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
2. Solderless breadboard
3. Electronics Components construct bread boarded temperature sensor
    a. 1x 1N457 PN Diode
    b. 1x 1KΩ Current limiting resistor for diode
    c. 3x 1KΩ Resistor for Amplifying Circuit
    d. 2x 10KΩ potentiometer for Amplifying Circuit
    e. 1x 5KΩ potentiometer for Amplifying Circuit
    f. 1x 5KΩ Resistor for Amplifying Circuit
    g. 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. Bench 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.

Be sure to document this process using photos and your lab notebook as you will be writing a report documenting the development 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 are going to calibrate our temperature sensor in a similar procedure to what we did with the SkeeterSat earlier in the sensor. This means we need to prepare the diode in a similar way that we did to the thermistor. We need to solder longer wires to the leads of the diode and then cover it with either heat shrink or liquid electrical tape to waterproof it.

Unlike the thermistor the diode is polarized. Remember the diode stripe will mark the cathode which is the negative side of the diode. Since our heat shrink or liquid tape will cover the markings it is good idea to use two different color wires and record which one is was connected to positive and negative. (If possible, a good idea is to use the standard red for positive and black for negative). When done we should end up with have a diode ready to use for temperature sensor like 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 lets just use one we should have multiple like 1kΩ.

We put the resistor first so that we can use the voltage on the positive side of the diode and the ground as our signal. We want to supply the sensor from our bench supply at around ~12V if possible, but if no other supply is available use the 5V pin from the Arduino.

Our output signal that we will send to the amplifying circuit will be between the between the diode and the resistor as shown in figure 2.

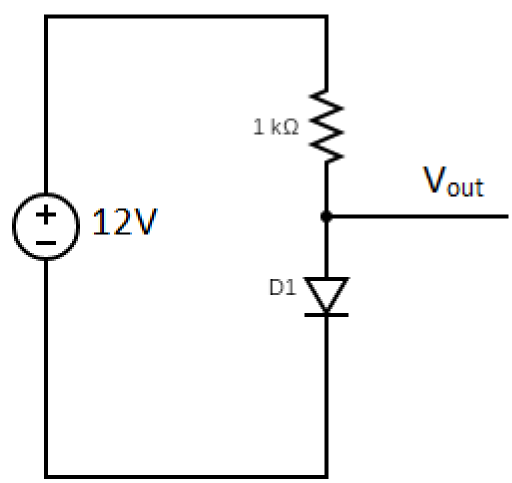

Figure 2 Setup for using diode as a temperature probe

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 temperature sensor is 450 mV and maximum value is 800 mV. This a good starting place but most likely will not match your diode exactly.

The base is 450 mV and the span are just 350 mV. So if we plot the vales for .45 to .8 V input to 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 which means that we need a noninverting amplifier.

We can also see that we will need an offset.

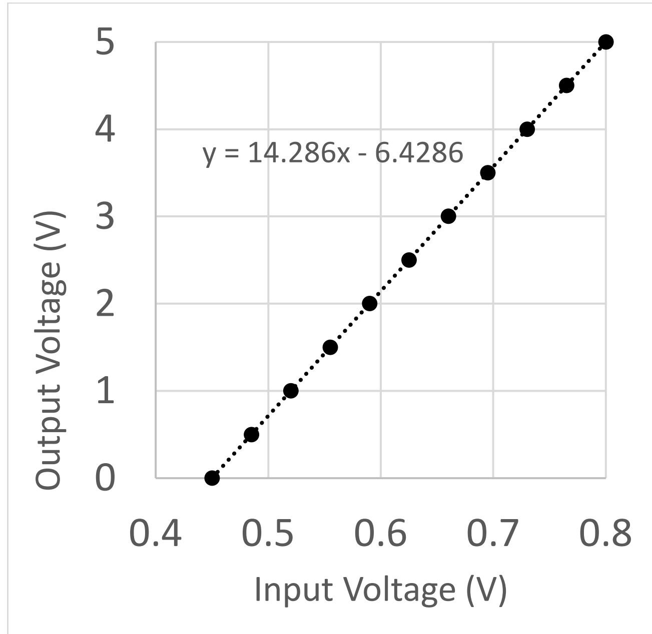

Figure 3 Linear plot of expected diode output range to ADC input range

From those 2 pieces of information, we determine we need a noninverting amplifier with offset. The schematic for that circuit is shown in figure 4.

So we now need to determine the values for the components in our circuit. Remember V_{input} is the signal coming from our diode temperature sensor.

The equation for the output of the circuit is:

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

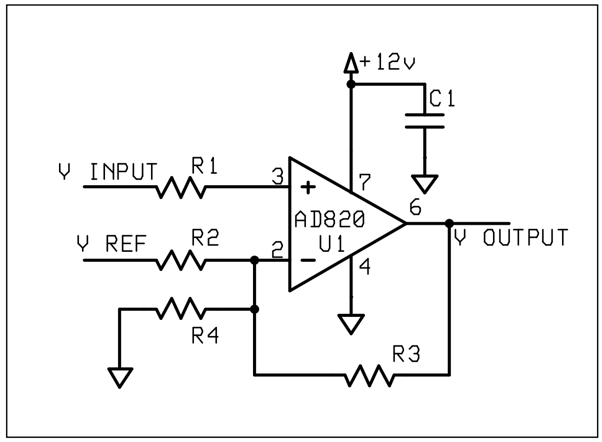

Figure 4 Noninverting Amplifier with offset

Remember V_{INPUT} is the voltage from output of our temperature sensor.

To determine the values for our components we match Equation 1 and Equation 2 to get to more equations for our components.

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 we notice is that C_1 and R_1 are not part of the equations.

C_1 is a capacitor to smooth out potential noise in the from the power supply. If we were designing a permanent circuit, we would want to include it but here it is optional. A typical value would be 0.1 uF.

R1 similarly is just to limit the possible current flow into the op amp. Again, this is optional but good practice so we can select a common value like 1 KΩ, but we can leave out if we are

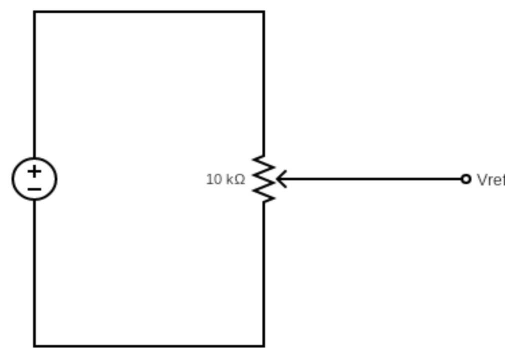

Figure 5 Providing V_{REF} via potentiometer

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 will cover the 3.3V pin and if we used the default pin headers, we will be unable to plug in to the top. In that case we have 2 options we could solder a wire in to the hole next to the 3.3V pin to create a 3.3V jumper like our TX and RX jumpers. Or we can use the remaining potentiometer as voltage divider as shown in figure 5. We simply connect the voltage source to the outer pins of the potentiometer and draw V_{REF} from the middle pin, by adjusting the potentiometer we can adjust V_{REF}. One thing to be aware of this method will cause V_{REF} to vary with the voltage source voltage so be sure make sure it does not change over the course of the calibration.

Since we have 2 equations and 3 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 $$3.3\left[\frac{R_3}{R_2}\right] = 6.4286$$

We now have 2 equations and 2 unknows that we can solve for.

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

These obviously are not "off-the-shelf" component values. We also may need to adjust our gain and offset once we have built our circuit.

To accomplish this, these resistors will be implemented as the series combination of a fixed and a variable resistor (*potentiometer*), as shown in **Figure 6**. Connect one of the outer terminals back to the center terminal of the potentiometer, then adjust the potentiometer screw until the series combination reaches the desired value. Also, while we have calculated the resistance to the third decimal places remember our other resistors have uncertainty of around 5% and diode voltage may vary, so do not worry about getting R3 and R2 to the exact values needed.

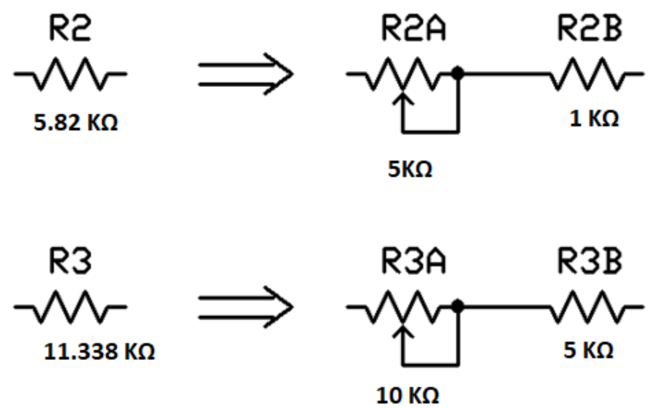

Figure 6 By combining a fixed resistor and potentiometer is series we are able to configure R2 and R3 to desired values.

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

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).

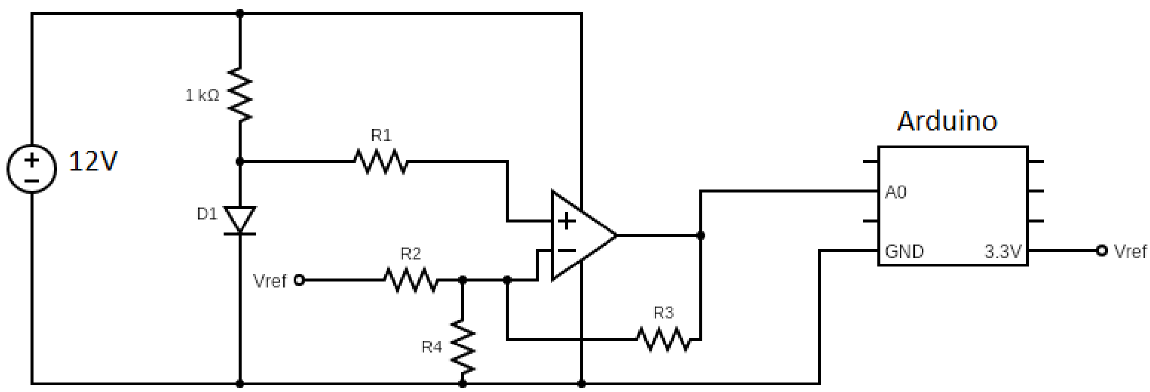

Figure 7 Combined sensor, amplifying, and Arduino circuit. Note analog A0 is not required any of the analog pins may be used. Also, the point where the leads cross left of R3 is not a connection.

Part 4: Calibration Procedure:

Now we need to construct the circuit on the breadboard and hook it up to the Arduino so we can record ADC values to determine the calibration.

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

Then use a water bath to determine a calibration function from ADC value to temperature. The function that will take ADC as input and output temperature in degrees C. You should see a linear equation in this case. Your calibration equation should be of the form:

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

You should expect to see the diode voltage decrease with an increase in temperature so the slope of your line should be negative. Determine the minimum and maximum value of the temperature you can measure (The temperatures at 0 ADC and 1024 ADC) as well as the temperature resolution (the change in temperature corresponding to 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. At a minimum your 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.

A simple sketch that outputs the ADC value will be useful. You can also take 2 quick measurements (for example at room temperature and at 50C). Then you determine the slope of that line and determine the slope of the that line and determine if we need to adjust our amplifying circuit before collecting our calibration data.

Once this complete, we can record data at a number of different temperatures and perform a full fit in order to generate a calibration equation.

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 writes that to a data record.

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 records should be .csv files that include the following elements:

- Data broken into multiple files
- Each data records must be uniquely identifiable by some sort of numbering system
- Calibrated temperature recorded in degrees C
- Time and Date including day, month, and year
- Altitude
- Status of the GPS fix at time the data was recorded

You may record additional data and are free to choose the format of the data, 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 be writing a report documenting your circuits, your calibrations procedures, and your data logging software for **A16.02 –Capstone Report: Sensors Interface and Calibration Report.**

The report should include narrative description of developing the sensor and calibration, sample data record and file format, tables of measured data, electronic circuit diagram(s), a description of the software needed to read and store the data.

References:

R16.04 AD820A Data Sheet