Name: _____ Date: _____

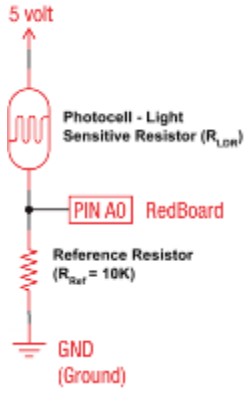
DIY Light Sensor

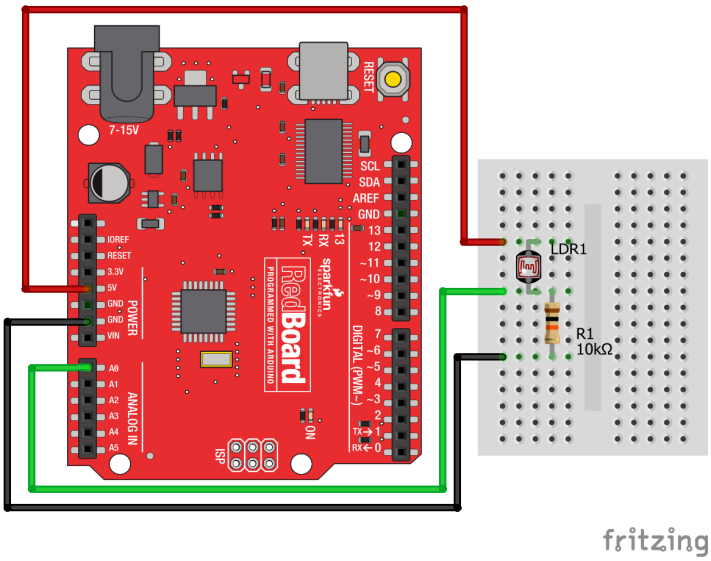
Introduction

This devices is called a photocell, or sometimes also called a light dependant resistor (LDR). THe material is made from Cadmium Sulfide (CdS), and decreases in resistance when exposed to light.

At full brightness, the resistance is ~100 Ohms. In a totally dark area (0 Lux), the resistance is ~8 - 10MOhms. This is quite a range - making this little sensor great for a variety of applications. For a majority of sensing applications, we are going to use a 10 kOhm pull-down resistor with the photocell to create a voltage divider circuit.

Circuit Diagram / Wiring





Write an equation for the voltage across the photoCell:

On the Analog Inputs of the Arduino, the microcontroller converts voltages to a numeric value (LSB[1]) from 0 to 1023. 5V = 1023 and 0V = 0. Revise the formula you have above to now be in units of (LSB). (Hint: the "voltage divider" ratio will come in handy).

Example Code

```
int rawAnalogReading;  // variable used to store the raw reading

/*************************************************************************/
void setup()
{
  Serial.begin(9600);
  Serial.println("Light Level Readings taken using Ardunio");
}

void loop()
{
  rawAnalogReading = analogRead(A0);  // reads in value from A0 [0-1023]
  Serial.println(rawAnalogReading);
  delay(500);  // short 0.5 second delay to slow down the loop for readability
}
```

So, that's it. Upload this and open up the Serial Monitor by clicking on the magnifying glass. 

The Serial Monitor is simply a terminal window that reports back any data it receives. The Arduino is currently printing (sending) data back to the computer. You should notice that the lights on the Arduino TX/RX lights blink back and forth.

What value does it show when you shine a light on it? Does this make sense? Work out the voltage divider equation and justify the results.

Code to Note

Using the Serial Monitor is a great way to debug a circuit or log readings and results back to a central computer. At the very top of the program inside the setup() function, we have a line that reads:

```
Serial.begin(9600);
```

This command starts up the Serial communication protocol on the Arduino. It sets everything up so that the Arduino can start sending and receiving data. Because the data comes across on just two wires (one for

[1] LSB stands for least-significant bit. It's really a unit-less quantity, but we need a name for the units.

transmit, and one for receive), the data rate must be agreed upon at the start. The 9600 indicates the data rate that the 1's and 0's will be transmitted (and received).

To send a line of text,

```
Serial.println("This is my text message back to the computer.");
```

This prints a line and adds a carriage-return / line-feed (moves the cursor to the next line). If you want to print multiple things right next to each other on the same line, we use a slightly different version of the same instruction:

```
Serial.print("This text stays on one line. ");
Serial.print("And, this will appear right next to it");
```

What's this good for? Well -- we can log light levels to the serial monitor for any given situation. Knowing the resistance ranges of the light detector, we can tailor the sensitivity for different scenarios.

Data Collection Example

download example code: https://codebender.cc/sketch:93683

```
/*****************************************************************************
 * Simple example of data collection using the photoresistor on an Arduino.
 *
 * This example prints out data to the Serial Monitor terminal window with a timestamp and
 * the raw analog value (0 - 1023) - uncalibrated. This data can be copied and pasted into
 * your favorite graphing, analysis, or spreadsheet program.
 *
 * After uploading this code, open up the Serial Monitor at 9600 baud.
 *
/*****************************************************************************/

int rawAnalogReading;  // variable used to store the raw reading

/*****************************************************************************/
void setup()
{
  Serial.begin(9600);
  Serial.println("Light Level Readings taken using Ardunio");
}

void loop()
{
  rawAnalogReading = analogRead(A0);  // reads in value from A0 [0-1023]

  Serial.print(millis());   // prints out the number of milliseconds fromt when
                            // the Arduino is powered up / program is run.
  Serial.print("\t");       // prints a tab character

  Serial.println(rawAnalogReading);

  delay(500);  // short 0.5 second delay to slow down the loop for readability
}
```