





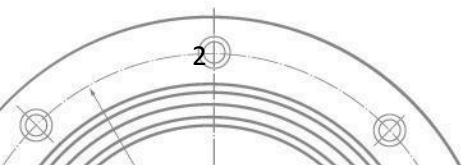
**Overview:**

- Students will copy code to observe a Light Dependent Resistor.

**Objectives:**

*Students will be able to:*

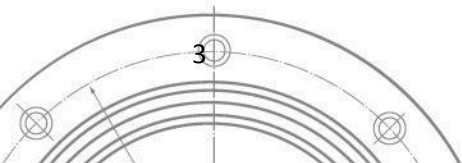
- Determine what an LDR is
- Test code to determine light intensity
- Read the Serial Monitor to determine data output





## Computer Science Teacher Association Standards

- 1A-CS-02 Use appropriate terminology in identifying and describing the function of common physical components of computer systems.
- 1B-AP-12 Modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
- 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended
- .



## Materials:

- [Frog Microcontroller Trainer \(FMT\)](#) from 1<sup>st</sup> Maker Space
- USB Power Cord
- Arduino IDE
- PC or Mac
- Chromebooks if using Arduino Create for Education App
- Introduction to Microcontroller Trainer Lesson (if needed as a resource)

## Preparation:

- Ensure students have a compatible device and have downloaded the Arduino IDE software on device.
- Ensure all students know how to properly connect the FMT to the device via the USB cable.

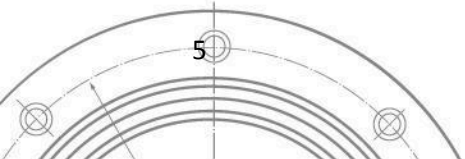
## Background Information:

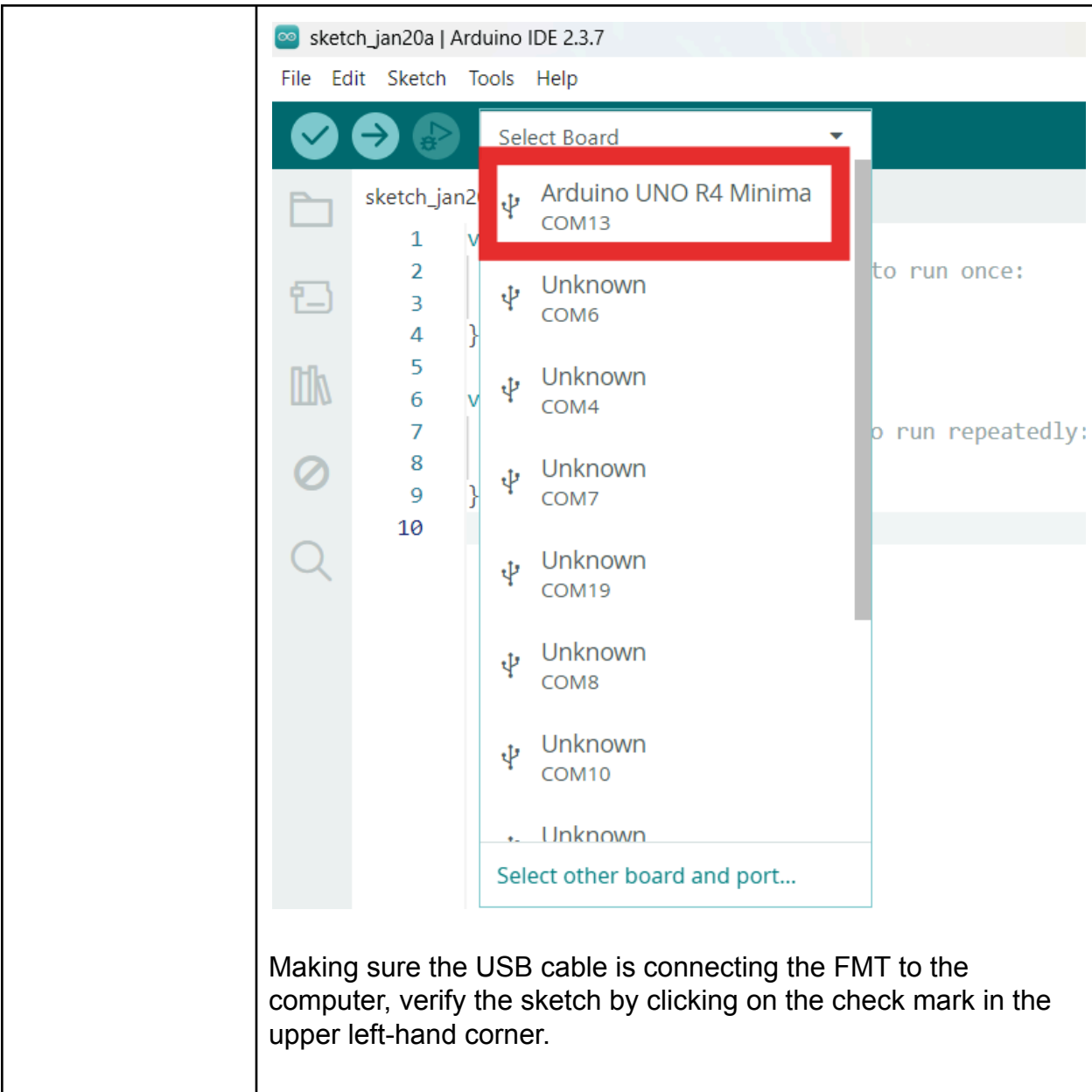
A light-dependent resistor, or LDR for short, is a special kind of electronic component that can sense light. It's like having a tiny eye that can tell how bright or dark it is around it. When the LDR is in the dark, its resistance gets bigger. When light shines on it, its resistance becomes smaller.

Lesson Elements													
<b>Basic Part Identification</b> — 10 minutes	Have students identify the Light Dependent Resistor (LDR) on the FMT.												
<b>Access 1MS Code</b> — 15 minutes	<p>Visit the 1st Maker Space <a href="#">website</a> and access the library of code developed specifically for the FMT. Here, you will find a folder with various code written for each major component of the trainer. The teacher will locate the sketch by clicking 8.00-Light Sensor.</p> <table border="1"><tbody><tr><td><b>Project 7.01 Writing Text to the Screen</b> In this code, you'll learn how to display text and numbers on the OLED screen! Read More →</td><td><b>Project 7.02 Reaction Game Using OLED</b> This Arduino sketch is for a reaction game using an OLED display and two buttons (SW1 and SW2)! Read More →</td><td><b>Project 7.03 Drawing Shapes with the OLED</b> We can do more than just text on the OLED. In this project you'll learn how to draw shapes too! Read More →</td><td><b>Project 8.00 Light Sensor</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and prints the values to the serial monitor! Read More →</td></tr><tr><td><b>Project 8.01 Max and Min Brightness</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and continuously monitors and updates the maximum and minimum brightness values observed! Read More →</td><td><b>Project 8.02 Mapping Light</b> This Arduino sketch reads the analog value from a light sensor connected to pin A2, maps it to a corresponding brightness level for an LED connected to pin 13 (LED1), and adjusts the LED brightness accordingly! Read More →</td><td><b>Project 9.00 Using the Temp Sensor</b> This Arduino sketch reads the analog value from a temperature sensor connected to analog pin A3 and prints the temperature value (represented by the analog reading) to the serial monitor! Read More →</td><td><b>Project 9.01 Getting an Actual Temperature Reading</b> This Arduino sketch reads the analog value from the temperature sensor connected to analog pin A3 and converts it into actual temperature readings in both Celsius (°C) and Fahrenheit (°F) using functions for reading the temperature and performing the conversion! Read More →</td></tr><tr><td><b>Project 9.02 Doing Something Based on Temperature</b> In this project, you'll learn how to create a dynamic temperature-responsive display using the MC Trainer! Read More →</td><td><b>Project 10.00 Decoding IR</b> This Arduino sketch demonstrates how to use an Infrared (IR) receiver module to receive and decode IR remote control signals! Read More →</td><td><b>Project 10.01 Sending IR</b> In this project, you'll discover how to transmit infrared (IR) data using the MC Trainer and the IRremote library! Read More →</td><td><b>Project 11.00 Using the Board as a Mouse</b> This Arduino sketch demonstrates how to use the MC Trainer board as a mouse input device! Read More →</td></tr></tbody></table>	<b>Project 7.01 Writing Text to the Screen</b> In this code, you'll learn how to display text and numbers on the OLED screen! Read More →	<b>Project 7.02 Reaction Game Using OLED</b> This Arduino sketch is for a reaction game using an OLED display and two buttons (SW1 and SW2)! Read More →	<b>Project 7.03 Drawing Shapes with the OLED</b> We can do more than just text on the OLED. In this project you'll learn how to draw shapes too! Read More →	<b>Project 8.00 Light Sensor</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and prints the values to the serial monitor! Read More →	<b>Project 8.01 Max and Min Brightness</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and continuously monitors and updates the maximum and minimum brightness values observed! Read More →	<b>Project 8.02 Mapping Light</b> This Arduino sketch reads the analog value from a light sensor connected to pin A2, maps it to a corresponding brightness level for an LED connected to pin 13 (LED1), and adjusts the LED brightness accordingly! Read More →	<b>Project 9.00 Using the Temp Sensor</b> This Arduino sketch reads the analog value from a temperature sensor connected to analog pin A3 and prints the temperature value (represented by the analog reading) to the serial monitor! Read More →	<b>Project 9.01 Getting an Actual Temperature Reading</b> This Arduino sketch reads the analog value from the temperature sensor connected to analog pin A3 and converts it into actual temperature readings in both Celsius (°C) and Fahrenheit (°F) using functions for reading the temperature and performing the conversion! Read More →	<b>Project 9.02 Doing Something Based on Temperature</b> In this project, you'll learn how to create a dynamic temperature-responsive display using the MC Trainer! Read More →	<b>Project 10.00 Decoding IR</b> This Arduino sketch demonstrates how to use an Infrared (IR) receiver module to receive and decode IR remote control signals! Read More →	<b>Project 10.01 Sending IR</b> In this project, you'll discover how to transmit infrared (IR) data using the MC Trainer and the IRremote library! Read More →	<b>Project 11.00 Using the Board as a Mouse</b> This Arduino sketch demonstrates how to use the MC Trainer board as a mouse input device! Read More →
<b>Project 7.01 Writing Text to the Screen</b> In this code, you'll learn how to display text and numbers on the OLED screen! Read More →	<b>Project 7.02 Reaction Game Using OLED</b> This Arduino sketch is for a reaction game using an OLED display and two buttons (SW1 and SW2)! Read More →	<b>Project 7.03 Drawing Shapes with the OLED</b> We can do more than just text on the OLED. In this project you'll learn how to draw shapes too! Read More →	<b>Project 8.00 Light Sensor</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and prints the values to the serial monitor! Read More →										
<b>Project 8.01 Max and Min Brightness</b> This Arduino sketch reads the light intensity from a light sensor connected to analog pin A2 and continuously monitors and updates the maximum and minimum brightness values observed! Read More →	<b>Project 8.02 Mapping Light</b> This Arduino sketch reads the analog value from a light sensor connected to pin A2, maps it to a corresponding brightness level for an LED connected to pin 13 (LED1), and adjusts the LED brightness accordingly! Read More →	<b>Project 9.00 Using the Temp Sensor</b> This Arduino sketch reads the analog value from a temperature sensor connected to analog pin A3 and prints the temperature value (represented by the analog reading) to the serial monitor! Read More →	<b>Project 9.01 Getting an Actual Temperature Reading</b> This Arduino sketch reads the analog value from the temperature sensor connected to analog pin A3 and converts it into actual temperature readings in both Celsius (°C) and Fahrenheit (°F) using functions for reading the temperature and performing the conversion! Read More →										
<b>Project 9.02 Doing Something Based on Temperature</b> In this project, you'll learn how to create a dynamic temperature-responsive display using the MC Trainer! Read More →	<b>Project 10.00 Decoding IR</b> This Arduino sketch demonstrates how to use an Infrared (IR) receiver module to receive and decode IR remote control signals! Read More →	<b>Project 10.01 Sending IR</b> In this project, you'll discover how to transmit infrared (IR) data using the MC Trainer and the IRremote library! Read More →	<b>Project 11.00 Using the Board as a Mouse</b> This Arduino sketch demonstrates how to use the MC Trainer board as a mouse input device! Read More →										



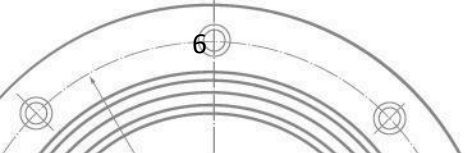
	<p>Here, one can find the sketch to share with students. Students can copy the code by simply clicking on the blue copy button.</p> <p><b>Project Code:</b></p> <pre>//////////////////////////////////// //8.00 - Reading Light Intensity  byte lightSensorPin = A2;  void setup() {   pinMode(lightSensorPin, INPUT);   Serial.begin(9600); }  void loop() {   Serial.print("The light level is at: "); Serial.println(analogRead(lightSensorPin)); }</pre>
<p><b>Connect the Trainer and Verify the Sketch – 10 minutes</b></p>	<p>Plug the FMT into your computer’s USB port. Using the dropdown box near the top, select the Arduino Uno R4 Minima and appropriate COM port. <b>Note:</b> Mac computers may automatically select the port.</p>





The screenshot shows the Arduino IDE interface. At the top, the title bar reads "sketch\_jan20a | Arduino IDE 2.3.7". Below it is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". The main workspace is divided into three panes: a file explorer on the left showing a folder named "sketch\_jan20a" with files numbered 1 through 10; a central editor pane with a "Select Board" dropdown menu open, listing "Arduino UNO R4 Minima COM13" (highlighted with a red box) and several "Unknown" boards with COM ports (COM6, COM4, COM7, COM19, COM8, COM10); and a serial monitor on the right with the text "to run once:" and "o run repeatedly:". In the top-left corner of the IDE, there are three icons: a checkmark, a right-pointing arrow, and a play button.

Making sure the USB cable is connecting the FMT to the computer, verify the sketch by clicking on the check mark in the upper left-hand corner.

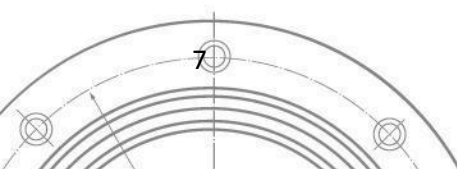
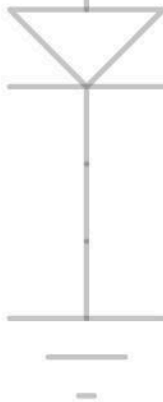


```
0-Reading_Light_Intensity.ino
1 ////////////////////////////////////////////////////
2 //8.00 - Reading Light Intensity
3 // Onboard the MCU Trainer is a LDR.
4 // LDR stands for "light dependent resistor".
5 // This means that the LDR changes its resistance based on the light applied to it.
6 // This makes it very easy to determine how much light is shining on the board.
7 // We can do that by doing an analogRead on the pin it is connected to.
8
9 byte lightSensorPin = A2;
10
11 void setup() {
12 //We are reading a voltage, so the pin is an input.
13 pinMode(lightSensorPin, INPUT);
14 Serial.begin(9600);
15 }
16
17 void loop() {
18 //This line reads the voltage across the LDR and prints it out
19 Serial.print("The light level is at: "); Serial.println(analogRead(lightSensorPin))
20 delay(1000);
21 }
22
23 ////////////////////////////////////////////////////
24
```

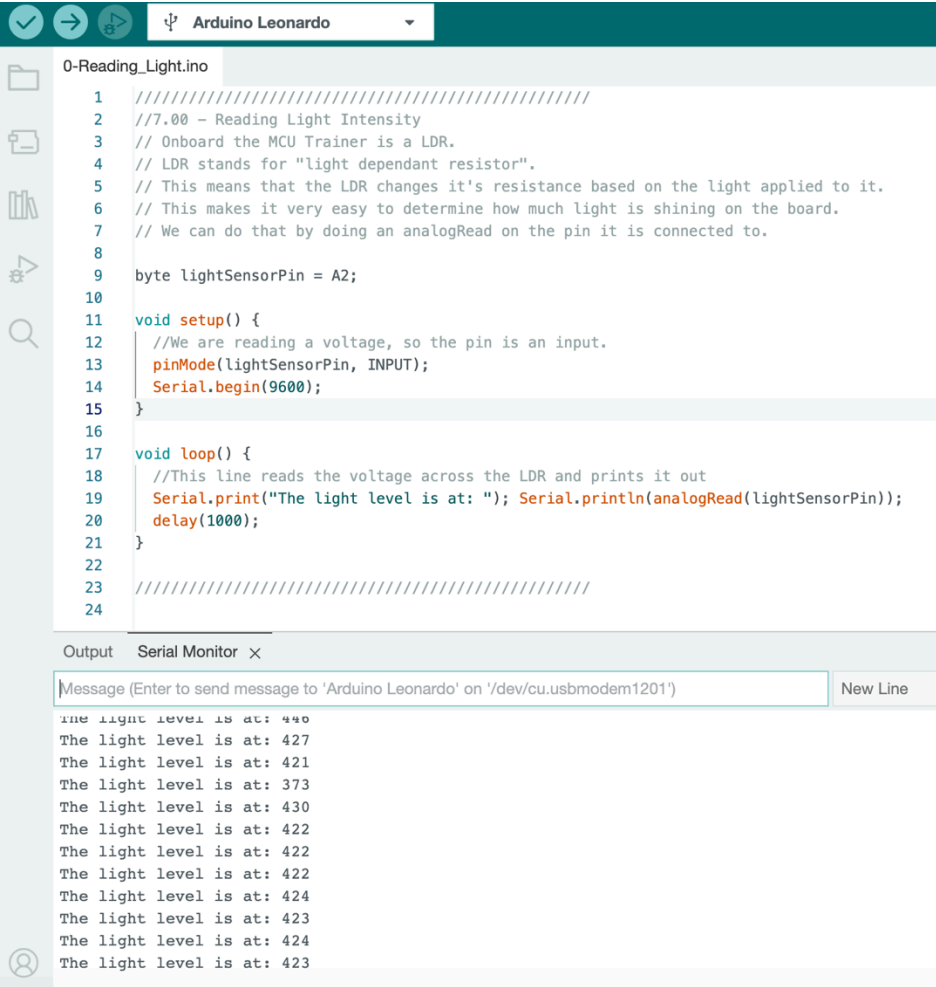
Once verified, upload the sketch to the trainer by clicking the right-facing arrow. It will turn yellow.

```
0-Reading_Light_Intensity.ino
1 ////////////////////////////////////////////////////
2 //8.00 - Reading Light Intensity
3 // Onboard the MCU Trainer is a LDR.
4 // LDR stands for "light dependent resistor".
5 // This means that the LDR changes its resistance based on the light applied to it.
6 // This makes it very easy to determine how much light is shining on the board.
7 // We can do that by doing an analogRead on the pin it is connected to.
8
9 byte lightSensorPin = A2;
10
11 void setup() {
12 //We are reading a voltage, so the pin is an input.
13 pinMode(lightSensorPin, INPUT);
14 Serial.begin(9600);
15 }
16
17 void loop() {
18 //This line reads the voltage across the LDR and prints it out
19 Serial.print("The light level is at: "); Serial.println(analogRead(lightSensorPin))
20 delay(1000);
21 }
22
23 ////////////////////////////////////////////////////
24
```

Students will open the Serial Monitor in the upper right-hand corner. Here, students will see the light level increase or decrease based on the amount of light. The number that appears



will increase as the amount of light decreases. As students pass their finger over the light sensor, they should see the number change.

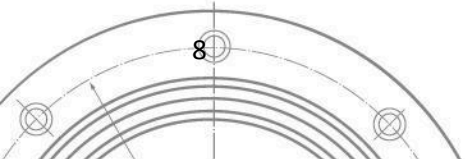


```
1 ////////////////////////////////////////////////////
2 //7.00 - Reading Light Intensity
3 // Onboard the MCU Trainer is a LDR.
4 // LDR stands for "light dependant resistor".
5 // This means that the LDR changes it's resistance based on the light applied to it.
6 // This makes it very easy to determine how much light is shining on the board.
7 // We can do that by doing an analogRead on the pin it is connected to.
8
9 byte lightSensorPin = A2;
10
11 void setup() {
12 //We are reading a voltage, so the pin is an input.
13 pinMode(lightSensorPin, INPUT);
14 Serial.begin(9600);
15 }
16
17 void loop() {
18 //This line reads the voltage across the LDR and prints it out
19 Serial.print("The light level is at: "); Serial.println(analogRead(lightSensorPin));
20 delay(1000);
21 }
22
23 ////////////////////////////////////////////////////
24
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Leonardo' on '/dev/cu.usbmodem1201') New Line

```
The light level is at: 446
The light level is at: 427
The light level is at: 421
The light level is at: 373
The light level is at: 430
The light level is at: 422
The light level is at: 422
The light level is at: 422
The light level is at: 424
The light level is at: 423
The light level is at: 424
The light level is at: 423
```





The screenshot displays the Arduino IDE interface for an Arduino Leonardo. The sketch file is named "0-Reading\_Light.ino". The code in the sketch is as follows:

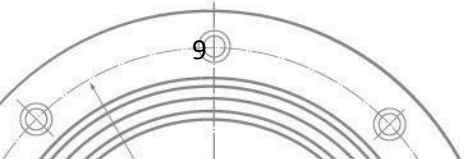
```
1 //////////////////////////////////////////////////
2 //7.00 - Reading Light Intensity
3 // Onboard the MCU Trainer is a LDR.
4 // LDR stands for "light dependant resistor".
5 // This means that the LDR changes it's resistance based on the light applied to it.
6 // This makes it very easy to determine how much light is shining on the board.
7 // We can do that by doing an analogRead on the pin it is connected to.
8
9 byte lightSensorPin = A2;
10
11 void setup() {
12 //We are reading a voltage, so the pin is an input.
13 pinMode(lightSensorPin, INPUT);
14 Serial.begin(9600);
15 }
16
17 void loop() {
18 //This line reads the voltage across the LDR and prints it out
19 Serial.print("The light level is at: "); Serial.println(analogRead(lightSensorPin));
20 delay(1000);
21 }
22
23 //////////////////////////////////////////////////
24
```

The Serial Monitor window shows the following output:

```
Message (Enter to send message to 'Arduino Leonardo' on '/dev/cu.usbmodem1201') New Line
The light level is at: 527
The light level is at: 531
The light level is at: 854
The light level is at: 889
The light level is at: 900
The light level is at: 902
The light level is at: 903
The light level is at: 902
The light level is at: 906
The light level is at: 905
The light level is at: 910
The light level is at: 912
```

**Modify the Sketch**  
–  
10 Minutes

Once students have successfully run the sketch, they will now make a modification to change the program. One possibility is to change the sketch to allow for all LEDs to light up after the LDR reaches a certain level.



```
Arduino Leonardo
0-Reading_Temperature.ino
9  byte lightSensorPin = A2;
10 byte LED1 = 13;
11 byte LED2 = 6;
12 byte LED3 = 7;
13 byte LED4 = 8;
14 void setup() {
15     //We are reading a voltage, so the pin is an input.
16     pinMode(lightSensorPin, INPUT);
17     pinMode(LED1, OUTPUT);
18     pinMode(LED2, OUTPUT);
19     pinMode(LED3, OUTPUT);
20     pinMode(LED4, OUTPUT);
21
22     Serial.begin(9600);
23 }
24
25 void loop() {
26     //This line reads the voltage across the LDR and prints it out
27     Serial.print("The light level is at: ");
28
29     int lightValue = analogRead(lightSensorPin);
30     Serial.println(lightValue);
31
32
33     if (lightValue > 500) {
34         digitalWrite(LED1, HIGH);
35         digitalWrite(LED2, HIGH);
36         digitalWrite(LED3, HIGH);
37         digitalWrite(LED4, HIGH);
38     }
39 }
```

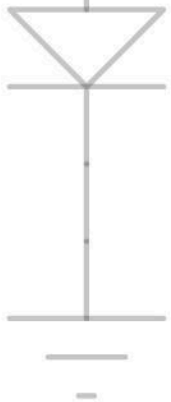
Output Serial Monitor x

Message (Enter to send message to 'Arduino Leonardo' on '/dev/cu.usbmodem1101')

New Line 9600 baud

```
The light level is at: 150
The light level is at: 150
The light level is at: 150
The light level is at: 150
The light level is at: 149
The light level is at: 149
The light level is at: 150
```

All LEDs should respond by lighting up when the LDR is covered.





**Career Exploration:**

- A good resource is the [IDOE Career Explorer database](#).
- Another good resource for career information is the Bureau of Labor Statistics

**Practical Application:**

The application of LDR's can be found in light-sensitive switches, burglar alarm systems where a beam of light appears when a sensor is triggered and in camera exposure.

**Additional Resources:**

- 1<sup>st</sup> Maker Space Frog Microcontroller Library
- 1<sup>st</sup> Maker Space Learn Arduino with the Frog Microcontroller

**Performance Assessment/Check for Understanding :**

- Was the student able to upload the sketch and use the Serial Monitor to see if the LDR is working?
- Was the student able to modify the sketch to show all LEDs lit as a response to the LDR light exposure?

Additional Feedback:

