

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

CS 220

Oct. 30

## 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 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 Procedure

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

1. Visit the `developer.mbed.org` website and sign-in.
2. Find the mbed Handbook page and scroll down to the `Serial` interface and visit its page.
3. Import the `Serial_HelloWorld_Mbed` program.
4. Open the program in the development IDE. 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. The following is an example showing how to print the `float` variable `x` to two decimal places:

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

Note that programmers traditionally use `i`, `j`, and `k` for single letter integer variables and `x`, `y`, and `z` for single letter floating point variables. For readability purposes, floating point constants are written with a decimal point:

```
float x = 0.0;
```

5. Connect your KL25Z to your workstation and download your program.
6. Open the Tera Term terminal application. In the New Connection window, select the Serial connection and set the port to the `mbed Serial Port`. Reset the KL25Z and output should begin to appear on the terminal.

Note that each time you download a new program onto the KL25Z, the terminal connection is broken. To re-establish the connection, you'll need to open the File menu, disconnect, and then make a New Connection. Yeah, it's a bit of a hassle, but you'll live.

7. 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 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 times the interface's reference voltage yields the sampled voltage.

The `AnalogIn` interface's reference voltage is 3.3V. Write 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  $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . 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  $0^{\circ}\text{C}$ . According to the TMP36's datasheet, a reading of 0.75V corresponds to a temperature of  $25^{\circ}\text{C}$ .

Write the equation relating the temperature in degrees Celsius to the voltage read from

the TMP36 sensor:

(c) Write 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. **If you took MA 160 from me, you had better get this right the first time!!!**

8. 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 (check Wikipedia to learn about this circuit). This `AnalogIn` interface will be connected to the TMP36 temperature sensor.

9. 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:

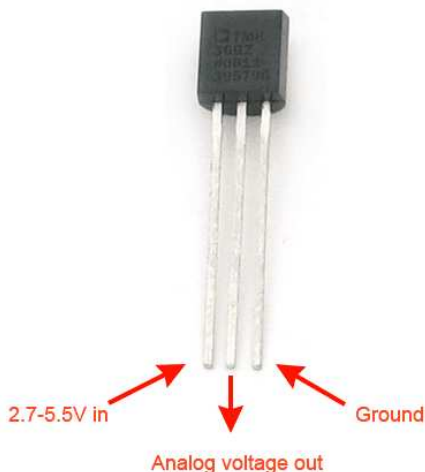
```
9 72.03 [72.00, 72.15] <0.15>
10 72.20 [72.00, 72.20] <0.20>
```

A statement similar to this one should do the trick:

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

Download the program onto the KL25Z.

10. Now it's time to connect the TMP36 to the KL25Z. Remove the TMP36 and one of the  $10.0\mu\text{F}$  capacitors (small, blue body) from the anti-static bag. Leave the second capacitor and the  $750\Omega$  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  $V_{\text{in}}$  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!**

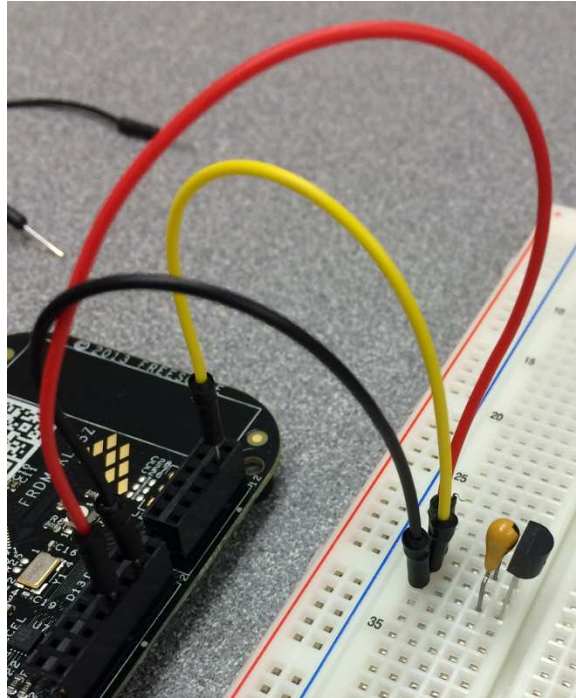
11. Grab your breadboard and insert the TMP36's three pins into three adjacent rows (each pin **must** be in its own numbered row), with the sensor's flat face facing you. Work with rows numbered 30 and higher. (Next week you'll be working with an LCD panel that you'll need rows 1 through 29 for.) Connect 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.

Challenge: What is the purpose of this capacitor?

12. For the following wiring steps, use the separate KL25Z pin diagram to locate KL25Z pins.

13. Remove power from the KL25Z. Using a red wire, connect pin one of the TMP36 to the P5v\_USB pin across from pin PTB9 on the KL25Z. Using a black wire, connect pin three of the TMP 36 to the GND pin across from 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.

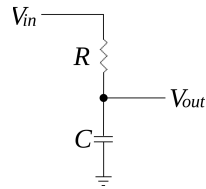
14. Double-check your wiring. Triple-check if you want to be absolutely certain that everything's right.
15. Re-apply power to the KL25Z. Re-connect the terminal program. The temperature should now be displayed on the terminal program. Take 60 temperature readings, noting the difference between the minimum and maximum readings. Record your results.
16. 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. Try taking 3 samples and then 16 samples, producing 60 temperature readings each time. The samples should be taken 1 ms. apart.

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

```
#define SAMPLES 3
```

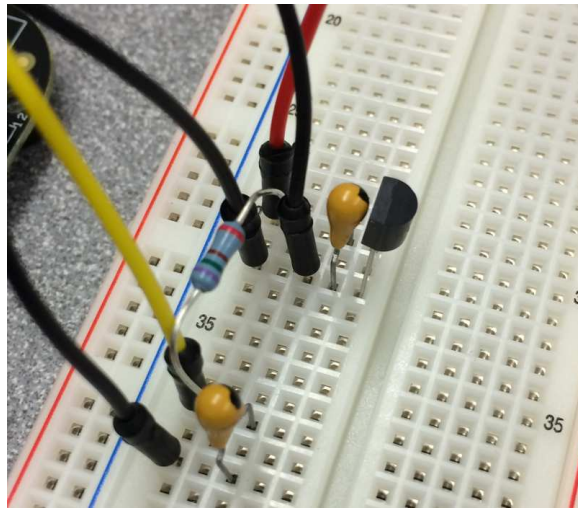
to make it easy to change the number of samples between each run of the program. Record and compare your results.

17. Next, we'll condition the output of the TMP36 by sending its signal through a passive low-pass filter (look it up in Wikipedia) to clean it up a bit. Here's a circuit diagram of a low-pass filter:



Challenge: How is this circuit similar to a voltage divider? How does it differ in a useful way?

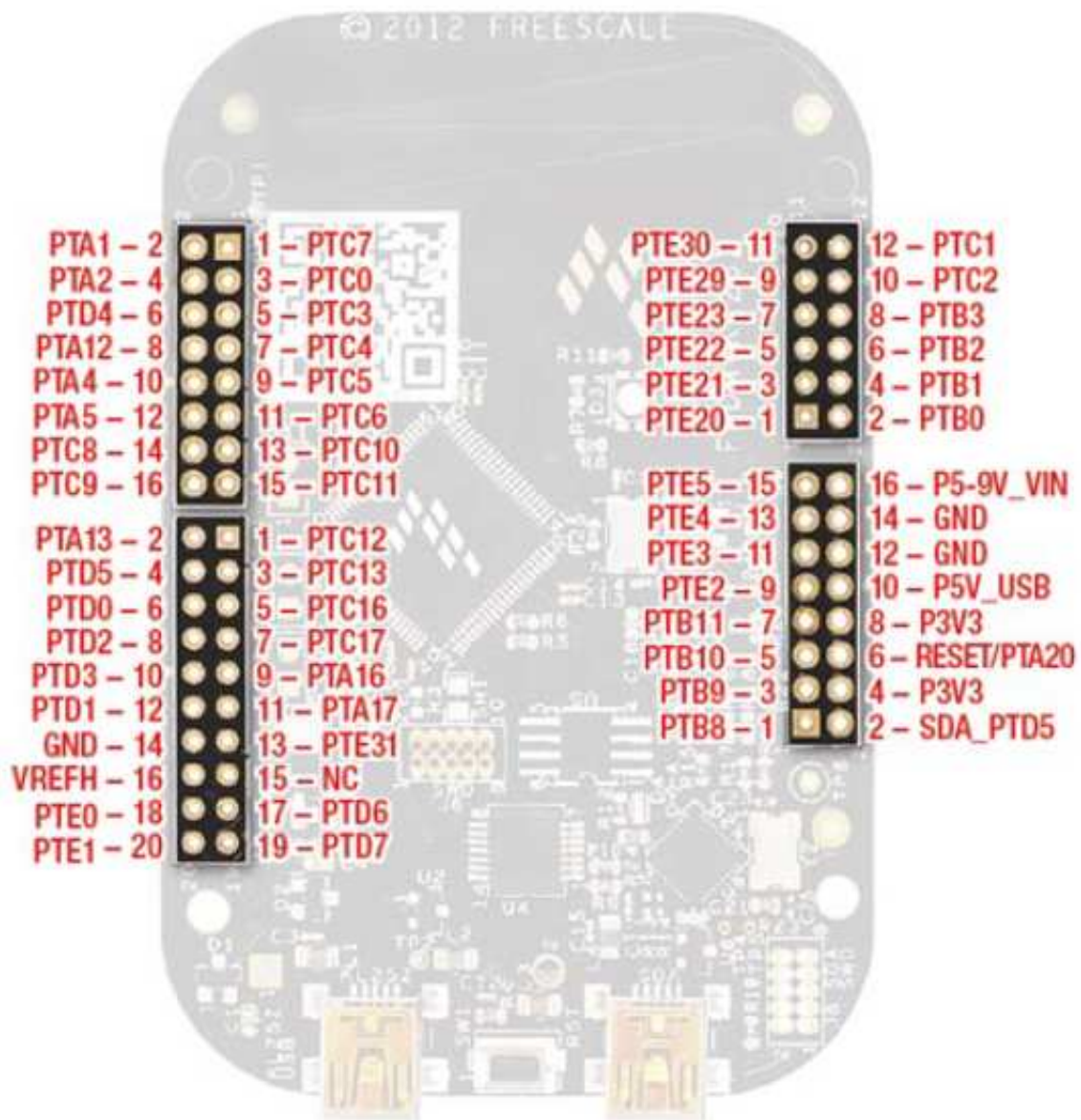
18. Remove power from the KLZ25. Disconnect the jumper wire connected to pin 2 of the TMP36 (the output pin). Leave the KLZ25 end of the jumper wire connected.
19. Remove the resistor and second capacitor from the anti-static bag. Connect one end of the resistor to pin 2 of the TMP36 and the other end 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 a black 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).
20. Your circuit should now be similar to this:



Challenge: What is the cutoff frequency of the low-pass filter that you just built?

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

# KL25Z Pin Diagram



USB connector-end of the KL25Z