Module: Resistive Sensors

Introduction

In your Sparkfun kit, among other parts and components, you will find a plastic bag full of a variety of smaller devices. Many of the devices in the bag are sensors – devices that react to their environment by transforming a movement, a change in temperature, or a change reflectivity of a surface into a voltage. This voltage or signal can be used in a circuit to perform an operation based on the output voltage of the sensor. In this module you will learn to characterize and use a subset of these sensors – those that react to their environment by changing resistance. Other sensors use diodes, transistors, or special material properties to work but a large class of sensors simply change resistance in response to external stimuli. In our kits we have several resistive sensors – see table below.

My bag of sensors

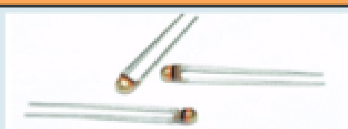
	sensor	description
	thermistor	Changes resistance in response to changes in temperature
	Flex sensor	Changes resistance in response to changes in the amount that the sensor is bent
	photocell	Changes resistance in response to changes in the amount of light hitting the sensor

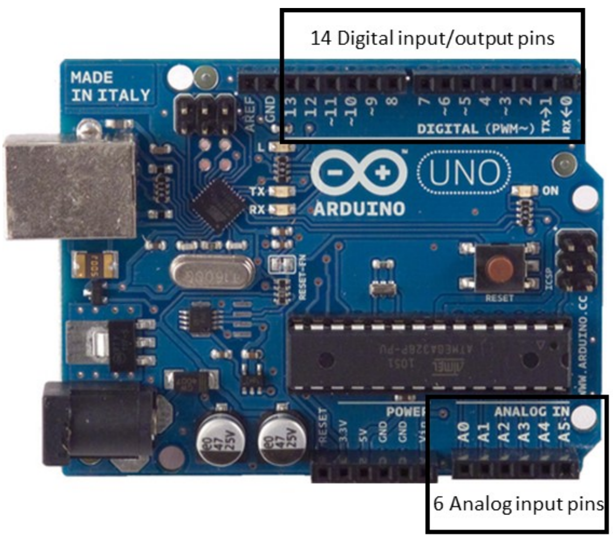
Table showing the resistive sensors in the Sparkfun kit

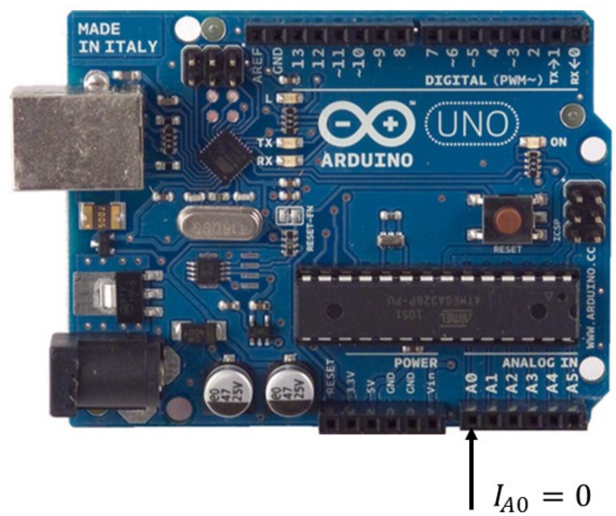
Notes:

After you have characterized the sensors it is time to connect them to a control circuit. This module leads you through the process of interfacing a resistive sensor to the Arduino/RedBoard.

Designing Circuitry Needed to Utilize the Photocell in a Project

For most simple designs the resistive sensors are connected to well-designed circuitry that responds to changes in voltage without drawing any current. This means that the behavior of any circuit connected to it is not modified. The Arduino is intended to behave this way. The figure below (left) shows the Arduino Uno and indicates which pins can be used to input voltage. It is the A0-A5 pins that you will be using for most of your sensor interfaces because these pins can read a voltage that varies from 5-0 V. Through an onboard device call an Analog-to-Digital converter the continuous voltage levels are mapped onto an integer value that varies from 0 at 0 V to 1023 at 5 V. The other pins are digital inputs implying that the board only recognizes only 2 voltage ranges designated 'HIGH' and 'LOW'. You will start by connecting your sensor circuit to the pin labeled A0. As shown in the figure below (right) you will assume that the Arduino is perfect and that the circuitry inside the board draws no current.



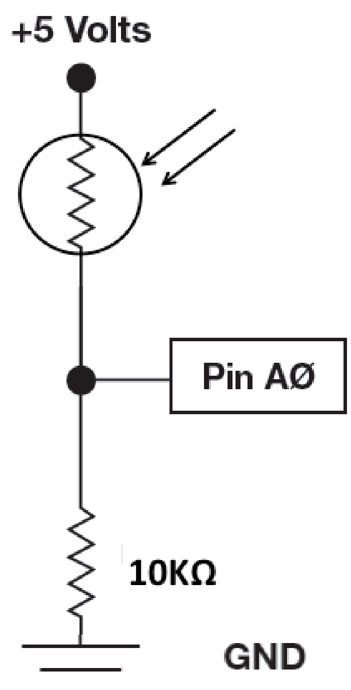


All the signal inputs are designed to draw **no current** from the circuitry connected to it. Like an ideal measuring device the Arduino is designed NOT to modify the behavior of the connected circuitry.

Notes:

Circuit interfacing the Photocell and the Arduino

The general approach to designing a very simple circuit using a restive sensor is to design a voltage divider circuit that attaches to one of the anaolog pins of the Arduino – in this case Pin A0. If you look in the SIK manual at Experiment 6 you will see that the photocell is used. In this experiment the photocell is used to turn on and off an LED. It is recommended that you work through this experiment as it also generalizes from the photocell to resistive sensors as a class of devices. The circuit used to connect the photocell to the Arduino is shown below.



Notes:

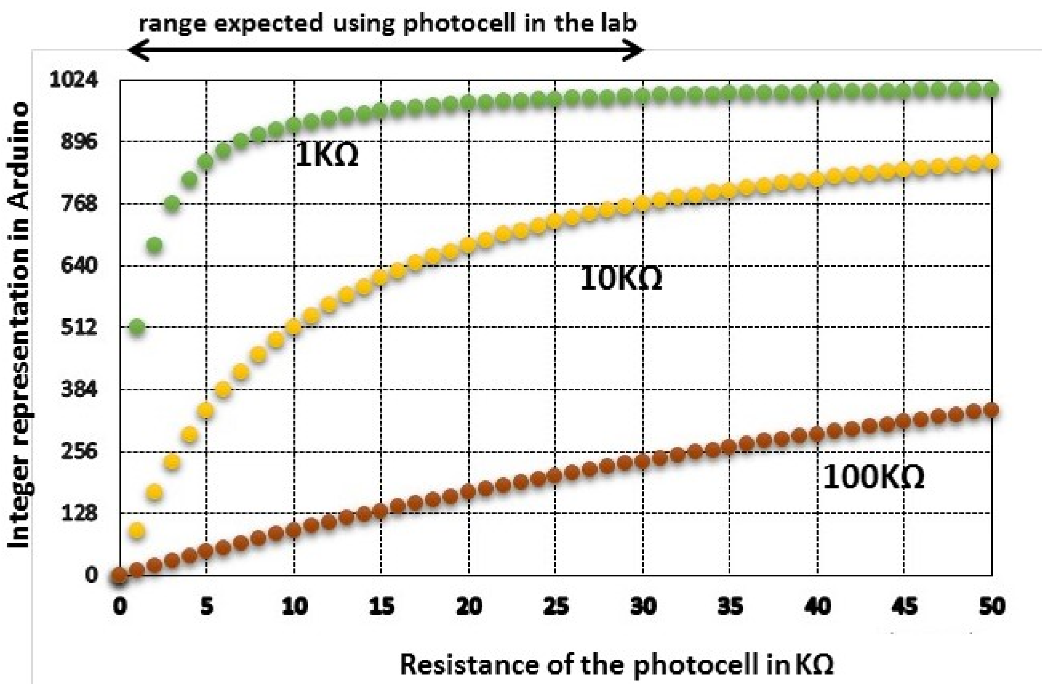
As you can see the photocell is connected in series with a 10KΩ resistor – let's call it R_1. The voltage across the 10KΩ resistor is connected to Pin A0 of the Arduino. Using the digitized data from Pin A0, code running on the Arduino can respond to the proximity of a light source. The folks at Sparkfun designed the voltage divider circuit knowing something about how the photocell works – this is why they chose a 10KΩ resistor. Let's see why they might have made that choice. The important things to remember are: i) that in the dark the resistance of the photocell is very high >1MΩ, ii) in dim light – conditions similar to those in the lab – the resistance is ~8-20KΩ, iii) the resistance lowers as more light hits the photocell.

Question 1: What are the extremes? If the resistance of the photocell varies between 1MΩ (essentially an open circuit) and 0Ω (a short circuit) what is the variation in voltage at Pin A0?

Question 2: This voltage range is largest range and is relatively independent of the choice of R_1 as long as R_1 is much less than 1MΩ. Why?

Notes:

Suppose the expected range of resistances encountered in the lab is ~100Ω (when the light source is very close to the photocell) to ~30KΩ (light source is far away and the photocell sees only the ambient light conditions). To have the greatest sensitivity to this range of light intensity you want the range of voltages generated when the photocell has this range of resistances to be as large as possible. You can optimize this range when choosing the value for R_1. The graph below shows how the resistance of the photocell maps onto the integer representation available to the Arduino code for 3 choices of R_1 – 1KΩ, 10KΩ, and 100KΩ.



Question 3: Using the graph or your own intuition explain why using the 10KΩ resistor as the value of R_1 is a decent choice?

Question 4: If the range of resistances obtained using another sensor is 100 – 5KΩ instead what might be a better choice for R_b?

Notes: