified and a tone will play, but the pitch will not be accurate. For best results, just remember to keep the octave parameter in the range 2-5.

SHARPS AND FLATS

What if the note needs a sharp or flat? Simple. The commands &C3 : &C# will play a middle C then a C-sharp. (Try it.) And the commands &E : &E! will play an E then an E-flat. (Sorry about the exclamation point, but the Apple doesn't have a "flat" symbol.)

DURATION

And what if the note needs a different duration? Again, very simple. Try this: &CE3 : &D : &E : &F : &G : &A4 : &B : &CH. Now you hear the C scale played twice as fast as before (except the last note), because the tones are eighth notes instead of quarter notes. The 'E' on the first note indicates an **Eighth** note duration. And the 'H' on the final note indicates a **Half** note. The available durations are **Whole, Half, Quarter, Eighth, Sixteenth, and Thirty-second** notes. The duration parameter, if specified, must follow immediately after the pitch parameter(s). (The octave, if specified, must come after any duration parameters.)

T.U.N.E.S. also allows you to have "dotted" notes. (The "dot" extends the duration by one half of the original value.) Thus, &CH.3 plays a middle C with a duration of a dotted half (the equivalent of three quarter notes in duration); &DQ. plays a dotted quarter (the equivalent of three eighth notes); and so forth.

And finally, you can make triplet notes. Three quarter-note triplets, for example, equal the duration of two normal quarter notes. This is encoded with a command such as &CQ' : & : &, where the apostrophe means "triplet note." To make triplet eighth notes,

continued on next page

SOURCE FILE: TUNES

```
0000: 1 *****
0000: 2 *
0000: 3 *
0000: 4 *
0000: 5 *
0000: 6 *
0000: 7 *
0000: 8 *
0000: 9 *
0000: 10 *
0000: 11 *
0000: 12 *
0000: 13 *****
0000: 14 *
```

----- NEXT OBJECT FILE NAME IS TUNES

```
9200: 15 ORG $9200 ;UP AGAINST DOS
9200: 16 *
9200: 17 *-----*
9200: 18 * ZERO PAGE VARIABLES:
9200: 19 *-----*
9200: 20 *
0000: 21 PITCH EQU $6 ;PITCH OF THE NOTES
0007: 22 DURATION EQU $7 ;NOTE DURATION
0008: 23 TEMPO EQU $8 ;TEMPO OF THE PIECE
0009: 24 OCTAVE EQU $9 ;NOTE OCTAVE (1-5)
001A: 25 DUR EQU $1A ;24-BIT DURATION DATUM
00F9: 26 DURVAR EQU $F9 ;24-BIT DURATION VARIABLE
9200: 27 *
9200: 28 *-----*
9200: 29 * MONITOR VARIABLES AND ROUTINES:
9200: 30 *-----*
9200: 31 *
0073: 32 HIMEM EQU $73 ;PROTECT PRG
00B8: 33 TXTPTR EQU $B8 ;APSOFT POINTER TO TEXT
00B1: 34 CHRGET EQU $0B1 ;APPSOFT CHARACTER GET
00B7: 35 CHRGET EQU $0B7 ;APPSOFT CHARACTER GOT
03F5: 36 AMPER EQU $3F5 ;APPSOFT AMPER LOCATION
E6F5: 37 GETBYTC EQU $E6F5 ;APSOFT RTN, EVAL EXPRN
C030: 38 SPKR EQU $C030 ;THE APPLE SPEAKER SWITCH
9200: 39 *
9200: 40 *-----*
9200: 41 * CONSTANTS:
9200: 42 *-----*
9200: 43 *
1700: 44 TCONST EQU $1700 ;TEMPO CONSTANT
0018: 45 TMIN EQU 24 ;MINIMUM TEMPO
00B3: 46 STOPTOK EQU 179 ;DEC TOKEN FOR "STOP"
00A6: 47 RESUMTOK EQU 166 ;DEC TOKEN FOR "RESUME"
00AE: 48 RESTORTK EQU 174 ;DEC TOKEN FOR "RESTORE"
9200: 49 *
```

LISTING 1

T.U.N.E.S.

TONE UTILITY AND NOTE ENCODING SYSTEM

BY S. SCOTT ZIMMERMAN

COPYRIGHT (C) 1983
MICROSPARC, INC.

three of which are equal in duration to two regular eighth notes or one regular quarter note, the command **&CE** should be used. If you want a **dotted triplet**, place the dot **before** the apostrophe; e.g., **&CQ.'** will generate a dotted quarter triplet note (whew!).

STACCATO

Now let's try some new things. Type **&CQS** : **&D** : **&E** : **&F** : **&G** : **&A4** : **&B** : **&CHL**. You recognize this as the C scale again, but with one important difference: the notes are staccato. This means that the tone duration is shortened and a rest (or pause) exists between the notes. The important thing to notice is that the tempo (160 beats/minute) remains unchanged. This is the power of T.U.N.E.S.: no longer do you need to do the calculations to make notes staccatoed; it's all automatic.

You can obtain Staccato notes simply by including an "S" in the parameter list after the pitch and duration, as shown. (If you forget to include the duration parameter and input an 'S' for Staccato, the program will think the 'S' is for a Sixteenth note and treat it as the duration parameter.) To go back to Legato, type the letter 'L' (after the pitch and duration parameters), as shown in the final note of the C scales above.

PAUSE (PORTATO)

Sometimes you may want notes which are not staccatoed but which have a slight pause between the notes; this is sometimes called portato or mezzo staccato. You can get this with the **Pause (or Portato)** command. Hence, **&F#QP** : **&** : **&** : **&** will play four F# quarter notes with a slight break between each note. Compare **&F#QP** : **&** : **&** : **&** with **&F#QL** : **&** : **&** : **&**. The Legato command turns off the Pause (as well as turning off the Staccato). If both Staccato and Pause commands are included in the parameter list, the note will be even shorter, with a concomitant longer rest (i.e., Pause and Staccato are additive).

To change from pure staccato to pure pause, do something like this: **&CQLS** : **&DQLP** : **&EQL**. The 'L' in the first note cancels out any pause or staccato that happens to be there from a previous note, and then the 'S' makes the note staccato. The 'L' on the second note cancels the staccato from the first note, and the 'P' makes it a pause note. Finally, the third note (and of course subsequent notes) will be set to legato with the final 'L'.

FERMATAS AND TIES

A fermata (hold) or a tied note is easily accomplished with T.U.N.E.S. For example, the command **&GWHW** will cause two whole notes to be tied. The 'H' following the pitch and the first duration parameters indicates a Hold, and the 'W' after the 'H' indicates a Whole note. You have to make sure that the Hold command comes **after** the pitch and the first duration parameter; otherwise, the program will think the 'H' stands for a Half note.

Okay, are you ready for a little quiz? Tell me what the command **&GISHH.3** will give. The correct answer is: pitch G-flat at octave 3 and a duration equal to the sum of a sixteenth note and a dotted half note. Notice that after the pitch and duration parameters are speci-

```

0060: 50 WHOLE EQU 96 ;DURATION NOTE VALUES
0030: 51 HALF EQU 48
0018: 52 QUARTER EQU 24
000C: 53 EIGHTH EQU 12
0006: 54 SIXTINTH EQU 6
0003: 55 THRTYSEC EQU 3
9200: 56 *
9200: 57 *-----*
9200: 58 * *
9200: 59 * START PROGRAM *
9200: 60 * *
9200: 61 *-----*
9200: 62 * SET AMPERSAND (&) POINTER AND INITIALIZE:
9200: 63 *-----*
9200: 64 *
9200:A9 4C 65 LDA #4C ;GET JMP OPCODE = 4C
9202:80 F5 83 66 STA AMPER ;STUFF AT AMPER VECTOR
9205:A9 54 67 LDA #)MAIN ;SET JUMP TO MAIN RTN
9207:8D F6 03 68 STA AMPER+1 ; FIRST LOB
920A:A9 92 69 LDA #)MAIN ; AND THEN HOB
920C:8D F7 03 70 STA AMPER+2 ;THE AMPER VECTOR HOB
920F:A9 1B 71 LDA #)DEFAULT ;SET THE HIMEM
9211:85 73 72 STA HIMEM ; TO PROTECT PROGRAM
9213:A9 92 73 LDA #)DEFAULT ;NOW DO IT FOR THE HOB
9215:85 74 74 STA HIMEM+1 ; AND HIMEM HOB
9217:20 1B 92 75 JSR DEFAULT ;SET THE DEFAULT VALUES
921A:60 76 RTS ;END OF INITIALIZATION
921B: 77 *-----*
921B: 78 *
921B: 79 * SET VARIABLE AND FLAGS TO DEFAULT VALUES:
921B: 80 *-----*
921B: 81 *
921B:A9 86 82 DEFAULT LDA #6 ;INIT PITCH TO MID C
921D:8D F6 95 83 STA PCHINDX
9220:A9 40 84 LDA #77 ;INIT THE PITCH TO MID C
9222:85 06 85 STA PITCH
9224:A9 18 86 LDA #24 ;INIT DURATION TO Q NOTE
9226:85 F9 87 STA DURVAR
9228:A9 26 88 LDA #38 ;INIT TEMPO TO ALLEGRO
922A:85 08 89 STA TEMPO ; I.E., 160 BEAT/MIN
922C:A9 03 90 LDA #3 ;INIT OCTAVE TO 3
922E:85 09 91 STA OCTAVE ; WHICH IS WHERE MID C IS
9230:20 34 92 92 JSR ZEROFLG ;ZERO THE FLAGS
9233:60 93 RTS ;END DEFAULT SUBROUTINE
9234: 94 *
9234: 95 *-----*
9234: 96 * ZERO THE FLAGS:
9234: 97 *-----*
9234: 98 *
9234:A9 00 99 ZEROFLG LDA #0 ;CLEAR FLAGS; ZERO VARS
9236:8D F2 95 100 STA STACC ; THE STACCATO FLAG
9239:8D F3 95 101 STA PAUSE ; THE PAUSE FLAG
923C:8D F4 95 102 STA REST ; THE REST FLAG
923F:8D F5 95 103 STA QUIET ; THE QUIET FLAG-NO SOUND
9242:8D F7 95 104 STA OLDNTE ; FLAG FOR PREVIOUS NOTE
9245:8D F8 95 105 STA DURSTACC ; THE STACCATO REST
9248:8D F9 95 106 STA DURSTACC+1
924B:8D FA 95 107 STA DURPAUSE ; THE PAUSE REST
924E:8D FB 95 108 STA DURPAUSE+1
9251:85 FA 109 STA DURVAR+1 ; THE DURATION HOB
9253:60 110 RTS ;END ZEROFLG SUBROUTINE
9254: 111 *
9254: 112 *-----*
9254: 113 * MAIN FOR THE PROGRAM (MAIN):
9254: 114 *-----*
9254: 115 *
9254:20 6F 92 116 MAIN JSR PARMSORT ;GO SORT THE PARAMETERS
9257:20 69 93 117 JSR SETPITCH ;THEN SET THE PITCH
925A:20 4E 94 118 JSR SETDUR ; AND THEN THE DURATION
925D:AD F5 95 119 LDA QUIET ;SHOULD NOTE BE SOUNDED?
9260:F8 06 120 BEQ PLAY ;YES, GO PLAY THE NOTE
9262:A9 08 121 LDA #0 ;CLEAR THE QUIET FLAG
9264:8D F5 95 122 STA QUIET ; FOR NEXT NOTE
9267:60 123 RTS ;NO, JUST RETURN
9268:20 24 95 124 PLAY JSR NOTE ;NOW PLAY THE NOTE(S)
926B:20 E8 94 125 JSR SILENCE ;BREAK IF STACC OR PAUSE
926E:60 126 RTS ;AND WE'RE DONE!
926F: 127 *
926F: 128 *-----*
926F: 129 * SORT THE INPUT PARAMETERS (PARMSORT):
926F: 130 *-----*
926F: 131 *
926F:20 B7 00 132 PARMSORT JSR CHRGT ;GET CHARACTER; END LINE?
9272:D0 09 133 BNE PSTOK1 ;NOT END; THERE'S A CHAR
9274:68 134 PLA ;END, POP OFF RETURN VAL
9275:68 135 PLA ; BOTH BYTES
9276:20 24 95 136 JSR NOTE ;GO PLAY THE CURRENT NOTE
9279:4C E8 94 137 JMP SILENCE ;MAKE REST THEN QUIT
927C: 138 *
927C:60 139 PSEND RTS ;END OF PARMSORT ROUTINE
927D: 140 *
927D:C9 B3 141 PSTOK1 CMP #)STOPTOK ;IS IT THE STOP TOKEN?
927F:D0 0A 142 BNE PSTOK2 ;NO, SO CHECK NEXT TOKEN
9281:A9 60 143 LDA #)60 ;GET OPCODE FOR RTS

```

fied, the Hold parameters and the octave parameter can be listed in any order.

You can also have multiple holds. Time the note **&CWHWHWHWHWHWHWH** with a stopwatch. You should get just under 10.5 seconds. Of course, you ought to be able to **calculate** the duration: The default tempo is 160 beats/minutes, which is the number of quarter notes per minute. Each whole note equals four quarter notes, and the above note has seven whole note equivalents. The note should have a duration equal to $4 \times 7 = 28$ quarter notes. The total duration, therefore, should be $(28 \text{ beats}) / (160 \text{ beats/min}) = (0.175 \text{ min}) (60 \text{ sec/min}) = 10.5$ seconds!

TEMPO

Speaking of tempo, would you like to be able to make a tune go faster or slower than the default 160 beats/minute? Look at this command: **&FQT180**. After the pitch and duration parameters are specified, a 'T' for Tempo is given, followed by an integral number equal to the desired beats per minute.

Actually, the tempo is the only command in T.U.N.E.S. which can handle an Applesoft expression, provided the expression is an integer value in the range 24-255. For example, you could use the command **TP = 20 : &FQT(200-TP)** to make the tempo 180 beats/minute. And here's how to make a ritardando (gradual slowing): **TEMPO = 160 : &CQ3TEMPO : &D : &E : &F : &GQT(TEMPO - 30) : &AQ4(TEMPO - 60) : &BQT(TEMPO - 90) : &CW**. Because it evaluates an expression, the tempo command **must** be last in the parameter lists (following at least the pitch and duration parameters).

REST

My piano teacher used to say "a rest is as important as a note." T.U.N.E.S. allows you to specify rests with the proper duration. The command **&R** will provide a rest (delay with no sound) equal in duration to the most recent note played. You can explicitly specify the duration of a quarter rest, for example, with the command **&RQ**; and a whole rest with a dotted half-note hold with an **&RWHH** command.

AUXILIARY COMMANDS

I have now covered all the major commands. There remain four auxiliary commands which are quite useful. The first command, **&X**, automatically resets all of the original defaults (middle C, quarter note, legato, 160 beats/min), without actually playing a note. The second command, **&X[parameters]**, is just like any other command except an X takes the place of a pitch parameter. This allows you to set parameters without sounding a note. For example, **&XQT120 : &C : &D** will cause the C and D to sound as quarter notes at a tempo of 120 beats per minute.

The final two auxiliary commands are **&STOP** and **&RESTORE** (or alternatively, **&RESUME**). In my programs, I like to give the user an option of having sound or silence, which I accomplish by including the following statements in my Applesoft programs:

100 PRINT "DO YOU WANT SOUND? (Y/N)" : GET A\$

120 IF A\$ = "N" THEN &STOP : REM TURN OFF SUBSEQUENT SOUND

140 IF A\$ = "Y" THEN &RESTORE : REM RESTORES SOUND

9283:8D	24	95	144	STA	NOTE	;MODIFY NOTE TO JUST RTS
9286:8D	E8	94	145	STA	SILENCE	;MODIFY SILENCE TO RTS
9289:08	14		146	BNE	PSTOK4	; (ALWAYS) LEAVE PROGRAM
928B:			147 *			
928E:C9	AE		148	PSTOK2	CMP	#RESTORTK ;IS IT THE RESTORE TOKEN?
928D:08	02		149	BNE	PSTOK3	;NO, GO CHECK NEXT TOKEN
928F:F8	84		150	BEQ	PST	;YES, GO RESTORE SOUND
9291:			151 *			
9291:C9	A6		152	PSTOK3	CMP	#RESUMTOK ;IS IT THE RESUME TOKEN?
9293:08	18		153	BNE	PS1	;NO, GO HANDLE PITCH, ETC
9295:A9	A5		154	PST	LDA	##A5 ;YES, GET LDA OPCODE TO
9297:8D	24	95	155	STA	NOTE	; RESTORE THE NOTE RTN
929A:A9	AD		156	LDA	##AD	;GET LDA OPCODE, RESTORE
929C:8D	E8	94	157	STA	SILENCE	; THE SILENCE ROUTINE
929F:			158 *			
929F:28	B1	08	159	PSTOK4	JSR	CHRGET ;INCR TXTPTR FOR EXIT
92A2:68			160	PLA		;POP OFF RETURN ADDRESS
92A3:68			161	PLA		; TO EXIT PROGRAM NOW
92A4:68			162	RTS		;RET TO BASIC CALLING RTN
92A5:			163 *			
92A5:28	0A	93	164	PS1	JSR	PCHRTN ;GO HANDLE PITCH PARMS
92A8:28	88	93	165	JSR	DURRTN	;GO HANDLE DURATION PARMS
92AB:			166 *			
92AB:28	B1	08	167	PSLOOP	JSR	CHRGET ;GET NEXT PARAM
92AE:F8	CC		168	BEQ	PSEND	;IF END OF LINE, QUIT
92B0:88	08		169	BCS	PS2	;IF NOT NUMBER, CHECK 'S'
92B2:38			170	SEC		;CONVERT ASCII TO VALUE
92B3:E9	38		171	SBC	#'--\$80	
92B5:95	87		172	STA	OCTAVE	;YES, SET THE VALUE
92B7:4C	AE	92	173	JMP	PSLOOP	;GO BACK, CHECK NEXT PARAM
92BA:			174 *			
92BA:C9	53		175	PS2	CMP	#'S'-\$80 ;IS NOTE A STACCATO?
92BC:08	08		176	BNE	PS3	;NO, GO TO NEXT
92BE:A9	81		177	LDA	#1	;YES, SET STACCATO FLAG
92C8:8D	F2	95	178	STA	STACC	;NOW IT'S A STACCATO
92C3:4C	AE	92	179	JMP	PSLOOP	;GO BACK, CHECK NEXT PARAM
92C6:			180 *			
92C6:C9	58		181	PS3	CMP	#'P'-\$80 ;IS NOTE A PAUSE?
92C8:08	08		182	BNE	PS4	;NO, GO TO NEXT
92CA:A9	81		183	LDA	#1	;YES, SET THE PAUSE FLAG
92CC:8D	F3	95	184	STA	PAUSE	;NOW IT'S SET TO PAUSE
92CF:4C	AE	92	185	JMP	PSLOOP	;GO BACK, CHECK NEXT PARAM
92D2:			186 *			
92D2:C9	54		187	PS4	CMP	#'T'-\$80 ;IS TEMPO TO BE RESET?
92D4:08	0F		188	BNE	PS6	;NO, GO TO NEXT
92D6:28	F5	E6	189	JSR	GETBYTC	;EVALUATE TEMPO EXPRSN
92D9:E8	18		190	CPX	#TMIN	;IS TEMPO LESS THAN TMIN?
92DB:88	02		191	BCS	PS5	;NO, SO GO SET IT
92DD:A2	18		192	LDX	#TMIN	;SO SET IT TO TMIN
92DF:84	08		193	PS5	STX	TEMPO ;STORE THE TEMPO VALUE
92E1:28	A8	94	194	JSR	TEMPCALC	;CALCULATE ACTUAL TEMPO
92E4:68			195	RTS		;TEMPO MUST BE LAST PARAM
92E5:			196 *			
92E5:C9	48		197	PS6	CMP	#'H'-\$80 ;IS NOTE TO BE LONGER?
92E7:08	66		198	BNE	PS7	;NO, GO TO NEXT
92E9:28	39	94	199	JSR	SETHOLD	;GO SET THE HOLD DURATION
92EC:4C	AE	92	200	JMP	PSLOOP	;GO BACK, CHECK NEXT PARAM
92EF:			201 *			
92EF:C9	4C		202	PS7	CMP	#'L'-\$80 ;IS IT LEGATO (SMOOTH)?
92F1:08	14		203	BNE	PS8	;NO, GO ON TO NEXT
92F3:A9	88		204	LDA	#8	;YES; CLEAR THE
92F5:8D	F2	95	205	STA	STACC	; STACCATO FLAG
92F8:8D	F3	95	206	STA	PAUSE	; THE PAUSE FLAG
92FB:8D	F8	95	207	STA	DURSTACC	;CLEAR DURATION VALUES
92FE:8D	F9	95	208	STA	DURSTACC+1	
9301:8D	FA	95	209	STA	DURPAUSE	
9304:8D	FB	95	210	STA	DURPAUSE+1	
9307:			211 *			
9307:4C	AE	92	212	PS8	JMP	PSLOOP ;GO BACK, CHECK NEXT PARAM
930A:			213 *			
930A:			214 *			
930A:			215 *			-----*
930A:			216 *			-----*
930A:			217 *			
930A:C9	52		218	PCHRTN	CMP	#'R'-\$80 ;IS NOTE A REST?
930C:08	06		219	BNE	XCHECK	;NO, GO CHECK IF AN X
930E:A9	81		220	LDA	#1	;SET THE REST FLAG
9318:8D	F4	95	221	STA	REST	
9313:68			222	RTS		;NO NOTE CALL NEEDED
9314:			223 *			
9314:C9	58		224	XCHECK	CMP	#'X'-\$80 ;SHOULD IT BE QUIET?
9316:08	18		225	BNE	CALCPCH	;NO, GO CALC THE PITCH
9318:28	B1	08	226	JSR	CHRGET	;IS THERE ANOTHER CHAR?
931B:08	08		227	BNE	QF	;YES, GO SET QUIET FLAG
931D:28	18	92	228	JSR	DEFAULT	;NO, SET DEFAULT VALUES
9320:68			229	PLA		;NOW POP OF ALL RETURNS
9321:68			230	PLA		
9322:68			231	PLA		; SO THE PROGRAM WILL
9323:68			232	PLA		; TERMINATE HERE
9324:68			233	RTS		;DONE.
9325:			234 *			
9325:A9	81		235	QF	LDA	#1 ;SET THE QUIET FLAG
9327:8D	F5	95	236	STA	QUIET	

With the appropriate coding in your program, you can allow the user to turn off (&STOP) and turn back on (&RESTORE or &RESUME) the sound generated by the T.U.N.E.S. routine

SUMMARY OF COMMANDS

Here is a summary of the commands available with T.U.N.E.S.

1. The general command, with the parameter options, is

& [pitch: A - G [#][!]] [R]est];duration: W,H,Q,E,S,T [.]["] [octave: 1-6] [L]egato] [S]taccato [P]ause [H]old [duration]; [T]empo [aexpr 24 - 255];

where any parameter type not specified defaults to the most recently specified one (or, if never specified, to the original defaults; viz., middle C, quarter note, legato, 160 beats/minute).

The various parameters are defined as follows:

a. The letters A-G represent the eight primary pitches of the musical scale. If any note parameter is specified, the pitch must be the first parameter after the ampersand.

b. Immediately following the letter for the pitch, a '#' means "sharp" and '!' means "flat." Only one of these may be specified in a single & command.

c. R means "rest." It must be the first parameter in the & command.

d. After the pitch or rest parameter(s), W = whole, H = half, Q = quarter, E = eighth, S = sixteenth, and T = thirty-second for the duration of the note. The duration parameter, if specified, must always immediately follow the pitch parameter.

e. A dot (.) immediately following the duration parameter modifies the duration to give a dotted note.

f. An apostrophe (') immediately following the duration parameter modifies the duration to give triplet notes. The apostrophe must come after the dot if both are specified.

g. An integer number in the range 1-6 indicates the octave. Each octave spans the notes starting with A-flat and ending with G-sharp. For accurate pitches, use only the octave range 2-5. The octave parameter may be given anywhere after the pitch and duration parameters, unless no duration parameter is specified, in which case the octave number is listed immediately after the pitch parameter.

The following commands, if specified, must come after the pitch parameter(s) and duration parameter(s), but in no particular order, except that the Tempo parameter must come last.

h. An L sets the Duration mode to legato (no break between the notes). S sets the mode to staccato. (The tone duration is cut in half, and a silent pause, equal to the new duration, is made after the note.) P sets the mode to pause or

932A:A5 B8	237	LDA	TXTPTR	;THE TXTPTR MUST BE DECR	
932C:D0 02	238	BNE	QF1	; TO KEEP PARAMETER LIST	
932E:C6 B9	239	DEC	TXTPTR+1	; IN SEQUENCE WITHOUT	
9330:C6 B8	240	DEC	TXTPTR	; SKIPPING ANYTHING	
9332:60	241	RTS		;NO NOTE CALC NEEDED	
9333:	242	*			
9333:38	243	CALCPCPH	SEC	;PREPARE FOR SUBTRACTION	
9334:E7 40	244	SBC	#'Q'-\$80	;MAKE A=1, B=2, ETC.	
9336:0A	245	ASL	A	;MULT BY 2	
9337:AA	246	TAX		;TRANSFER PITCH TO INDEX	
9338:A9 08	247	LDA	#0	;NOT A REST, SO ZERO	
933A:8D F4 95	248	STA	REST	; THE REST FLAG	
933D:20 B1 08	249	JSR	CHRGET	;IS THERE A # OR ' ?	
9340:C9 23	250	CMP	#'Q'-\$80	;IS NOTE A SHARP?	
9342:D0 11	251	BNE	PR1	;NO, CONTINUE	
9344:E8	252	INX		;YES, SO GO UP ONE NOTE	
9345:E0 05	253	CPX	#5	;SHOULD NOTE BE #? ?	
9347:D0 04	254	BNE	PRE	;NO, SO GO SEE IF #	
9349:E8	255	INX		;YES, SO INCR INDEX TO C	
934A:4C 65 93	256	JMP	PSETNDX	;GO SET THE INDEX	
934D:	257	*			
934D:E0 08	258	PRE	CPX	#11	;SHOULD NOTE BE AN E#?
934F:D0 14	259	BNE	PSETNDX		;NO, SO ALL IS OKAY AS IS
9351:E8	260	INX			;YES, SO INCREMENT TO F
9352:4C 65 93	261	JMP	PSETNDX		;GO SET THE PITCH INDEX
9355:	262	*			
9355:C9 21	263	PR1	CMP	#'!'-\$80	; IS IT FLAT (EXCLAM INT)?
9357:D0 04	264	BNE	PR2		;NO, SO SET PITCH INDEX
9359:CA	265	DEX			;YES, SO GO DOWN ONE NOTE
935A:4C 65 93	266	JMP	PSETNDX		
935D:	267	*			
935D:A5 B8	268	PR2	LDA	TXTPTR	;SINCE CHAR IS NOT # OR !
935F:D0 02	269	BNE	PR3		; DECREMENT TEXT POINTER
9361:C6 B9	270	DEC	TXTPTR+1		
9363:C6 B8	271	PR3	DEC	TXTPTR	
9365:	272	*			
9365:8E F6 95	273	PSETNDX	STX	PCHINDEX	;AND STORE IT
9368:60	274	RTS			;END OF PCHRTN ROUTINE
9369:	275	*			
9369:	276				
9369:	277	*			-----*
9369:	278	*			SET THE PITCH (SETPITCH);
9369:	279	*			-----*
9369:AD F4 95	280	SETPITCH	LDA	REST	;IS THE NOTE A REST?
936C:F0 05	281	BEQ	SP0		;NO, GO SET PITCH
936E:A9 00	282	LDA	#0		;SET THE PITCH TO ZERO
9370:4C 85 93	283	JMP	SP3		
9373:	284	*			
9373:A6 09	285	SP0	LDX	OCTAVE	;GET OCTAVE OF THE NOTE
9375:AD F6 95	286	LDA	PCHINDEX		;GET THE PITCH INDEX
9378:	287	*			
9378:CA	288	SP1	DEX		;DONE ADDING 1? ?
9379:F0 06	289	BEQ	SP2		;YES, GO GET PITCH
937B:18	290	CLC			;PREPARE FOR ADDITION
937C:69 0E	291	ADC	#14		;14 NOTES/OCTAVE IN TABLE
937E:4C 78 93	292	JMP	SP1		;GO BACK
9381:	293	*			
9381:AA	294	SP2	TAX		;TRANSFER VALUE TO INDEX
9382:BD 9C 95	295	LDA	PITCHTAB,X		;GET THE NOTE VALUE
9385:85 06	296	SP3	STA	PITCH	;AND WE HAVE THE PITCH
9387:60	297	RTS			
9388:	298	*			
9388:	299	*			-----*
9388:	300	*			SET THE DURATION (DURRTN);
9388:	301	*			-----*
9388:	302	*			
9388:20 B1 08	303	DURRTN	JSR	CHRGET	;GET THE DURATION VALUE
938B:C9 57	304	CMP	#'W'-\$80		;WHOLE NOTE?
938D:D0 07	305	BNE	DR1		;NO, CHECK NEXT
938F:A9 60	306	LDA	#WHOLE		;SET WHOLE NOTE
9391:85 07	307	STA	DURATION		
9393:4C D6 93	308	JMP	DR8		
9396:	309	*			
9396:C9 48	310	DR1	CMP	#'H'-\$80	;HALF NOTE?
9398:D0 07	311	BNE	DR2		
939A:A9 30	312	LDA	#HALF		
939C:85 07	313	STA	DURATION		
939E:4C D6 93	314	JMP	DR8		
93A1:	315	*			
93A1:C9 51	316	DR2	CMP	#'Q'-\$80	;QUARTER NOTE?
93A3:D0 07	317	BNE	DR3		
93A5:A9 18	318	LDA	#QUARTER		
93A7:85 07	319	STA	DURATION		
93A9:4C D6 93	320	JMP	DR8		
93AC:	321	*			
93AC:C9 45	322	DR3	CMP	#'E'-\$80	;EIGHTH NOTE?
93AE:D0 07	323	BNE	DR4		
93B0:A9 0C	324	LDA	#EIGHTH		
93B2:85 07	325	STA	DURATION		
93B4:4C D6 93	326	JMP	DR8		
93B7:	327	*			
93B7:C9 53	328	DR4	CMP	#'S'-\$80	;SIXTEENTH NOTE?
93B9:D0 07	329	BNE	DR5		
93BB:A9 06	330	LDA	#SIXTNTH		
93BD:85 07	331	STA	DURATION		

portato. (The duration is slightly decreased with a short pause after the note.) The S and P parameters are additive; i.e., if both are specified in a single note, the tone duration will be shortened by both commands, and the pause after the note will be lengthened by both.

i. An H indicates a hold note. Immediately following the H, a duration parameter (W, H, Q, E, S, or T, with an optional dot and/or apostrophe) must be given. A maximum of 116 sets of holds can be given in a single note, although in actual practice, encoding more than three is rare.

j. As the **last** parameter specified, T sets the tempo to the number of beats per minute, equal to the Applesoft expression which follows the T. The expression should give an integer value in the range 24 (the slowest tempo) to 255 (the fastest tempo).

2. The following auxiliary commands are available with T.U.N.E.S. They are called "auxiliary" because they do not encode for a tone.

a. The command **&X** resets all the parameters to the original defaults — viz., middle C, quarter note, legato, 160 beats/minute — without playing a tone.

b. The command **&X[parameters]**, where the parameters can be any of those listed above except the pitch parameters, sets the specified parameters but makes no sound.

c. **&STOP** silences all subsequent & commands. The program behaves much as if the & commands of T.U.N.E.S. did not exist.

d. **&RESUME** or **&RESTORE** causes the sound from the & commands to resume.

3. To obtain pitches and durations not accessible from the normal parameters, you can do the following POKES:

a. **POKE 6,n** where n is a pitch in the range 1-255. Higher values of n give lower pitches. Thus, the lowest pitch available is given by **POKE 6,255** : &. Be sure to **not** include any parameters with the & command, or the POKES will be overridden with the specified pitch.

b. **POKE 249,i** : **POKE 250,j** where the duration is equal to $i + j * 256$, in the range 1-65535. Higher values of i and j give longer durations. The value of i, which is the low-order byte of a two-byte value, must be a number in the range 1-255; and j, the high-order byte, must be in the range 0-255.

c. **POKE 8,m** where m is the tempo in the range 1-255. A higher value of m gives a slower tempo. Thus, **POKE 8,1** : & is the fastest tempo possible. (In fact for many pitches, it is so fast that no sound will be made.)

Just to give you some idea of the range of possible durations from POKeing in the tempo and duration, if you **POKE 8,1** (the fastest tempo) and **POKE 249,1** : **POKE 250,0** (the shortest duration), the tone will be nothing more than a faint click, if any

```

93BF:4C D6 93 332 JMP DR8
93C2: 333 *
93C2:C9 54 334 DR5 CMP #'T'-'$80 ;THIRTY-SECOND NOTE?
93C4:D0 07 335 BNE DR6
93C6:A9 03 336 LDA #THRTYSEC
93C8:85 07 337 STA DURATION
93CA:4C D6 93 338 JMP DR8
93CD: 339 *
93CD:A5 B8 340 DR6 LDA TXTPTR ;DECR THE PRGM POINTER
93CF:D0 02 341 BNE DR7
93D1:C6 B9 342 DEC TXTPTR+1
93D3:C6 B8 343 DR7 DEC TXTPTR
93D5:60 344 RTS
93D6: 345 *
93D6:A9 00 346 DR8 LDA #0 ;REINIT DURATION HOB
93D8:85 FA 347 STA DURVAR+1
93DA:A9 00 348 LDA #0 ;CLEAR THE OLD NOTE FLAG
93DC:8D F7 95 349 STA OLDNOTE ; SINCE IT'S A NEW NOTE
93DF:A5 07 350 LDA DURATION ;GET THE NEW DURATION
93E1:85 F9 351 STA DURVAR ; AND STORE IT
93E3: 352 *
93E3:20 B1 00 353 JSR CHRGET ;CHECK IF DOTTED NOTE
93E6:C9 2E 354 CMP #'.'-'$80 ;IS IT A DOT?
93E8:D0 13 355 BNE DR10 ;NO, GO CHECK IF TRIPLET
93EA:A5 07 356 LDA DURATION ;GET THE OLD DURATION
93EC:8D FC 95 357 STA TEMPDAT ;STORE IT TEMPORARILY
93EF:4A 358 LSR A ;DIVIDE BY TWO TO GET THE
93F0:18 359 CLC ; ADDITNL TIME, THEN ADD
93F1:6D FC 95 360 ADC TEMPDAT ; TO THE OLD DURATION
93F4:85 F9 361 STA DURVAR ; TO THE NEW DURATION
93F6:90 02 362 BCC DR9 ;DO WE NEED ANOTHER BYTE?
93F8:E6 F9 363 INC DURVAR ;YES, INCREMENT IT
93FA:20 B1 00 364 DR9 JSR CHRGET ;GET NEXT CHAR
93FD: 365 *
93FD:C9 27 366 DR10 CMP #'/'-'$80 ;IS IT TRIPLET (' / ')?
93FF:D0 CC 367 BNE DR6 ;NO, GO DECREMENT TXTPTR
9401:06 F9 368 ASL DURVAR ;SHIFT TO MULTIPLY BY 2
9403:26 FA 369 ROL DURVAR+1 ;PUT CARRY IN DURTN HOB
9405: 370 *
9405: 371 * DIVIDE THE NEW DURATION BY THREE:
9405: 372 *
9405:A9 00 373 LDA #0 ;INIT PARTIAL ANSWER
9407:8D FC 95 374 STA TEMPDAT
940A:8D FD 95 375 STA TEMPDAT+1
940D:A9 03 376 LDA #3 ;LOAD THE DIVISOR (3)
940F:8D FE 95 377 STA TEMPENST
9412: 378 *
9412:A0 10 379 LDY #16 ;SIXTEEN-BIT DIVISION
9414:06 F9 380 TRDIV1 ASL DURVAR
9416:26 FA 381 ROL DURVAR+1
9418:2E FC 95 382 ROL TEMPDAT
941B:2E FD 95 383 ROL TEMPDAT+1
941E:38 384 SEC
941F:A0 FC 95 385 LDA TEMPDAT
9422:ED FE 95 386 SBC TEMPENST
9425:AA 387 TAX
9426:A0 FD 95 388 LDA TEMPDAT+1
9429:E9 00 389 SBC #0
942B:90 08 390 BCC TRDIV2
942D:8E FC 95 391 STX TEMPDAT
9430:8D FD 95 392 STA TEMPDAT+1
9433:E6 F9 393 INC DURVAR
9435:88 394 TRDIV2 DEY
9436:D0 DC 395 BNE TRDIV1
9438:60 396 RTS ;END OF DURRTN
9439: 397 *
9439: 398 *-----*
9439: 399 * SET THE DURATION IF THERE'S A HOLD (SETHOLD):
9439: 400 *-----*
9439: 401 *
9439:A5 FA 402 SETHOLD LDA DURVAR+1 ;GET THE OLD DURATION
943B:48 403 PHA ; AND STORE FOR LATER USE
943C:A5 F9 404 LDA DURVAR ;LOB SECOND
943E:48 405 PHA
943F:20 88 93 406 JSR DURRTN ;CHECK HOLD DURATN VALUE
9442:18 407 CLC ;PREP TO ADD DURATION
9443:68 408 PLA ;GET OLD DURATION BACK
9444:65 F9 409 ADC DURVAR ;ADD THE LOB
9446:85 F9 410 STA DURVAR ;SAVE NEW DURATION
9448:68 411 PLA ;GET THE OLD HOB
9449:65 FA 412 ADC DURVAR+1 ;ADD THE HOB
944B:85 FA 413 STA DURVAR+1 ;SAVE IT
944D:60 414 RTS ;END OF SETHOLD ROUTINE
944E: 415 *
944E: 416 *-----*
944E: 417 * SET THE DURATION (SETDUR):
944E: 418 *-----*
944E: 419 *
944E:A0 F7 95 420 SETDUR LDA OLDNOTE ;IS THIS AN OLD NOTE?
9451:D0 4F 421 BNE DLOND ;YES, SO NO CHANGE NEEDED
9453:A0 F2 95 422 LDA STACC ;IS STACCATO FLAG SET?
9456:F0 0E 423 BEQ CHKPAUSE ;NO, SO CHECK PAUSE
9458:46 FA 424 LSR DURVAR+1 ;DIVIDE DURATION BY TWO
945A:66 F9 425 ROR DURVAR

```

sound is made at all. At the other extreme, if you **POKE 8,255** (the slowest tempo) and **POKE 249,255** : **POKE 250,255** (the longest duration), and then play the note ('&' with no parameters), the tone will last for 6 minutes and 50 seconds!

4. To amplify the sound of your T.U.N.E.S. music or sound effects, you can route the sound from the cassette output port (on the back of your Apple) into the amplifier of your stereo system, or into the audio port of your monitor (if your monitor has a speaker). You will need to modify the T.U.N.E.S. program to access the "cassette out" memory location (-16352 = SC020) rather than the speaker location (-16336 = SC030). This requires two POKES: **POKE 38269,32** and **POKE 38295,32**.

DEMO PROGRAMS

Listing 2 shows an Applesoft program demonstrating the power and flexibility of T.U.N.E.S. The musical example is **Musetta's Waltz Song** from **La Bohème** by Puccini. The printed music for this piece is shown in **Figure 1**.

This piece was chosen because it has a great diversity of notes: legato, staccato, pause, grace notes, holds, ritardandos, triplets, rests, etc. You can learn much about how to enter T.U.N.E.S. music into Applesoft programs by comparing the written music with the program listing.

Listing 3 gives an Applesoft program of Bach's **Invential VIII**. It shows how easily fast-paced music (in this case **vivace**) is handled by T.U.N.E.S.

Listing 4 is a demonstration of various sound effects possible with T.U.N.E.S. Actually, T.U.N.E.S. was not written as a sound effects program, but with a little imagination and luck, you can come up with some other interesting sound effects for your computer games.

HOW TO LOAD T.U.N.E.S.

T.U.N.E.S. is an assembly language program (shown in **Listing 1**). While the program was written with the BIG MAC assembler from the Apple PugetSound Program Library Exchange (A.P.P.L.E.), the listing shown was made with the DOS Tool Kit Editor/Assembler. Specialized assembler features, such as macros, were not used in T.U.N.E.S. so that the code would be as compatible as possible with the many different assemblers used by Nibble readers.

The Letters section at the front of this issue contains an article on "How to Enter Assembly/Machine Language Programs in Nibble." This provides most of the information you need to type in the program if you don't have an assembler. After you have entered all the machine code as instructed, type **BSAVE TUNES,AS9200,LS400** to save the binary program to disk.

945C:A5 F9	426	LDA DURVAR	;SAVE THE NEW DURATION
945E:8D F8 95	427	STA DURSTACC	; IN STACC REST DURATION
9461:A5 FA	428	LDA DURVAR+1	
9463:8D F9 95	429	STA DURSTACC+1	
9466:	430 *		
9466:AD F3 95	431	CHKPAUSE LDA PAUSE	;IS PAUSE FLAG SET?
9469:F0 37	432	BEQ SDEND	;NO, NO CHANGE NEEDED
946B:A5 F9	433	LDA DURVAR	;GET PRST DURATION VAR
946D:8D FC 95	434	STA TEMPDAT	;STORE FOR MANIPULATION
9470:A5 FA	435	LDA DURVAR+1	
9472:8D FD 95	436	STA TEMPDAT+1	
9475:4E FD 95	437	LSR TEMPDAT+1	;DIVIDE IT BY 8
9478:6E FC 95	438	ROR TEMPDAT	; BY DOING THREE
947B:4E FD 95	439	LSR TEMPDAT+1	; 16-BIT RIGHT ROTATIONS
947E:6E FC 95	440	ROR TEMPDAT	
9481:4E FD 95	441	LSR TEMPDAT+1	
9484:6E FC 95	442	ROR TEMPDAT	
9487:38	443	SEC	;PREPARE FOR SUBTRACTION
9488:A5 F9	444	LDA DURVAR	;MAKE THE DURATION 7/8
948A:ED FC 95	445	SBC TEMPDAT	; OF WHAT IT WAS
948D:85 F9	446	STA DURVAR	
948F:A5 FA	447	LDA DURVAR+1	; AND THE PAUSE 1/8 OF
9491:ED FD 95	448	SBC TEMPDAT+1	; ORIGINAL DURATION
9494:85 FA	449	STA DURVAR+1	
9496:AD FC 95	450	LDA TEMPDAT	;GET THE PAUSE LENGTH
9499:8D FA 95	451	STA DURPAUSE	;AND SAVE FOR LATER PAUSE
949C:AD FD 95	452	LDA TEMPDAT+1	
949F:8D FB 95	453	STA DURPAUSE+1	
94A2:	454 *		
94A2:A9 01	455	SDEND LDA #1	;SET OLD NOTE FLAG
94A4:8D F7 95	456	STA OLDNOTE	
94A7:60	457	RTS	;END OF SETOUR ROUTINE
94A8:	458 *		
94A8:	459 *		
94A8:	460 *	CALCULATE TEMPO (BEATS/MIN) (TEMPCALC):	
94A8:	461 *		
94A8:	462 *		
94A8:A9 00	463	TEMPCALC LDA #0	;INIT PARTIAL ANSWER
94AA:8D FC 95	464	STA TEMPDAT	
94AD:8D FD 95	465	STA TEMPDAT+1	
94B0:A9 80	466	LDA #>CONST	;LOAD THE TEMPO CONSTANT
94B2:8D FE 95	467	STA TEMPCNST	; AND SAVE FOR USE
94B5:A9 17	468	LDA #<CONST	

FIGURE 1

MUSETTA'S WALTZ SONG (from La Bohème)

Allegro G. Puccini

TUNES CODE: &X &REATX &B &D &G &F#S' &G &F# &DE &B &J3
FOOTNOTES: (1) (2) (3) (4) (5)

&F#T &G &F#S &EES &G &D &G# &C# &AE4L &R
(6) (7) (8) (9) (10)

&RH &AES &R &RQ &AE &R &RQ &AE &R &RQ

Tempo di valse lento

&DH4LT(X-50) &DS3 &F# &A4 &D &C#11 &DS3 &F# &A4 &C#
(11)

If you have the BIG MAC assembler, you can enter the assembly code as shown in **Listing 1, with the following modifications:**

1. Change the operand labels that begin with #> to #<, and change #< to #>. For example, in the initialization section (just after "START PROGRAM"), the command LDA #>MAIN should be changed to LDA #<MAIN; and LDA #<MAIN should be changed to LDA #>MAIN. (I can't imagine how assemblers can possibly use opposite conventions, but they do.) Many other assemblers use a completely different convention; in one assembler the low order byte is simply #, and the high order byte is #/. Check your assembler's documentation for the proper expression.

2. Change ASCII labels such as #N'-S80 to #N'. These expressions are prevalent throughout PARMSORT, PCHRTN, and DURRTN subroutines.

3. The mnemonics LSR and ASL in BIG MAC do not require a label. Therefore, program line numbers 245 and 358 should read **ASL** and **LSR**, respectively.

If you have some other assembler, you are on your own. The best advice for adapting the T.U.N.E.S. source code to your assembler is, first, to know the proper conventions for your assembler; and second, to watch your assembled (object) code to make sure that it is the same as that given in the left hand margin of **Listing 1**.

After you have assembled the program, SAVE it as the binary file **TUNES**. The origin address is \$9200 (decimal 37316) and the length is \$400 (decimal 1024). This is high in memory, up against DOS in the 48K Apple. If you have a computer with less memory, or if your Applesoft program sets MAXFILES greater than 3, you will have to assemble TUNES at an appropriately lower memory address.

HOW THE PROGRAM WORKS

When you BRUN TUNES, three things happen:

1. The &-vector at \$3F5 (decimal 1013) is set so that Applesoft will jump to the MAIN routine of TUNES when an '&' is encountered.
2. HIMEM is set below the TUNES routine to protect it from Applesoft string variables, which normally occupy that region of memory.
3. The TUNES default values are set and the various flags are zeroed.

The MAIN routine has the major function of calling the various subroutines for setting up the note values (pitch, duration, etc.) and then calling the tone routine itself. The first and principal subroutine called by MAIN is PARMSORT, which sorts the input parameters. PARMSORT and the subroutines which it calls use two important ROM Applesoft routines, CHRGET and CHRGOT. Both of these routines use the Applesoft vector TXTPTR, located at \$D0 and \$B9 (decimal 184 and 185). TXTPTR points at the next character or instruction of the Applesoft program. Thus, when an '&' is encountered and

```

94B7:8D FF 95 469 STA TEMPCNST+1
94BA: 470 *
94BA: 471 *
94BA: 472 * DIVIDE: TEMPO = TEMPCNST/TEMPO
94BA: 473 *
94BA:A0 10 474 LDY #16 ;SIXTEEN-BIT DIVISION
94BC:0E FE 95 475 TC1 ASL TEMPCNST
94BF:2E FF 95 476 ROL TEMPCNST+1
94C2:2E FC 95 477 ROL TEMPDAT
94C5:2E FD 95 478 ROL TEMPDAT+1
94C8:38 479 SEC
94C9:AD FC 95 480 LDA TEMPDAT
94CC:E5 08 481 SBC TEMPO
94CE:AA 482 TAX
94CF:AD FD 95 483 LDA TEMPDAT+1
94D2:E9 00 484 SBC #0
94D4:90 09 485 BCC TC2
94D6:0E FC 95 486 STX TEMPDAT
94D9:8D FD 95 487 STA TEMPDAT+1
94DC:EE FE 95 488 INC TEMPCNST
94DF:88 489 TC2 DEY
94E0:08 DA 490 BNE TC1
94E2: 491 *
94E2:AD FE 95 492 LDA TEMPCNST ;PUT THE ANSWER INTO THE
94E5:85 08 493 STA TEMPO ;NEW TEMPO VALUE
94E7:60 494 RTS ;END OF TEMPCALC ROUTINE
94E8: 495 *
94E8: 496 *
94E8: 497 * SET 'REST' FOR STACCATO AND/OR PAUSE (SILENCE):
94E8: 498 *
94E8: 499 *
94E8:AD F2 95 500 SILENCE LDA STACC ;IS IT A STACCATO NOTE?
94ED:0D F3 95 501 ORA PAUSE ;OR A PAUSE?
94EE:00 01 502 BNE SPAUSE ;YES, SO GO MAKE PAUSE(S)
94F0:60 503 RTS ;RETURN WITH NO PAUSE
94F1: 504 *
94F1:A5 FA 505 SPAUSE LDA DURVAR+1 ;SAVE ORIGINAL DURATION
94F3:48 506 PHA
94F4:A5 F9 507 LDA DURVAR ;LOB AND HOB
94F6:48 508 PHA
94F7:AD F8 95 509 LDA DURSTACC ;GET REST DURTN FOR STACC
94FA:85 F9 510 STA DURVAR ;MAKE IT PRESENT DURATION
94FC:AD F9 95 511 LDA DURSTACC+1
94FF:85 FA 512 STA DURVAR+1
9501:18 513 CLC ;NOW ADD IN PAUSE DURTN
9502:A5 F9 514 LDA DURVAR
9504:60 FA 95 515 ADC DURFAUSE
9507:85 F9 516 STA DURVAR
9509:A5 FA 517 LDA DURVAR+1
950B:60 FB 95 518 ADC DURFAUSE+1
950E:85 FA 519 STA DURVAR+1
9510:A5 06 520 LDA PITCH ;GET THE PRESENT PITCH
9512:48 521 PHA ;SAVE IT
9513:A9 00 522 LDA #0 ;ZERO PITCH TO MAKE REST
9515:85 06 523 STA PITCH
9517: 524 *
9517:20 24 95 525 JSR NOTE ;GO 'UNFOUND' THE PAUSE
951A: 526 *
951A:68 527 PLA ;GET BACK THE OLD PARMS
951B:85 06 528 STA PITCH
951D:68 529 PLA
951E:85 F9 530 STA DURVAR
9520:68 531 PLA
9521:85 FA 532 STA DURVAR+1
9523:60 533 RTS ;END OF SILENCE ROUTINE
9524: 534 *
9524: 535 *
9524: 536 * PLAY THE NOTE (NOTE):
9524: 537 *
9524: 538 *
9524:A5 FA 539 NOTE LDA DURVAR+1 ;SAVE DURATION VARIABLE
9526:48 540 PHA ;FIRST THE HOB
9527:A5 F9 541 LDA DURVAR ;THEN THE LOB
9529:48 542 PHA
952A: 543 *
952A: 544 * MULTIPLY DURVAR BY 16
952A: 545 *
952A:A2 04 546 LDX #4 ;MAKE FOUR SHIFTS
952C:06 F9 547 MULT ASL DURVAR ;SHIFT LEFT LSB
952E:26 FA 548 ROL DURVAR+1 ;SHIFT LEFT MSB W/ CARRY
9530:CA 549 DEX
9531:D0 F9 550 BNE MULT
9533: 551 *
9533: 552 * MULTIPLY DURVAR BY TEMPO AND SAVE IN DUR:
9533: 553 *
9533:A5 08 554 LDA TEMPO ;SAVE THE TEMPO
9535:48 555 PHA
9536:A9 00 556 LDA #0 ;INITIALIZE TO ZERO
9538:85 1A 557 STA DUR
953A:85 1E 558 STA DUR+1
953C:85 1C 559 STA DUR+2
953E:85 FB 560 STA DURVAR+2 ;FIRST TWO BYTES DONE
9540:A0 08 561 LDY #8 ;MULTIPLIER IS 8 BIT

```


the program jumps to TUNES, TXTPTR is pointing at the character after the ampersand.

The routine CHRGET, located at \$00B1 (decimal 177) does three things:

1. It increments TXTPTR.
2. It loads the accumulator with the character being pointed at by TXTPTR.
3. It sets the Zero flag of the Processor Status Register if TXTPTR points at the end of a line, and clears the Carry flag if the accumulator holds an ASCII number (0-9).

The routine CHRGOT, located at \$00B7 (decimal 183), carries out the same function as CHRGET but does **not** increment TXTPTR; i.e., CHRGOT loads the accumulator with the character already being pointed at by TXTPTR.

The T.U.N.E.S. subroutines use CHRGET and CHRGOT to get each parameter, one at a time, and then appropriately modify the variables (pitch, duration, octave, etc.) or set the various flags (Staccato flag, Rest flag, etc.).

Another Applesoft ROM routine used by T.U.N.E.S. is GETBYTC located at \$E6F5 (decimal 59125). Its general function is to evaluate the Applesoft expression pointed at by TXTPTR. GETBYTC is used in PARM-SORT after a 'T' is encountered to evaluate the expression for the tempo.

The heart of T.U.N.E.S. is the NOTE subroutine, which plays the note encoded in the & parameter list. NOTE first multiplies the duration variable (DURVAR) by 16 to allow for longer notes, and then multiplies that value by the TEMPO. The end result is that DURVAR is a 24-bit number, able to equal any values from 0 to 16,777,215. This accounts for the wide range of possible durations. The sound loop always makes DURVAR passes, whether or not the speaker is toggled. The PITCH variable dictates how often the speaker is toggled during each pass, thereby determining the frequency (pitch) of the sound. Thus, pitch and duration are kept independent, greatly simplifying the evaluation of the variables needed for the desired note.

I have tried as much as possible to keep the various functions of T.U.N.E.S. separated into short modules (subroutines), and to explain the function of each module with comments in the source listing. Hopefully, this will help you understand the code if you are an assembly language programmer.

LIMITATIONS OF T.U.N.E.S.

Now that I have touted T.U.N.E.S. as being powerful and easy to use, I will warn you of some of its limitations:

1. Although the tone pitches are as closely in tune as possible within the tone routine algorithm, there are limits imposed by the 8-bit frequency variable. Some of the pitches will bother you musicians with a good ear. But for the rest of us, music encoded as part of an Applesoft BASIC program will sound just fine.
2. The tempo variable is quite accurate, but you should understand that your Applesoft program may affect the time **between** notes. Obviously, if you place several lines

9542:46 08	562	MULTLOOP	LSR	TEMPO	;IS LSB CLEAR?
9544:90 13	563	BCC	ROTBIT		;YES, GO ROTATE
9546:18	564	CLC			;NO, SO ADD VALUE
9547:A5 F9	565	LDA	DURVAR		
9549:65 1A	566	ADC	DUR		;INTERMED RESULTS ADDED
954B:85 1A	567	STA	DUR		;AND SAVED HERE
954D:A5 FA	568	LDA	DURVAR+1		
954F:65 1B	569	ADC	DUR+1		
9551:85 1B	570	STA	DUR+1		
9553:A5 FB	571	LDA	DURVAR+2		
9555:65 1C	572	ADC	DUR+2		
9557:85 1C	573	STA	DUR+2		;NOW ALL BYTES ADDED
9559:06 F9	574	ROTBIT	ASL	DURVAR	;ROTATE BITS TO MULTIPLY
955B:24 FA	575	ROL	DURVAR+1		
955D:26 FB	576	ROL	DURVAR+2		
955F:88	577	DEY			;DONE?
9560:D0 E0	578	BNE	MULTLOOP		;NO, GO BACK
9562:	579	*			
9562:68	580	PLA			;RESTORE TEMPO
9563:85 08	581	STA	TEMPO		
9565:68	582	PLA			;RESTORE DURATION VAR
9566:85 F9	583	STA	DURVAR		
9568:68	584	PLA			
9569:85 FA	585	STA	DURVAR+1		
956B:	586	*			
956B:	587	*	START SOUND LOOP:		
956B:	588	*			
956B:A6 06	589	LDX	PITCH		;GET FREQUENCY (PITCH)
956D:D0 05	590	BNE	LOOP		;GO SOUND IF NOT ZERO
956E:A9 01	591	LDA	#01		;MODIFY CODE FOR NO SPKR
9571:8D 7D 95	592	STA	SOUND+1		;UNUSED LOC IS ACCESSED
9574:CA	593	LOOP	DEX		;SHOULD WE CLICK SPEAKER?
9575:F0 05	594	BEQ	SOUND		;YES, CLICK WHEN PITCH=0
9577:EA	595	NOP			;DELAY TO EQUALIZE CYCLE
9579:EA	596	NOP			; TIME IF NO SOUND
9579:4C 81 95	597	JMP	DELAY		;NO CLICK; GO DELAY
957C:8D 30 C0	598	SOUND	STA	SPKR	;CLICK SPEAKER
957F:A6 06	599	LDX	PITCH		;RESTART DELAY INDEX
9581:A5 1A	600	DELAY	LDA	DUR	
9583:D0 0C	601	BNE	DECLOB		;24-BIT DECREMENT
9585:A5 1B	602	LDA	DUR+1		
9587:D0 06	603	BNE	DECMIB		
9589:A5 1C	604	LDA	DUR+2		
958B:F0 09	605	BEQ	QUIT		;ALL ZERO?
958D:C6 1C	606	DEC	DUR+2		;NO, CONTINUE
958F:C6 1B	607	DECMIB	DEC	DUR+1	
9591:C6 1A	608	DECLOB	DEC	DUR	
9593:4C 74 95	609	JMP	LOOP		
9596:A9 30	610	QUIT	LDA	#30	;RESTORE TO ACCESS SPKR
9598:8D 7D 95	611	STA	SOUND+1		
959B:60	612	RTS			;END OF NOTE ROUTINE
959C:	613	*			
959C:	614	*			
959C:	615	*	TABLE OF PITCH VALUES (PITCHTAB):		
959C:	616	*			
959C:	617	*			
959C:FF FF FF	618	PITCHTAB	DFB	,\$FF,\$FF,\$FF,\$FF,\$FF,\$FF,\$FF,\$FF	
959F:FF FF FF					
95A2:FF FF					
95A4:FF FF F6	619	DFB	,\$FF,\$FF,\$F6,\$F6,\$E8,\$D8,\$CF,\$C3		
95A7:F6 E8 DB					
95AA:CF C3					
95AC:88 AC A4	620	DFB	,\$88,\$AC,\$A4,\$A4,\$9B,\$92,\$8A		
95AF:A4 9B 92					
95B2:8A					
95B3:82 7B 7B	621	DFB	,\$82,\$7B,\$7B,\$74,\$6D,\$67,\$62		
95B6:74 6D 67					
95B9:62					
95BA:5C 57 52	622	DFB	,\$5C,\$57,\$52,\$52,\$4D,\$49,\$45		
95BD:52 4D 49					
95C0:45					
95C1:41 3E 3E	623	DFB	,\$41,\$3E,\$3E,\$3A,\$37,\$34,\$31		
95C4:3A 37 34					
95C7:31					
95C8:2E 2B 29	624	DFB	,\$2E,\$2B,\$29,\$29,\$27,\$25,\$23		
95CB:29 27 25					
95CE:23					
95CF:21 1F 1F	625	DFB	,\$21,\$1F,\$1F,\$1D,\$1B,\$1A,\$18		
95D2:1D 1B 1A					
95D5:18					
95D6:17 16 14	626	DFB	,\$17,\$16,\$14,\$14,\$13,\$12,\$11		
95D9:14 13 12					
95DC:11					
95DD:10 0F 0F	627	DFB	,\$10,\$0F,\$0F,\$0E,\$0D,\$0C,\$0B		
95E0:0E 0D 0C					
95E3:0B					
95E4:0A 09 08	628	DFB	,\$0A,\$09,\$08,\$08,\$07,\$06,\$05		
95E7:08 07 06					
95EA:05					
95EB:04 03 03	629	DFB	,\$04,\$03,\$03,\$02,\$01,\$00,\$00		
95EE:02 01 00					
95F1:00					
95F2:	630	*			
95F2:	631	*			
95F2:	632	*	RESERVE VARIABLE SPACE:		

of BASIC code or REM statements between the ampersand calls to T.U.N.E.S., there will be a time delay between the notes. For this same reason, sixteen different sixteenth notes will not have **exactly** the same duration as one whole note, because of the BASIC "overhead" time. However, the difference will be very slight and totally unnoticed by 98% of us.

3. I admit that there is a bug in my program. The problem is this: Let's say you want T.U.N.E.S. to play a thirty-second note at pitch A. You type the command **&AT**. But you have innocently spelled "AT", an Applesoft **reserved word!** Since reserved words are tokenized into one-byte values, T.U.N.E.S. does not interpret the results correctly. Two other Applesoft reserved words are potential problems, **GET** and **DEL**. The command **&GET160** would be used to sound an eighth note G at a tempo of 160 beats/minute, and **&DEL** would be used to sound an eighth note D in the Legato mode, but you won't get what you want. And **&GR** will not sound the note G followed by a rest.

Since these bugs would occur very rarely, I have chosen not to add the additional code necessary to rectify the problem. Instead, I am warning you of the bug, and will now tell you how to fix it. Instead of **&AT**, use the command **&XT : &A**. You will get the same result without using a reserved word. Instead of **&GET160**, use **&XET160 : &G**; and instead of **&DEL**, use **&XEL : &D**.

4. There are several musical features which are difficult to encode using T.U.N.E.S. For example, the slur, the trill, the mordent, the turn, double-dotted notes, etc., are all difficult to generate with T.U.N.E.S. . . . but not impossible. It just takes some extra effort and creativity. Even then, I don't guarantee the results.

Two severe limitations of most simple tone routines such as T.U.N.E.S. are (1) the lack of musical dynamics such as piano (soft), forte (loud), crescendo (gradually get louder), and diminuendo (gradually get softer); and (2) the lack of multiple voices. (You only get one tone at a time.) There are commercially available programs with some of these capabilities, but then you lose ease of use, which is T.U.N.E.S.' forte (pun intended).

5. To use T.U.N.E.S. effectively, you must be able to read music. (And even if you **do** read music, you need to have music available for the tune you want.) The only solution is to learn to read music, which really isn't too hard. Many dictionaries and most encyclopedias contain enough information for you to learn the notes. But if you want to play **Death March** in your adventure game and don't have the music, you'll just have to get to a piano and try to pick out the tune, write down the notes that sound correct, and then encode them with T.U.N.E.S.

```

95F2:      633 *-----*
95F2:      634 *
95F2:      635 STACC DS 1      ;FLAG SET IF STACCATO
95F3:      636 PAUSE DS 1     ;FLAG SET IF PAUSE
95F4:      637 REST DS 1     ;FLAG SET IF REST
95F5:      638 QUIET DS 1    ;FLAG SET IF NO SOUND
95F6:      639 PCHIDX DS 1   ;PITCH INDEX STORAGE
95F7:      640 OLDNOTE DS 1  ;FLAG SET IF OLD NOTE
95F8:      641 DURSTACC DS 2 ;THE STACC REST DURATION
95FA:      642 DURPAUSE DS 2 ;THE PAUSE DURATION
95FC:      643 TEMPDAT DS 2  ;TEMPORARY DATA STORAGE
95FE:      644 TEMPCNST DS 2 ;TEMPO CONSTANT MEM LOC

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

The image shows two musical staves. The first staff is marked "ritardando" and contains a sequence of notes: a quarter note G4, a quarter note A4, a quarter note B4, a quarter note C5, a quarter note B4, a quarter note A4, and a quarter note G4. The second staff is marked "a tempo" and "rit." and contains a sequence of notes: a quarter note G4, a quarter note A4, a quarter note B4, a quarter note C5, a quarter note B4, a quarter note A4, and a quarter note G4. Below the staves are assembly code snippets with comments:

 &BH,HEP (12) &BEL &C#ET(X-60) (13) &ET (14) &DET(X-75) &C# &B

 &DET(X-50) (15) &G3 &GQS &GQLHE &AE4P &BET(X-60) &DTL &C#EPT(X-75) &B &A (17)

Even with these limitations, I think you'll find T.U.N.E.S. an enjoyable addition to your library of utility software.

BIBLIOGRAPHY

Here are some articles which gave me direction and inspiration:

Bender, Jack H., "Pitch & Rhythm on the Apple" in **Call-A.P.P.L.E.**, June 1982, Vol. 5/ No. 6, p. 15.

Bongers, Cornelis, "Applesoft's CHARGET Routine" in **Call-A.P.P.L.E.**, March 1982, Vol. 4/No. 3, p. 21. An in-depth analysis of CHARGET.

Crossley, John, "Applesoft Internal Entry Points" in **All About Applesoft, Call-A.P.P.L.E. in Depth**, p. 51. This gives a brief explanation of CHARGET, CHRGOT, TXTPTR, and other Applesoft routines and pointers. The article was reprinted from **The Apple Orchard**.

Irwin, Peter, "Amp-L-Soft" in **Nibble**, Vol. 3/No. 7, 1982, p. 19. This assembly language program contains a simple tone routine, but its main value to me was as a model of how to write effective ampersand routines.

Scanlon, Leo J., "Let There Be Music" (Chapter 15), **Apple II Assembly Language Exercises**, John Wiley & Sons, New York, 1982, p. 124.

Voskuil, Jon, "Music Machine" in **Softside**, November 1981 (Vol. V, No. 2), p. 61.

Wagner, Roger, "Assembly Lines" in **Soft-talk**, January 1982 (Vol. 2, No. 5), p. 37.

This and previous related articles in the series explain how to write tone routines in assembly language. These articles are also found in Roger Wagner's, **Assembly Lines: The Book**, Softalk Publishing, North Hollywood, CA, 1982.

Footnotes (For Figure 1):

1. Sets the default parameters.
2. An eighth-note rest. At the same time, sets the octave to 4 and the tempo to X. (X is the tempo input variable in the program; see **Listing 2**, line 260.)
3. The tone is an eighth note, pitch B, octave 4. Only the note need be specified because the default is eighth note, octave 4. (See footnote 2.)
4. This is an F#, sixteenth-note triplet. The triplet is indicated by the apostrophe ('). The following two notes (G and F#) are also triplets.
5. The G is in octave 3, so "3" must be specified. End of measure one.
6. The grace notes, F# and G, are set to a duration of a thirty-second note.
7. This F#, written as an eighth note, is set to a sixteenth note to maintain the rhythm, since the preceding grace notes take up time.
8. The 'S' in this command sets the **Staccato** mode.
9. The 'L' resets the **Legato** mode (notes connected).
10. An eighth note rest ends measure two.

11. "Tempo di valse lento" means "slow waltz tempo." The X-50 sets the tempo at 50 beats/minute slower than the introduction.
12. The note is a dotted-half, pitch B, with an eighth-note hold. The "P" sets the Pause or Portato mode (notes separated by a slight pause).
13. The "ritardando" means to gradually get slower. The X-60 slows the tempo from the previous rate of X-50 beats/minute.
14. The grace note is given the duration of a thirty-second note.
15. The "a tempo" means "at tempo." In other words, the ritardando is over and the original tempo should resume.
16. This note is pitch G, duration quarter note. Legato mode is set (since the previous note was a staccato). The final "HE" means to hold an extra eighth note (tie).

17. The example musical score stops here. For the remainder of the music, type all of Listing 2.

KEY PERFECT 4.0		
CODE	ADDR#	ADDR#
29B2	022B	027A
2BF5	027B	02CA
2AFD	02CB	031A
24BD	031B	036A
297D	036B	03BA
2995	03BB	040A
2A27	040B	045A
2B6A	045B	04AA
272B	04AB	04FA
2E4B	04FB	054A
28EE	054B	059A
2713	059B	05EA
1ED5	05EB	062A

TOTAL PROGRAM CHECK IS : 0400

```

CHECK CODE 3.0
ON: TUNES
TYPE: B
LENGTH: 0400
CHECKSUM: A6

```

ED, T U N E S, Helper, and Digger are available on diskette for an introductory price of \$19.95 plus \$1.50 shipping/handling (\$2.50 outside the U.S.) from NIBBLE, P.O. Box 325, Lincoln, MA 01773. Offer expires 1/15/84.

LISTING 2

```

10 REM *****
11 REM * T.U.N.E.S DEMO 1 *
12 REM * BY S. SCOTT ZIMMERMAN *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC. *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
120 IF PEEK (37403) = 169 AND PEEK (38385) = 0 AND
    PEEK (1014) = 84 AND PEEK (1015) = 146 THEN 160
    : REM CHECK IF 'TUNES' IS THERE AND INITIALIZED
140 PRINT CHR$ (4);"BRUN TUNES"
160 TEXT : HOME : VTAB 2: PRINT "** COPYRIGHT 1983 BY
    MICROSPARC, INC. **"
180 VTAB 6:A$ = "MUSETTA'S WALTZ SONG"; HTAB (4) - LEN
    (A$) / 2: INVERSE : PRINT A$: NORMAL : PRINT :A$
    = "FROM 'LA BOHEME' BY PUCCINI": HTAB (4) - LEN
    (A$) / 2: PRINT A$
200 PRINT : PRINT :A$ = "PROGRAMMED BY": HTAB (4) - LEN
    (A$) / 2: PRINT A$:A$ = "S. SCOTT ZIMMERMAN": PRINT
    : HTAB (4) - LEN (A$) / 2: PRINT A$
220 PRINT : PRINT :A$ = "WITH": HTAB (4) - LEN (A$)
    / 2: PRINT A$:A$ = "T.U.N.E.S.": PRINT : HTAB (4
    1 - LEN (A$) / 2: INVERSE : PRINT A$: NORMAL
240 PRINT :A$ = "TONE UTILITY AND NOTE ENCODING SYSTE
    M": HTAB (4) - LEN (A$) / 2: PRINT A$
260 X = 150: VTAB 23: CALL - 869: PRINT "WHAT TEMPO (
    IN BEATS/MIN)? ";X: POKE 36,( PEEK (36) - 3): INPUT
    "":X$: VTAB 23: HTAB 1: CALL - 868
280 IF X$ = "" THEN 380
300 X = VAL (X$)
320 IF X > 255 THEN X = 255
340 IF X < 41 THEN X = 41
360 REM NOTES TO "MUSETTA'S WALTZ SONG"INTRODUCTION:
380 & X: & RE4TX: & B: & D: & G: & F#S: & G: & F#: &
    DE: & B: & G3: REM MEASURE 1
400 & F#T: & G: & F#S: & EES: & G: & D: & G#1: & C#1: &
    AE4L: & R: & RH: & AES: & R: & RQ: & AE: & R: & R
    Q: & AE: & R: & RQ: REM END OF MEASURE 4
420 REM START 3/4 TIME SECTION:
440 & DH4L(X - 50): & DS3: & F#1: & A4: & D: & C#H: &
    DS3: & F#1: & A4: & C#1: & B#1: & BEL: & C#E1(X -
    60): & ET: & DET(X - 75): & C#1: & B: & DET(X - 50
    ): & G3: & GGS: & GDLHE
460 & AE4P: & BET(X - 60): & DTL: & C#E1P(X - 75): &
    B: & A: REM END OF FIGURE 1
470 & XTLT(X - 50): & A: & C#E1: & F#3: & F#Q3: & F#Q4
    HE
480 & AE3: & B: & D: & F#1: & A4: & DH: & DS3: & F#1: &
    A4: & D: & C#H
500 & DS3: & F#1: & A4: & C#1: & B#1: & BE: & C#1: & E
    T: & DE: & C#1: & B: & AS'T(X - 70): & B: & AE5T(X
    - 100): & F#E4T(X - 70): & E: & D: & B: & F#E3H
    ET(X - 50): & AE4: & AQ: & EE3S: & F#1: & XEL: & D
    : & R: & D: & REHQHH.
520 REM END OF MUZETTA'S WALTZ SONG
540 VTAB 23: CALL - 869: PRINT "DO YOU WANT TO HEAR
    IT AGAIN? (Y/N)"
560 VTAB 23: HTAB 37: INVERSE : PRINT " "; & CTS5T25
    5: & RW: HTAB 37: NORMAL : PRINT " ": IF PEEK (-
    16384) < 128 THEN & GT4: & RW: IF PEEK (- 1638
    4) < 128 THEN 560
580 X = PEEK (- 16384): POKE - 16368,0
600 IF X = 217 THEN 260
620 IF X = 206 THEN END
640 GOTO 560

```

LISTING 3

```

10 REM *****
11 REM * T.U.N.E.S DEMO 2 *
12 REM * BY S. SCOTT ZIMMERMAN *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC. *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
120 IF PEEK (37403) = 169 AND PEEK (38385) = 0 AND
    PEEK (1014) = 84 AND PEEK (1015) = 146 THEN 160
    : REM CHECK IF 'TUNES' IS IN PLACE
140 PRINT CHR$ (4);"BRUN TUNES"
160 TEXT : HOME : VTAB 2: PRINT "** COPYRIGHT 1983 BY
    MICROSPARC, INC. **"
180 VTAB 6:A$ = "INVENTIO VIII ": HTAB (4) - LEN (A
    $) / 2: INVERSE : PRINT A$: NORMAL : PRINT :A$ =
    "BY JOHANN SEBASTIAN BACH": HTAB (4) - LEN (A$)
    / 2: PRINT A$
200 PRINT : PRINT :A$ = "PROGRAMMED BY": HTAB (4) - LEN
    (A$) / 2: PRINT A$:A$ = "S. SCOTT ZIMMERMAN": PRINT
    : HTAB (4) - LEN (A$) / 2: PRINT A$
220 PRINT : PRINT :A$ = "WITH": HTAB (4) - LEN (A$)
    / 2: PRINT A$:A$ = "T.U.N.E.S.": PRINT : HTAB (4
    1 - LEN (A$) / 2: INVERSE : PRINT A$: NORMAL
240 PRINT :A$ = "TONE UTILITY AND NOTE ENCODING SYSTE
    M": HTAB (4) - LEN (A$) / 2: PRINT A$
260 X = 126: VTAB 23: CALL - 869: PRINT "WHAT TEMPO (
    IN BEATS/MIN)? ";X: POKE 36,( PEEK (36) - 3): INPUT
    "":X$: VTAB 23: HTAB 1: CALL - 868
280 IF X$ = "" THEN 380
300 X = VAL (X$)
320 IF X > 255 THEN X = 255
340 IF X < 41 THEN X = 41
360 REM NOTES TO "BACH'S INVENTIO VIII":
380 & RETX: & FE3S: & A4: & F3: & C4: & F3: & FE4L: &
    E5: & D: & C: & D: & C: & B: & A: & B: & A: & G
    3: REM MEASURES 1-2
400 & FES: & A4: & C: & A: & F: & C: & AS5L: & C: & B
    : & C: & A: & C: & B: & C: & A: & C: & B: & C:
    REM MEASURES 3-4
420 & F4: & A5: & G4: & A5: & F4: & A5: & G4: & A5: &
    F4: & A5: & G4: & A5: & D4: & F: & E: & F: & D: &
    F: & E: & F: & D: & F: & E: & F: REM MEASURES 5-6
440 & BES: & G3: & D4: & E: & F: & D: & G5L: & A5: &
    G4: & F: & E: & F: & E: & D: & C: & D: & C: & B:
    REM MEASURES 7-8
460 & AES: & DSL: & C: & B: & C: & B: & A: & G3: & A4
    1: & G3: & F: & E: & F: & E: & D: & C: & E: & C#4: &
    B: & C#5: & E#5L: REM MEASURES 9-10
480 & FES: & C4: & E3: & C4: & D3: & B4: & CGL: & RQ:
    & : & RE: & C#5: & E: & C: & G: & C: REM MEASURE
    S 11-13
500 & C#5L: & BS: & A: & G4: & A5: & G4: & F: & E: &
    F: & E: & D: & C: & B: & C: & A5: & C4: & A5: &
    B: & A5: & C4: & A5: & A4: & A5: REM MEASURES 1
    4-15
520 & B!E4S: & G3: & B: & G3: & D4: & G3: & GEL4: &
    F5: & E: & D: & E: & D: & C: & B: & A: & B: &
    A: & G#5S: & B: & D: & B: & G: & D: REM MEASUR
    ES 16-18
540 & B!E5S: & C#4: & B: & C#4: & B: & C#4: & D: &
    A: & F: & D: & A5: & F4: REM MEASURES 19-20
560 & G5L: & F: & G: & B: & C4: & B: & D4: & B: &
    E4: & B: & C4: & C4: & B: & F4: & E: & F: & A5: &
    B4: & A5: & C#4: & A5: & D4: & A5: & B4: & A5: REM
    MEASURES 21-22

```

LISTING 4

```

10 REM *****
11 REM * TUNES SOUND EFFECTS *
12 REM * BY S. SCOTT ZIMMERMAN *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC. *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
120 TEXT : HOME : VTAB 2: PRINT "** COPYRIGHT 1983 BY
MICROSPARC, INC. **: VTAB 4:A* = "T.U.N.E.S. SO
UND EFFECTS": HTAB (41 - LEN (A*)) / 2: INVERSE
: PRINT A*: NORMAL
140 PRINT A* = "BY S. SCOTT ZIMMERMAN": HTAB (41 - LEN
(A*)) / 2: PRINT A*
160 PRINT CHR* (4): "BRUN TUNES"
180 VTAB 17: INVERSE: PRINT "SOUND EFFECTS": NORMAL
200 GOSUB 2000: PRINT "WHISTLE (?)": & X: POKE 0,1: FOR
I = 240 TO 250: POKE 6,252 - I: & : NEXT : & X: &
RH: & X: POKE 0,1: FOR I = 2 TO 50: POKE 6,1: & :
NEXT : GOSUB 1000: IF A* = "Y" THEN 200
220 GOSUB 2000: PRINT "MONSTER ENCOUNTER (?)": & X: FOR
I = 1 TO 10: & CT,3: & C#1: NEXT : GOSUB 1000: IF
A* = "Y" THEN 220
240 GOSUB 2000: PRINT "SPACE SHIP MOVE (?)": & X: POKE
0,5: FOR I = 1 TO 30: POKE 6,( RND (1) * 20 + 10)
: & : NEXT : GOSUB 1000: IF A* = "Y" THEN 240
260 GOSUB 2000: PRINT "TICK-TOCK": & X: FOR I = 1 TO
3: & CTSPST255: & RW: & GT4: & RW: NEXT : GOSUB 1
000: IF A* = "Y" THEN 260
280 GOSUB 2000: PRINT "CTRL-G (BEEP) EMULATOR": & X: FOR
I = 1 TO 3: & A#5: & RQ: NEXT : GOSUB 1000: IF A
* = "Y" THEN 280
300 GOSUB 2000: PRINT "SNAKE (?)": & X: FOR I = 1 TO
50: POKE 0,1: POKE 6,3: & : NEXT : GOSUB 1000: IF
A* = "Y" THEN 300
320 GOSUB 2000: PRINT "LITTLE BIRD": & X: FOR I = 1 TO
3: POKE 0,1: FOR J = 2 TO 6: POKE 6,1: & : NEXT J
: & X: & RQ: NEXT I: GOSUB 1000: IF A* = "Y" THEN
320
340 GOSUB 2000: PRINT "REFEREE WHISTLE (?)": & X: POKE
0,1: FOR I = 1 TO 50: & A6: & A#6: NEXT : GOSUB 1
000: IF A* = "Y" THEN 340
360 GOSUB 2000: PRINT "LOUDER REFEREE WHISTLE (?)": &
X: POKE 0,1: FOR I = 1 TO 50: & C6: & C#6: NEXT :
GOSUB 1000: IF A* = "Y" THEN 360
380 GOSUB 2000: PRINT "AMBULANCE SIREN": & X: FOR I =
1 TO 4: & CQ3: & G2: NEXT : GOSUB 1000: IF A* = "
Y" THEN 380
400 GOSUB 2000: PRINT "'CHARGE' FANFARE": & GE3PT230:
& CE4: & E: & GQL: & EE: & GH: GOSUB 1000: IF A*
= "Y" THEN 400
420 GOSUB 2000: PRINT "TELEPHONE RING": & X: POKE 0,6
: POKE 6,27: FOR I = 1 TO 30: & : NEXT : GOSUB 10
00: IF A* = "Y" THEN 420
440 GOSUB 2000: PRINT "STATIC (GEIGER COUNTER)": & XS
2ST255: FOR I = 1 TO 80: POKE 0, RND (1) * 10: &
A: NEXT : GOSUB 1000: IF A* = "Y" THEN 440
460 HOME : END
1000 VTAB 20: HTAB 1: CALL - 868: PRINT "DO YOU WANT
TO HEAR IT AGAIN? (Y/N) ": GET A*: PRINT A*: RETURN
2000 VTAB 20: HTAB 1: CALL - 868: RETURN

```

T.U.N.E.S. Tone Utility and Note Encoding System (Cont.)

```

580 & E4: & D: & E: & G: & A: & G: & B: & G: & C#1: &
G: & A: & G: & FEL: & D: & B!ES: & D: & G3: & F4:
REM MEASURES 23-24
600 & EEL: & C: & AES: & C: & F3: & E!4: & DSL: & F: &
E!: & F: & D: & F: & E!: & F: & D: & F: & E!: & F
: & B!: & D: & C: & D: & B!: & D: & C: & D: & B!
: & D: & C: & D: REM MEASURES 25-27
620 & G3: & B!4: & A: & B!: & G3: & B!4: & A: & B!: &
G2: & B!4: & A: & B!: & EE3S: & C: & G: & E: & B!
4: & G3: REM MEASURES 28-29
640 & CS4L: & D: & C: & B!: & A: & B!: & A: & G3: & F
: & G: & F: & E!: & DES: & GSL: & F: & E! & F: &
E: & D: & C: & D: & C: & B!: REM MEASURES 30-31
660 & A: & B!: & A: & G2: & FE: & FS3: & E: & FE: &
A: & B!: & FET(X - 10): & AET(X - 20): & FET(X -
30): & G2ET(X - 45): & E3: & FQL: & RGHQ: REM MEA
SURES 32-33
680 REM END OF BACH'S INVENTIO VIII
700 VTAB 23: CALL - 868: PRINT "DO YOU WANT TO HEAR
IT AGAIN? (Y/N)"
720 VTAB 23: HTAB 37: INVERSE: PRINT " ": & CT5ST25
5: & RW: HTAB 37: NORMAL: PRINT " ": IF PEEK ( -
16384) < 128 THEN & GT4: & RW: IF PEEK ( - 1638
4) < 128 THEN 720
740 X = PEEK ( - 16384): POKE - 16380,0
760 IF X = 217 THEN 260
780 IF X = 206 THEN END
800 GOTO 720

```