

Raymond Martin
Austin Wells
Brandon Guerra

Python Code:

```
import serial, time, random
from matplotlib import pyplot as plt
import numpy as np
```

```
class serialPort():
    def __init__(self, port):
        self.ser = serial.Serial(port, 57600, timeout=1)
        self.ser.flush()

    def write(self, payload):
        self.ser.write(payload + '\r\n')
        time.sleep(1)

    def read(self):
        payload = self.ser.read()
        return ord(payload)

    def movingTimeSeries(self, size):
        self.tsData = [-1 for i in range(size)]
        self.ax = plt.axes(xlim=(0, size), ylim=(0, 255))
        self.line, = plt.plot(self.tsData)
        plt.ion() # interactive plots can do not block on "show"
        plt.show() # show the plot on the screen

    def updateTS(self, point):
        self.tsData.insert(0, point)
        self.tsData.pop()
        self.line.set_ydata(self.tsData) # set the data
        plt.draw() # and draw it out
        #time.sleep(0.1) # simulate some down time
        plt.pause(0.0001) # pause so that the drawing updates

    def graph(self, x, y):
        #sample x and y in numpy
        # 100 equally spaced array of numbers from 0.0 to 1
        #x = np.linspace(0, 1, 100)
        # take sine of the value
        #y = np.sin(2 * 3.14159 * t)
        plt.plot(x, y) # plot them
        plt.show()

    def close(self):
        self.ser.close()
```

```

if __name__ == '__main__':
    #example of polling from a serial port
    ser = serialPort('/dev/tty.usbserial-A50271RK')
    ser.movingTimeSeries(300)

    while 1:
        out = ser.read()
        #print(out)
        ser.updateTS(out)

```

Arduino Code:

```

byte i;

// the setup routine runs once when you press reset:
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(57600);
}

// the loop routine runs over and over again forever:
void loop() {
    int val = analogRead(A0);
    byte val_byte = map(val, 0, 1023, 0, 255);
    //Serial.print(val_byte);
    Serial.write(val_byte);

    delay(100);
    // print out the value:
    /*Serial.write(i);
    i += 1;

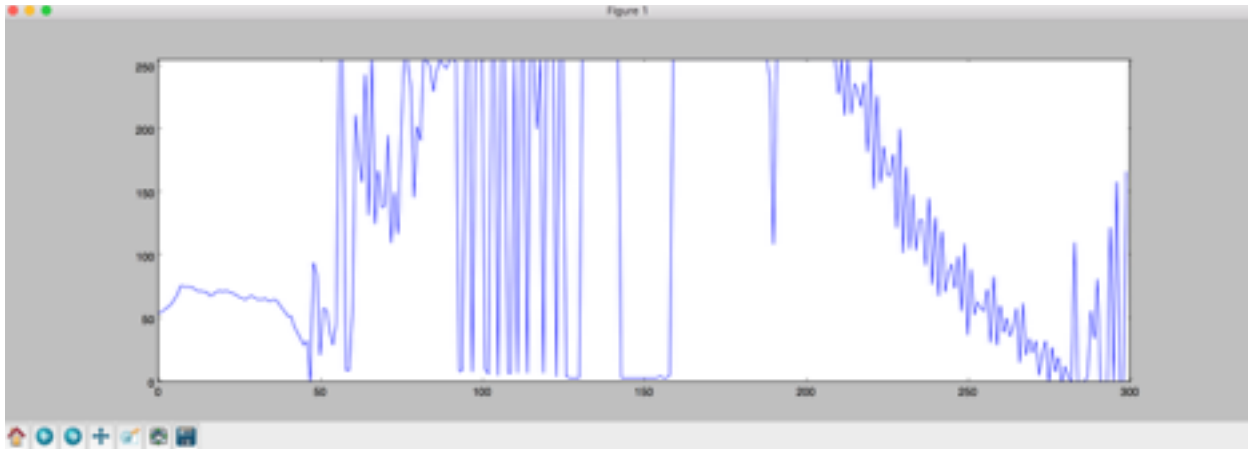
    if(i>=255)
        i=0;

    delay(10);    // delay in between reads for stability
    */
}

```

Screen Shot:

Part Three:



Yes, it looks different. People have different moisture levels and may effect the voltage with it.

For Thought:

1. Map the value to 16 bits, and then read 16 bits in python instead of 8 bits. ex: `serial.read(2)` (2 is for 2 bytes = 16 bits)
2. We could read in more than one value at a time to decrease the amount we update the plot. We could also not print values which are no required.
3. We are sending data at less than 57600 baud rate, because there is overhead of the plotting updating and there functions consuming computer resources.