**MEGN 200 – Worksheet 5**

**Light Source Locator Assignment**

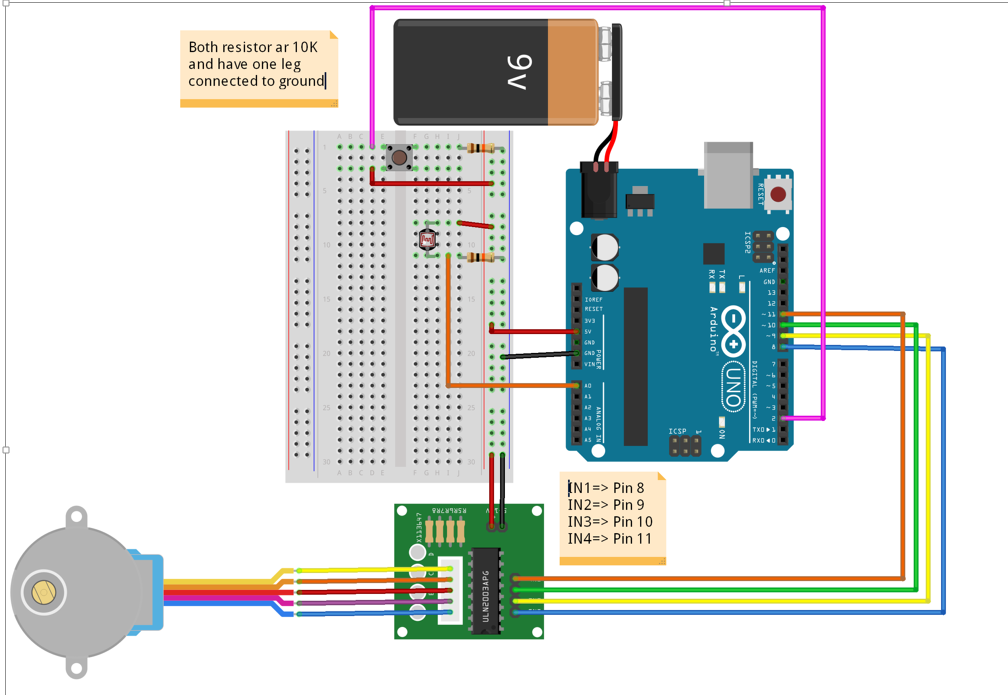
This effort will be to develop a mechanism and software that will scan 180 degrees and identify the brightest light source using only parts available in the Elegoo R3 Super Starter Kit provided for this section.

Parts required from the Elegoo kit:

|  |  |
| --- | --- |
| Arduino Uno w/USB cable | 9V battery (optional, USB power is sufficient) |
| Breadboard | Assorted jumper wires |
| Stepper motor w/ ULN2003 driver | Cardboard & Tape (provided by the student) |
| Push Button | Photoresistor |
| 2x 10K resistors | UNO Shield w/ small breadboard (optional) |

**Part 1: Test to make sure your stepper motor and driver are working (*0 pts, no submission*)**

NOTE: The wiring diagram below can be used for all three parts, and the longer female-male cables are needed to connect the stepper driver. Furthermore, you do NOT need the 9V battery (USB power is sufficient), but it could be helpful for additional power. For Part 1, the stepper motor should function as follows: The stepper motor should rotate 360o (2048 steps) clockwise, and then 360o (2048 steps) counter-clockwise. To achieve this, open and upload the provided Arduino code “Part1\_stepper\_oneRevolution.ino”. This code uses the built-in Arduino library called Stepper.h.



**Part 2: Photo Resistor Test & Measurements *(7 pts).***

1. Open the code “Part\_2\_Photo\_Resistor\_Test.ino”
2. In the code, insert an if/else statement. This if/else statement should consistently turn the built-in LED attached to pin 13 on when the photoresistor is covered and off when the photoresistor is not covered. Refer to the basic BLINK code to see how to turn the built-in pin 13 LED on/off.
3. NOTE: When covering the photoresistor, you must cover it fully with a large piece of paper or your entire hand to prevent most ambient light from reaching it. **Submit your commented code for Part 2 as a link. No video is required**. *(****3 pts****)*
   1. COMMENTED CODE LINK HERE
4. Approximately what analogRead value do you read when the photoresistor is uncovered vs.

covered? What value did you use as a threshold to consistently turn the LED on and off?

*(****1.5 pts****)*

analogRead value, uncovered: \_\_\_\_\_\_\_\_\_

analogRead value, covered: \_\_\_\_\_\_\_\_\_

Threshold value: \_\_\_\_\_\_\_\_\_

What voltages do these 10 bit digital readings equate to? *(****1.5 pts****)*

Uncovered: \_\_\_\_\_ [V]

Covered: \_\_\_\_\_\_ [V]

Threshold value: \_\_\_\_\_ [V]

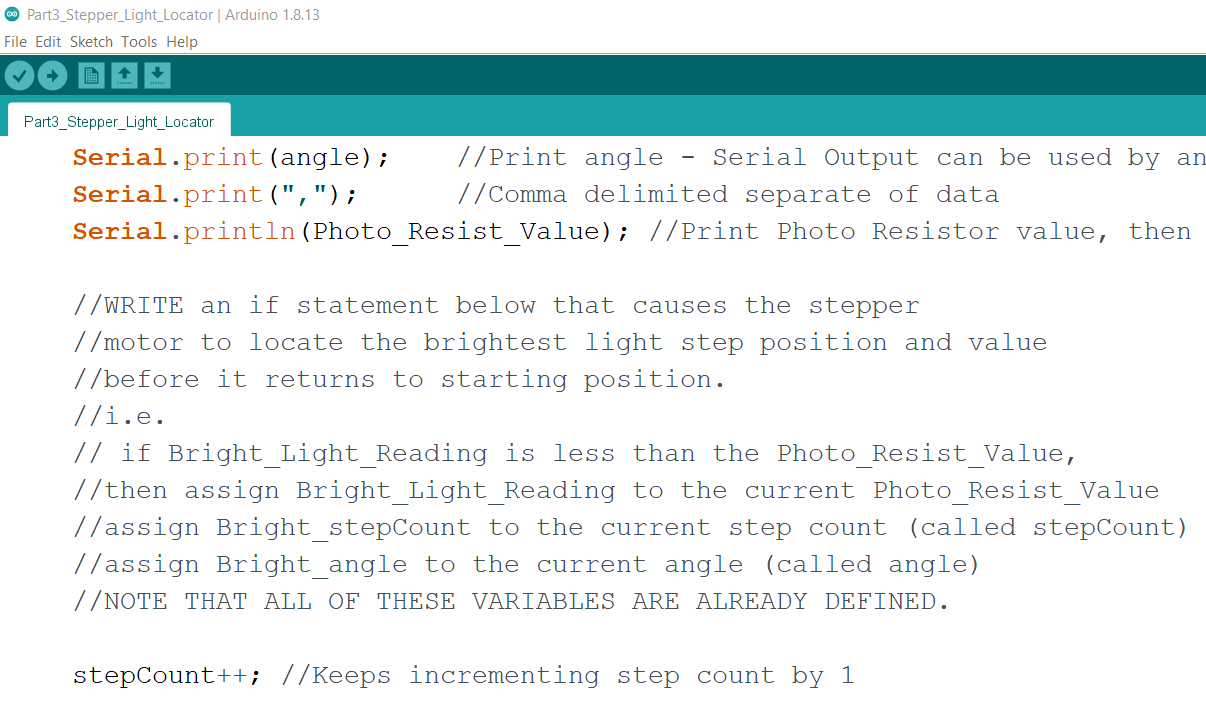
1. When covered, does your ADC value ever reach 0? Why or why not? *(****1 pt****)*

**Yes/No**

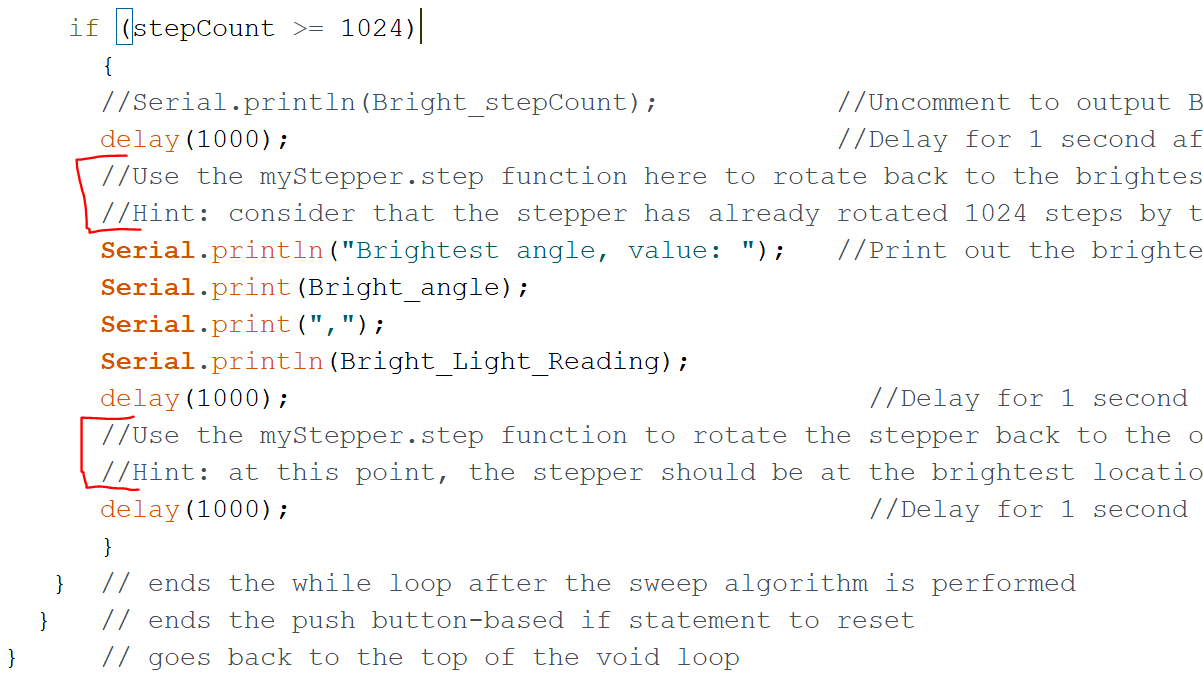
**Why/Why not?**

**Part 3: Locating a Bright Light Source (similar to a solar tracker) *(8 pts)***

1. This part involves *slightly* changing the wiring diagram above. You will need to “extend” the photoresistor by removing it and replacing its connections with the male sides of the longer male-to-female cables from the kit. Next, using tape, attach the photoresistor to a piece of cardboard, and connect the photoresistor’s ends to the female portions of the male-to-female cables. As you will see, this can be a cable management challenge to ensure your wires do not twist over others. **See photos on page 4.** For further clarification, please watch the **Part 3 Demonstration Video** (<https://www.youtube.com/watch?v=ZGnqL_YdwIY>).
2. Insert an if statement into the code that causes the stepper motor to locate the brightest light step position and value before it returns to starting position. A screenshot of the code insertion location is below:



1. When the “start scan” button is pressed the stepper should begin a 180 degree (1/2 a revolution) sweep. Remember the stepper library uses the starting position as 0 degrees/steps and all steps are referenced relative to this starting position.
2. During this scan the device should store the highest reading received from the photo resistor voltage divider input at A0. To start the scan, **press the push button on your breadboard.** **Then,** **shine a light on the photo resistor at any time** during the scan and note the direction that the pointer is pointing. You’ll need to store the brightest location as described in Step 2.
3. After the full scan is complete, the device should pause for 1 second, and then return to the location where it recorded the highest photoresistor light reading, and then return to the starting position (as shown in the video). The provided started code will only rotate 180 degrees - the code needs to be modified as described in the comments to accomplish this (see screenshot below).



1. Once the code and circuit is working, run the code 3 times, each time shining a light at **different locations** along the scan. Remember to press the push-button before each scan to start it and after the last scan was completed. **Make sure to record all three sweeps in your video**. **Write down the angles (in degrees) and the brightness values in the table below.** You can print out the final brightness and angle values using the Serial Monitor and Serial.print() statements in the code. (**3 pts**)

|  |  |  |  |
| --- | --- | --- | --- |
| **Reading** | **1** | **2** | **3** |
| **Angle Location (o)** |  |  |  |
| **Brightness Value** |  |  |  |

1. **Submit your video of your 3 scans working (introduce yourself as usual and explain the system).** *(****5 pts****)*
   1. VIDEO LINK HERE

**Example Design of Photo Resistor Light Scanner (needs cardboard and tape)**

|  |  |  |
| --- | --- | --- |
|  |  |  |