U.Porto Journal of Engineering, 3:3 (2017) 42-47
ISSN: 2183-6493
DOI 10.24840/2183-6493_003.003_0005

Received: 8 April, 2017
Revised: 10 May, 2017
Published: 24 November, 2017

Real Time Temperature Monitoring

Temperature Measurement through a Thermistor Inserted into a Properly Sized Circuit using MatLab and Arduino Platforms

Jorge Silva

Department of Electrical and Computer Engineering, Faculty of Engineering University of Porto, Porto, Portugal (up201610377@fe.up.pt) ORCID 0000-0001-9808-9864

Abstract

The purpose of this project is to measure the temperature through a thermistor inserted into a properly sized circuit.

This implementation was designed to be possible, given the characteristics of the used sensor, a measurement in the range of -5 to 270 ºC, although the temperatures of interest of this application are between 18 and 35 ºC.

For the analysis of this measure, as well as for the control, an application was developed in MatLab. This software allows, not only programming and processing the acquired signal, but also serves as a graphical interface where it is possible to monitor the entire process in real time. For the acquisition of the analog signal, was uses an Arduino UNO board.

Inside of the MatLab environment, a screen has been developed that allows visualizing the temperature evolution in real time, as well as verifying if it is inside the stipulated limit. The developed application also allows the system to stop and store test data in a file for later calculations.

For a better framing, this circuit can be considered as a system developed for temperature control of a typical classroom.

That being said, the is then to control the temperature, stipulating limit values, to visualize its evolution in real time through an indicator and a graph and to collect data for later calculations.

Keywords: Temperature, Thermistor, MatLab, Arduino, Temperature Control, Temperature Monitoring

Type: Research Article
Open Access Peer Reviewed CC BY

1. Introduction

In the framework of the project, measuring temperature becomes an extraordinarily important factor, because in this case the fact that it can reach higher values can lead to a heating of an entire system, as well as could cause serious damage. Therefore, for its measurement, it was decided to choose a suitable chain of measurement, in order to obtain values that are, at least, acceptable.

Consequently, this work also has the great advantage of connecting the MatLab and Arduino software, when associated to an acquisition circuit.

1.1. Temperature measure

Temperature measurement is one of the most common and most important measurements in process control.

There are phenomena that are influenced by temperature, and consequently, are used to measure it: variation of resistance of a conductor (resistance probes - RTD's); resistance variation of a semiconductor (thermistors); f.e.m. created in the union of two metals (thermocouples).

The application discussed in this article was developed with the use of an NTC thermistor (Negative Temperature Coeffcient) as a sensor, which has temperature-dependent resistance inversely proportional. There are also PTC thermistors (Positive Temperature Coeffcient), which have as their peculiarity to have a transition point, only from a certain temperature will exhibit an ohmic variation with the temperature variation. However, the NTC is more used than the PTC, due to the greater ease of being manufactured.

1.2. NTC thermistor

Thermistors, or thermal resistors (temperature sensitive resistors), are semiconductors with high thermal resistance coefficients (Figure 1).

Figure 1: 22 kΩ NTC

Most NTC's used in instrumentation are sintered compounds of metal oxides, such as manganese, nickel, cobalt, copper, iron and uranium, with resistance from 0.5 to 75 MΩ and allow to measure temperatures from -100 to 300 ° C. These devices include a resin based encapsulation that serves as protection and stabilization (Johnson 1990).

Due to the fact that it is manufactured with semiconductor material, the NTC is one of the temperature sensors that give the greatest variation of output voltage by temperature variation, but the reality of being so sensitive makes it behave in a non-linear way (Helfrick and Cooper 1994; Johnson 1990). The curve that defines the relation between temperature and resistance has an exponential behavior (Figure 2).

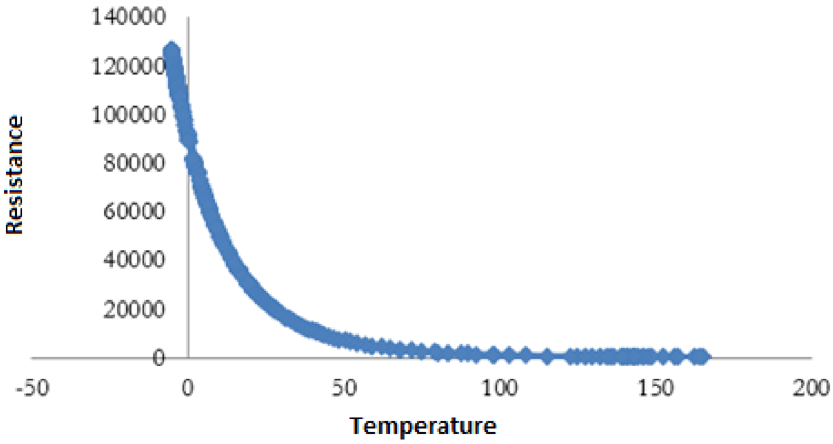


Figure 2: Temperature-resistance ratio in a measurement between -5 and 270 °C

From Figure 2, it can be concluded that the resistance of the thermistor is in fact extremely sensitive to temperature, so it is necessary to analyze the exponential that characterizes the temperature as a function of the resistance (Equation 1).

$$R_T = R_0 \, e^{\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)} \quad (1)$$

Where, R_0 is the resistance of the thermistor at the reference temperature (25 °C); T is the temperature to be obtained in Kelvin; T_0 is the reference temperature in Kelvin; β is a constant of the thermistor supplied by the manufacturer, given in Ohm.

Likewise, the expression that will be used to calculate the temperature measured in the thermistor will be as follows (Equation 2).

$$T = \beta\left(\frac{T_0}{\beta + T_0 + ln\left(\frac{R}{R_0}\right)}\right) \tag{2}$$

This conversion method used is called “B Parameter” (beta). It is not of the most precise methods, but nevertheless it is of simple application.

Thermistor characteristics make it one of the most widely used temperature sensors in a variety of application areas, such as agriculture (temperature control in greenhouses), electrical sector (motor safety, generators and transformers), electronic components (protection against heat sinks in personal computers) and even in the hospital area (temperature control in surgical blocks and incubators) and so many others (Soisson 2002).

2. Materials and methods

Arduino is a low cost, affordable electronic prototyping platform designed to allow the development of interactive system control. In addition, all the material (software, hardware and libraries) is open source, that is, it can be used without the need of payment of licenses. Its platform consists essentially of two parts, the hardware (Figure 3) and the software.

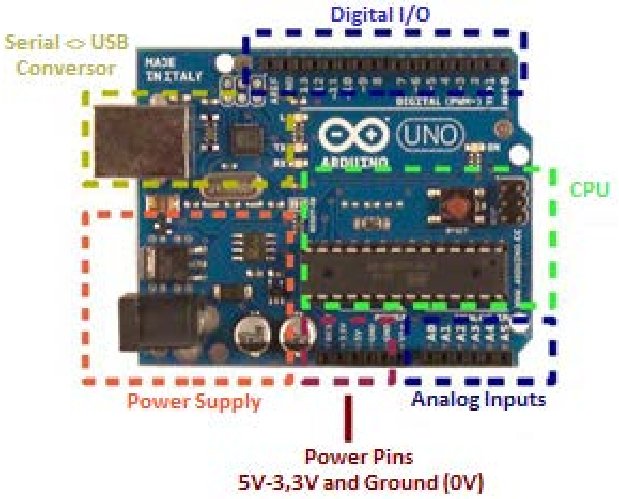


Figure 3: Arduino UNO board

The hardware (Arduino UNO board) is very simple, but very efficient. It consists of a microprocessor (ATMega 328) that has 28 pins. It is through these pins that provide access functions of the microprocessor, send data to its memory and trigger external devices (LEDs, motors, relays), configuring them as analog inputs and digital outputs (Campa 2009).

The microprocessor also has 3,3 and 5V fixed power outputs, ideal for this type of applications, and a 6 channel analog-to-digital converter (for 6 pin analog inputs) with a 10 bit resolution that returns integer numbers from 0 to 1023 bits. To convert a bit reading to voltage values, assuming a 5V power supply, is performed the following conversion (Equation 3).

$$V_{pin} = \frac{Reading \times 5}{1023} \tag{3}$$

The Arduino development environment (software) is a C/C ++ compiler, which allows create various input and output operations.

However, throughout the development of the project came the idea of developing a GUI (Graphical User Interface).

Arduino in spite of the characteristics mentioned above does not have a graphical user interface. From this, came the idea of using the MatLab software functions to process the analog signal acquired from the Arduino UNO board.

MatLab (Matrix Laboratory) is a high-level programming environment that has application and programming features that other platforms do not allow. It is one of the most used mathematical tools in Engineering that offers the user innovation and sophistication, and the nonchalance with some items, such as variable declaration, memory allocation, pointer usage and other routine tasks.

In addition to the many advantages, the main motivation for using this software in this project is the possibility of constructing graphs, in which a set of data can be visualized and manipulated in different ways.

2.1. Connecting MatLab to Arduino UNO board

In order to connect MatLab to Arduino UNO board it was necessary to implement (in Arduino) a server of analog and digital outputs and inputs that will allow MatLab to control the Arduino UNO board, so that it understands all the commands sent.

After the connection, was also developed a code in Arduino that allows to read the analog input, to calculate the temperature and to activate the digital outputs, just to verify the good operation of the plate and the components used.

2.2. Experimental setup

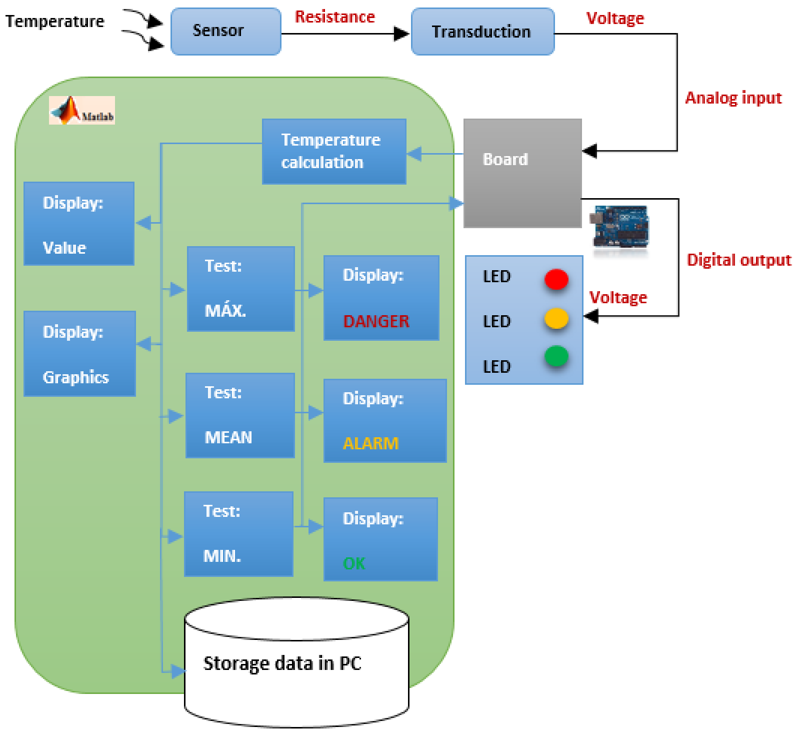


Figure 4: Block diagram of the project

The idea was to assemble the complete circuit (Figure 5) that allowed the signal transmission to the acquisition board, configure MatLab in the most appropriate way to receive the signal from the board, and already in the environment of the software, to develop an algorithm that allows the processing of the data, its storage in a file, and the activation of digital outputs.

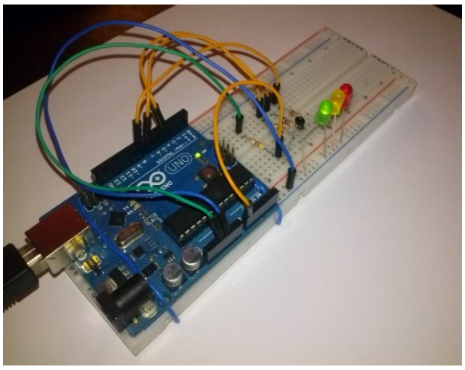
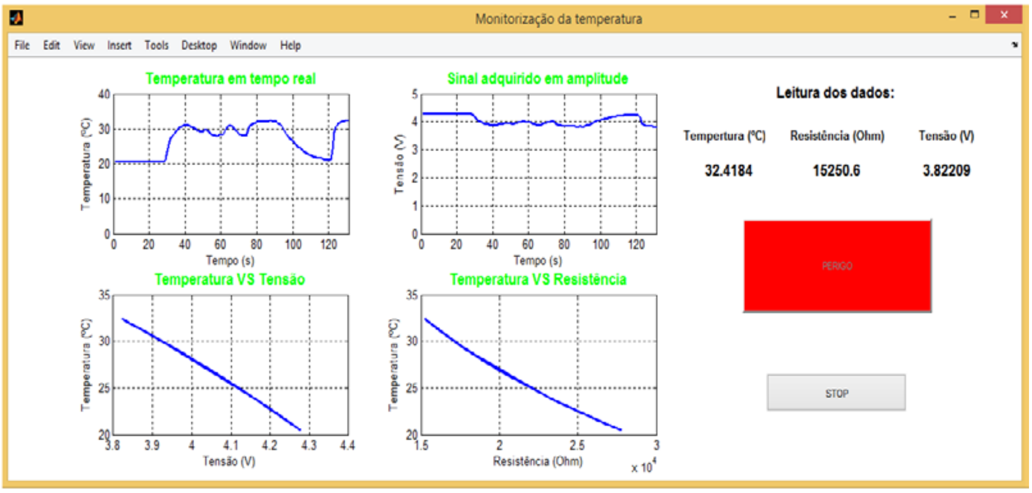


Figure 5: Circuit assembly

The resistance variation of the thermistor is converted to voltage through a transducer, which is constituted by a voltage divider between the thermistor and a 4,7 kΩ resistor. The digital outputs consist of the activation of LEDs, protected with 1 kΩ resistors, according to the limits stimulated in the algorithm.

3. Results and discussion

The final result of the project was translated into what was proposed in the objectives, initially and in the long term of the project: a graphic screen that allows monitoring in real time the temperature (Figure 6). In the same screen is possible to see the acquired signal (voltage) over time, a variation of the temperature as a function of the resistance and the voltage, numerical indicators that indicate the value of the temperature, voltage and resistance, display that changes color depending on the state of the process, and a stop button that allows to close the cycle.



Leitura dos dados:

Tempertura (°C)	Resistência (Ohm)	Tensão (V)
32.4184	15250.6	3.82209

Figure 6: Graphic interface developed in MatLab

At the same time as the algorithm that performs all the calculations and represents the graphical interface, another algorithm, also developed in this project, receives the execution data and stores it in an EXCEL worksheet (in a file of type .xlsx), to further calculations and a better analysis of the results obtained.

The stored data allowed to trace the calibration curve of the thermistor. In order to cover a greater possible range of temperatures, ice was used, which cooled the thermistor to -5 °C and a heat source (fire) which heated the thermistor to about 270 °C (Figure 7).

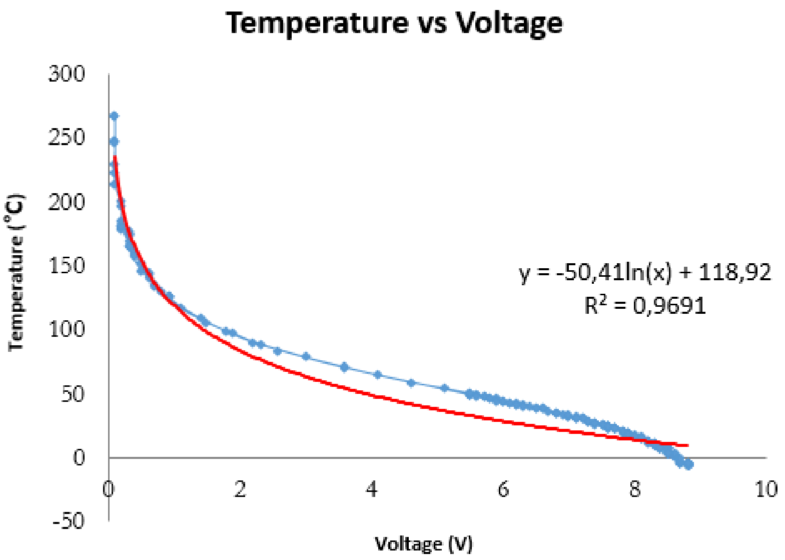


Figure 7: Temperature as a function of the voltage of the acquired signal

A good approximation was obtained with a logarithmic fit, where the correlation is 0,9691, and the resulting equation is:

$$T = -50{,}41\ln(V) + 118{,}92 \tag{4}$$

4. Conclusions

It is emphasized in a somewhat informal outline that felt the goal fulfilled. Timely planning, research and reinforcement of competencies made one feel a full sense of mission accomplished.

Having verified the advantage of the use of MatLab software with the low cost Arduino UNO board, it is evaluated as potential future developments the analysis of other sensors, such as humidity, light and pressure sensors.

It was perhaps necessary to delve into some detail, to improve failures and to reformulate strategies and techniques, but as it was verified the obligation to prevail for the KISS (keep it small and simple) orientation and also for the prioritization of what was essential to the detriment of what would be ancillary, ended up not creating illusions that were difficult to operate.

However, the results obtained constitute a solid starting point for future projects in this area, once the observed constraints have been suppressed.

References

Campa, G. 2009. "Learning basic mechatronics concepts using the Arduino board and MatLab". Webinar [The Mathworks]. Accessed April 8, 2017. https://www.mathworks.com/matlabcentral/fileexchange/27843-legacy-matlab-and-simulink-support-for-arduino--slides-and-examples.

Helfrick, A. D. and Cooper. 1994. *Modern electronic instrumentation and measurement techniques*. Prentice Hall.

Johnson, C. D. 1990. *Process control instrumentation technology*. Fundação Calouste Gulbenkian.

Soisson, H. E. 2002. *Instrumentação industrial: padrões e unidades de calibração... controle ambiental*. Curitiba: Hemus. 852-89-0145-9.