Feedback-Based Optimising Automatic Tuner

For a Small Transmitting Magnetic Loop

Joshua J Mardling

VE1CEN

27/02/2018

Contents

About the Author

I am an amateur radio operator with a focus on CW and Digital modes. I am an electrical engineer, and I work in the radio engineering industry, so I enjoy applying what I learn at work to what I do at home. I live in an environment that is not friendly to large antennas, lacking trees for a dipole or the space for a tower, so I make the most of my situation by designing and building magnetic loops. While hopping around the bands, I found that the traditional manual tuning procedure of my loop antenna was slowing me down. Irritation being the mother of invention, I decided to design my own automatic tuner, based on a simple sense antenna providing feedback, and now I happily QSY at the touch of a button. I hope that this design helps further the popularity of magnetic loop antennas in the increasingly space-restricted home environment.

Abstract

Small Transmitting Magnetic Loops (STMLs) are novel antennas that offer a small physical footprint and potentially high efficiency in exchange for comparably small operating bandwidth when held against, for instance, a resonant dipole or ¼ wave vertical. Typical STMLs achieve impedance matching by way of a variable capacitor series-connected to the radiating loop element. By varying the capacitance, the loop can be matched to the transmitting system at a range of frequencies. Typical home-made magnetic loops achieve remote tuning with a motor-controlled capacitor, manually controlled from the operator's position. The purpose of this project is to outline the design, construction, and successful deployment of a simple, cost-effective, easily-constructed automatic impedance matcher for a STML. This design does not require complicated forward and reflected power sampling and attendant calculations, but rather achieves a match by maximising radiated signal strength.

Theory of Operation

The crux of the design is the use of a simple sense antenna, placed in a fixed position in the centre of the radiating loop, with a simple 3-component passive rectifier, feeding to an Arduino Analog to Digital Converter (ADC) input (the Arduino Uno has several). The sense antenna converts radiated RF to a DC voltage that directly corresponds to radiated signal strength. In a magnetic loop antenna, peak power radiation is achieved when the capacitor is tuned to resonate the inductance inherent in the antenna, resulting in a purely real impedance. This impedance is transformed up to 50Ω via the feed method; either by a geometrical relation between the size of the radiating element and the feed element, in the case of a Faraday coupling feed, or by a gamma match, or by a carefully selected turn ratio on a coupling toroid. Regardless of the feed method, the important thing to note is that the maximum radiated power, minimum SWR, and thus an optimal match, all coincide and all rely on the tuning capacitor. Thus, to minimise SWR, it is sufficient to maximise radiated power. It is much easier to sample radiated power than it is to calculate SWR, and this is the basis of the design.

The Arduino microcontroller samples the rectified DC voltage during its tuning procedure, and, provided that the stepper motor provides sufficient resolution, the Arduino can reliably find the specific position of the stepper motor that provides the maximum radiated power and thus the minimum SWR.

System Description

The system is comprised of a stepper motor for capacitor actuation and a simple sense antenna with rectifier, with control achieved by an Arduino Uno microcontroller. These components allow for the control and automatic tuning of any magnetic loop antenna.

Complete System Schematic

The Each system component is described in detail below, however for clarity, the entire schematic is shown in Figure 1 Complete System Schematic.

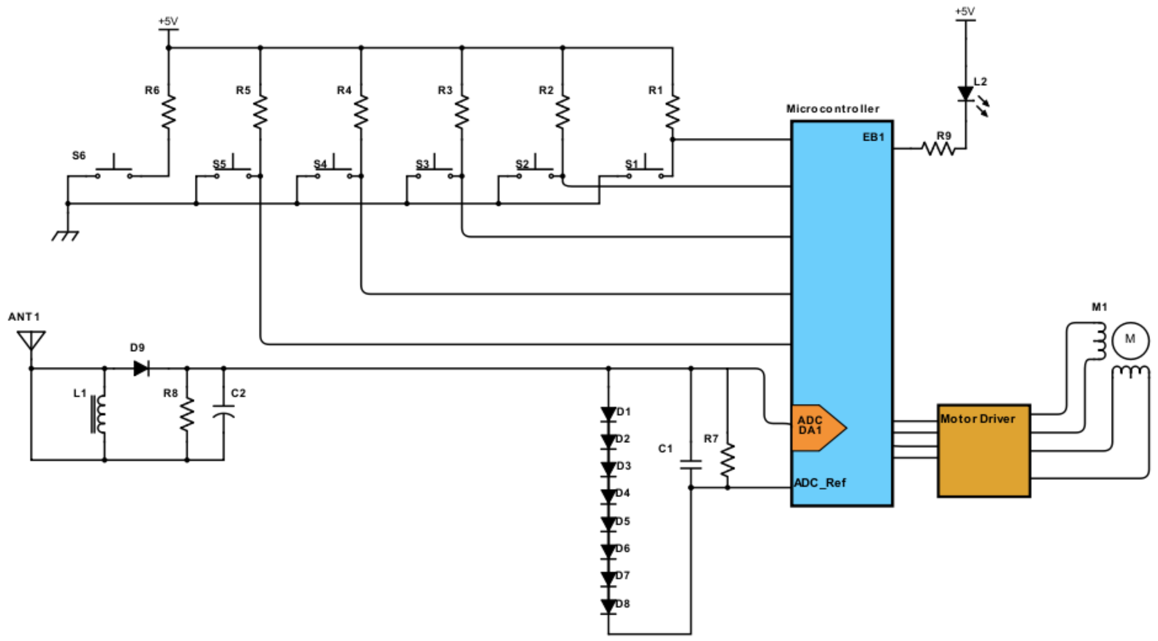

Figure 1 Complete System Schematic

The BOM below includes descriptions of all components that appear in Figure 1 Complete System Schematic.

Table 1 Complete Bill of Materials

Schematic Designation	Part Description
R1-R6	1kΩ, 5% Resistor
R7,R8	100kΩ, 5% Resistor
S1-S5	Off-Mom Pushbutton
S6	Off-Mom Microswitch
D1-D8	1N914 (or other UHF Diode)
D9	1N914
C1,C2	470pF, 50V
L1	2.5mH, 400mA Inductor
L2	Red LED

The H-bridge motor controller must be selected to match the specifications of the stepper motor; this is left to the experimenter.

Sense Antenna/Rectifier

The sense antenna need not be large. The rectified voltage can be adjusted somewhat by changing the position or the length of the sense antenna. The Arduino Uno ADC can measure DC voltages between 0V and 5V, if the analogue reference voltage (voltage against which the measured voltage is compared) A_{ref} is also 0V.

Sense Antenna Schematic

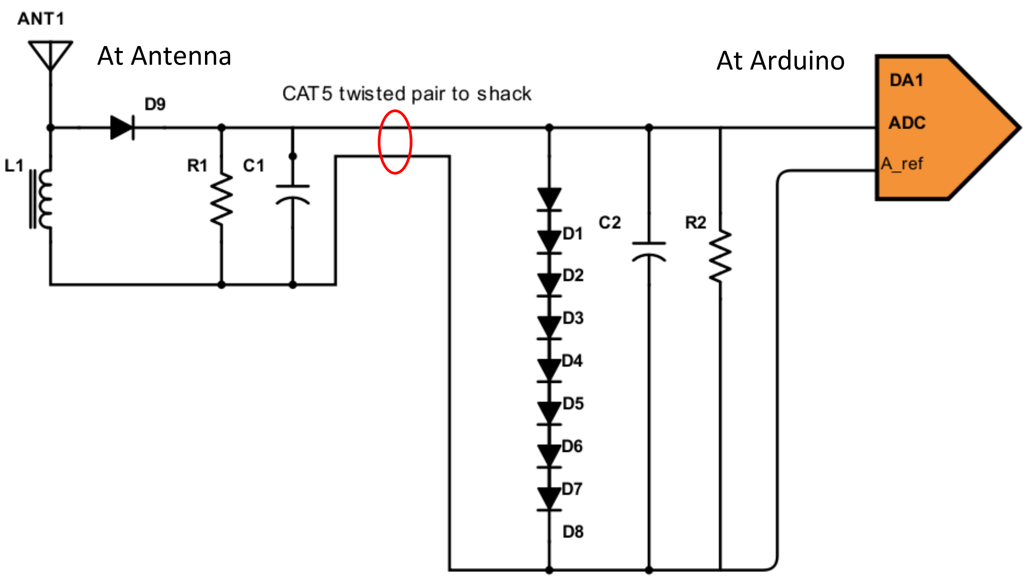

Figure 2 Sense Antenna Schematic

Table 2 Sense Antenna Components

L1	C1,C2	R1/R2	D9	D1-D8
2.5mH	470pF	100kΩ	1N914	1N914

The vital aspect of this circuit is to ensure that L1 is sufficiently large to provide an RF ground on the negative side of C1, and for C1/C2 to be large enough to provide a steady rectified output, but not so large that the circuit is slow to respond to changes in the rectified voltage. The values listed in *Table 2 Sense Antenna Components* are provided as examples that were used during the development of this design. If the sensed voltage exceeds the 5V range of the ADC, a trimmer pot can be used to divide the voltage down into a useful range. The diode stack D2-D9 will begin conducting when a voltage exceeding $8*V_{forward\ diode}$ is exceeded at the input to the ADC. Assuming a typical $V_{forward\ diode}$ of 0.6V, eight of these diodes in series will begin conducting at 4.8V, which is sufficient to protect the Arduino's ADC input from excessively high voltage during tuning or, more importantly, during full-power operation, when rectified voltages can be much higher.

Stepper Motor and Capacitor

The capacitor used in this project is a junk-box Hammond Canada air-variable from an old home-made antenna tuner, rated at $3kV_{pk}$ and with a measured capacitance range from ~50pF to ~500pF. Based on online calculations[1] this should tune the STML to resonance on 40m up to 20m. Additional capacitance could be added to increase the range down to 80m, but the efficiency approaches 1% on that band given the dimensions of the radiating element. The design of the loop antenna itself is outside the scope of this report, however several online resources are available to aid the experimenter in coming up with an approximate set of dimensions and required capacitance ranges for any given frequency and efficiency.

Because of the high Q of a STML, and because air-variable capacitors traverse their entire capacitance range in 180 degrees of rotation (whereas vacuum-variable capacitors require dozens of turns to move from maximum to minimum capacitance), precise control over the capacitor position is required. Direct-coupling a typical 1.9 degrees/step motor to the capacitor would not have provided sufficient resolution. Therefore, a 6:1 gear reduction was used to couple the capacitor to the stepper motor, increasing the resolution. Electrical isolation of the stepper motor from the live capacitor rotor shaft was provided by a surplus ceramic disk shaft isolator taken from a linear amplifier. The capacitor and stepper motor drive module as well as the shaft isolator are illustrated below in Figure 3 Capacitor and Drive Module.

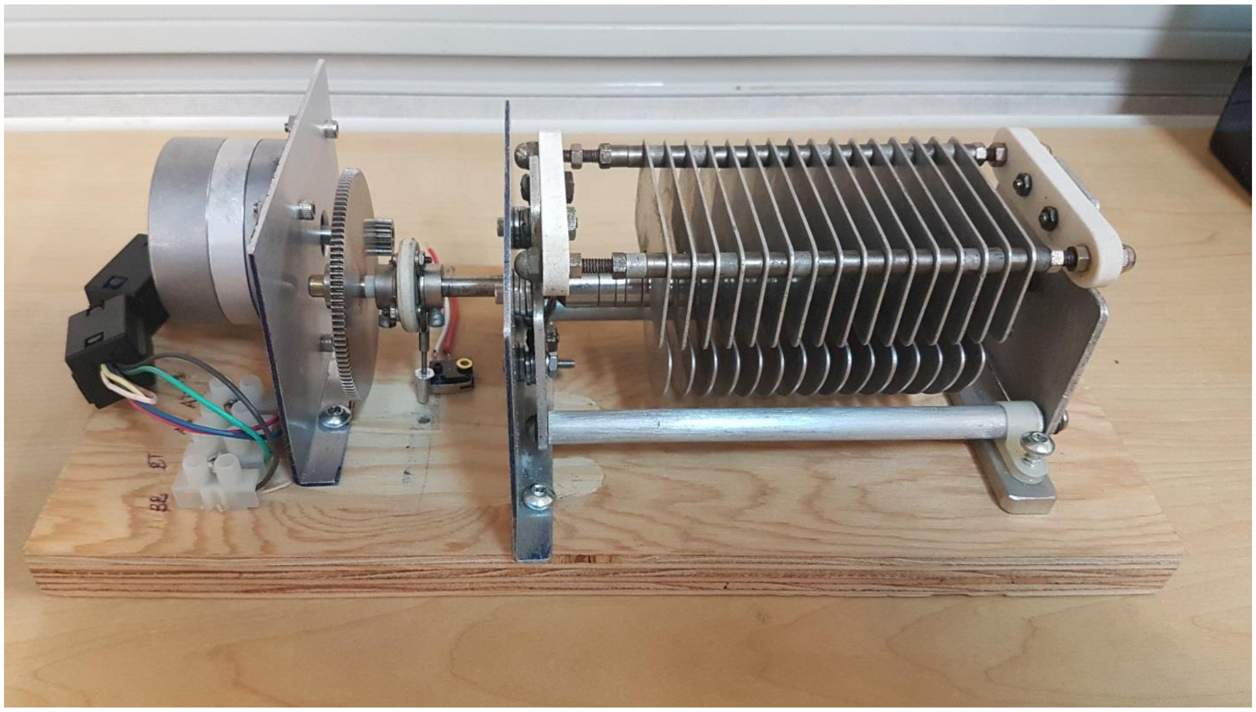

Figure 3 Capacitor and Drive Module (mounting plates by Jeff VE1ZAC)

To speed the operation of the tuner, the tuning algorithm must be able to move to pre-programmed capacitances and tune in a narrow band around that region. To accomplish this, the microcontroller needs to have a reference “zero” position that corresponds to a consistent capacitance. A simple Off-Mom microswitch and actuator-arm made from a lug and rivet integrated into the ceramic drive isolator allows the algorithm to automatically zero the capacitor during initialisation, thereby allowing specific capacitances to be mapped to stepper motor indices. Detailed photographs of the microswitch and actuator appear below:

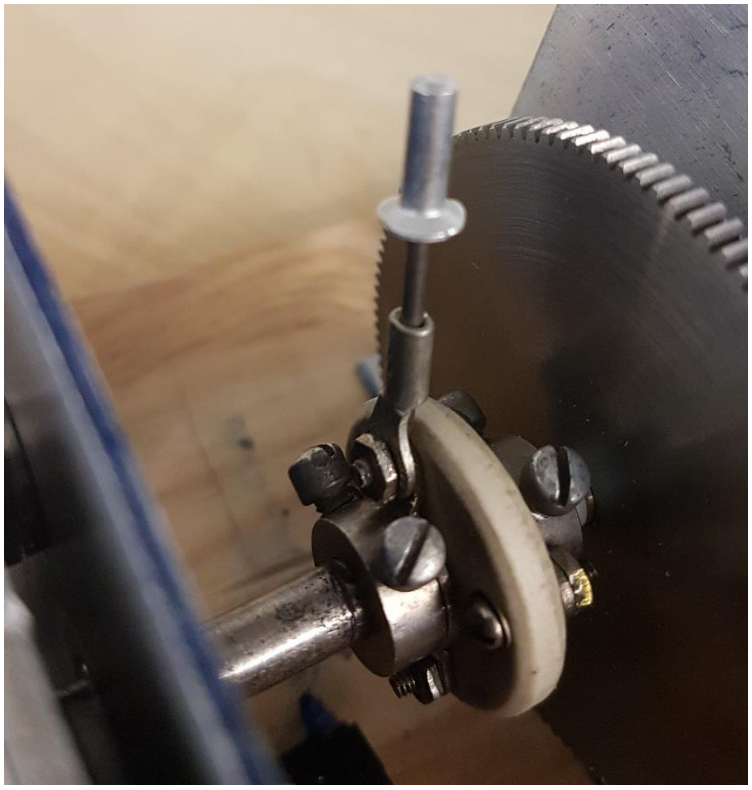

Figure 4 Microswitch Actuator Arm

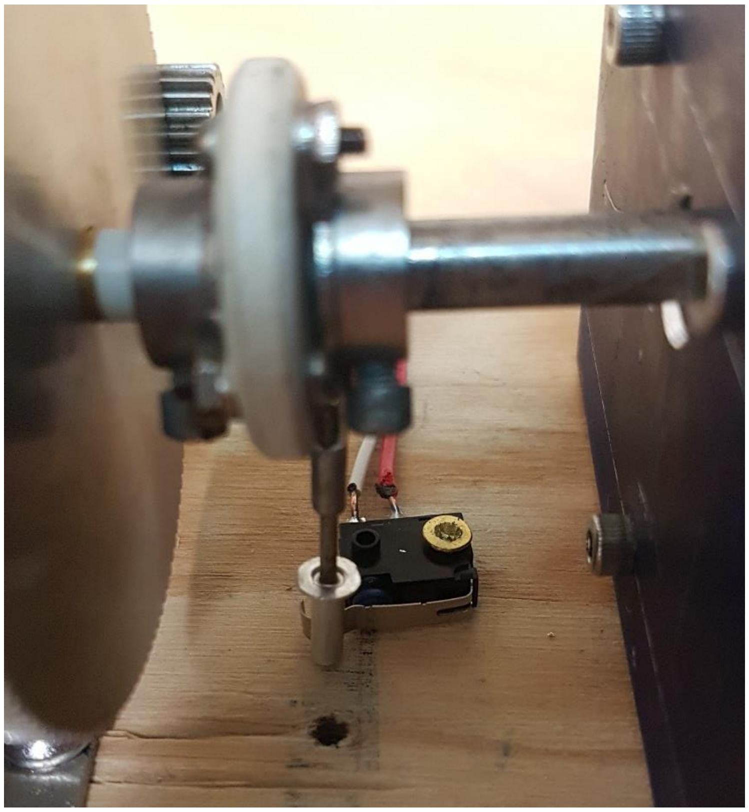
Figure 5 Microswitch Assembly

On initialisation, the microcontroller simply rotates the capacitor counter-clockwise at full speed until the microswitch is depressed. This consistent motor position becomes the new “zero” reference.

System Operation

The tuning algorithm is sufficiently simple to allow for fast, multi-band, precision tuning, but still operate on an inexpensive (~$5 USD) Arduino Uno replica board. The basic functionality of the tuning algorithm is described in this section. The code is heavily commented, with any user-specific variables clearly labelled for simple adaptation to anyone’s specific use-case. The code is available at the following link, and is free for anyone to use and modify as they wish:
https://create.arduino.cc/editor/JMardling/67243a05-af85-4b7b-832c-b67a56f8b77c/preview

List of User-Customisable Variables

The following variables are set by the user to either customise their implementation (number of available modes, motor speed) or to tailor the algorithm to their specific hardware (number of steps in one motor revolution, maximum number of steps allowed, etc.) These variables can be changed by the user without requiring any additional modifications to the source code, provided that the new value falls within the range specified. Note that while other variables can also be modified, they may require additional modifications to the source code to work correctly with non-default values. The variables appear in the source code under the heading “User-Defined Variables” and are outlined below in Table 3 User-Defined Program Variables.

Table 3 User-Defined Program Variables

Name	Default Value	Function	Min Value	Max Value
memScaler	5	Divides step numbers down to fit in a single EEPROM address	1	Unbounded
maxSteps	1200	Number of steps required to reach max capacitance from min capacitance "zero" position	1	Unbounded
stepsPerRevolution	200	Number of steps in a single revolution of the stepper motor	1	Unbounded
fastDelay	10	Time delay between successive motor steps (lower value is faster)	Motor Dependant	Unbounded
slowDelay	100	Time delay between successive motor steps (higher value is slower)	Motor Dependant	Unbounded
searchSteps	50	Number of steps above and below the user-defined starting position for a given mode that the algorithm will search for an optimal tune (Larger number results in wider tuning range for given mode, but slower tune time)	1	maxSteps
numModes	5	Number of modes allowed for storing tune-positions. (See *Table 5 Sample Modes and Frequencies* for example)	1	512

Status Indicator LED Codes

The microcontroller communicates with the user using a single red LED, which blinks with a frequency of 3Hz. Most blink codes have only one possible meaning, however others can have multiple meanings. Note that when the mode button is pressed, the number of subsequent blinks indicates the currently selected mode. The number of possible modes is user-adjustable (**numModes)**, and the LED indication will automatically expand to cover all available modes. The following table outlines all possible LED blink-codes and their meanings

Table 4 LED Status Flash Codes

Number of Blinks	Primary Meaning	Secondary Meaning
1	Tune Button: Beginning Tuning Procedure	Mode Button: Mode 1 Selected
2	Init: Capacitor Zeroing Begin/End	Mode Button: Mode 2 Selected
3	Tune Button: Tune Procedure Complete	Mode Button: Mode 3 Selected
4	Write Mode Button: EEPROM write complete	Mode Button: Mode 4 Selected
5	Mode Button: Mode 5 Selected	
10	Write Mode Button: EEPROM write error / mode not saved	
Continuous Flashing	Manual Move Buttons: Minimum/Maximum Capacitance Reached	

Initialisation Procedure

The operation of the tuning algorithm is best described by outlining a typical user interaction. On start-up, the controller rotates the capacitor anti-clockwise at full speed until it detects that the zeroing microswitch has been actuated. At this point, the rotation stops, and that position becomes the "zero reference" position for the tuner. All viable capacitor positions are between the zero position, and the step position representing the number of steps required to achieve maximum capacitance. Rotation is not allowed below the zero point, or above the maximum point.

On start-up, the microcontroller reads in the "starting tune-positions" corresponding to each of the 5 available modes (the number of available modes can be expanded by modifying **numModes**) from EEPROM. In this way, once the correct starting position corresponding to a given mode is set by the user, it can always be recalled, even after power cycling the controller. **By default, all positions are "0" until set by the user.** The **starting position corresponds to a capacitor position that is within the tuners seek range of the correct tuned position.** This range is set to +/- 50 steps by default, but can be modified by changing the **searchSteps** variable. A larger seek range means that a successful tune can be achieved on a given mode across a wider range of frequencies, at the expense of tune time. If the operating frequency has changed so much that a tune cannot be found within the step-range of the starting position of the corresponding mode, then a new mode can be programmed by the user to cover the extended frequency range. Though this is loop-dependant, see *Table 5 Sample Modes and Frequencies* as an example of a typical distribution of modes and resulting frequency coverage.

Table 5 Sample Modes and Frequencies

Mode	Minimum Tunable Frequency	Maximum Tunable Frequency
1	7.000MHz	7.080MHz
2	7.080MHz	7.170MHz
3	10.100MHz	10.180MHz
4	14.000MHz	14.080MHz
5	14.050MHz	14.100MHz

Manual Operation

As in a typical manual remote magnetic loop tuner, the capacitor can be controlled manually without impacting any saved mode information. Regardless of the current mode, the capacitor can be adjusted manually using the **CW** (clockwise) or **CCW** (counter-clockwise) buttons. Single steps are achieved with momentary presses of either button, continuous movement by holding the button down continuously. When either button is held down, the first 10 steps are very slow for fine adjustments (see **slowDelay** variable), after which the controller transitions to high speed mode for coarse adjustments (see **fastDelay** variable). The capacitor will not rotate below the zero point or past the maximum capacitance point **(Status LED will flash continuously while button is pressed).**

Initial Mode Programming

Modes must be programmed the first time the controller is used, as well as any time the user wants to change the initial tune starting position for a given mode. In either case, the procedure is the same. The mode to be programmed must be selected by pressing the "**Mode Select**" button to cycle to the desired mode. The current mode is indicated by the **Status LED**. The number of blinks corresponds to the selected mode. Once the desired mode is selected, the initial position can be programmed using the following procedure:

1. Set radio to AM mode
2. Set the radio to the desired frequency of operation
    a. If you want a band of operation for that mode, set the frequency to the middle of that band
3. Observe the noise level at this frequency either by ear (preferred method) or using the S-meter
4. Manually adjust the position of the capacitor using the **CW (Clockwise)** or **CCW (counter-clockwise)** buttons (see Manual Operation section above)
5. As the capacitor rotates, listen for a sharp peak in noise level. Stop when the noise level sounds like it has been maximised
    a. This does not need to be exact
6. Press the **"Program Mode"** button once. The **Status LED** will flash 5 times, indicating successful programming of the new starting position for the current mode

Repeat this procedure for as many frequencies (modes) as you desire. Remember that there are 5 modes by default, and if more are needed, this can be changed.

Tuning Procedure

Tuning is performed by following the procedure below:

1. Select the desired mode, which corresponds to the desired range of frequencies
    a. Ex) To operate between 7.000MHz and 7.080MHz, Mode 1 would be required (see Table 5 Sample Modes and Frequencies).
2. Select the desired operating frequency on your transmitter
3. Select "AM Mode" on your transmitter
4. Adjust your transmitter to output minimum tuning power
5. Key the transmitter ON
6. Press **"Tune"** once
7. Wait until the **Status LED** flashes 3 times indicating that the tuning procedure has completed
8. Verify acceptable SWR
9. Un-key the transmitter and begin normal operation

When the tune button is pressed with a low-powered carrier provided by the transmitter, the algorithm first moves the capacitor to the start position corresponding to the selected mode. The correct capacitance should be within +/- **searchSteps** of the start position, otherwise the tuner will fail to find a correct capacitance. After moving to the start position, the microcontroller averages 10 ADC samples of the voltage produced by the sense antenna-rectifier. This serves as a baseline reading for further relative signal strength comparisons. The higher the voltage measured by the ADC for a corresponding capacitor position, the closer the antenna is to resonance.

After determining a baseline rectifier voltage for further comparison, the tuning algorithm begins seeking within the available range of capacitance for an optimum value. Beginning at the starting position, the controller steps the motor counter-clockwise one step at a time. After each step, it again averages 10 ADC samples. For each averaged sample, if the reading exceeds the previous maximum voltage, it stores the new maximum voltage and corresponding position as the current optimal position. After seeking **searchSteps** counter-clockwise, the controller jumps the capacitor back to the starting position and repeats the procedure, seeking clockwise. This seeking procedure is illustrated below in Figure 6 Capacitor Search Algorithm.

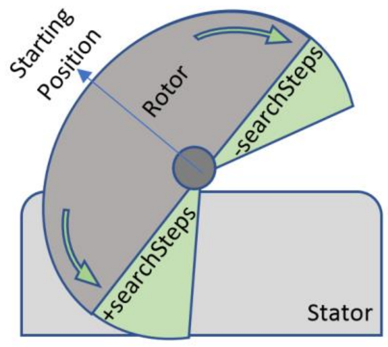

Figure 6 Capacitor Search Algorithm

At the end of the seek procedure, the microcontroller has recorded a maximum voltage reading from the ADC and the corresponding capacitor position that produced the voltage maximum. This position corresponds to the optimal capacitance within +/- **searchSteps** of the starting position for the selected mode; the controller moves to the optimal position, and indicates that it is done by **flashing the Status LED 3 times.**

Note that these voltage measurements are relative, and thus the tuner cannot tell if it has reached an acceptable SWR. If the tune position is not located within the algorithms seek range, it will merely move to the optimal position in that range, which will not result in an acceptable match. Therefore, it is vital for the user to check the SWR before beginning full-power operation.

Appendix A: List of Abbreviations

STML: Small Transmitting Magnetic Loop

ADC: Analog to Digital Converter

(V)SWR: Voltage Standing Wage Ratio

LED: Light Emitting Diode

Off-mom: Switch type, normally off, momentary on

A_{ref}: Analog Reference Voltage

Appendix B: Source Code

The most up-to-date source code is available at the following link: https://create.arduino.cc/editor/JMardling/67243a05-af85-4b7b-832c-b67a56f8b77c/preview

[1] Online Magnetic Loop calculator:

http://www.66pacific.com/calculators/small-transmitting-loop-antenna-calculator.aspx