

From bits to real values: converting sensor readings



Online guide,
always up to date

From the number the converter returns to degrees, volts and percent moisture. With a soil sensor calibrated on two points and an analog temperature sensor, both read with the ADS1115.

By **Ada Intelisano** · informaticait.it technical team

IN SHORT

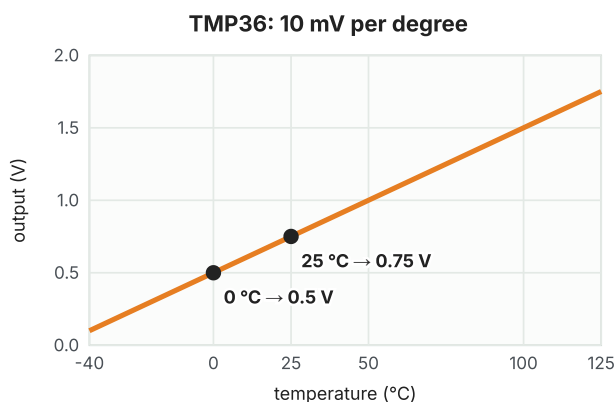
- Every conversion takes two steps: **from the number to volts** (depends on the converter) and **from volts to the quantity** (depends on the sensor).
- If the datasheet gives a formula, use it: the **TMP36** gives 0.5 V at 0 °C and 10 mV per degree.
- If there is no formula, you **calibrate on two points**: read the soil sensor in the air (0 %) and in water (100 %), then draw the straight line.
- The conversion does not make the sensor more accurate: 16 bits on a ± 3 °C TMP36 are still ± 3 °C.

Step 1: from the number to volts

The converter returns an integer. To get volts you multiply it by the value of one step, that is the full scale divided by the number of levels:

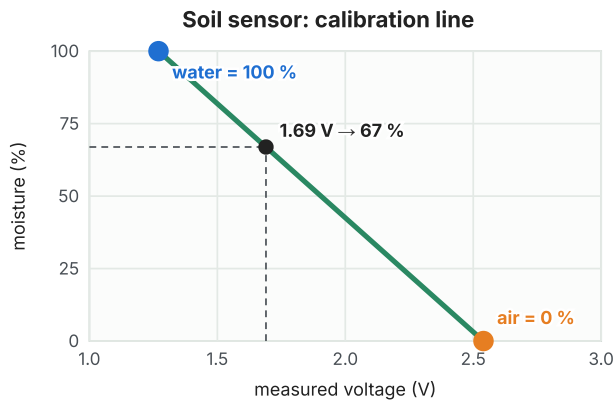
Converter	One step is worth	Volts =
Arduino UNO, 5 V reference	$5 \div 1024 \approx 4.9 \text{ mV}$	$\text{reading} \times 5 \div 1024$
ESP32 (Arduino)	the chip works it out with its calibration	$\text{analogReadMilliVolts}(\text{pin}) \div 1000$
ADS1115, GAIN_ONE ($\pm 4.096 \text{ V}$)	$4.096 \div 32768 = 0.125 \text{ mV}$	$\text{ads.computeVolts}(\text{reading})$

Step 2: from volts to the quantity



With the datasheet formula. The Analog Devices TMP36 gives a voltage proportional to temperature: 500 mV at 0 °C and 10 mV for every degree, so $T = (V - 0.5) \times 100$. At 25 °C the output is 750 mV. It runs on 2.7 to 5.5 V and draws less than 50 μA .

informaticait.it



informaticait.it

With two-point calibration. The DFRobot SEN0193 capacitive sensor gives a voltage that **drops** as moisture rises, but every unit is a bit different and there is no formula. DFRobot explains how to do it: read the sensor **in the air** (dry soil, 0 %) and **in a glass of water** without going past the line (100 %). Between the two points you draw a straight line: $\text{moisture} = (V_{\text{air}} - V) \div (V_{\text{air}} - V_{\text{water}}) \times 100$.

Complete example with the ADS1115

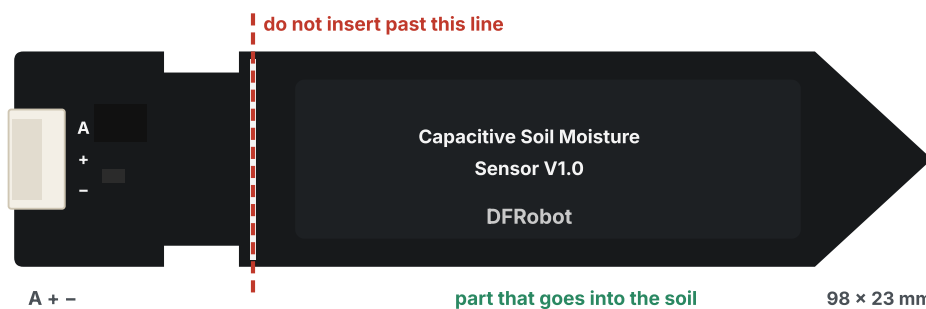
Calibration: 2.54 V in the air, 1.27 V in water (these are the values from the DFRobot example, 520 and 260 on the Arduino UNO).

Channel A0 reads **13520** → $13520 \times 0.125 \text{ mV} = \mathbf{1.690 \text{ V}}$ → $\text{moisture} = (2.54 - 1.69) \div (2.54 - 1.27) \times 100 = \mathbf{67 \%}$.

Channel A1 (TMP36) reads **5960** → $5960 \times 0.125 \text{ mV} = \mathbf{0.745 \text{ V}}$ → $T = (0.745 - 0.5) \times 100 = \mathbf{24.5 ^\circ C}$.

The capacitive soil sensor

DFRobot SEN0193

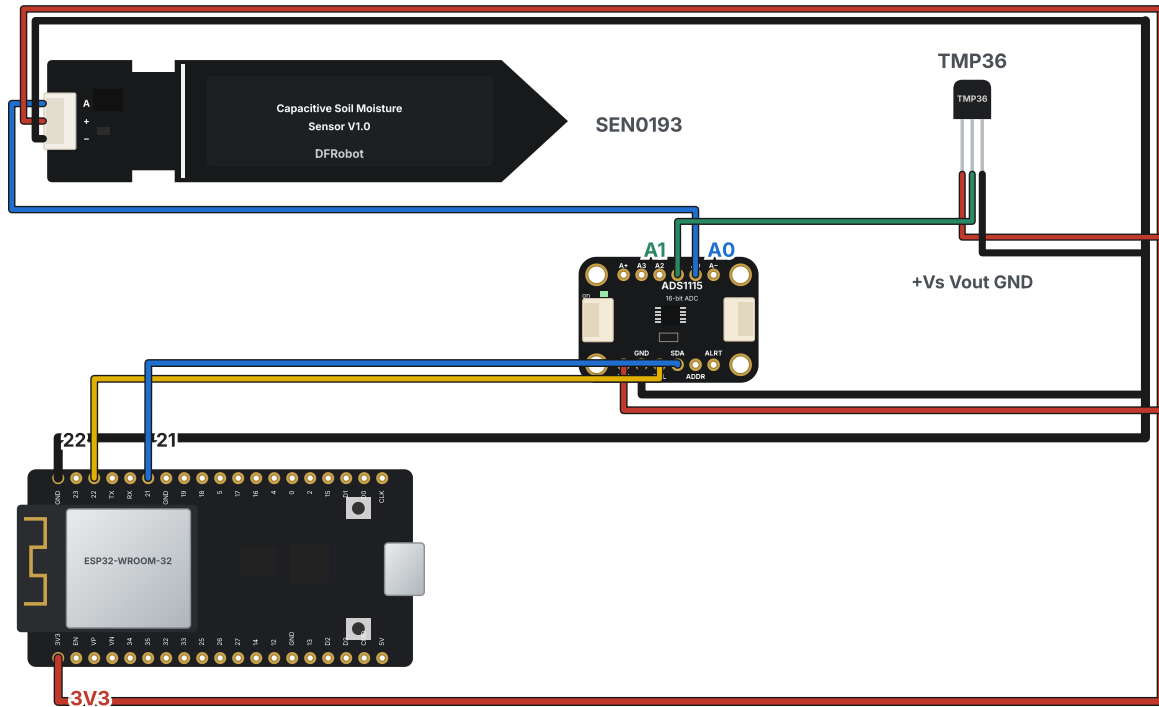


informaticait.it

DFRobot SEN0193: 98 × 23 mm, 3.3 to 5.5 V supply (it has its own regulator), 3-pin connector A, +, -. The components on the upper part are **not waterproof**: push it into the soil only up to the white line. DFRobot warns that the reading also changes with depth and with how compact the soil is: the calibration holds for the way you install it.

Wiring with the ESP32 and the ADS1115

ESP32 + ADS1115: soil sensor on A0, TMP36 on A1



informaticait.it

Everything at 3.3 V: the ADS1115, the soil sensor and the TMP36 take their power from the ESP32's 3V3. The soil sensor goes to input **A0**, the middle output of the TMP36 to **A1**. With the Arduino UNO you use 5 V and the same sketch.

The sketch

```
// From bits to real values with the ADS1115:
// A0 = capacitive soil moisture sensor (DFRobot SEN0193): two-point calibration, air and water
// A1 = TMP36 analog temperature sensor: 10 mV per degree, 500 mV at 0 degrees
// Same sketch for Arduino UNO and ESP32. Library: Adafruit ADS1X15.
#include <Adafruit_ADS1X15.h>

Adafruit_ADS1115 ads;

// Calibration: write YOUR two readings in volts here (see the guide, step 2).
// These are the values from the DFRobot example (520 and 260 on Arduino UNO, that is 2.54 V and 1.27 V):
// every sensor is a little different, so measure yours.
const float V_AIR = 2.54;    // dry sensor, in the air
const float V_WATER = 1.27; // sensor in a glass of water, not past the line

void setup() {
  Serial.begin(115200);
  if (!ads.begin(0x48)) {
    Serial.println("ADS1115 not found");
    while (true) delay(1000);
  }
  ads.setGain(GAIN_ONE);    // +/-4.096 V full scale, 1 bit = 0.125 mV
}

void loop() {
  // 1) bits -> volts
  float vSoil = ads.computeVolts(ads.readADC_SingleEnded(0));
  float vTmp36 = ads.computeVolts(ads.readADC_SingleEnded(1));

  // 2) volts -> real value
  // moisture: straight line through the two calibration points (more water = lower voltage)
  float moisture = (V_AIR - vSoil) / (V_AIR - V_WATER) * 100.0;
  moisture = constrain(moisture, 0.0, 100.0);
  // temperature: formula from the TMP36 datasheet
  float tempC = (vTmp36 - 0.5) * 100.0;

  Serial.print("Soil: ");
  Serial.print(vSoil, 3);
```

```

Serial.print(" V = ");
Serial.print(moisture, 0);
Serial.print(" %   TMP36: ");
Serial.print(vTmp36, 3);
Serial.print(" V = ");
Serial.print(tempC, 1);
Serial.println(" degrees C");
delay(1000);
}

```

Calibrate first: leave the sensor in the air and note the voltage printed, then put it in a glass of water up to the line and note the second one. Write the two values in V_AIR and V_WATER and upload the sketch again.

Other sensors, other formulas

Sensor	From volts to quantity	Good to know
TMP36	$T = (V - 0.5) \times 100$	2.7–5.5 V; in the TO-92 package (grade G) the maximum error is $\pm 3^\circ\text{C}$ at 25°C
LM35	$T = V \times 100$ (0 V at 0°C)	needs at least 4 V of supply: with the ESP32 at 3.3 V it is outside its specifications
NTC 10 kΩ, B 3950	resistance from the divider, then the Beta equation: $1/T = 1/T_0 + \ln(R/R_0)/B$ (T in kelvin, $T_0 = 298.15\text{ K}$)	away from $25\text{--}50^\circ\text{C}$ the Beta formula drifts from the manufacturer's table (calculated on the Adafruit 4890 table: about 0.5°C at 0°C , over 2°C at -40°C); "3950" thermistors from different brands do not have the same curve
Voltage divider	$V_{in} = V \times (R1 + R2) \div R2$	for voltages higher than the converter's supply, for example a battery

Common mistakes

What happens	Why	Fix
Moisture above 100 % or below 0 %	the reading falls outside the two calibration points	clamp the result (constrain) or calibrate again
The soil sensor fails after a few weeks	pushed in past the line: the electronics at the top are not waterproof	respect the line; DFRobot suggests heat-shrink tubing
Different readings when you move the sensor	depth and soil compactness change the output	calibrate the sensor in the position where it will stay
LM35 reading wrong with the ESP32	powered at 3.3 V, below the 4 V minimum	use the TMP36, which works from 2.7 V
Temperature "accurate" to a hundredth but wrong	the ADC resolution is not the sensor's accuracy	compare with the $\pm^\circ\text{C}$ in the datasheet
NTC reading a few tenths too high	self-heating: the divider current warms it up	power the divider only during the reading

The method works everywhere. Photoresistors, pressure sensors with a voltage output, amplified load cells: if the datasheet has a formula, use it; otherwise, two known points and a straight line. For strongly curved sensors you use more points or a lookup table.

Sources: DFRobot, wiki and schematic of the SEN0193 "Capacitive Soil Moisture Sensor" (calibration procedure and warnings) · Analog Devices, TMP35/TMP36/TMP37 datasheet (Rev. H); Texas Instruments, LM35 datasheet (SNIS159H) · Texas Instruments, ADS111x datasheet (SBAS444E); Adafruit, Adafruit_ADS1X15 library · Adafruit, "Thermistor" guide and datasheet of thermistor 4890 (resistance-temperature table) · Microchip, ATmega328P datasheet; Espressif, Arduino-ESP32 documentation (ADC)