MEGN 200 – Worksheet 5

Light Source Locator Assignment

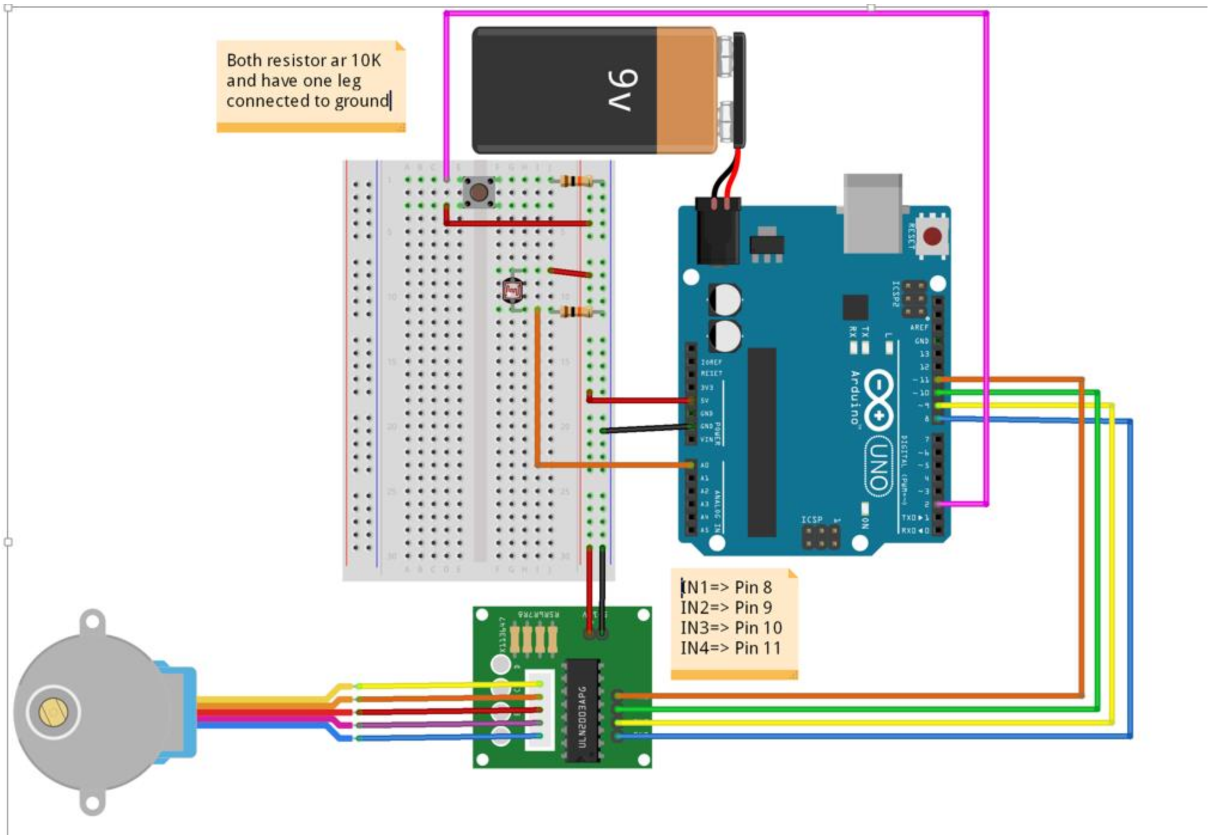
This effort will be to develop a mechanism and software that will scan 180 degrees and identify the brightest light source using only parts available in the Elegoo R3 Super Starter Kit provided for this section.

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.


Both resistor ar 10K and have one leg connected to ground

IN1=> Pin 8
IN2=> Pin 9
IN3=> Pin 10
IN4=> Pin 11

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**)*
    a. 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]

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

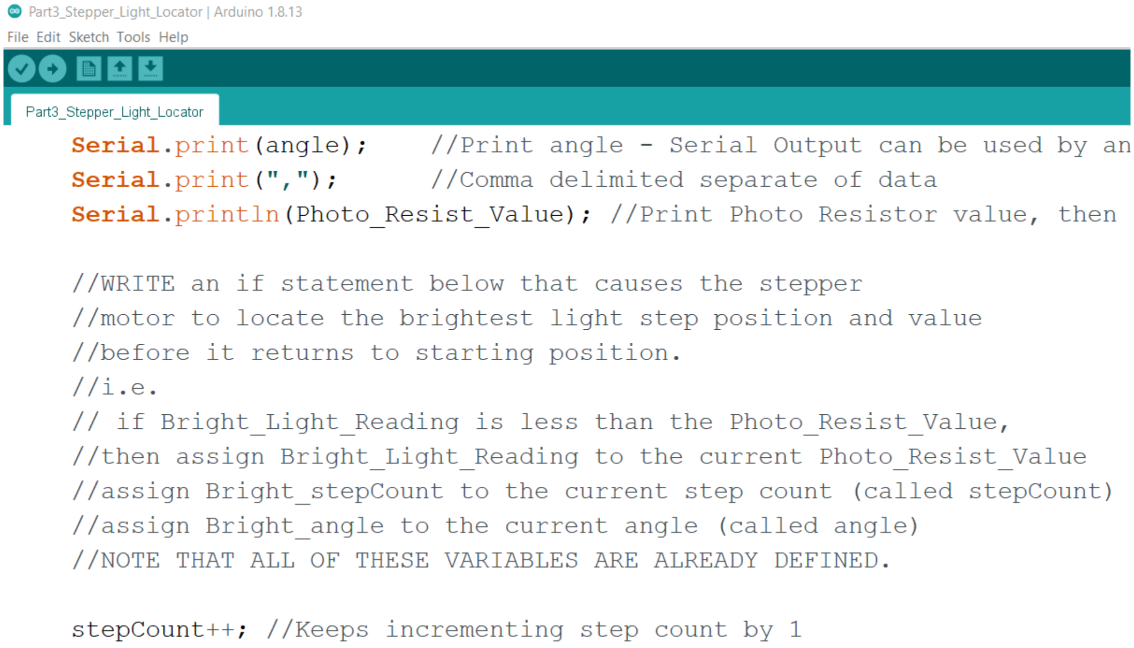
   Yes/No

   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:

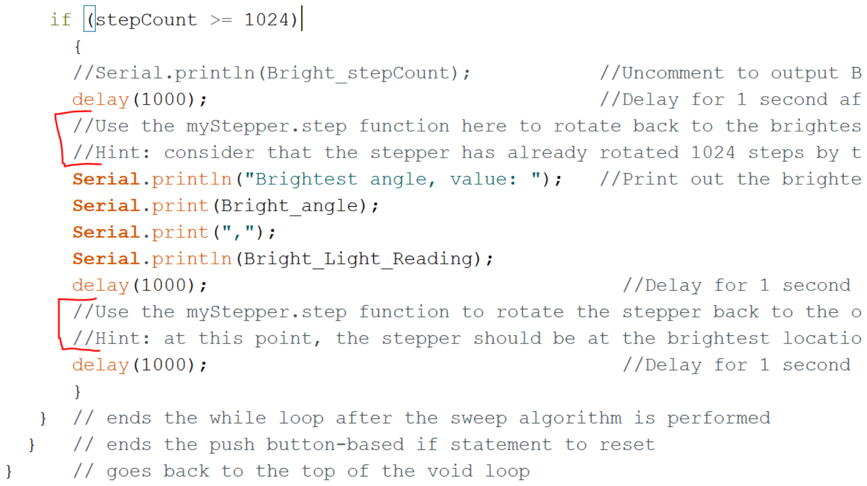

Part3_Stepper_Light_Locator | Arduino 1.8.13
File Edit Sketch Tools Help
Part3_Stepper_Light_Locator

```
    Serial.print(angle);    //Print angle - Serial Output can be used by an
    Serial.print(",");      //Comma delimited separate of data
    Serial.println(Photo_Resist_Value); //Print Photo Resistor value, then

    //WRITE an if statement below that causes the stepper
    //motor to locate the brightest light step position and value
    //before it returns to starting position.
    //i.e.
    // if Bright_Light_Reading is less than the Photo_Resist_Value,
    //then assign Bright_Light_Reading to the current Photo_Resist_Value
    //assign Bright_stepCount to the current step count (called stepCount)
    //assign Bright_angle to the current angle (called angle)
    //NOTE THAT ALL OF THESE VARIABLES ARE ALREADY DEFINED.

    stepCount++; //Keeps incrementing step count by 1
```

3. 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.
4. 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.
5. 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).



```
    if (stepCount >= 1024)
      {
      //Serial.println(Bright_stepCount);           //Uncomment to output B
      delay(1000);                                  //Delay for 1 second af
      //Use the myStepper.step function here to rotate back to the brightes
      //Hint: consider that the stepper has already rotated 1024 steps by t
      Serial.println("Brightest angle, value: ");   //Print out the brighte
      Serial.print(Bright_angle);
      Serial.print(",");
      Serial.println(Bright_Light_Reading);
      delay(1000);                                    //Delay for 1 second
      //Use the myStepper.step function to rotate the stepper back to the o
      //Hint: at this point, the stepper should be at the brightest locatio
      delay(1000);                                    //Delay for 1 second
      }
   }  // ends the while loop after the sweep algorithm is performed
  }   // ends the push button-based if statement to reset
}     // goes back to the top of the void loop
```

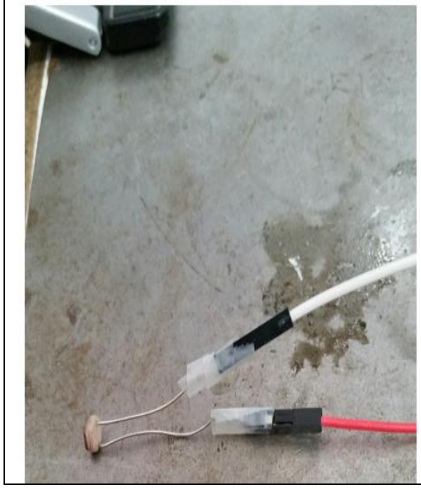
6. Once the code and circuit is working, run the code 3 times, each time shining a light at **different locations** along the scan. Remember to press the push-button before each scan to start it and after the last scan was completed. **Make sure to record all three sweeps in your video**. **Write down 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			

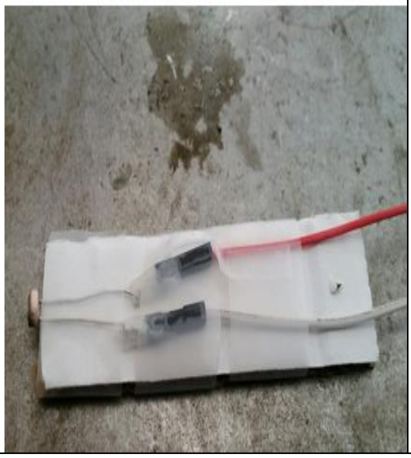
7. **Submit your video of your 3 scans working (introduce yourself as usual and explain the system).** *(**5 pts**)*
    a. VIDEO LINK HERE

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

1) Twist photo resistor leads to jumpers and tape



2) Twist photo resistor leads to jumpers and tape



3) Mount to stepper shaft