- **Approach 1:** Gathering the data and sending the results to: **Google Spreadsheet on your Google Drive**.

A core part of the "**Internet of Things**" movement is the idea of devices that gather data and send it to the Internet. That data is then acted on or observed for later. It's a simple concept and has been going on for a while but lately it's been getting cheaper and easier to do.

Google Spreadsheets

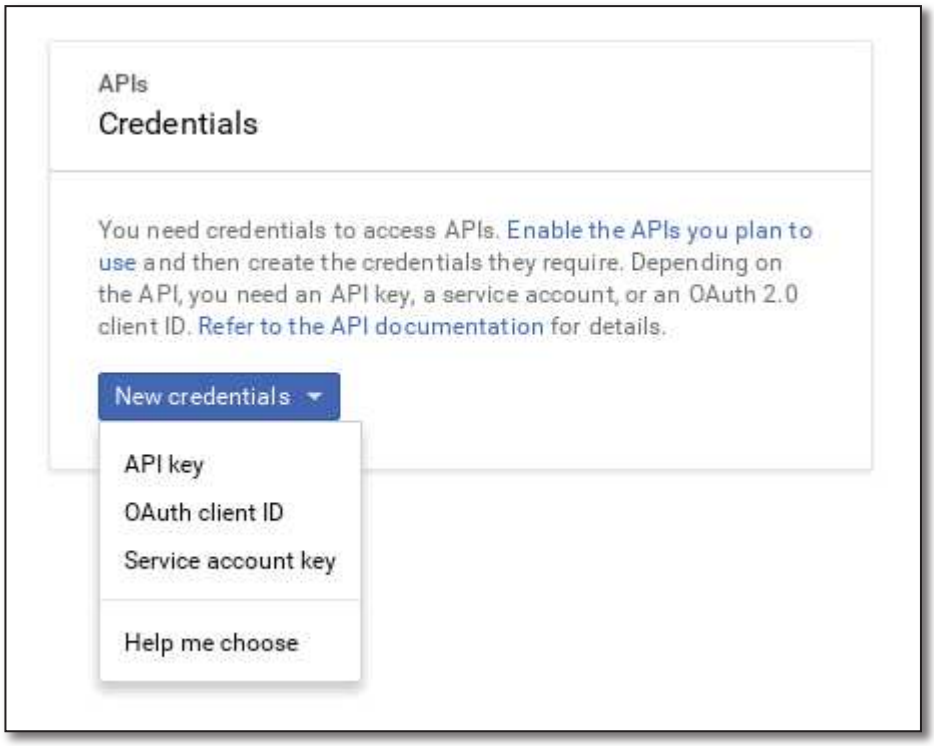
You can output data to a Google Spreadsheet application. You will need to setup OAuth with Google, and create a JSON file. The steps are as mentioned below:

Using OAuth2 for Authorization (OAuth Credentials)

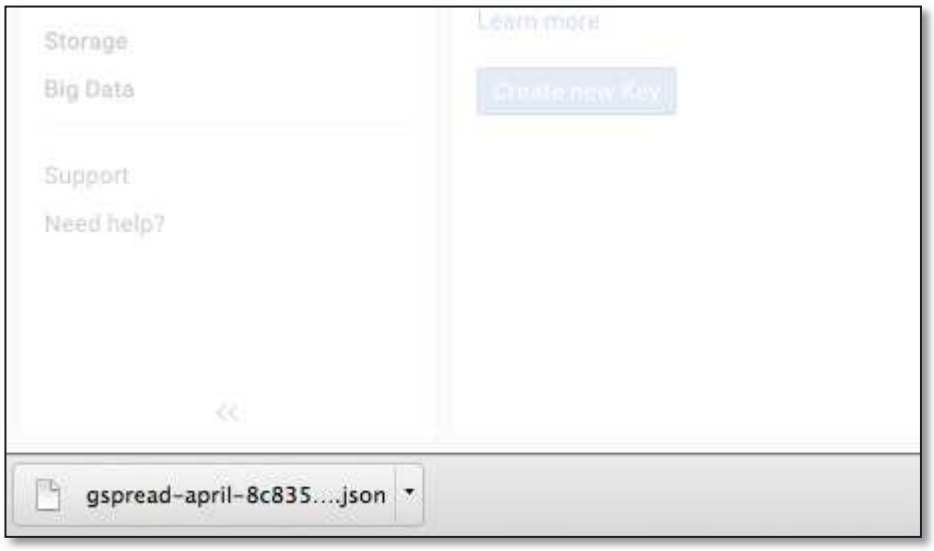
1. Head to Google Developers Console and create a new project (or select the one you have.)

2. Under "API & auth", in the API enable "Drive API".

3. Go to "Credentials" and choose "New Credentials > Service Account Key".



4. You will automatically download a JSON file with this data.



5. This is how this file may look like:

```
{
    "private_key_id": "2cd … ba4",
    "private_key": "-----BEGIN PRIVATE KEY-----\nNrDyLw … jINQh/9\n-----END PRIVATE
KEY-----\n",
    "client_email": "473 … hd@developer.gserviceaccount.com",
    "client_id": "473 … hd.apps.googleusercontent.com",
    "type": "service_account"
}
```

You'll need *client_email* and *private_key*.

6. Install oauth2client:

```
pip install --upgrade oauth2client
```

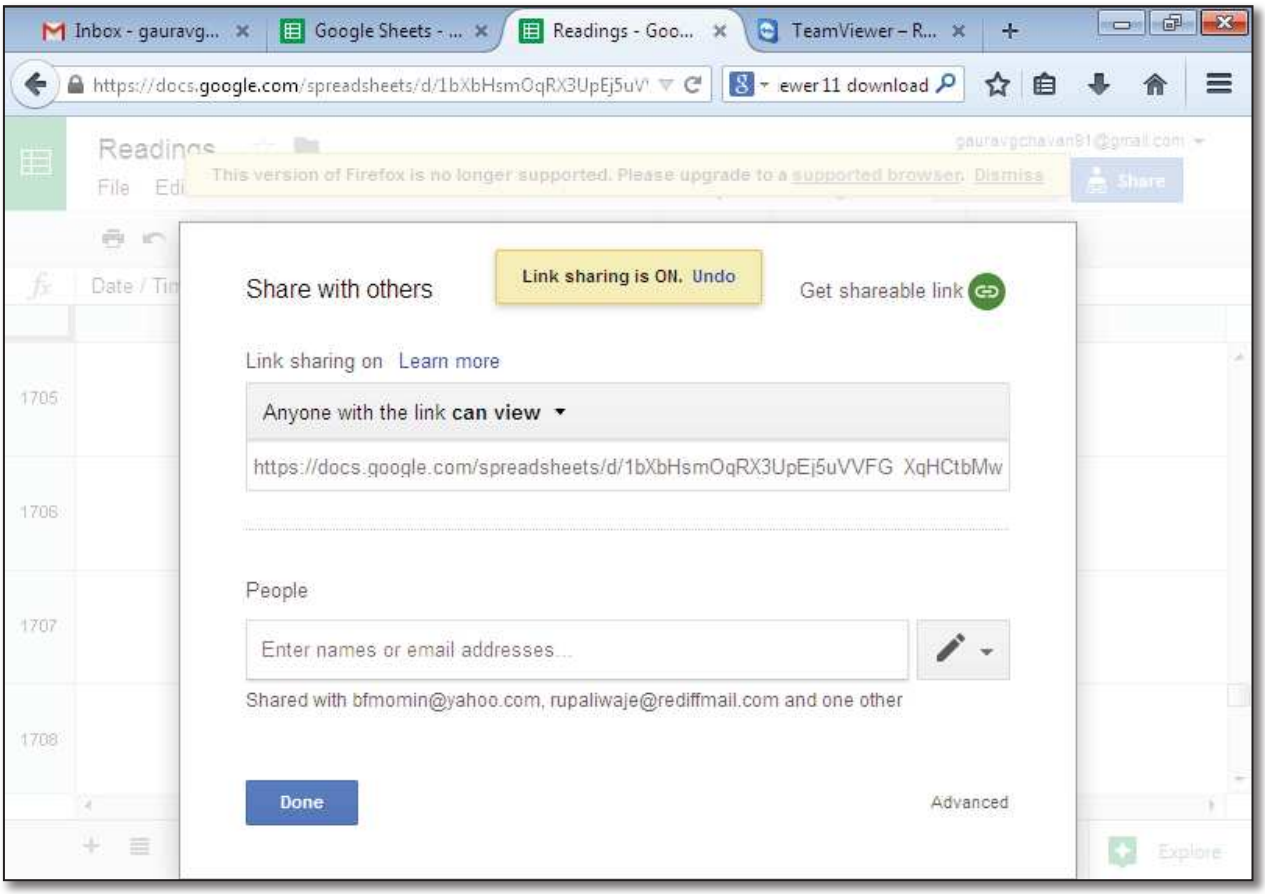
7. Depending on your system setup you may need to install PyOpenSSL:

```
pip install PyOpenSSL
```

You will want to store the generated **JSON** file in the **MiniWeatherStation.py** folder. One thing you will need to is open up that **OAuth JSON** file and look for "**client_email**". It should look like this:

"client_email": "1985453359310-asdlkjried8ss98eeEic@developer.gserviceaccount.com",

Take note of that email address value and go to your Google spreadsheet in a web browser. Using the **File -> Share...** menu item share the spreadsheet with **read access** to the email address found above.



8. Next, open up the **WeatherStationmod.py** file and edit:

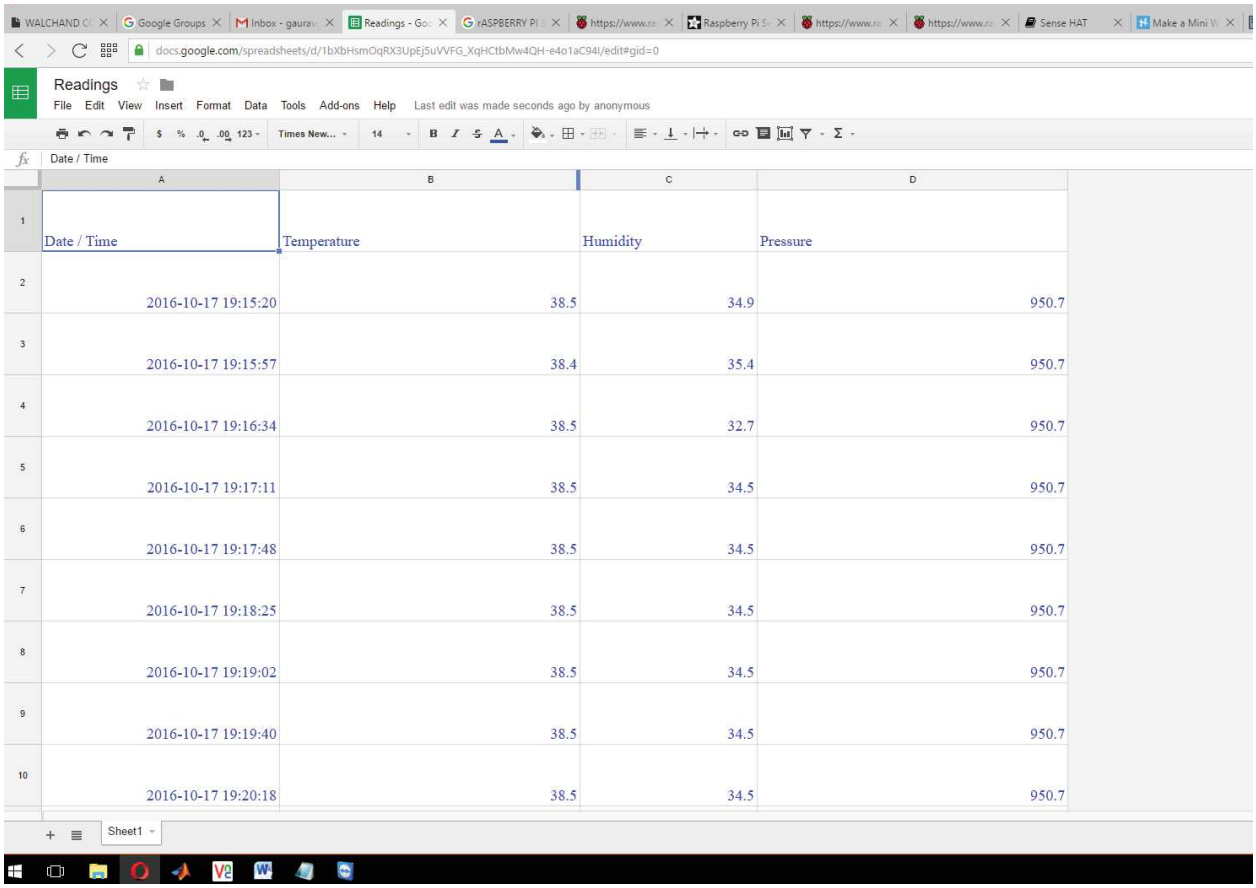
```
sudo nano WeatherStationmod.py
```

Replace the **GDOCS_OOAUTH_JSON** value with the name of your **JSON** file you downloaded. Set the **GDOCS_SPREADSHEET_NAME** with the name of your sheet. Save it.

Type in :

```
sudo python WeatherStationmod.py
```

If all your information is correct, it will start running and adding rows to your spreadsheet every 10 seconds.



Readings

File Edit View Insert Format Data Tools Add-ons Help Last edit was made seconds ago by anonymous

Date / Time

	A	B	C	D
1	Date / Time	Temperature	Humidity	Pressure
2	2016-10-17 19:15:20	38.5	34.9	950.7
3	2016-10-17 19:15:57	38.4	35.4	950.7
4	2016-10-17 19:16:34	38.5	32.7	950.7
5	2016-10-17 19:17:11	38.5	34.5	950.7
6	2016-10-17 19:17:48	38.5	34.5	950.7
7	2016-10-17 19:18:25	38.5	34.5	950.7
8	2016-10-17 19:19:02	38.5	34.5	950.7
9	2016-10-17 19:19:40	38.5	34.5	950.7
10	2016-10-17 19:20:18	38.5	34.5	950.7

Sheet1

Code:

```
#!/usr/bin/python

import json

import sys

import time

import datetime

# libraries

import sys

import urllib2

import json

import gspread

from oauth2client.client  import SignedJwtAssertionCredentials

from sense_hat import SenseHat


# Oauth JSON File

GDOCS_OAUTH_JSON       = 'My Project-452d7668e39b.json'


# Google Docs spreadsheet name.

GDOCS_SPREADSHEET_NAME = 'Readings'


# How long to wait (in seconds) between measurements.

FREQUENCY_SECONDS=5


def login_open_sheet(oauth_key_file, spreadsheet): """Connect to Google Docs spreadsheet and return the first worksheet."""
```

```
try:
json_key = json.load(open(oauth_key_file))
credentials=SignedJwtAssertionCredentials(json_key['client_email'],json_key['private_key'],
['https://spreadsheets.google.com/feeds'])
gc = gspread.authorize(credentials)
worksheet = gc.open(spreadsheet).sheet1
return worksheet
except Exception as ex:
print 'Unable to login and get spreadsheet.  Check OAuth credentials, spreadsheet name, and
make sure spreadsheet is shared to the client_email address in the OAuth .json file!'
print 'Google sheet login failed with error:', ex
sys.exit(1)


sense = SenseHat()
sense.clear()
print 'Logging sensor measurements to {0} every {1}
seconds.'.format(GDOCS_SPREADSHEET_NAME, FREQUENCY_SECONDS)
print 'Press Ctrl-C to quit.'
worksheet = None


while True:
# Login if necessary.
if worksheet is None:
worksheet = login_open_sheet(GDOCS_OAUTH_JSON,
GDOCS_SPREADSHEET_NAME)
```

```
        # Attempt to get sensor reading.
        temp = sense.get_temperature()
        temp = round(temp, 1)
        humidity = sense.get_humidity()
        humidity = round(humidity, 1)
        pressure = sense.get_pressure()
        pressure = round(pressure, 1)

        # 8x8 RGB
        # sense.clear()
        info = 'Temperature (C): ' + str(temp) + 'Humidity: ' + str(humidity) + 'Pressure: ' +
str(pressure)
sense.show_message(info, text_colour=[255, 0, 0])

        # Append the data in the spreadsheet, including a timestamp
        try:
                worksheet.append_row((datetime.datetime.now(), temp,humidity,pressure))
        except:
                # Error appending data, most likely because credentials are stale.
                # Null out the worksheet so a login is performed at the top of the loop.
                print 'Append error, logging in again'
                worksheet = None
                time.sleep(FREQUENCY_SECONDS)
                continue
        # Wait 30 seconds before continuing
        print 'Wrote a row to {0}'.format(GDOCS_SPREADSHEET_NAME)
        time.sleep(FREQUENCY_SECONDS)
```