Arduino Beginners LDR Project

Wouldn’t it be cool if we could eliminate darkness? In this beginner arduino project, I have posted a very simple project that focuses on eliminating darkness.  
What it does is that- whenever a room gets dark due to a fused bulb or any other factors, a light bulb automatically turns ON. This can be used even as an emergency lighting system. It can be used to automatically turn a light ON whenever there isn’t sufficient light in a room.  
Here, in order to detect the intensity of light or darkness, we use a sensor called LDR (Light Dependent Resistor). The LDR is a special type of resistor which allows higher voltages to pass through it (low resistance) whenever there is high intensity of light. And passes a low voltage (high resistance) whenever it is dark. We can take advantage of this factor of the LDR and use it in our Arduino LDR sensor DIY project. Check this [link](https://www.kitronik.co.uk/blog/how-an-ldr-light-dependent-resistor-works/) to know more about LDR.

**What are the stuff required to do this project? Hardware:**

1. [Arduino](http://robokits.co.in/shop/index.php?main_page=product_info&cPath=6_72&products_id=388) or an arduino clone board ([freeduino](http://robokits.co.in/shop/index.php?main_page=product_info&cPath=6_72&products_id=388" \t "_blank)), or make your own custom arduino board with this [tutorial.](http://diyhacking.com/make-arduino-board-and-bootload/)
2. [LDR](http://www.myroboshop.com/product.php?id=MRBS137) (you can buy it online or from a local electronics store very cheaply)
3. 5V [SPDT Relay](http://www.tenettech.com/product/1655/relay-spdt-sealed)
4. 9V Battery and connector
5. Connecting wires
6. 100K resistor

**Software:**

[Arduino IDE](http://arduino.cc/en/main/software)

**How does it work?**  
This system works by sensing the intensity of light in its environment. Here, the sensor that can be used for detecting light is an LDR. You can buy it from any local electronics store or online, for a very cheap price.  
The LDR gives out an analog voltage when connected to Vcc (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, greater the intensity of light, greater will be the corresponding voltage from the LDR. Since the LDR gives out an analog voltage, it is connected to the analog input pin of the arduino. The arduino, with its inbuilt ADC (Analog to Digital Converter) then converts the analog voltage (from 0-5V) into a digital value in the range of (0-1023). Thus, when there is sufficient light in its environment or on its surface, the converted digital values read from the LDR through the arduino will be in the range of 800-1023. For more info on the LDR, check this out: [LDR/photoresistor](http://en.wikipedia.org/wiki/Photoresistor).

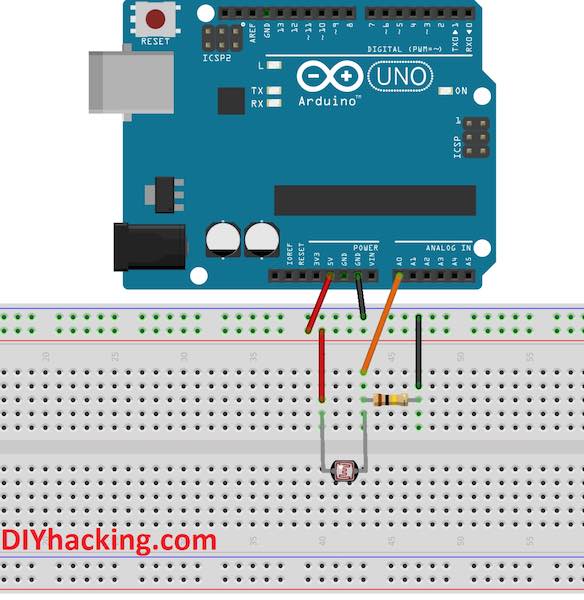
[](http://301o583r8shhildde3s0vcnh.wpengine.netdna-cdn.com/wp-content/uploads/2014/11/LDRblock.jpg)

Arduino LDR Sensor working

Furthermore, we then program the arduino to turn ON a relay. And correspondingly turn ON an appliance (light bulb), when the light intensity is low (can be done by covering the surface of the LDR with any object). That is, when the digital values read are in the higher range than usual.

**Step 1:  Arduino LDR Sensor Connections**

First of all, you need to connect the LDR to the analog input pin 0 of arduino. You have to use a voltage divider configuration to do this. The connection diagram for the arduino is as given below:

[](http://301o583r8shhildde3s0vcnh.wpengine.netdna-cdn.com/wp-content/uploads/2014/10/ConnLDR-min.jpg)

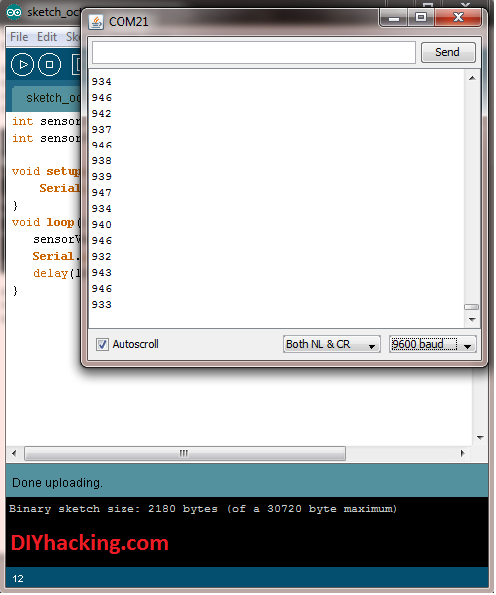
Arduino LDR connections

Here, one leg of the LDR is connected to VCC (5V) on the arduino. And the other to the analog pin 0 of the arduino. A 100K resistor is also connected to the same leg and grounded.

**Step 2: Testing the code for Arduino LDR Sensor**

Now, after connecting the LDR to arduino, we can check for the values coming from the LDR via arduino. For this, connect the arduino via USB to your PC and open up the arduino IDE or software. Now, paste this code and upload it to arduino:

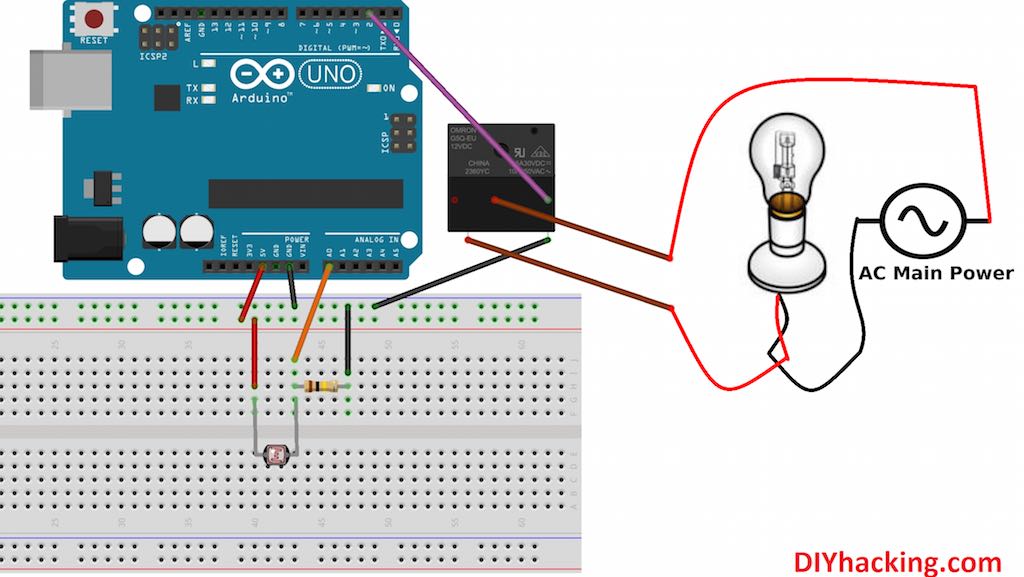
int sensorPin = A0; // select the input pin for ldr  
int sensorValue = 0; // variable to store the value coming from the sensor  
void setup() {  
Serial.begin(9600); //sets serial port for communication  
}  
void loop() {  
sensorValue = analogRead(sensorPin); // read the value from the sensor  
Serial.println(sensorValue); //prints the values coming from the sensor on the screen  
delay(100);  
}  
After uploading the code, click on the button on the arduino IDE called “Serial monitor”. This will open a new window, which prints different values on the screen. Now, test out the sensor, block its surface from light and see what all values you get on the serial monitor. This is how it looks:

[](http://301o583r8shhildde3s0vcnh.wpengine.netdna-cdn.com/wp-content/uploads/2014/11/serialMonitor.png)

Arduino LDR Sensor – Serial Monitor

**Step 3: Connecting the relay to arduino**

A relay is an electro mechanical switch; it can be used to turn ON/OFF an appliance working on AC/DC. Here, when the arduino supplies HIGH voltage (5V) to the relay, it turns it ON (the switch is ON), else it remains OFF. You can learn more about it in detail [here](http://www.learningaboutelectronics.com/Articles/How-to-connect-a-single-pole-double-throw-relay-in-a-circuit).  
In this project, we are using a 5V SPDT (Single Pole Double Throw) relay. One terminal of the relay coil is connected to arduino digital pin 2, and the other end to GND. We are connecting a light bulb to it as well. Since we are dealing with high power AC voltages, do take proper precautions. If you are still confused about connecting a relay to an appliance, check this [link](https://www.controlanything.com/Relay/Device/RELAY_LOGIC). The overall circuit is now as shown:

[](http://301o583r8shhildde3s0vcnh.wpengine.netdna-cdn.com/wp-content/uploads/2014/10/NoMoreDarkness-min.jpg)

Arduino LDR Sensor and Relay – Connection Diagram

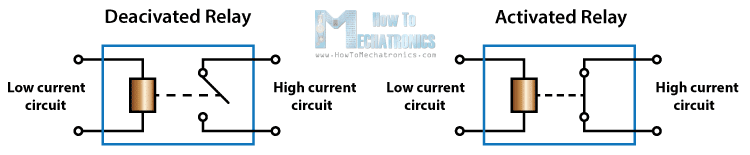
After connecting the arduino as shown above, next we need to test it by uploading the final code to arduino. The final sketch can be found here: [arduino code](http://diyhacking.com/projects/ArduinoNoMoreDarkness.ino" \t "_blank).

In this sketch, we have set a threshold light value as 700. But it can vary for your projects, you need to find out the particular value below which the light bulb should turn ON. This needs to be done after testing it empirically. So basically, the arduino turns ON the light bulb (via the relay) whenever the light intensity falls below 700. And when it is above 700, it turns the light bulb OFF. Finally a video showing it in action:

In this Arduino Tutorial we will learn how to Control High Voltage Devices using the Arduino Board. You can watch the following video or read the written tutorial below.

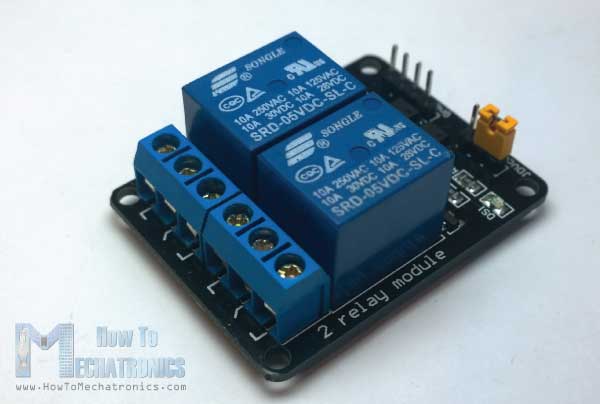
## Overview

We can control High Voltage electronic devices using relays. A Relay is actually a switch which is electrically operated by an electromagnet. The electromagnet is activated with a low voltage, for example 5 volts from a microcontroller and it pulls a contact to make or break a high voltage circuit.

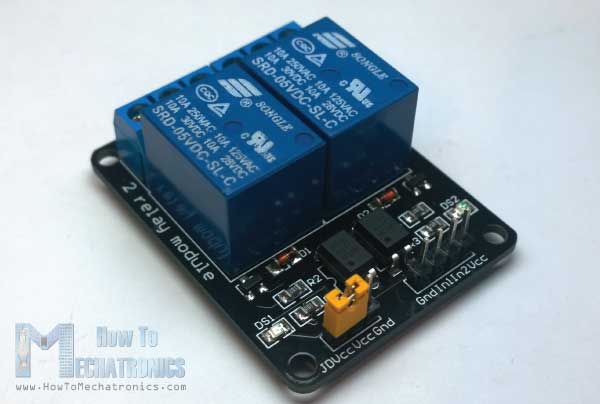


## HL-52S Relay Module

As an example for this Arduino Relay Tutorial we will use the HL-52S 2 channel relay module, which has 2 relays with rating of 10A @ 250 and 125 V AC and 10A @ 30 and 28 V DC. The high voltage output connector has 3 pins, the middle one is the common pin and as we can see from the markings one of the two other pins is for normally open connection and the other one for normally closed connection.



On the other side of the module we have these 2 sets of pins. The first one has 4 pins, a Ground and a VCC pin for powering the module and 2 input pins In1 and In2. The second set of pins has 3 pins with a jumper between the JDVcc and the Vcc pin.  With a configuration like this the electromagnet of the relay is directly powered from the Arduino Board and if something goes wrong with the relay the microcontroller could get damaged.



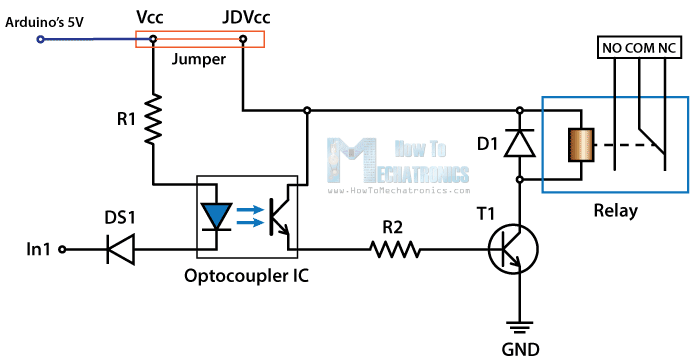
## Components needed for this tutorial

* 5V Relay Module – [[Get One](http://howtomechatronics.com/recommends/2-channel-relay-module-optocoupler-protection-bg/)]
* Arduino Board – [[Get One](http://howtomechatronics.com/recommends/arduino-mega-board/)]
* Cable, Plug, Socket

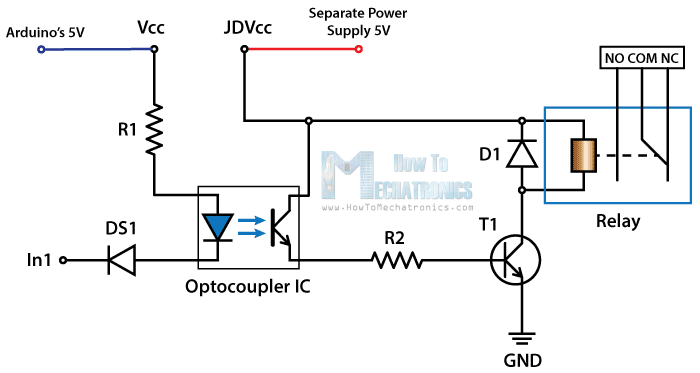
\*Please note: These are affiliate links. I may make a commission if you buy the components through these links.  
I would appreciate your support in this way!

## Circuit Schematic

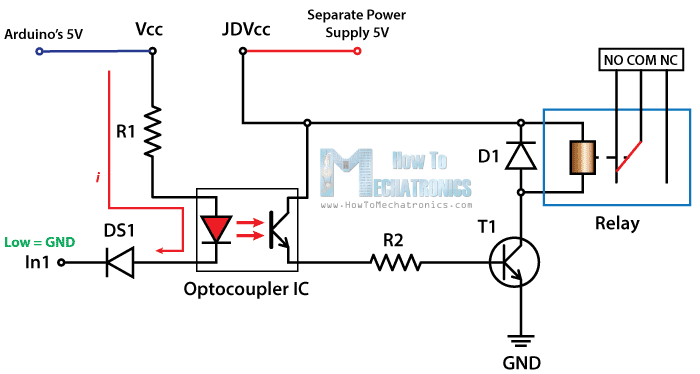
For better understanding let’s see the circuit schematics of the relay module in this configuration. So we can see that the 5 volts from our microcontroller connected to the Vcc pin for activating the relay through the Optocoupler IC are also connected to the JDVcc pin which powers the electromagnet of the relay. So in this case we got no isolation between the relay and the microcontroller.



In order to isolate the microcontroller from the relay, we need to remove the jumper and connect separate power supply for the electromagnet to the JDVcc and the Ground pin. Now with this configuration the microcontroller doesn’t have any physical connection with the relay, it just uses the LED light of the Optocoupler IC to activate the relay.



There is one more thing to be noticed from this circuit schematics. The input pins of the module work inversely. As we can see the relay will be activated when the input pin will be LOW because in that way the current will be able to flow from the VCC to the input pin which is low or ground, and the LED will light up and active the relay. When the input pin will be HIGH there will be no current flow, so the LED will not light up and the relay will not be activated.



## High Voltage Warning

Before we continue with this tutorial, I will warn you here that we will use High Voltage which if incorrectly or improperly used could result in serious injuries or death. So be very caution of what you are doing because I take no responsibility for any of your actions.



### How to use the relay module with the High Voltage devices

First let’s take a look at the circuit diagram. As previously described we will use a 5V Adapter as a separate power supply for the electromagnet connected to the JDVcc and the Ground pin. The Arduino’s 5V pin will be connected to the Vcc pin of the module and the pin number 7 to the In1 input pin for controlling the relay. Now for the HIGH Voltage part we need a power plug, a socket and a cable with two wires. One of the two wires will be cut and connected to the common and the normally open pin of the module output connector. So with this configuration when we will activate the relay we will get the high voltage circuit closed and working.

Here’s how made the cable. So I bought a plug, a socket and a cable. Then I carefully cut the cable and cut one of the wires as shown in the picture below and connect them to the normally open connection pins of the relay module. Also connected the ends of the cable to the plug and the socket.



\*Note: Make sure you use the other wires, not the “Yellow & Green” wire as it is meant to be used for Ground.

Here’s the final appearance of my cable ready for use. However before you use your cable make sure it’s working properly. You can check it using a multimeter or test it with a low voltage first.



## Source Code

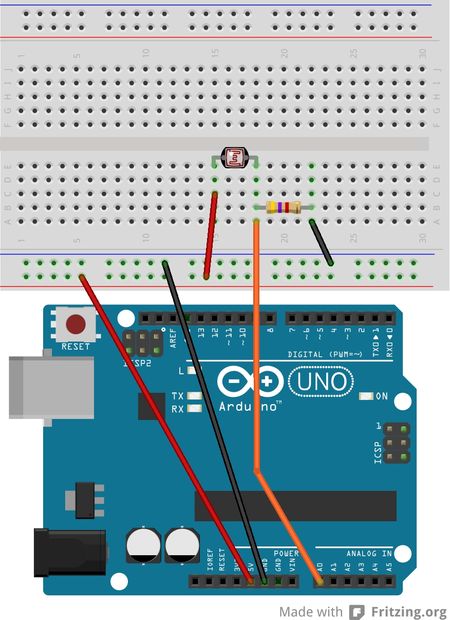
Now what’s left for this tutorial is to make a simple code and test relay module how it will work. Here’s the simple code, we will just use the pin number 7 for controlling the relay, so we will define it as output and make a program that will just activate and deactivate the relay each 3 seconds. I will mention once again here that the input of the module works inversely so a logic low at the input will actually active the relay and vice versa.

1. **int** in1 = 7;
2. **void** setup() {
3. pinMode(in1, OUTPUT);
4. digitalWrite(in1, HIGH);
5. }
6. **void** loop() {
7. digitalWrite(in1, LOW);
8. delay(3000);
9. digitalWrite(in1, HIGH);
10. delay(3000);
11. }

There is a demonstration of this example at the end of the video of this tutorial. I tested 3 devices on it. First a 100W light bulb, then a desk lamp and a fan heater. All of these devices work on 220V.

So that’s how we can control any High Voltage Device using Arduino or actually any other microcontroller. And of course the possibilities are now endless, for example we can control the devices using [TV Remote](http://howtomechatronics.com/tutorials/arduino/control-any-electronics-with-a-tv-remote-arduino-ir-tutorial/" \t "_blank), Bluetooth, SMS, Internet, and so on.

## Step 1: How it Works

[](http://www.instructables.com/file/F87P9PCHLSHJAAN/)

The top of the Potential Divider is 5V; the bottom is at 0V and the middle (connected to A0) is some value between 5V and 0V that varies as the LDR resistance varies. Remember the LDR resistance varies with Light so the Voltage at A0 will too.  
  
If you want to be very precise and technical then you can work out the Voltage on A0 as :  
  
Va0 = 5 \* R1/(R1+R2)  
  
where Va0 is the voltage at A0 pin, R2 is the top resistor value, R1 is the bottom resistor value;  
  
e.g. R1 = 10k, R2 = 5k => Va0 = 5 \* 10000/(10000 + 5000) = 5 \* 10/15 = 3.33V

The LDR has a high value when no light is present. The value of resistance of the LDR depends on the type. In this case it's about 10k. As the light level increases the resistance drops, which makes the current increase (by Ohm's Law), which in turn, makes the voltage at A0 (Va0) increase.

Electrically this is what's happening (skip this if you're not interested in this): The LDR and resistor are in series with the applied voltage (5V), so the current flowing through them is the same (the A0 pin draws virtually zero current). So the current through the resistor is (by Ohm's Law):

I = 5 / (R1+R2)

Now, the voltage across the resistor is applied to A0. Again, by Ohm's Law that is:

Va0 = I \* R1

Substituting the equation for I back in to this equation we get:

Va0 = 5 \* R1/(R1+R2)

See what it looks like on a Breadboard.

LDR SENSOR

**Arduino UNO Tutorial 8 - NightLight**

In this Arduino UNO tutorial, we are going to use a Light Dependent Resistor (LDR) to create a simple childrens bedroom nightlight which turns on automatically when it gets dark and turns off when it gets light.

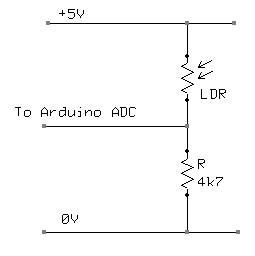
An LDR's resistance changes depending upon the amount of light hitting the sensor.  For the LDR we are going to use the resistance reduces as the light falling on the device increases. Used in conjunction with a 4.7K resistor this forms a simple voltage divider where the voltage across the LDR changes dependent upon the light.

We can then input this into one of the Analog to Digital inputs in the Arduino to measure the voltage. Then its a simple matter of checking whether the value is above or below a threshold value and to turn one of the outputs on or off.

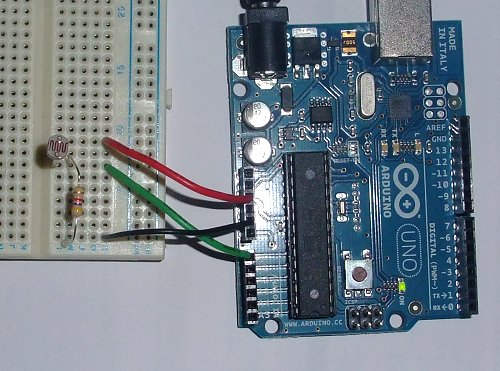
The LDR we are using is available [here](http://www.hobbytronics.co.uk/ldr)



The circuit diagram is shown below. As the light increases, the LDR's resistance drops and hence the voltage across it drops. Thus the voltage across the resistor increases, so the voltage into the Arduino ADC increases. The opposite is true as it gets darker.



Here is the circuit laid out on a breadboard. 5V and 0V are taken from the Arduino. The input goes to pin A0



Below is the Arduino Sketch. In this sketch we are simply turning on the built-in LED if the ADC value drops below a specific value. To make a nightlight, a brighter led (with limiting resistor ~220 ohms) can be connected to the pin 13 output.

In the code you will notice that there are some serial output statements that are commented out. If you uncomment these you will see on the serial monitor the current value of the voltage being read by the Arduino ADC input. This value is between 0 and 1024. Cover the LDR with your hand and shine a light on it to see the effect.

Change the value in the code where the LED is switched on to an appropriate value.

/\*

\*\* Nightlight LDR test program

\*\* Created 06 Feb 2010

\*\*

\*\* This example code is in the public domain.

\*\* www.hobbytronics.co.uk

\*/

int sensorPin = A0; // select the input pin for the ldr

unsigned int sensorValue = 0; // variable to store the value coming from the ldr

void **setup**()

{

  pinMode(13, OUTPUT);

  //Start Serial port

**Serial**.begin(9600); // start serial for output - for testing

}

void **loop**()

{

  // read the value from the ldr:

  sensorValue = analogRead(sensorPin);

  if(sensorValue<400) digitalWrite(13, HIGH); // set the LED on

  else digitalWrite(13, LOW); // set the LED on

  // For DEBUGGING - Print out our data, uncomment the lines below

  //Serial.print(sensorValue, DEC); // print the value (0 to 1024)

  //Serial.println(""); // print carriage return

  //delay(500);

}