

# S2014, BME 101L: Applied Circuits Lab 1

## Temperature measurement—Thermistor

Kevin Karplus

March 28, 2014

## 1 Design Goal

The design goal for this week’s pair of labs is to build a temperature monitor that can record temperature measurements at frequent time intervals (say, once a second) using a microprocessor.

We want maximum sensitivity of the monitor at a specified operating temperature  $T_{op}$ , but the range should be at least 0°C to 100°C. (Note:  $T_{op}$  will not be announced until Thursday’s lab period starts, so do your calculations symbolically—you’ll need some numbers you won’t have until Wednesday anyway.)

## 2 Background

### 2.1 Thermistors

A thermistor is a semiconductor device whose resistance varies with temperature. Read

[http://en.wikipedia.org/wiki/Electrical\\_resistance](http://en.wikipedia.org/wiki/Electrical_resistance)

<http://en.wikipedia.org/wiki/Thermistor>

Thermistors are “negative thermal coefficient” devices, which means that the resistance decreases as the temperature increases (unlike most metals, which have a positive thermal coefficient of resistance, increasing resistance as temperature increases).

There are various formulas used for estimating the resistance at a particular temperature (or, equivalently, the temperature for a particular resistance). The simplest such formula, used on many specification sheets for thermistors, is the “B” equation:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \left( \frac{R}{R_{T_0}} \right) ,$$

where  $T$  is the temperature in degrees Kelvin,  $R$  is the resistance,  $B$  is a parameter that depends on the material used in the thermistor, and  $R_{T_0}$  is the resistance at some calibration temperature  $T_0$ . We can also write this as

$$R = R_{T_0} e^{B(1/T - 1/T_0)}$$

or

$$R = R_\infty e^{B/T} ,$$

where  $R_\infty$  is the projected “resistance at infinity”,  $R_{T_0} e^{-B/T_0}$ .

For example, the thermistor we will use in this lab (NTCLE413E2103F520L) has a data sheet at <http://www.vishay.com/doc?29078> (Note: on future labs we won’t be providing URLs for the data sheets—you’ll be expected to look for, download, and read the data sheets without being specifically instructed to do so. We will help you try to understand the most important parts of each data sheet.)

The thermistor data sheet unpacks the part number to mean

NTC A negative temperature coefficient thermistor

L Leaded—that is, the device has wires coming out of it (unlike some surface mount devices that are soldered directly to a printed-circuit board, for example)

E413 describes the wires and encapsulation: 30-gauge wires with the thermistor encapsulated in PVC and epoxy, rated for a maximum of 105°C. So this thermistor would not be suitable for measuring the temperature in an autoclave.

E2 Tin-alloy on the wires.

103 A nominal resistance of 10kΩ at 25°C. We'll see a lot of parts labeling where a number like 103 should be interpreted as 10.E3 or  $10 \times 10^3$ .

F The nominal resistance should be accurate to  $\pm 1\%$

520 The leads are 52mm long.

L The  $B$ -value is “low”, between 3000 and 3500.

Further down in the data sheet, they give the  $B_{25/85}$  value as 3435. This is the  $B$ -value that they got from a pair of measurements: the resistance at 25°C and the resistance at 85°C. They also provide a table of measured or calculated resistances (they don't say which) for the 10kΩ 1% low- $B$  device we are using. You are not to use this table in the lab, but to make your own independent measurements—though you can check against this table to see if the measurements you are making are close to the specs.

The  $B$  equation is an ok approximation if you are keeping fairly close to the reference temperature, but using just 2 measurements to estimate  $B$  is a bad idea, particularly if you plan to use the thermistor at lower temperatures than either measurement. If you want to use the  $B$  equation, you need to fit the best values for  $B$  and  $R_{T_0}$  (or  $R_\infty$ , the extrapolated resistance at infinite temperature) for a number of measurements across the temperature range you are interested in. When Prof. Karplus did this lab at home, he made 63 measurements, but that would take too long for this lab—10 measurements should be plenty, as long as they span the full range of interest and are made fairly accurately.

There is a better formula for approximating the relationship between thermistor resistance and temperature: the Steinhart-Hart equation [http://en.wikipedia.org/wiki/Steinhart-Hart\\_equation](http://en.wikipedia.org/wiki/Steinhart-Hart_equation):

$$\frac{1}{T} = A_0 + A_1 \ln(R) + A_3 (\ln(R))^3 .$$

It is much messier to invert the Steinhart-Hart equation to get  $R$  in terms of  $T$ , if you need to do that, but having an extra parameter allows much closer fitting of the temperature vs. resistance curve.

## 2.2 Voltage dividers

A *voltage divider* consists of a serial connection of two resistances, as shown in Figure 1. A voltage is applied across the pair of resistors and measured across one of the resistors. We will be using voltage dividers in one form or another for almost every lab this quarter—they are a fundamental design tool for electronics.

The theory of voltage dividers is easily derived from *Ohm's Law*:  $V = IR$  and *Kirchhoff's current law*: in equilibrium all the current into a node sums to 0. If you are not already familiar with those concepts, go look them up on line in *All About Circuits* or Wikipedia.

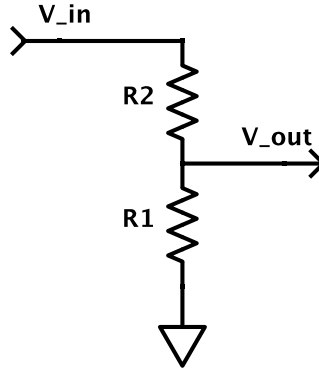


Figure 1: Voltage divider circuit. If there is no current through the output wire, then the current through the two resistances is the same, and the output voltage can be easily calculated by using ratios. Reminder: the triangle pointing down is the symbol that we use for *ground*, the node that we measure other node voltages from.

Note that if there is no current through the output wire, then the current through  $R_1$  must be the same as the current through  $R_2$  and we can compute the voltages across the two resistors:

$$\begin{aligned} V_{in} - V_{out} &= IR_2 \\ V_{out} - 0 &= IR_1 \end{aligned}$$

Adding the two equations, we can see that  $V_{in} = I(R_1 + R_2)$ , which gives us the rule for resistances in series: they add. We can solve for the current:  $I = \frac{V_{in}}{R_1 + R_2}$ , and plug that formula into the equation for  $V_{out}$  to get the formula for the measured voltage:

$$V_{out} = V_{in} \frac{R_1}{R_1 + R_2} ,$$

where  $R_1$  is the resistance across which the voltage  $V_{out}$  is measured,  $R_2$  is the other resistance, and  $V_{in}$  is the voltage across the two resistors in series.

For this lab, you will use a voltage divider to convert the temperature-varying resistance of the thermistor to a temperature-varying voltage that can be measured and recorded by a microcontroller.

If you want higher temperature to produce a higher voltage, which of  $R_1$  and  $R_2$  is the thermistor and which a fixed resistor?

### 2.3 Analog-to-Digital Converter

The microprocessors used in this course contain an *analog-to-digital converter* (*ADC*). The Freedom KL25Z boards have 16-bit ADC converters (range 0 to  $2^{16} - 1 = 65535$ ), while the Arduino boards we used previously had only a 10-bit resolution (that is, values from 0 to 1023). The Freedom board can measure voltages from any of several different pins (B0, B1, B2, B3, C0, C1, C2, D1, D5, D6, E20, E21, E22, E23, E29, E30) as *single-ended* inputs—that is the value reported is the difference between the voltage on the pin and the ground pin on the microcontroller. (The Arduino boards can measure voltages on one of the 6 analog input pins A0 through A6.)

The voltage on the analog pins must be limited to be within the power rails of the microprocessor. That is, on the KL25Z boards  $0 \leq V_A \leq 3.3v$  and on most Arduino boards  $0 \leq V_A \leq 5v$ . Putting negative voltages or larger positive voltages on the board may damage the chips on the board.

It is not possible to measure voltages simultaneously on different inputs, but the microprocessor can switch the ADC rapidly from one input to another, so that measurements can be made at *almost* the same time.

The ADC is *ratiometric*, which means that the output value is  $65536V_A/V_{Aref}$ , where  $V_A$  is the voltage on the analog input pin, and  $V_{Aref}$  is a reference voltage being compared to. The default reference voltage in the data acquisition software we are using is the power-supply voltage to the board, though it is possible to switch to an external reference. For this lab we'll only be using the power-supply reference.

We have provided data acquisition software that can be run on the Freedom KL25Z board or an Arduino connected with a USB cable to a larger computer. On the computers in the lab, this software and documentation should be installed in directory `C:\ProgramFiles\PteroDAQ\`

You can get a copy to run on your own computer at <http://bitbucket.org/abe.k/PteroDAQ/get/default.tar.bz2>

(extensions `.gz` and `.zip` are also available, if your machine does not understand bzip format). Note: this is the latest version and may be a beta release. You can find earlier release versions at <https://bitbucket.org/abe.k/PteroDAQ/downloads> under the “Tags” tab.

The software provides options for getting either the raw ADC readings (integer values from 0 to  $2^{16} - 1$ ) or converting to voltages. For this assignment, we are mainly interested in the ratio of the measured voltage to the reference voltage, rather than the voltage itself.

### 3 Pre-lab assignment

Read the Agilent 34401A User’s Guide to figure out how to use the meter to make resistance and voltage readings:

<http://cp.literature.agilent.com/litweb/pdf/34401-90004.pdf>

Before Wednesday (preferably before Tuesday), install gnuplot on your own computer from

<http://gnuplot.sourceforge.net/>

Installing on a Mac OS X system seems to still be a bit of a pain—for some suggestions, see the comments at

<http://gasstationwithoutpumps.wordpress.com/2012/01/02/installing-gnuplot-a-nightmare/>, particularly the comment

<http://gasstationwithoutpumps.wordpress.com/2012/01/02/installing-gnuplot-a-nightmare/#comment-6732>

You will need gnuplot for Wednesday’s class.

**Pre-lab Exercise:** for the circuit in Figure 2, what is the voltage  $V_{out}$ ?

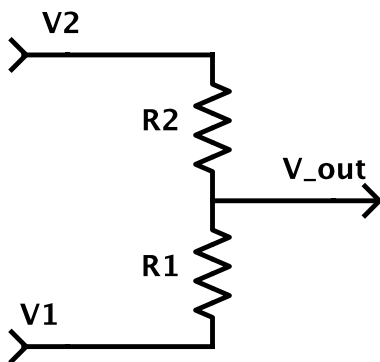


Figure 2: What is the voltage  $V_{out}$ ?

This lab consists mainly of measurements, but you do have one design task: to pick an appropriate resistor value for voltage divider. The goal is to have the maximum sensitivity of the output at particular temperature, which is a simple optimization problem. Before Wednesday, do the algebra to figure out the formula for the voltage ratio  $V_{out}/V_{ref}$  as a function of temperature, using the B equation in Section 2.1 and the voltage divider formula.

You will want to choose your resistor in your voltage divider to maximize the sensitivity (the change in voltage for a change in temperature) for an arbitrary operating temperature  $T_{op}$ . Note that you are not trying to maximize the voltage, but the derivative of the voltage with respect to temperature. Feel free to use symbolic tools like Mathematica or Wolfram Alpha, but be sure to write up the formula you are optimizing and how you optimized it. You should end up with a formula for the resistance  $R$  in terms of  $T_{op}$  and the parameters for the thermistor. Have this ready to be checked on Wednesday, but turn in the write-up of the optimization as part of design report on Friday.

Note that design reports do not consist just of the final design, but include the assumptions and thinking that lead to that design, so that an engineer reading the report can figure out why the design is the way it is, and what they can change to adapt it to a different need.

## 4 Parts, tools, and equipment needed

For the first lab, we will be issuing the parts and tool kit needed for the labs for the rest of the quarter.

### Parts for this lab from kit:

- NTCLE413E2103F520L thermistor
- 4 alligator clips
- resistors (for Thursday lab)
- wire (available in lab)
- breadboard (optional)

### Tools for this lab:

- wire cutters
- wire strippers
- screwdriver (for alligator clips)
- thermometer

### Equipment in lab:

- small cups for water baths
- secondary containment tubs to prevent spills
- thermoses with hot and cold water
- ice bucket
- hot pot to boil water
- multimeter (for measuring resistance)

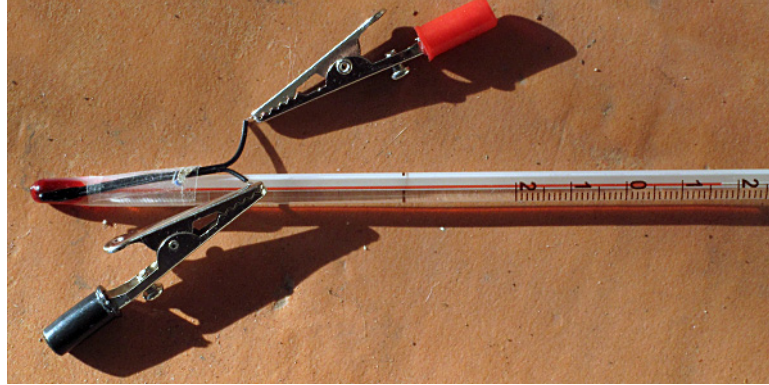


Figure 3: Thermistor with leads split for about 1cm, taped to a thermometer so that it is in contact with the bulb. The alligator clips are not attached to wires in this picture—that wiring should be done before the clips are attached to the thermistor.

## 5 Procedures

### 5.1 Sorting parts

For the first part of the lab, students will be learning to identify the parts in the parts kit, and doing some organizational work that will save time later on (like writing the value of the capacitors with a Sharpie on all the bags of capacitors).

We'll also make some clip leads and a power-supply lead

To connect up to the thermistor, it will be handy to have clip leads that can attach to the rather small and fragile wires on the thermistor. Cut a couple of pieces of wire about 2 feet (60cm) long. Strip the insulation off the last 5–6mm of each end, and tighten the screw on the alligator clip to hold the wire.

Later in the quarter we'll need power-supply wires to connect from the bench power supply to various breadboard circuits. Rather than have students cut new wires each time (at 10 cents a foot, the wire costs add up pretty fast), each student should cut a 4-foot length of red wire and 4-foot length of black wire and twist them to make a “twisted-pair”. It is conventional in digital electronics to use black for the “ground” (0v) wire and red for the positive (5v or 3.3v) supply wire. We won't need the this power-supply wire for the first lab, but it should be stowed with the parts and used for the rest of the quarter.

### 5.2 Setting up

The thermistor has two leads that are very close together, which can be handy if you are soldering the thermistor to a printed circuit board, but which is inconvenient for this lab. Separate the two leads about 1cm by pulling the wires apart. It may be necessary to make a small cut between the two wires at the end, to start the split. Although a pocket knife or razor is good for this initial cut, it should be doable with the diagonal cutters in the toolkit.

We'll provide water at three or more temperatures (near boiling, ice water, and roughly room temperature), which you can mix to get other temperatures. You'll make your water bath in coffee cups. Because water and electronics don't mix well, you must keep the coffee cup inside the secondary containment tub at all times.

In order to calibrate your thermistor, you will need to measure its temperature and resistance simultaneously. To measure the temperature, we'll use a liquid-filled student thermometer calibrated in °C.

Fastening the thermometer and the thermistor together so that the bulbs are in contact will minimize the temperature difference between them.

### 5.3 Measuring resistance

You will need to use the bench multimeter to measure the resistance of the thermistor. Make sure that the probes are plugged into the high and low jacks labeled for voltage and resistance, not current. By convention, black probes are used for the “ground” and red probes for measured voltage. Although this polarity does not matter for measuring the thermistor resistance, it is good to get in the habit of connecting the probes up correctly.

You will need to select the resistance measurement (the button labeled  $\Omega$ ) and automatic continuous measurement. If the “MAN” for manual or “TRIG” for trigger lights are on on the display, try fiddling with the buttons (shift-trigger may help). Better would be to read the Agilent 34401A User’s Guide <http://cp.literature.agilent.com/litweb/pdf/34401-90004.pdf>

Depending on what sort of probe is connected to the multimeter, you may be able to connect them directly to the thermistor wires, or you may need to use alligator clip leads to connect them.

For making precise measurements, it helps to subtract off the measurement of the wires connecting up to the device being measured, by shorting them together and recording that measurement.

There is another technique that does an even better job of compensating for wiring resistance: using 4-wire measurement and Kelvin clips. For an explanation, see [http://www.allaboutcircuits.com/vol\\_1/chpt\\_8/9.html](http://www.allaboutcircuits.com/vol_1/chpt_8/9.html)

Although the Agilent 34401A is capable of 4-wire measurement, we do not have the Kelvin clip probes for simultaneously measuring current and voltage, so this technique is not available to us. In any case, the resistances we’re looking at on the thermistors are large enough compared to the wire resistance that we don’t really need the extra accuracy obtainable with 4-wire measurement.

You will want to measure the temperature and resistance of the thermistor at the same time (since the uninsulated water baths will equilibrate to room temperature fairly quickly). Having one person hold and read the temperature, while the other records the temperature and resistance makes the recording easier. You can either record the measurements in a lab notebook, or type them directly into a computer file. You will need to have them in a computer file before Wednesday’s class.

The computer file should start with a number of comment lines, giving the names of the people making the measurements, the date, the part being measured, and the headings for columns (the data acquisition software records the data automatically, but you need to enter the the rest of the information as “notes”) Here is an example of the sort of heading I have in mind:

```
# Data collected Sat 23 Jun 2012
# by Kevin Karplus
# calibration for Vishay BC Components NTCLE413E2103F520L thermistor
# using Fluke 8060A multimeter
# degrees_F      kohm
192      1.30
190      1.37
186      1.452
180      1.65
176      1.718
...
```

Note: your measurements will be in  $^{\circ}\text{C}$  not  $^{\circ}\text{F}$ , and you will be measuring with a different meter. The “#” characters are there to hide the comments from gnuplot, which you can use for plotting and for fitting parameters of models to the data.

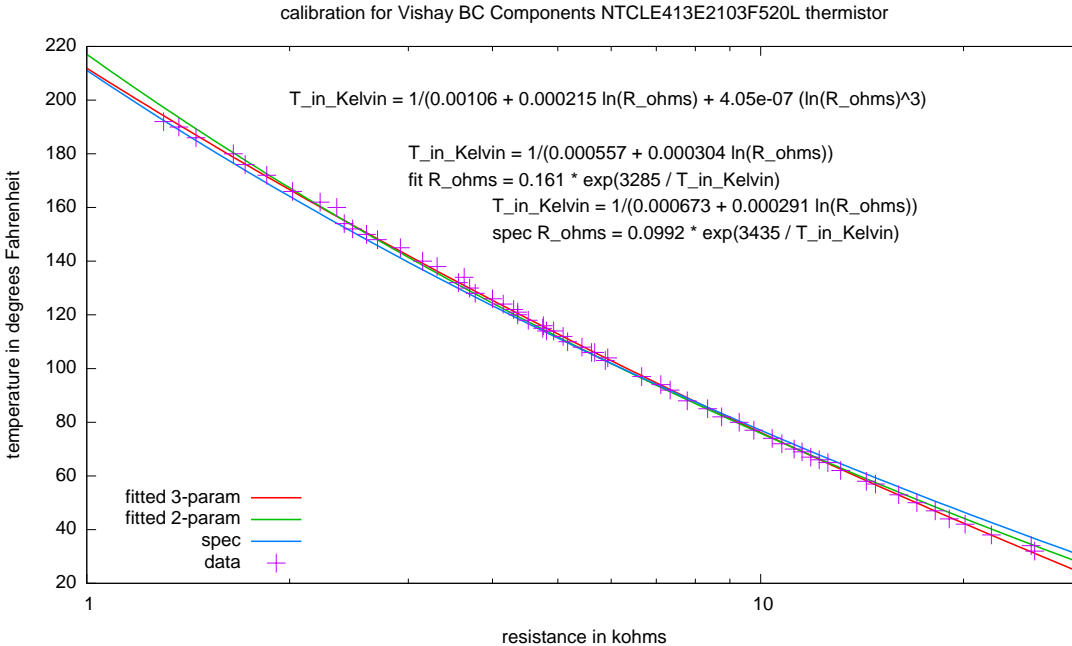


Figure 4: Plot produced from measurements with a Fahrenheit thermometer. Your plot should use Celsius measurements.

## 5.4 Fitting parameters with gnuplot

Gnuplot is a handy, free program for plotting data and fitting models. There are many other programs available with similar capabilities (some free, some cheap, some expensive), but gnuplot has been around for a long time and is likely to continue to be available for a long time to come.

Gnuplot can be used interactively to explore data, but can also be used with a script file that does everything consistently. I find it more useful, usually, to use a script file from the beginning, and edit it to make changes, rather than trying to remember all the commands I have typed in.

Because this is the first time most of you have used gnuplot, we'll spend Wednesday's lecture developing a gnuplot script to produce plots like the the one in Figure 4, which fits the Steinhart-Hart model and a simpler 2-parameter model to the data. Bring a computer with gnuplot installed and your data already in a computer file to Wednesday's class, so that you can work along with the class discussion.

Before fitting the data, we'll first see how to plot functions (like the B equation), then how to plot data, and finally how to fit a function to the data.

We'll plot and fit the data with temperature as a function of resistance, because the Steinhart-Hart equation is simpler in that form than as resistance as a function of temperature (as one might wish to plot it for a physics experiment, as temperature is clearly the independent variable). The T vs. R plot is also the more useful plot for a calibration curve for using the thermistor later, where we want to change a measured resistance into the appropriate temperature value.

You need to plot temperature in  $^{\circ}\text{C}$  as a function of the voltage  $V_{out}$  for your thermistor and resistor voltage divider circuit. You may wish to do two plots: for the voltage ratio  $V_{out}/V_{in}$  and for  $V_{in} = 3.3v$ .

In future assignments, you'll have to do more of the scripting from scratch, so learn what each of the commands in the script do. Gnuplot has a `help` command that you can use interactively to look at the documentation, and there is on-line documentation at <http://www.gnuplot.info/documentation.html>



## 5.5 Measuring voltage with a multimeter

Choose an appropriate series resistor for the operating point  $T_{op}$  specified during the lab. Be sure to record  $T_{op}$ , the resistance chosen, and the schematic diagram for your circuit in your lab notebook.

You can hook up your series resistor and thermistor either using the alligator clip leads or using the breadboard. For a single connection like this, the breadboard does not offer any advantage, but for future, more complicated circuits you will find that the breadboard simplifies quick wiring and is more reliable than clip leads (though much less reliable than soldered connections).

The adjustable 6v outputs on the Agilent E3631A power supplies will provide your input voltage divider. Before hooking up your resistor and thermistor, figure out how to set the voltage to 3.3v with a current limit of 0.1A (100mA) and measure the voltage of the power supply with the multimeter. You should always set the voltage and current limits of the power supply before turning on the output of the supply—you do not want to damage your circuits or blow fuses in the multimeters by unintentionally providing a large current.

Note that the “output on/off” button should be used to turn the power on and off to your circuit, without having to power down the Agilent E3631A itself. Turning off the power completely means having to reset the voltage and current limits when you turn it back on. These power supplies are high-quality ones and should provide very accurate voltage sources. Once you’ve confirmed that the output voltage is correctly set on the binding posts that you expect, you can rely on the power supply’s reporting of the voltage.

The color coding of the power supply terminals, with black for ground and red for the positive voltage, is standard (on the  $\pm 25$ v supply they also use red for the negative supply, with black reserved for the common ground). Please use red wires for your positive power-supply and black wires for your ground throughout this course (and don’t use those colors for other signals). Following this convention will make your circuits much easier to check and debug.

Measure the temperature and voltage for a few different temperatures, both close to and far from  $T_{op}$ . Verify that the voltages are what you expect from your previous calibrations.

## 5.6 Soldering headers on KL25Z boards

The Freedom KL25Z boards do not come with female headers attached, so we need to solder some on, to make connecting to your circuits easier. Illustrated instructions for doing this can be found at <http://gasstationwithoutpumps.wordpress.com/2014/02/23/soldering-headers-on-a-freedom-board/>

There are many learn-to-solder videos and tutorials on the web—if you’ve never soldered before, it might worth looking at a few of them.

## 5.7 Recording voltage measurements

We are providing some data acquisition software to use with the Freedom KL25Z and Arduino boards. This program was written by a high-school student, and he welcomes feedback. The program and documentation are available from the BitBucket repository:

[http://bitbucket.org/abe\\_k/PteroDAQ/get/default.tar.bz2](http://bitbucket.org/abe_k/PteroDAQ/get/default.tar.bz2)

(extensions tar.gz and .zip are also available, if your machine does not understand bzip format).

The software consists of two parts: a small program that runs on the microprocessor board and a Python program that runs on a laptop or desktop machine. The microprocessor code makes measurements and communicates them over the USB connection to the Python program on host computer, which provides a graphical user interface, file I/O, and configuration information.

To use the data acquisition software, you need to download the microprocessor program to the board. This only needs to be done once, as the program is stored in flash memory and will remain on the board

until another program is downloaded to replace it. The download procedure is different on the KL25Z boards and the Arduino boards.

### 5.7.1 Downloading to KL25Z boards

Connect the KL25Z board to a USB port on your computer using the USB connector labeled “SDA” on the KL25Z board. The KL25Z board should appear as a flash drive on the computer. You can download the data acquisition software by dragging the `kl25_daq.bin` file to this “flash drive”.

If you are using a Mac, Linux, or a Windows 8 machine, the download will probably not work, because the firmware written by P&Micro for the OpenSDA chip on the KL25Z board was not properly tested, and only works with a few versions of Windows. (They claim that newer versions of the P&E Micro firmware fixes the problem, but I’ve not been able to verify that—the KL25Z boards still seem to be shipping with old firmware.)

To download software from a Mac, you first need to download different firmware using a Windows machine like the ones in the lab. I recommend the MBED firmware from [mbed.org](http://mbed.org/handbook/Firmware-FRDM-KL25Z): <http://mbed.org/handbook/Firmware-FRDM-KL25Z> has the instructions and download link for the firmware. Once you have the MBED firmware installed, you should be able to download the `kl25_daq.bin` program using almost any computer.

### 5.7.2 Downloading to Arduino boards

You must have the Arduino development software installed to download to Arduinos, as the program is recompiled from source code every time you do a download.

To download the program to an Arduino, connect the Arduino to the computer with a USB cable, and open the `arduino_daq.ino` file in the Arduino environment. On the lab computers, this program should be in

```
C:\ProgramFiles\DAQ\arduino_daq\
```

Set the Arduino board type and serial port in the “Tools” menu, and download `arduino_daq.ino` to the Arduino. Once the program is downloaded, the Arduino will retain it in flash memory, even if power is removed. The Arduino environment can be closed after the download is done, as it is not needed again.

### 5.7.3 Running the Data Acquisition System

Note that you will not be using the bench-top power supply, but supplying power from the microprocessor board. Because the analog inputs have to stay in the range 0 to 3.3v of the KL25Z (0 to 5v on most Arduino boards), using power from the board makes it much less likely that you will accidentally damage your microprocessor board by providing an out-of-range voltage.

Hook up 3 wires between your microprocessor and your thermistor circuit: GND to GND,  $V_{in}$  to 3.3v (5v on Arduino boards), and  $V_{out}$  to one of the pins that you can measure analog inputs on (say B0 on the KL25Z board or A0 on an Arduino).

The python program is started with `python C:\ProgramFiles\PteroDAQ\daq` (or whatever directory the program is installed in). Python looks for `__main__.py` in that directory.

The first thing to do is to configure what information you want recorded and when. For example, you may want to record the analog input every 200 msec, and you may want to scale it so that 1.0 is full-scale (rather than actual voltage or raw readings).

You also have to decide what reference you will use for the analog-to-digital conversion. Since the A-to-D is ratiometric, and we are interested in measuring  $V_{out}/V_{in}$ , it is probably easiest to use the default reference (the 3.3v or 5V supply on the microprocessor board). It is possible to use an external analog

voltage reference, as long as both the reference and the analog input stay within legal ranges, but that is not recommended for this lab.

Record a few minutes worth of data (say of hot water cooling down towards room temperature) and save it in a file. You should be able to plot the file easily with gnuplot, though getting proper scaling and labeling of the plot may take some effort. You want to plot temperature (in °C) on the y-axis vs. time (in seconds) on the x-axis.

## 6 Demo and writeup

In the lab, the students need to show one of the instructors that they can correctly measure resistance and temperature, fit parameters of a model to the data, select an appropriate resistor for a given  $T_{op}$ , correctly measure voltage from a voltage divider, and record a time series of measurements with the data acquisition system.

A careful writeup of the lab is due on Friday, the day after the lab.

In addition to all the usual stuff described in the lab report guidelines, the lab report should contain at least the following (though not necessarily in this order):

- A statement of the problem.
- Any pre-lab analysis and design that was done.
- A plot of measurements made and any modeling done (for example, the temperature vs. resistance calibration curve in Figure 4). The plot should use the units measured ( $\Omega$  or  $k\Omega$  and °C, for example).
- A schematic for the voltage divider, including part number for the thermistor and value for the resistor.
- A calibration plot of temperature vs. voltage (or voltage ratio) for the voltage divider, showing both measured data and the model built using the equations for thermistor behavior and voltage dividers. Note that this plot will require you to do some scripting in gnuplot or a similar plotting tool.
- A plot of temperature vs. time from the output recording. Again, this plot will require some scripting.