Summary:

Students will combine the skills they have learned over the semester to build a simple benchtop data logger.

Materials:

This activity will be done by each student. The student should have available the following materials, equipment, and supplies:

1. AD-820 operational amplifier integrated circuits in DIP-8 package (Or alternative Op Amp TLV271P)
2. Solderless breadboard
3. Electronics Components construct breadboarded temperature sensor
    a. 1x 1N457 PN Diode
    b. 4x 1KΩ Resistors
        i. 1x Current limiting resistor for diode
        ii. 3x Resistors for Amplifying Circuit
    c. 1x 10KΩ potentiometer for Amplifying Circuit
    d. 1x 5KΩ potentiometer for Amplifying Circuit
    e. 1x 5KΩ Resistor for Amplifying Circuit
    f. 1x 0.1 uF capacitor for Amplifying Circuit (Optional)
4. Small hookup wire, clip leads, and soldering tools
5. Multimeter (for measuring resistances and voltage)
6. Arduino Mega
7. Power Supply
8. Arduino programming cable
9. Heat shrink or liquid electrical tape
10. Water bath, ice, hotplate for calibration
11. Thermometer for calibration

The laboratory should also be equipped with the following:

1. Flat worktables sufficient to seat all students with plenty of workspace

Introduction:

In this activity, we are going to combine the skills we have learned in the previous lessons to build a simple datalogger that will record the temperature along with additional data from the GPS into a simple, realistic data recorded on the SD card using the Arduino.

This temperature sensor should have a minimum temperature of -50C or lower. A maximum temperature of +50C or higher. And resolution of 1C or better.

Be sure to document this process using photos and your lab notebook as you will be writing a report documenting the development of your data logger.

Part 1: Preparing the Temperature Sensor

We are going to use the diode as our temperature sensor. If you recall, when we first worked with the Arduino's ADC, we saw that the voltage drop across the diode would vary when we held the diode in our fingers. This is because the voltage will vary with the temperature of the diode.

We will calibrate our temperature sensor using a procedure like what we did with the SkeeterSat earlier in the semester. This means we need to waterproof the diode in a similar way that we did to the thermistor. So, cover it with either heat shrink or liquid electrical tape to waterproof it.

We also need to solder wires to the leads of the diode long enough to allow us to submerge it in a water bath. Unlike the thermistor, the diode is polarized. The diode stripe will mark the cathode which is the negative side of the diode. Since our heat shrink or liquid tape will cover these markings, it is a good idea to use two different color wires and record which one is connected to positive and negative. (If possible, use the standard red for positive and black for negative). When completed, your diode should look something like the one pictured in Figure 1.

Figure 1 Soldered and insulated diode for temperature sensor. Note bare wire ends may be used instead of crimped connection

Part 2: The Temperature sensor circuit:

The circuit for the temperature sensor is simple. We just need to connect a voltage source to the diode in series with a resistor to limit the current. The value for that resistor is not important, so let's just use a common value from our parts kit, like 1kΩ but any value from 1-10 kΩ would also work.

We connect the resistor to the positive supply to use the voltage on the positive side of the diode as our signal and connect the negative to our common ground for the circuit. For the diode, the voltage supplied is not critical, but we will want to use the same supply for our amplifying circuit. For that reason supply ~12V if possible, but if no other supply is available, use the 5V pin from the Arduino.

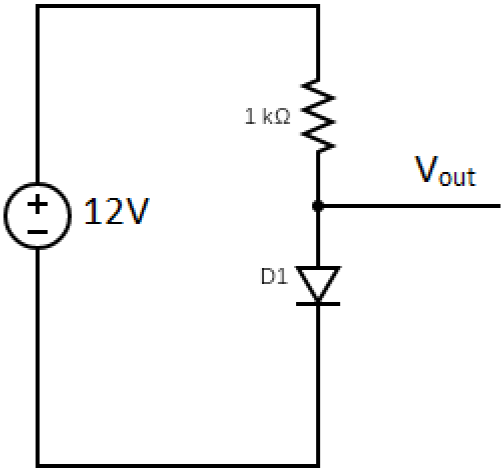

Figure 2 Setup for using diode as a temperature probe

The output signal (V_{OUT} in Figure 2) will connect to the input of the amplifying circuit (V_{INPUT} on Figure4).

Part 3: The signal conditioning circuit:

We want to get the best sensor resolution possible and match the output of our amplified signal to the range of the Arduino ADC (0-5V).

The typical range for the diode used as a temperature sensor is 450 mV, and the maximum is 800 mV. This is a good starting point, but we should expect different output from each diode.

The base is 450 mV, and the span is just 350 mV. So, if we plot the values for .45 to .8 V input versus the 0 to 5V output we get the line shown in figure 3.

The graph in **Figure 3** represents a transfer function between the output and the input of the signal conditioning circuit, given by the equation of the straight line:

Eq. 1 $V_{\mathrm{OUT}} = 14.286V_{\mathrm{IN}} - 6.4286$

From this equation, we can determine what kind of amplifying circuit. We have a gain > 1, meaning we need a non-inverting amplifier.

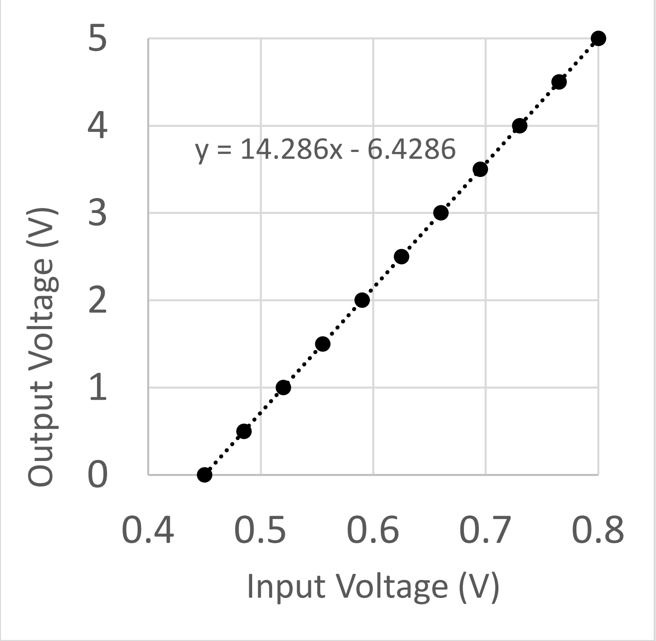

Figure 3: Linear plot of expected diode output range to ADC input range

We can also see that we will need an offset.

From those 2 pieces of information, we need a non-inverting amplifier with offset. Looking up how to build one we will find a schematic like the one shown in Figure 4.

The equation for the output of the circuit is:

Eq. 2 $$V_{\mathrm{OUTPUT}} = V_{\mathrm{INPUT}}\left[1 + R_3\left(\frac{1}{R_2} + \frac{1}{R_4}\right)\right] - V_{\mathrm{REF}}\left[\frac{R_3}{R_2}\right]$$

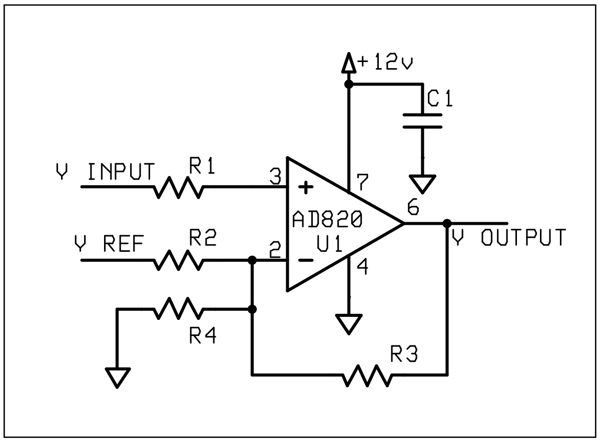

Figure 4 Noninverting Amplifier with offset

Now we need to determine the values for the components in our circuit. Remember, V_{INPUT} is the signal coming from our diode temperature sensor. V_{OUTPUT} is the signal that we will send to the ADC.

To determine the values for the components we match Equation 1 and Equation 2 to equations for resistor values.

Eq. 3 $$R_3\left(\frac{1}{R_2}+\frac{1}{R_4}\right) = 13.286$$

and

Eq. 4 $$V_{REF}\left[\frac{R_3}{R_2}\right] = 6.4286$$

The first thing to notice is that we have two equations with four unknowns, V_{REF}, R_2, R_3, and R_4. This means we can just pick convenient values for two of them based on what we have available.

The first thing we notice is that C_1 and R_1 are not part of the equations, so they can be freely selected without changing the gain of the amplifier.

Capacitor C_1 is called a bypass capacitor. Its purpose is to smooth out potential noise from the power supply. If we were designing a permanent circuit, we would want to include it, but for our test circuit, it can be omitted. A typical value would be 0.1 uF.

R1 just limits the possible current flow into the Op Amp. Again, this is optional, so we can select a common value like 1 KΩ or leave it out.

For $V_{REF,}$ we want something in the single-volt range; conveniently, we have the 3.3V supply on the Arduino, so use that. However, the GPS shield covers the 3.3V pin, so we cannot plug a wire into the top. The easiest option is to solder a wire into the hole next to the 3.3V pin to create a 3.3V jumper like our TX and RX jumpers. So V_{REF} is ~3.3V.

NOTE: The Arduino 3.3V regulator voltage will vary somewhat from Arduino to Arduino.

Now we are down to three unknowns (R2, R3, and R4) we can pick a common value for one of them and then solve for the other 2. Let's choose **R4 to be 1KΩ**. If we put in the selected values to our equations, we get:

Eq. 5 $$R_3\left(\frac{1}{1000}+\frac{1}{R_2}\right) = 13.286$$

and

Eq. 6 $$3.3\left[\frac{R_3}{R_2}\right] = 6.4286$$

Now we can solve for R2 and R3

Doing so we get the values **R3=11.338 KΩ** and **R2=5.820 KΩ**.

These obviously are not "off-the-shelf" component values. Also, small variations in V_{REF}, diode voltage, and R4 are going to cause our equations to be inexact. So, we will want to be able to tweak our circuits without completely rebuilding them.

To accomplish this, use a series combination of a fixed and a variable resistor, as shown in **Figure 6**. For the potentiometer connect one of the outer terminals back to the middle terminal on the breadboard. This means as we adjust the screw, we are partially bypassing the resistor. So we can change it from 0 to its max value(5K or 10K). So we build that on the breadboard and then adjust the potentiometer screw until the series combination reaches the desired value.

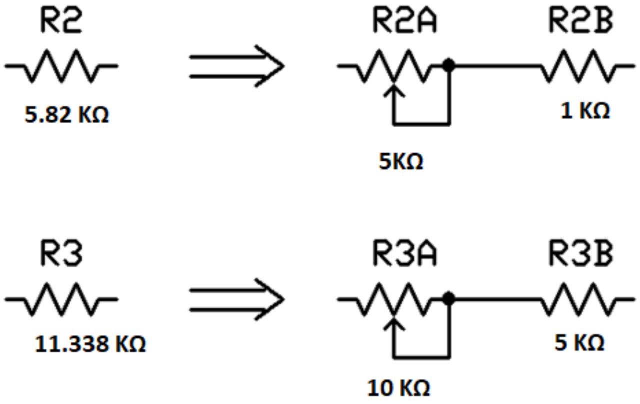

Figure 5: By combining a fixed resistor and potentiometer is series we can configure R2 and R3 to desired values.

When building this circuit, it is important to remember we have calculated the resistance to the third decimal place, but the actual values for the diode and resistances are not that exact. So do not worry about getting R3 and R2 to the exact values calculated. Just get it close.

The values we have determined are summarized in Table 1 below:

Table 1: Amplifying Circuit Component Values

R1 (Optional)	1 KΩ
R2	5.82 KΩ
R2A	5 KΩ Pot.
R2b	1 KΩ
R3	11.338 KΩ
R3A	10 KΩ Pot.
R3B	5 KΩ
R4	1 KΩ
C1 (Optional)	0.1 uF

We then connect the output of our amplifier to the Arduino like full circuit shown in Figure 7.

Note: A key point is that the ground of our temperature sensor, our amplifying circuit, and our Arduino should all be connected together so they are referencing the same 0V(ie grounds connected).

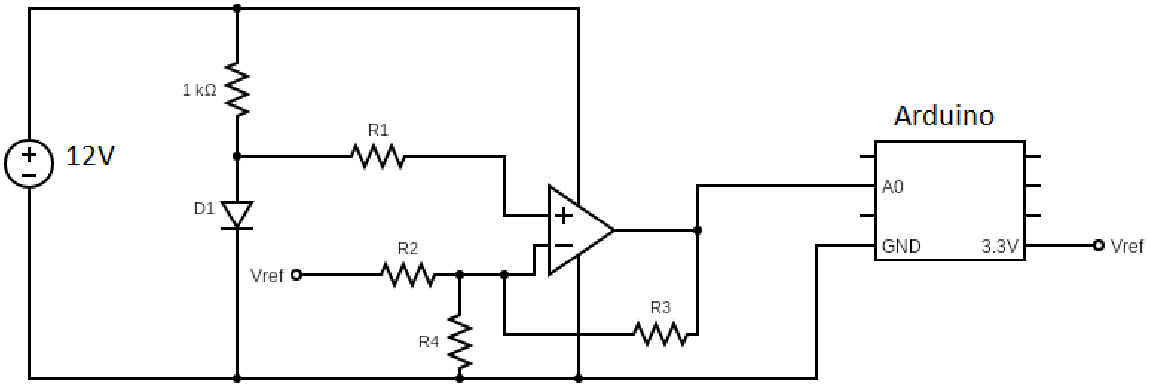

Figure 6: Combined sensor, amplifying, and Arduino circuit. Note analog instead of A0 any of the analog pins may be used. Also, the point where the lines cross to the left of R3 is not a connection.

Part 4: Calibration Procedure:

To calibrate, write a simple sketch that will read the voltage output of the amplified temperature sensor and outputs the ADC value to the serial monitor.

First, we need to take some measurements to check our circuit and possibly make adjustments to ensure we meet the range and resolution requirements.

For our final calibration, we want to make several measurements from 0-100C to show our sensor is linear. But if we do that every time we adjust the potentiometers, it will take an excessive amount of time. Instead, we can just make two measurements after each adjustment.

Use a water bath and thermometer to provide a known temperature just like we did with the SkeeterSat. We could do a hot bath and cold bath, or just use room temperature for one of the two points. Plot the two points with ADC on the X-axis and temperature on the Y. Then you calculate the equation of the line through those points you will get equation 7 below.

Eq. 7 $T = m * ADC + b$

You should expect to see the diode voltage decrease with an rise in temperature so the slope of your line should be negative.

From equation 7, we can get our resolution, minimum, and maximum values. To determine the minimum and maximum temperature values, you just plug in the minimum and maximum ADC values (The temperature max will be at 0 ADC and the minimum at 1023 ADC). The temperature resolution is just the slope (the change in temperature corresponding to the change in 1 ADC).

We will adjust our circuit as if we were going to use it in a balloon experiment to measure atmospheric temperature. The circuit should be able to measure from -50C to +50C and have a resolution of 1C. You may need to adjust your amplifier gain and offset to achieve this.

NOTE: You do not need to be at exactly -50C for your min; if your minimum is a lower temperature than that, you have met the requirement. The same applies to the maximum and resolution requirements.

Once we have finalized our amplifier values, we can take measurements at several temperatures and perform a full fit to generate a calibration equation. This might be slightly different from the equation of the line we used for the quick check while adjusting the amplifier but it should be close.

Ideally, we take calibration data all the way down to -50C, but we cannot do that without more specialized equipment. Be sure to record this calibration data so that it can be included in your report as proof of your calibration equation and to show the sensor is linear.

Part 5: Building A Datalogger

Once you have determined the calibration, it is time to combine the circuit you just built with the GPS shield to make a simple data logger that reads the output of the sensor and combines it with GPS data, and then writes that to data files on the SD card.

You will develop a simple Arduino program that reads your temperature sensor, reads the GPS, and writes data packets to the SD card.

Your data should be in .csv files that include the following elements:

- Data broken into multiple files
- Data set must be uniquely identifiable by some sort of numbering system
- A file header to identify what each column measures in the file
- Calibrated temperature recorded in degrees C
- Time and Date including day, month, year, hours, minutes, seconds
- Altitude
- Status of the GPS fix at time the data was recorded

You may record additional data and are free to choose the data format, but it should include the elements above at a minimum. Also, do not worry if you are unable to operate the sensor in a location where you can get a GPS fix.

Be sure to take pictures and notes in your lab notebook as you will write a report documenting your circuits, your calibration procedures, and your data logging software for **A16.02 –Capstone Report: Sensors Interface and Calibration Report.**