**IoT Principles: Assignment 1** Ryan Bannon – x14488478

For the purposes of this assignment, I was required to gather sensor data from any three sensors of my choice, send these readings to the cloud via dweet.io with Python, retrieve the data through JavaScript and display the results on three different charts in a HTML page.

Distance: Using the Grove Pi Ultrasonic Ranger, I developed a getDistance() method in my dweet.py (Python file) that calculates the length in which an object is away from the sensor in centimetres. The result is then sent to my dweet thing named ‘ryan-iot-ca1’. Then the JavaScript in my High Charts HTML file retrieves this value and plots it on a graph. I have chosen a column chart for this reading as I feel it best suits this type of sensor data. I have also set some plotlines on the chart, these are horizontal lines on the chart that give the user a visual of whether there is an object quite close to the sensor or relatively far away from it.

Light Level: Using the Grove Pi Light Sensor, I created a getLight() method also in the dweet.py file. The result of this is then sent to the cloud where the JavaScript can consume the data. This is displayed as a spline chart with a small png image of a light bulb on the markers for each value that is received.

Temperature: Finally, using the Grove Pi DHT sensor, I developed a getTemperature() method in the dweet.py file. This method calculates the temperature of its environment. Similar to the two previous sensor readings, the result from this sensor is transmitted to the cloud, and is then plotted on a chart in the High Charts.html file. I chose to implement an area spline chart for this reading. The reason being, I was able to set zones in the plot options, therefore the area spline values will display as blue if the temperature is below 10 degrees (cold), orange if they are between 10 and 20 (warm), and red if it is above 20 (hot).

Other Information:

* I have created a config.py file that simply contains my thing ID name. This configuration information is read in from the dweet.py file.
* I was unable to store the readings in a local database. I used SQLite3 and managed to create the database, however, inserting the values into the tables proved problematic. I have left the lines of code that left me with errors in the dweet.py file, but have commented them out.
* The sensors used and the above methods submit the information to dweet.io. For visual purposes of the incoming data more so at the beginning of this project, I used this link: <https://dweet.io/follow/ryan-iot-ca1>. This showed me that the data I was gathering was in fact being delivered to my dweet thing. Initially I used fake data to test the system by returning random numbers between 1 and 100, before integrating the sensors script.
* Using a JavaScript while loop, I ensured that the latest 100 readings are stored in HTML Local Storage.

Before this assignment, I had never used a Raspberry Pi for a project. As a result, I have learned a lot about using the Pi, sensors, and transmitting this data to the cloud.