Physics 222

**Lab Practicum: Making a colorful thermometer**

Objectives:

• Building a simulated circuit with a low-voltage analog sensor, calibrating and testing it

• Writing code to run the sensor, and to trigger actions at particular threshold temperatures

Introduction:

The Analog Devices TMP36 Temperature Sensor is the most common digital device used to measure the ambient temperature of a system. However, what the device *actually* outputs is a voltage. Fortunately, the output voltage is proportional to the temperature, so it’s a matter of calibrating the voltage to conform with the temperature.

Open the data sheet from the manufacturer, Analog Devices; fortunately, the University of Michigan Engineering Program has already scanned that in for you: <http://ctms.engin.umich.edu/CTMS/Content/Activities/TMP35_36_37.pdf> .

The sensor is shown in Figure 4. T-3 (TO-92).

Quoting from the data sheet: “The TMP36 is specified from −40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply… Both the TMP35 andTMP36 have an output scale factor of 10 mV/°C.”

From this information, derive the linear formula for the temperature sensor (e.g., y = mx + b form). Remember that the analog inputs of the Arduino convert a voltage input between 0 and 5 V into an integer between 0 and 1023.

Write **the Arduino code** that receives input from the TMP36 sensor and outputs the Celsius temperature to the serial monitor using the formula you determined above. Some tips:

• One particular line that will be helpful is **Serial.println(temp\_C);** which will cause a line space to appear between data entries.

• The analogRead command receives input from the sensor and converts it into a variable (note the sensor output is connected to Arduino analog port 0):

**float voltage = analogRead(A0);**

• Add a 1 second delay (**delay(1000)**) between readings and limit the number of readings, unless you want to be overwhelmed by data. Note the delay value is in milliseconds.

As an example of Arduino code, here is a program called “ReadAnalogVoltage” which causes the Arduino microprocessor to read an analog input from pin A0, convert that reading into a decimal voltage, then prints that voltage onto the Serial monitor.

Also note that this code is extensively documented (commented) – everything after the “//” in a line is ignored by the computer. I will be looking at your code’s documentation; basic rule: more explanatory comments = more points.

// the setup routine runs once when you press reset

void setup() {

// initialize serial communication at 9600 bits per second

Serial.begin(9600);

}

// the loop routine runs over and over again forever

void loop() {

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

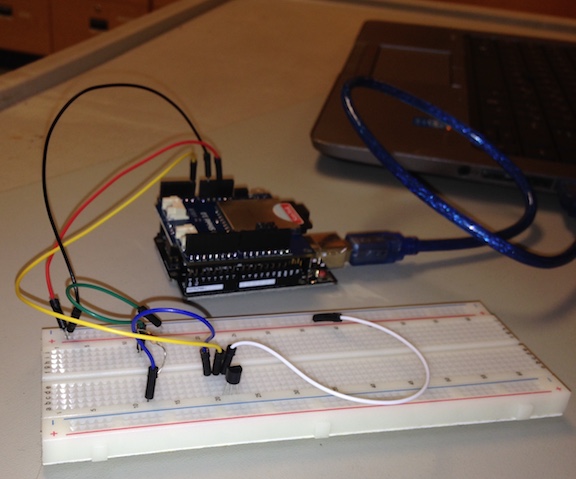
// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage = sensorValue \* (5.0 / 1023.0);

// print out the value you read on the Serial monitor:

Serial.println(voltage);

}

This is what the Arduino and breadboard look like when students set up this task during lab.

Note the numerous connections between the Arduino and the breadboard.

Circuit diagrams

As you build the hardware, take photos and notes, so that you will be able to draft a final circuit diagram for the presentation. Show the connections between the Arduino Board and the TMP36 sensor; this means drawing pin-to-pin lines (the sensor and the Arduino may be represented as squares, with labeled pins on each).

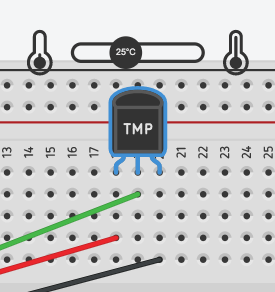
Safety

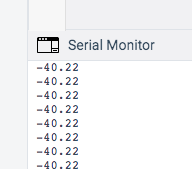
Had you been using an actual TMP36 and Arduino, you’d see that both the Arduino Board and the TMP36 sensor are easily damaged by static electricity. Furthermore, if the TMP36 sensor was incorrectly wired, it would have grown UNTOUCHABLY HOT QUICKLY. If this had occurred, you would have simply **cut the power** to the board.

Suggested procedure

1. On Tinkercad, select a small breadboard, an Arduino and a TMP36 device.

2. Connect the TMP36’s pin 1 to 5 V on your Arduino, the middle pin (Out) to A0 or some other analog input pin on your Arduino, and pin 3 to GND.

3. Enter the code shown in the Introduction section. If you have coded and wired everything correctly, a temperature slider should appear above the TMP36, as shown in the screenshot to the right.

4. To read your output, open the “Serial Monitor” whose icon is located at the bottom of the coding window. When the simulation is running, you should be able to see a running record of the temperature the sensor is experiencing; ideally, it will match the temperature you set the slider on! Also, unless you want a TON of output, throw in a delay(1000); statement.

5. If needed, modify the code to obtain more accurate temperatures from the sensor. Generally this means tweaking the numerical constants in your formula. Make sure you are getting accurate results before proceeding!

6. Modify the circuit so several LEDs are wired into circuit parallel to the temperature sensor (that is, the LEDs are not in series with the sensor; in fact, they should be deriving their power from one of the digital power pins like 9 or 13). Make sure each LED is in series with 220 ohm resistor so they don’t blow up.

7. Modify the Arduino code so that one color LED lights up if the sensor temperature is less than 10° C, a different color LED lights up when the temperature is between 10° C and 25° C, and yet another color LED lights up at temperatures more than 25°C. Remember to document (comment) all the critical lines in your code. For an added challenge, make use of more LEDs so you have the digital equivalent of an analog thermometer!

8. Make sure to take screenshots of the code and the circuit for inclusion in your presentation.

In your PowerPoint (or other presentation software) slide deck:

• Title slide, with the name of the project and the full names of the team members. Indicate affiliation with North Seattle College

• A goals slide – what did you hope to accomplish? How much time did you budget for the work? Who is going to do what on this project?

• A schematic slide, showing the circuit with **standard electrical symbols**. The Arduino and TMP36 may be represented as boxes with numbered or labeled pins emerging from them.

• A code slide, showing the actual code used to run the sensor, including extensive documentation

• A results slide – did your project work? What were the difficulties encountered, and how were they overcome, including any personnel issues? How much time DID you spend (everyone) on this project?

Either show a video or do a live demonstration of your color LED temperature sensor showing how it changes color from cold (0°C) to hot (50°C).