# MICROCONTROLER BASED TEMPERATURE SENSORS AND RESISTANCE MEASURMENT

**Abstract -** *This research explores microcontroller applications in temperature sensing and resistance measurement. It covers various Arduino-compatible sensors, emphasizing their pros and cons, and discusses effective thermal anchoring techniques. The study also presents standard resistance measurement methods and proposes an alternative approach using voltage amplification for enhanced precision. The findings offer practical insights for researchers and enthusiasts working with microcontrollers in experiments requiring precise temperature control and resistance monitoring.*

## INTRODUCTION

This article centers on the Arduino microcontroller board [1], highlighting that alternative boards such as MBED, Hawkboard, Raspberry Pi, or Odroid exist, offering potentially lower costs or superior features compared to Arduino. However, our choice of Arduino is driven by the robust user community, a crucial factor in our interactions with students.

Our experience with Arduino is primarily derived from undergraduate project-based physics labs [2] initiated within the Fundamental Physics Department of Université Paris Sud. These labs aim to provide students with hands-on experience in experimental physics. Students select a subject for a week-long project, design and build the experiment using available lab equipment, and explore physical phenomena using affordable materials and low-cost boards.

This article will detail two projects developed by third-year students. The first project focuses on quantifying the magnetocaloric effect in Gadolinium, while the second, popular among students, involves measuring the resistive transition of a high critical temperature superconductor (HTCS). The techniques employed in these projects can be adapted for experiments requiring the measurement of low voltage while varying the setup temperature, particularly in simple transport characterization of samples in research laboratories.

Throughout the article, we will address two critical aspects of these measurements: thermometry and thermal anchoring on one hand, and the measurement of resistances on the other.

## Determining the temperature of an object using microcontrollers

Temperature stands out as one of the most frequently employed experimental control parameters to induce variations in the properties of a physical system. Numerous temperature sensors are available, catering to different temperature ranges. Instead of compiling an exhaustive list, this paper concentrates on the commonly encountered sensors that are compatible with Arduino readouts. Additionally, we will examine fundamental techniques to establish effective thermal contact between the sample and the thermometer.

### i. Sensor Types

a. Built-in Arduino Sensors

There exists a variety of temperature sensors commonly included in standard Arduino kits. For instance, the Arduino Starter Kit is equipped with a TMP36 low voltage temperature sensor [3], which can also be individually purchased for approximately $1. The operational principle of this sensor relies on the temperature-dependent voltage drop across a diode.

The noteworthy advantage of such thermometers is their direct compatibility with Arduino without the need for additional electrical circuits. Additionally, with the corresponding library downloaded, temperature readings in ◦C can be effortlessly obtained through the computer interface, eliminating the necessity for calibration.

However, it's crucial to acknowledge that these sensors have limitations in terms of accuracy and operational range. Taking the TMP36 sensor as an example, it provides a precision of ±2◦C within the temperature range of -40◦C to +125◦C [3].

These sensors are highly convenient for less demanding temperature readouts, such as atmospheric probes used in student projects [4]. However, they are not well-suited for the level of precision required in many research laboratory experiments.

### b. Thermocouple

Thermocouples represent economical and robust thermal sensors readily available in the industrial market, typically priced at around $15. They offer a broad temperature range, such as from 200◦C to +1250◦C for a type K thermocouple [5], making them suitable for applications involving elevated temperatures. This characteristic sets them apart as one of the few thermometers reliably functional at temperatures significantly higher than room temperature.

Additionally, thermocouples are highly practical for measuring the temperature of small-sized samples. Only the hot junction between the two metals needs to be in contact with the area requiring temperature monitoring. However, the accuracy of temperature readings depends on the thermal stability of the cold junction, and the precision of temperature measurement is lower compared to using a thermistor. The voltage to be measured is small, with the sensitivity of a thermocouple being on the order of tens of microvolts per Kelvin. This sensitivity decreases as the temperature decreases; for example, a type K thermocouple has a sensitivity of 40 $\mu V.K^{-1}$ at room temperature but reduces to 10 $\mu V.K^{-1}$ at liquid nitrogen temperature.

It's important to note that amplification of the voltage signal is necessary to read the temperature with Arduino. Some chips offer ready-to-use thermocouple amplifiers for microcontrollers, such as the MAX31856 breakout, providing a resolution of a quarter of a kelvin when using the Adafruit library [6], with an accuracy of a few kelvins. Alternatively, better sensitivity can be achieved with a custom-made amplifier (as discussed below) and careful attention to detail.

### c. Platinum Thin Resistive Films

Platinum thin films serve as practical and highly reliable resistive thermometers, operating typically within the temperature range of 20 K to 700 K [7]. Consequently, they are well-suited for cryogenic applications, extending down to temperatures reached by liquid nitrogen, as well as for moderate heating scenarios. A notable feature of these sensors is that their response is solely determined by the resistance value at 0◦C [8].

The widely used platinum resistance is known as Pt100, possessing a resistance of 100 Ω at 0◦C and priced at approximately $3 to $5. These thermal sensors exhibit a typical precision of about 20 mK up to 300 K and approximately 200 mK above room temperature. Furthermore, their magnetic field-dependent temperature errors are well-documented [7].

While it is feasible to integrate these resistances into a dedicated Arduino resistance-to-temperature converter, such as the MAX31865 [9], a more straightforward approach often involves directly measuring the resistance with a dedicated electrical circuit, as elaborated in Section III. This approach proves particularly convenient for low or high-temperature measurements, where the Arduino board cannot be maintained at the same temperature as the thermometer and the sample.

### ii. Thermal anchoring

For the temperature measurement to be relevant, the thermometer must be in good thermal contact with the sample. How to achieve a good thermal anchoring is a subject of investigation in itself, but in this section we will outline a few standard tech- niques, focusing on the low temperature case.

To cool down a sample at low temperatures, one could use a Peltier module, but the simplest – and not so expensive – way is to use liquid nitrogen. Some basic safety measures have to be taken to manipulate this cryogenic fluid: use protection glasses, gloves, work in a well-ventilated room and, above all, ensure that it is poured into a vessel that is not leak-tight to allow natural evaporation of liquid nitrogen and avoid pressure build-up in the vessel. Once these precautions are observed, the manipulation is relatively safe.

To ensure that the thermometer indeed probes the sample temperature, the most obvious technique is to solidly attach the thermometer to the sample using good thermal conductors. The thermal sensor can, for instance, be glued onto a cop- per sample holder. The glue then has to retain its properties at the probed temperature range. In the low temperature case, one frequently uses GE 7031 varnish which sustains very low temperatures and can easily be removed with a solvent. Alternatively, the thermometer could be mechanically fixed with a spring-shaped material whose elasticity is maintained at a low temperature, such as CuNi sheets. Upon cooling, the spring-shaped material will continue to apply pressure onto the thermometer, thus ensuring a good mechanical and thermal contact with the sample holder.

Another approach involves thermally insulating the thermometer and the sample from the external environment while placing them in contact with a common thermal bath. This insulation can be achieved by immersing them in a container filled with glass beads of a few millimeters in diameter [10] (as shown in the inset of Fig. 7) or alternatively, sand. These materials provide effective thermal insulation for the sample and thermometer system, isolating it from the external surroundings while allowing for significant thermal inertia.

Moreover, when operating at temperatures close to 77 K, these materials help minimize liquid nitrogen evaporation, resulting in a gradual return of the temperature to room temperature. For instance, with a volume of 1 L of beads initially immersed in liquid nitrogen, the temperature may reach back to 300 K over a span of 3 to 4 hours. The heat exchange between the sensor and the sample is facilitated through the evaporated N2 gas, ensuring homogeneous temperature distribution throughout the entire volume. An alternative method for achieving efficient thermal contact between the sensor and the sample through gas exchange is elaborated in Ref. [11].

## III. Measuring Resistances with Microcontrollers

Microcontroller inputs provide readings of electric potentials. Measuring resistances is a slightly more intricate process than simply connecting a resistance to an ohmmeter. From an educational standpoint, this complexity is valuable as it affords students the opportunity to experiment with the concept of resistance and recognize that even seemingly straightforward measurements can pose challenges. In the following sections, we will introduce standard methods for measuring resistances, with a specific emphasis on the low-resistance case.

### i. Current-Voltage Measurement

The most straightforward arrangement for measuring a standard resistance involves the voltage divider setup depicted in Fig. 1. In this configuration, the resistance of interest ($R_0$) is connected in series with a reference resistance (R_ref). The voltage drop across both resistances is regulated by the 5 V output from the board. The potential V1 at a specific point in the circuit can be read using one of the microcontroller’s inputs and should be close to 5V.

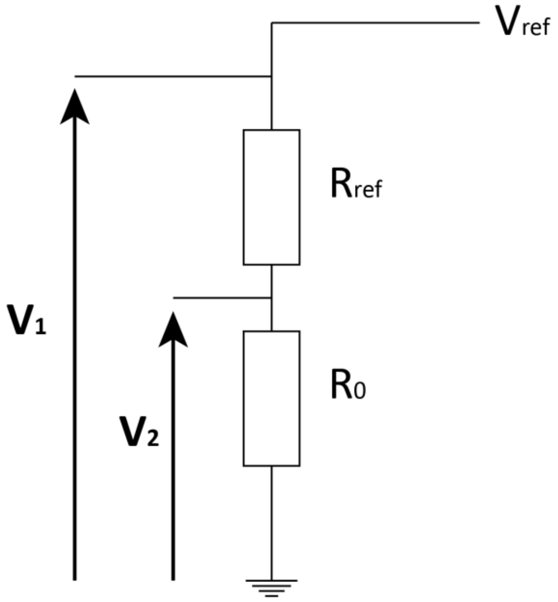

Figure 1: Schematic representation of the current-voltage set-up.

The potential V2, which is read by a second input, corresponds to the voltage drop across the unknown resistance ($R_0$). $R_0$ can then be determined using the simple relation:

$$R_0 = \frac{V_2}{V_1 - V_2} R_{ref} \qquad (1)$$

Monitoring V1 provides better precision by directly observing the current. The 5 V output from the Arduino may experience variations over time. To enhance voltage stability, it can be beneficial to use an external power source for the microcontroller instead of relying on the computer's USB output. It's worth noting that when Rref is much greater than $R_0$, the current through the circuit can be considered constant, which is often convenient for resistance measurements not requiring high precision.

However, this method has limitations when dealing with small values of $R_0$. Given that the ultimate resolution of an Arduino UNO board is approximately 1 mV with Vref = 1.1 V, $R_0$ should not be smaller than about 2 * 10^(-4) * R(ref). For instance, in the case of a standard commercial high-temperature superconductor (HTCS) sample, the normal state resistance is often on the order of a few tens of mΩ. To observe the resistance drop across the critical temperature (Tc) of a superconductor, R(ref) should be in the range of a few ohms. Such resistances are commercially available or can be custom-made with relative ease.
Achieving good precision (within the order of a few mΩ) is feasible by employing a lengthy copper wire. Commercially available copper wires with a diameter of 0.2 mm, for instance, exhibit a resistance of approximately 0.5 Ω/m. However, to optimize measurement precision, it is imperative to amplify V2. Additionally, utilizing this approach for measuring small resistances may result in the circuit current surpassing the maximum allowed current at the microcontroller's output. Subsequently, we will explore an alternative method for measuring small resistances.

### ii. Wheatstone Bridge

Another method for determining resistance, offering high precision, is the Wheatstone bridge. The measurement principle is depicted in Fig. 2. In this configuration, R1 and R3 are fixed-value resistances, whereas R2 is an adjustable resistance, and R0 represents the resistance of interest. The relationship between the potentials V1 and V2 is then expressed as:

$$\underset{2 \qquad 1}{\underline{\bar{R}_2\ \overline{\qquad}\quad (2)}} - \frac{R_0}{R_1 + R_2 \qquad R_0 + R_3} V$$

The bridge is deemed "balanced" when R2 is adjusted to the point where V1 and V2 are equal. At this balanced state, the resistances are interconnected by the following relationship:

$$R = \underline{R_2R_3}0 \quad R_1 \qquad (3)$$

The precision attainable with the Wheatstone bridge method, especially when integrated with a microcontroller, is comparable to that of the current-voltage measurement approach. However, this method proves impractical when working with variable resistances, such as R0, as the bridge must be closely maintained near balance at each measurement point. Notably, it is not well-suited for measuring the resistive transition of a superconductor.

### iii. Voltage Amplifier

For the practical measurement of small voltages, and consequently small resistances, the most effective solution is to amplify the potential difference across the resistance. This amplification is achieved through standard voltage amplification setups utilizing operational amplifiers, available in either single-ended or differential input configurations. In the single-ended case, as depicted in Fig. 3, the output potential is determined by:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2} \qquad (4)$$

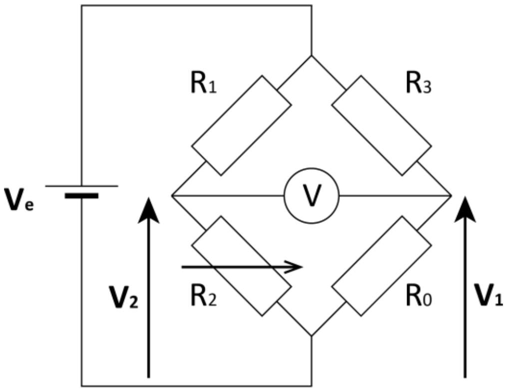

Figure 2: Schematic representation of the Wheatstone bridge: $R_1$ and $R_3$ are fixed resistances, $R_2$ is a tunable resistance, and $R_0$ is the resistance of interest.

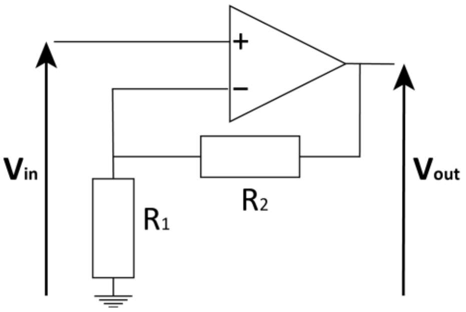

Figure 3: Voltage amplification.

The input voltage $Vin$ can then be amplified at will, depending on the ratio $R2/R1$. The output voltage $Vout$ can be read by the microcontroller.

This amplification method offers significantly higher precision compared to previously mentioned methods. It eliminates the need for tuning at each data point and can be applied to measure various small voltages, such as the voltage drop across a superconductor, potential difference across a thermocouple, or derivation of thermoelectric coefficients (Seebeck or thermopower). Going beyond this straightforward amplification may involve more complex methods, such as fabricating a microcontroller-based lock-in amplifier, as demonstrated in Ref. [12].

### iv. Using Another ADC than Arduino's

The specifications of the Arduino Digital-Analog Converter often limit the accuracy of the aforementioned measurements. The Arduino ADC provides, at best, 10 bits on the 1.1 V internal reference voltage and can only measure voltage in single-ended configurations.

An alternative is to use another microcontroller with a superior ADC. For instance, the low-cost FRDM-KL25Z from NXP [14] features an ADC capable of measuring voltage in single-ended or differential input configurations with 16 bits on 3.3 V.

While the ease-of-use and a large user community may favor Arduino, an external ADC can be employed when better resolution or differential mode configuration is required. For example, the ADS1115 chip [15] has been tested, offering the capability to measure 4 single channels or 2 differential channels with 16 bits on 4.1 V. The possibility of preamplification up to 16 times improves resolution to 8 μV per bit, compared to the standard 5 mV (or 1 mV with the 1.1 V internal reference).

The ability to measure voltage differentially with a resolution better than 10 μV opens up various possibilities for physics measurements, such as strain gauge or four-wire configuration resistance measurement, direct measurement of a thermocouple, or superconductor resistance across the transition.

The main drawback of this method is its complexity compared to using the Arduino ADC. It requires the installation of a library (with tutorials available online, e.g., Ref. [15]). Additionally, an external ADC is generally less robust than Arduino's ADC, necessitating careful monitoring of the voltage input to prevent damage.

## IV. Evidencing a Magnetocaloric Effect with Microcontrollers

To exemplify these methodologies, we will elaborate on the measurement of the magnetocaloric effect. This phenomenon involves the temperature change experienced by a magnetic material when subjected to a varying magnetic field.

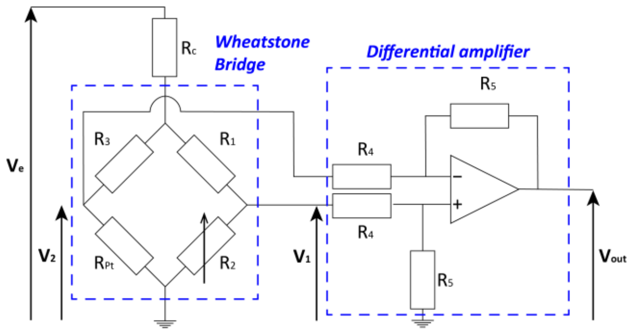

Figure 4: Measurement of the resistance of a Pt resistive thermometer with a Wheatstone bridge and amplified by an opamp-based circuit.

## IV. Evidencing a Magnetocaloric Effect with Microcontrollers

To explore the magnetocaloric effect, we focused on Gadolinium (Gd) due to its paramagnetic properties and Curie temperature (TCurie) close to room temperature (TCurie = 292 K). In our experiment, a 2.242 g Gd sample was exposed to a neodymium magnet generating a magnetic field with a maximum strength of 0.51 T at a temperature of approximately 298 K. For a more detailed explanation of the phenomenon, please refer to Ref. [16].

The challenge in this experiment was measuring the minute temperature difference induced by the application of a magnetic field. To achieve this, a Pt100 thermistor was placed in thermal contact with the Gd sample using thermal paste. The resistance change was measured by a Wheatstone bridge with the following characteristics: R1 = R3 = 100 Ω, R2 was set at 108 Ω to approximate balance at the specified temperature, and Ve = 5 V via Arduino's internal source. An additional resistance (Rc = 800 Ω) was included in series to limit the current through the Pt100, preventing excess heating of the thermometer. Ve is then replaced by (R1 + R2) Ve in Eq. (2). The off-balance potential difference (V2 - V1) was differentially amplified with a gain of 100 (R4 = 1.5 kΩ and R5 = 150 kΩ). The resulting voltage Vout was then read by the board (Arduino Mega in this case), using 2.56 V as Arduino's ADC reference voltage [17]. The overall read-out circuit is schematically depicted in Fig. 4. The temperature was then determined using the linear relationship of Pt100 response in the range [273 K - 323 K]:

$$T\,[K]=2.578R_{pt}\,[\Omega]+15.35 \qquad (5)$$

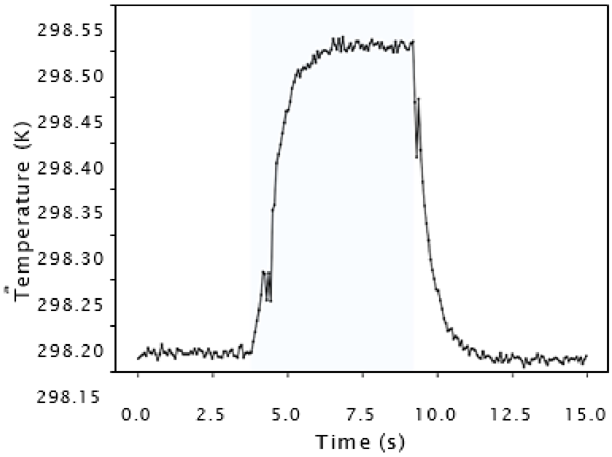

Figure 5: Magnetocaloric effect in a Gd sample submitted to a 0.51 T magnetic field (blue background) before going back to the zero-field situation (white background). Each data point corresponds to the average of 50 measurements. The noise level is of the order of10 mK.

As illustrated in Fig. 5, the magnetocaloric effect is clearly visible with an amplitude of about $\Delta T = 0.33 \pm 0.01$ K and a time scale of a few seconds. The resolution of the setup corresponds to 50 mK (18 mΩ). Each data point in Fig. 5 corresponds to an average of 50 measurements so that the effective noise that can be observed is of about 10 mK, or about 5 mΩ in resistance. This yields a relative precision for the measurement of a few 10^(-5), which is remarkable given the simplicity of the apparatus. When the magnet is taken away from the Gd sample, the temperature decreases back to its initial value, as predicted by the isentropic character of the magnetocaloric effect.

## V. Measuring a Superconducting Resistive Transition with Microcontrollers

In the second experiment, we aim to detail the measurement of the superconducting resistive transition of a High Critical Temperature Superconductor (HCTS). In these compounds, the critical temperature (Tc), below which the sample becomes superconducting and exhibits zero resistance, is higher than 77 K. Therefore it can be easily observed by cooling the sample down to liquid nitrogen temperature and subsequently warming it back up.

**References**

[1] Arduino Project Website, https://www.arduino.cc/

[2] F Bouquet, J Bobroff, M Fuchs-Gallezot, L Maurines, Project-based physics labs using low-cost open-source hardware, Am. J. Phys. 85, 216 (2017).

[3] TMP36 specifications, https://www.arduino.cc/en/uploads/ Main/TemperatureSensor.pdf

[4] V K Merhar, R Capuder, T Maro˘sevi˘c, S Arta˘c, A Mozer, M S˘tekovi˘c, Vi˘c goes to near space, The Physics Teacher 54, 482 (2016).

[5] Thermocouple operation temperature range, https://www.omega.co.uk/techref/ colorcodes.html

[6] Thermocouple amplifier, https://www.adafruit.com/product/3263

[7] Characteristics of Pt thermometers are available at the following website, https://www.lakeshore.com/Documents/ LSTC_Platinum_l.pdf

[8] Platinum RTD sensor resistance to temperature conversion tables are available at the following websites, https://www.omega.com/techref/pdf/ z252-254.pdf, www.intech.co.nz/products/temperature/ typert/RTD-Pt100-Conversion.pdf

[9] RTD-to-Digital converter, https://www.maximintegrated.com/en/ products/sensors/MAX31865.html

[10] G Ireson, Measuring the transition temperature of a superconductor in a pre-university laboratory, Physics Education 41, 556 (2006).

[11] L M Le´on-Rossano, An inexpensive and easy experiment to measure the electrical resistance of high-Tc superconductors as a function of temperature, Am. J. Phys. 65, 1024 (1997).

[12] K D Schultz, Phase-sensitive detection in the undergraduate lab using a low-cost microcontroller, Am. J. Phys. 84, 557 (2016).

[13] R Henaff, G Le Doudic, B Pilette, C Even, J M Fischbach, F Bouquet, J Bobroff, M Monteverde, C A Marrache-Kikuchi, A study of kinetic friction: The Timoshenko oscillator, Am. J. Phys. 86, 174 (2018).

[14] https://os.mbed.com/platforms/KL25Z/

[15] https://www.adafruit.com/product/1085

[16] V Percharsky, K A Gscheider Jr, Magnetocaloric effect and magnetic refrigeration, J. Magn. Magn. Mater. 200, 44 (1999).

[17] https://www.arduino.cc/reference/ en/language/functions/analog-io/ analogreference/