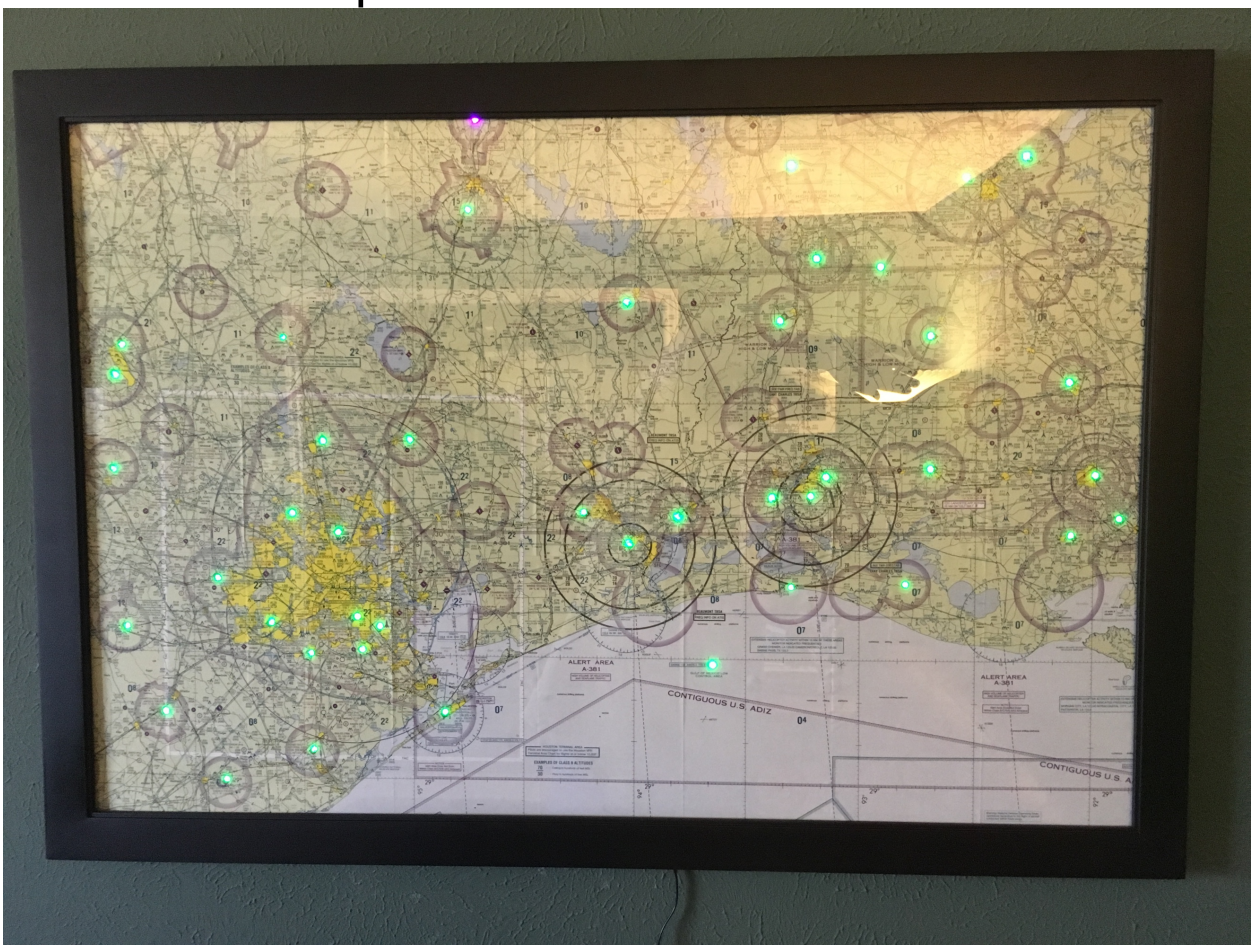
Neopixel Metar Sectional Chart



So, I've had several people ask me to do a true step by step build instruction on making a weather reporting neopixel sectional map.
There are a couple of videos on youtube and some reference available on the web, but was unable to find a complete how to, so here's mine.

A couple of issues I addressed while working on mine included the thickness of the map and frame being used. Seems some of the LEDs guys are using are HUGE. I ordered a set to look at them.

The frames of their charts needed to be 3 inches thick to contain the LEDs. I knew there could be a better way and found another set of LEDs that work very well.

Makes for a very thin chart!

Another is an issue when the internet connection was reset. The program would hang and a couple of simple lines of code now give it an “on err” path to keep it running.

One last comment before we begin. The LEDs I used will run on a 2 amp wall wart, but while playing with the setup on a full size Pi 3B, I had trouble with the large LEDs everyone else uses. If you get a lightning bolt sign in the top left of the screen, that’s a sign that you could use a larger power supply. My test setup has a 10amp PS just to make up make sure I didn’t have issues. That said, the one on my wall runs fine on a 1 amp wall wart lighting 57 LEDs.

Required Items:

CanaKit Raspberry Pi Zero W (Wireless) Starter Kit with Official Case - 8 GB Edition (Amazon)

BTF-LIGHTING 50x WS2812B Pre-soldered Black leds with Black wire 5V WS2812 IC Built-in & 10cm Cable Addressable Idividually LED Module String Panel (Amazon)

Map(s) of the area you want to cover (2 maps are required to cover a sectional, but you can use multiple sectionals to get the coverage you require. ie if you fly from an airport on the edge of the sectional, you might want the adjoining sectional to give equal coverage around your base of operations. National maps or IFR charts are another possibility)

Materials:

Foam Board 5mm

Frame for hanging (24x36 works well)

Spray Glue

Shoe Goo

Masking tape

Velcro

Toothpicks

Computer Tools:

Powered USB hub

Keyboard/Mouse

Adequate power supply 5v 2amp min

HDMI capable monitor with HDMI cable

Hand Tools:

Nippers

Drill

½ in brass tube (homemade hole cutter)

Rolling Pin

Razor knife

Soldering iron fine tip w fine solder


CanaKit


LED STRIP
BTF-LIGHTING®
WE COULDN'T IMAGINE
HOW BEAUTIFUL LIFE WITHOUT YOU
WS2812B LED Modules
SKU:WS2812BLEDW-B12
Cable : Black
Cable Size : 10cm between leds
LED Quantity: 50leds
Working voltage: DC5V
Maximun power: 15W
BTF-LIGHTING
RoHS


ELMER'S
Spray
Adhesive
ALLOWS FOR REPOSITIONING
DRIES PERMANENT
FAST TACK, DRIES CLEAR
11 OZ (311g)


WARNING
ADVERTENCIA:
Original
Shoe Goo
SHOE REPAIR
AND
PROTECTIVE
COATING
REPARACIÓN DE ZAPATOS
Y REVESTIMIENTO PROTECTOR
CLEAR
TRANSPARENTE
3.7 fl oz (109.4 mL)

Electronics:

The first thing we need to do is get our Pi up and running. We need to get it hooked up to the USB hub, keyboard, mouse and monitor. We also will insert the micro SD card that has setup software on it. When the Pi boots the first time it will boot to NOOBS which is an OS installation tool. On the bottom of the page select your location (USA) and then select wifi networks and enter the required wifi info. This will allow the Pi to connect to your network. Once connected we are going to install Raspbian as the OS, and then click install and yes at the confirmation/warning box.

Now go fix a sandwich or polish a spinner or something for a bit.

When you return, you will be notified that the OS has been installed. Click OK leads to reboot.

Once rebooted, Open the terminal program. Fourth from top left.

Next, we will update the Pi's software catalog using the terminal program. Start by opening the terminal program. From there we are going to enter the following lines.

Red is actual entries!

Type sudo apt-get update wait

Type sudo apt-get upgrade, confirm Y wait

Type sudo apt-get dist-upgrade, confirm Y wait

Type sudo apt-get install build-essential python-dev git scons swig, confirm Y wait

Type git clone https://github.com/jgarff/rpi_ws281x.git

Type cd rpi_ws281x

Type scons

Type cd python

Type sudo python setup.py install

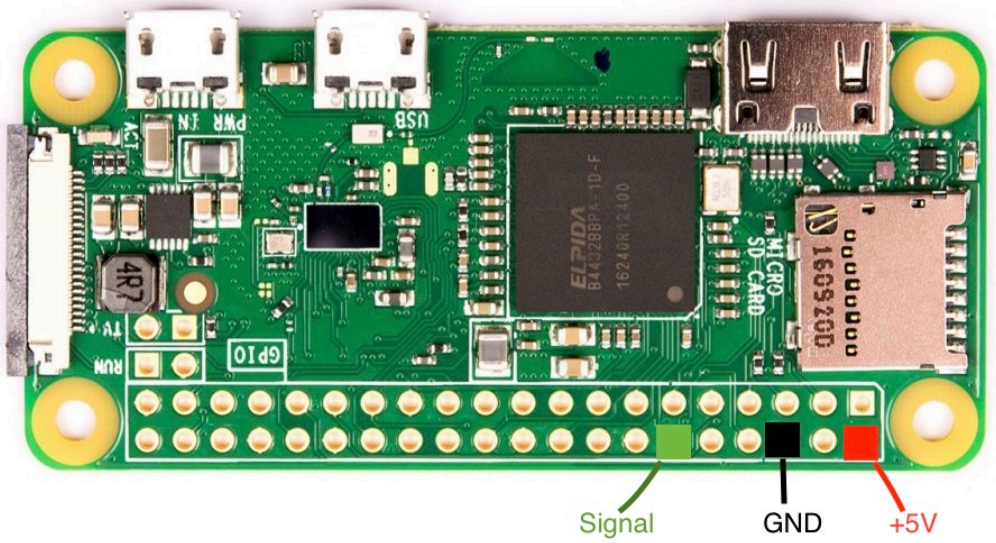
Now we'll shut the Pi down

Type sudo shutdown -h now wait for about 30 seconds

So now you have a new micro computer to love. Unplug all the connections. Now we're going to try to kill it! (hopefully NOT!)

We need to attach the LED string pigtail to the Pi Zero. There is a front to the LED string and the pigtail that goes on the Pi should look like this.

LED strings are typically 50 count and can be daisy-chained. This requires some delicate soldering skills. It might be a phone a friend moment. We need to connect to pin 2 with the positive lead (red,) then pin 6 with the ground (white) and pin 12 with the signal (green.) Pay careful attention to the quality of these solder joints and make sure no two pin pads are bridged with solder!



Now you can plug the Pi back into the peripherals and the LED string and reboot.

If all went well with the soldering, you should now be back to the Pi desktop.

Now the fun begins.

Open the terminal again and type in the following to get the software needed for the map.

Type git clone https://github.com/lhatton/WXsect.git

This will retrieve the software needed to run the map and put it in a new folder named WXsect. You can see the files in the package by changing to the WXsect directory and displaying the contents.

In terminal…. cd WXsect then ls.

The package contains five files:

> “metar.py” is the actual python script that retrieves and displays the led colors.
>
> “airports” is the list of airports that “metar.py” will retrieve in the order in which they are on the LED string.
>
> “strandtest.py” is a program you can use to test the LED string.
>
> “launcher.sh” is an easily edited script program ran on boot to start either “meter.py” or “test.py” or both. By default “metar.py” is the one that will run when “launcher.sh” is run. Editing “launcher.sh” is easier than modifying the boot instructions on the pi.
>
> “slowmetar.py” is a version of metar.py that takes a second between lighting each LED. It can be useful for confirming your “airports” file later in the build.

Now lets see how your LEDs work

Type sudo python strandtest.py

This should present you with all kinds of LED gyrations.

Control-C to stop.

If that worked you can sudo python metar.py to display my list of BPT centered airports on my HOU sectional chart.

Again Control-C to stop.

THE MAP:

Start by finding which side of the foam board you want to be the front. Any bow should be convex looking at the front.

Tape an old map to the foam board, taking into account what airport you want in the center. A couple to four points will do (this is temporary.)

Mark a couple of index points for when you flip the map so as to make it lineup perfect. Then take a tack and poke thru all the airports you want to mount LEDs at. This will leave a small hole in the foam board we can use to locate where we want to cut holes in the foam board. Remember to make sure the airports/locations you mark have WX reporting. Foreflight is a real help here.

Flip the map (or add any additional maps), line up your index marks, tape it down and repeat the marking process.

Now remove the map and use a pen to highlight all your pinholes.

Now here is where some tool fabrication takes place. Take the brass tube and notch the end with the corner of a file in four places. I taped up an old drill bit to about the right size to snugly fit in the other end of the brass and then taped it together to make a deadly foam board hole saw.

Using your new tool and one of the spindles the LEDs came on, drill holes at all the pin mark locations of your foam board. Scrap foam board and a scrap piece of wood underneath make for clean cuts.

When finished you should have a swiss cheesed piece of foam board.

Iron your charts you plan on using for the finished map.

Tape your first new chart at the index marks and make sure your holes match your airports (backlight.) Then lift the non taped side and spray glue the foam board. Carefully place the map back down and use a rolling pin, center outward) to ensure good smooth adhesion. Now take the taped side up and glue down as well. Repeat this procedure with any additional or flip side maps

At this point I mounted my foam board in the frame. The frame and glass made it easier to work with.

Turn the framed chart face down.

Using the placement of your Pi as a starting point, figure out the most efficient path between all your airports temporary taping as you go. It took me a couple of tries before I was happy with the route.

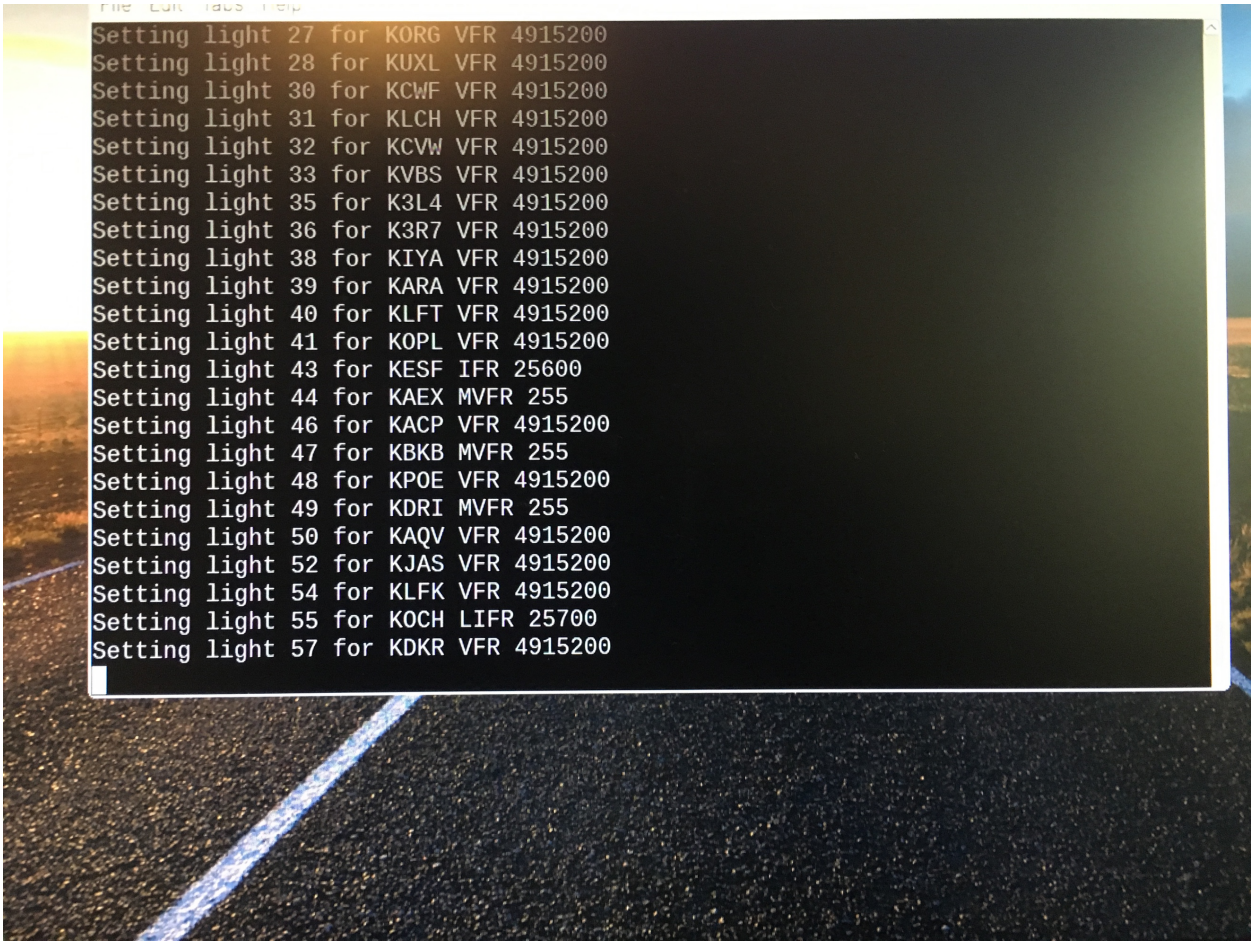
Using the Shoe Goo, glue the LEDs to the foam board. I tried to keep the glue off the chart, just on the back of the LED and the board. Use tape to position the led where you want it and add the glue blob when you are happy with the position. I used a toothpick to make sure the LED is down in the hole occasionally as the glue gets tacky.

Once the LEDS are firmed up, make a list of the LED locations as well as any that were not used (some cities are more than one LED apart.) This list will be edited into the “airports” file on the next step.

Making your AIRPORTS list match your chart.

Using the file browser (two folders on top menu) you can right click the airports file in WXsect with the text editor, you can now list the airports you ran LEDs to. Use NULL where an LED has to be behind the foam board. I've included a "slowmetar.py" file that will illuminate them one at a time to help verify your airports file.

Run it in terminal from the WXsect folder cd WXsect and then sudo python slowmeter.py



The program waits for 10 minutes before it polls the WX service again. Again any time you want to get out of the program Control-C.

I

n the next step we will make “launcher.sh” run on boot.

Again in terminal

Change directories as follows…
cd /home/pi/.config/lxsession/LXDE-pi

edit the autostart file sudo nano autostart

arrow down to the first blank line and add

@lxterminal -e sh /home/pi/WXsect/launcher.sh

Then control-X, then Y, then enter

Now every time you start the pi it will run metar.py or whatever you specify in launcher.sh

Reboot either in the desktop or terminal sudo reboot

That’s it. I’m looking at a few variations, but this one works well, is thin and is easy enough to build.

Les