This control is designed for managing 4 different loops antennas. You can manage every antenna without interfere in the rest. The power supply is 12v. This is not a commercial design it is made for a ham amateur only for the enjoy of the rest of the community.

The controller can manage 4 different loops antennas independently.

It has 64000 step for every antenna

14 memories for antenna.

You can define up limit and down limit.

!!!! VERY IMPORTANT!!!

The controller has 4 memory banks (1 memory bank for antenna). If you want to erase a memory bank push UP & DOWN buttons simultaneously.

Just in case you need to erase the whole data push DOWN & MENU buttons simultaneously.

The controller has five push buttons:

MENU –this button selects between MEM/ANT/SAVE/ADJUST/BACKLASH/SPEED/DISABLE POLOLU AND MICROSTEP functions.

UP/DOWN – used for the next functions:

-Increase and decrease manually the stepper motor (normal and adjust functions).

-Save memory in the save memory function

-Modify backlash/speed/micro step and disable pololu functions.

MEM UP/ MEM DOWN – used to select the memories and to change the antennas.

All the functions return to MEM function after 3 or 8 seconds.

Functions:

--MEM—

In this position you can select the desired memory. If you don’t have any number stored, NO DATA will be shown in the display. Remember that MEM14 is the upper limit. You need to store in this position the maximum step you want to move your capacitor. For select a memory push MEM UP / MEM DOWN.

--ANT—

In this position you can select the antenna between 1 and 3. For choosing an antenna push MEM UP / MEM DOWN.

--SAVE—

Once SAVE is shown in the left corner, you must select the desired number of memory (between 1 and 14) and push UP or DOWN buttons to save.

After this will appear a new screen in which you can save the frequency. Introduce the frequency this way:

-Buttons UP & DOWN to select MHZ (1000 KHz) Up to 59 MHZ

- Buttons MEMP & MEMDOWN to select KHZx100 Up to 59 MHZ

-Rotary encoder to select KHZ.

-Push MENU button to save the frequency or wait 4 seconds.

Remember that this is only a tag not a real frequency.

Remember that in position 14 you must save the upper limit.

--ADJUST—

ADJUST function allows to move the stepper motor without increasing or decreasing any number in the display. It is useful when we need to find the 0 position manually. Sometimes it is necessary for calibrating stored memories. Once adjusted one of them, the rest are calibrated too.

--BACKLASH—

Backlash compensation from 0 to 200. In this position you select the value you consider effective in your system. In order to not to complicate the software, I have decided to compensate only when decreasing. So If you want to as more accurate as possible, before storing a position:

Ej—step 1750

1. increase a bit more the value ---1765
2. decrease the value to the desired position --1750
3. save it --1750 save

Remember to do this if you want to be accurate in the recorded positions.

Just in case you don’t need backlash compensation put the value in 0.

--SPEED—

This function stablishes the maximum speed in automatic movement (memories ). 3 is the max speed (3milisecons pause in every step) 20 is the min speed (20 milliseconds pause in every step). You must adjust the speed in order not to broke your capacitor. I could have used 1 millisecond but the speed was dangerous for almost every system.

--DIS POLOLU—

Pololu is the driver that is in charge to move the step motor. During its work, pololu introduces a lot of rf noise in the antenna. Some people has designed its system in order to not to be affected by this noise. In case you can’t deal with the noise you can disable the pololu after every movement. This happens automatically if you choose “Y“. In case we chose “N” the pololu never disables. Don’t disable the pololu is more accurate but noisier.

--MICROSTEP—

On the cnc shield you’ll find three jumpers you can set to modify the Microstep.

<https://blog.protoneer.co.nz/arduino-cnc-shield-v3-00-assembly-guide/#Assemble>

Microstep menu uses a compensation to be more accurate when we use micro stepping in the pololu. For no compensation or no micro stepping you can use 0 compensation.