Overview

A mini-project to build a system to automatically measure temperature. Topics include:

- Thermistor
- Arduino micro-controllers
- Analog-to-Digital Converter (ADC)
- Op-amp Follower

You are allowed 6 pages for your Report for this lab.

Background preparation before the lab

- Read through these Lab instructions and do any research needed to accomplish the lab tasks.
- Read notes from Lecture 5.
- Look at Vishay NTCLE100E3103HB0 thermistor datasheet
- Look at the Arduino IDE. If downloaded onto your own computer, allows you to
  - write and compile code so it is ready to try when you come into the lab.
  - control and take data from an Arduino.

Answer these questions in your lab notebook Before The Lab

The Instructor or TA will ask to see the answers in your notebook during the lab. We expect to see a sentence or two and/or calculations that justify your answer.

1. When a fixed voltage is applied across a thermistor, the current through the thermistor changes monotonically with temperature. True or False?

2. What value of $R_{ref}$ have you decided to use in Exercise 1b?

3. What are the minimum and maximum voltages that can be safely measured on an Arduino analog input pin?

4. A 10-bit analog-to-digital converter has a voltage range of 5 V. A signal that is sent into this ADC gets discretized with a smallest voltage step size of $\sim 5$ mV. True or False?

1) Measuring a thermistor’s output

A thermistor is a resistor whose resistance depends on temperature as shown in Figure 1. You will be using a Vishay NTCLE100E3103HB0 thermistor, an NTC type with a negative temperature coefficient, i.e., its resistance decreases with temperature. Its nominal resistance at 25 C is $R_{25} = 10k\Omega$.

The thermistor’s datasheet provides the following formula for converting its resistance to temperature in Kelvin:

$$\frac{1}{T(R)} = A_1 + B_1 \ln \frac{R}{R_{25}} + C_1 \left( \ln \frac{R}{R_{25}} \right)^2 + D_1 \left( \ln \frac{R}{R_{25}} \right)^3$$

*largely based on previous write-ups by Amar Vutha (2016-2018)
The constants for this thermistor type (in units of $K^{-1}$) are $A_1 = 3.354016 \times 10^{-3}$, $B_1 = 2.884193 \times 10^{-4}$, $C_1 = 4.118032 \times 10^{-6}$, $D_1 = 1.786790 \times 10^{-7}$.

Figure 1: Resistance versus temperature for a negative temperature coefficient (NTC) thermistor. $R_{25}$ is the nominal resistance of the thermistor at 25°C.

a) Measure the resistance of the thermistor, $R_{th}$, using a multimeter. Record the room temperature, $T_{room}$, and verify that $R_{th}$ is consistent with the curve shown in Figure 1.

b) In order to automatically record the thermistor’s resistance, it is convenient to produce a voltage that depends on its resistance. Do this using a voltage divider as shown in Figure 2. Use a value of $R_{ref} \sim 1 - 5k$, and $V_{in}$ of +5 V from your HP 6235 DC power supply. Measure and record the value of $R_{ref}$ before you put it in. Measure the output voltage of the divider when you touch the thermistor’s head with your fingers. Estimate your hand’s temperature using the calibration curve in Figure 1, and the calculated division ratio of the divider.
2) Arduino Microcontroller

The voltage produced by the thermistor divider circuit is a complicated function of the temperature, so it would be convenient to directly convert the thermistor’s output into a temperature. A digital device called a microcontroller can automate this measurement and calculation. Microcontrollers are extremely versatile devices, and are handy for many tasks where complicated algorithms and nonlinear operations need to be implemented.

Arduino Setup

WARNING: To avoid zapping - and possibly destroying - your Arduino with a static discharge, it is best to always touch a ground (e.g.
on your scope) before touching your Arduino.

Plug in your Arduino board into any available USB port on the lab computer.
- You may use your own laptop if it has the correct ports and you have downloaded the Arduino IDE.
- Start up the Arduino program on the desktop, and configure the software to communicate with an Arduino Uno board.
  - Answer “No” to any Arduino update requests on the lab computers.
- Make sure that the Board under the Tools menu is chosen to be “Arduino/Genuino Uno”, and the Port is chosen to be COM4 (Arduino/Genuine Uno).

To upload and run a program (called a “sketch”) to the Arduino, you first need to compile it, and upload it (using the button shaped like a right arrow). Try the simple Blink example first:
- Select File → Examples → 01.Basics → Blink
- Compile Blink by clicking on Sketch → Verify/Compile or on the Verify button shaped like a check mark.
- Upload Blink to the Arduino by clicking on Sketch → Upload or on the Upload button shaped like a right arrow.
- The yellow “L” LED on your Arduino should start blinking every second.

Play with the code to become familiar with the programming interface, e.g. change the code so the LED blinks in different patterns. After modifying the code, it is usually sufficient to click Upload to compile and upload it.
- Note that the delays in the code are in milliseconds

3) Voltage source

A device’s power supply typically provides only one voltage, but it is often the case that that different components within the device require different voltages. For example, the op-amp used later in this exercise may use ±15V, but we only want a maximum of 5 V input into the Arduino. Another common issue is that the input voltage is not known exactly a priori, but we need to supply a precise voltage to a circuit.
In this task you will use an LM7805 voltage regulator IC to generate a stable voltage. The pin labels are listed in the IC’s datasheet.

a) Wire up a 7805 regulator as follows:

- connect the +18 V output of the HP 6235 DC power supply to the input pin of the 7805 (pin 1)
- connect the power supply COMMON output to the ground pin of the 7805 (pin 2).
- also connect the COMMON output to the ground provided by the oscilloscope

![Circuit for testing the LM7805 voltage regulator. Choose capacitors between 1 nF and 1 µF.](image)

Measure the DC voltage at the output of the 7805 (pin 3) using a multimeter or oscilloscope. Verify that the 7805 puts out a reasonably constant voltage over a large range of power supply voltages.

b) Attach resistors ranging from 30 Ω to 1 MΩ to the output of the regulator, and verify that the 7805 puts out a constant voltage regardless of the load resistance.

4) Arduino Analog Input

**WARNING:** Applying voltages outside the range of 0 to +5 V to the Arduino analog input pins may destroy the Arduino’s Analog-to-Digital Converter (ADC).

The input impedance of the Arduino ADC is smaller than $R_{ref}$, so if the ADC is fed directly by the thermistor divider, $V_{out}$ will sag leading to inaccurate temperature measurements. Therefore we will construct an op-amp follower to sit between the divider and the microcontroller. The follower presents a large input impedance to the divider, and a small output impedance to the microcontroller.

a) Op-amp follower

Wire up an op-amp follower using an LF356, as shown in Figure 4. An op-amp follower is just a non-inverting amplifier (Lab 4 Fig. 2) with $R_2 = 0$ and $R_1 = \infty$, hence a gain of $V_{out}/V_{in} = 1$. The 1N751A Zener diode to ground ensures that the output of the op-amp never exceeds 5.1 V, protecting the Arduino input. Apply a variety of waveforms to the input of the follower, and verify that the output faithfully tracks the input, but never exceeds 5.1 V.
b) Analog Input Test

To learn how to measure voltages using the Arduino:

- connect your LM7805 voltage regulator circuit output to a resistive divider with a fixed resistor in series with a potentiometer (a small variable resistor)
- feed the voltage across the potentiometer into your zener-protected op-amp follower circuit.
- feed output of op-amp into pin A0 of your Arduino.

Use the File → Examples → 01.Basics → AnalogReadSerial example to read the voltage from pin A0.

- To see the output from the running sketch, click Tools → Serial Monitor
  - It is best to change the delay in the sketch to at least 1000 so the numbers don’t scroll too fast
  - Make sure the Autoscroll box below the output screen is ticked.
- If pin A0 does not work, e.g. because it has been fried by a previous student applying to large a voltage, try another pin and change the sketch code accordingly.

By changing the potentiometer, you can feed various voltages into the Arduino. By measuring the input voltage with a multimeter or oscilloscope, and comparing it to the values reported by the Arduino, calibrate the Arduino voltage measurements and report the conversion factor between volts and the Arduino output.

5) Arduino Temperature Measurement

Replace your potentiometer divider circuit with your thermistor divider circuit. Power your thermistor divider with the output of your LM7805 voltage source circuit.

Write an Arduino program that converts the output of the thermistor circuit into temperature and prints it out on the serial port. A basic program to get you started, arduino_thermistor.pde, is available on the course resources page.

Use Tools → Serial Plot to show the temperature as a function of time, while heating up the thermistor with your hand and then removing your hand to let it cool down. Take a photo of the plot and include it in your report, noting on the photo the times when you are touching the thermistor.

- It is also possible, with a more advanced sketch, to save the data to a file on the computer so it can be plotted offline.
- If you optionally connect the thermister to longer wires, it can be placed in ice water or boiling water for further calibration.