mbed Lab 2: I'm Sensing a Change in Temperature

CS 220

Introduction

No computer is really that useful unless it can interact with the world around it. In addition, quantities such as temperature are analog, whereas computers are inherently digital. In this lab, you'll use the mbed's `Serial` interface to facilitate communication between the KL25Z board and a serial terminal program running on the host computer. You'll use the mbed's `AnalogIn` interface to read temperature values encoded as analog voltages from an analog temperature sensor, the Analog Devices TMP36.

Lab Objectives

1. Use the mbed's `Serial` interface to display text on a terminal program running on the host computer.

2. Use the mbed's `AnalogIn` interface to read analog values from a TMP36 temperature sensor, converting them to Fahrenheit temperatures and displaying the temperature continuously.

3. Use a few techniques for cleaning-up noisy analog signals in order to present a more stable temperature reading.

4. Develop an appreciation for why analog computers are exceedingly rare.

Lab Parts List

1. KL25Z board and Quick Reference Card

2. USB cable

3. Breadboard

4. Analog Devices TMP36 analog temperature sensor.

5. Two 10.0μF capacitors, blue.

6. One 750Ω resistor, tan with colored stripes

7. Four wires

The temperature sensor, capacitors, and resistor are in the static shielding bag.

Lab Procedure

Read each of the following steps **completely** before acting on them.

1. You'll be using a terminal program to display output from the mbed board on your computer. Let's prepare for that before moving on. You won't actually need to use the terminal program until later on during the lab.

   If you're using your personal Windows PC, follow this link (`https://ttssh2.osdn.jp/index.html.en`) to download and install `Tera Term`.

   Windows and Mac users, refer to this document (`http://phoenix.goucher.edu/~kelliher/f2015/cs220/terminalApplications.pdf`) for using a terminal program with the mbed board's `Serial` interface.

2. Visit the `developer.mbed.org` website and sign-in.

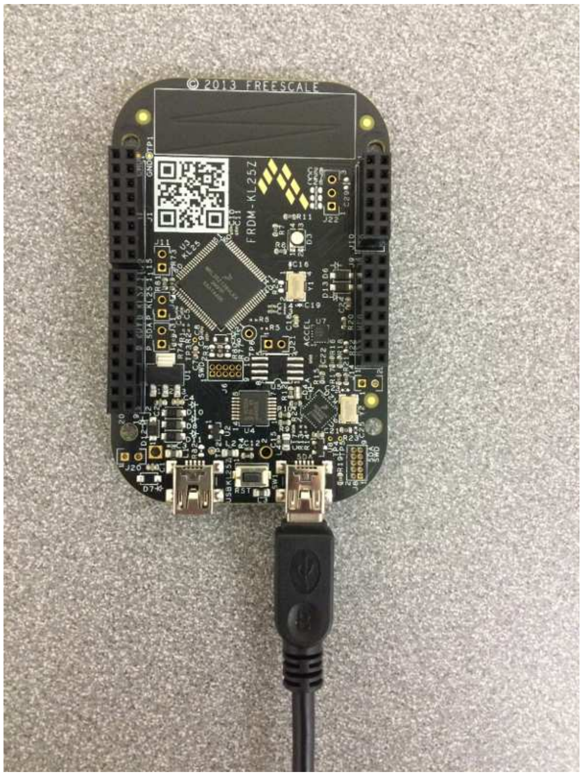
3. The documentation for the `Serial` interface is here (`https://os.mbed.com/docs/v5.6/reference/serial.html`). Visit this page.

4. Import the Serial Hello World program. When the *Import Program* dialog box opens, check the update box so that you pick up the most recent versions of any library files.

5. Open the program in the development IDE. As a warm-up exercise, modify the program by replacing the body of the `while` loop with a body that prints floating point values to two decimal places, starting at `0.00` and incrementing by `0.01`. Print a new value every 0.25 sec. (Use the `wait()` function.) The following is an example showing how to print the `float` variable `x` to two decimal places:

   ```
   float x = 0.0;
   ...
   pc.printf("%.2f\r\n", x);
   ```

6. Connect your KL25Z to your computer and compile and download your program. Remember to connect the USB cable to the KL25Z like so:



7. Open your terminal application. Reset the KL25Z and output should begin to appear on the terminal.

8. Okay, it's time to dust-off your algebra skills. You're going to have to develop three linear equations and then compose them into a single linear equation.

   (a) The first equation relates the analog value read from an `AnalogIn` interface to the actual sampled voltage. A read operation performed on an `AnalogIn` will return a `float` value in the range $[0.0, 1.0]$. This value is a scaling factor. The scaling factor multiplied by the interface's reference voltage yields the sampled voltage.

   The `AnalogIn` interface's reference voltage is 3.3V. Derive the equation relating the sampled voltage to the value read from the `AnalogIn` interface:

(b) The TMP36 sensor is calibrated to indicate temperatures in degrees Celsius from −40°C to 125°C. To be capable of indicating negative temperatures, all temperature readings are biased by +0.5V. Therefore, a sensor reading of 0.5V corresponds to a temperature of 0°C. According to the TMP36's documentation, a reading of 0.75V corresponds to a temperature of 25°C.

Derive the equation relating the temperature in degrees Celsius to the sampled voltage computed in the previous step.

(c) Finally, derive the equation relating the temperature in degrees Fahrenheit to the temperature in degrees Celsius (It's okay to look this equation up on the Internet.):

(d) The TMP36's output pin will be connected to an `AnalogIn` interface on the KL25Z. Composing the three equations that you just wrote, write the equation relating the temperature in degrees Fahrenheit to the value read from the `AnalogIn` interface:

Check your work: If the reading from the `AnalogIn` interface is 0.22391, the temperature is approximately 75°F.

9. The documentation for the `AnalogIn` interface is here (`https://os.mbed.com/docs/v5.6/reference/analogin.html`). Back in the mbed IDE, add an `AnalogIn` interface to your program, configuring it for KL25Z pin `PTC2`. Only a few of the KL25Z's pins can be used with the `Analogin` interface, because the interface makes use of an analog-to-digital converter circuit (If you are interested, you can check Wikipedia to learn about this circuit). This `AnalogIn` interface will be connected to the TMP36 temperature sensor.

10. Modify the body of your program's `while` loop to print a degrees Fahrenheit temperature reading once a second. In addition to the current temperature reading, also print the minimum and maximum temperatures and the difference between them; similar to this:

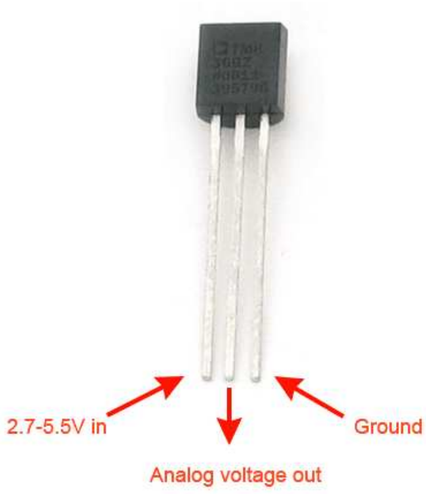
    ```
     9 72.03 [72.00, 72.15] <0.15>
    10 72.20 [72.00, 72.20] <0.20>
    ```

    A statement similar to this one should work:

    ```
    pc.printf("%3d %.2f [%.2f, %.2f] <%.2f>\r\n", i, temp, min, max, max-min);
    ```

    where `i` is the iteration count. Download the program onto the KL25Z.

11. Now it's time to connect the TMP36 to the KL25Z. Remove the TMP36 and one of the 10.0μF capacitors (small, blue body) from the anti-static bag. Leave the second capacitor and the 750Ω resistor (striped tan body) in the anti-static bag for now. The TMP36 looks like this:

    In this figure, note that the TMP36's flat face is facing you. The pins are numbered, from the left, one through three. I.e., pin one is the V_{in} pin. If you reverse the power and ground connections to the TMP36, when power is applied the sensor will quickly overheat and then die a horrible death. **Don't let the sensor suffer this fate!**

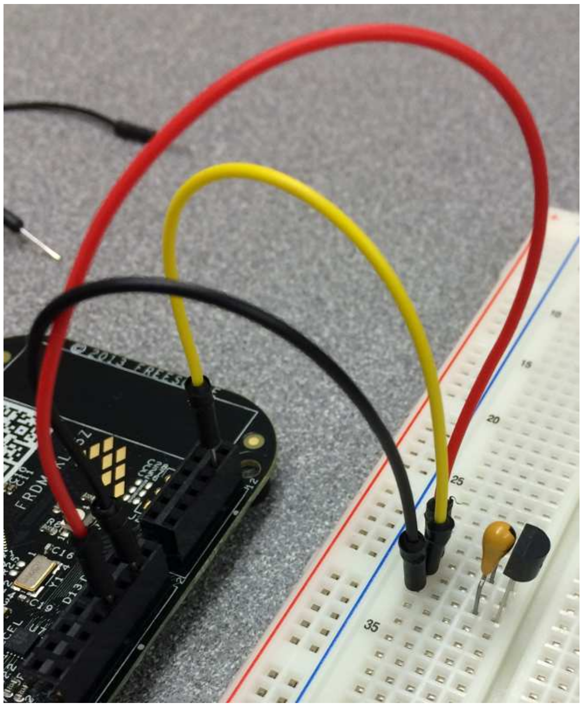
12. Grab you breadboard and insert the TMP36's three pins into three adjacent rows (each pin **must** be in its own numbered row), with the sensor's flat face facing you. Use breadboard

rows that don't already have other components in them. Rows 42 through 58 on the left-hand side of the board are a good choice. Connect the capacitor so that one of its pins is in the same row as pin one of the TMP36 and the other pin is in the same row as pin three of the TMP36.

13. For the following wiring steps, use the KL25Z Quick Reference Card to locate KL25Z pins.

14. Remove power from the KL25Z. Grab a wire and connect pin one of the TMP36 to the `P5v_USB` pin across from pin `PTE2` on the KL25Z. Using a second wire, connect pin three of the TMP36 to the `GND` pin next to pin `PTE4`. Finally, using a third wire, connect pin two of the TMP36 to pin `PTC2` of the KL25Z.

    When finished, your wiring should be similar to this:

    (I used a tan-colored capacitor; remember, your capacitors are blue.) Leave an air gap between the TMP36 and the capacitor; otherwise the capacitor might heat the sensor, leading to incorrect readings.

15. Double-check your wiring. Triple-check with me if you want to be absolutely certain that everything's right.

16. Re-apply power to the KL25Z. If necessary, re-connect the terminal program. The temperature should now be displayed on the terminal program. Take 60 temperature readings, noting the difference between the minimum and maximum readings. Record your results.

17. You will likely notice that the temperature display isn't very steady. Let's fix that. Modify your program so that several sample temperatures are read and averaged, with the average temperature printed and used to determine the ongoing maximum and minimum temperature readings. Try taking 3 samples and then 16 samples. The samples should be taken 1 ms. (0.001 sec.) apart. Again, run your program long enough to print 60 readings.

    You'll find it useful to use a `#define` constant at the top of your source file

    ```
    #define SAMPLES 3
    ```

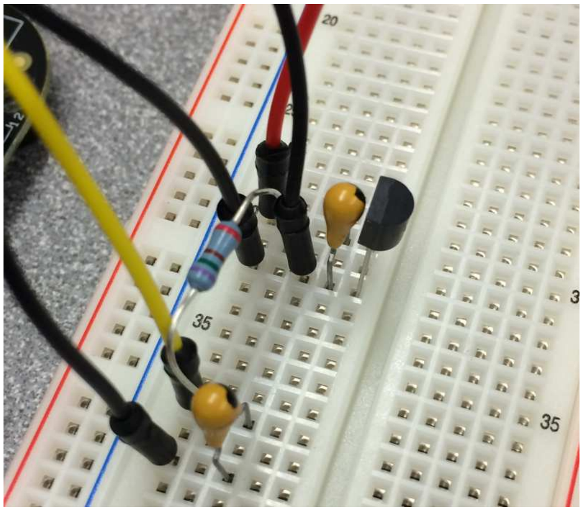
    to make it easy to change the number of samples between each run of the program. Record and compare your results.

18. Next, we'll condition the output of the TMP36 by sending its signal through a passive low-pass filter. You can look this up in Wikipedia if you are interested, but we are going to treat this as a black box for the purpose of this lab. This filter should clean up our messy analog input. Let's give it a try.

19. Remove power from the KLZ25. Disconnect the jumper wire connected to pin 2 of the TMP36 (the output pin). Leave the KLZ25 end of the jumper wire connected.

20. Remove the resistor and second capacitor from the anti-static bag. Connect one end of the resistor to pin 2 of the TMP36 and the other end to an open breadboard row. Reconnect the jumper wire to this row. Connect one end of the capacitor to this row and the other end to another free row. Finally, take another jumper wire and use it to jump the TMP36's ground pin (pin 3) and the unconnected end of the capacitor (the end that's in a row all by its lonesome).

21. Your circuit should now be similar to this:



22. Double-check your wiring. Triple-check if you want to be absolutely certain that everything's right.

23. Re-apply power to the KL25Z. Re-connect the terminal program. Repeat the experiment, taking 60 temperature readings using 1, 3, and 16 samples. Record and compare your results.

24. To finish-up, disconnect both ends of the USB cable. You'll be using the temperature sensor circuit again in the next lab, so carefully transfer the KL25Z board and the breadboard to the storage box, put everything else back into the storage box, and carefully close its lid.