Name: ____ **Date:** ____________________ **Class:** ________________

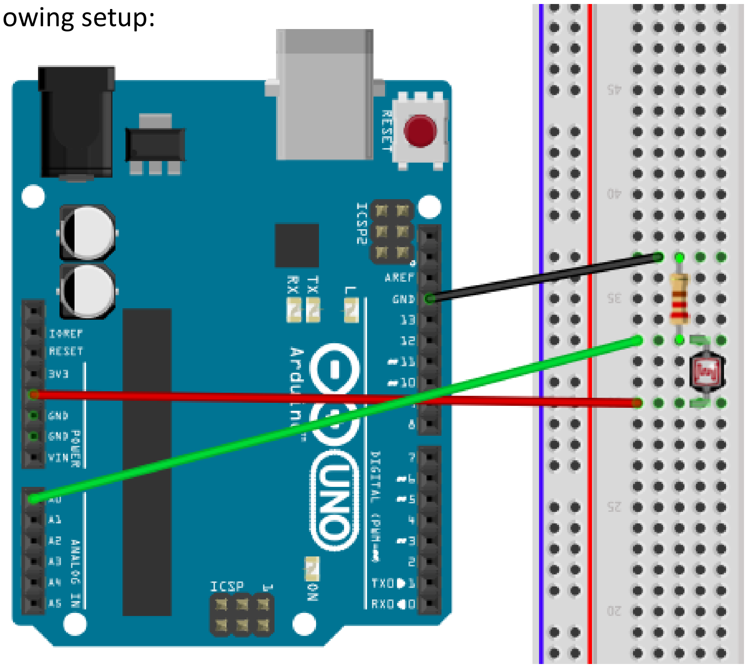
Activity #4: Using a Photoresistor Handout

Overview

In this activity, we will learn how to use the Arduino to collect measurements (signals) from a photoresistor—a sensor that detects light.

Construct the Hardware

1. Gather the following materials:
   - 1 Arduino, USB cable & breadboard
   - 1 220Ω resistor (red-red-brown)
   - 2 jumper wires
   - 1 photoresistor (aka light-dependent resistor or photocell)
2. Construct the following setup:



   A. Connect 1 jumper wire from GND ("ground") to one of the rows. Then place 1 leg of the resistor in the same row as the GND wire. Place the other leg of the resistor in a new row.
   B. Place 1 leg of the photoresistor in the previous row. Place the other leg of the resistor in a new row.
   C. Connect 1 jumper wire from the previous row to the 5V pin.
   D. Connect 1 jumper wire from analog pin A0 to the row that has the legs of both the resistor and photoresistor.
   E. Check that your setup matches the setup in the diagram above (or at the front of the room).

Create the Program

3. Type up the associated program for Activity #4 (see the next page).
4. Save the file as **Sensor_Reading**.

5. To check that the Arduino can compile your code: “Verify” it. Do this by either pressing **Command + R** or by clicking on the Verify button (it looks like a check mark).

```
/*
Activity 4: Sensor Reading
Collect measurements (signals) from a sensor using an Analog Input Pin.
*/

int SensorPin = A0;

int Sensor_value = 0;

void setup() {

         //Setup a connection for serial communication between the Arduino and the computer.
        Serial.begin(9600);
        Serial.println("Getting started.... ");
}

void loop() {

        Sensor_value = analogRead(SensorPin);

        Serial.println(Sensor_value);

        delay(200);
}
```

Troubleshoot Errors

6. Error messages pop up in orange text at the bottom of the IDE window. The computer tells you where (what lines) it believes the errors in the program are located (though it is not always right).
7. Start by reading the first error message and checking that line of code. Compare what you have written to the original program. Common errors:
   - If you get the message “________ was not declared in this scope,” it is possible you misspelled one of the variables or functions.
   - If you get the message “expected ‘;’ before ____” — then you forgot to add a semi-colon at the end of the previous line.

Run the Program

8. Using the USB cable, connect the Arduino to the computer. Expect to see some lights flash on the Arduino board. Then the Arduino starts to execute the last program that was uploaded to it.
9. From the “Tools” menu in the Arduino IDE, make sure the appropriate serial port is selected.
10. Now upload the “Sensor_Reading” sketch. To do this, hit **Command + U.** (Or, click on the Upload Button in the Arduino IDE, which looks like an arrow pointing to the right.)
11. Wait a few seconds. Then open the serial monitor by pressing **Shift + Command + M**. (Alternately, click on the magnifying glass icon at the top right corner of the Arduino IDE window to open the serial monitor).

12. Expect to see a list of numbers start to form. Move your hand over the photoresistor (up and down) and observe how the numbers change.

Reflection Questions

1. What do you think the function **analogRead( )** does? What information must go in its parentheses for it to work?

2. What do you think the function **Serial.println( )** does? What information must go in its parentheses for it to work?

3. What are the lowest and highest readings you get on the serial monitor? Why do you get these numbers? Read the following two pages from the Arduino website and then answer this question.
   - http://arduino.cc/en/Tutorial/AnalogInputPins
   - http://arduino.cc/en/Reference/analogRead

Alternatives

1. Figure out how to create a nightlight by adding an LED to your setup that is controlled by the photoresistor. Make it so the LED turns on when the photoresistor is covered; otherwise, the LED stays off. Sketch the new setup below and describe the changes you made to the program to accomplish this.

2. Building off the first alternative, replace the photoresistor with a force-sensitive resistor (FSR) pad. Make it so that when you push down on the FSR, the LED goes on; otherwise, the LED stays off. Use alligator clips to connect the FSR to the system.

3. *For Programmers:* Building off the first alternative, create an adaptable nightlight such that its level of brightness is dependent on the level of light received by the photoresistor. There are a few ways to do this. One way is to use the **map( )** function. If you need some help, watch Jeremy Blum's tutorial on Analog Inputs (see link below), from time 8:39 to 11:45 in the video.

Further Reading

Learn more about analog inputs and photoresistors in *Getting Started with Arduino* by Massimo Banzi; look at Chapter 5, pgs. 60-62. Download a pdf of the book at: http://it-ebooks.info/book/1338/.

Watch Jeremy Blum's *Tutorial 04 for Arduino: Analog Inputs* (18-minute video) at: https://www.youtube.com/watch?v=js4TK0U848I&index=4&list=PLA567CE235D39FA84.