

6. Measuring and Displaying Light Levels

What you will need:

Arduino UNO

CdS photoresistor

Mini breadboard

10 k Ω resistor (a smaller value may work better in some applications)

3 M/M jumper wires – red, black, and some other color

It's often useful to know whether it's light or dark or how much light is present. If an approximate determination of light levels will be enough, cadmium sulfide (CdS) photoresistors like the one in Figure 6.1 are simple passive devices that are very easy to interface with an Arduino. The resistance of these devices varies depending on how much light is falling on them. They are available with a range of resistance values, high when it's dark and decreasing as the intensity of light falling on them increases. They are widely available for around \$1. The housing of the one shown in the photo is about 0.5×0.7 cm. It has a resistance of more than 1 M Ω in total darkness and as little as around 100 Ω in very bright sunlight.



Figure 6.1. Cadmium sulfide (CdS) photoresistor.

Hardware layout:

Figure 6.2 shows the breadboard layout for this project. The photoresistor, in series with a 10 K Ω resistor, is powered from the 5 V pin, but you could also use the 3.3 V pin. Unlike the powered analog temperature sensor discussed previously in Chapter 5, this passive device can be connected in either direction to power and ground – like other resistors, there's no “plus” or “minus” orientation.

Recall that the maximum continuous current from a digital output pin should be no more than 20 mA. For the power pins, the maximum currents are much higher: 500 mA for the 5 V power pin and 200 mA for the 3.3 V pin when the UNO is powered through a USB cable. If the board is powered from an external source, you can draw more than 500 mA from the 5 V pin, but I do not recommend it. Drawing too much current from an external source will heat up the onboard voltage regulator and it will cut off the power until it cools down – not a good result!

It's unlikely that the resistance of a CdS photoresistor will ever be low enough to draw more than 500 mA through the 5 V power pin on its own, but that is beside the point for this project because a fixed series resistor is required not just to limit current flowing through the circuit in bright light, but to produce a signal that is proportional to the light falling on the photoresistor.

Suppose the fixed resistor is 10 k Ω and that in very bright light the photoresistor is 100 Ω . Then the total resistance in the circuit is 10.1 k Ω . The voltage across the fixed resistor is $5 \cdot 10,000 / 10,100 = 4.95$ V. Now suppose the photoresistor is 1 M Ω in total darkness. Then the voltage across the fixed resistor is $5 \cdot 10,000 / 1,010,000 = 0.0495$ V. This gives a very wide range of voltage values that are related in some nonlinear way to the amount of light falling on the photoresistor.

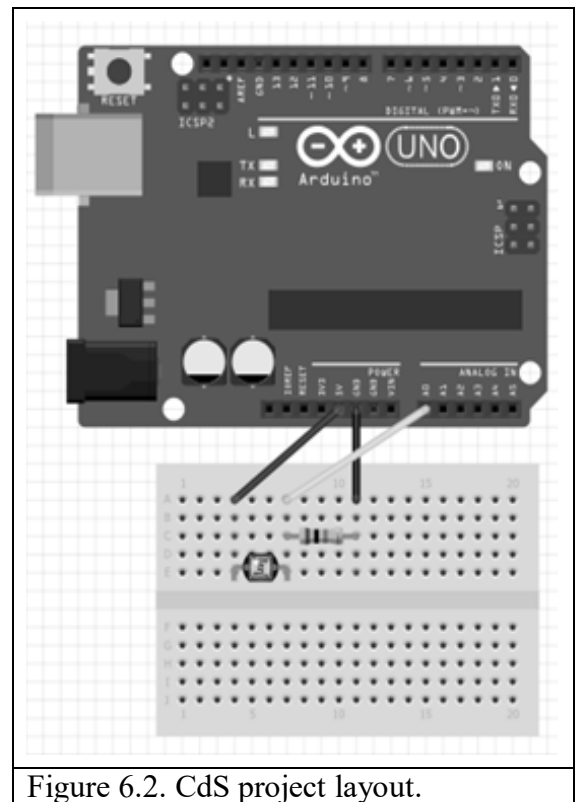


Figure 6.2. CdS project layout.

Writing the code:

You could certainly write some pseudocode or draw a code flow sketch as shown for the temperature sensor project in Chapter 5. But the basic code design for this project is exactly the same as for the project in Chapter 5:

1. Assign an analog input pin to read output across the resistor shown in the layout.
2. Optionally, convert the digitized voltage into an analog voltage.
3. Display the output in the serial port window.

In fact, the easiest way to write code for this project is not to start from scratch, but to open the temperature sensor code, rename and save it for this project, and make a few changes – in variable names, for example. If you wish to use the output across the resistor to take other actions, then more thought about the code design would be required.

Figure 6.3 shows the code with some output. The first couple of values result from shining a bright LED flashlight on the CdS resistor; its resistance is very low so almost all the voltage drop is across the 10 K resistor. The lowest values result from covering the resistor with my cupped hand (nearly total darkness) so the CdS resistance is high and much less of the voltage drop is across the 10 K resistor. Depending on your application, you might want to change the series resistor to provide a different range of voltage values in response to light. The only limiting factor is to make sure that the series resistor is large enough to limit the current drawn to acceptable levels in the brightest possible lighting conditions.

It might not be necessary to convert the digital values on pin A0 back to analog voltages. I've done this just to clarify the operation of the circuit; for example, if the resistance of the CdS resistor is 10 k Ω , then the voltage measured across the 10 k Ω resistor will be 2.5 V – half the total of 5 V across the circuit. If you want to control some other device based on the light level, it's probably just as useful to use the digitized input voltage without bothering to convert it to its analog voltage value.

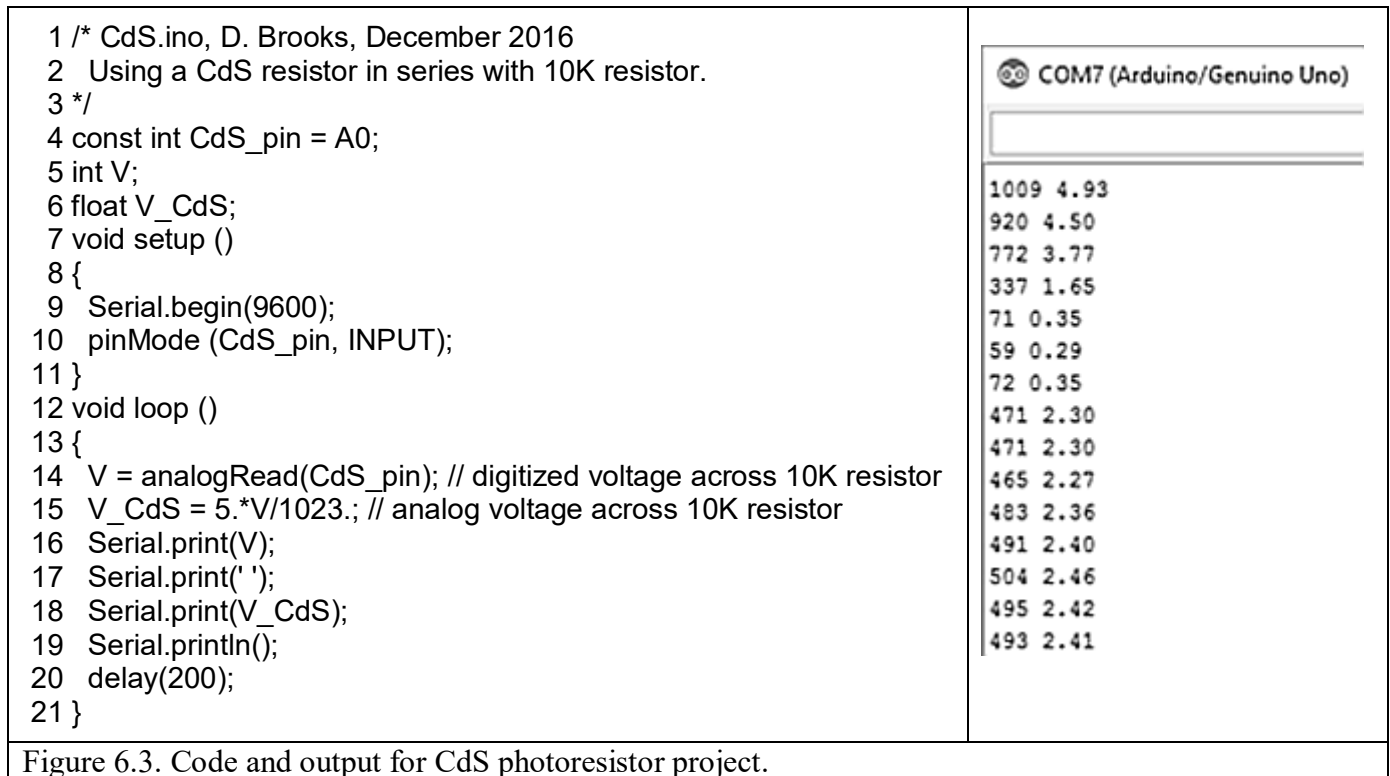


Figure 6.3. Code and output for CdS photoresistor project.

Additional project idea:

With a CdS photoresistor and a small laser, you can build an intrusion alarm. The components are shown in Figure 6.4. Caulk or epoxy the CdS photoresistor into one end of a piece of pipe – I used 1/4" (inside diameter) schedule 80 (thick walled) PVC plumbing pipe, but any small diameter pipe will work. The pipe keeps out stray light and the length of the pipe determines how critical alignment of the laser source is to the pipe. Small red laser modules operating on from 3 to 5 V are widely available online for less than \$1 each. You can power them with any appropriate DC source, including the 3.3 V power pin on an Arduino board. Don't use the 5 V power pin unless the laser module is specifically rated for this voltage. You could also use a handheld battery-powered laser pointer for a more portable light source, but a handheld device may be difficult to point accurately down a long tube several feet away.

As for the "alarm" part of the project, you could use something quiet and unobtrusive like an LED or a very loud and obnoxious buzzer like the ID 1536 from www.adafruit.com (\$1). Those buzzers are low-current devices so you turn them on and off by setting a digital pin HIGH or LOW without needing a current limiting resistor.

Your code should look at the digitized voltage from a fixed resistor in the CdS photoresistor circuit and respond accordingly to a threshold value based on whether the photoresistor is illuminated or not. (See the discussion of `if...` statements in Chapter 7.) Once you've assembled your alarm, you will probably have to send some output to the serial port window to figure out a good value for the threshold.

If you don't put a `delay()` statement in your `loop()` function, the code will respond essentially instantaneously to the illuminating beam being blocked. With data logging hardware and more code (see Chapter 10), you could use this alarm to monitor and record the passage of animals, for example.

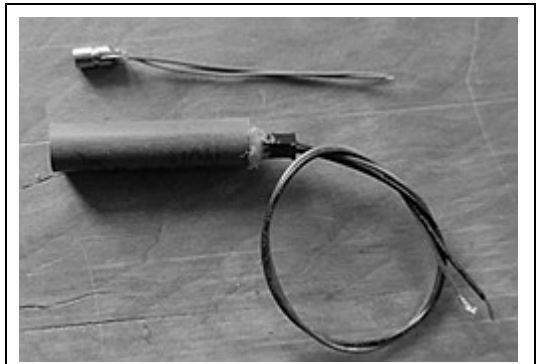


Figure 6.4. CdS photoresistor inside a tube and a small red laser, for intrusion alarm.