

KL6M AUTOMATIC POLARITY ROTATION SYSTEM

By Mike Melum, KL6M, June 2023



My Earth-Moon-Earth adventure started in mid 1999 here in Anchorage, Alaska on 144 MHz. With my meager yagi array I was able to work the two monster stations, W5UN and KB8RQ. I quickly learned that Faraday rotation in the ionosphere is a huge problem with working 'normal' size stations. I changed bands to 432 MHz later in 1999 so that I could more easily build an array of multiple yagis.

With sixteen 2.2 wavelength NBS yagis and 600w I was able to work 50 CW initials in less than one year on 432, but I specifically concentrated on stations either very large, or with polarity rotation capability. I often had a very difficult time working fixed polarity stations, if I was able to work them at all.

In 2000 I constructed my 30 foot dish system and one of the top requirements was to be able rotate polarity and what a HUGE improvement that made in QSO rate (along with 30dBi gain of the dish). So for 20 years of 432 EME operation I roughly compensated for Faraday by realizing the difference between received signal angle and geometric angle and rotated my transmit angle that many degrees on the other side of the geometric angle. In other words:

$$T = 2D - R$$

Where: T is my transmit polarity angle
D is geometric angle between me and the desired station
R is the angle where I receive the best signal

The following example in Figure 1, demonstrates simply how this works:

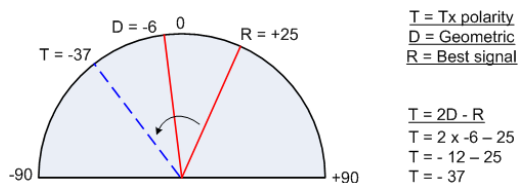


Figure 1. Sample, that demonstrates $T=2D-R$

I personally use a program called SkyMoonWin written by Dave, W5UN to determine the other station's geometric polarity, as shown in this screen capture below. With this particular program, the other stations' grid can be entered to pin point his location, or one can left click in his approximate area to place it on the map. Then the geometric polarity difference is indicated on the lower right, which I have highlighted with a square.

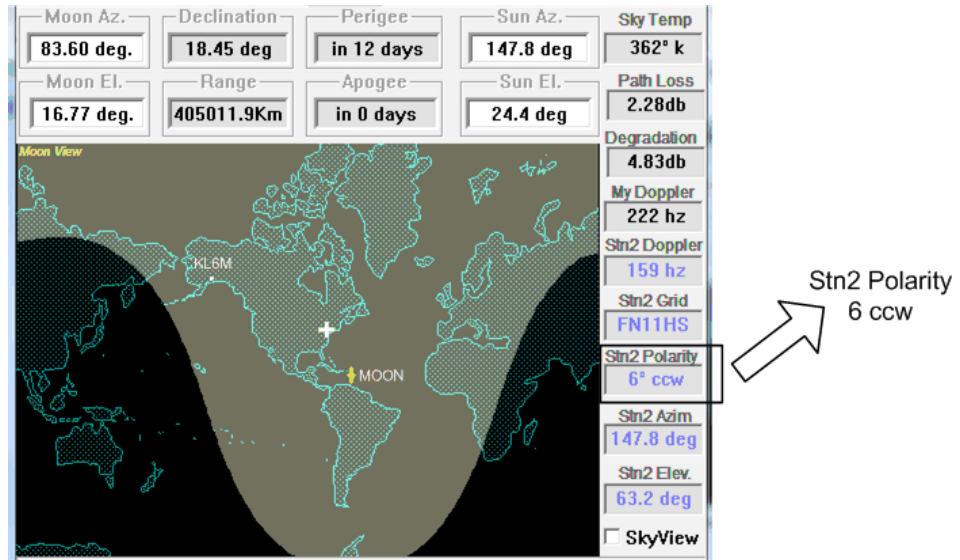


Figure 2. Screen shot of SkyMoonWin (by W5UN)

I thought about automating T=2D-R for years and a couple years ago I finally designed a system to do this. I decided to use a PIC 16F690 microcontroller and embark on some C-language programming.

Block diagram of my system is below in Figure 3:

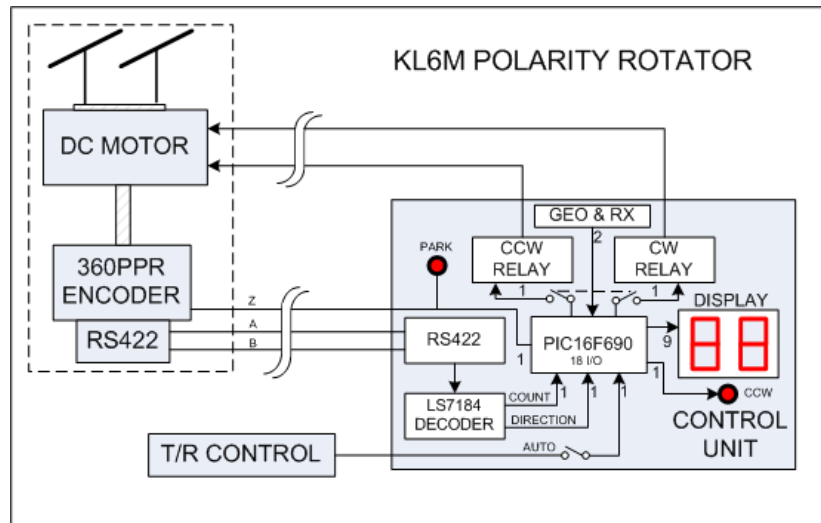


Figure 3. System Block Diagram

The PIC 16F690 has exactly the number of I/O ports needed for this project. I'm now wishing I had chosen a device with more ports just in case, but this seems to be ok. I did a wire-wrap breadboard prototype first and then drew up a layout in KiCad and sent away to JLCPCB in Hong Kong for some boards, amazingly economical, shown in Figure 5 .

I flow charted the steps needed to program this project as shown below in figures 7 & 8. I designed the code step by step, first getting the LED displays working in a loop. I interrupt the display loop whenever I get a change in the count from the incremental encoder. The operation of the system is as follows, using the first example of T=2D-R in Figure 1, and referring to the front panel shown below in Figure 4:

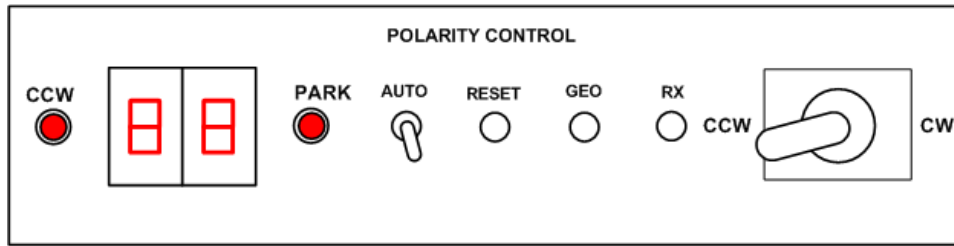


Figure 4. Control Panel

CCW and Park are LEDs, CCW is essentially a minus sign. The display will indicate +/- 99 degrees. Auto is a toggle switch that enables automatic Faraday compensation by sensing the T/R switch. Reset, GEO and RX are push button switches, and the right most toggle is for manual control.

Step one is to determine the geometric angle. This assumes both stations are horizontally polarized. Look at the SkyMoonWin in Figure 2 above for the location of the desired station. The plus is approximately on New Jersey and the polarity difference is 6 degrees counter clockwise. Manually rotate the feed to -6 and press the "GEO" button on the control panel to store the position.

Step two is Rotate the feed in search of the best signal, and for example purposes, shown at 25 degrees clockwise as shown in the example in Figure 1, and then press the "RX" button on the control panel.

The processor in the control panel will now compute the best angle to transmit to a station in the New Jersey area using T=2D-R, and if the "Auto" switch is selected, will automatically rotate to that position when the Transmitter is enabled. When returning to Receive, the rotor will automatically return to the optimum RX position.

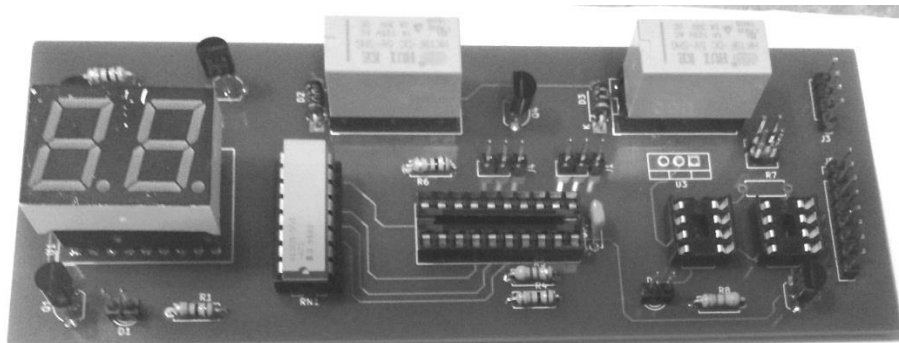


Figure 5. Finished circuit board

Here are some more sample cases that demonstrate how to keep the result within the range of my motor from -90 to +90 degrees:

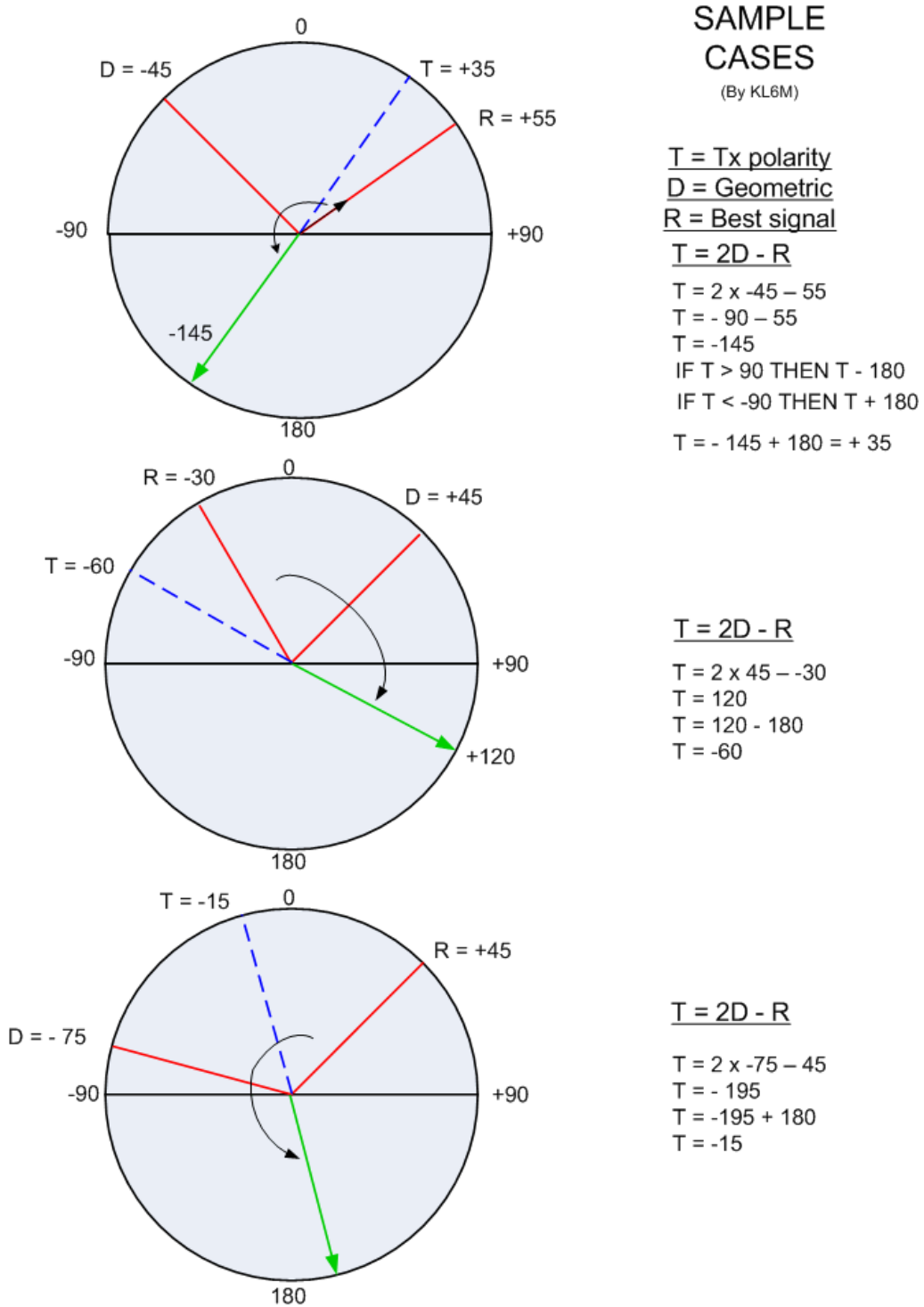


Figure 6. More Sample Cases

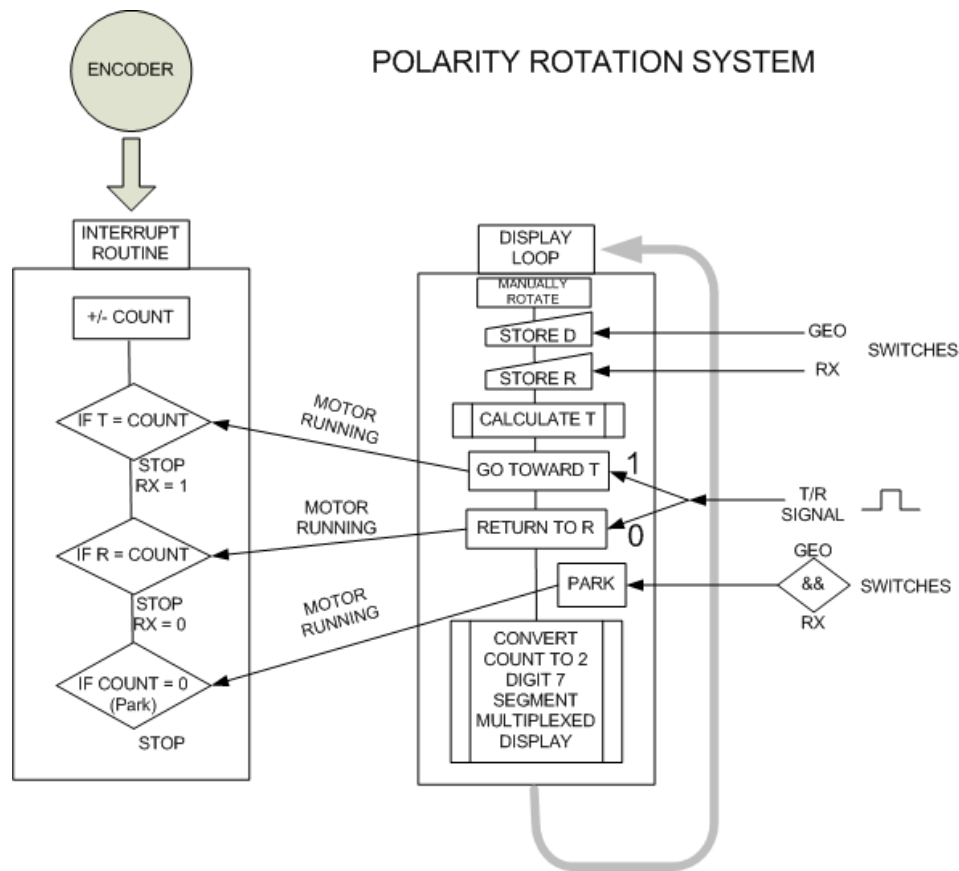


Figure 7. Overall Flow Diagram

DESCRIPTION:

The program loops continuously updating the two 7-segment multiplexed digits. A momentary-Off-momentary switch allows manually moving the DC motor clockwise or counter clockwise.

An incremental quadrature 360 ppr encoder is attached to the motor and sends the A/B pulses through a RS422 line driver to the control unit line receiver to a LS7184 decoder that outputs clock and direction to the MPU.

The clock is connected to the MPU interrupt port. The interrupt counts clock pulses either plus or minus. The encoder outputs a Z-pulse at zero degrees which resets the count on every zero crossing to eliminate errors from noise etc.

A "Park" routine senses RX and GEO switches pressed simultaneously and sends the motor to zero position where, within the interrupt routine it cuts off the motor relay. This is a function typically executed when shutting down. All of the above works great on the prototype using the polrotW.pdf program that can be found on the web url mentioned below.

The Automatic Faraday Compensation routine works as follows: I manually rotate to a specific GEO position and press the GEO button to store the count at that position.

Then I search for the polarity of the strongest signal, and then press RX to store that position. The MPU then calculates T which is the optimum transmit polarity. No action is taken until an external T/R signal shows up at RA0 when "Auto" is enabled.

Then the motor travels to the T position and stops there for the duration of the T/R signal. When the signal stops, the motor returns to the RX position and terminates the process until another T/R occurs.

Note: All motor initiations and GEO and RX store functions need to start within the display loop and the "stop" code must be within the interrupt routine.

Another flow chart was developed to implement the automatic Faraday compensation functions:

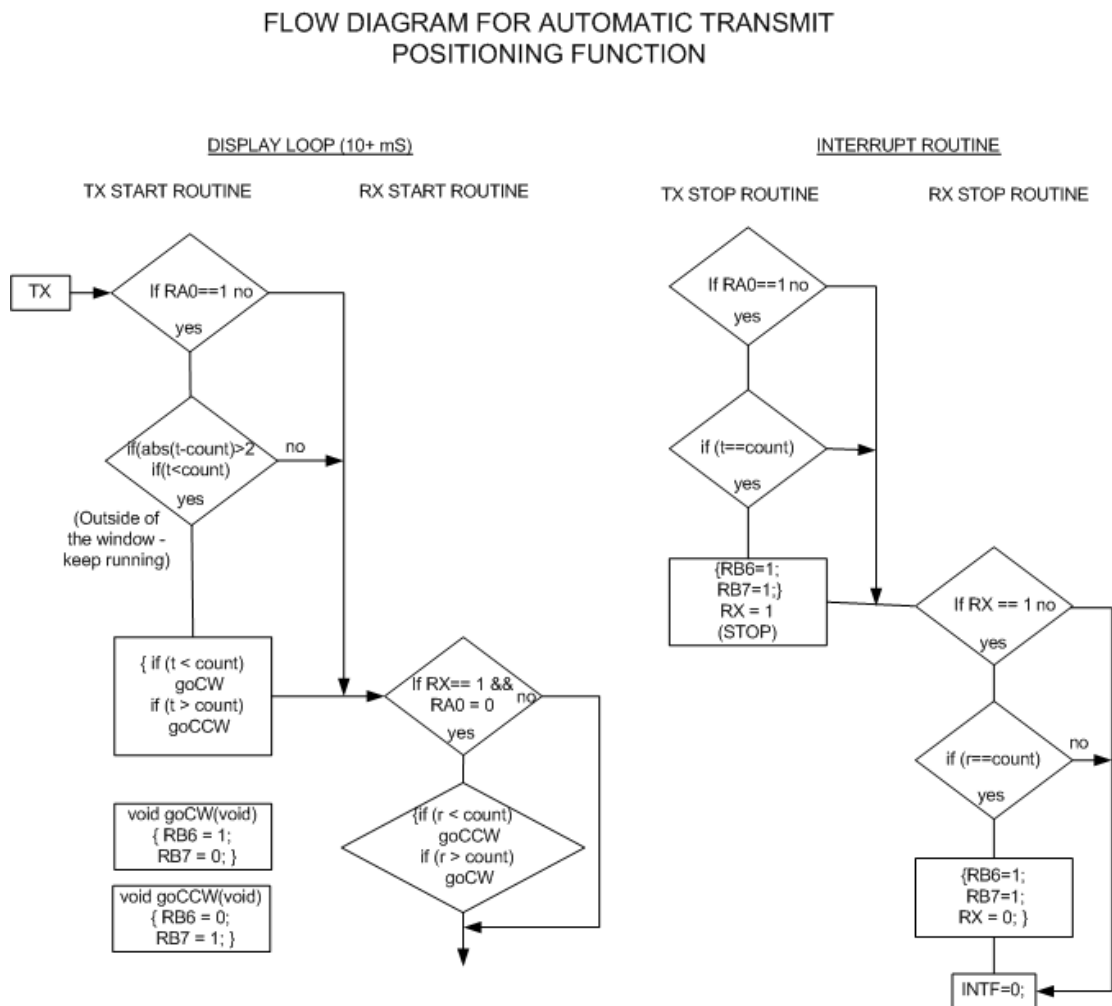


Figure 8. Flow chart for the automatic Faraday compensation

I decided not to include the C-code here since it is about four pages long, but it is available on the web along with more detail and better photos at the following url: <https://kl6m.com/polarity/Polarity.html> Also, many thanks to my friends, Bob, NL7S and Richard, AL6P for their assistance on this project.