**ASEE2018 IoT workshop handout**

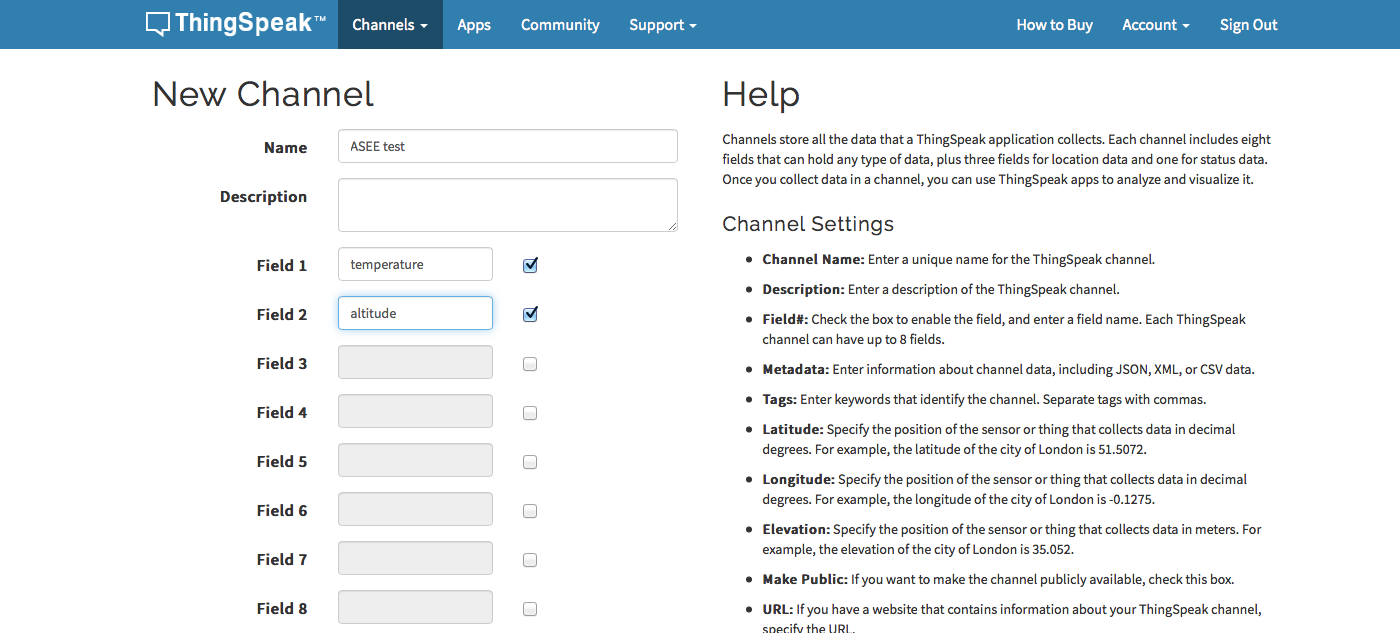
**Sign up on ThingSpeak**

-go to thingspeak.com

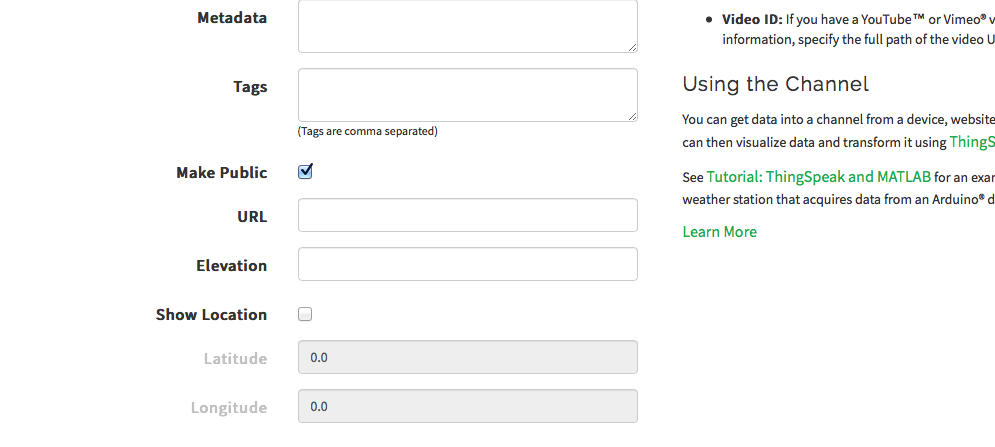
-sign up for an account

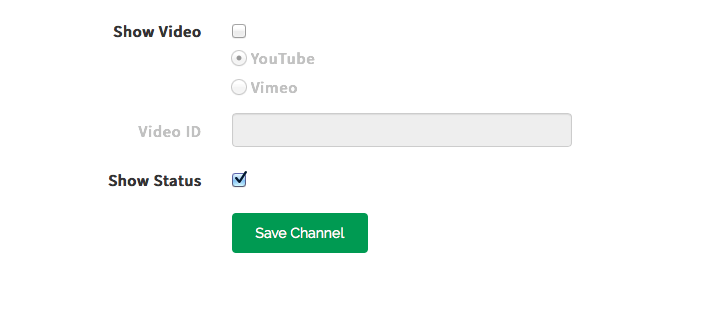
**We’ll create a channel**

Click on “my channels”, then click “new channel”. You’ll see this page:

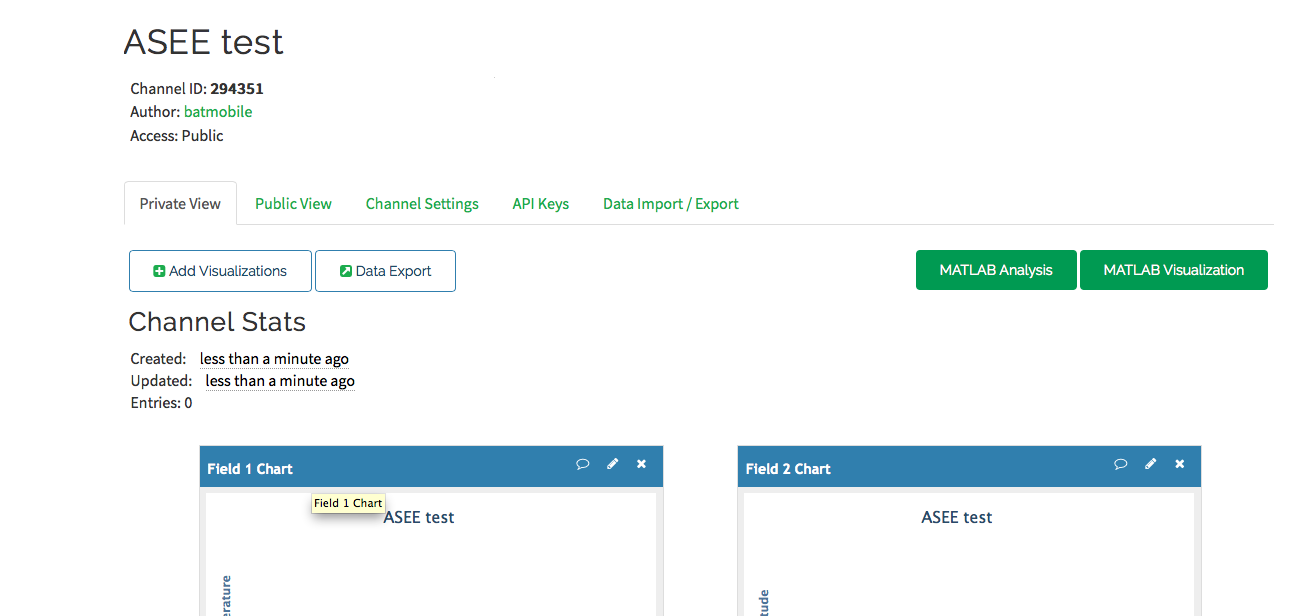


I called my channel “ASEE test” and I filled in two fields, labeled “temperature” and “altitude”

Next, I moved down the page and selected ”make public” :

And finally I clicked “save channel”:

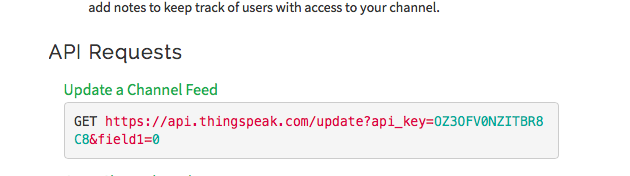
On the next page I’ll get an overview of the channel and some options for navigating the channel I created.



The most important things are the channel ID, which will allow you to find a reference your channel and the tabs under the author and access listings.

There are a few tabs to navigate, The Public view tab allows you to preview what is seen on the web, the channel settings tab controls the functionality of the channel and the API keys tab allows you to communicate with the channel over the web.

Let’s test you feed, go to the bottom of the page and look for “Update a Channel Feed”, copy the whole URL, but leave off the “GET”.



Paste it into a browser navigation bar and hit enter;

Screen Shot 2017-06-27 at 7.21.03 PM.png

You can change the value after the “=” and hit enter. Check your public feed for updates. Is it working?

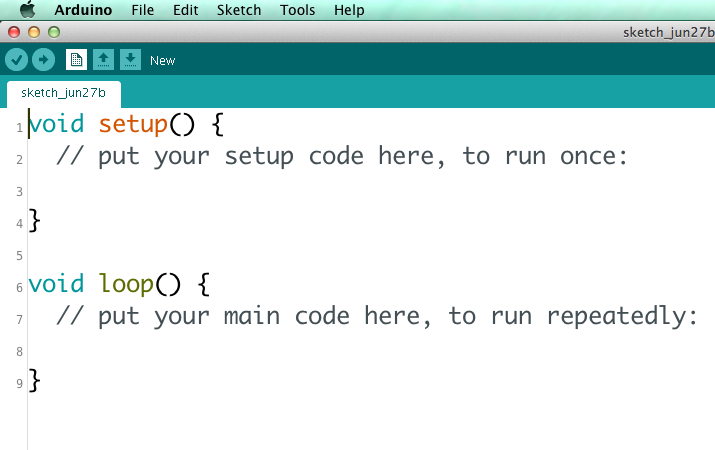
**Arduino: Code and Hardware**

Next, from the thumb drive, I’ll open the Arduino environment.

Select from the “AR” drive the folder for your operting system(Mac or Windows), in windows you’ll need to navigate one more folder in.

Click on the blue Infinity symbol with the positive and negative symbols in the center. 

If all went well you should see an editor window like this:



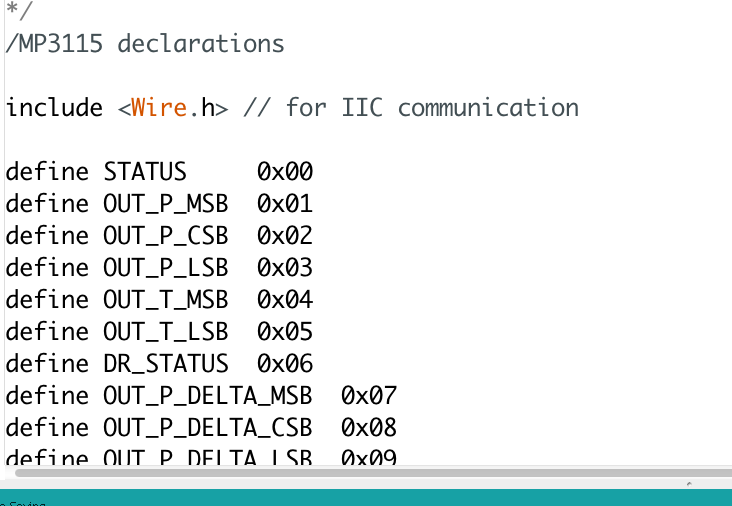
Next, we’ll open example code into the Arduino editor.

Under file in the Arduino environment, click open and open the “workshop Arduino code “ folder and open the “thingspeak\_example” code.

If you don’t have a pre-built drive with the Arduino code

From the Github link (<https://github.com/balboajeff/ASEEIoT>)or the shared folder(<http://bit.ly/ASEEcode>) open the “ThingSpeakTestASEE\_sendData” code. Copy all the code and paste it into the new sketch you opened in Arduino.

If all goes according to plan you should have this at the top of your sketch:



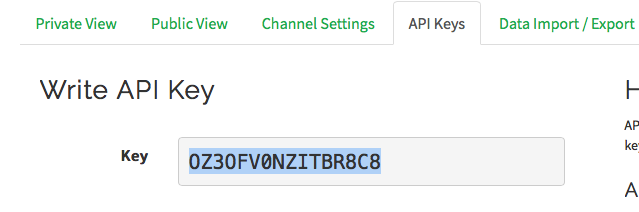
About 70 lines down in the code we want to check the SSID for the wireless and change the stream ID for ThingSpeak.

*We have purchased conference wifi and we’ll get you set up with the password and ID*

If there are problems with the wifi, we have a router and you can log on to that.

The wireless should be “sparkfun-workshop”and the password is “sparkfun”.

From your API Keys on your ThingSpeak dashboard grab the write key:



Replace the “streamId key in the quotation marks with your key:

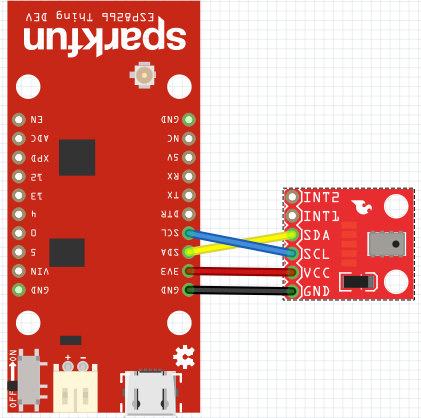


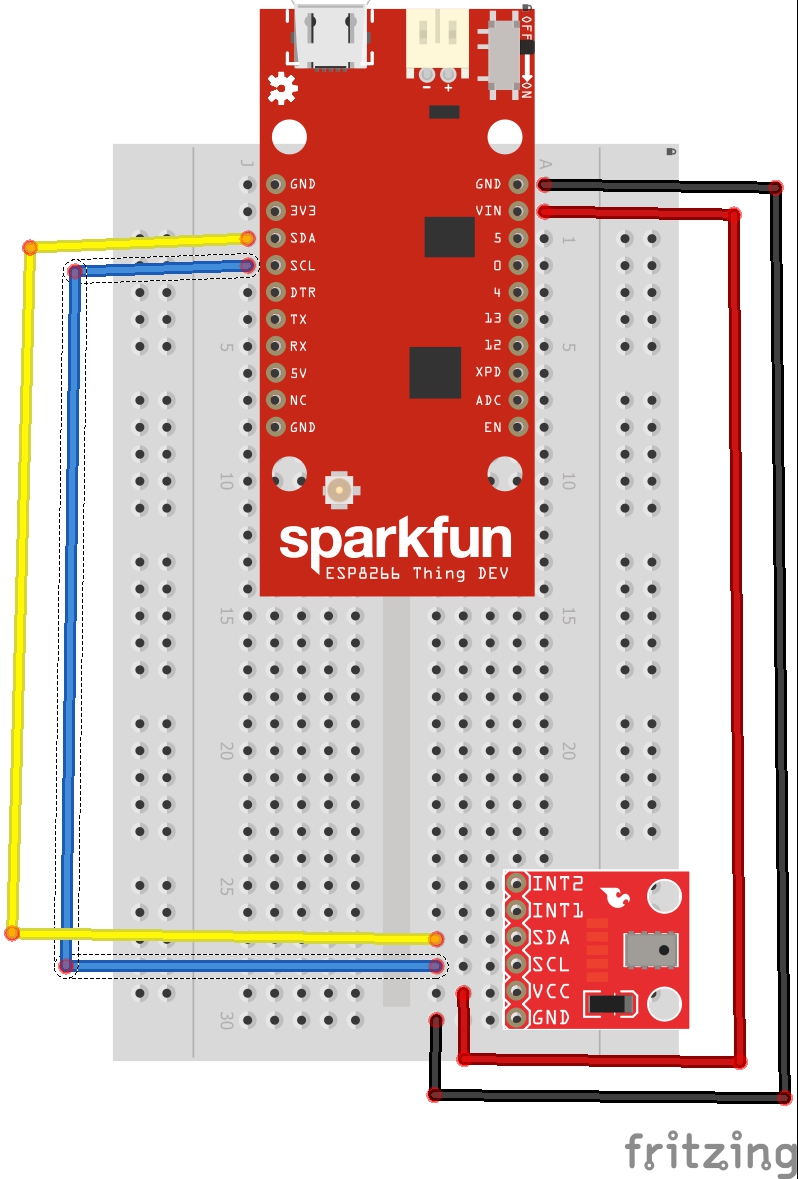
It should now read like this:



**Now for the hardware**

The hookup is as follows:

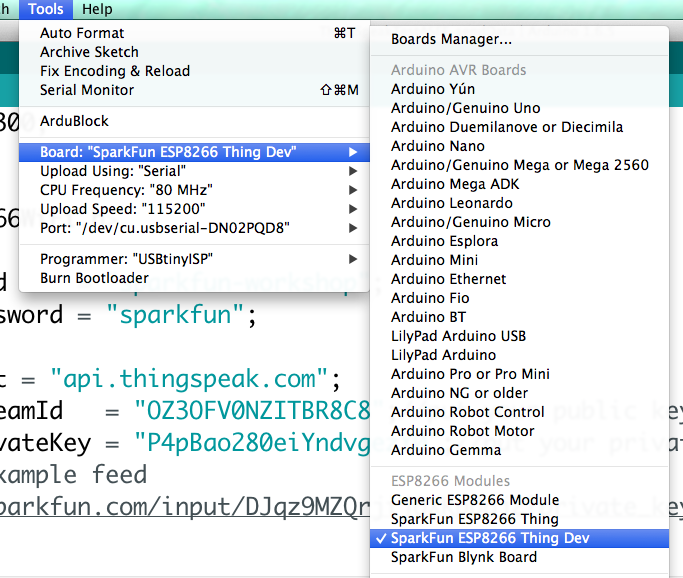


****

**Now we’re ready for upload**

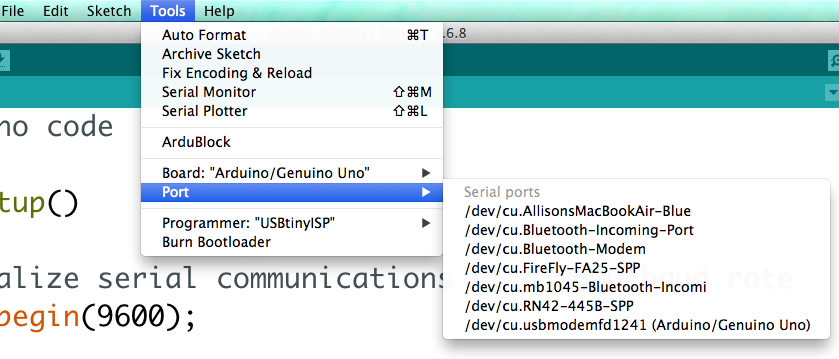
Plug the usb cable into the ESP8266 and the other end into your laptop.

From the “tools” menu in Arduino select “boards” and then find “SparkFun ESP8266 ThingDev”, click on it:

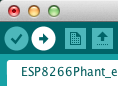


Now we’ll need to select the “Port”.

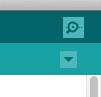
From the tools menu, select port and in windows look for a com port numbered above 3 for Mac it will be a /dev/cu.usbserialxxxxx(some hex #)

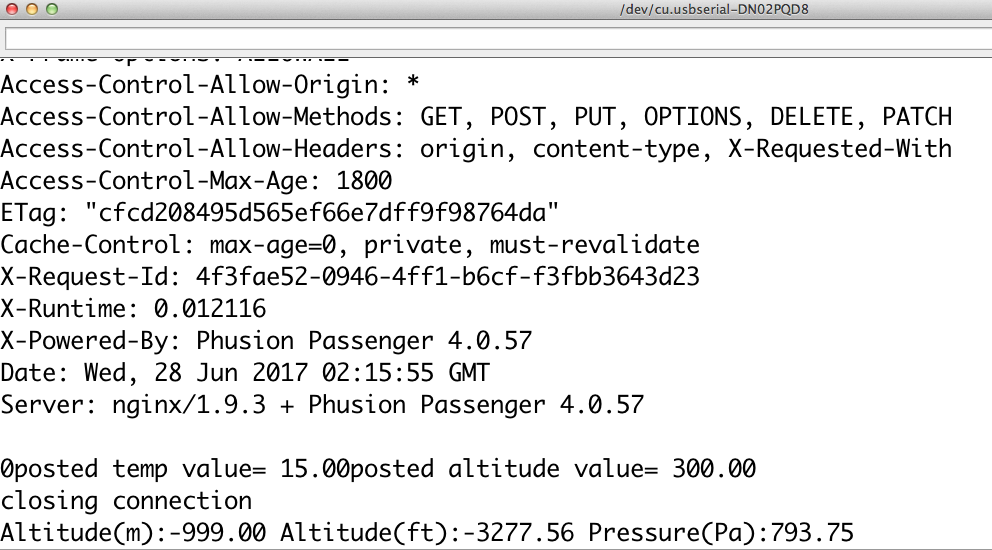
:

Now we can upload the code, click on the upload button on the tool bar:



The code will take a little while to upload, when it’s done we’ll open the serial terminal( the magnifying glass in the upper left hand corner) and check the return message from ThinkSpeak:





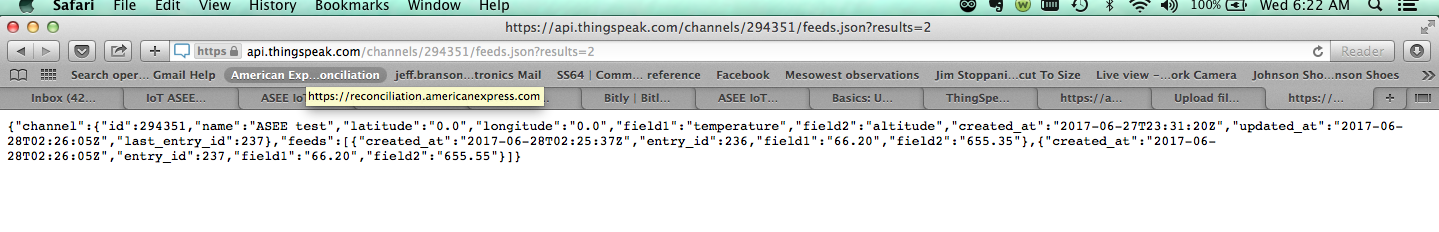
Now check your public channel to see if temperature and altitude are posting.

**Getting data from the API**

In the same way we can send dat we can also request data. From your API Keys tab on your channel page you can grab a sample API request from the box at the bottom that says “Get Channel Feed”. You’ll grab the URL portion That looks like this:

<https://api.thingspeak.com/channels/294351/feeds.json?results=2>

If you plug this into a web browser and hit enter, your browser will display the returned data structure:



If you read through this, you’ll see all the different data returned by ThingSpeak.

For our example we’re only interested in the temperature, so we’ll restructure the request and resend it to see what we get back.

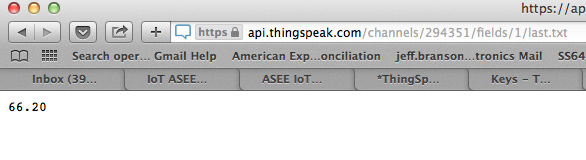
I will take the “Get Channel Feed” example

<https://api.thingspeak.com/channels/294351/feeds/1/Last.text>

https://api.thingspeak.com/channels/294351/fields/1/last.txt

and append the end to like like the second example above.

When I run this example URL, i get a significantly trimmed return:

This is a much cleaner set of text to have to deal with parsing than the full return.

**Data retrieval from an API with Arduino**

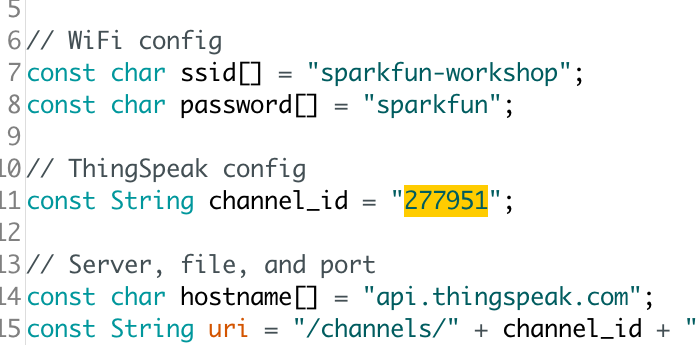
We’ve already sent data to ThingSpeak, let’s try asking the service for data from an Arduino and then performing a physical action with it.

With an ESP8266 and some example code we’ll build an analog thermometer using a servo motor.

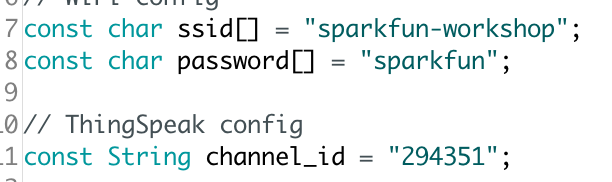
Here is the wiring of your circuit :

And now to code our example.

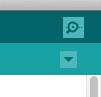
Open the “ThingSpeakTestASEE\_getData” example code with the Arduino editor.

We’ll need to change the example code to match your channel ID, here is what the example code looks like:

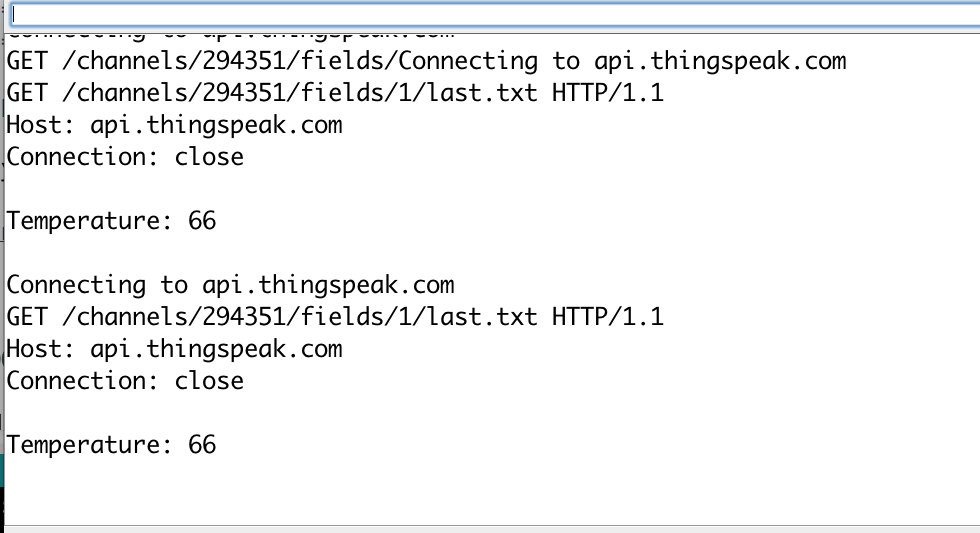
And here is the change using my test ID:

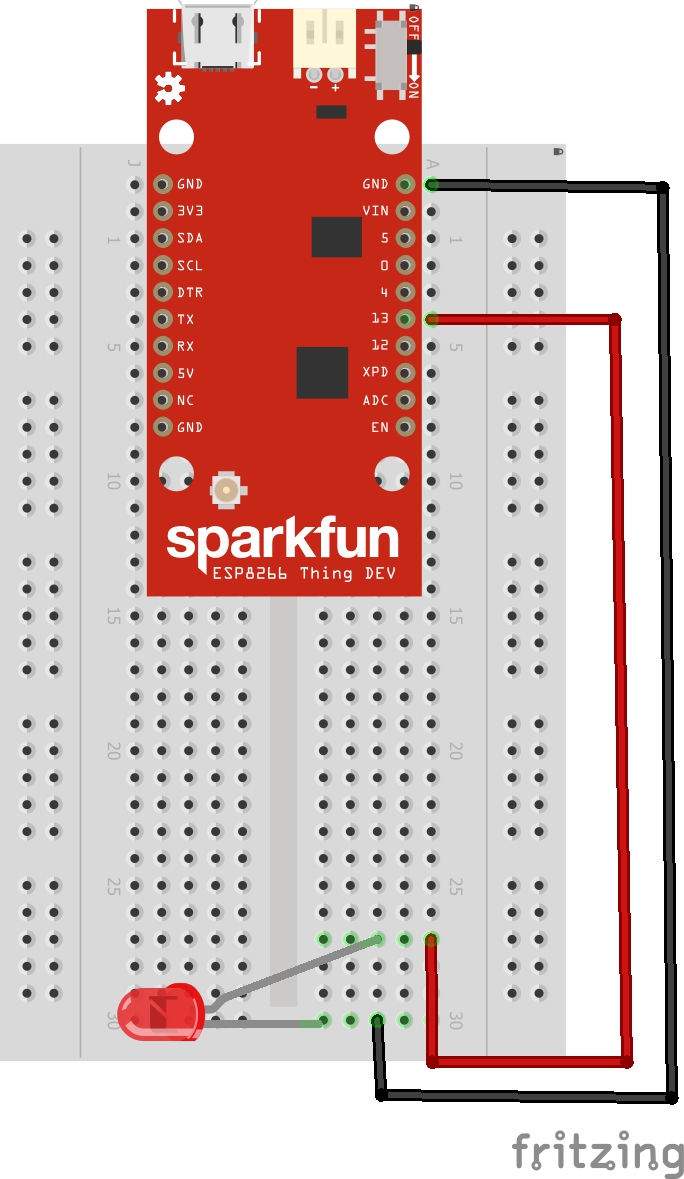


**Checking the feed**

I can check the returns from ThingSpeak to confirm that the wireless is connected and I’m reading the incoming data from ThingSpeak by opening the serial monitor with the magnifying glass icon in the upper right corner of Arduino:

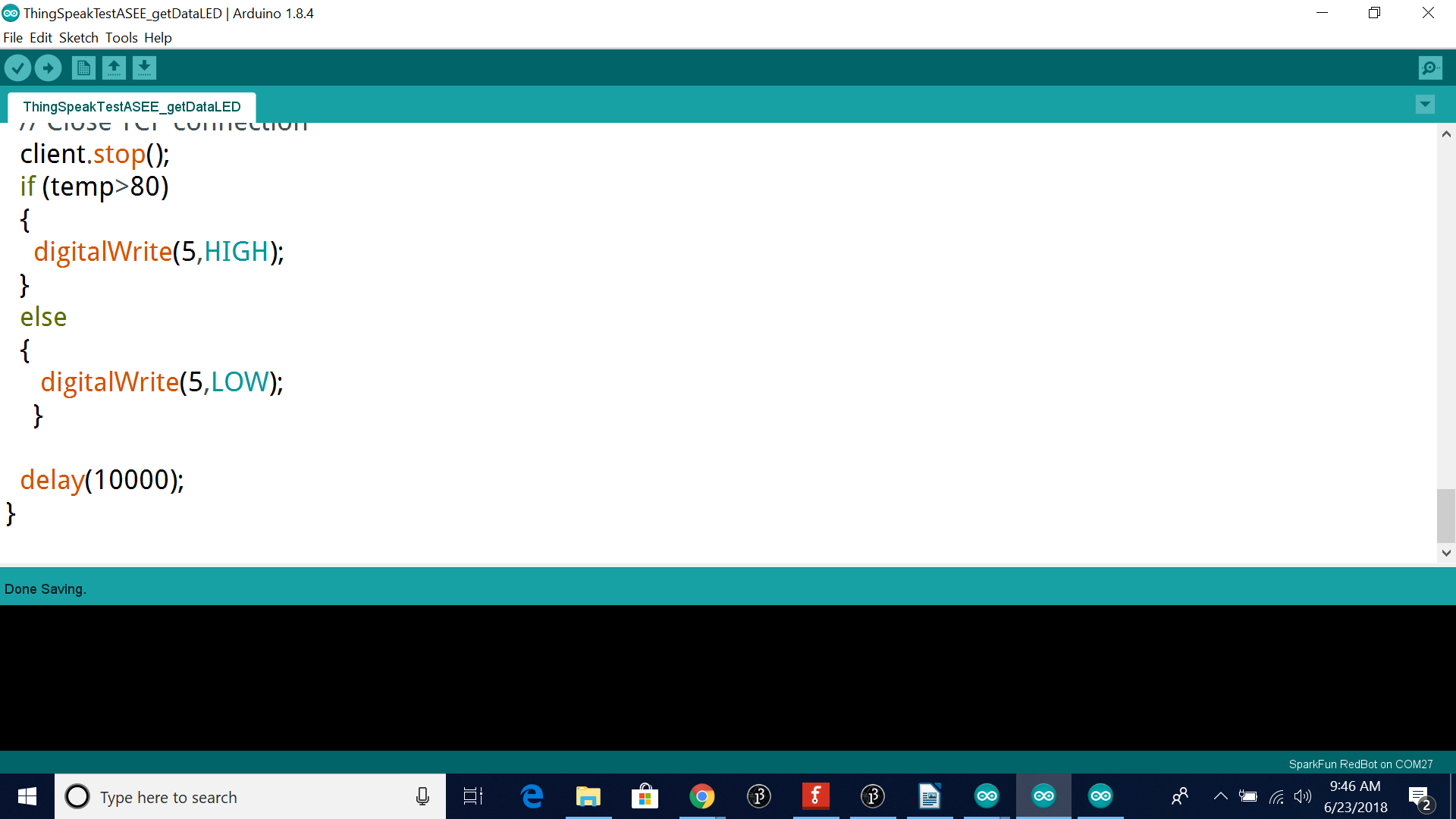
When I open the monitor, I should get this header:



I have built an if statement in the bottom of the code that checks tempeture and turns on an Led connected to pin 13 if the value rises above a certian level.

The wiring follows this diagram.

You can change the value in the if statement to reflect the conditions in your area.



**Processing and data**

I have built a couple simple example of getting dat into a stand-alone app use the Processing language.

Open the folde entitled “workshop processing code” and we can run these code examples from the Processing software installed on the drive.

***Make sure to enter your channel ID after “channels/”***

**Running a Python3 script in Trinket.io**

Python is extremely popular these days and I have built a small example of running our data in Trinket.io.

The code to enter into the Trinket window is as follows(make sure to enter *your* channel ID after “channels/”):

#start code

#make sure to enter your channel ID after “channels/”

import urllib.request

import json

contents = urllib.request.urlopen("https://api.thingspeak.com/channels/294351/fields/1/last.txt").read()

varab = json.loads(contents)

print(varab)

#end code

**Going further**

there are extra buttons and potentiometers in your kit, as well as LEDs, please feel free to experiment with the added hardware!

**Additional IoT ideas**

<http://bit.ly/dishiot>

Boards manager url(install in preferences)

<https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json>

