

Libraries needed:

senseHat() - for sensing hardware readings

requests() - http module to allow send or receive data.

Web Platform:

dweet.io, freeboard.io

Description:

dweet.io is simple publishing and subscribing for machines, sensors, devices, robots, and gadgets (we just call them things). We call published messages 'dweets'. dweet.io enables your machine and sensor data to become easily accessible through a web based RESTful API, allowing you to quickly make apps or simply share data. dweet is free to use. If you want to reserve a thing name and keep your dweets private, it's \$1.99 per lock per month.

To send the sensed data, we just append the sensed data to url passed using **request.post()** method. Example,

```
url='https://dweet.io/dweet/for/http1991?' + 'temp='+ a + '&humidity=' + humidity + '&pressure=' + pressure  
r=requests.post(url)
```

To dweet from your thing, simply call a URL like:

<https://dweet.io/dweet/for/my-thing-name?>

where, we Just replace my-thing-name with a unique name.

Source Code:

```
from sense_hat import SenseHat
import time
import requests
import subprocess

sense = SenseHat()

def Readings():

    ##### Get CPU Temperature #####
    cpu_temp = subprocess.check_output("vcgencmd measure_temp", shell=True)
    array = cpu_temp.split("=")
    array2 = array[1].split('"')
    cpu_temp= float(array2[0])
    print ("CPU_TEMPERATURE: %.2f" %cpu_temp)

    ##### Calculate Calibrated Temperature #####
    temp = round(sense.get_temperature(),2)
    temp_calibrated = round(temp - ((cpu_temp - temp)/5.466),2)
    a=str(temp_calibrated)
    print("Temperature: %s %C" % a)

    # print temp_calibrated

    ##### Calculate Humidity #####
    humidity = str(round(sense.get_humidity(),2))
    print("Humidity: %s %rH" % humidity)

    ##### Calculate Pressure #####
    pressure = str(round(sense.get_pressure(),2))
    print("Pressure: %s Millibars" % pressure)

    # append the readings to URL
    url='https://dweet.io/dweet/for/htp1991?' + 'temp='+ a + '&humidity=' + humidity + '&presure=' + pressure
    r=requests.post(url)
    |
while True:

    Readings()
    time.sleep(10)
    print("readongs recorded")
```

Source Code Description:

In the program, we have defined a function which will take readings every 10 seconds. For that purpose, we have used **sleep()** method from time module.

We have calculated cpu temperature, and then calculated original readings using temperature calibration methodology formula as shown below:

```
temp_calibrated = round(temp - ((cpu_temp - temp)/5.466),2)
```

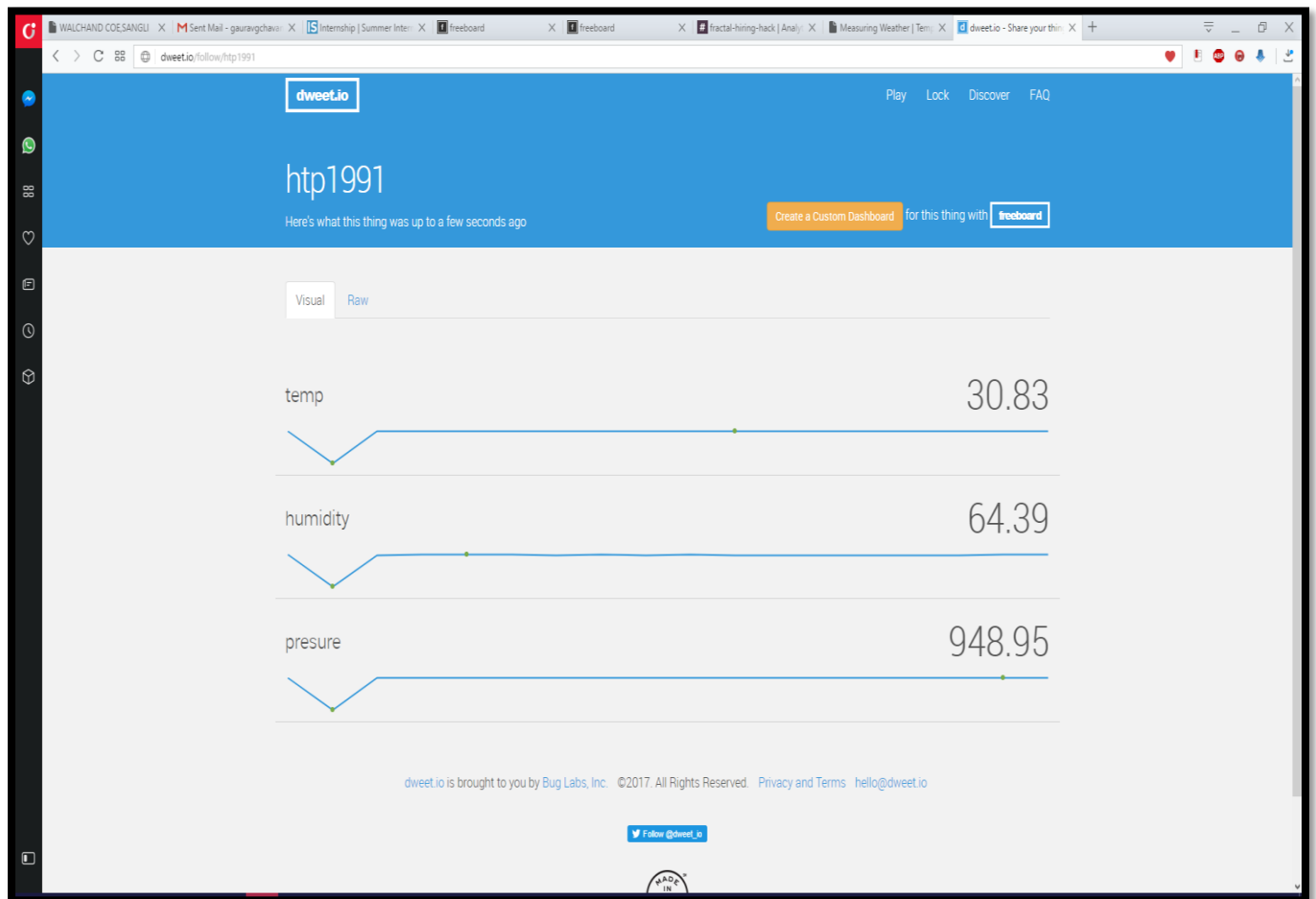
Temperature, Humidity and pressure are computed using `sense.get_temperature()`, `sense.get_humidity()`, `sense.get_pressure()` methods respectively.

The readings are then appended as a string to url.

In the program, we have used `htp1991`, as unique name, as shown in source code. The delay was set to 10 seconds as given in `time.sleep()` method.

To view our dweet.io dashboard, simply type `dweet.io/follow/my-thing` name which in this case is dweet.io/follow/htp1991.

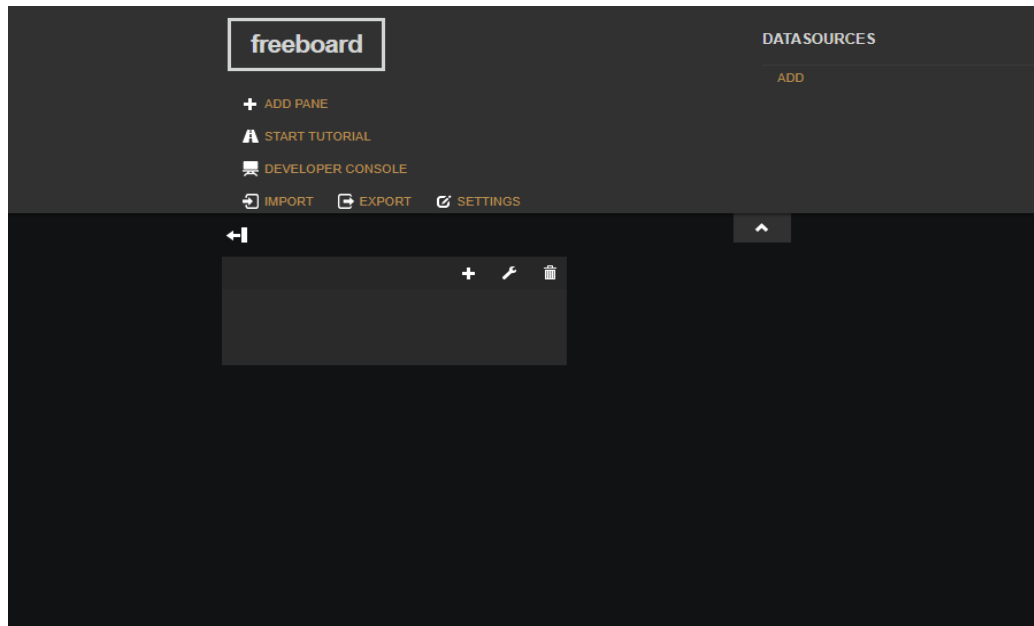
The URL looks like follows:



Dweet.io URL

Now, Freeboard.io is a platform where we can create an interactive dashboard for free. For that purpose, we need to sign-up at freeboard.io and start creating a dashboard.

We need to add pane which is shown as below:



Add pane example

In its settings, we need to select the type of it, for weather, we can set it to gauge, and set the data source type to dweet.io, and thing name to the unique name which is htp1991 in this case as shown below:

A screenshot of the 'DATASOURCE' configuration form in Freeboard.io. The form is titled 'DATASOURCE' and has a subtitle 'A datasource for connecting to things at dweet.io.' It contains several input fields: 'TYPE' with a dropdown menu showing 'Dweet.io', 'NAME' with the value 'SenseHat', 'THING NAME' with the value 'htp1991' and a hint 'Example: salty-dog-1', and 'KEY' which is empty. Below the 'KEY' field is a note: 'If the thing is not locked, you can ignore this field'. At the bottom left is a checkbox for 'SHOW FULL PAYLOAD' which is currently unchecked and labeled 'NO'. At the bottom right are 'SAVE' and 'CANCEL' buttons. A small vertical toolbar is visible on the right edge of the form.

Adding Data Source to dashboard

So in add pane we can select the type of pane, it can be gauge, sparkline, bar etc. Gauge was chosen to display humidity readings as shown below:

The 'WIDGET' configuration panel shows the following settings:

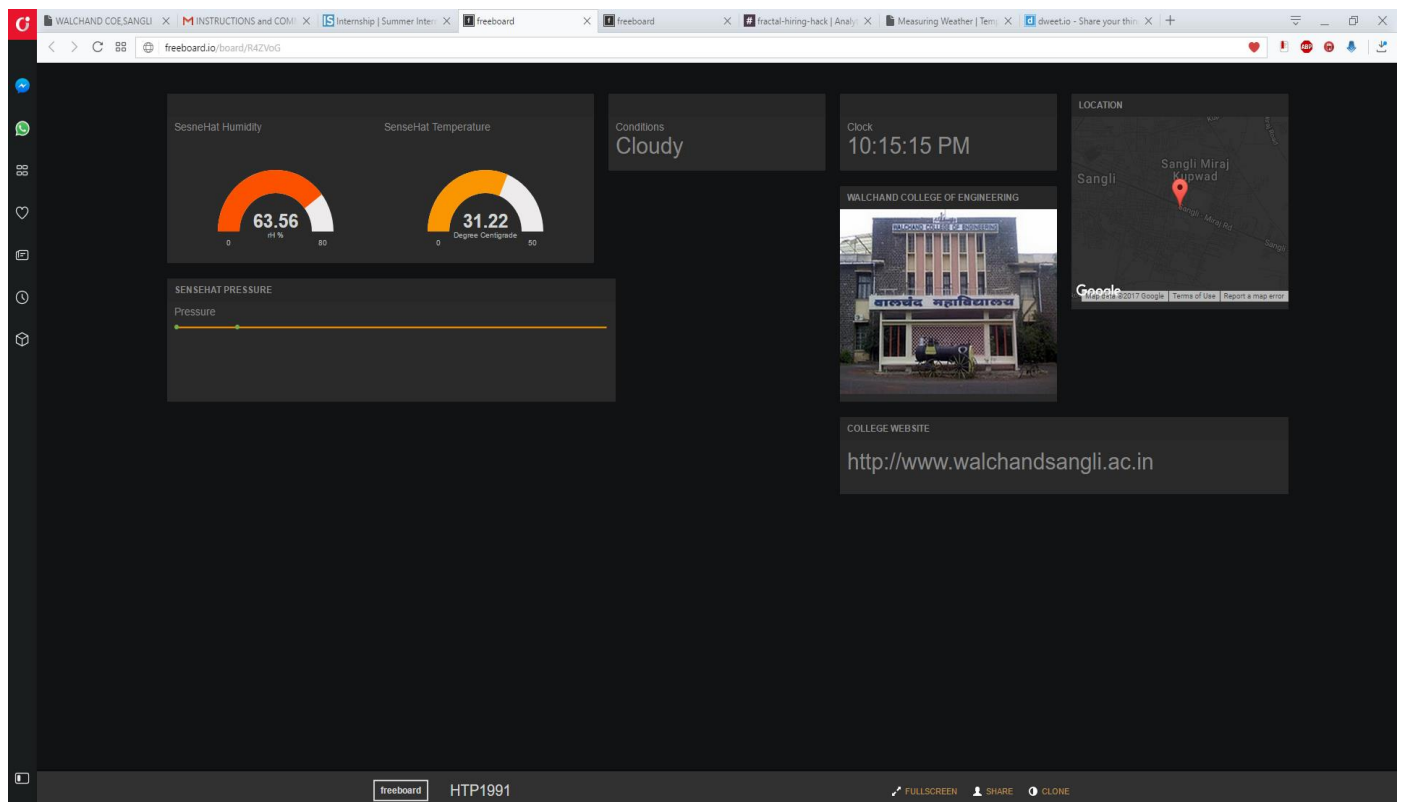
- TYPE: Gauge
- TITLE: Humidity
- VALUE: `datasources["SenseHat"]["humidity"]`
- UNITS: (empty)
- MINIMUM: 10
- MAXIMUM: 85

Buttons: + DATASOURCE, JS EDITOR, SAVE, CANCEL

Setting minimum and maximum values

The minimum and maximum values were set in add pane settings as shown above.

Similarly, the temperature, pressure, location, weather conditons pane were added in a dashboard. As shown below:



Final Dashboard Layout.