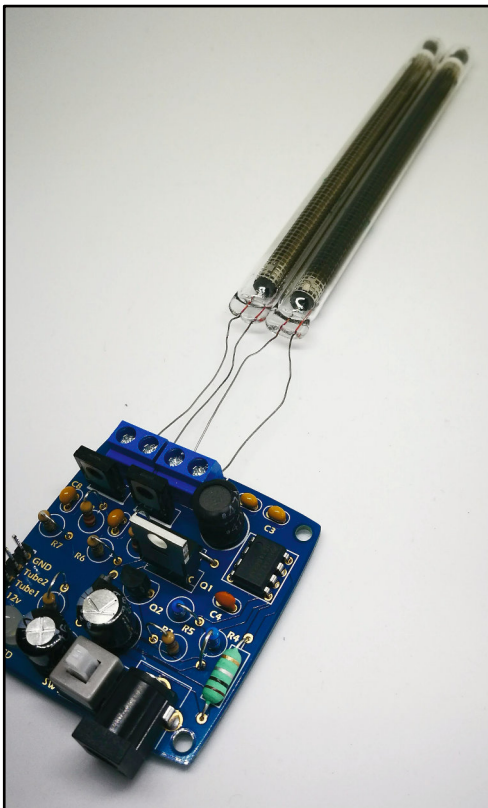Check carefully that your kit looks like mine above if you are happy then you are ready to go ahead and test it.

The Nixie tubes will only work in one polarity, with the front of the tube upward from the table insert the legs and screw into place.

If you apply power at this point you will not see the tubes light up, however, there WILL be high voltage on your board, so you must be very careful not to touch the board.

In order to see the tubes glow you will need to connect an Arduino, microcontroller, Raspberry Pi etc to the Tube1 and Tube2 inputs on the PCB. A very simple test is to connect these to two Arduino pins, connect the ground pin to the ground on the Arduino and use digitalWrite to set them high. This will set both Tube1 and Tube2 to 5v, and the tubes will light up to their maximum height...

"Wait, I did that but the tube is only half lit!"

This is what you will most likely see. These tubes are Russian new old stock (NOS). That means they have been sitting in a box, for decades, unused. Worry not. Your tubes simply need to be "burnt in". By driving the tubes at a maximum for about 30-60mins you will see over time the top of the orange glow creep up towards the top of the tube. Some will move faster than others, but they will get there, you might even need to leave it for a few hours. Once they have reached the top you will no longer need to burn them in and now whenever you use them they will reach their full height.

Once your tubes are burnt in, you are ready to try something more interesting. I have written a couple of example Arduino sketches to help you get going. You can download them here. When you look at these sketches you will see that I am not using the default PWM function, I am in fact using the PWM frequency library, the latest version of which you can download from here. This is for an important reason. Unlike an LED, which can react extremely quickly to changes, the Neon glow discharge inside a Nixie tube behaves quite differently. Making sudden changes can cause the bottom of the plasma to no longer be locked to the bottom of the tube, and instead, float in the middle. This is very undesirable. So in order to avoid this, the PWM signal is filtered before the controlling transistor. This low pass filter smooths the PWM signal but also limits how quickly changes to the tube height can be made. The standard Arduino PWM function uses a very slow frequency of 500Hz. Using a low pass filter for such a low frequency means it is also difficult to make quick changes to the tube height. For this reason, I have opted to use the PWM frequency library, this allows us to use a much higher PWM frequencies. I chose of 10kHz. Now our low pass filter can smooth the PWM signal and allow for quick tube changes.

If, however, you decide you really want to use the Arduino PWM functions then it is still possible. Though you might want to change the values of the low pass filters into the transistors, R6, C7 and R7, C8. By reducing the time constant, increasing the values of R and C, then this will filter the lower PWM frequency, but limit the speed at which tube changes can be made. If you will only ever make small changes, for example, driving a clock, then this might not be a problem. For more information on low pass filters see here.

Thanks and I hope you make something cool with your Nixie Bargraph Driver!