Theory of Operation

It is suggested that a copy of the schematic be printed out as reference guide to this description of the High Powered PEMF Theory of Operation. The Arduino Nano that is employed in this configuration is the mastermind component that makes this configuration possible. It monitors and controls the complete sequence for generating pulses and controlling everything. The Arduino determines Speed and Timing settings and communicates the operation status through a single multifunction LED.

The PEMF has a master on off switch and an inline 8 Amp SLOW BLOW fuse. All components are rated well above the 8 amps so it is possible to use a 10 Amp Slow Blow if necessary. Anything less will pop the fuse and anything more would be unsafe.

The very 1st requirement of the circuit is to monitor the incoming AC line and detect when there is a Zero Crossing. This is accomplished by the following components; D1, R1, D2, R2, R3, Q1, and R4. R4 is a potentiometer and adjusted to deliver a 5 volt Zero Crossing square wave pulse to the Arduino.

D3 is used to deliver only the positive half of the incoming AC to SCR1. Under software control and very critically time, SCR1 is fired through D4 by the Arduino. R3 is used to create a minimum current draw to force the SCR1 to fire on and off cleanly.

D5 insures that C1 charges through D5 and discharges through SCR2. D6 helps control the back EMF from the coil so the polarized capacitor (C1) is not reversed biased. SCR2 is critically fired through D7 by the Arduino. When SCR2 is fired and depending on the exact RC time constant of the C1 capacitor and the coil resistance, the pulse will last approximately 3 milliseconds. Under Arduino control this discharge voltage source is completely from C1 and not from any AC voltage. This is because SCR1 has been turned off and stays off after the capacitor has been fully charged.

The capacitor is charged by 3 critically timed pulse from the incoming half wave rectified 60 AC source. This is done to help minimize the current draw surge by the capacitor when charging. There is no current limiting resistor employed and this PEMF will cause poorly designed CFL and LED lighting sources to flicker. However, a well designed house hold LED will not noticeable flicker unless you have millisecond vision acuity.

The Arduino software speaks for itself on how and when the required firing sequences are accomplished. Out of the 60 positive half cycles available every second, 5 pulse times are used for the combined charge and discharge cycle. 3 pulses to used to charge capacitor, 1 pulse time to discharge the capacitor through the coil, and 1 pulse time lost because of software overhead.

This is the reason for a maximum 12 hertz frequency limit. NOTE: the Arduino software was written for a 60 cycle environment. A 50 cycle environments will necessitate a small modification to the Arduino sketch. Sketch is geek talk meaning software program. This of course will limit this PEMF circuit to a maximum of 10 pulses per second.

RSW1 and RSW2 are six position rotary switches used to give the Arduino sketch 6 choices for Speed and Run Time control. Trm1 and Trm2 are thermistors mounted on SCR1 and SCR2 respectively. The thermistors are monitored by the Arduino to see how hot the SCRs are getting. At approximately 60 degrees Celsius the unit will shut down and turn the LED on solid. The LED is normally flashing at the speed selected rate when the PEMF is running. The LED is off in all other conditions.

The single button switch SW2 is used to start and stop the PEMF run cycles. It will turn off the LED if the LED is in a continuous on state from heat. Yes, you can restart the PEMF by pressing the button switch and a new run state will commence. However if the SCRs are already close to the 60 degree Celsius limit the PEMF will quickly shut down and not reach the requested runtime settings. So give the unit time to cool down if it stops because of thermal protection.

The SCRs are rated for 150 degrees Celsius so as can be seen there is a large safety margin being incorporated in this PEMF circuit design. This was done to help insure a long component life time. It is possible that I may have been too cautious and the 60 degree limit could be raised to something higher. This will be determined when more runtime data has been accumulated. It might be possible to raise the limit to 90 or possibly 100 degrees Celsius.

The 12 volt power supply is a self contained module and its purpose is obvious. It provides 12 volts to run the Arduino.

Suggestions on what could be documented to improve the clarity of this technical description are welcome.

Respectfully,

Steve