

METAR SECTIONAL CHART

Gavilan College Aviation Maintenance Program

Written By: Nick Swanson & Herb Spenner

INTRODUCTION:

The goal of this project is to display live weather data on an aviation map. The weather data is extracted, over the Internet, from the Aviation Weather Center (www.aviationweather.gov). The project was inspired by a video on HomeBuilt Help's YouTube channel. This project takes their idea a step further by documenting and simplifying the hardware build, software installation, and software configuration.

(HomeBuilt Help website - <http://www.homebuilthelp.com>)

(HomeBuilt Help video - https://www.youtube.com/watch?v=Yg61_kyG2z)

The Metar Map project is a live aviation map that displays current flight categories (VFR, IFR, LIFR, MVFR) for airports. The project uses a Raspberry Pi to download METAR weather information from the Aviation Weather Center and send the data to a strand of LED lights displaying the appropriate color for each airport on the map. The project is great for someone who wants to learn basic computer and wiring skills, or to just have a cool piece of art to place in your home or office. The software has been written to be plug and play with very minor editing needed to configure which airports are to be displayed.

BASIC PARTS LIST:

(With the dynamic nature to the electronics industry, specific part numbers are not provided. Depending on when you purchase your hardware, you may be buying a newer version.)

Raspberry Pi (<https://www.raspberrypi.org>) - The computer is from the Raspberry Pi Foundation. They are robust, inexpensive single board computers. The model you choose should have Wi-Fi capability. The project was tested with Pi 3 Model B and Pi Zero W models. These are available from many different vendors.

Micro SD Card and USB Reader –At least 8GB Micro SD card and USB reader. See the Raspberry Pi Foundation website for details on the SD card requirements. The USB reader will be used to transfer software from your computer to the SD card. These are available from many different vendors.

Raspberry Pi case – A case is not necessary but it will make it much easier to mount the Pi. (Note: Vendors, like Amazon, sell Pi kits that include the Pi, SD card, reader and case. We found these a very attractive option.)

WS2811 LED String – The WS2811 is a LED controller that enables a computer to control the colors displayed on RGB LEDs. Companies combine the controllers and LEDs in a string of lights that look like Christmas tree lights. We used a string with 50 LEDs. The LEDs were 12mm in diameter and spaced about 3" apart. You will most likely not need all 50 LEDs. You just cut off the ones you don't use. Search the web for "**WS2811 LED String**" and you should find many vendor options.

Level Shifter - The output from the Pi and the input for the LEDs run at different voltages. A level shifter is needed. We use a TXB0104 Bi-Directional Level Shifter TXB0104 from adafruit. The adafruit product id is 1875 (<https://www.adafruit.com/product/1875>). There are other options but we found the TXB0104 inexpensive and most important – easy to install.

5 Volt Power Supply – Make sure the power supply you choose can supply enough smooth power for the Pi, Lights, and level shifter. The hardware needs less than about 5A. We had difficulty finding good power supplies at the 5A level. We suggest you use at least a 10A power supply.

The map, frame and support hardware have many options. We leave these up to your imagination and personal preference.

GOOD LUCK AND HAVE FUN!

Raspberry Pi Installation Instructions

Initial Pi Setup:

The first thing that needs to be done is load the basic operating system and configurations files to an SD card and load it on a Pi.

1. Download Raspbian Stretch **Lite** from Raspberry Pi Website.
<https://www.raspberrypi.org/downloads/raspbian/>
2. Download Etcher Application. This will allow you to write the Raspbian image to your SD card.
<https://etcher.io>
3. Place the SD card in the USB reader and plug the reader in your computers USB port. Open Etcher. Select the Raspbian file for the image. Select your SD card for the drive. Click Flash.
4. Next, you will be creating a ssh (secure shell) key to be enable you to remotely communicate with the Pi. MAC and Linux users can use native editors such as Text Editor. Windows native editors don't support Linux file formats well. We recommend **Windows users use an application called Notepad ++** <https://notepad-plus-plus.org/download/v7.5.6.html> (this will allow you to save the files in a Linux format)
 - Open text editor application
 - Leave the file blank. (no text)
 - Click "Save As", name the file **ssh**, and **save the file to a convenient location**. (The file should not have any extension like .doc or .txt and should be in "text" format.)
 - Move the **ssh** file, you created, onto your SD card's top directory.
5. You will also need to create a file which will activate the WiFi (WLAN) for the Pi.
 - Open text editor application.
 - Copy and paste the text below into the text editor application.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
country=US
network={
    ssid="YourNetworkName"
    psk="YourNetworkPassword"
    key_mgmt=WPA-PSK
}
```

****Note:** (make sure there is a blank line at the end of the file and the file is in UNIX/Linux text format like the one used in step 4. **ssid and psk are case sensitive**)

- Replace "YourNetworkName" with the name of your WiFi network. Replace "YourNetworkPassword" with the password for your WiFi network.
- Click "Save As" and name the file **wpa_supplicant.conf** . (if on a windows machine, make sure to change the file type to Linux type.) Save the file to a convenient location.

- Move the **wpa_supplicant.conf** file, you created, onto your SD card's top directory.

CONNECTING TO YOUR Pi:

Now that you have the Raspbian image, ssh, and network configuration files on the SD card, it's time to connect to the Pi.

1. Insert the SD card into the Raspberry Pi and power it up. (you should see a solid red light and a blinking green light.)
2. Next you will need to determine which IP (network) address your Pi is using, so you can connect to the Pi remotely.
 - Use an IP scanner to find the address assigned to your Pi.

****Note:** We recommend a scanner named Advanced IP Scanner, but you can use whatever you are comfortable with. <https://www.advanced-ip-scanner.com/>

- Press Scan and scroll through results until you find which address has been assigned to the Pi. (write it down)

****Note:** The scanner should automatically populate your current network connection, however if it does not you will need to enter a range for your network (Ex: 192.168.0.1-250)

3. You are now ready to remotely connect to the Pi. For Mac or Linux users, you will be using the Terminal application. Windows users will need to download a ssh application such as Putty. (Found Here) <https://www.putty.org>

3a. MAC/LINUX USERS (TERMINAL)

- Open Terminal
- Type in **ssh pi@192.168.0.101** (Use the IP Address you found using the scanner that is assigned to your Pi.) Press ENTER
- You should then be prompted for a password. The default password is **raspberry** (this can be changed and will be covered in a later step.) Press ENTER
- If you connected successfully, you will be prompted with a message asking if you would like to connect. Type in **YES**. Press ENTER

3b. WINDOWS USERS (PUTTY)

- Open Putty.
- Enter the IP Address you found using the IP Scanner for your Pi into the field that says "host name (or IP Address)
- Make sure the ssh option is selected for the connection type.
- You can save this configuration by entering a name where it says, "saved sessions". Then Click Save
- Press Open. You should now see a separate window open
- Type **pi** and press ENTER.

- You should then be prompted for a password. The default password is **raspberry** (this can be changed and will be covered in a later step.) Press ENTER
- If you connected successfully you will be prompted with a message asking if you would like to connect. Type in **YES**. Press ENTER

Congratulations! You are now Connected to Your Pi.

4. To change the default password follow the steps below.

- While connected to the pi, type **passwd**. Press Enter.
- Type in the current password (the default starting: “raspberry”). Press Enter.
- Enter in a new password. (this will be case sensitive). Press Enter.
- Re-type new password. Press Enter.

The password has now been changed. You can repeat these steps if you need to change the password in the future.

INSTALLING METAR SOFTWARE ONTO YOUR Pi:

The next step will be to download all the software and scripts for the lights to display the Metar information.

1. Connect to the Pi. Enter the following text into your command window(Terminal/Putty). This updates the base Pi OS and install git (a software control system).

sudo apt-get update

sudo apt-get -y upgrade

sudo apt-get -y install git

****Note:** All the commands (bold text) in these steps are case sensitive. You MUST copy them exactly as they are written here.

- Press ENTER (IF you are prompted with any questions during Installation, answer YES)
2. Next, download the Metar script by entering the following command.

git clone https://github.com/GavilanAviation/MetarMap

- Press ENTER (IF you are prompted with any questions during Installation, answer YES)

3. Then finally, install the script by entering the following command.

sudo MetarMap/installscript.sh

- Press ENTER (IF you are prompted with any questions during Installation, answer YES)

TESTING THE SOFTWARE:

This will test if the software has been installed correctly. Before running any of these commands, insure you are in the Metar Map directory (enter the command: **cd ~pi/MetarMap**)

1. Test the application installation by entering the following command.

./runmetarscript.sh AirportMMCheck.py

****Note:** It may take a second for the script to download the data, but you will see each airport you entered on the configuration show up with its current flight category as seen below.

The screen should look this:

KMRY

Airport: MONTEREY; ID: KMRY; Flight Category: VFR; LED #: 1

KSNS

Airport: SALINAS; ID: KSNS; Flight Category: VFR; LED #: 2

KWVI

Airport: WATSONVILLE; ID: KWVI; Flight Category: VFR; LED #: 3

•
•
•

If you are having problems at this point, check the Pi has access to the Internet and that you have completed all the software installation. To check if you are connected to the Internet, enter the following command: **ping -c 5 www.google.com**

The response to the ping command should be something like this:

```
PING google.com(sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e)) 56 data bytes
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=1 ttl=54 time=14.7 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=2 ttl=54 time=14.6 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=3 ttl=54 time=14.1 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=4 ttl=54 time=14.0 ms
64 bytes from sfo07s26-in-x0e.1e100.net (2607:f8b0:4005:802::200e): icmp_seq=5 ttl=54 time=14.2 ms
```

If the response is not like the above, you either are not connected to the internet, or you have a typo in the <http://www.google.com> name.

CONFIGURING THE METAR SCRIPT:

After installing the scripts, you will need to configure the airports your wish to show. The script retrieves information from the Aviation Weather Center that uses ICAO abbreviations for airports. The configuration file, ConfigMetarMap.py, contains a section that matches the LEDs to the correct airports. Since your map will mostly be laid out differently than the one used to develop the software, you will have to change the configuration to meet your requirements.

Each LED is individual addressable by it's position on the string. The first LED in the string is 1, second is 2, third is 3, . . . Once you have placed your LEDs on your map, you will then setup the software to tell it which LED will light for which airport.

The script will read the configuration file. Software languages are very literal and picky. The configuration file must follow an exact syntax or the software will not run.

Inside the file **ConfigMetarMap.py**, is an array called **CONSTANT_airportsArray**. The array starts with **=**[and ends with **]**. These symbols must not be removed or changed. Inside the symbols is a list of pairs that map the correlations between airports and LEDS. A pair consists of **["XXXX",#]**. The **XXXX** is the ICAO airport abbreviation and the **#** is the LED number. The **XXXX** must be enclosed in " " (quotation) marks. Each pair is must be followed by a coma (,) to indicate there is another pair to follow. The last pair doesn't have a coma (,) after it.

Remember that the Pi command lines interface is case sensitive.

1. Start the text editor nano. Make sure you are in the Metar Map Directory (**cd ~pi/MetarMap**).
 - Enter **nano ConfigMetarMap.py**
2. To configure the file, replace the airports with the 4 letter ICAO codes (all capital letters) for the airports and their corresponding LED numbers you want to include in your map.

****Note:** The airports you choose must be ones that broadcast Metar data. If not, there will be no data to send to the Pi and nothing will light up.

- Additional airports may be added. Make sure each pair is followed by a coma (,) except for the last pair. The airport and LED number don't have to be in any particular order.

****Note:** You may find it easier to layout your map with lights before setting the light numbers.

```
CONSTANT_airportsArray =[
    ["KMRY",1] ,
    ["KSNS",2] ,
    ["KWVI",3] ,
    ["KCVH",4] ,
    ["KE16",5] ,
    ["KRHV",6] ,
    ["KSJC",7] ,
    ["KPAO",8] ,
    ["KSQL",9] ,
    ["KSFO",10] ,
    ["KHAF",11] ,
    ["KOAK",12] ,
    ["KAPC",13] ,
    ["KO69",14] ,
    ["KLVK",15] ,
    ["KCCR",16]
]
```

3. Save the updated file.

- Press **Control+O** and then **Enter** to write the changes change to disk.
- Press **Control+X** to exit the editor.

****Note:** If your string has more than 50 LEDs, you must change the **LED_COUNT** constant, in **ConfigMeatarMap.py**, to match the number of LEDs used.

TESTING THE SOFTWARE:

There are a few tests you can run to make sure the airport data is being downloaded and the lights work correctly. Before running any of these commands, insure you are in the Metar Map directory (enter the command: **cd ~pi/MetarMap**)

1. **Airports:** To test the airports you have entered in the configuration you will run a test called **AirportMMCheck.py**. The script will use the configuration file (see above) to gather information from Aviation Weather Center

- Enter the Command **./runmetarscript.sh AirportMMCheck.py**

****Note:** It may take a second for the script to download the data, but you will see each airport you entered in the configuration show up with its current flight category as seen below.

KMRY

Airport: MONTEREY; ID: KMRY; Flight Category: VFR; LED #: 1

KSNS

Airport: SALINAS; ID: KSNS; Flight Category: VFR; LED #: 2

KWVI

Airport: WATSONVILLE; ID: KWVI; Flight Category: VFR; LED #: 3

****Note:** If you see an error like “airport not found” Make sure you typed in the code for that airport correctly with all capital letters. If it still has an error, the airport likely does not broadcast Metar data and you will need to remove it from the configuration.

2. **Lights:** To test that the lights are working, run a test called **strandtest.py**

- Enter the Command **./runmetarscript.sh strandtest.py**
- Make sure the lights are functioning properly.
- To stop the test press (control+C)

****Note:** Your lights must be powered and connected to the Pi for this test to work.

3. **Individual Light:** To check individual LED placement and colors, you can use **LEDCheckColor.py**

- Enter the Command **`./runmetarscript.sh LEDCheckColor.py`**
- The following questions will appear:

Enter which LED to light: (enter the LED number)

Enter the red value(0 to255): (enter a number between 0 and 255)

Enter the green value(0 to255): (enter a number between 0 and 255)

Enter the blue value(0 to255): (enter a number between 0 and 255)

A single LED will light up with the colors you have picked. White is 255 for all colors. The LEDs are RGB, so you can play with different colors.

****Note:** The software follows the industry standard for colors:

LIFR (Low Instrument Flight Rules) – Magenta;

IFR (Instrument Flight Rules) – Red;

MVFR (Marginal Visual Flight Rules) – Blue;

VFR (Visual Flight Rules) – Green.

If you wish to change the colors used for your map and are comfortable changing a configuration file, the array `FlightCatDef` in `ConfigMetarMap.py` defines the colors used.

RUNNING THE SCRIPT:

Once you have everything installed and have tested to make sure you Pi is working properly and sending the correct signals to the lights, you can run the Metar Script.

1.) To start the Script, Enter the Command **`./runmetarscript.sh MetarMapMain.py`**

- You should see the airports begin to populate on your screen and show the current flight category. Check to make sure the corresponding lights are showing the correct color as well.

****Note:** You may notice a situation where one of the lights may not light up. This is normal. If the software can't retrieve the current status for an airport, the LED is turned off. There are two reasons for not received data for a properly configured airport:

1. This is an issue with the Aviation Weather Center server not communicating with the airport (usually clears up in an hour or two)

2. The airport is not currently broadcasting the Metar info (airport is closed, etc...).

The script is set up to retry the download 1 more time if the data for an airport is not found. If you would like the software to make more retries, change `CycleRetries` in `ConfigMetarMap.py`

SETTING UP A CRON JOB:

The software is written to be started over and over again. Each time it is started, the software updates the LED/weather status. The alternate method would be to have the software constantly looping through the program over and over. We feel that our method takes less power and will recover in case the software hangs for some reason.

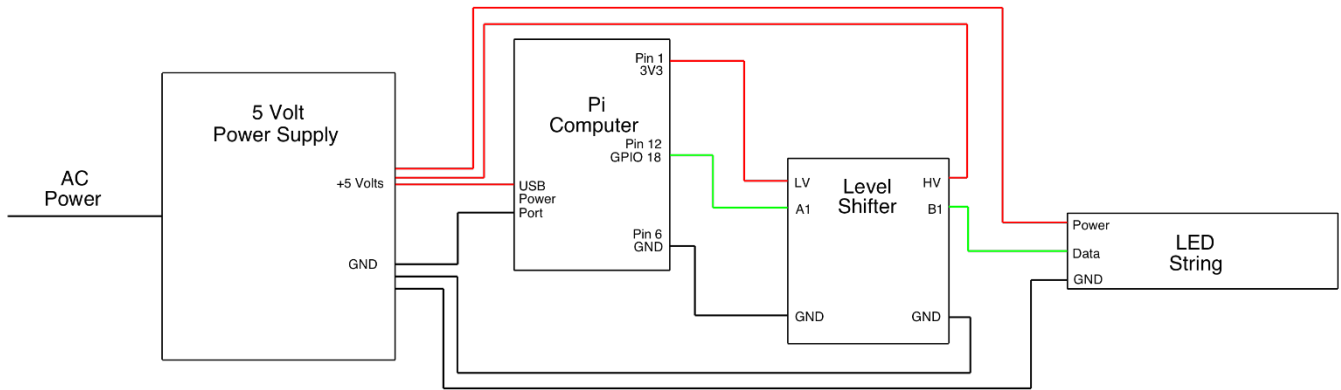
The mechanism used to periodically restart the software is a native Linux faculty called “cron”. We have written an installation script that will automatically install the “cron job” to restart the software every 10 minutes. If you would like to have a different update period and are comfortable with “cron jobs”, feel free to create your own or modify ours.

****Note:** When the script restarts you will see the lights turn off and restart as the data is downloaded. This is normal.

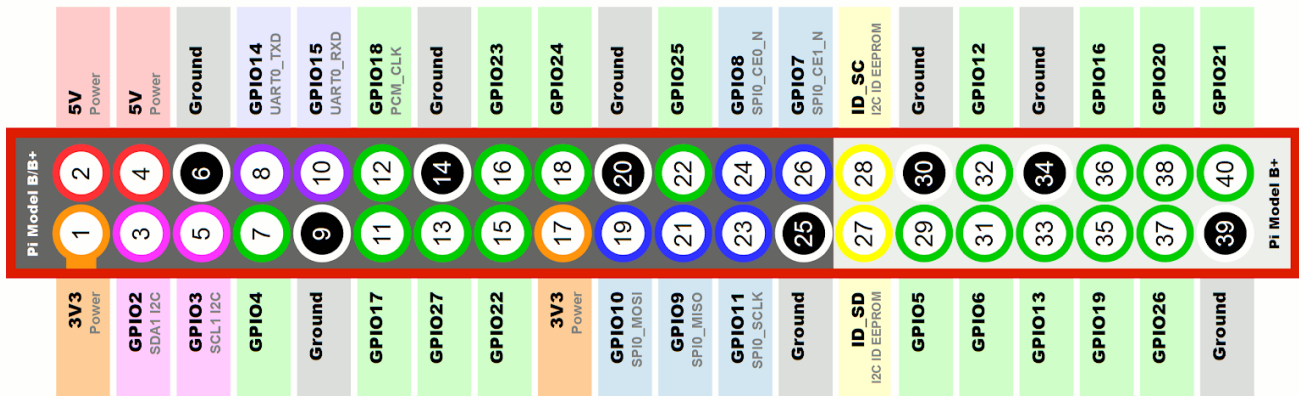
1. Enter the command `./installscript.sh cron` (it should say cron job created)

Wiring Installation Instructions

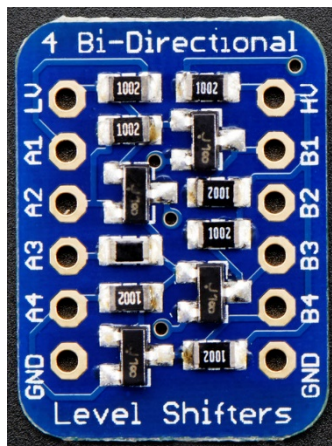
Metar Map Schematic



Pi Pin Header



Level Shifter



INSTRUCTIONS:

1. Attach an AC power cord to the power supply.
2. Connect a Micro USB cable to the + 5 volts and GND on the power supply.
3. Plug the Micro USB cable into the Pi's USB power port.
4. Connect Level Shifter HV junction to the + 5 volts on the power supply.
5. Connect the LED power lead to the + 5 volts on the power supply.
6. Connect the level shifter GND (right side) to the GND on the power supply.
7. Connect the LED GND lead to the GND on the power supply.
8. Connect Pi pin 3 (3v3) to the Level Shifter LV junction.
9. Connect Pi pin 6 (GND) to the Level Shifter GND (left side).
10. Connect Pi pin 12 (GPIO 18) to the Level Shifter A1 junction.
11. Connect Level Shifter B1 junction to the LED data line.

Note: The GND referenced above is the ground for the 5 volts, not the “earth connection” for the AC power.

LED INFORMATION:



1. The LEDs used during the software development look like the one pictured above.
2. **The LED string is uni-directional. There is an input side and an output side. The Level Shifter B1 must be connected to the input side. The input side is usually the female connector.**
3. Most LED strings come with connectors on each end. We just cut off the connector and soldered directly to the wire.
4. The input side connector is usually marked with an arrow pointing in or the word input. The input connector is usually the male connector.
5. Most of the LED strings on the market have only about 3” between LEDs. That was not enough for our map. To maintain the string integrity, we didn’t cut the string apart. We would skip a LED and leave it glued to the back of the map. An example: the first airport would use LED #1; the leads were not long enough to use LED #2 for the second airport; LED #3 was used for the second airport. The software configuration (ConfigMetarMap.py) would include airports for LED #1 and #3. Since LED #2 was not in the software configuration, it didn’t light up or use power. LED #2 just served as an extension.
6. For our LEDs: red lead – 5 volt power; white lead – data; and blue lead – GND.

Pi INFORMATION:

1. The Pi header has a 5 volt pin that can be used to power the Pi. We found this input to be very finicky with input voltage levels. We don't recommend using Pins 2 or 4 to power the Pi. We recommend the Micro USB port be used to power the Pi.
2. The Pi Zero W doesn't come with header pins. It has just holes where the leads can be soldered to. We found this easier to implement than the Pi computers that come with header pins.
3. For the Pis with headers, we used premade jumpers that plugs nicely into the header. Search the web for "GPIO jumper wires". Make sure you get the ones with male connectors on at least one end. We cut off the other end of the jumper and soldered it directly to the Level Shifter.
4. We added a little glue to help secure the USB connector and the GPIO connectors. We are aircraft mechanics, we use the belt and suspenders approach.

CONSTRUCTING THE MAP:

Since everyone's map will be a little different, we leave the exact construction up to you. Here are a few suggestions.

1. Use a standard picture and add a wood border to provide room to attach the electronics to the back.
2. Add plywood to the back of the picture frame to provide the enough strength for the electronics. To get a flatter gluing surface, use the laminate wood like that used in Ikea type furniture
3. Iron the map flat and then glue it to the plywood back. After the glue is dry, drill the holes for the LEDs and glue the LEDs through the plywood back.
4. Epoxy the electronics, LEDs and wires to the plywood. Hot glue will melt over time.
5. Use a long AC power cord. This eliminates the need to add an extension cord to use the map.
6. Add ventilation holes on the top and bottom of the border to provide cooling ventilation for the computer.

Configuration File Suggestion:

All of the setting can be found in ConfigMetarMap.py

Display is too bright:

Change LED_BRIGHTNESS to a lower value. Try 75 or 100 as a starting value.

```
LED_BRIGHTNESS = 255      # Set to 0 for darkest and 255 for brightest
```

Display Flight Category Examples:

The software has an option that will illuminate 4 LEDs in the flight category colors. By turning these lights on and labeling them, the map will have a color index that defines the flight categories.

Change:

```
DisplayFlightCategory = False
```

to

```
DisplayFlightCategory = True
```

Change the following entries to correspond to the LEDs used in the color index:

```
LIFR_LED_Num = 17      # Low Instrument Flight Rules LED Number
```

```
IFR_LED_Num = 18       # Instrument Flight Rules LED Number
```

```
MVFR_LED_Num = 19      # Marginal Visual Flight Rules LED Number
```

```
VFR_LED_Num = 20       # Visual Flight Rules LED Number
```