**SMART WEATHER STATION**

**Background:**

I am an international student and the weather in my country is consistent. You don’t need to see the weather every time. Here in Melbourne the weather keeps on changing every minute. It can be hot in afternoon and cold in the evening. Temperature can fall about 15 degrees in few hours.

**Problem Statement:**

Weather can change regularly in Melbourne. It can be difficult for a person to judge weather from inside the room. Setting up a weather station can be very expensive and requires a huge space.

**Requirements:**

There is a need to make a weather station that is cheap and small. It will click the picture of the weather to capture great moments. It needs to store the data to compare it and it needs to send a message if weather condition changes.

Designing Principles:

1. The models need to be small.
2. It needs to be cheap.
3. It needs to be safe from water as it is going to be kept outside.
4. It needs to be tested for various conditions.
5. Raspberry pi should get proper wifi signal.

Prototype Architecture:

I have used various sensors available in market. It includes tipping bucket rain gauge, wind vane, wind gauze, DHT11 (temperature sensor), LDR (Light sensor) and more.

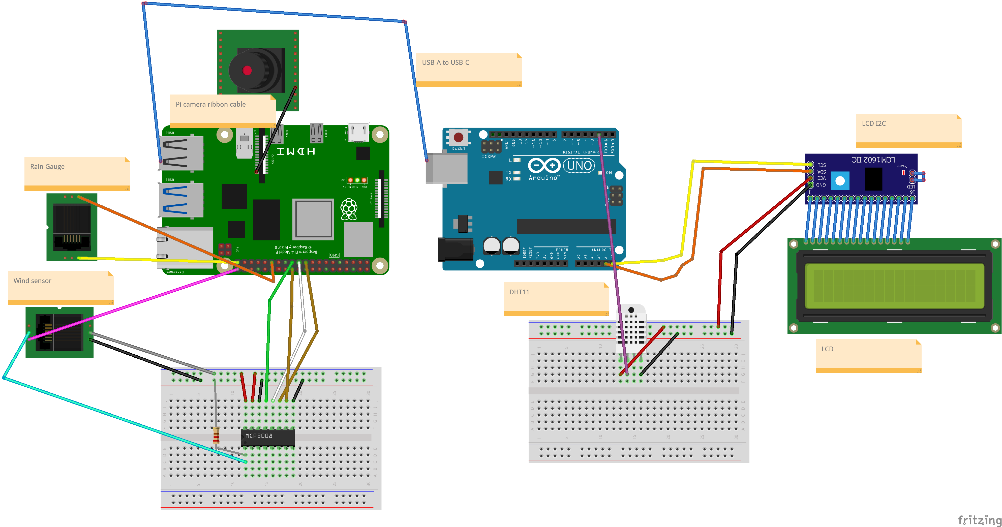


Fig 1

The above picture shows the circuit diagram I have used.

I have setup the wiring in the waterproof box with some holes to let the wires go and out for the sensors and power.

I have used MCP3008 which is used to convert analog signals to digital.

Here I have connected wind wane with wind anemometer to reduce the number of wires and then the wire from wind anemometer contains the wire for both. Here rain gauge, wind wane and anemometer are connected to raspberry pi and temperature sensor is connected to Arduino. After measuring temperature Arduino sends data to raspberry pi using the wire.

Raspberry pi uses read serial function for it.

Then raspberry pi sends data to thingspeak where it creates the graph related to value.

How different sensors work.

Wind Anemometer:

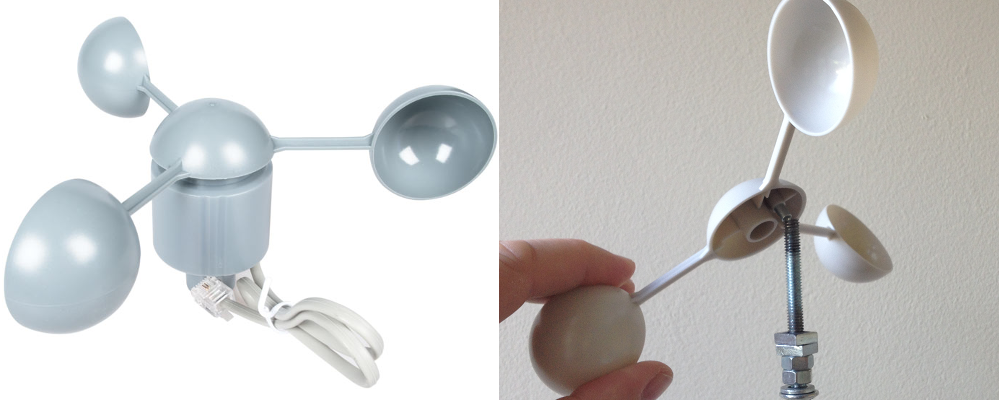


Fig 2

It measures the speed of the wind. It has little reed switch which works as a switch. It works on the principle of magnetic field. When the dial moves the reed switch will close and make the circuit closed and then it will make it open.

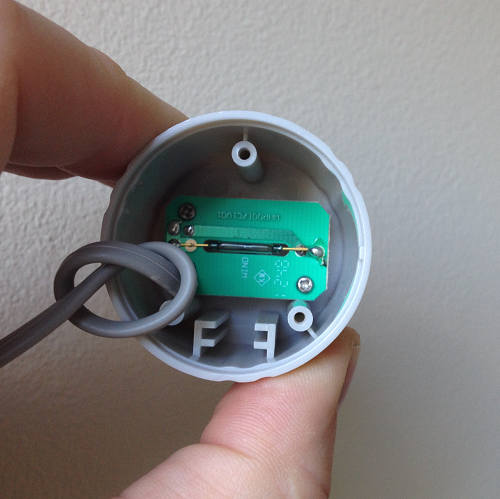


Fig 3

Here is how it is placed in the sensor.

In completing one circle the reed switch will close the circuit two times. We can count the number of times the current passes and divide it by two to get the number of rounds the propellor has turned. Then we can calculate the distance it has travelled and divide it by time to get the wind speed.

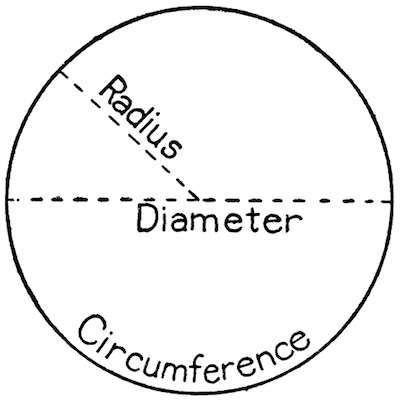


Fig 4

So we know the radius of propellor. We can calculate the distance travelled which is circumference as 2πr. Then we know the time(t) it has taken. We also know the circles it has made(n). So, it makes the formula for wind speed as.

N X 2π X r/t

This will give us the speed of the wind.

Wind Vane:



Fig 5

This also works on the reed switches. But it uses 8 reed switches. The direction in which reed switch is on is the direction the wind is flowing. Different reed switches has different resistances and hence making different voltages flowing through each point.

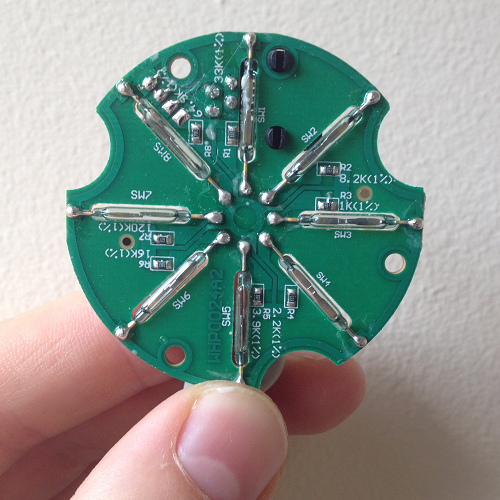


Fig 6

Here is the wind vane board.

There can be times when vane is between two reeds and can activate both together and hence releasing different voltages. Hence, there are mostly 16 types of voltages. The direction will be determined by using a separate resistor to calculate the voltage.

Rain Gauge:



Fig 7

Rain gauge collects the rain water and tells the amount of rain that has fallen. It works on the principle of tipping bucket.

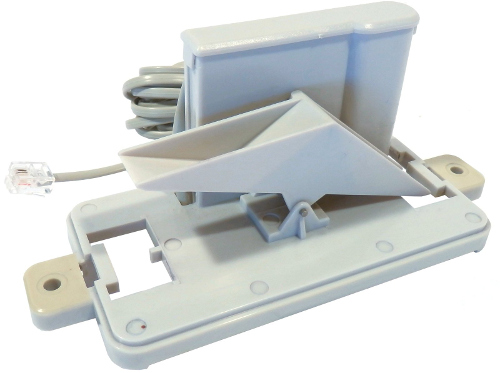


Fig 8

Water first enters the hole and starts collecting in small bucket risen top. When it gets full due to the heavy weight it goes down and removes all water. Then same thing happens to one now at top.

There are tipping sensors attached to the two points to calculate how many time the bucket has been tipped. The number of tips then can be multiplied by amount of water stored by one to get the amount of rain that has fallen

GitHub link:

<https://github.com/kinshuk9449/weather>

Testing Approach:

I have tried my project under various conditions. I kept it outside in the real weather and checked the temperature with the weather app in mobile.

I also tried to try different things like putting water myself in rain gauge. I applied some pressure to turn the anemometer. Then I turned the wind vane at various locations.

User Manual:



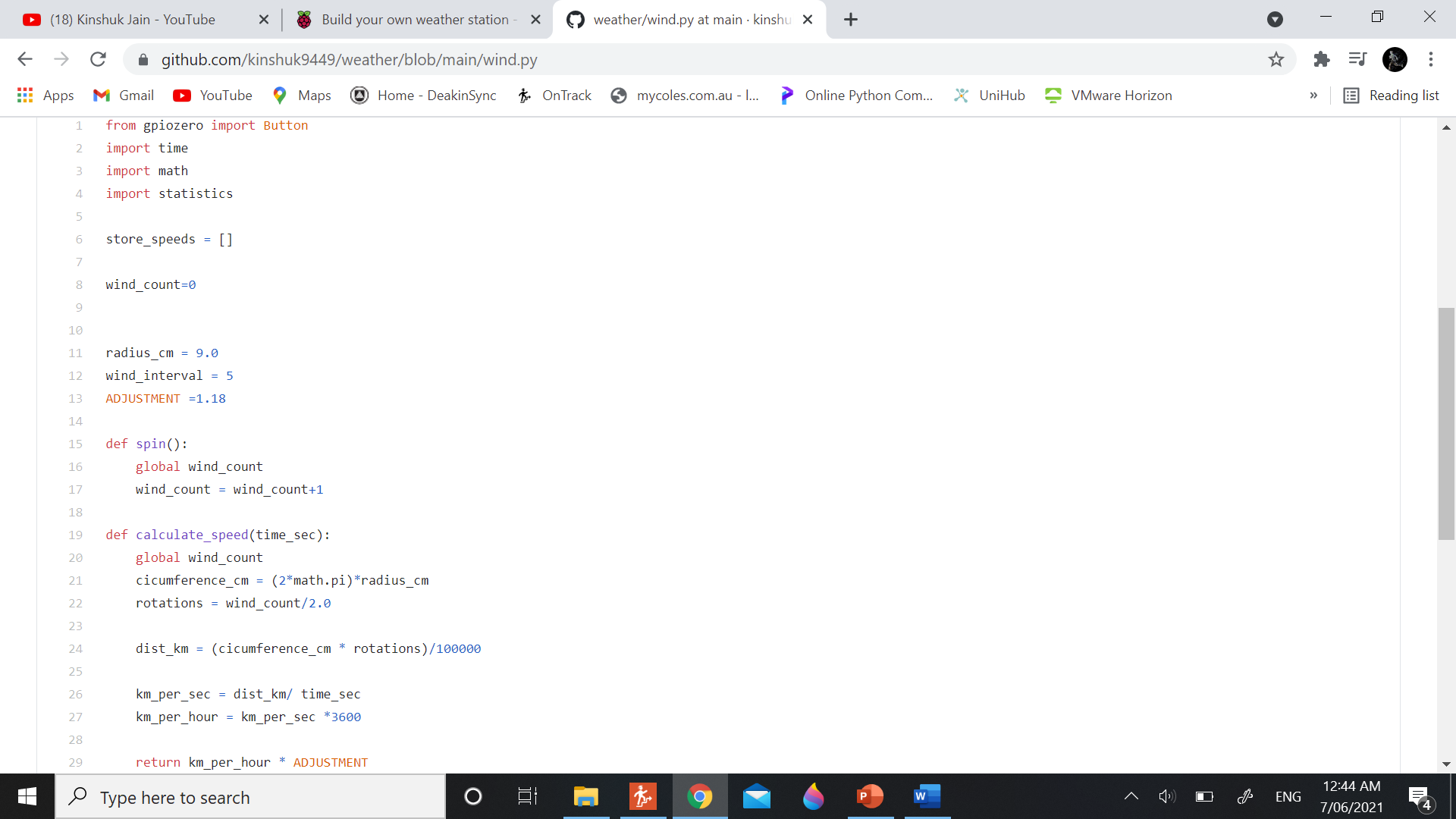
Fig 9

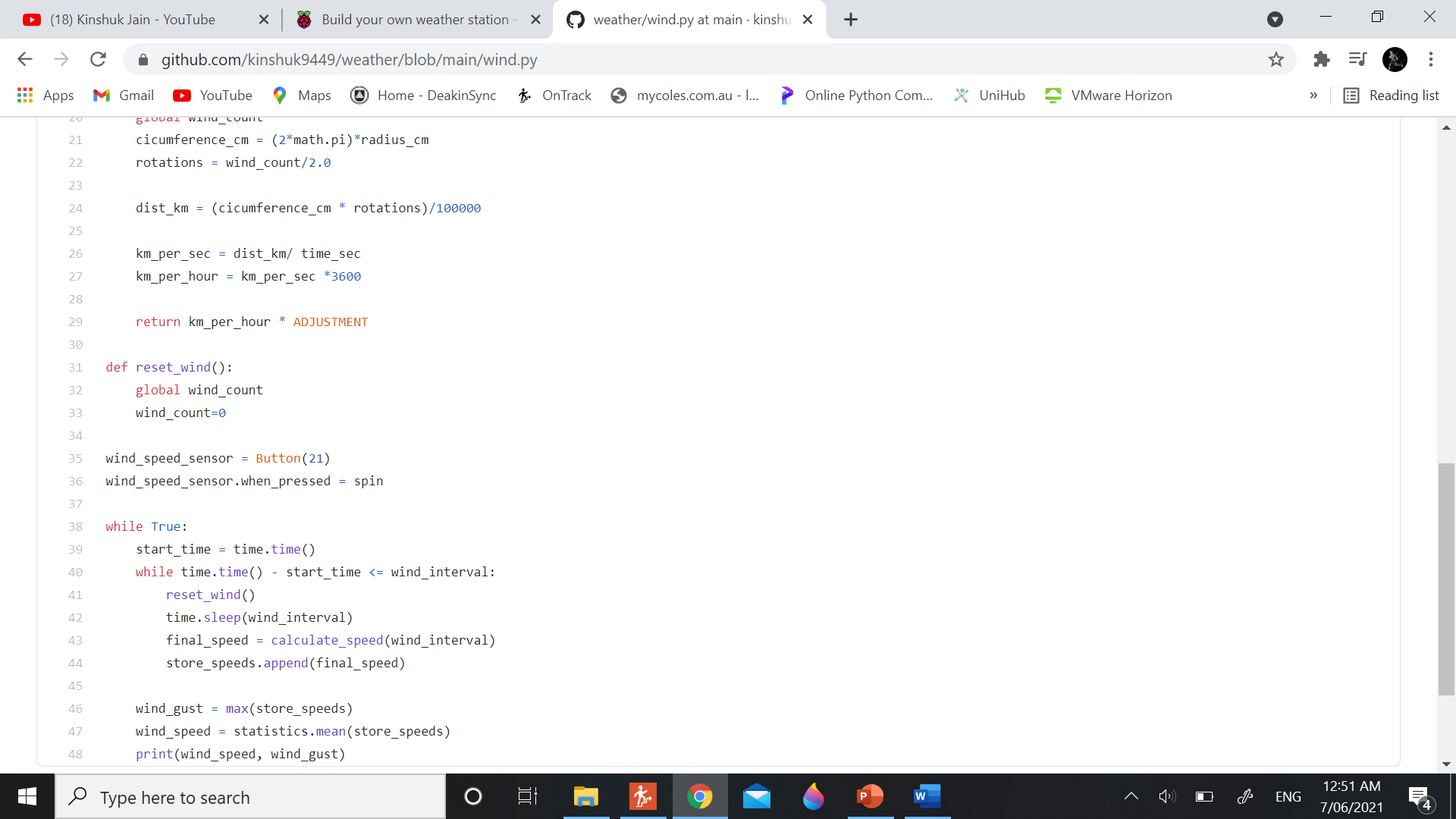
Wind anemometer:

The wire for wind anemometer is the third and fourth in RJ11. Either we can get RJ11 breakout or we can remove RJ11 connector and connect wires directly. Third wire will be connected to GPIO21 and fourth wire to GND.

As talked, we will calculate the number of times the circuit has closed.

For this we will be using gpiozero library which makes the sensor act as a button and counts the number of times it has been pressed. It makes the process easy.





Here we have created a function called spins which increases the count of spins whenever it is called. We call this function whenever the sensor is turned on.

Calculate speed calculates the speed of the wind depending upon the distance/time. Time taken is given as an input to this function. As the count will be twice the number of actual rotations, we will divide the spins by 2 to get the number of rotations.

As the radius and time is in cm and sec respectively. Hence, we convert the speed in km/h by multiplying the result by 3600/100000.

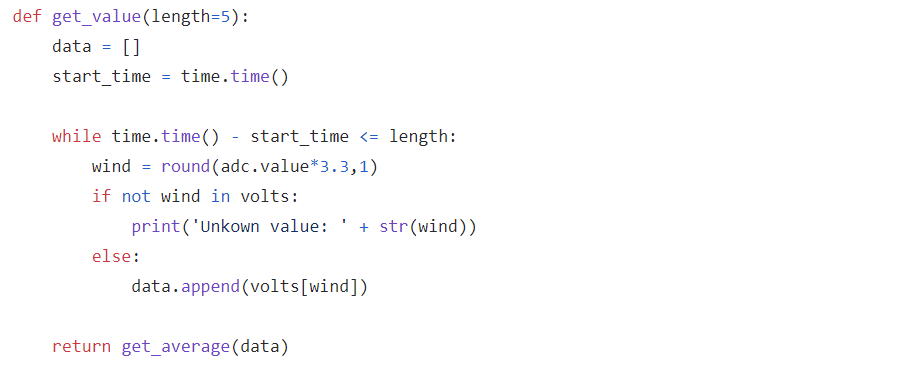
Wind Vane:

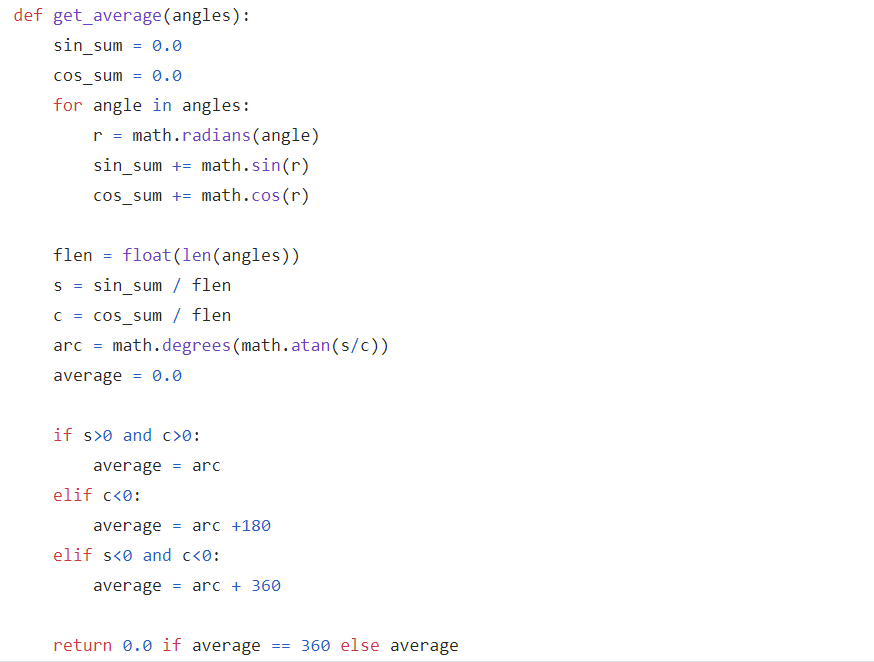
Second and fifth point in RJ11 connector are for the wind vane. But wind vane gives the data in analog, so we need to convert it into digital value. We use MCP3008 as converter. MCP3008 will be connected as shown in the circuit.

Second wire from RJ11 will be connected to first pin of analog side of MCP3008. A 4.7K resistor will also be connected with A0 of MCP3008 and gnd. Fifth wire from wind sensor is connected to 5V of raspberry pi.



These are some of the voltages that are produced by wind vane. We have set the dictionary which records the voltage and angle related to it. The voltages can be calculated using divider.py file by changing the resistance that has been added in the circuit.





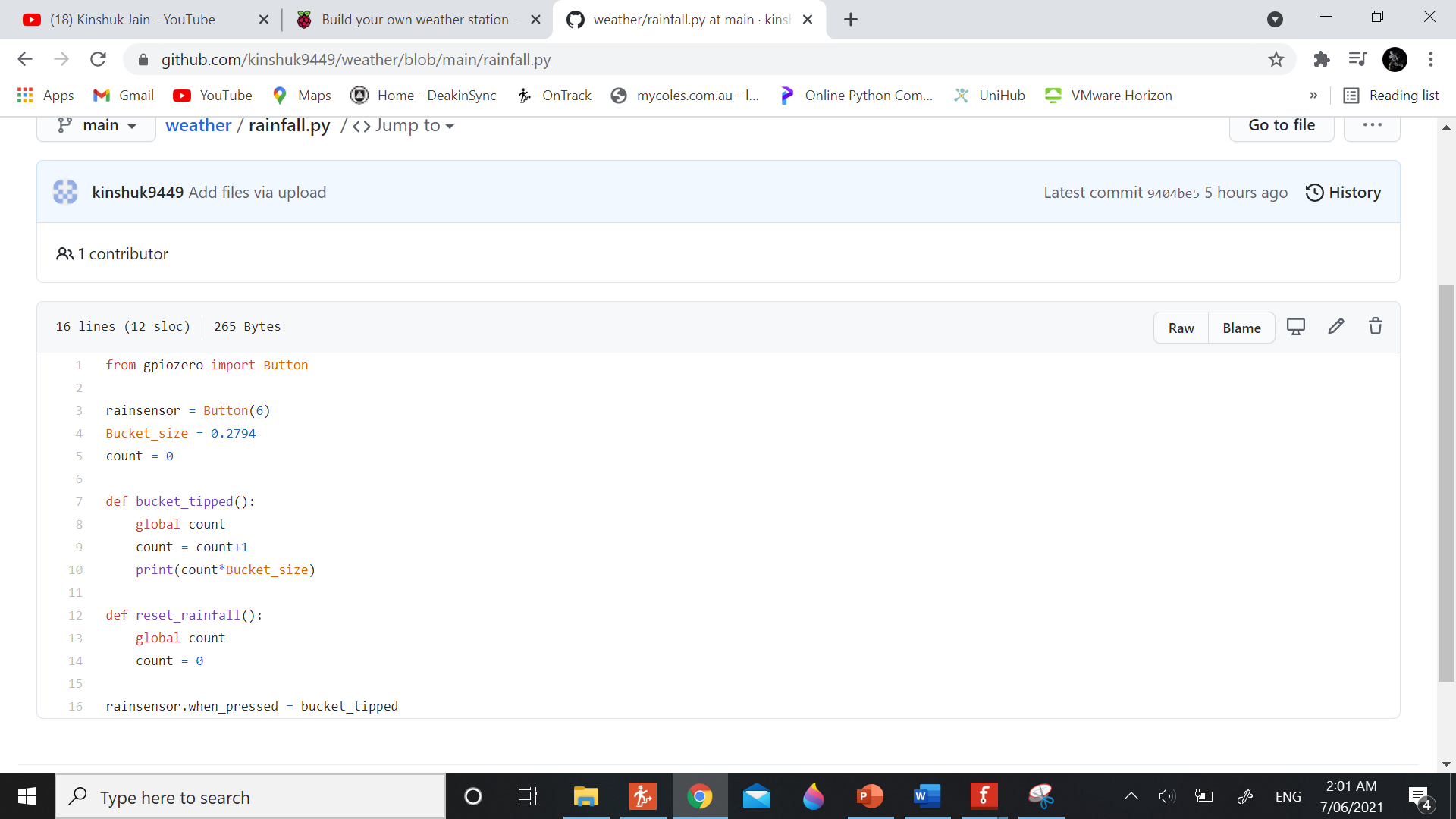
This is a normal code to calculate the angle as we just see if the volts in inside the list or not. If it is there it uses the angle next to it.

To calculate more accurate angle, we calculate the wind angle for five seconds and then we get the average of it. Math behind this a little complicated which can be seen from the site given below.

<https://en.wikipedia.org/wiki/Directional_statistics>

Rainfall:

Rainfall sensor also has a RJ11 connector. There are only two cables in it. Third wire will connect to GND and fourth wire will connect to GPIO6.



This is the simple code for calculating the number of times the bucket has tipped. It also uses gpiozero which makes the process very easy.

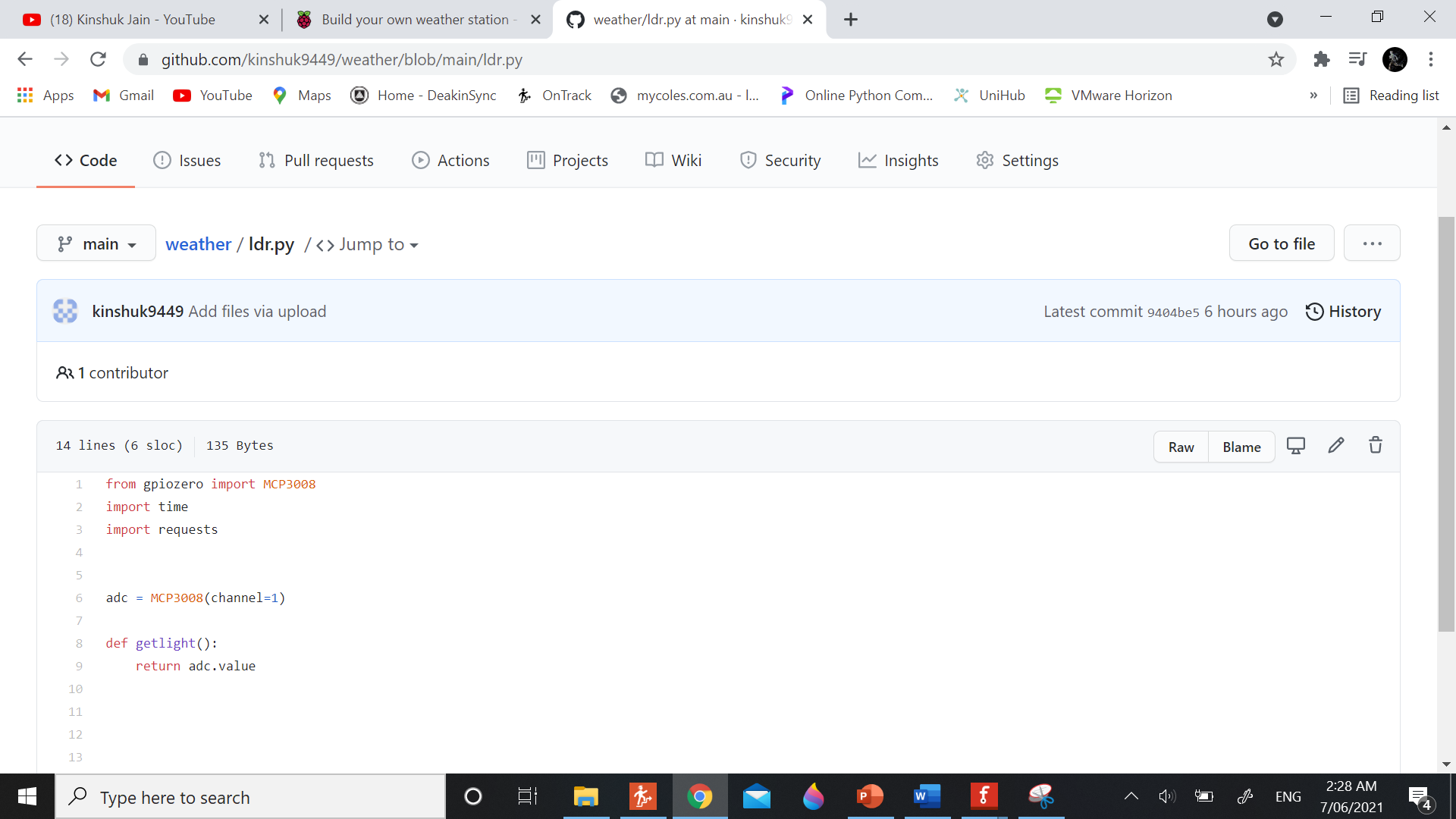
It counts the number of times the bucket has tipped and multiply it with 0.2794 (amount of water bucket can hold).

LDR:



Fig 10

It detects the amount of light being put on it. We will be using the other empty port of MCP3008 for this. We need to connect ldr’s one pin with 10k resistor which further will connect to gnd and also to A1 of MCP3008. Second pin of ldr will be connected to 5v.



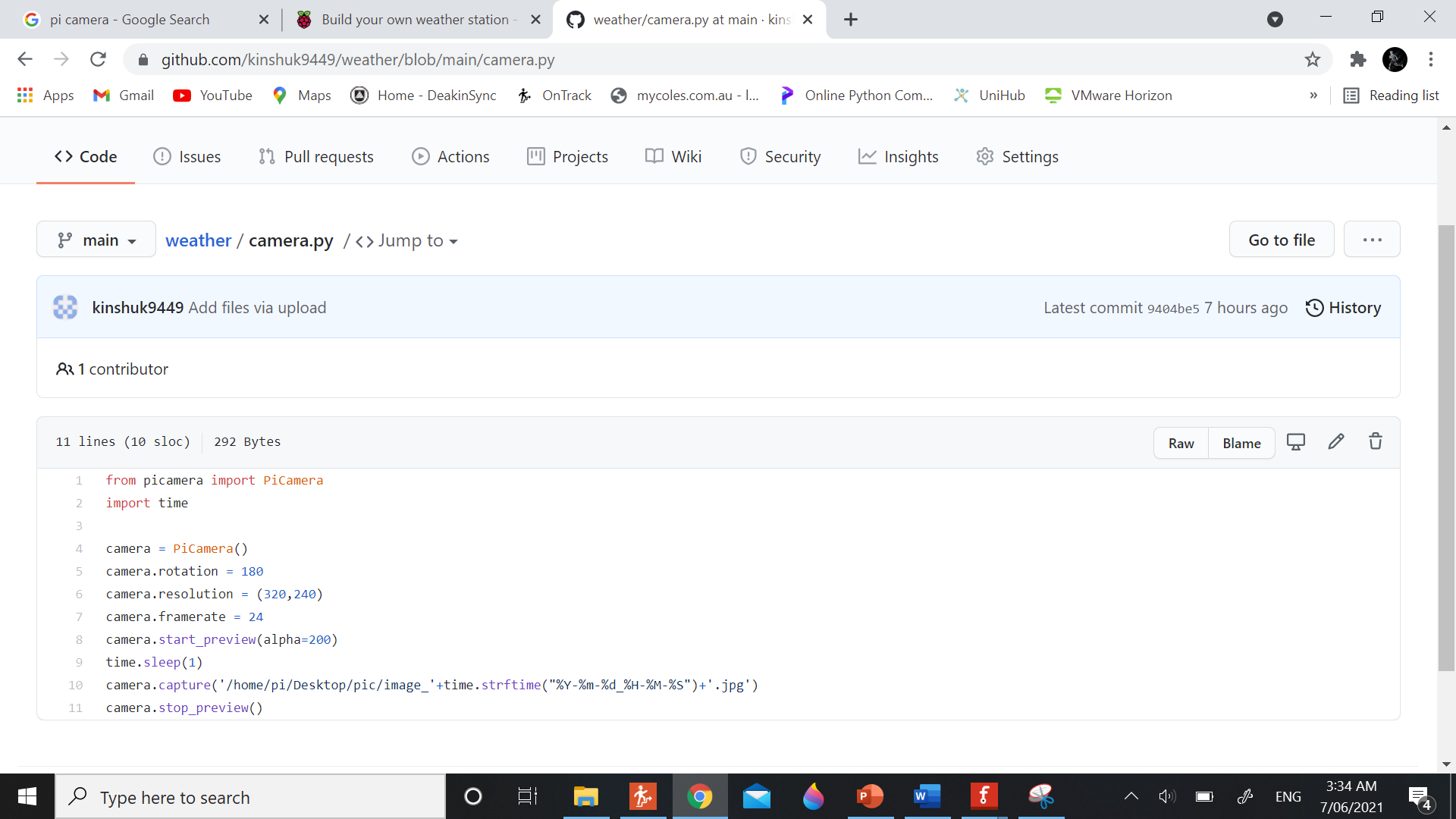
It is just a simple code that reads the data from sensor.

Pi Camera:



Fig 11

This a pi camera which is connected to the small socket provided in raspberry. You just need to insert the strap inside the board.



This code that helps to capture image and store it with different name every time so that picture doesn’t get overlap. To click picture we need one second of sleep to allow camera to see colours properly.

DHT11:

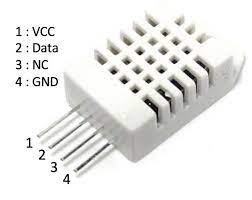


Fig 12

This is the sensor which detects the humidity and temperature around the sensor. This sensor will be connected to Arduino as we want to divide the voltage.

Below is how the sensor will be connected.

VCC -> 5V

Data -> 2

NC -> Null

GND -> GND

There will also be lcd that will be connected to Arduino. I have used I2C for lcd so that I don’t have to use so many connections with Arduino.



Fig 13

Here is the connection of LCD with Arduino:

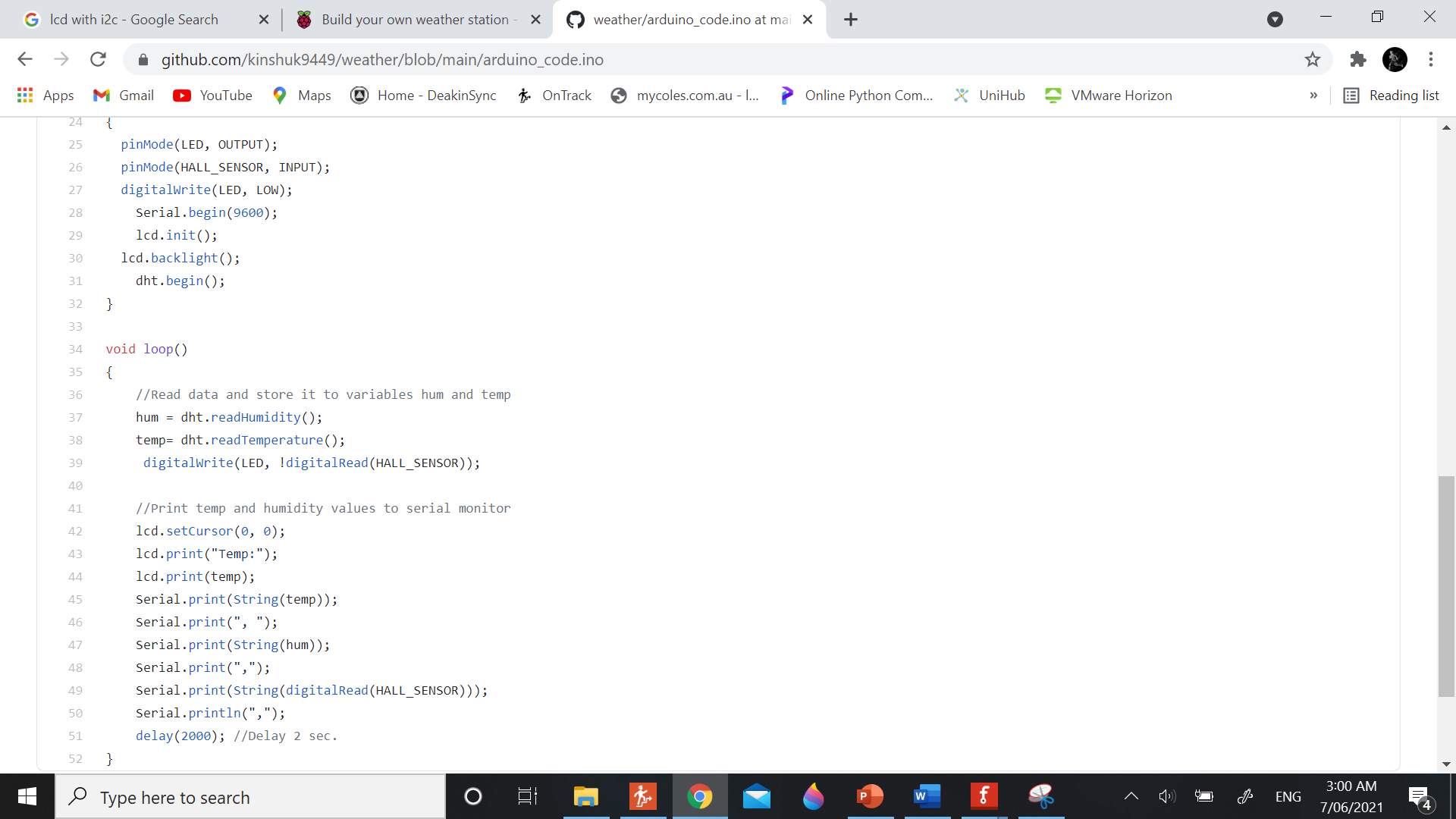
SCL -> A5

SDA -> A4

GND -> GND

VCC -> 5V

LCD will be used to display the temperature at all time.



This is the code for the Arduino. Here we use dht module which helps to get the data from the sensor. Here we print the result on the serial monitor.

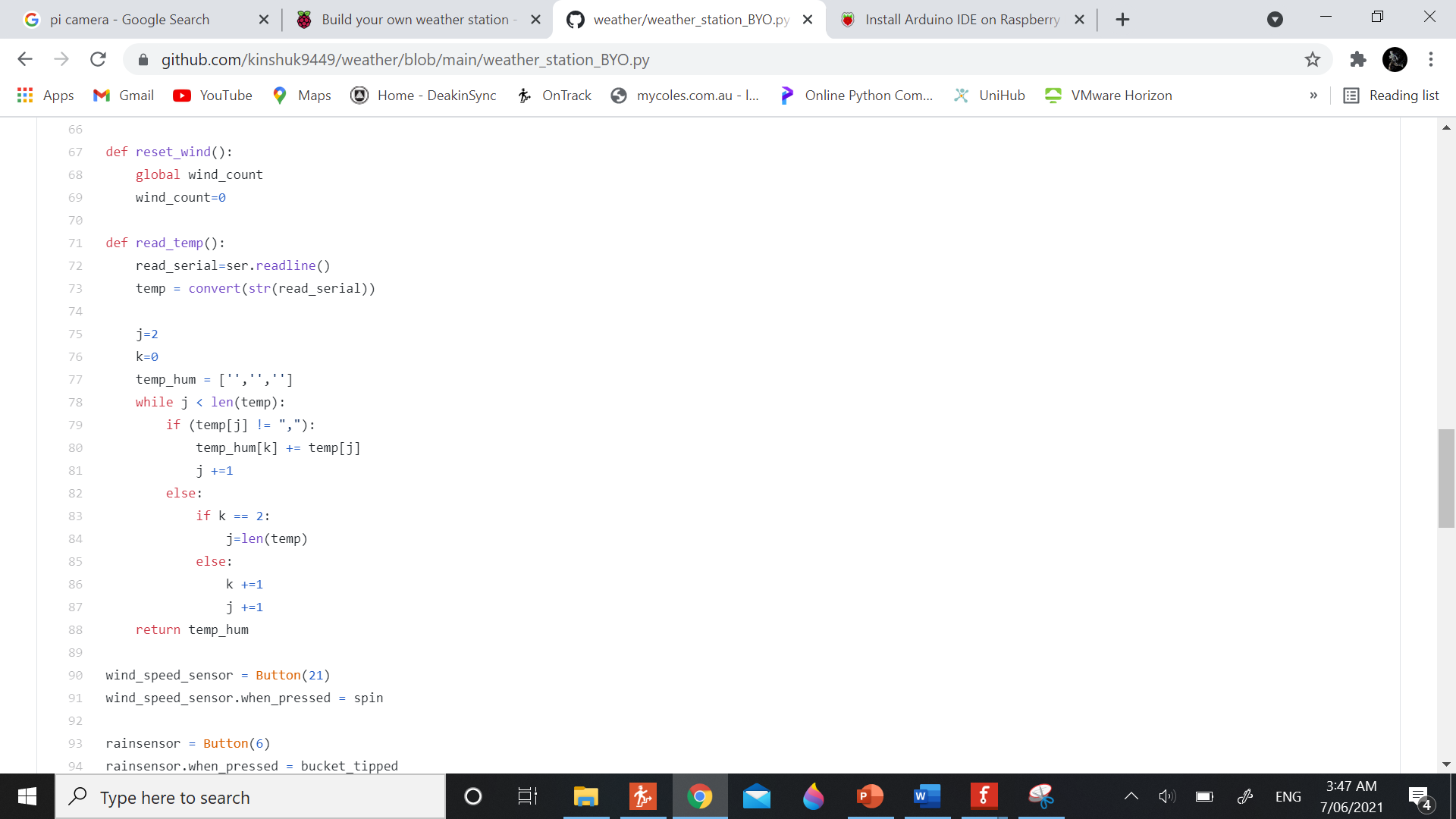
Connect Arduino and Raspberry pi:

To connect raspberry pi and Arduino you just need to take the cable of Arduino and plug it into raspberry pi port.

You can download Arduino in raspberry pi by following the tutorial given on the site given below:

<https://www.raspberrypi-spy.co.uk/2020/12/install-arduino-ide-on-raspberry-pi/>

We will be using read serial function in raspberry pi to read the data from Arduino. This is the reason we are printing the content on serial monitor in Arduino. Whenever there will be something on Arduino serial monitor it will be read by raspberry pi.



This function will read the temperature and humidity and store it in the array.

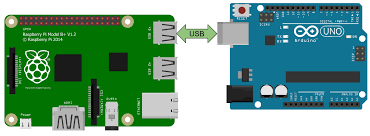


Fig 14

This is the way they will be connected.

Upload data on web:

I have used thingspeak to upload data. You need to create an account on thing speak and then create a new channel. There you add different fields.

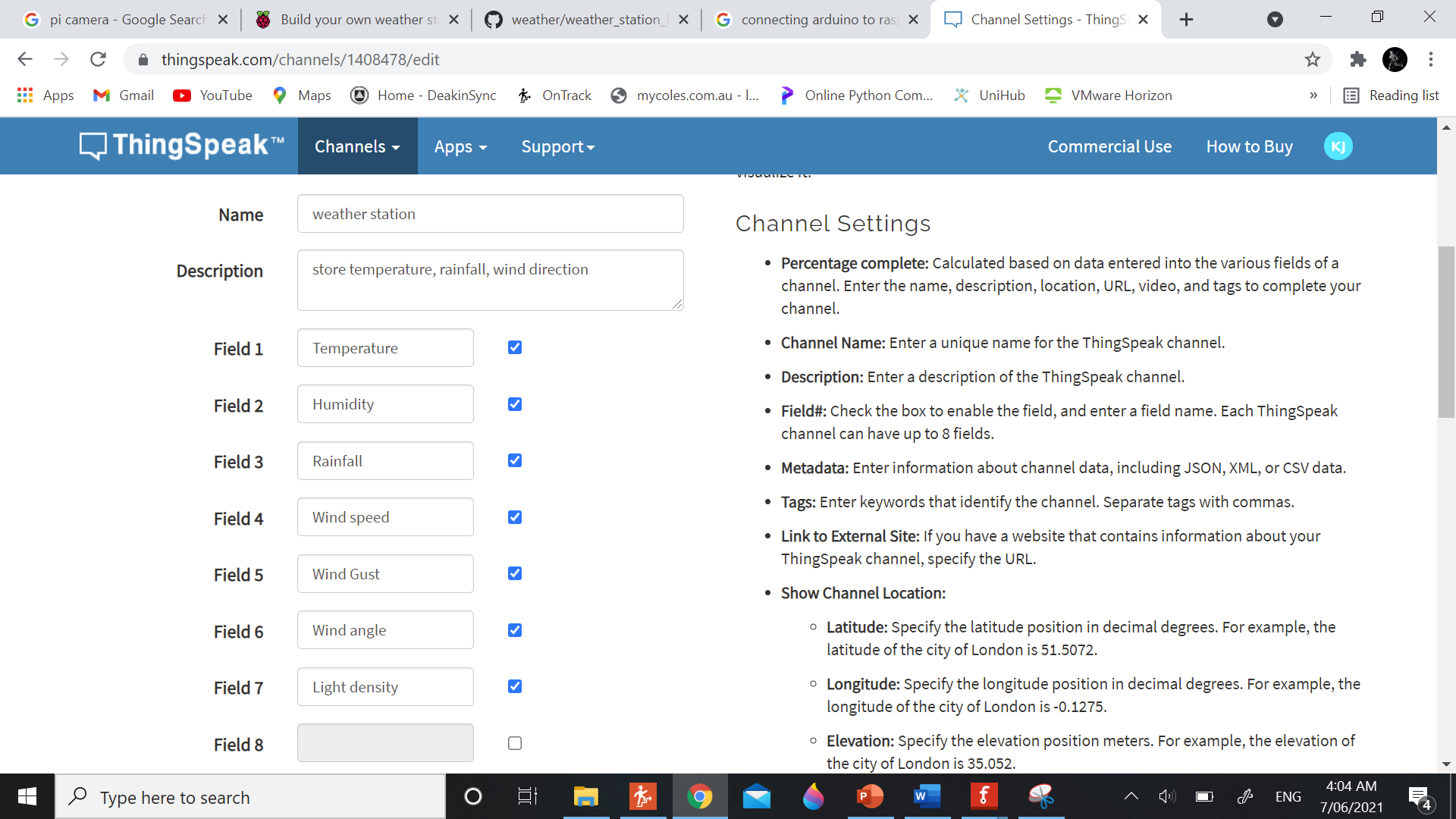


Fig 15

These are the fields that I have created.

Then you need to go the API keys and note the API.

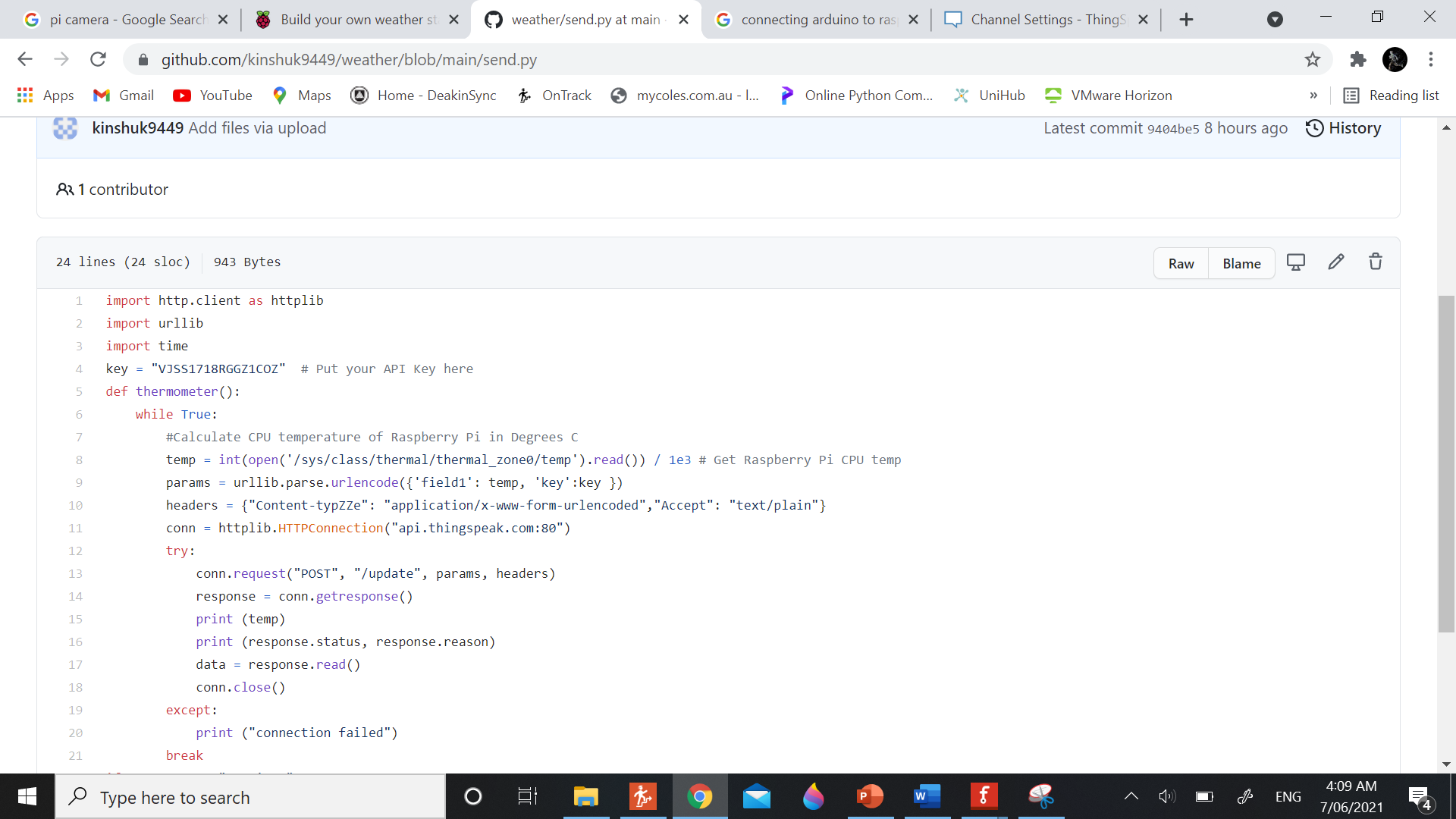


Fig 16

This is an example code. Here you need to add your API in the place of key. Also, you need to add more fields as in this it only posting one field.

So, you need to add total of seven fields storing specific values.

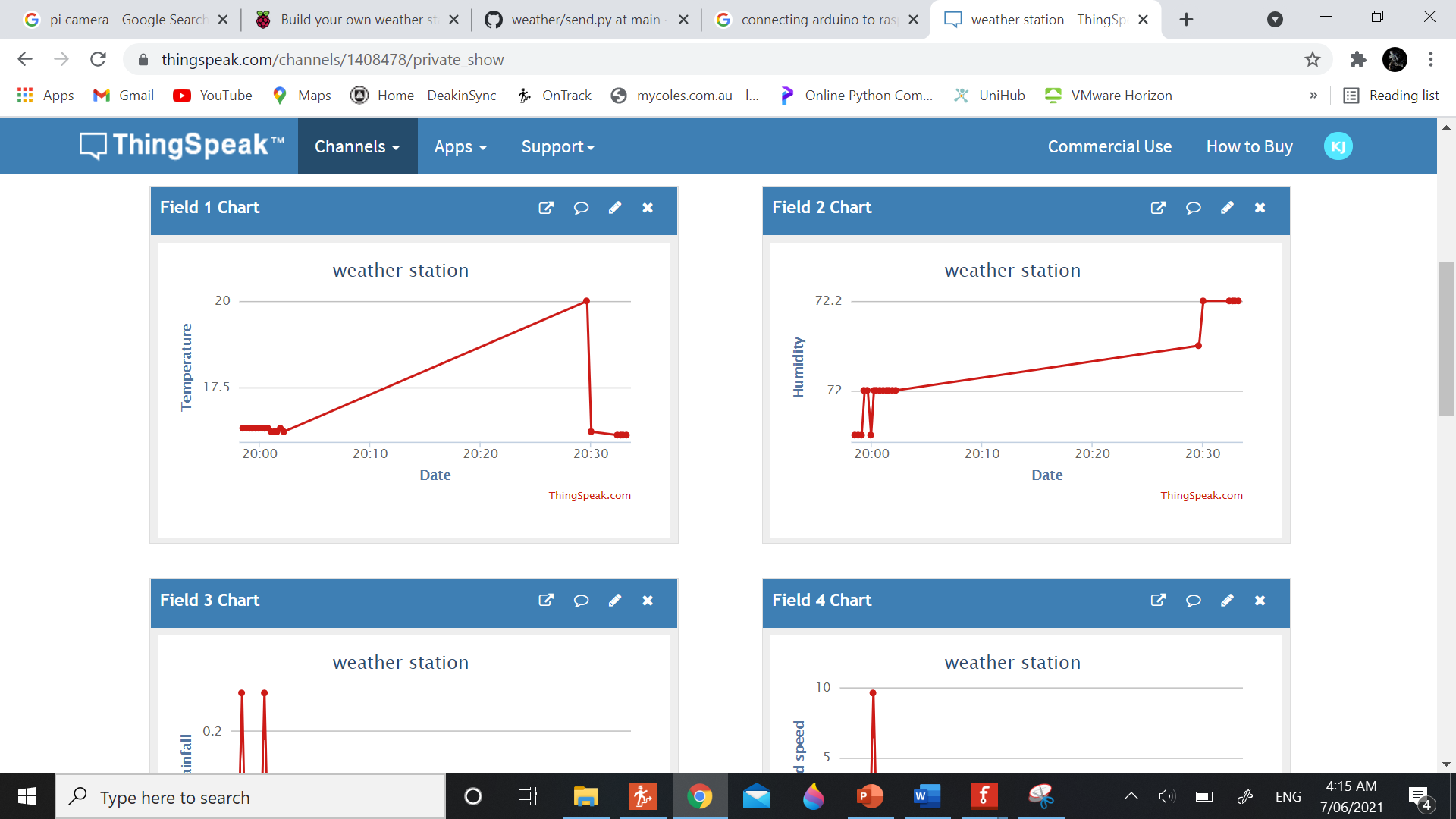


Fig 17

Sending notifications:

I have used IFTTT to send the notification. In IFTTT I am using webhooks services. Whenever an event is generated it sends the notification to the app.

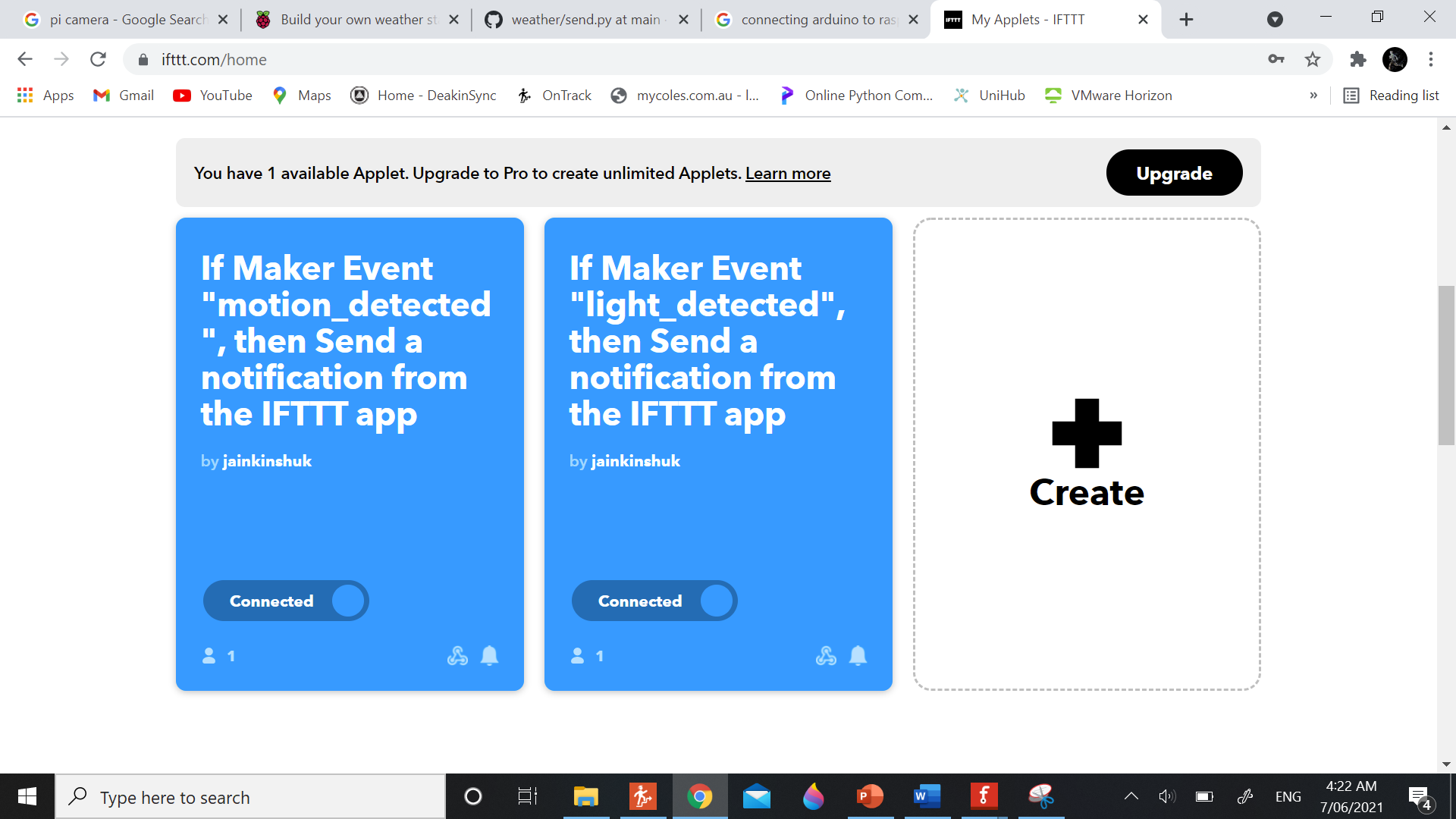
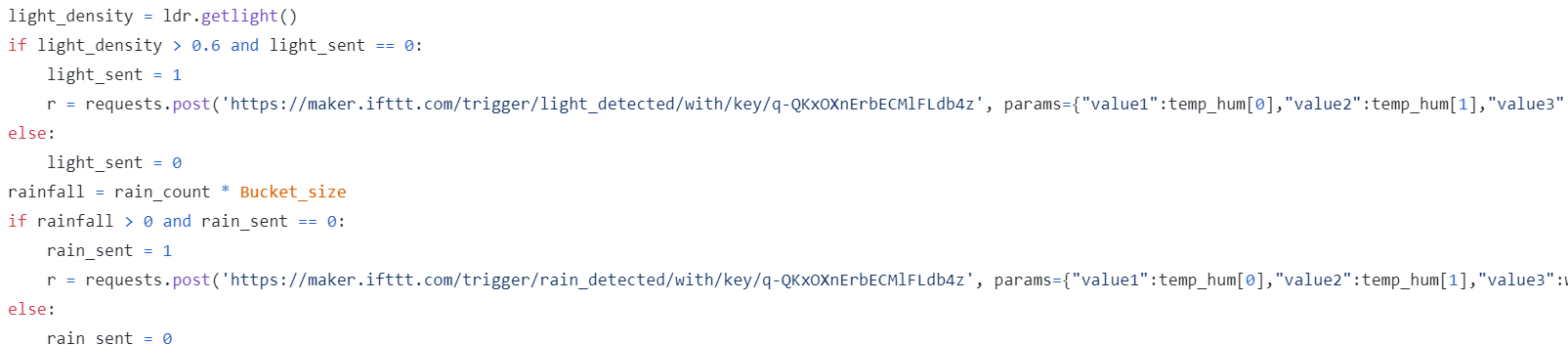


Fig 18

These are the applets that I have created. One is to see when it starts raining and other is to see when sun will rise.



This the code which sends the event online and then the notification is sent.

In the box you need to fill the unique key. It can be found by going to my services and then selecting webhooks. Then go to My documentation and then you can find the unique key.

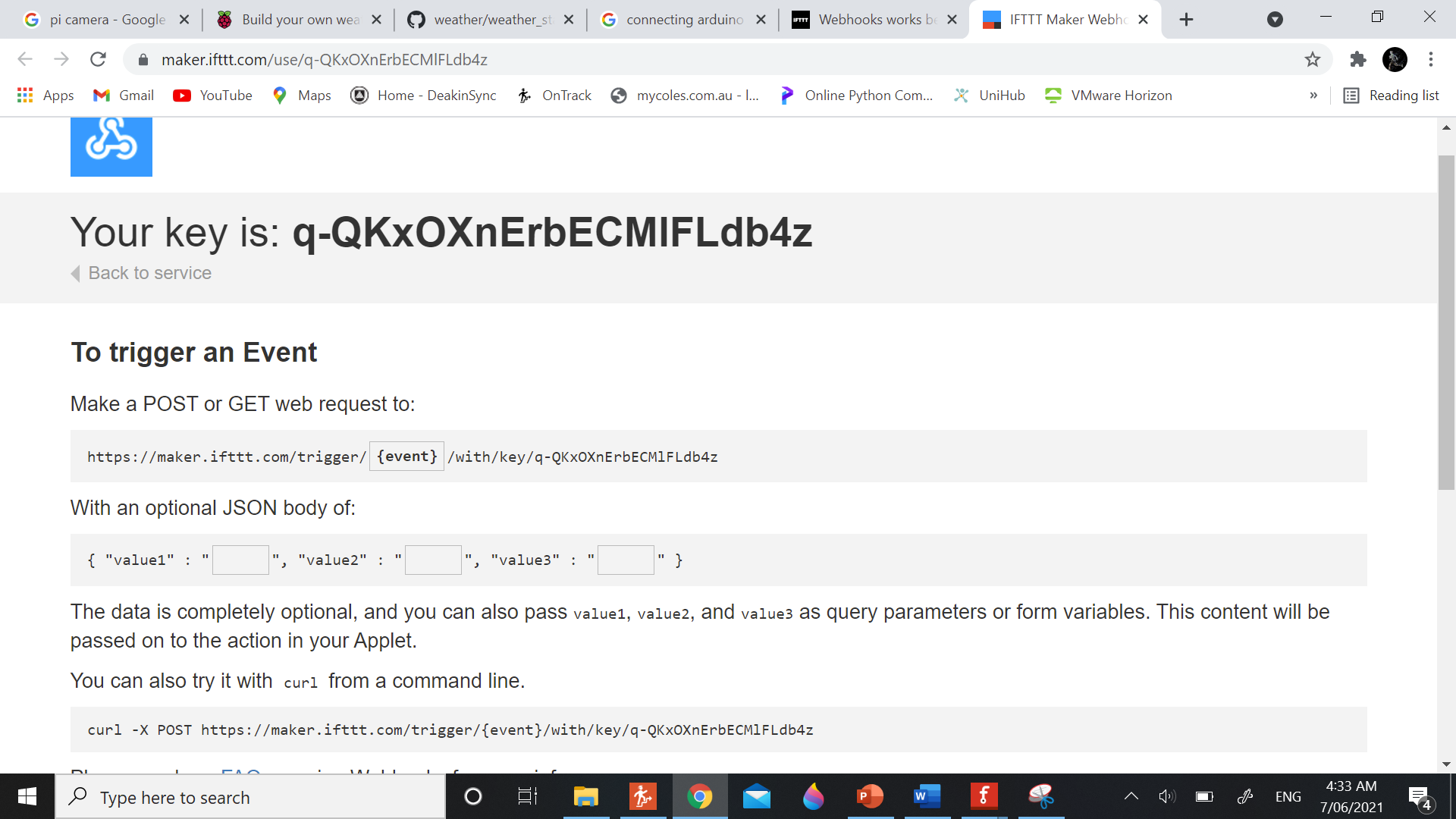


Fig 19

The place where there is rectangle will be your unique key.

Setting everything up:

Now you need to set everything up and combine the code in one file so that every sensor can work together.

Wind file will act as a base and then you can add the code and change the things a bit.

Complete file can be located here:

<https://github.com/kinshuk9449/weather/blob/main/weather_station_BYO.py>

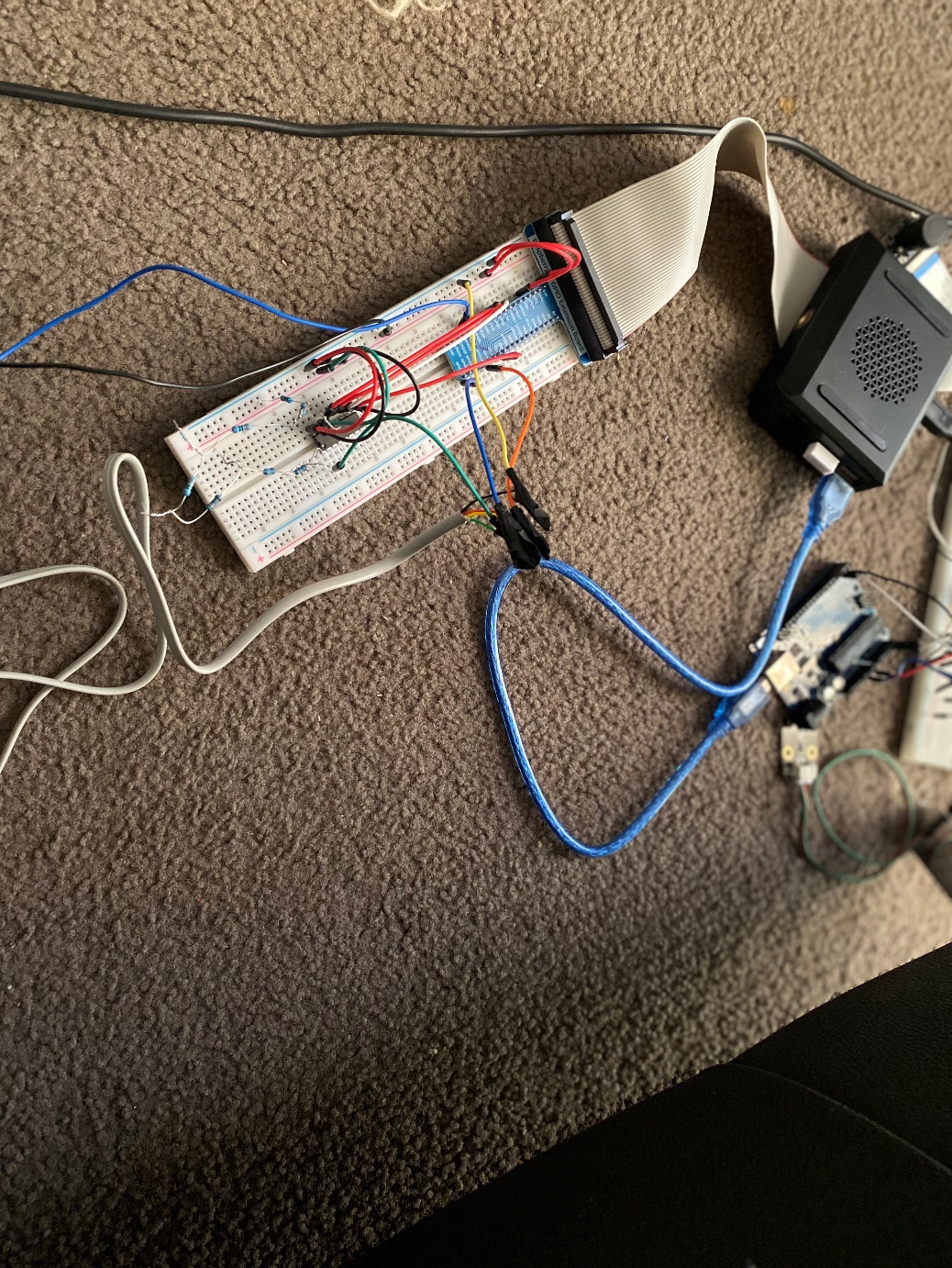


Fig 20



Fig 21

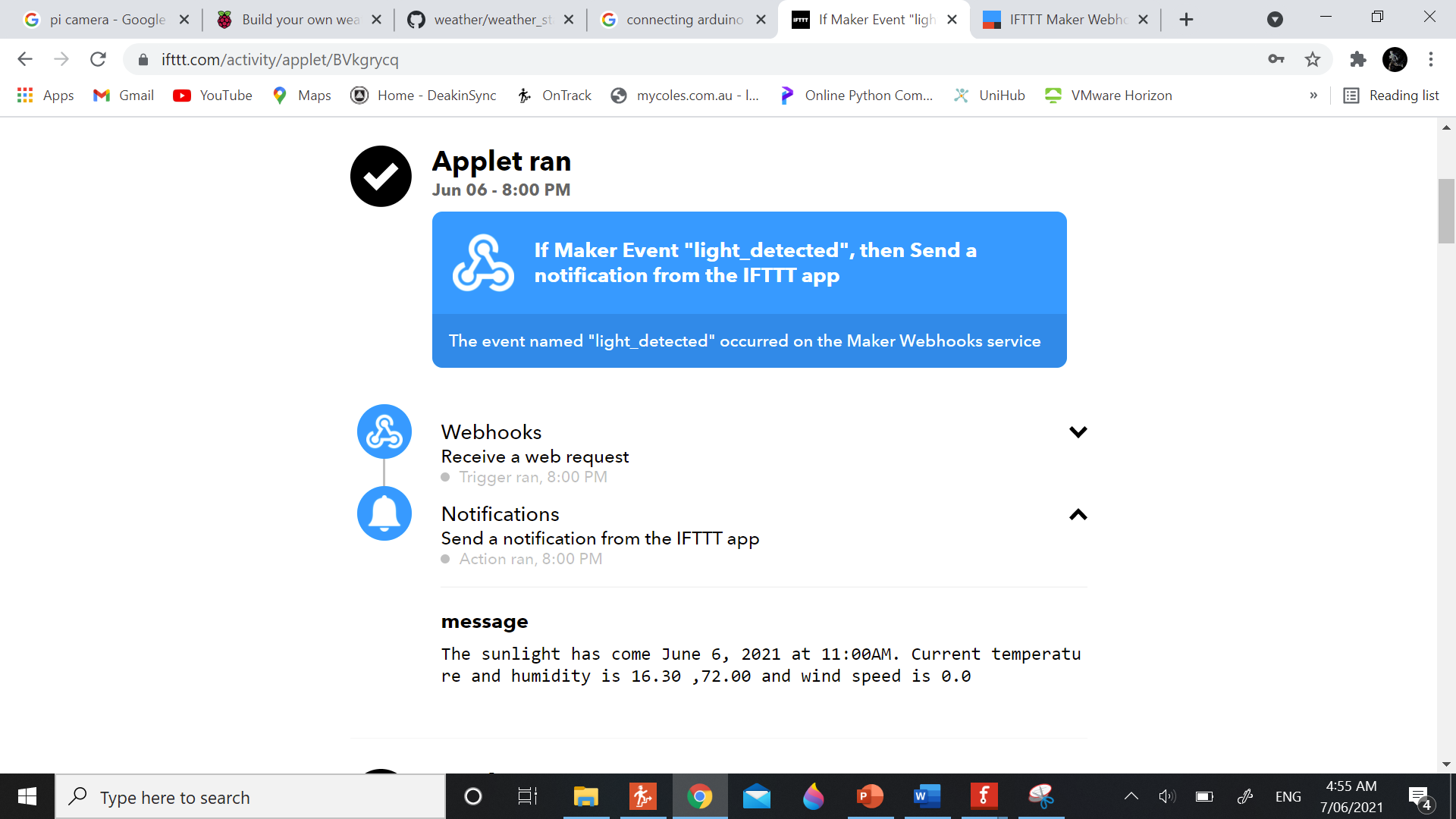


Fig 22

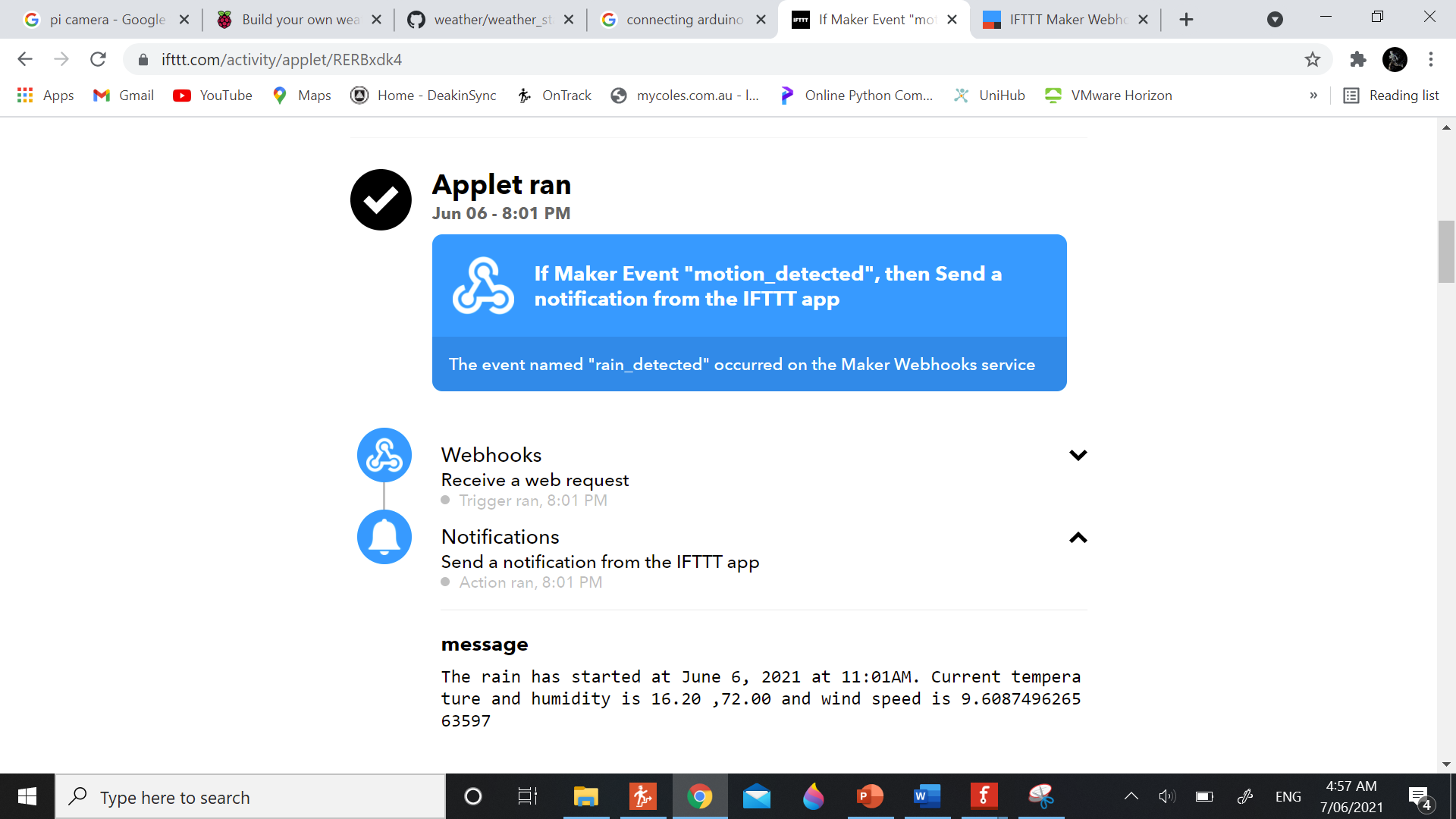


Fig 23

Conclusion:

Weather station was made properly and was tested at different tests.

It sends the notification whenever the light is there or if there is a rain. The project in real can be made in just 200 dollars and would require just a small space in garden.

I really enjoyed working during project. I liked to face some challenges and then solve it. I came to know the importance of taking small steps at one time and then combine the code. If I would have been given second chance, I would have made the model more representable and effective. It was really nice.