

03 - Sensing

Our first three projects all involved controlling LEDs. We showed you how to turn them on, off, and dim them.

Here we will introduce you to using Sensors. Most sensors are some type of variable resistor. So, let's start with one of those.



SKILL REQUIREMENTS

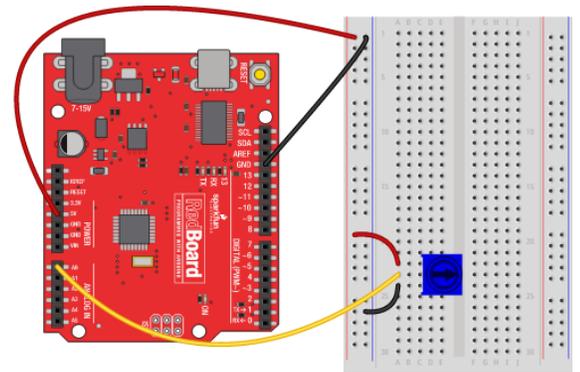
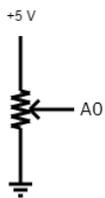
| | | | | | |
|------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| ELECTRICAL PROTOTYPING | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ROBOTICS | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SOLDERING | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| PROGRAMMING | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| DIY | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

MATERIALS LIST

- Computer
- Arduino UNO
- USB Cable
- LEDs
- Potentiometer
- Breadboard
- Light Sensor

STEP 1: Wiring / Background

Starting simple, let's find the blue potentiometer in your kit. It has 3 legs on it. Connect one of the outer legs to 5V, the other outer leg to GND, and the center leg A0 on the Arduino board. A potentiometer is a variable resistor. The resistance between the center leg and the two outer legs changes as you twist the knob. This circuit behaves like a simple voltage divider. As you twist the knob, the voltage on the center leg will also vary between 5V and GND.



STEP 2: Example Code

Open up this example code (codebender.cc/sketch:139963), and upload it to your board. After uploading, it click on the Serial Monitor button in codebender and click **[Connect]**. If you get errors, try using the Serial Monitor in the Arduino IDE as well. Simply open up Arduino and click on the magnifying glass in the upper right corner of the window. Now, twist the knob. Do you see values coming across your screen? This is the Arduino's reading of the voltage on pin A0. Now, let's look at the code. In the `setup()`, you see a new command:

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

This initializes the Serial communication on the Arduino. This is an interface that will allow us to send / receive data between the Arduino and another device (like your computer). The 9600 indicates the baud rate or speed for the data. There are other speeds available, but 9600 is one of the most commonly used.

In the `loop()`, we declare an integer variable called `val` and we assign it a value equal to `analogRead(A0)`; The `analogRead()` command reads the voltage on A0 and returns a value. The next couple lines of code are:

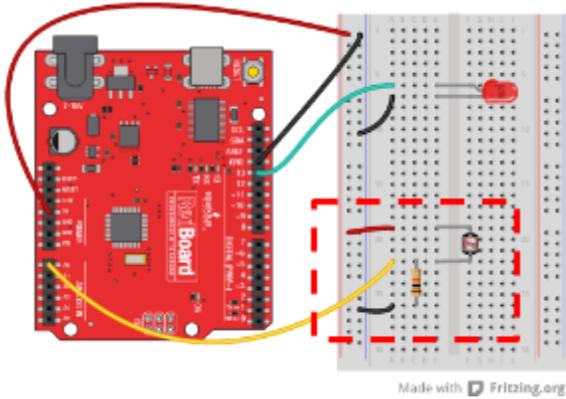
```
Serial.print("Val = ");  
Serial.println(val);  
delay(100);
```

The first line prints out the text: "Val = " to the Serial port using the `Serial.print()` command. The next line prints out the variable "val" with a new-line character - notice that this command is `Serial.println()`. This is what tells the Arduino to also send a new-line character. Finally, the `delay()` just slows things down a bit so that data isn't streaming across the screen at a crazy fast rate.

- Now, change the text around to read something else. What if you wanted to put units after the value? Add the text “raw units” after the value of the sensor is printed. Make sure that all of this text shows up on the same line.

The `Serial.print()` command is really useful for building troubleshooting routines or debugging your code.

STEP 3: Building a Night Light



This circuit is similar to the one above. But, instead of using a potentiometer, we are going to use a light sensor. The light sensor is really a light dependant resistor. Look at where the light sensor circuit is. Notice that one side of the circuit is connected to 5V and the other is connected to GND? This is that same “voltage divider” circuit.

Now, we also need a light. So, connect up an LED to pin 13.

After you’ve completed the circuit, find the example code at (codebender.cc/sketch:139983) and upload it to your board. Open up the Serial Monitor again and see what range of values the light sensor shows. Try covering up the sensor with your hand. Notice what the “average” value of the sensor is when the sensor is lit (i.e. not covered).

Write down your average light level (uncovered) here: _____

Write down your average light level (covered) here: _____

As you cover the light sensor up, does the value go up or down? _____

Let’s look at how the code works. Notice that we changed the name of the variable from “val” to “light”. It’s important to choose variable names that are descriptive and easy to remember. (Note: variable names must start with an alphabetical character and not contain any special characters or spaces.) Inside the `loop()`, you will see an `if-else` statement.

```

if(light < 500)
{
  digitalWrite(13, HIGH);
}
else
{
  digitalWrite(13, LOW);
}

```

STEP 4: Modify the code

The 500 is the “threshold” value that we based our night light. If the light is less than (<) 500, turn the LED on, else (otherwise) turn the LED off. Use your values from above to modify the threshold value for your night light. How can you make it so that it’s really sensitive?

Ideas / Challenge For Going Further

- Add more LEDs and use the light sensor to create a bar graph display.
- Use the light level to change the rate of a blink or a pattern.
- Use the `map()` function to scale your input values. Read more about `map()` on the Arduino resources page.



Visit us at: learn.sparkfun.com