



Apple Guitar Tuner

Perhaps the most expensive and complex frequency standard on the planet at the moment with the possible exception of a few atomic clocks, an Apple can emit just the right tones to tune a guitar by.

by Jim Dawson

Rub an Apple the right way and it will make some fairly decent noises. Polish it and you will end up with a pretty fair musical standard. The program in this feature won't make the cats that hang out in your yard sing any better than usual, but it will bring your guitar up to perfect pitch.

Few things in life sound much worse than an out of tune musical instrument. An out of tune guitar can make an alley full of cats swearing their undying love to each other sound downright pleasant in comparison. Even relative tuning has its weaknesses. Unless the person setting the standard for his instrument or for the group

has perfect pitch, the standard is valid only for the particular instrument or group for a specific time.

The solution to all this is a tuning standard, one which is right all the time and about which there is no debate. This little program uses the sound generating capabilities of the Apple to provide an easily used tuning standard. A short assembler language routine is POKed into memory by the BASIC calling program and provides open string notes for tuning six and twelve string guitars.

Strings

If you have only a six string guitar, the program can be readily cut in half by removing the menu and all reference to twelve stringed instruments. Delete line 230 for starters and delete... or don't bother typing... lines 430 to 690. Lines 820 to 880 are not required for six string guitars and the lines referring to

IF FLAG = 1

such as line 700, can be replaced by a PRINT statement if you wish.

The timing loops involved in this tuning standard depend for their own accuracy on the degree of precision used in Cupertino... or Taiwan... when the Apple you're using was assembled and tested. It is unlikely, however, that the crystal oscillator frequencies in different Apples will vary more than a few hertz one from the other. A single machine language clock cycle in the Apple is, all being well, 0.97779×10^{-6} seconds in duration. It would take a rather extravagant departure from this value to make any perceptible difference in the actual tones produced. If you find any adjustment is necessary in running this program... my own version was checked against a Korg tuning standard... the values for P can be modified slightly. A larger value for P will sharpen a note and a lower version will flatten it.

If you are sufficiently interested to have read to this point, I am going to assume you know how to tune your guitar, listen for a beat note and so on. Using this program, though, you tune all the strings open. No fretting is needed.

One particular problem came up in writing the program. While a perfectly valid low E is generated by the assembler language routine, the next open string note, a low A, produced a sound which sounded to my ears, at least, more like a slightly sharp F than an A. The Korg standard, however, showed a perfect A. True, one could sing it or tune a string to it if one were to start with the sound produced by the Korg. For that matter, if both the Korg and the Apple were producing the A simultaneously, the Apple A was perfectly acceptable. Used independently, though, it sounded like a sharpened F.

Apple Guitar Tuner

The mystery remains to be explained. Perhaps it is an internal resonance in the case, a harmonic produced by the speaker itself, or a phenomenon of aural perception of a series of square waves with extremely rapid rise and fall times and relatively wide spacing between successive waves. Whatever the case, tune your low A string one octave down from the middle A produced by the Apple. It is worth noting, as an aside, that many musicians prefer to tune instruments such as the bass guitar from a standard one or even two octaves higher than the actual value of the note required. It is vastly harder to hear subtle changes in value of low tones as a string is tightened or loosened slightly.

The Program

Except for the assembler language sound generating routine, something you needn't worry about directly, since it is POKEd into memory by BASIC, the program is so simple as to need no comment. In fact, if you are not into 6502 assembler language, you can skip this part altogether and get on with typing the program and making music.

The disassembled listing below that the machine language portion of the program really starts at location \$0302, decimal 770. The lengths of notes are POKEd into \$0300 and the pitch into location \$0301, decimal locations 768 and 769 respectively, by the BASIC program.

A series of nested loops ensure that each note is played at the correct pitch and held for a reasonably long period of time, about ten seconds, so that the musician using the tuner has a fairly good chance of tuning a given string at the first attempt. Any open string guitar note may be played at will, however, and may of course be repeated as often as necessary. If you find the notes played are too long or too short, you might want to adjust L in the Applesoft program to suit your needs. Note, however, that neither L nor P may exceed two hundred and fifty-five.

There is, in the Apple an unfortunate correlation between the pitch of a note and the length of time it will sound for. Some compensation is introduced in the algorithm by including the value for pitch in the actual running of the program as a factor in modifying the time for which a note will sound. Further adjustment can be made by playing with L. In the case of the G, the maximum possible value for L of 255 would produce a tone for some five seconds only, shorter in any case than the other notes. A compromise is made in this case. L is made slightly shorter and in line 770, GOSUB 790 is invoked twice.

The ideal tuning standard would produce a sine wave free of harmonic content altogether. Apples do not send sine waves to their speakers, alas. The best they can manage is a series of square waves of ex-

Program one, the Applesoft guitar tuner

```

100 HOME : VTAB 7: HTAB 14: PRINT "=====": PRINT : PRINT : PRINT : HTAB 14: PRINT
=====
110 VTAB 9: HTAB 15: PRINT "GUITAR TUNER"
120 VTAB 14: HTAB 6: PRINT "(C) 1985 - ORMSTOWN, QUEBEC": PRINT
130 HTAB 19: PRINT "BY": PRINT
140 HTAB 15: PRINT "JIM DAWSON"
150 VTAB 23: PRINT "      PRESS ANY KEY TO GO ON..."; GET A$
160 FOR M = 770 TO 795
170 READ C
180 POKE M,C
190 NEXT
200 DATA 172,01,03,174,01,03,232,208,253
210 DATA 169,04,32,168,252,173,48,192,136
220 DATA 208,239,206,00,03,208,231,96
230 GOTO 920
240 HOME : VTAB 22: PRINT "<<(STRINGS NUMBERED LOWEST TO HIGHEST)>>"
250 POKE 32,6: PRINT
260 VTAB 4: PRINT "STRING NOTE  PRESS..."
270 PRINT "=====": PRINT
280 PRINT "1ST.  LOW E  [CTRL] E"
290 PRINT "2ND.   A    [CTRL] A"
300 PRINT "3RD.   D    [CTRL] D"
310 PRINT "4TH.   G    [CTRL] G"
320 PRINT "5TH.   B    [CTRL] B"
330 PRINT "6TH.  HIGH E   E"
340 VTAB 23: PRINT "<<(PRESS 'ESC' TO QUIT)>>"
350 VTAB 18: PRINT "WHICH NOTE, PLEASE? "; GET A$
360 IF A$ = CHR$(27) THEN TEXT : HOME : END
370 TEXT : HOME
380 P = ASC (A$)
390 IF P = 3 OR P = 6 THEN 250
400 IF P > 7 AND P < 69 THEN 250
410 IF P > 69 THEN 250
420 GOTO 720
430 HOME : VTAB 22: PRINT "<<(STRINGS NUMBERED LOWEST TO HIGHEST)>>"
440 FLAG = 1
450 POKE 32,6: PRINT
460 VTAB 1: PRINT "STRING NOTE  PRESS..."
470 PRINT "=====": PRINT
480 PRINT "1ST.  LOW E  [CTRL] E"
490 PRINT "2ND.  MID E  ME "
500 PRINT "3RD.  LOW A  [CTRL] A"
510 PRINT "4TH.  MID A  MA "
520 PRINT "5TH.  LOW D  [CTRL] D"
530 PRINT "6TH.  MID D  MD "
540 PRINT "7TH.  LOW G  [CTRL] G"
550 PRINT "8TH.  MID G  MG "
560 PRINT "9TH.   B    [CTRL] B"
570 PRINT "10TH. B    [CTRL] B"
580 PRINT "11TH. E     E"
590 PRINT "12TH. E     E"
600 TEXT : VTAB 23: PRINT "      PRESS 'Q' TO QUIT"
605 N$ = "":A$ = ""
610 HTAB 6: VTAB 18: PRINT "WHICH NOTE, PLEASE? ";A$; : IF A$ = "E" THEN 720
611 GET N$
615 P = PEEK ( - 16384); IF P < 8 THEN 720
620 IF N$ = "Q" THEN HOME : END

```


Apple Guitar Tuner

tremely short duration spaced at specific intervals. The speaker is, in fact, plucked with brief pulses... if this were done once only, all you would hear would be a faint click. If, however, the clicks follow each other regularly and rapidly enough, the ear is fooled into thinking it hears a recognizable note.

The actual duration time of the pulse which clicks the speaker does not vary. What does change is the delay in the Apple WAIT subroutine at \$FCA8 and the variable delay in \$0308 to \$0310. Both these variables affect the pitch produced, the former by introducing a relatively subtle change in pitch for each unit of increment or decrement. The actual formula is

$$T = 0.5(x + 27x + 26)$$

where the delay value, x , is in the accumulator. T is in microseconds.

This is the same routine that is used by the Applesoft SPEED command, by the way.

*302L

0302-	AC 01 03	LDY	\$0301
0305-	AE 01 03	LDX	\$0301
0308-	E8	INX	
0309-	D0 FD	BNE	\$0308
030B-	A9 04	LDA	#04
030D-	20 A8 FC	JSR	\$FCA8
0310-	AD 30 C0	LDA	\$C030
0313-	88	DEY	
0314-	D0 EF	BNE	\$0305
0316-	CE 00 03	DEC	\$0300
0319-	D0 E7	BNE	\$0302
031B-	60	RTS	
031C-	00	BRK	
031D-	00	BRK	
031E-	00	BRK	
031F-	00	BRK	
0320-	00	BRK	
0321-	00	BRK	
0322-	00	BRK	
0323-	00	BRK	

Program two. The machine code to make noises.

In any case, as long as the train of square waves is produced with the correct interval between them, the ear will hear an open string note and tuning the guitar accurately presents no problem.

If your next hootnanny is not a success, you will not be able to blame badly tuned guitars, and your local prima donna who had heretofore though herself endowed with perfect pitch may find her thinking being revised by your Apple tuning standard.

CNI

```

625 A$ = A$ + N$: IF LEN (A$) < 2 THEN VTAB 18: HTAB 25: PRINT A$: GOTO 610
630 IF LEN (A$) > 2 THEN 430
640 IF LEN (A$) = 1 THEN P = ASC (A$): GOTO 720
650 IF A$ = "MA" THEN P = 47:L = 160: POKE 780,04:K$(1) = "MID A":J$(1) = "4TH.": GOSUB 800: GOSUB 790: GOTO 700
660 IF A$ = "MD" THEN P = 106:L = 125: POKE 780,04:K$(1) = "MID D":J$(1) = "6TH.": GOSUB 800: GOSUB 790: GOTO 700
670 IF A$ = "ME" THEN P = 170:L = 10: POKE 780,30:K$(1) = "MID E":J$(1) = "2ND.": GOSUB 800: GOSUB 790: GOTO 700
680 IF A$ = "MG" THEN P = 149:L = 75: POKE 780,04:K$(1) = "MID G":J$(1) = "8TH.": GOSUB 800: GOSUB 790: GOTO 700
690 GOTO 700
700 IF FLAG = 1 THEN 430
710 GOTO 240
720 GOSUB 800
730 IF P = 1 THEN P = 47:L = 160: POKE 780,04:K$(1) = "A":J$(1) = "2ND.": GOSUB 800: GOSUB 790: GOTO 700
740 IF P = 2 THEN P = 72:L = 135: POKE 780,04:K$(1) = "B":J$(1) = "5TH.": GOSUB 800: GOSUB 790: GOTO 700
750 IF P = 4 THEN P = 94:L = 27: POKE 780,30:K$(1) = "D":J$(1) = "3RD.": GOSUB 800: GOSUB 790: GOTO 700
760 IF P = 5 THEN P = 152:L = 10: POKE 780,45:K$(1) = "LOW E":J$(1) = "1ST.": GOSUB 800: GOSUB 790: GOTO 700
770 IF P = 7 THEN P = 19: POKE 780,04:K$(1) = "G":J$(1) = "4TH.": GOSUB 800:L = 200: GOSUB 790: GOSUB 790: GOTO 700
780 IF P = 69 THEN P = 125: POKE 780,04:L = 110:K$(1) = "E":J$(1) = "6TH.": GOSUB 800: GOSUB 790: GOTO 700
790 POKE 768,L: POKE 769,P: CALL 770: RETURN
800 HOME :K$ = "THE NOTE BEING PLAYED IS ":I$ = " STRING"
810 IF FLAG = 0 THEN 890
820 IF P = 47 THEN J$(1) = "STRINGS #3 & #4":I$ = "": GOTO 890
830 IF P = 106 THEN J$(1) = "6TH.": GOTO 890
840 IF P = 94 THEN J$(1) = "5TH.": GOTO 890
850 IF P = 170 THEN J$(1) = "2ND.": GOTO 890
860 IF P = 149 THEN J$(1) = "8TH.": GOTO 890
870 IF P = 72 THEN J$(1) = "STRINGS 9 & 10":I$ = "": GOTO 890
880 IF P = 125 THEN J$(1) = "STRINGS 11 & 12":I$ = "": GOTO 890
890 VTAB 8: HTAB 8: PRINT K$:K$(1): PRINT : PRINT : HTAB 16: PRINT J$(1):I$
900 IF P = 47 THEN VTAB 14: HTAB 5: PRINT "<<TUNE LOW 'A' ONE OCTAVE DOWN>>"
910 RETURN
920 HOME : VTAB 6: HTAB 16: PRINT "=====": VTAB 10: HTAB 16: PRINT "=====": VTAB 8: HTAB 17: PRINT "OPTIONS"
930 VTAB 13
940 HTAB 10: PRINT "A] 6 STRING GUITAR": PRINT
950 HTAB 10: PRINT "B] 12 STRING GUITAR": PRINT : PRINT
960 PRINT " TYPE ";: INVERSE : PRINT "LETTER";: NORMAL : PRINT " TO SELECT CHOICE --> ";: GET A$
970 IF A$ > "B" OR A$ < "A" THEN 920
980 A = ASC (A$) - 64: ON A GOTO 240,430
1000 P = ASC (A$): ON A GOTO 1010,1020,1030,1040,1050,1060,1070
1010 PRINT "A": RETURN
1020 PRINT "B": RETURN
1030 PRINT "INVALID ENTRY": POP : GOTO 700
1040 PRINT "D": RETURN
1050 PRINT "E": RETURN
1060 PRINT "INVALID ENTRY": POP : GOTO 700
1070 PRINT "G": RETURN

```