

Sense Your City: Seeeduino Setup Instructions

Thanks for being a part of the Sense Your City Project! Let's get started building, fabricating, and programming our sensors. This is going to be so. much. fun.

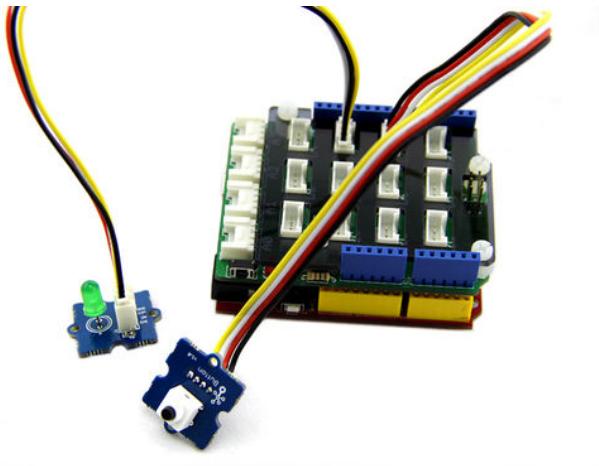
First, open the package, lay out all the items on the table in front of you.

You should have the following:

- 1 [Seeeduino](#)
- 1 [Grove Base-Shield](#)
- 6 Grove Sensors (and cables):
 - [Digital Light Sensor](#)
 - [Sound Sensor](#)
 - [Temperature & Humidity](#)
 - [Air Quality Sensor](#)
 - [UV Sensor](#)
 - [Dust Sensor](#)
- 1 Plastic Housing
- 1 Micro USB Cable

Plug the Grove Base Shield into the Seeeduino.

The black colored base shield fits perfectly on top of the red Seeeduino. Make a delicious circuit sandwich.



Find the switch on the Grove Base Shield and make sure it is switched to "5V" side.

This means that we will be powering all sensors with 5 Volts of electricity.

Plug a cable into each of the Grove sensors.

The cable only fits one way so don't force anything. The connectors on the cables fit the sockets on the sensors, so they should fit perfectly together. One thing to bear in mind is the colors of the wires correspond to the white writing on the sensor boards. The red wire is always connected to VCC - which supplies power. The black wire is always connected to GND (Ground).

Once all the sensors are connected to cables, plug the other ends into the Grove Shield. Each sensor has a specific, labelled port that we will plug it into.

Sensor	Port
Digital Light Sensor	I2C
Sound Sensor	A0
Temperature & Humidity Sensor	D4
Air Quality Sensor	A1
UV Sensor	A2
Dust Sensor	D7

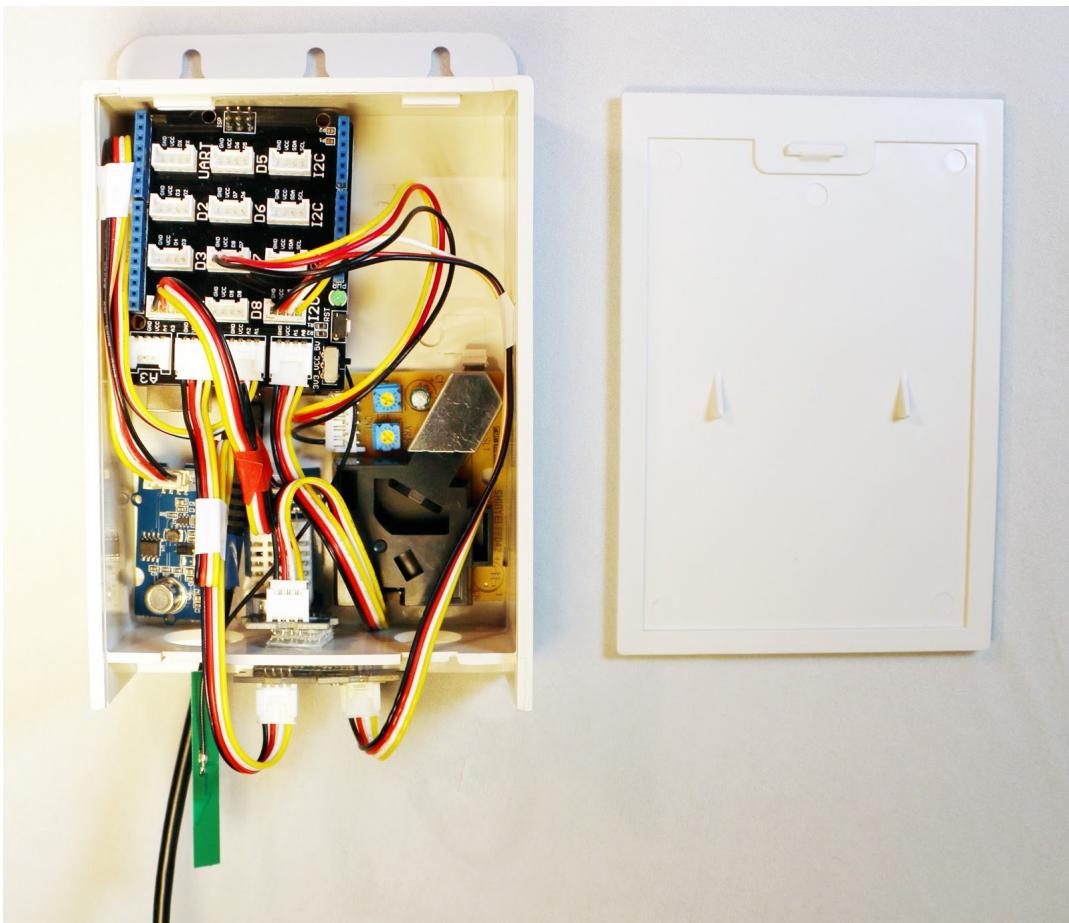
Plug the micro USB cable into the Seeeduino.

The small end of the USB cable goes into the red Seeeduino. Later on, we will plug the other end into our computer to program our node. Once our sensor is finished, we will plug the USB cable into an adapter and power outlet.

Install the sensor in the case

Once all of the electronic parts are connected, we are ready to install everything into the housing. For this part you might want to have some extra tools on hand. You don't *need* any extra tools but I'm going to use some tape and velcro. The top of the housing has a tab with three small holes. The very bottom of the housing has two large holes. This case will protect our electronics from water and weather, but will also allow our sensors to measure the environment outside the case, through the two large holes on the bottom.

Note: To secure the sensors in place and affix them to the case, I used a bit of [dual lock adhesive](#) to keep all the parts in place. You can also use velcro, or, if in a bind, electrical tape.



Put the Seeeduino inside of the box.

The letters "A0...A1...A2....A3" on the black Grove Shield should be facing downwards towards the bottom of the case.

Feed the micro USB cable through the left hole in the bottom of the case and plug it into the Seeeduino.

Using adhesive (tape, velcro, dual-lock, etc), secure the Air Quality Sensor all the way on the left so it hovers just above the hole.

Space is pretty limited in the case, so make sure the sensor is all the way to the left, the printed circuit board will be touching the plastic case.

Secure the Temperature and Humidity Sensor next to the right of the Air Quality Sensor, in the middle of the case.

Secure the Dust Sensor on the right side of the case so it hovers over the right hole.

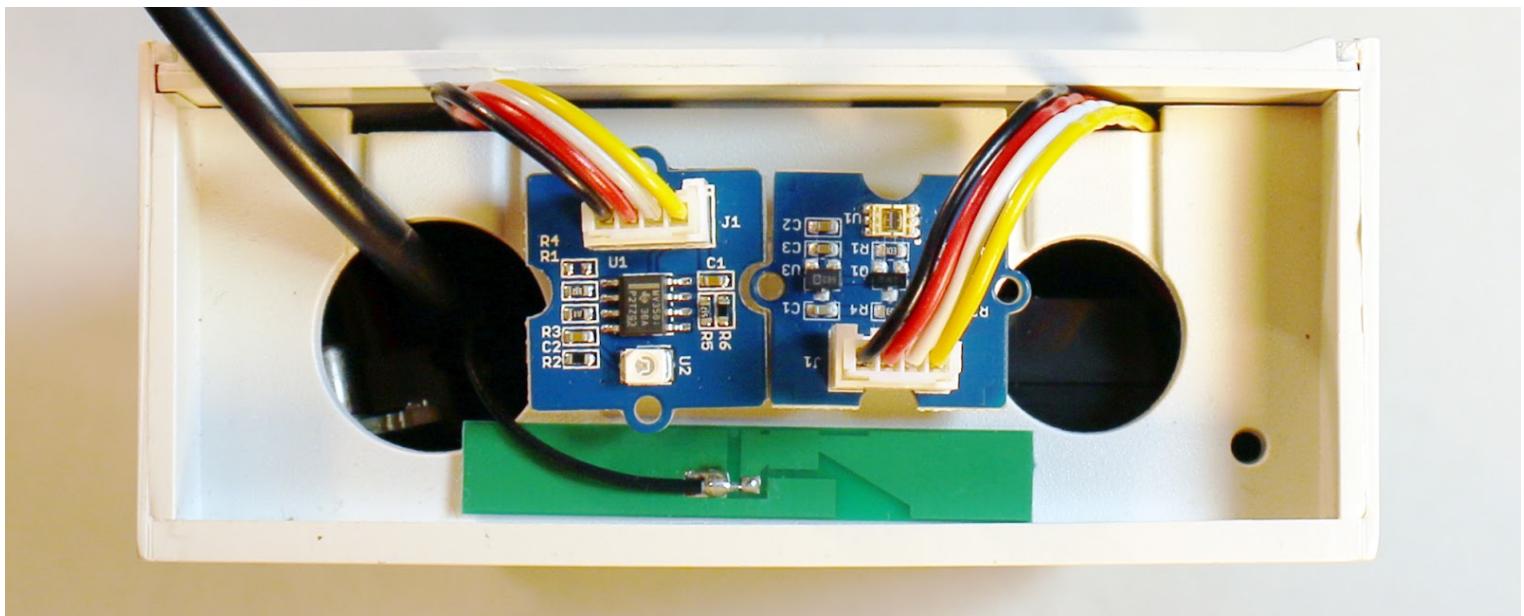
Make sure the metallic part of the Dust Sensor is pointing towards the top of the case.

Secure the Sound Sensor right above the Temperature and Humidity Sensor in the middle of the case.

It should be facing upwards towards the top of the case (where the tab with three holes is located). Make sure you leave enough clearance so you can slide the back of the case on. You can see where it slides on, there is a small ridge. The white connector at the end of the jumper cable should be below the ridge where the case slides on.

Feed the cables for the UV and Light sensors through the notches on the bottom of the case, and attach both sensors to the case so they are facing down.

The sensors are shaped so can put them together like a puzzle, you can tape them together with a single piece of adhesive and attach them to the case.



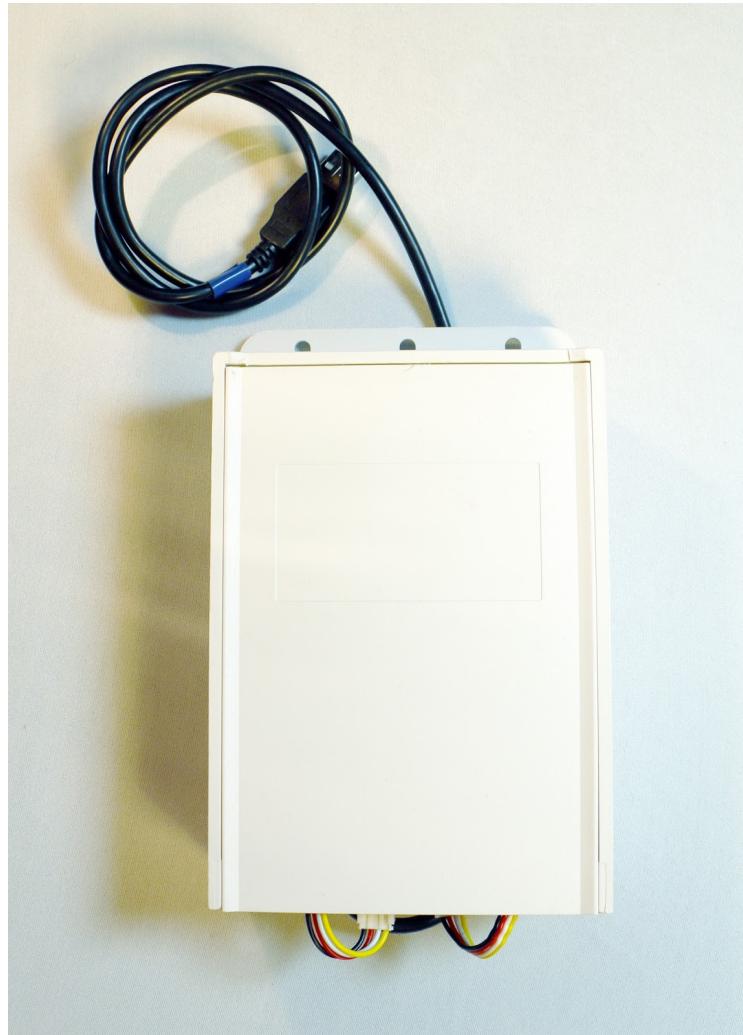
Run the antenna outside of the left hole on the bottom of the box, peel off the adhesive backing, and secure it to the case.

The antenna is the green circuit board hanging freely from the Seeeduino. Once you attach it, your box will look like a smiling robot, oh how cute is that.

Once all of the sensors are in place. Slide on the top of the case on.

If the case does not slide on easily, you will need to tidy up your wires and make sure that the UV and Light sensors' jumper cables fit in the notches. If the wires are a mess you can fold them and tape them together, or to parts of the case, so that it looks neater and fits better. If you use several pieces of colored tape to do this, it will be easier to identify which wire is which incase you need to troubleshoot later on.

Use a wire tie, string, or tape to attach the USB cable to the middle tab at the top of the case. The sensor will need to hang vertically from the USB cable. We don't want it to flip upside down somehow and expose our vulnerable sensors to rain or harsh elements - make sure it hangs securely. The USB cable should be secured between two of the holes in the top tab. You may also want to tape the USB cable to the back of the housing as well.



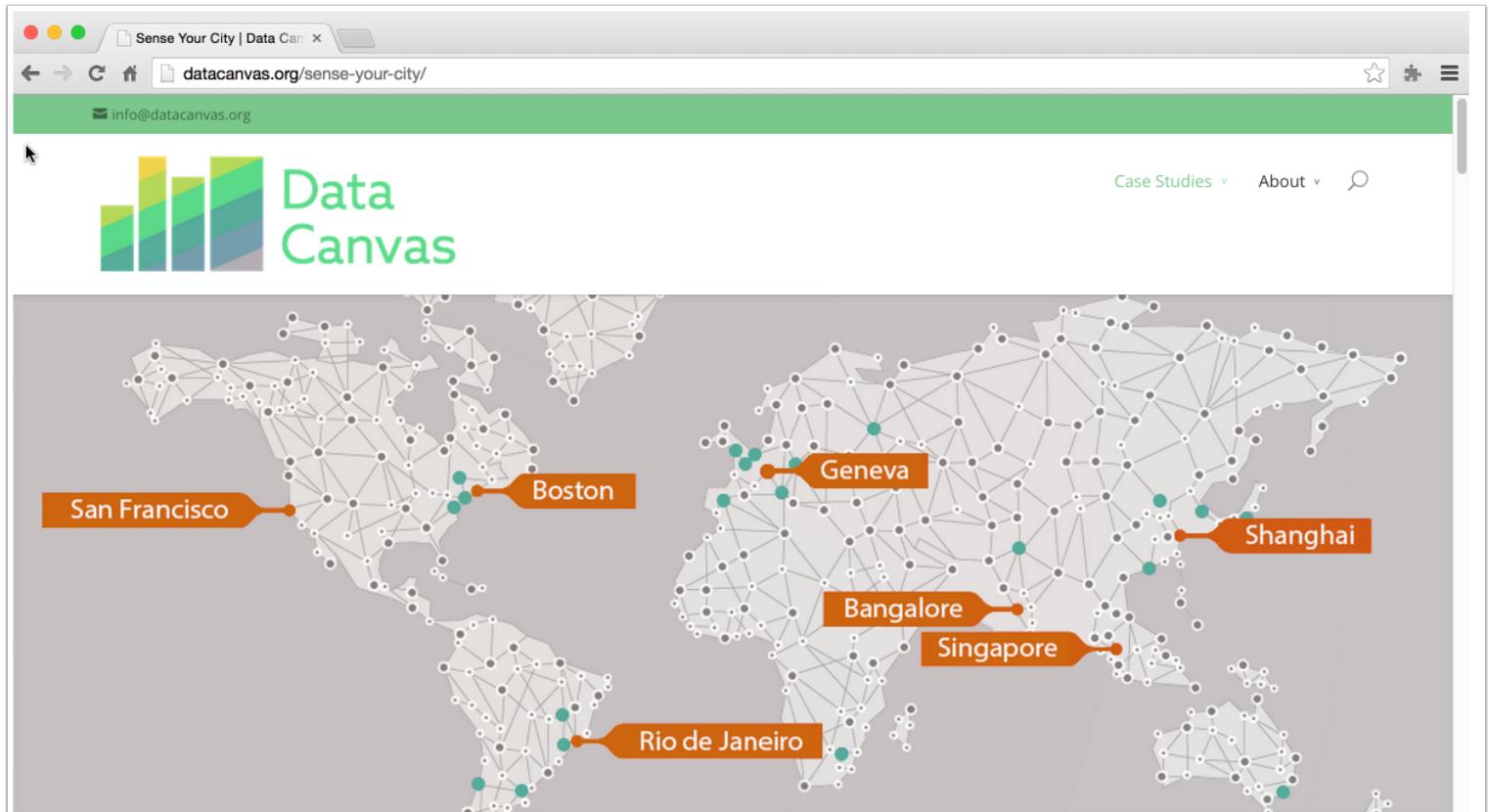
Add some pizzazz to your sensor by decorating the cover of your sensor node.

The cover is your own blank canvas and you are the next Picasso, draw some abstract weather patterns or merely scribble your twitter handle. Just make sure you add some pizzazz, everyone loves pizzazz, it's what made Picasso famous.

Program the Seeeduino.

Before getting started, download the code resources [here](#).

You will need to acquire a private key from Localdata, which will be assigned to you individually. Each sensor has its own unique Public ID and Private Key so that we can identify all unique data streams, we will also incorporate GPS coordinates, to plot all the sensors on a map.



Download and install Arduino 1.5.8 software from <http://arduino.cc/en/Main/Software>.

Windows has an easy installer. For the Mac OS X version, drag the downloaded Arduino icon to your applications folder. I had most success with the Mac OS X Zip file for Java 6, Arduino prompted me to install a version of Java 6 - I did so and had no problems after doing so.

The screenshot shows the Arduino Software download page. At the top, there's a navigation bar with links for Buy, Download, Products, Learning, Forum, Support, Blog, LOG IN, and SIGN UP. Below the navigation bar, there are links for Mac OS X, Linux: 32 bit, 64 bit, source, Examples, Foundations, and FAQ. The main content area features a teal header "Arduino 1.5.8 BETA (with support for Arduino Yún and Arduino Due boards)". Below this, a note says: "If you have the Arduino Yún or Due you must download the 1.5.8 version. Refer to the [Yún getting started page](#), or [Due getting started page](#) for specific details about those boards. Please discuss any issues in the [Yún forum](#) or [Due forum](#)". There are also links for "Download" and "Arduino 1.5.8 (release notes)" which lead to a list of download links for Windows, Mac OS X, and Linux.

Plug your sensor into a computer with the USB cable.

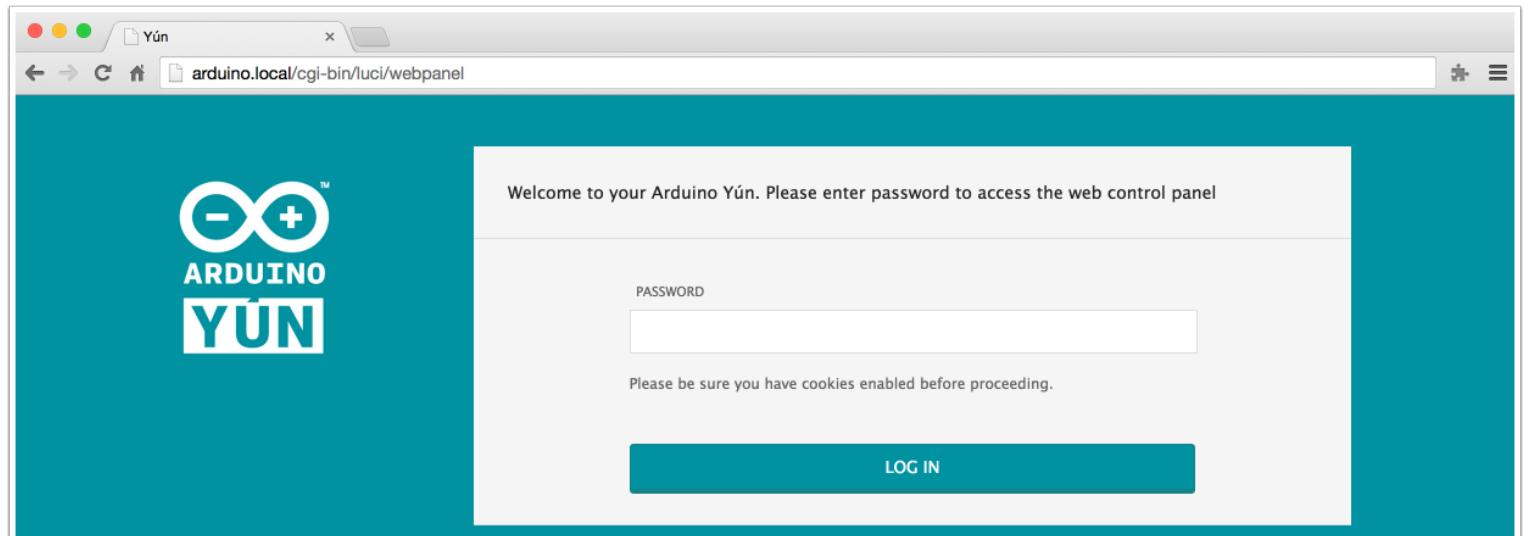
The sensor needs two things: power and a connection to the internet. Always make sure the sensor is within range of your Wifi router (maybe 50 ft or so). The first time you turn it on you will configure it to connect to your own Wifi network. If you change the location of your sensor, or your Wifi network settings (such as your password), you will need to reset and reconfigure your Seeeduino.

Connect to the Seeeduino.

After a few moments, check your wifi settings and look for an available network containing the word “Seeeduino” that’s your Seeeduino network, select it and connect to it. If you do not see any network named Seeeduino try one or all of the following: Unplug and replug your USB cable, wait a moment for the network to show up. Move closer to your wifi router. If after repeated tries you do not see any networks, you can plug an Ethernet cable into the Seeeduino and connect it to your wifi router, which will guarantee that your sensor is connected to your network.

Log in to the Seeeduino, when prompted for a password enter “seeeduino.”

In a web browser, go to the following address: 192.168.240.1. This is the default IP address of your Seeeduino, in the next step you will give it a name, which will allow you to connect to it by directing your web browser to `your_name_here.local`



....

Configure the Seeeduino to access your own wifi network.

Give your board a name, something catchy, anything that you will remember. Create a password that is at least 8 characters. Select your Wifi Network from the list of “Detected Wireless Networks,” enter your Wifi network’s password. If you ever move your Seeeduino to another wifi network you will need to repeat this process, and change the Wireless Parameters accordingly.



For more advanced network configuration features, see the [advanced configuration panel \(luci\)](#)

YÚN BOARD CONFIGURATION ?

YÚN NAME * Aethera

PASSWORD *****

CONFIRM PASSWORD *****

TIMEZONE * America/Los Angeles ▼

WIRELESS PARAMETERS ?

CONFIGURE A WIRELESS NETWORK

DETECTED WIRELESS NETWORKS EAB1023 (WPA2) ▼ [Refresh](#)

WIRELESS NAME * EAB1023

SECURITY WPA2 ▼

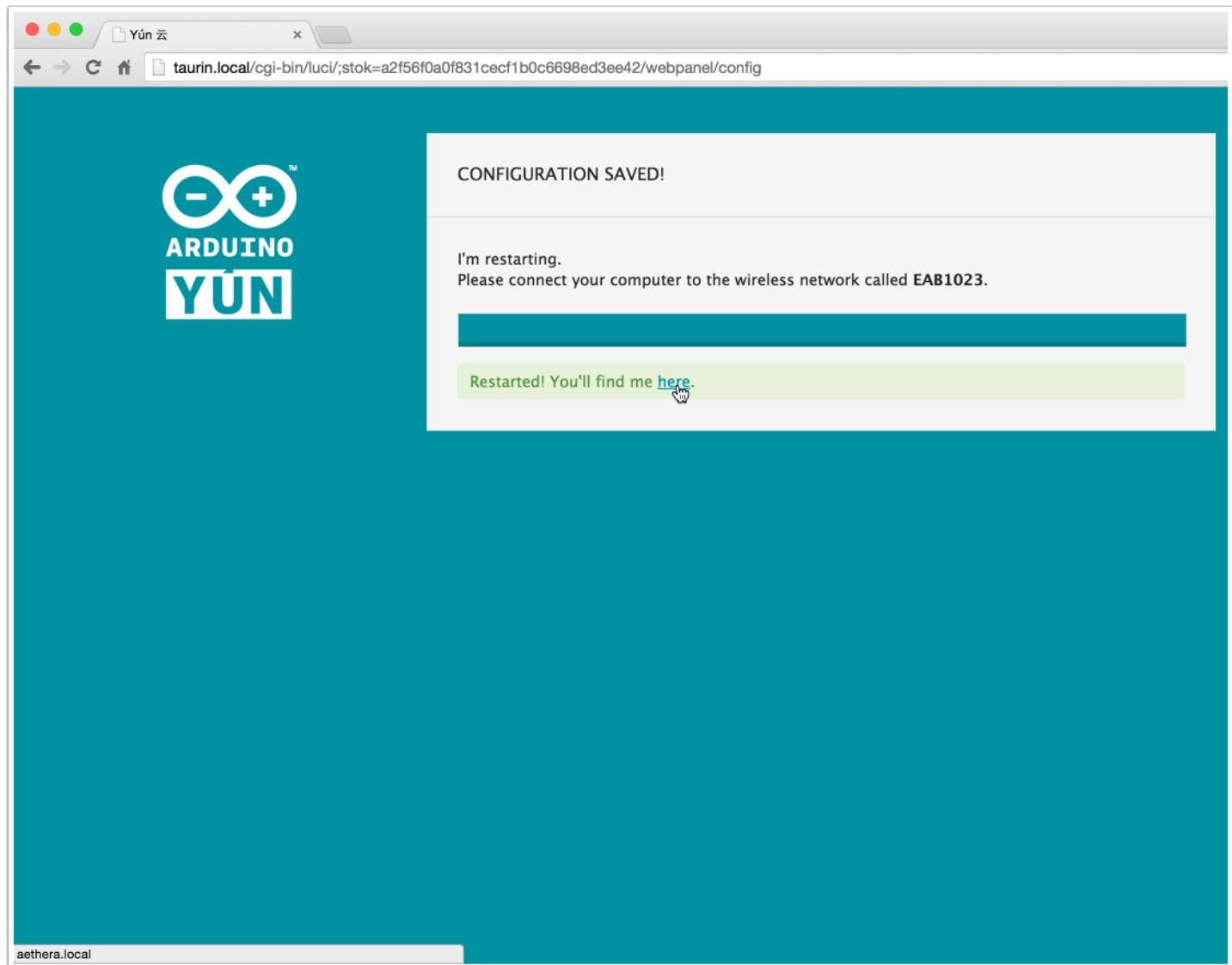
PASSWORD * *****

[DISCARD](#)

[CONFIGURE & RESTART](#)

Click the “Configure and Restart” button

The Seeeduino will restart and log on to your wireless network. If you had to plug your sensor directly into a router with an Ethernet cable earlier, you should now be able to remove it.



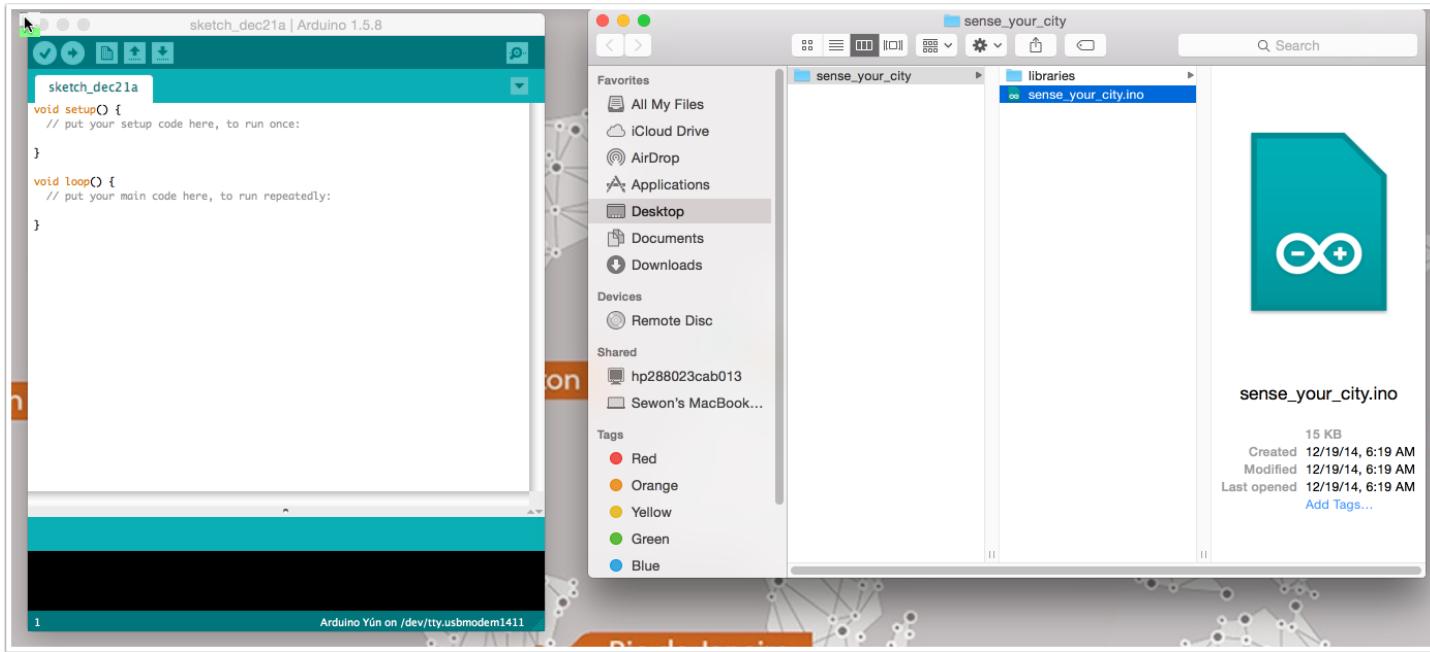
Disconnect your computer from the Seeeduino wireless network and connect your computer to your usual wifi network.

Once connected to your usual wifi network, direct your web browser to yourseeduinosname.local. You should see the same Arduino interface as before, but this time on your own wifi network. This means your Seeeduino has properly connected to your wifi network. w00t!

....

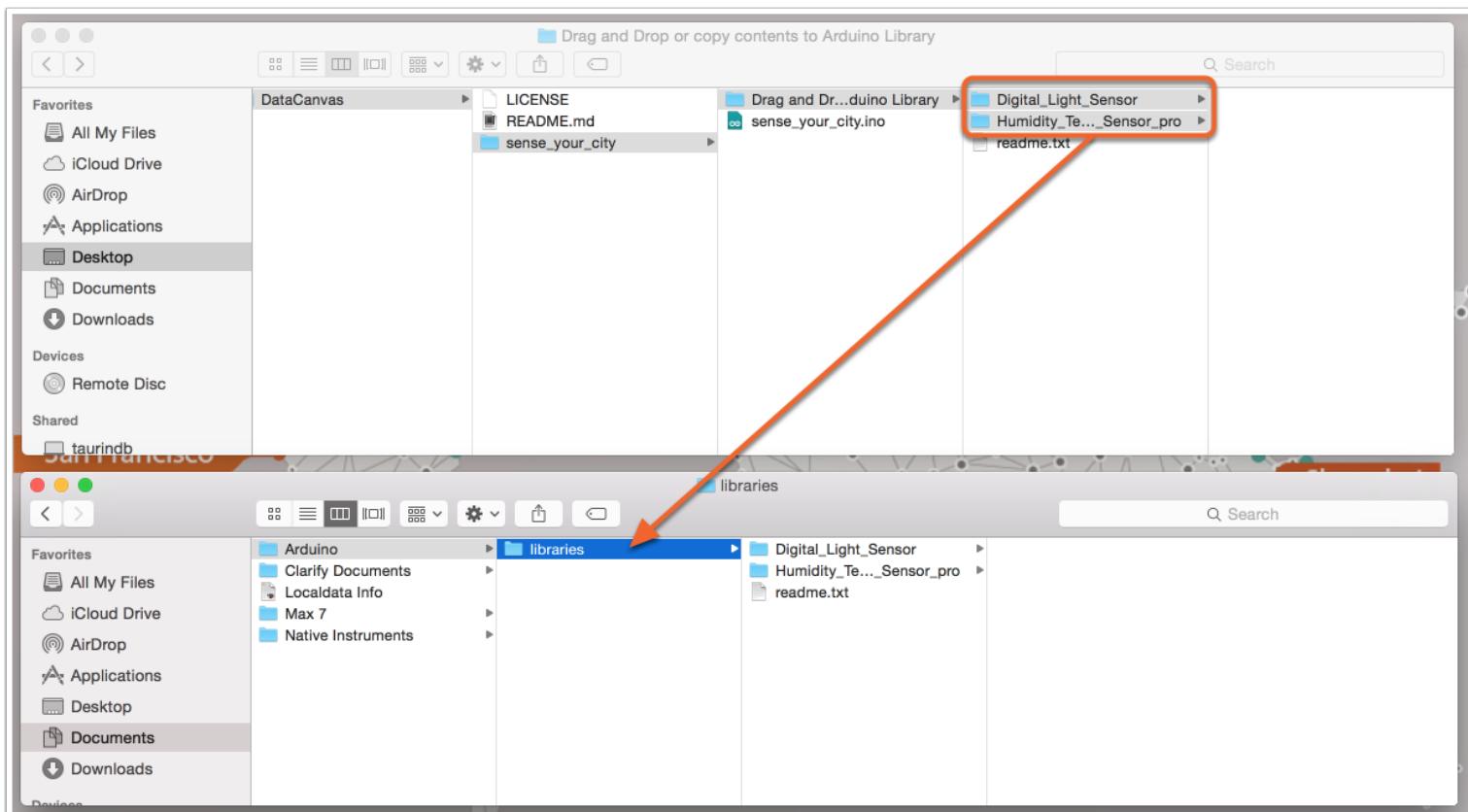
Open the `sense_your_city.ino` file.

It should open in the Arduino IDE (Integrated Development Environment).



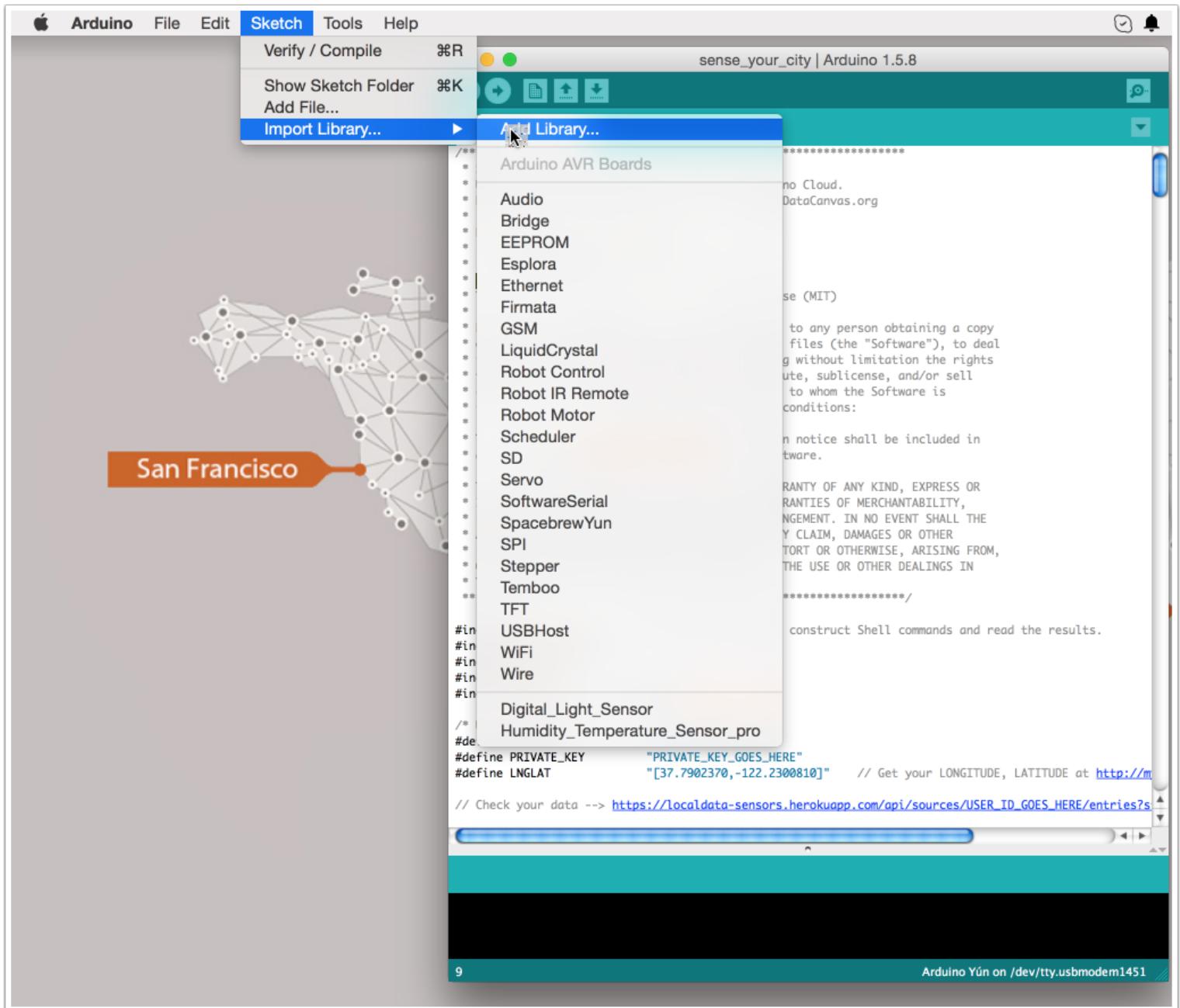
Add the libraries in your `sense_your_city` folder to your local Arduino Library.

You can do this two ways. The easiest way is to copy the two folders located in `sense_your_city/libraries` to your Arduino library folder. On OS X the local Arduino library is located at `/Users/yourusername/Documents/Arduino/libraries`.

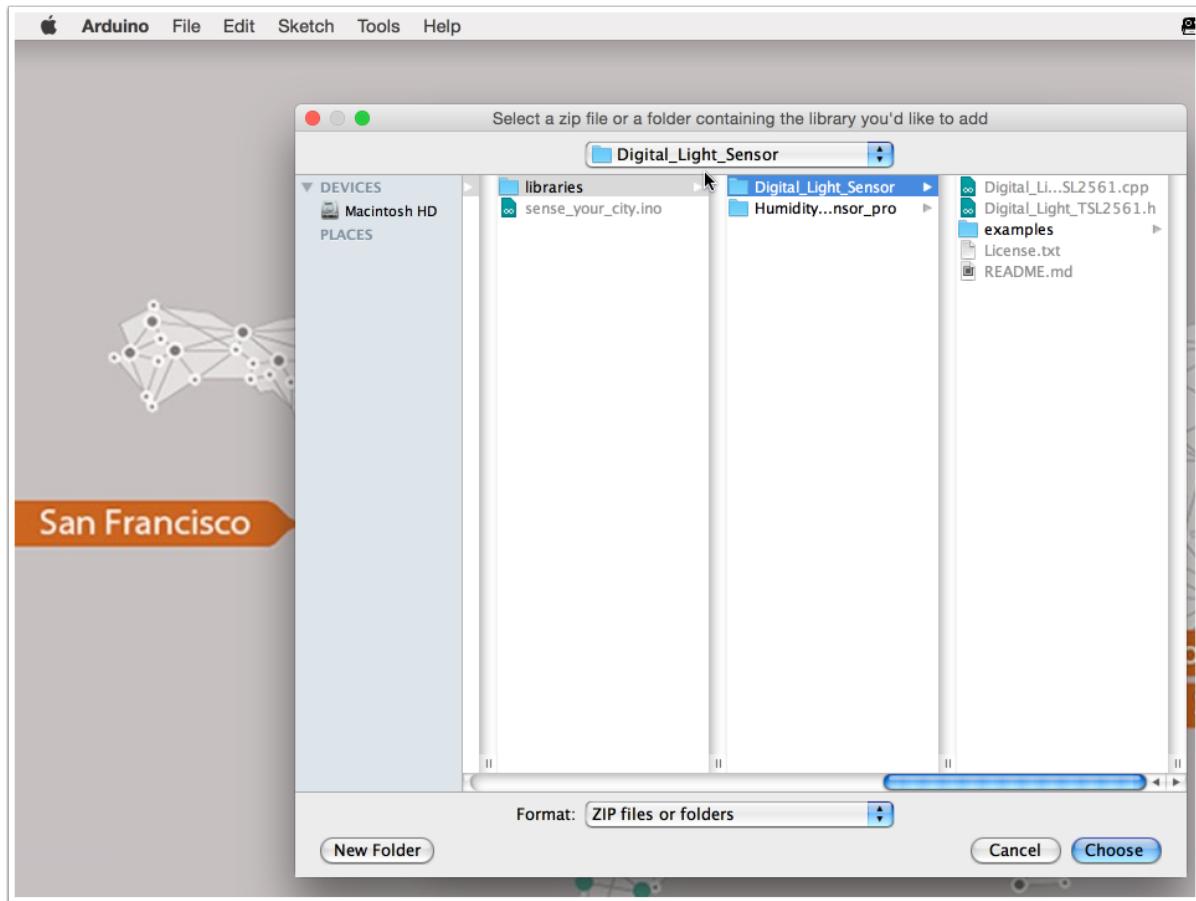


The standard way to add libraries is by using the “Add Library...” function found in the “Sketch” menu in the Arduino IDE.

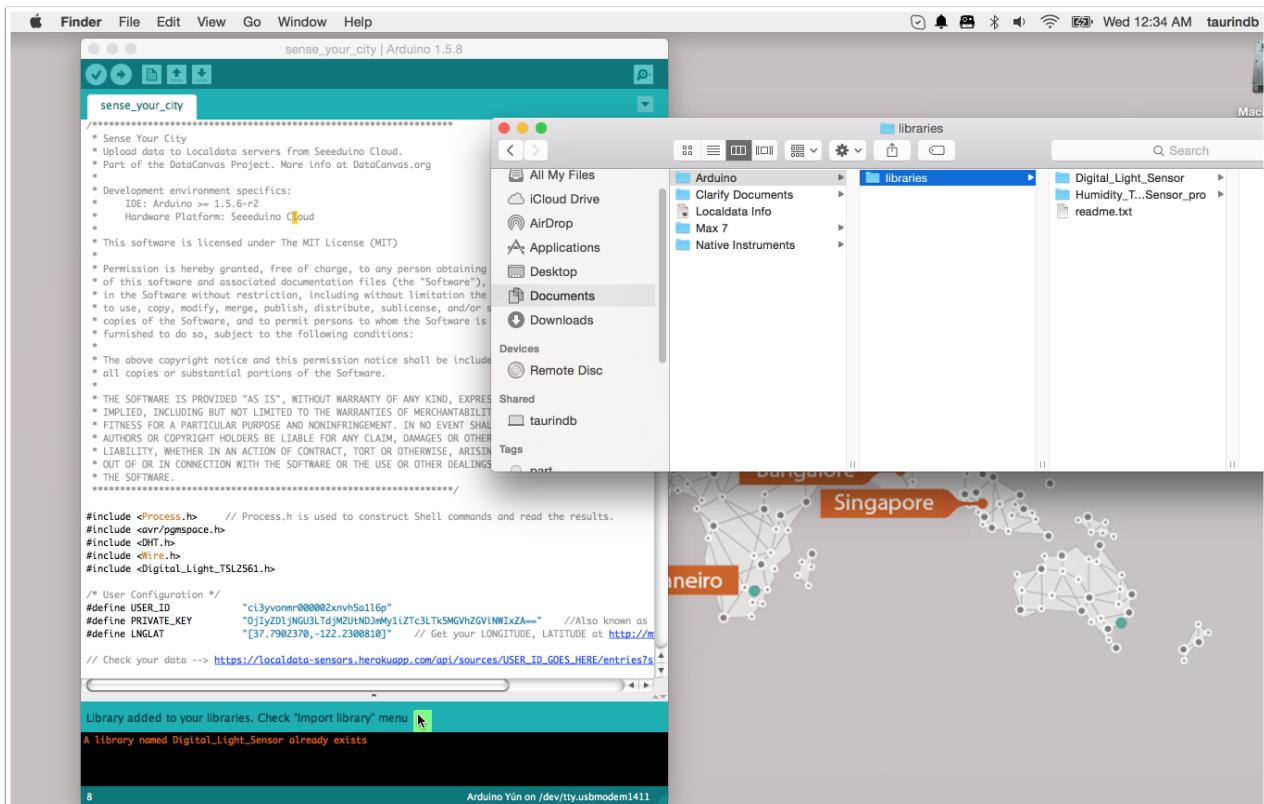
Go to Sketch > Import Library... > Add Library...



Add both of the folders included in the *sense_your_city/libraries* to your Arduino Library. Select each folder individually and click “Choose.”

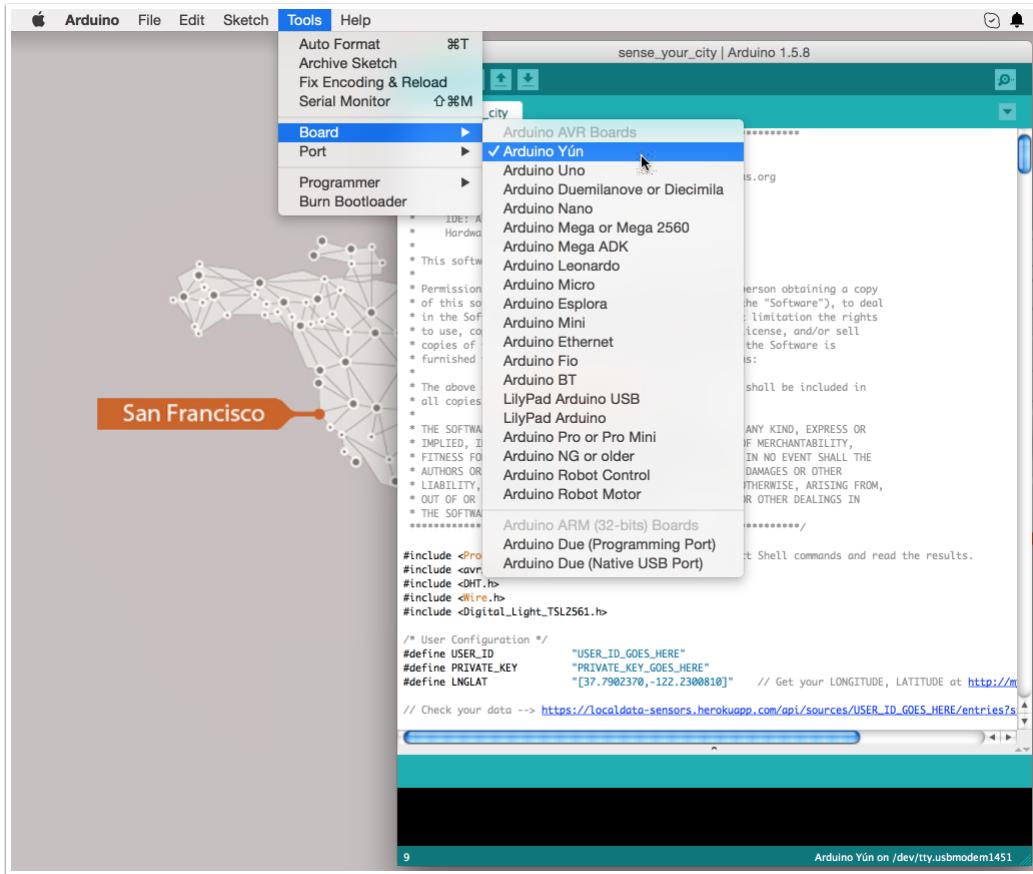


If the library was added, the Arduino IDE will display that it was successful.

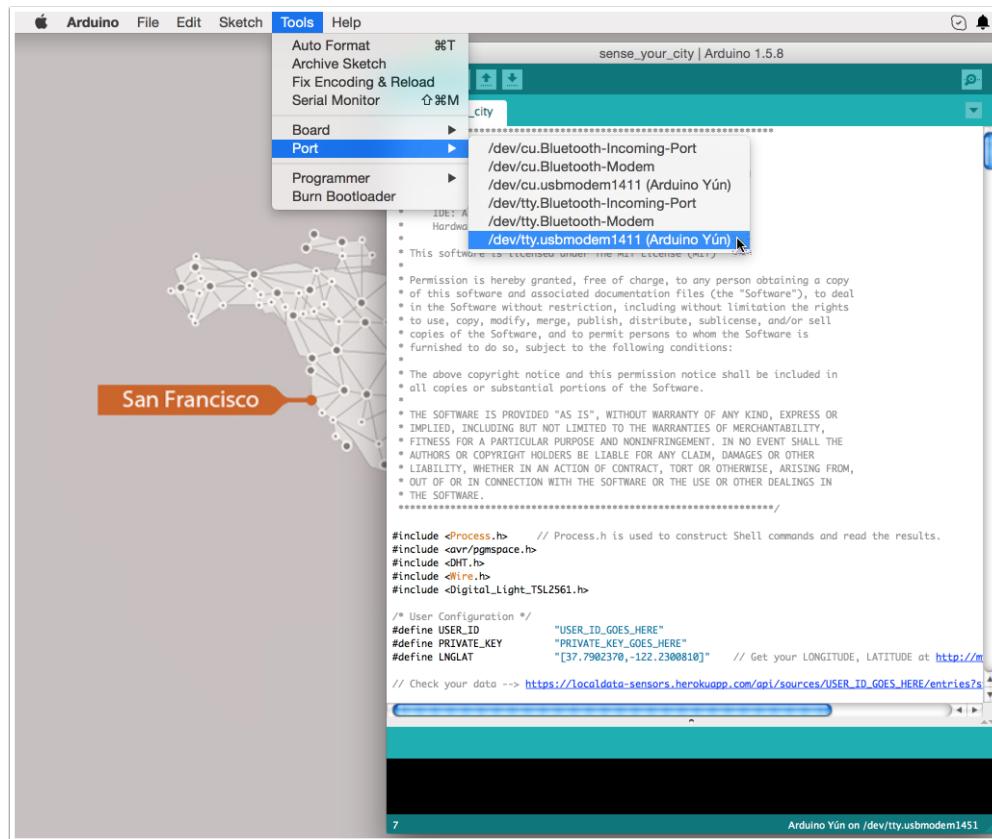


Confirm that your computer is configured for your board.

Go to Tools > Board > Arduino Yún



Next go to Tools > Port > /dev/tty.usbmodem (Arduino Yún)



Make the necessary modifications to the `sense_your_city.ino` file.

First, copy your user ID and private key into the space provided.

Local Data Key — Edited

id	token	encoded auth value
ci3yvonnmr0200ds02xvh5a116p	22d9c4e7-7c3e-42f3-be77-990eadeb5b1d	OjlyZDijNGU3LTdjM2UtNDJmMy1iZTc3LTk5MGVhZGVINWlxZA==

To confirm that data is arriving:

<https://localdata-sensors.herokuapp.com/api/v1/sources/ci3yvonnmr000002xvh5a116p/entries?startIndex=0&count=10>

```

sense_your_city §

THE SOFTWARE.

#include <process.h> // Process.h is used to construct Shell commands and read the results.
#include <avr/pgmspace.h>
#include <DHT.h>
#include <Wire.h>
#include <Digital_Light_TSL2561.h>

/* User Configuration */
#define USER_ID "ci3yvonnmr0200ds02xvh5a116p"
#define PRIVATE_KEY "OjlyZDijNGU3LTdjM2UtNDJmMy1iZTc3LTk5MGVhZGVINWlxZA==" //Also known as "LONGITUDE,LATITUDE" Get your LONGITUDE, LATITUDE at http://mygeop...
#define LNGLAT

// Check your data --> https://localdata-sensors.herokuapp.com/api/sources/USER_ID_Goes_Here/entries?st...

/* Define sensor locations on Grove */
#define pin_uv A2 // UV sensor
#define pin_sound A0 // Sound sensor
#define pin_dust 7 // Dust sensor (D7)
#define pin_air_quality A1 // Air Quality sensor
#define pin_dht 4 // Humidity and Temperature sensor (D4)
#define DHTTYPE DHT22 // DHT 22 (AM2302)

#define FP(string_literal) (reinterpret_cast<const __FlashStringHelper *>(string_literal))

/* Define Data Sampling Functions */
Float iReadTemperature(void);
Float iReadHumidity(void);
Float iReadDensityDust(void);
unsigned long iReadLux(void);
Float iReadUVRawVol(void);
int iReadSoundRowVol(void);

/* Particulate Matter sensor */
unsigned long dust_starttime;
unsigned long duration_starttime;
unsigned long duration;
unsigned long sampletime_ms = 30000;
unsigned long lowpulseoccupancy = 0;
Float ratio = 0;
Float concentration = -1;
int sensorValue=0;

Done uploading.

at java.awt.Desktop.checkFileValidation(Desktop.java:191)
at java.awt.Desktop.open(Desktop.java:252)
... 40 more

```

Arduino Yun on /dev/tty.usbmodem1411

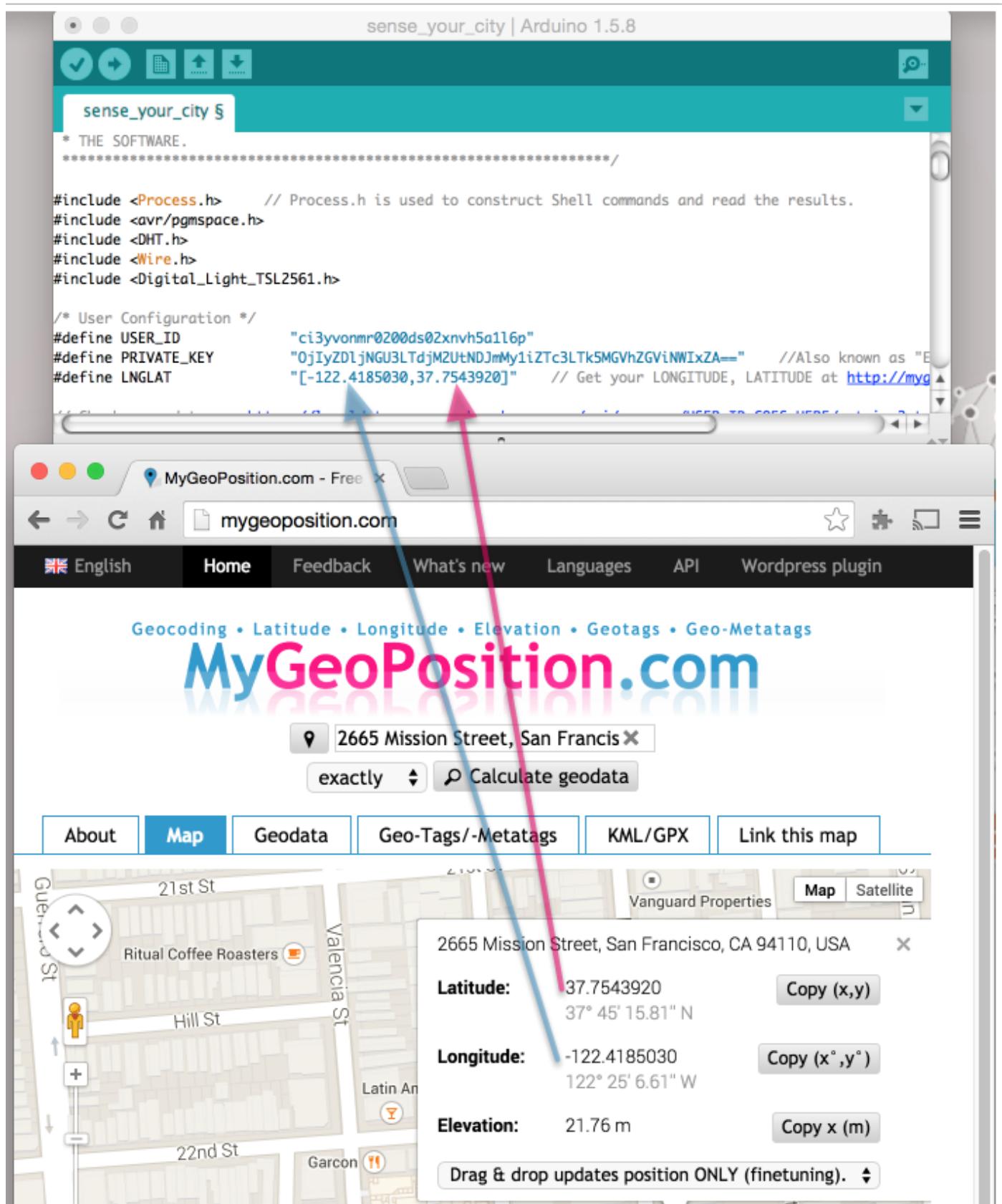
Add your GPS Coordinates

Go to mygeoposition.com and enter the address where the sensor will be installed. Click the “Calculate geodata” button. Click the “Copy (x,y)” button to copy your GPS coordinates and paste them into the Arduino IDE.

The screenshot shows a web browser window for 'MyGeoPosition.com - Free'. The URL 'mygeoposition.com' is in the address bar. The page title is 'Geocoding • Latitude • Longitude • Elevation • Geotags • Geo-Metatags' followed by the main logo 'MyGeoPosition.com'. Below the logo is a search bar with the address '2665 Mission Street, San Francisco, CA 94110' and a dropdown menu set to 'exactly'. A button labeled 'Calculate geodata' is to the right. Below the search bar is a navigation menu with tabs: About, Map, Geodata, Geo-Tags/-Metatags, KML/GPX, and Link this map. The main content area is a Google Maps interface showing the location at 2665 Mission Street. A callout box displays the coordinates: Latitude 37.7543920, Longitude -122.4185030, and Elevation 21.76 m. Buttons for 'Copy (x,y)', 'Copy (x,y)', and 'Copy x (m)' are provided. A note says 'Drag & drop updates position ONLY (finetuning)'. The map shows surrounding streets like Valencia St, 21st St, 22nd St, and 23rd St, along with various businesses and landmarks.

Reverse GPS Coordinates.

After pasting your coordinates, make sure to reverse them so longitude is first and latitude is second.



Verify and upload the code.

Click the check button to verify that the code modifications you made are OK. Next, click the play button to upload your code to the Seeeduino Cloud.

```
sense_your_city | Arduino 1.5.8

sense_your_city §

while(true)
{
    uint32_t t = millis();
    p.runShellCommandAsynchronously(F("ntpd -n -q -p 0.openwrt.pool.ntp.org"));
    // Block until clock sync is completed
    while(p.running() && millis() - t < 10000);
    if(p.running())
    {
        Serial.println(F("fail to sync time, retry..."));
        p.close();
        delay(1000);
        continue;
    }
    else break;
}
//light up the led
digitalWrite(13, HIGH);
}

// Return timestamp of Linino
unsigned long timeInEpoch()
{
    Process time;           // process to run on Linuino
    char epochCharArray[12] = ""; // char array to be used for atol

    // Get UNIX timestamp
    time.begin(F("date"));
    time.addParameter(F("+%s"));
    time.run();

    // When execution is completed, store in charArray
    while (time.available() > 0) {
        //millisAtEpoch = millis();
        time.readString().toCharArray(epochCharArray, 12);
    }

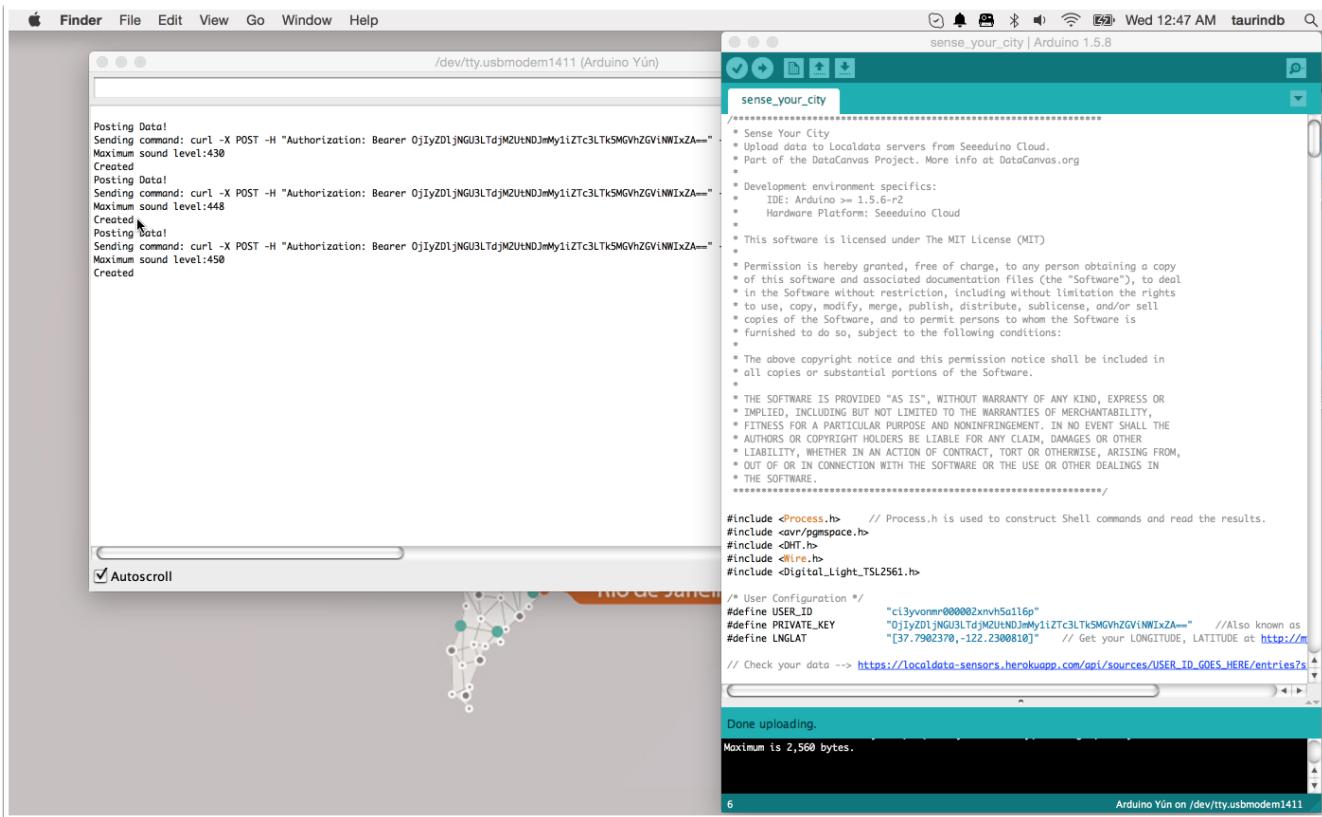
    // Return long with timestamp
    return atol(epochCharArray);
}

Uploading...
Sketch uses 26,226 bytes (91%) of program storage space. Maximum is 28,672 bytes.
Global variables use 985 bytes (38%) of dynamic memory, leaving 1,575 bytes for local variables.
Maximum is 2,560 bytes.

18 Arduino Yún on /dev/tty.usbmodem1411
```

Open the serial monitor, to make sure the connection is working.

Goto Tools > Serial Monitor. If everything is working, you will see new window that contains incoming communication from the Seeeduino, including verification that you are uploading online.

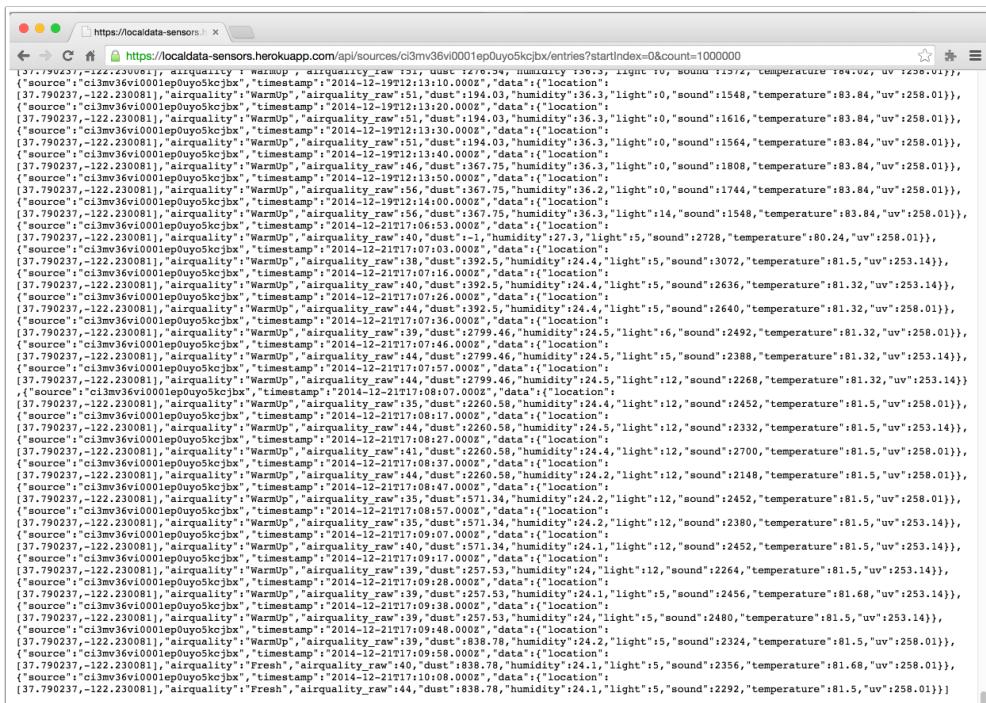


Check the server to verify that your code is uploading.

In a web browser, go to the following address:

https://localdata-sensors.herokuapp.com/api/sources/USER_ID_Goes_Here/entries?startIndex=0&count=1000

Be sure to change USER_ID_Goes_Here to the user ID you received. Press your browser's reload button several times over the course of several minutes to make sure data is uploading online. Data should be uploading about every 30 seconds or so. If everything is working, do a super funky celebration dance.



Node Installation

After we have confirmed that our sensor is successfully uploading data to the web, we are ready to install it outside. Place the sensor somewhere interesting - a busy street corner, near a public transit station, next to a public park, etc. Make sure the sensor is always safe from harsh weather conditions. If you predict some terrible weather is about to happen, it is a good idea to unplug or move your sensor somewhere safer.

Unplug the Seeeduino's USB cable from your computer and plug it into USB adapter that will plug into a power outlet.

You can use any standard USB port to supply your sensor with power, just make sure the Seeeduino is always powered and check on it from time to time. You can tell if the Seeeduino is powered by looking into the bottom holes of your housing. You should always see some LEDs blinking inside the case.

Hang your sensor outside by the USB cord.

At Gray Area, I used a utility clip to secure the USB cord to a ledge outside. Perhaps the best way to hang your sensor is to secure the USB cord to an object inside your dwelling and hang the sensor out your window. At SEEED, the sensor hung from an office window, the USB cable was secured and clipped to a desk inside. There are many ingenious ways to make sure that your node does not fall down to the ground and cause a tragic, tragic accident.



Once your sensor is installed, double check to make sure it is still uploading data online.

Check the [Localdatalink](#) to make sure data is still posting to our servers. If your sensor is not uploading data, it is probably too far away from your wifi router. Try moving the sensor closer to the router and turn the power off and on to reset it.

Celebratory (Funky) Victory Dance

If you have verified that your sensor is up and running, do a celebratory victory dance, make it extra funky. When you get home, go straight to your kitchen and make yourself the most delicious sandwich you've eaten in years. You did my friend, you really did it. Now it's time for us to make some use of this data - follow along with the project at DataCanvas.org to see how artists, developers, and enthusiasts are turning this environmental data into meaningful media. Be a part of the conversation at <https://www.facebook.com/groups/datacanvas/> and stay tuned!

Troubleshooting (In progress):

- If you encounter any problems connecting your sensor to your wifi network: hold the button on the top left hand corner (opposite the ethernet port) of the Seeeduino to reset it to its factory settings. Consult the “Program Your Sensor” section on how to reconfigure your wifi settings.