UNIVERSIDAD
Panamericana

Homework 2: Microprocessors

Student: Sergio Andrés Becerra Gándara 0244963

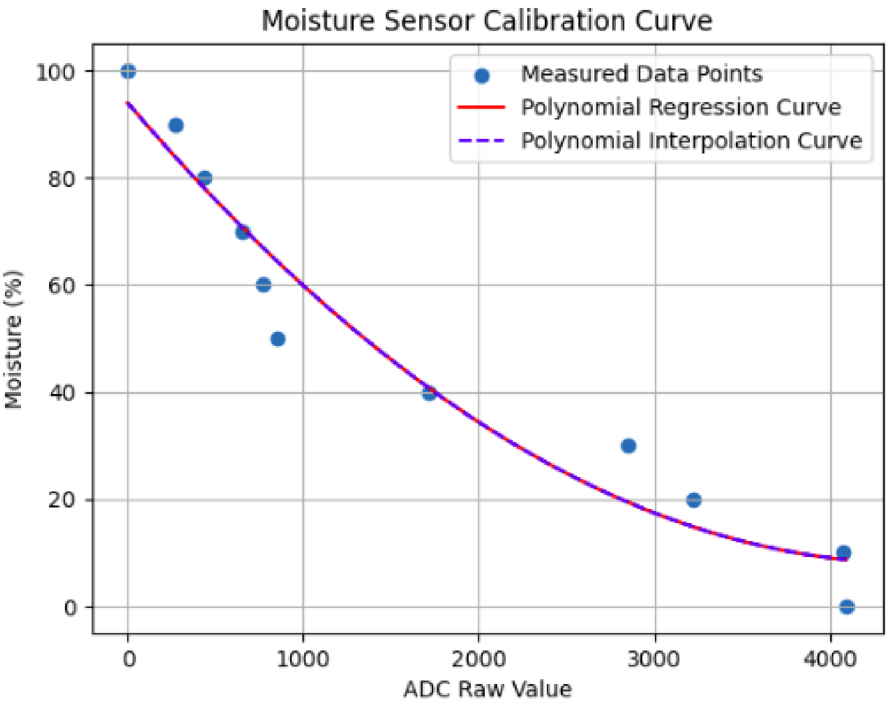
Professor: Dr. Luis Emilio Tonix Gleason

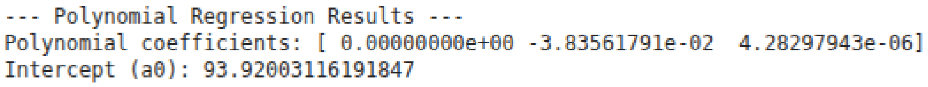
September 2025

1. Setup and calibration procedure

For this practice we used a glass flask, cables, two nails, an ESP32, water and dirt. We made a cheap homemade humidity sensor connected to the ADC from the ESP32. In my case I used a voltage divisor because in the normal connection the ADC values were not accurate and had a lot of noise. Due to this voltage divisor my result values were upside down, this meaning that when the dirt is completely dry the ADC has a value of 3.3V and when the nails are "touching"the value of the ADC is 0V. I did the calibration starting with dry dirt and then adding 5ml of water and measuring. I had only 50g of dirt so 5ml of water sounded reasonable for every measurement. Then I added this measurements to a .csv file and exported to a Python script to obtain the following graph:

1.1. Calibration curve





```
--- Polynomial Regression Results ---
Polynomial coefficients: [ 0.00000000e+00 -3.83561791e-02  4.28297943e-06]
Intercept (a0): 93.92003116191847
```

As we can see the behavior of this data is polynomial and we obtained the coefficients in order to use them in our calibration function in the ADC code.

The behavior of the sensor is polynomial which is logical because the dirt absorbs the water and when it is saturated or filled with a lot of water the data goes up in a non linear way because of the saturation and we obtain this behavior.

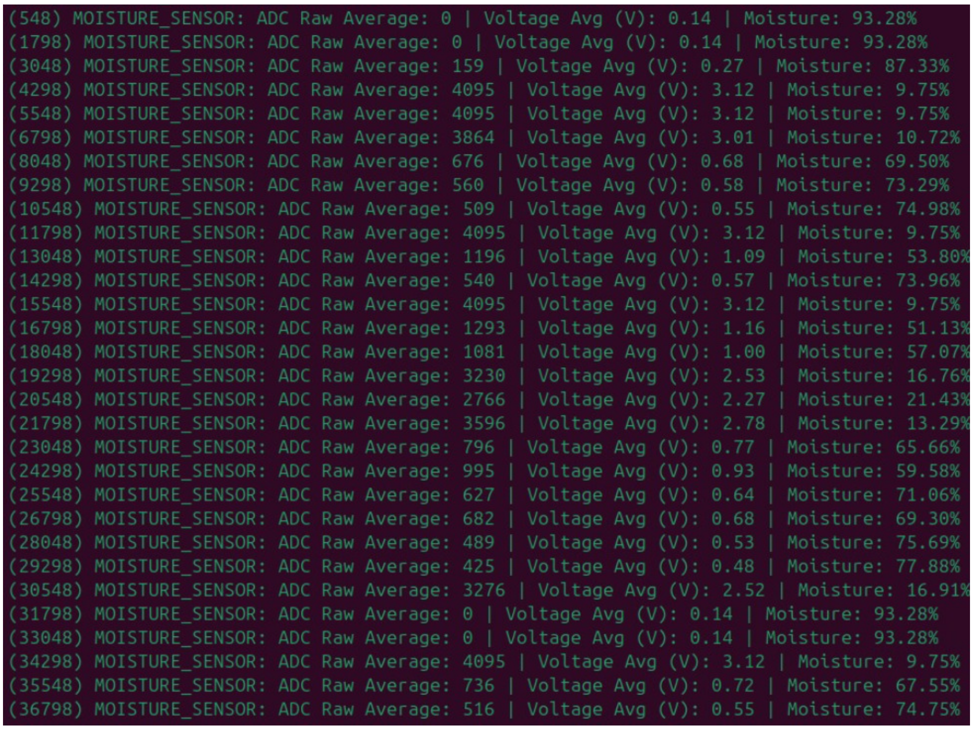
1.2. Calibration function in terms of moisture %

```c
float moisture_map(float voltage_value)
{
    // Polinomial regression coefficients
    const float a0 = 93.92003116191847;
    const float a1 = 3.83561791e-02;
    const float a2 = 4.28297943e-06;

    // The formula for the second-degree polynomial regression: y = a0 + a1*x + a2*x^2
    // where y is the moisture percentage and x is the voltage.
    float moisture_percentage = a0 + a1 * voltage_value + a2 * (voltage_value * voltage_value);

    // Ensure the percentage is within the 0-100% range
    if (moisture_percentage < 0) {
        moisture_percentage = 0;
    }
    if (moisture_percentage > 100) {
        moisture_percentage = 100;
    }

    return moisture_percentage;
}
```

Listing 1: Calibration function code

We added this function to the ADC one shot code so we can take the readings and transform them into a moisture percentage output.

2. Data obtained



```
(548) MOISTURE_SENSOR: ADC Raw Average: 0 | Voltage Avg (V): 0.14 | Moisture: 93.28%
(1798) MOISTURE_SENSOR: ADC Raw Average: 0 | Voltage Avg (V): 0.14 | Moisture: 93.28%
(3048) MOISTURE_SENSOR: ADC Raw Average: 159 | Voltage Avg (V): 0.27 | Moisture: 87.33%
(4298) MOISTURE_SENSOR: ADC Raw Average: 4095 | Voltage Avg (V): 3.12 | Moisture: 9.75%
(5548) MOISTURE_SENSOR: ADC Raw Average: 4095 | Voltage Avg (V): 3.12 | Moisture: 9.75%
(6798) MOISTURE_SENSOR: ADC Raw Average: 3864 | Voltage Avg (V): 3.01 | Moisture: 10.72%
(8048) MOISTURE_SENSOR: ADC Raw Average: 676 | Voltage Avg (V): 0.68 | Moisture: 69.50%
(9298) MOISTURE_SENSOR: ADC Raw Average: 560 | Voltage Avg (V): 0.58 | Moisture: 73.29%
(10548) MOISTURE_SENSOR: ADC Raw Average: 509 | Voltage Avg (V): 0.55 | Moisture: 74.98%
(11798) MOISTURE_SENSOR: ADC Raw Average: 4095 | Voltage Avg (V): 3.12 | Moisture: 9.75%
(13048) MOISTURE_SENSOR: ADC Raw Average: 1196 | Voltage Avg (V): 1.09 | Moisture: 53.80%
(14298) MOISTURE_SENSOR: ADC Raw Average: 540 | Voltage Avg (V): 0.57 | Moisture: 73.96%
(15548) MOISTURE_SENSOR: ADC Raw Average: 4095 | Voltage Avg (V): 3.12 | Moisture: 9.75%
(16798) MOISTURE_SENSOR: ADC Raw Average: 1293 | Voltage Avg (V): 1.16 | Moisture: 51.13%
(18048) MOISTURE_SENSOR: ADC Raw Average: 1081 | Voltage Avg (V): 1.00 | Moisture: 57.07%
(19298) MOISTURE_SENSOR: ADC Raw Average: 3230 | Voltage Avg (V): 2.53 | Moisture: 16.76%
(20548) MOISTURE_SENSOR: ADC Raw Average: 2766 | Voltage Avg (V): 2.27 | Moisture: 21.43%
(21798) MOISTURE_SENSOR: ADC Raw Average: 3596 | Voltage Avg (V): 2.78 | Moisture: 13.29%
(23048) MOISTURE_SENSOR: ADC Raw Average: 796 | Voltage Avg (V): 0.77 | Moisture: 65.66%
(24298) MOISTURE_SENSOR: ADC Raw Average: 995 | Voltage Avg (V): 0.93 | Moisture: 59.58%
(25548) MOISTURE_SENSOR: ADC Raw Average: 627 | Voltage Avg (V): 0.64 | Moisture: 71.06%
(26798) MOISTURE_SENSOR: ADC Raw Average: 682 | Voltage Avg (V): 0.68 | Moisture: 69.30%
(28048) MOISTURE_SENSOR: ADC Raw Average: 489 | Voltage Avg (V): 0.53 | Moisture: 75.69%
(29298) MOISTURE_SENSOR: ADC Raw Average: 425 | Voltage Avg (V): 0.48 | Moisture: 77.88%
(30548) MOISTURE_SENSOR: ADC Raw Average: 3276 | Voltage Avg (V): 2.52 | Moisture: 16.91%
(31798) MOISTURE_SENSOR: ADC Raw Average: 0 | Voltage Avg (V): 0.14 | Moisture: 93.28%
(33048) MOISTURE_SENSOR: ADC Raw Average: 0 | Voltage Avg (V): 0.14 | Moisture: 93.28%
(34298) MOISTURE_SENSOR: ADC Raw Average: 4095 | Voltage Avg (V): 3.12 | Moisture: 9.75%
(35548) MOISTURE_SENSOR: ADC Raw Average: 736 | Voltage Avg (V): 0.72 | Moisture: 67.55%
(36798) MOISTURE_SENSOR: ADC Raw Average: 516 | Voltage Avg (V): 0.55 | Moisture: 74.75%
```

As we can observe the data worked as said, that 3.3V means the dirt is dry and 0V means the dirt is totally wet. We realize that there is some error around 7 to 9 % because when the ADC Raw Value is 0 it has a little bit of voltage which makes the error visible and the same when the Raw Value is 4095 because ir never hits the 3.3V mark. Still i think those are pretty good results and the sensor works in a very god way.