Assignment 6 report

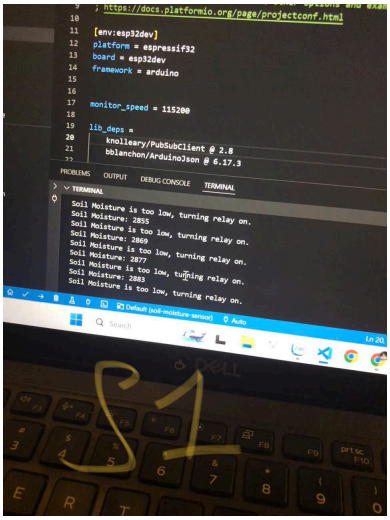
In this assignment, i gathered soil moisture sensor readings, measured as values from 0-4000 (using ESP32- 3.3Volt). To convert these into actual soil moisture readings, i need to calibrate your sensor, taking readings from soil samples, then calculating the gravimetric soil moisture content from these samples.

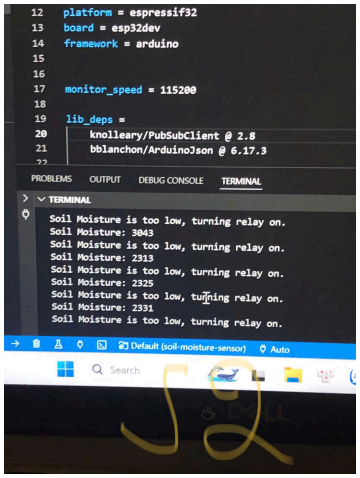
Part 1. Take a soil moisture reading using the soil moisture sensor and write down this reading.

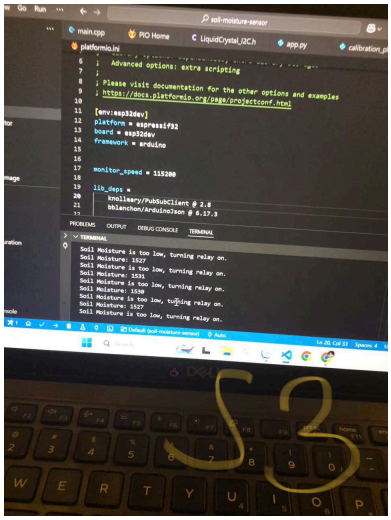
The data collected by 3 sample of soil is measured by kitchen **microscale** and esp32 microprocessor displayed through Laptop monitor and recorded as follow

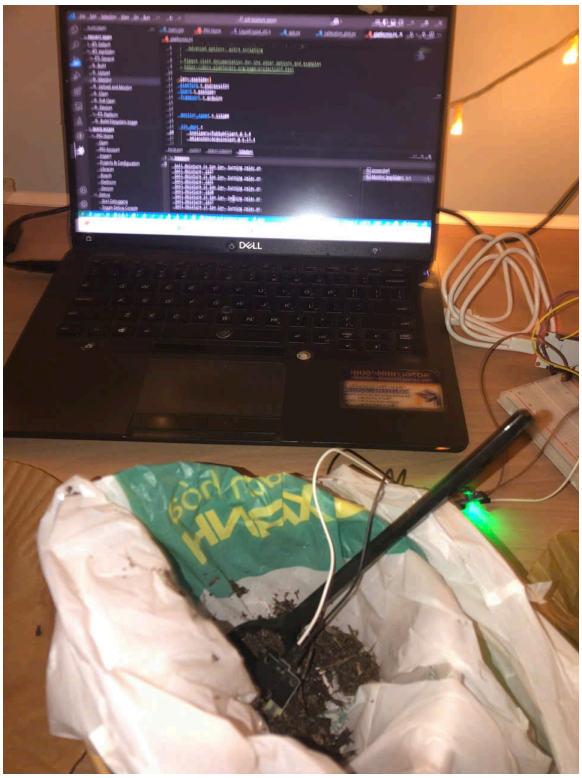
- **Sample 1: 2883 Volt / 11 Grams**
- **Sample 2: 2332 Volt / 13 Grams**
- **Sample 3: 1527 Volt/ 22 Grams**

The actual footages of the Test are given below









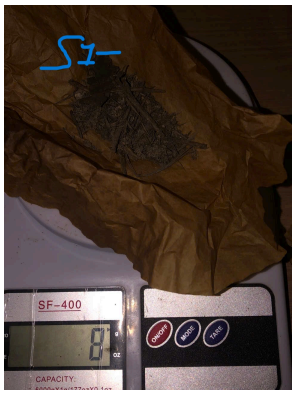
Part 2: Drying in the oven for completely dry the soil sample

The 3 soil sample is then dried in the oven for 3 hours with low heat and uncovered oven (For the moisture to disappear), then weigh again for final calculation.

- **Sample 1: 8 grams**
- **Sample 2: 8 grams**
- **Sample 3: 12 grams**

The result is depicted below for 3 sample:

*



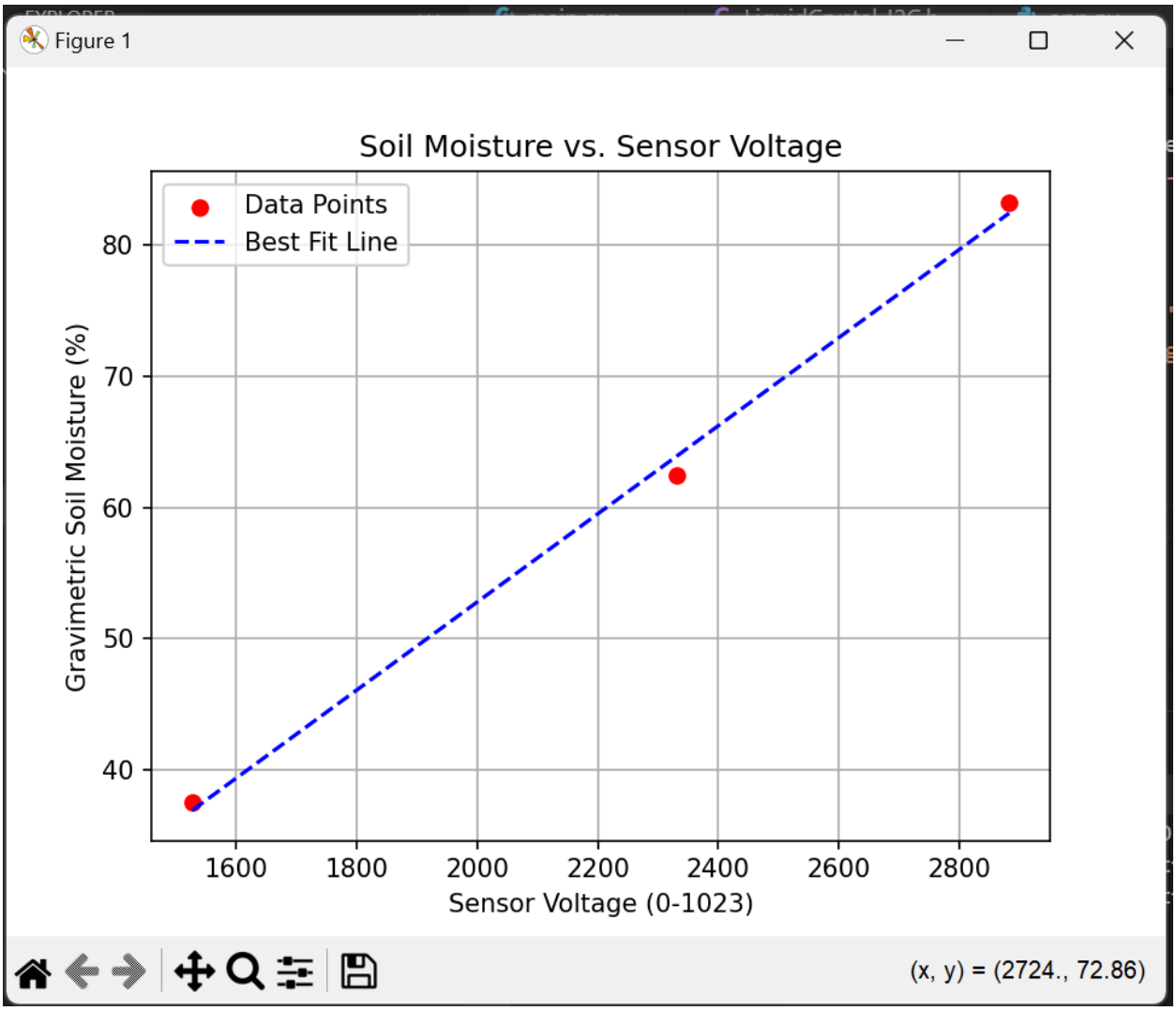
Part 3: gravimetric soil moisture calculation

$$\text{Soil moisture \%} = \frac{W_{wet} - W_{dry}}{W_{dry}} \times 100$$

Statistic table

Sample	1	2	3
Weight with moisture (gram)	11	13	22
Weight without moisture (gram)	8	8	12

Soil moisture calculated (%)	**37.5**	**62.5**	**83,3**



#Code for connecting Moisture sensor with Arduino (include the output of LED1606 using LiquidCrystal_I2C.h)

```
#include <Arduino.h>

#define M_PIN A1      // Moisture sensor pin
#define RELAY_PIN 4   // Relay connected to D4

void setup() {
   Serial.begin(9600);
   pinMode(M_PIN, INPUT);
   pinMode(RELAY_PIN, OUTPUT);
   digitalWrite(RELAY_PIN, LOW);  // Start with relay OFF
}

void loop() {
   int moisture = analogRead(M_PIN); // Read moisture sensor

   // Check moisture level and control relay
   if (moisture > 450) {
      Serial.println("Moisture level too low, turning relay ON");
      digitalWrite(RELAY_PIN, HIGH);  // Turn relay ON
   } else {
      Serial.println("Moisture level sufficient, turning relay OFF");
      digitalWrite(RELAY_PIN, LOW);  // Turn relay OFF
   }

   // Print moisture value to Serial Monitor
   Serial.print("Moisture Level: ");
   Serial.println(moisture);

   delay(1000);  // Wait 1 second before next reading
}
```

#Code of Jupiter notebook used for calibration graph

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Sample data: Voltage (sensor readings) and Gravimetric Soil Moisture (%)
voltage_values = [2883, 2331, 1527]  # Example voltage values
moisture_values = [83.3, 62.5, 37.5]  # Example moisture percentages

# 22 - 12 = 10 / 12 = 0.833  | 13 - 8 = 5 / 8 = 62,5 %  | 11 - 8 = 3 / 8 = 37,5 %

# Perform linear regression to find the best-fit line
slope, intercept, r_value, p_value, std_err = linregress(voltage_values, moisture_va
print(f"Equation: y = {slope:.4f}x + {intercept:.4f}")
# Generate x values for the line
x_values = np.linspace(min(voltage_values), max(voltage_values), 100)
y_values = slope * x_values + intercept

# Plot the data points
plt.scatter(voltage_values, moisture_values, color='red', label='Data Points')
plt.plot(x_values, y_values, linestyle='--', color='blue', label='Best Fit Line')

# Labels and title
plt.xlabel('Sensor Voltage (0-1023)')
plt.ylabel('Gravimetric Soil Moisture (%)')
plt.title('Soil Moisture vs. Sensor Voltage')
plt.legend()
plt.grid()

# Show the plot
plt.show()
```