

```

001 ;Small Magnetic Loop Antenna Autotuner Program, Version 5a
Quarter-Quarter for the Model 3 (Standardized) Autotuner Circuit.
002
003 ;Copyright: Andrew Cornwall / Enfield, Nova Scotia / Canada
004 ;This program is free for anyone to use for non-commercial
purposes.
005
006 ;Small Magnetic Loop Antenna Automatic Tuner Project Model 3
Circuit
007 ;This version of the Automatic Tuner program coincides with
the article,
008 ;'Experimental Automatic Tuner for Small Magnetic Loop
Antennas'
009 ;by Andrew Cornwall / VE1COR, published in the ; XXMonth-
XXMonth 2018;
010 ;edition of the "The Canadian Amateur" (TCA) magazine of the
011 ;Radio Amateurs of Canada. The article explains how to use
this program.
012
013 ;Notes:
014 ;The program assumes that a free-turning (360 degree) variable
015 ;capacitor is attached to the automatic tuner. It is possible
to modify the
016 ;circuit and progrm allowing for use with a 180 deg., half-
turn variable
017 ;capacitor. How to do this is explained on the website:
018 ;www.cornwaab.altervista.org.
019
020 ;The Model 3 circuit has optional provision to use a sampling
antenna relay to
021 ;reduce the current going into the automatic tuner's receiver
when high
022 ;wattage transmissions are taking place (i.e. not a low
wattage tuning
023 ;signal). Although the program provides for relay operation,
it does not
024 ;matter that a relay is omitted. Without a relay there will
not be the added
025 ;protection from possible overload, however, with transmtion
of up to 100
026 ;relay protection may not be needed.
027
028 ;This program is written for a PICAXE 18M2+ Microcontroller.
It should also
029 ;work with larger microcontrollers in the PICAXE family, but
the pin
030 ;assignments will have to be changed.
031
032 ; ***** Program Starts Below
*****
033 setfreq m32 ;This tells the 18M2+ to work at a 32 MHz
microcontroller

```

```

034 ;(internal) clock speed. This its maximum speed. I've run the
program at
035 ;slower rates, but why bother. The effect program
consideration of 32 MHz
036 ;is to speed up the pause statement by a factor of 8, e.g.
pause 8000 (ms)
037 ;is one second or 1000 ms.
038
039 SERTXD ("VE1COR Autotuner program, Direct Method Ver. 5a
(Quarter-Quarter)",CR,LF)
040
041 ;VARIABLES - Variables starting with 'W' have 16 bits and can
range in
042 ;value from 0 to 65535, those starting with 'B' have 8 bits
and range from
043 ;0 to 255. All variables are integers.
044
045 symbol AVG_VOLTS=W0 ;AVG_VOLTS is the measure of Sample
Antenna radiated
046 ;signal
047 symbol DELTA=W1
048 symbol MAX_VOLTS=W2
049 symbol REF_VOLTS=W3
050 symbol VOLTS=W4
051 symbol M=W5
052 symbol TEMP_VOLTS=W6
053 symbol NUMBER_STEPS=W7
054 symbol MANUAL=B24
055 symbol SWEEP=B25
056 symbol J=B27
057 symbol N=B26
058
059 ;PINS - Pin names, e.g. LED_RED, are assigned here for inputs
or outputs.
060 ;Most pins of the 18M2+ can be used as either an input or
output. Whether
061 ;a pin is an input or output is determined by how it used in
the program.
062 ;Pin references are labeled starting with 'B' or 'C' in the
18M2+,
063 ;and represent specific pin numbers 1 to 18 according to the
microcontroller's
064 ;diagram. Pin designations were made mostly for the
convenience of wiring.
065
066 symbol LED_RED=B.0 ;RED LED too high tuning signal. Output
HIGH or LOW
067 symbol LED_GREEN=B.1 ;GREEN LED manual mode. Output HIGH or
LOW
068 symbol LED_BLUE=B.2 ;BLUE LED too low tuning signal. OUTPUT
HIGH or LOW
069 symbol BUZZER=B.3 ;Buzzer. Output HIGH or LOW

```

```

070  symbol DIR=B.4 ;Determins turn direction of stepper motor.
Output HIGH or LOW
071  symbol STEPX=B.5 ;Move stepper moter one increment. Output
PULSE
072  symbol MS1=B.6 ;Determins stepper motor partial incremment.
Output HIGH or LOW
073  symbol MS2=C.0 ;Determins stepper motor partial incremment.
Output HIGH or LOW
074  symbol SLEEPX=B.7 ;Turns stepper motor ON or OFF. Outut HIGH
or LOW
075  symbol ADC=C.1 ;Analogue voltage input from receiver. Input
~0 to ~5 Volts
076  symbol RELAY=C.7; Optional relay - disconnects sampling
antenna from
077 ;receiver when tuning is completed. Output HIGH or LOW
078
079 ;Initial conditions
080 LOW LED_RED ;Off
081 LOW LED_BLUE ;Off
082 LOW LED_GREEN ;Off
083 LOW BUZZER ;Off
084 LOW STEPX ;Off
085 HIGH MS2 ;HIGH MS2 and LOW MS1 sets the stepper motor to move
at quarter step
086 LOW MS1
087 LOW SLEEPX ;NO power to the stepper motor when LOW
088 LOW DIR ;Direction at the Easy Driver needs to be something to
start,
089 ;which way does not matter
090 LOW RELAY ;Antenna disconnected to start
091 SWEEP=0 ;Sweep counts the number of active tuning tries later
on in the
092 ;program.
093 MANUAL=0 ;MANUAL MODE=1 when autotuner is in manual mode
094
095 NUMBER_STEPS=1640 ;1600 is a full turn for a 0.9 degree
stepper motor turning
096 ;at quarter step. The extra 40 steps accounts for possible
slipage in the
097 ;stepper motor-variable capacitor linkage. (I have not
encountered any.) The
098 ;NUMBER_STEPS variable can be changed when using a stepper
motor of a
099 ;different partial step (e.g. one-half step vs. one-quarter).
100
101 ; Below: pocessor self test and and announce time to start
tuning transmission
102 ;unless manual mode is desired.
103 for j=1 to 2
104     gosub BLINK_BLUE
105     gosub BLINK_RED
106

```

```

gosub BLINK_GREEN
107     gosub BLINK_BUZZER
108 next j
109
110 pause 16000 ;Wait 2 seconds for transmitter tuning signal to
start
111
112 MAX_VOLTS=0 ;Initialize maximum value variable
113
114 HIGH SLEEPX ;Turn on stepper motor power
115 HIGH RELAY ;Connect sampling antenna, if available, to the
receiver
116
117 For j=1 to 20 ;Move variable capacitor away from previous
tuned location.
118     pause 10
119     gosub MOVE
120 next j
121
122 ;Program section below finds maximum value of sample antenna
voltage
123 ;during one full turn of the variable capacitor.
124
125 BEGIN_MAX_VOLTS_SWEEP:
126 SERTXD ("STARTING MAX. VALUE SWEEP AT QUARTER-STEP",CR,LF)
127 for M=1 to NUMBER_STEPS
128     pause 20
129     gosub MOVE
130     gosub READ_AVG_VOLTS
131
132     if AVG_VOLTS>MAX_VOLTS then
133         MAX_VOLTS=AVG_VOLTS
134     endif
135
136     if AVG_VOLTS>10 then
137         sertxd ("TUNER POS: ",#M," VOLTS: ",#AVG_VOLTS, "
HIGHEST YET: ",#MAX_VOLTS,CR,LF)
138     endif
139 next M
140
141 ;Manual Mode triggered by near Zero MAX_VOLTS.
142
143 if MAX_VOLTS<20 then
144
145     LOW SLEEPX ;Turn off power to stepper motor.
146     MANUAL=1
147     sertxd ("STARTING MANUAL MODE",cr,lf)
148     HIGH LED_GREEN ;Green LED ON indicates autotuner is in
Manual Mode
149     MAX_VOLTS=0
150
151     REPEAT: ;Manual Mode continually returns to this point
152

```

```

153     gosub READ_AVG_VOLTS
154
155     if AVG_VOLTS>MAX_VOLTS then ;Record Maximum Volts so far
156         MAX_VOLTS=AVG_VOLTS
157     endif
158
159     sertxd ("MANUAL - VOLTS: ",#AVG_VOLTS,"   HIGHEST YET: "
, #MAX_VOLTS,cr,lf)
160     pause 2500 ;slow manual readings to about 3 times per
second
161     goto REPEAT
162 endif
163
164 ;FROM HERE IS ACTIVE AUTOTUNE MODE
165
166 sertxd (CR,LF,"ACTIVE TUNING AT QUARTER-STEP FOR APPROX.
MAXIMUM VOLTS ",#MAX_VOLTS, CR,LF)
167
168 DELTA=MAX_VOLTS/20 ;DELTA is a 5% factor to reduce autotune
target in
169 ;succesive active tuning sweeps if necessary
170
171
172 TRY_ANOTHER_SWEEP: ;This is the point where the program
returns if there
173 ;is more than another active tuning sweep is needed
174
175 SWEEP=SWEEP+1 ;Keeps track of number of active tuning sweeps
that is occuring.
176
177 REF_VOLTS=MAX_VOLTS-DELTA ;Lower autotune target with each
active sweep
178 MAX_VOLTS=REF_VOLTS ;Lowers MAX_Volts to prepare for next
possible
179 ;active sweep
180
181 sertxd ("STARTING ACTIVE TUNING SWEEP: ",#SWEEP," - REF_VOLTS:
",#REF_VOLTS,CR,LF)
182
183 ;Announce Active tuning is about to start for next tuning
sweep - number
184 ;sweep indicated by buzzer beeps and LED blinks
185 for j=1 to SWEEP
186     gosub BLINK_BUZZER
187     gosub BLINK_RED
188     gosub BLINK_BLUE
189 next j
190
191 ;Start of an active tuning sweep process
192 for M=1 to NUMBER_STEPS
193     pause 10
194     gosub MOVE
195     gosub READ_AVG_VOLTS

```

```

196
197 ;Send data to computer, if attached, only for radiated signal
values greater
198 ;than 10
199     if AVG_VOLTS>10 then
200         sertxd ("TUNER POS: ",#SWEEP," / ", #M," TUNED VOLTS:
",#AVG_VOLTS, CR,LF)
201     endif
202
203 ;Check to see if radiated signal is greater than or equal to
the target, if
204 ;YES go to final autotuning stage. If NOT continue the tuning
sweep.
205     if AVG_VOLTS>=REF_VOLTS then
206         goto QUIT_TUNING
207     endif
208
209 Next M
210
211 ;If the target radiated signal was not encountered during the
active tuning
212 ;sweep, try again with a lower target for up to four tries.
213 if M>NUMBER_STEPS then
214     if SWEEP<4 then
215         goto TRY_ANOTHER_SWEEP
216     endif
217     sertxd ("* UNABLE TO TUNE",CR,LF)
218     LOW SLEEPX ;Turn off power to stepper motor
219     goto EXIT_TUNING
220 endif
221
222 ;Active tuning should be completed now. The sequence below is:
turn off
223 ;power to stepper motor, perform a post tuning radiated signal
diagnostic
224 ;sound buzzer, and light status LEDs.
225
226 QUIT_TUNING:
227
228 LOW SLEEPX ;turn off power to stepper motor
229
230 For J=1 to 8 ;Final diagnostic. Is radiated signal about
stable?
231 gosub READ_AVG_VOLTS
232 sertxd ("FINAL VOLTS: ",#AVG_VOLTS, CR,LF)
233 pause 50
234 Next J
235
236 EXIT_TUNING: ;This junction is used when autotuning has
failed. Radiated
237 ;signal diagnostic (above) is bypassed when tuning is
completed.
238

```

```

239  LOW RELAY ;turn off relay thus removing sampling antenna from
reciever
240
241  gosub BLINK_BUZZER
242
243  sertextd ("TUNING FINISHED",CR,LF)
244
245  if MAX_VOLTS>875 then ; Blink red LED for possible receiver
saturation
246      HIGH LED_BLUE ;Blue LED stays ON
247      gosub BLINK_BUZZER ;second buzzer sound to indicate
possible problem
248      sertextd ("* HIGH SIGNAL SATURATION MAY HAVE OCCURED",CR,LF)
249      STAY_HERE_4:
250      gosub BLINK_RED
251      goto STAY_HERE_4
252  endif
253
254  if MAX_VOLTS<45 then ; Blink Bluse LED for signal too low to
tune
255      HIGH LED_RED ;RED LED stays ON
256      gosub BLINK_BUZZER ;;second buzzer sound to indicate
possible problem
257      sertextd ("* SIGNAL TOO LOW TO TUNE",CR,LF)
258      STAY_HERE_3:
259      gosub BLINK_BLUE
260      goto STAY_HERE_3
261  endif
262
263  ;Both red and blue LEDs lit means tuning completed OK
264  HIGH LED_BLUE
265  HIGH LED_RED
266
267  STAY_HERE_5:
268  goto STAY_HERE_5
269  end
270
271  ;SUBROUTINES ARE BELOW
272
273  ;Blinking LEDs and Buzzer sounds are first
274  BLINK_BUZZER:
275  HIGH BUZZER
276  pause 400
277  LOW BUZZER
278  pause 400
279  return
280
281  BLINK_BLUE:
282  HIGH LED_BLUE
283  pause 400
284  LOW LED_BLUE
285  pause 400
286  return

```

```

287
288 BLINK_RED:
289 HIGH LED_RED
290 pause 400
291 LOW LED_RED
292 pause 400
293 return
294
295 BLINK_GREEN:
296 HIGH LED_GREEN
297 pause 400
298 LOW LED_GREEN
299 pause 400
300 return
301
302 ;The MOVE subroutine causes the stepper motor to turn one step
or partial
303 ;step by sending a pulse (low-high-low) to the stepper motor
controller
304 MOVE:
305 pause 10 ;was 20
306 HIGH STEPX
307 pause 10 ;was 20
308 LOW STEPX
309 return
310 ;READ_AVG_VOLTS examines the ADC voltage level at pin C.1 and
interprets
311 ;it as a variable with the range of 0 to 1023, does some
processine then
312 ;provides the value to the rest of the program.
313 READ_AVG_VOLTS:
314
315 if AVG_VOLTS>10 and MANUAL=0 then 'pause measuring 2/3 second
for
316 ;transmitter SWR foldback function to relent
317     pause 6600
318 endif
319
320 TEMP_VOLTS=0
321 AVG_VOLTS=0
322 VOLTS=0
323
324 ;Voltage is read four times and averaged to smooth
inconsistencies
325 for N=1 to 4
326 pause 10
327 readadc10 ADC,VOLTS
328 TEMP_VOLTS=VOLTS/4
329 AVG_VOLTS=AVG_VOLTS+TEMP_VOLTS
330 next N
331
332 if AVG_VOLTS>875 then; while tuning of possible saturation and
333

```



```
;about 100 watt transmitter power limit
334     gosub BLINK_RED
335     sertxd ("Probable Saturation Measurement of : "
, #AVG_VOLTS, cr, lf)
336 endif
337
338 return
```