

```

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

```

```

microcontroller
035 ;(internal) clock speed. This its maximum speed. I've run the
program at
036 ;slower rates, but why bother. The effect program
consideration of 32 MHz
037 ;is to speed up the pause statement by a factor of 8, e.g.
pause 8000 (ms)
038 ;is one second or 1000 ms.
039
040 SERTXD ("VE1COR Autotuner program, Direct Method Ver. 5c Half-
Eighth step",CR,LF)
041
042 ;VARIABLES - Variables starting with 'W' have 16 bits and can
range in
043 ;value from 0 to 65535, those starting with 'B' have 8 bits
and range from
044 ;0 to 255. All variables are integers.
045
046 symbol AVG_VOLTS=W0 ;AVG_VOLTS is the measure of Sample
Antenna radiated
047 ;signal
048 symbol DELTA=W1
049 symbol MAX_VOLTS=W2
050 symbol REF_VOLTS=W3
051 symbol VOLTS=W4
052 symbol M=W5
053 symbol TEMP_VOLTS=W6
054 symbol NUMBER_STEPS=W7
055 symbol SWEEP_VOLTS=W8
056 symbol MANUAL=B24
057 symbol SWEEP=B25
058 symbol J=B27
059 symbol N=B26
060
061 ;PINS - Pin names, e.g. LED_RED, are assigned here for inputs
or outputs.
062 ;Most pins of the 18M2+ can be used as either an input or
output. Whether
063 ;a pin is an input or output is determined by how it used in
the program.
064 ;Pin references are labeled starting with 'B' or 'C' in the
18M2+,
065 ;and represent specidic pin numbers 1 to 18 according to the
microcontroller's
066 ;diagram. Pin designations were made mostly for the
convenience of wiring.
067
068 symbol LED_RED=B.0 ;RED LED too high tuning signal. Output
HIGH or LOW
069 symbol LED_GREEN=B.1 ;GREEN LED manual mode. Output HIGH or
LOW
070 symbol LED_BLUE=B.2 ;BLUE LED too low tuning signal. OUTPUT

```

HIGH or LOW

```

071  symbol BUZZER=B.3  ;Buzzer. Output HIGH or LOW
072  symbol DIR=B.4    ;Determins turn direction of stepper motor.
Output HIGH or LOW
073  symbol STEPX=B.5  ;Move stepper moter one increment. Output
PULSE
074  symbol MS1=B.6    ;Determins stepper motor partial incremment.
Output HIGH
075  ;or LOW
076  symbol MS2=C.0    ;Determins stepper motor partial incremment.
Output HIGH or LOW
077  symbol SLEEPX=B.7 ;Turns stepper motor ON or OFF. Outut HIGH
or LOW
078  symbol ADC=C.1    ;Analogue voltage input from receiver. Input
~0 to ~5 Volts
079  symbol RELAY=C.7; Optional relay - disconnects sampling
antenna from
080  ;receiver when tuning is completed. Output HIGH or LOW
081
082  ;Initial conditions
083  LOW LED_RED ;Off
084  LOW LED_BLUE ;Off
085  LOW LED_GREEN ;Off
086  LOW BUZZER ;Off
087  LOW STEPX ;Off
088  LOW MS2 ;LOW MS2 and HIGH MS1 sets the stepper motor to move
at half step
089  HIGH MS1
090  LOW SLEEPX ;NO power to the stepper motor when LOW
091  LOW DIR ;Direction at the Easy Driver needs to be something to
start,
092  ;which way does not matter
093  LOW RELAY ;Antenna disconnected to start
094  SWEEP=0 ;Sweep counts the number of active tuning tries later
on in the
095  ;program.
096  MANUAL=0 ;MANUAL MODE=1 when autotuner is in manual mode
097
098  NUMBER_STEPS=1640 ;1600 is a full turn for a 0.9 degree
stepper motor turning
099  ;at quarter step. The extra 40 steps accounts for possible
slipage in the
100  ;stepper motor-variable capacitor linkage. (I have not
encountered any.) The
101  ;NUMBER_STEPS variable can be changed when using a stepper
motor of a
102  ;different partial step (e.g. one-half step vs. one-quarter).
103
104  ; Below: pocessor self test and and announce time to start
tuning transmission
105  ;unless manual mode is desired.
106

```

```

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

```

```

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

```

```

196     SWEEP_VOLTS=AVG_VOLTS
197     pause 10
198     gosub MOVE
199     gosub READ_AVG_VOLTS
200
201     ;Send data to computer, if attached, only for radiated signal
values greater
202     ;than 10
203     if AVG_VOLTS>10 then
204     sertxd ("TUNER POS: ",#SWEEP," / ", #M," TUNED VOLTS:
",#AVG_VOLTS, CR,LF)
205     endif
206
207     ;Check to see if radiated signal is greater than or equal to
the target, if
208     ;YES go to final autotuning stage. If NOT continue the tuning
sweep.
209     if AVG_VOLTS>=REF_VOLTS then
210     goto QUIT_TUNING
211     endif
212
213 Next M
214
215     ;If the target radiated signal was not encountered during the
active tuning
216     ;sweep, try again with a lower target for up to four tries.
217     if M>NUMBER_STEPS then
218     if SWEEP<4 then
219     goto TRY_ANOTHER_SWEEP
220     endif
221     sertxd ("* UNABLE TO TUNE",CR,LF)
222     LOW SLEEPX ;Turn off power to stepper motor
223     goto EXIT_TUNING
224     endif
225
226     ;Active tuning should be completed now. The sequence below is:
turn off
227     ;power to stepper motor, perform a post tuning radiated signal
diagnostic
228     ;sound buzzer, and light status LEDs.
229
230 QUIT_TUNING:
231
232 LOW SLEEPX ;turn off power to stepper motor
233
234 For J=1 to 8 ;Final diagnostic. Is radiated signal about
stable?
235 gosub READ_AVG_VOLTS
236 sertxd ("FINAL VOLTS: ",#AVG_VOLTS, CR,LF)
237 pause 50
238 Next J
239
240 EXIT_TUNING: ;This junction is used when autotuning has

```

```

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

```

```

287  pause 400
288  LOW LED_BLUE
289  pause 400
290  return
291
292  BLINK_RED:
293  HIGH LED_RED
294  pause 400
295  LOW LED_RED
296  pause 400
297  return
298
299  BLINK_GREEN:
300  HIGH LED_GREEN
301  pause 400
302  LOW LED_GREEN
303  pause 400
304  return
305
306  ;The MOVE subroutine causes the stepper motor to turn one step
or partial
307  ;step by sending a pulse (low-high-low) to the stepper motor
controller
308  MOVE:
309  pause 10 ;was 20
310  HIGH STEPX
311  pause 10 ;was 20
312  LOW STEPX
313  return
314  ;READ_AVG_VOLTS examines the ADC voltage level at pin C.1 and
interprets
315  ;it as a variable with the range of 0 to 1023, does some
processine then
316  ;provides the value to the rest of the program.
317  READ_AVG_VOLTS:
318
319  if AVG_VOLTS>10 and MANUAL=0 then 'if signal detected pause
1/2 second;
320  ;for transmitter SWR foldback function to relent, and go to
eighth step
321      pause 3600 ; first of two pauses that total 1/2 second
322      HIGH MS1 ;change to eighth step if signal detected
323      HIGH MS2
324      pause 1200
325  else
326      HIGH MS1 ; keep (or revert to) half-stepping
327      LOW MS2
328      pause 10
329  endif
330
331  TEMP_VOLTS=0
332  AVG_VOLTS=0

```



```
333 VOLTS=0
334
335 ;Voltage is read four times and averaged to smooth
inconsistencies
336 for N=1 to 4
337   pause 10
338   readadc10 ADC,VOLTS
339   TEMP_VOLTS=VOLTS/4
340   AVG_VOLTS=AVG_VOLTS+TEMP_VOLTS
341 next N
342
343 if AVG_VOLTS>875 then; while tuning of possible saturation and
344 ;about 100 watt transmitter power limit
345     gosub BLINK_RED
346     sertxd ("Probable Saturation Measurement of : "
, #AVG_VOLTS, cr, lf)
347 endif
348
349 return
```