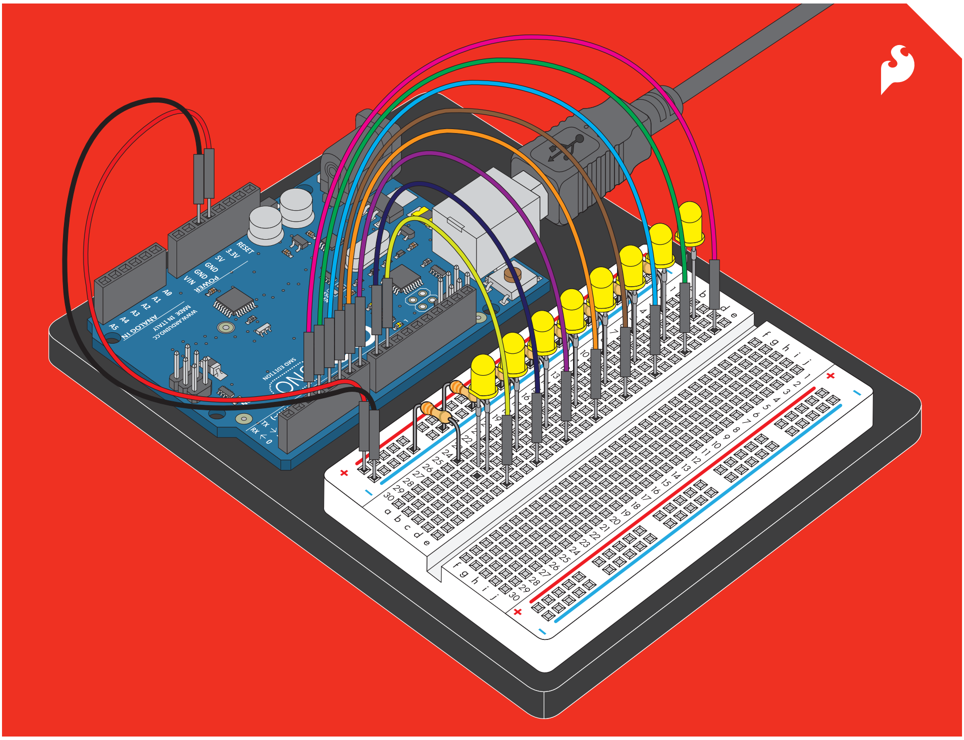
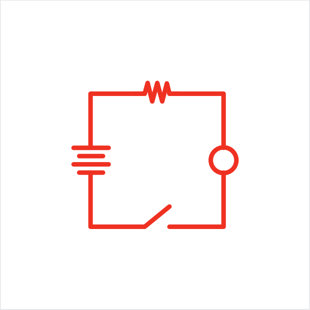


SIK GUIDE

Your Guide to the SparkFun Inventor's Kit for Arduino

Getting Started with Circuits



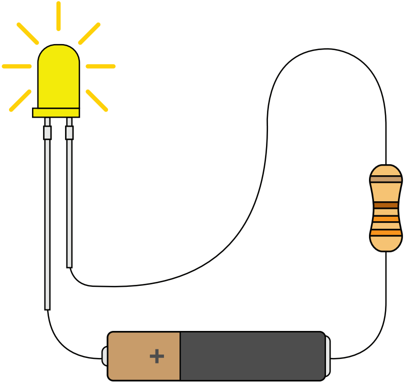
What is an Electrical Circuit?

A circuit is basically an electronics loop with a starting point and an ending point - with any number of components in between. Circuits can include resistors, diodes, inductors, sensors of all sizes and shapes, motors, and any other handful of hundreds of thousands of components.

Circuits are usually divided into three categories - analog circuits, digital circuits, or mixed-signal circuits. In this guide, you will explore all three sets of circuits.

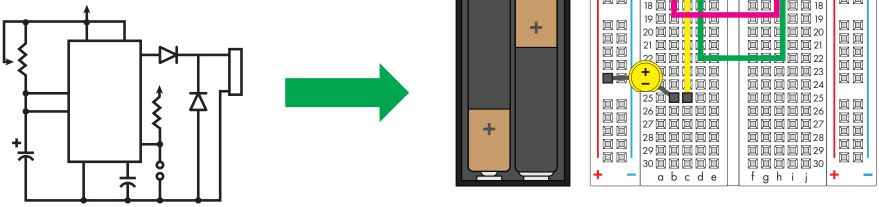
The World Runs on Circuits:

Everywhere you look, you'll find circuits. The cell phone in your pocket, the computer that controls your car's emissions system, your video game console - all these things are chock full of circuits. In this guide, you'll experiment with some simple circuits and learn the gist of the world of embedded electronics.

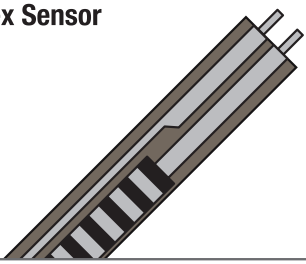


// Simple and Complex Circuits

In this guide, you will be primarily exploring simple circuits - but that doesn't mean you can't do amazing things with simple tools! When you've finished the SIK, your knowledge of circuits will enable you to explore amazing projects and unleash the power of you imagination.

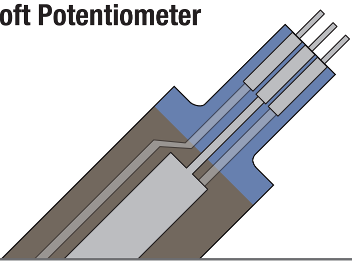


Flex Sensor



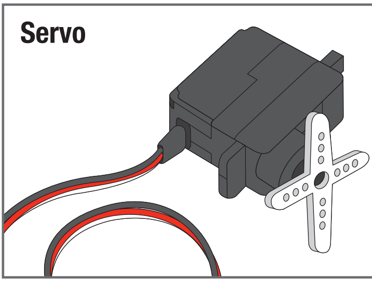
x1

Soft Potentiometer



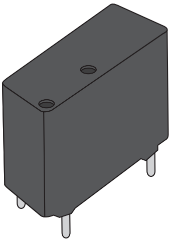
x1

Servo



x1

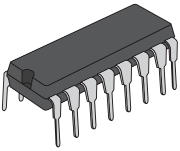
Relay

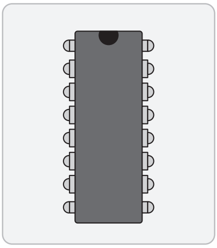


x1

Integrated Circuit

(IC)

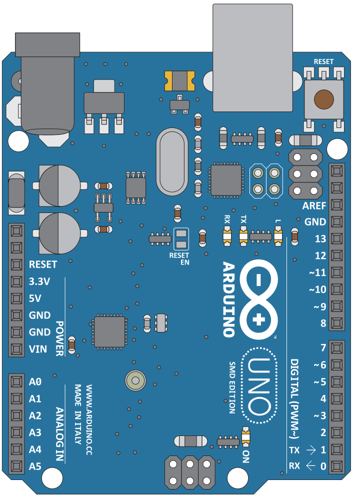




x1

Arduino Board

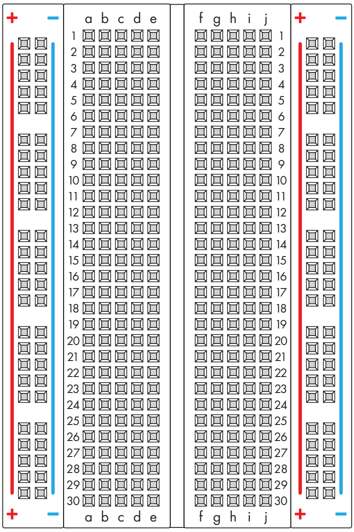
UNO - SMD Version



x1

Breadboard

Standard Solderless

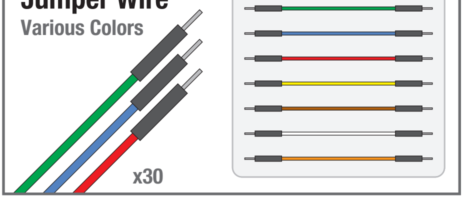


x1

Inventory of Parts

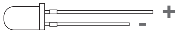
Jumper Wire

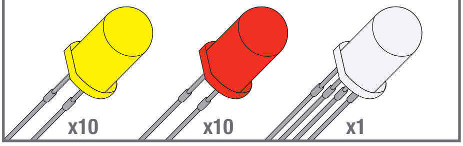
Various Colors



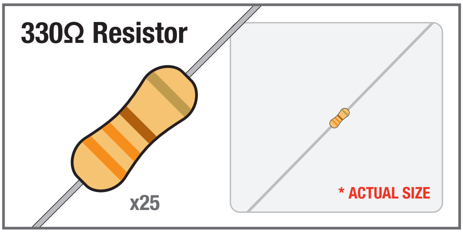
LED (5mm)

(Light Emitting Diode)

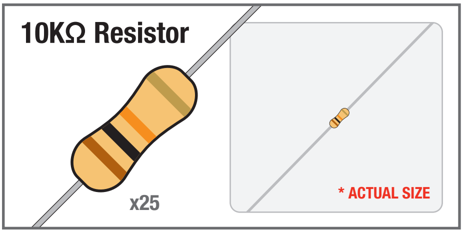




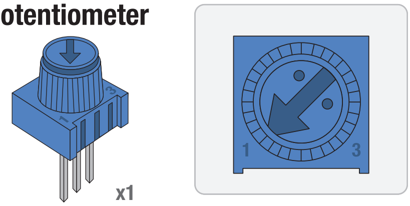
330Ω Resistor



10KΩ Resistor



Potentiometer



Diode

(1N4148)

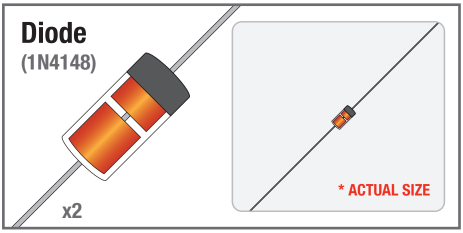
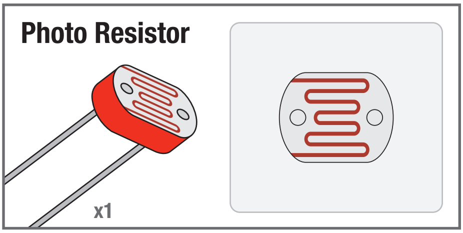
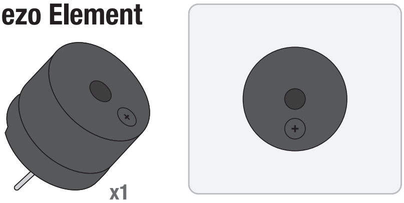


Photo Resistor

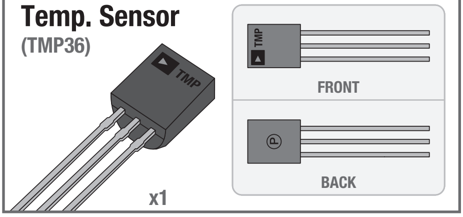


Piezo Element



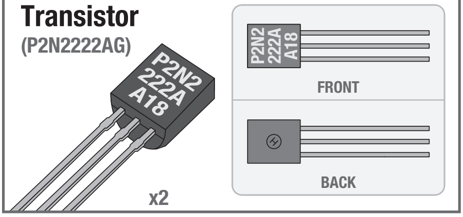
Temp. Sensor

(TMP36)

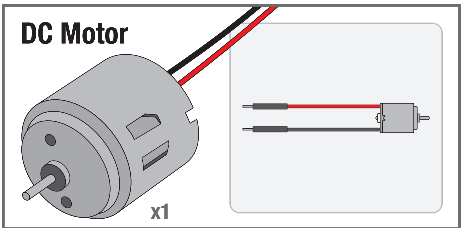


Transistor

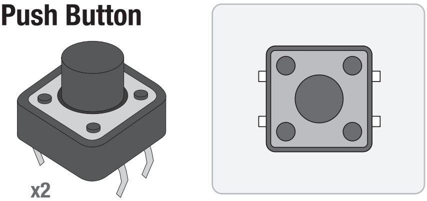
(P2N2222AG)



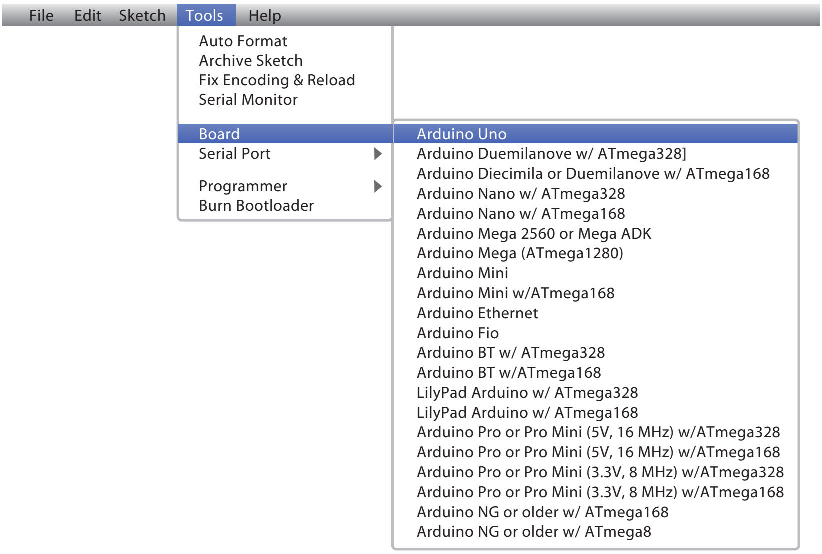
DC Motor



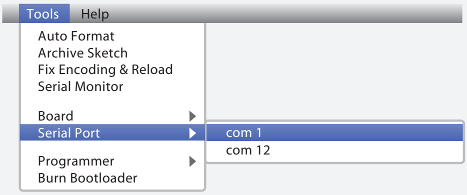
Push Button



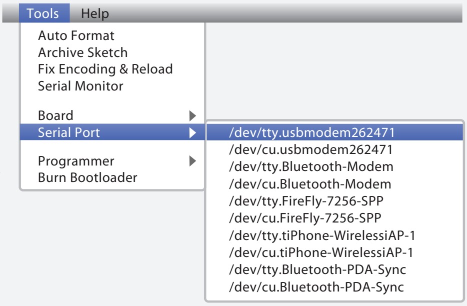
4 // Select your board: *Arduino Uno*


File
Edit
Sketch
Tools
Help
Auto Format
Archive Sketch
Fix Encoding & Reload
Serial Monitor
Board
Serial Port
Programmer
Burn Bootloader
Arduino Uno
Arduino Duemilanove w/ ATmega328]
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Mini w/ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8

// Select your Serial Device

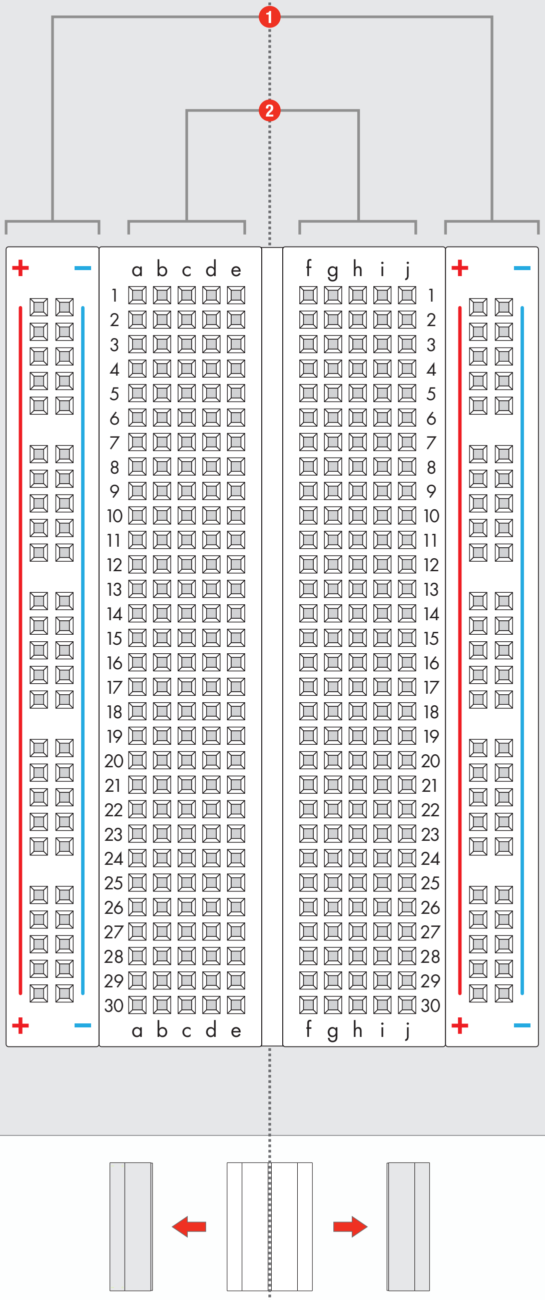


Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be **com3 or higher** (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with **/dev/tty.usbmodem** (for the Uno or Mega 2560) or **/dev/tty.usbserial** (for older boards) in it.

http://www.arduino.cc/playground/Learning/Linux

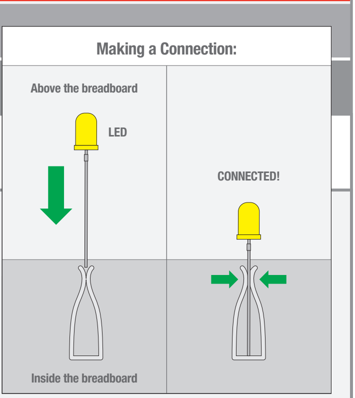


This line divides the board in half, restricting electricity to one half or the other.

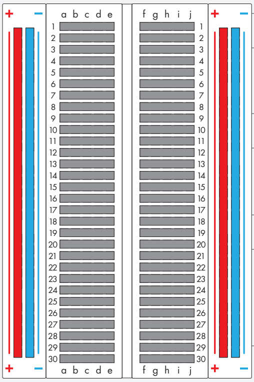
Breadboard

1 Vertical Connection (+ Power and - Ground // See Diagram Below)

2 Horizontal Connection (a-e & f-j // See Diagram Below)



How's it all connected?



+ Power:

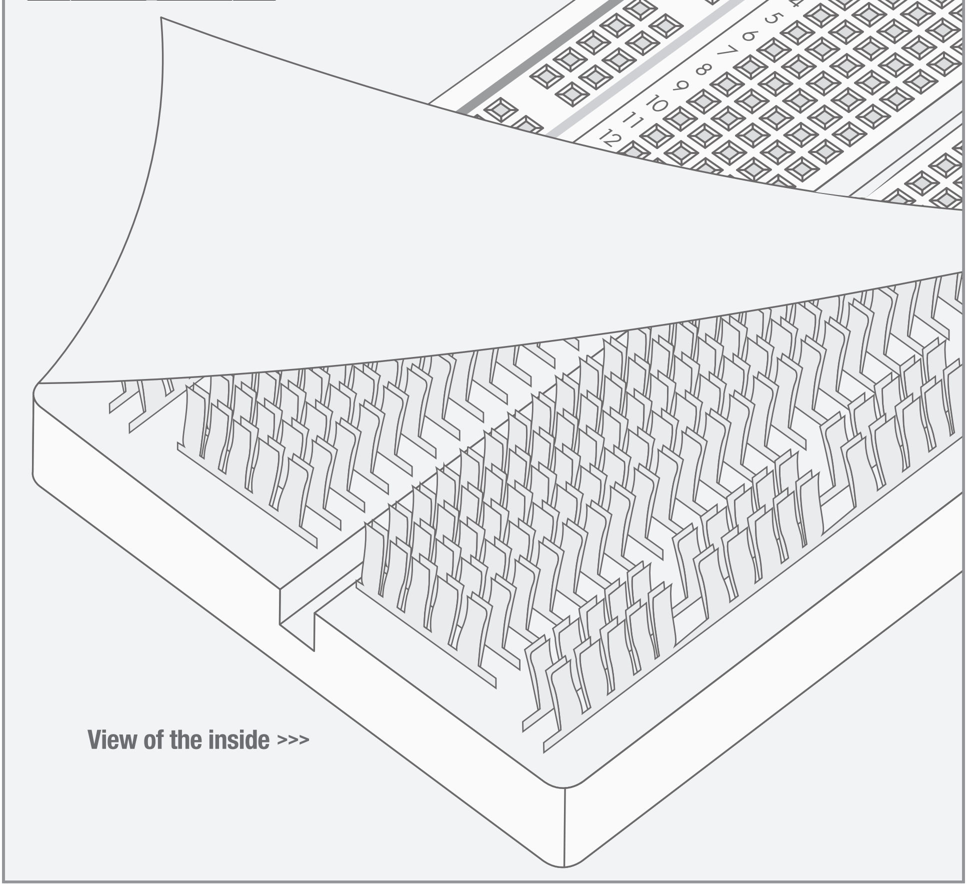
Each + sign runs power anywhere in the vertical column.

− Ground:

Each - sign runs to ground anywhere in the vertical column.

▬ Horizontal Rows:

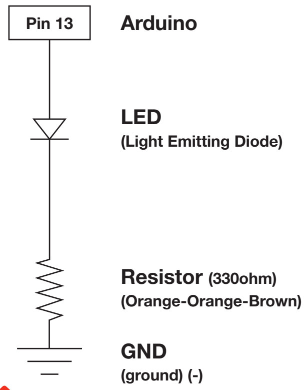
Each of these rows numbered 1-30 are comprised of five horizontal sockets. Components placed in the same row will be connected in a circuit when power is running.



View of the inside >>>

1

Blinking a LED



+ This is a schematic of your circuit.

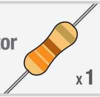
LEDs (light-emitting diodes) are small, powerful lights that are used in many different applications. To start off the SIK, we will work on blinking an LED. That's right - it's as simple as turning a light on and off. It might not seem like much, but establishing this important baseline will give you a solid foundation as we work toward more complex experiments.

+ Each Circuit begins with a brief description of the what you are putting together and the expected result.

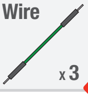
PARTS:

LED x 1

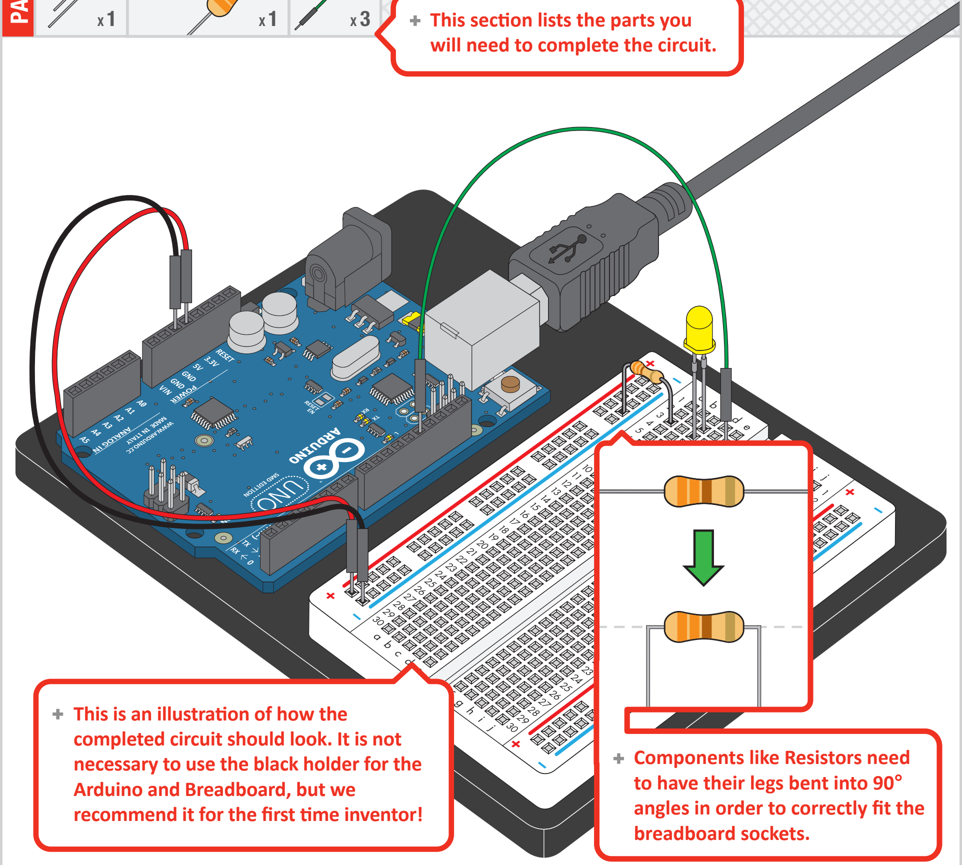
330Ω Resistor x 1



Wire x 3



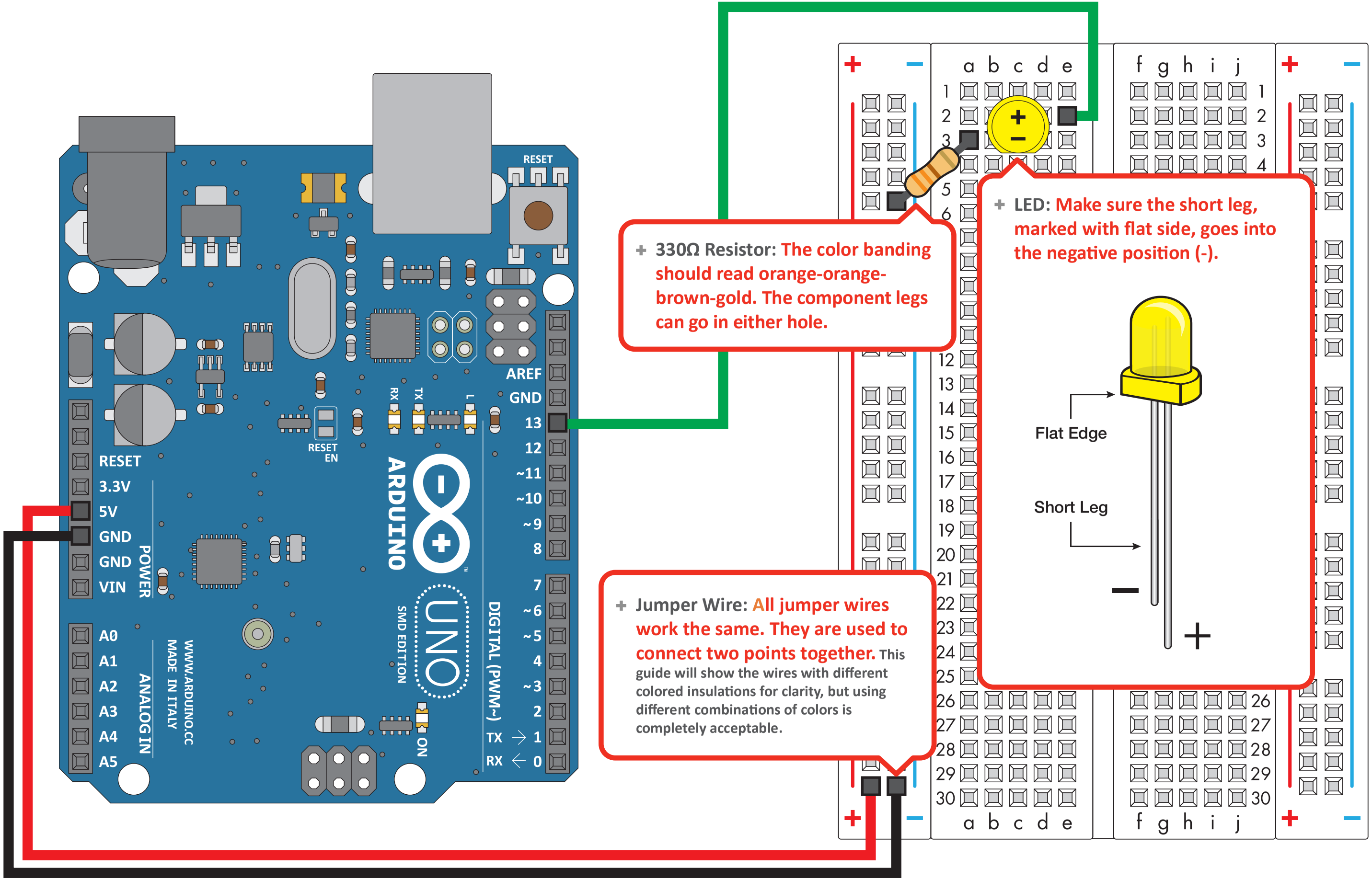
+ This section lists the parts you will need to complete the circuit.



+ This is an illustration of how the completed circuit should look. It is not necessary to use the black holder for the Arduino and Breadboard, but we recommend it for the first time inventor!

+ Components like Resistors need to have their legs bent into 90° angles in order to correctly fit the breadboard sockets.

Circuit 1: Blinking a LED



+ LED: Make sure the short leg, marked with flat side, goes into the negative position (-).

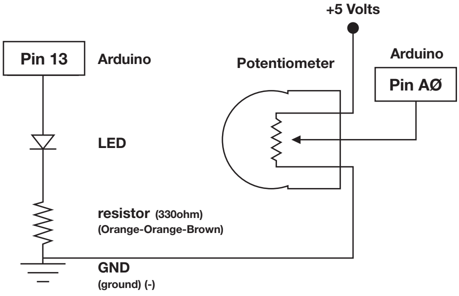
+ 330Ω Resistor: The color banding should read orange-orange-brown-gold. The component legs can go in either hole.

+ Jumper Wire: All jumper wires work the same. They are used to connect two points together. This guide will show the wires with different colored insulations for clarity, but using different combinations of colors is completely acceptable.

CIRCUIT #2

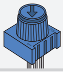
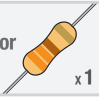
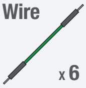
2

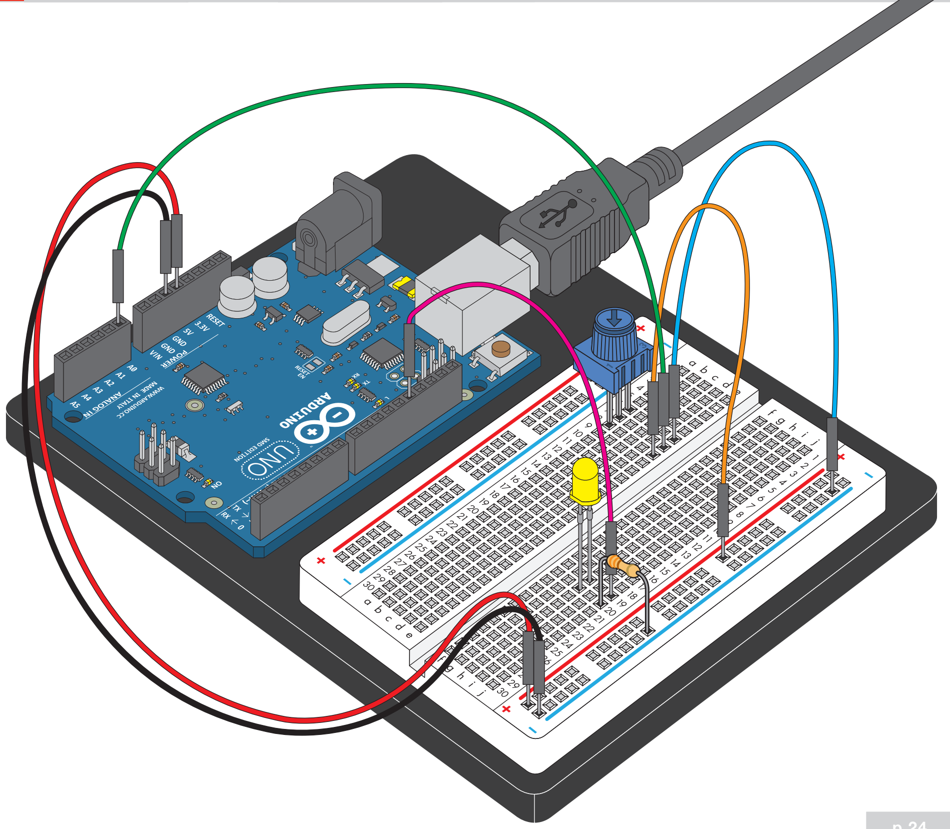
Potentiometer



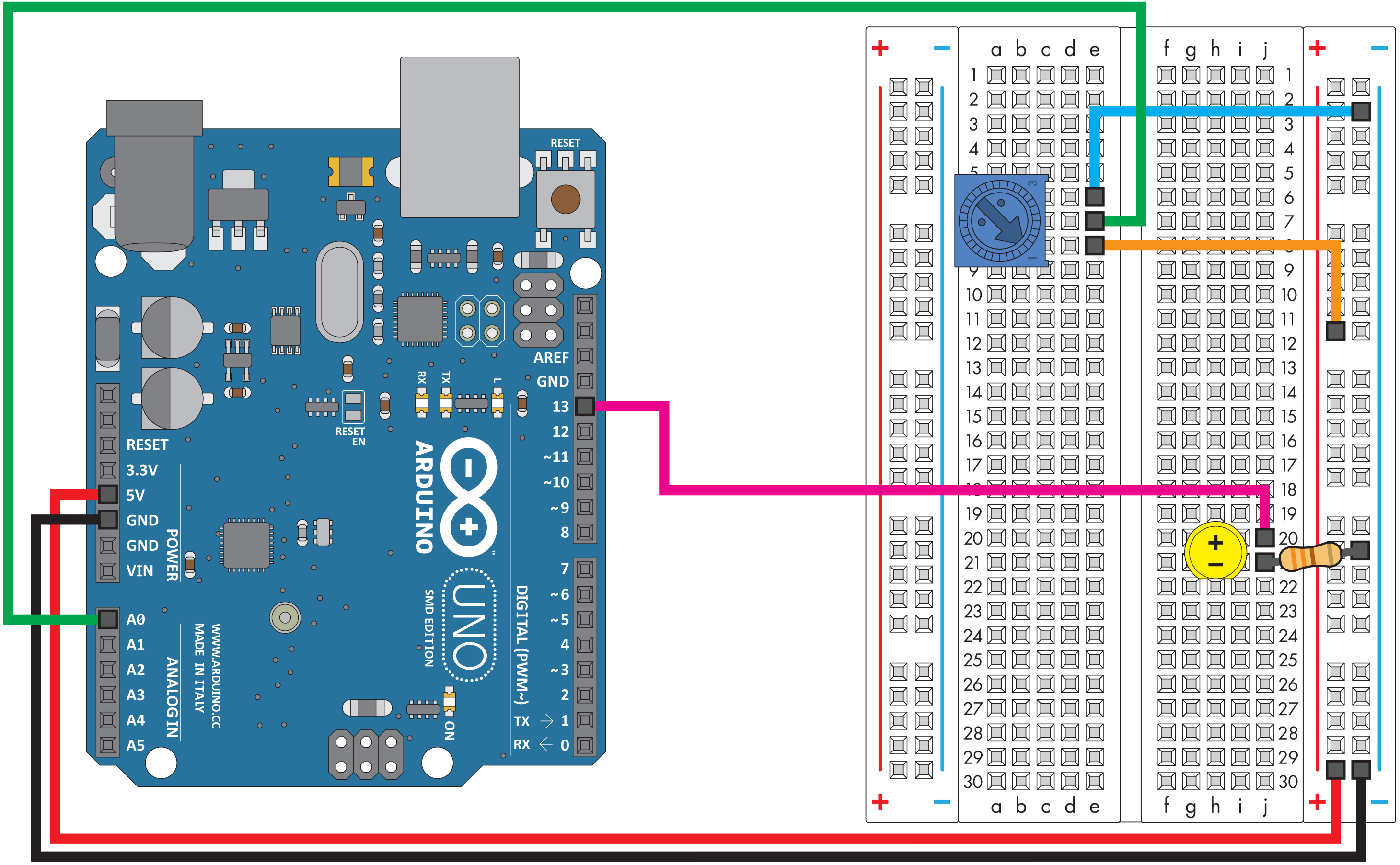
In this circuit you'll work with a potentiometer. A potentiometer is also known as a variable resistor. When it's connected with 5 volts across its two outer pins, the middle pin outputs a voltage between 0 and 5, depending on the position of the knob on the potentiometer. In this circuit, you'll learn how to use a potentiometer to control the brightness of an LED.

PARTS:

Potentiometer		x 1
LED		x 1
330Ω Resistor		x 1
Wire		x 6



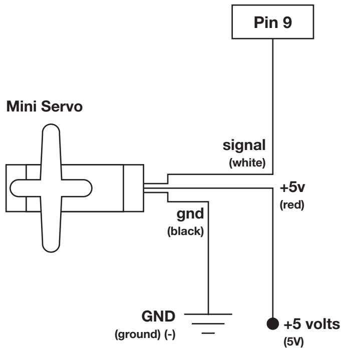
Circuit 2: Potentiometer



CIRCUIT #8

8

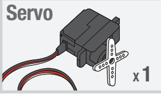
A Single Servo



Servos are ideal for embedded electronics applications because they do one thing very well that spinning motors cannot – they can move to a position accurately. By varying the pulse of voltage a servo receives, you can move a servo to a specific position. For example, a pulse of 1.5 milliseconds will move the servo 90 degrees. In this circuit, you'll learn how to use PWM (pulse width modulation) to control and rotate a servo.

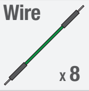
PARTS:

Servo

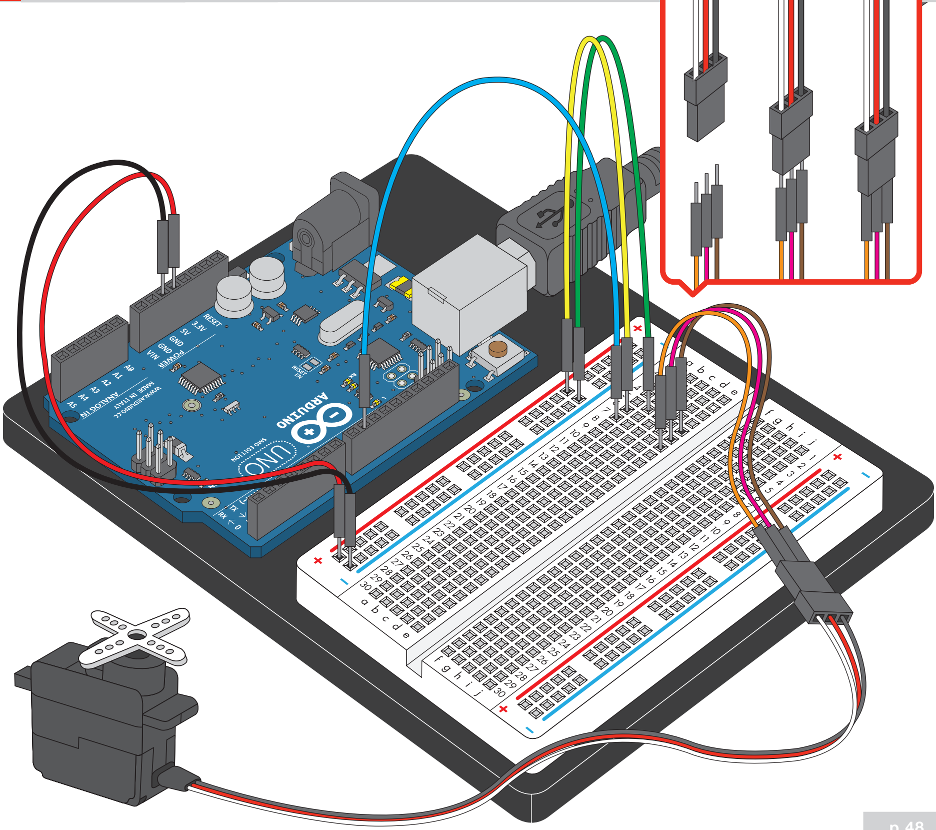


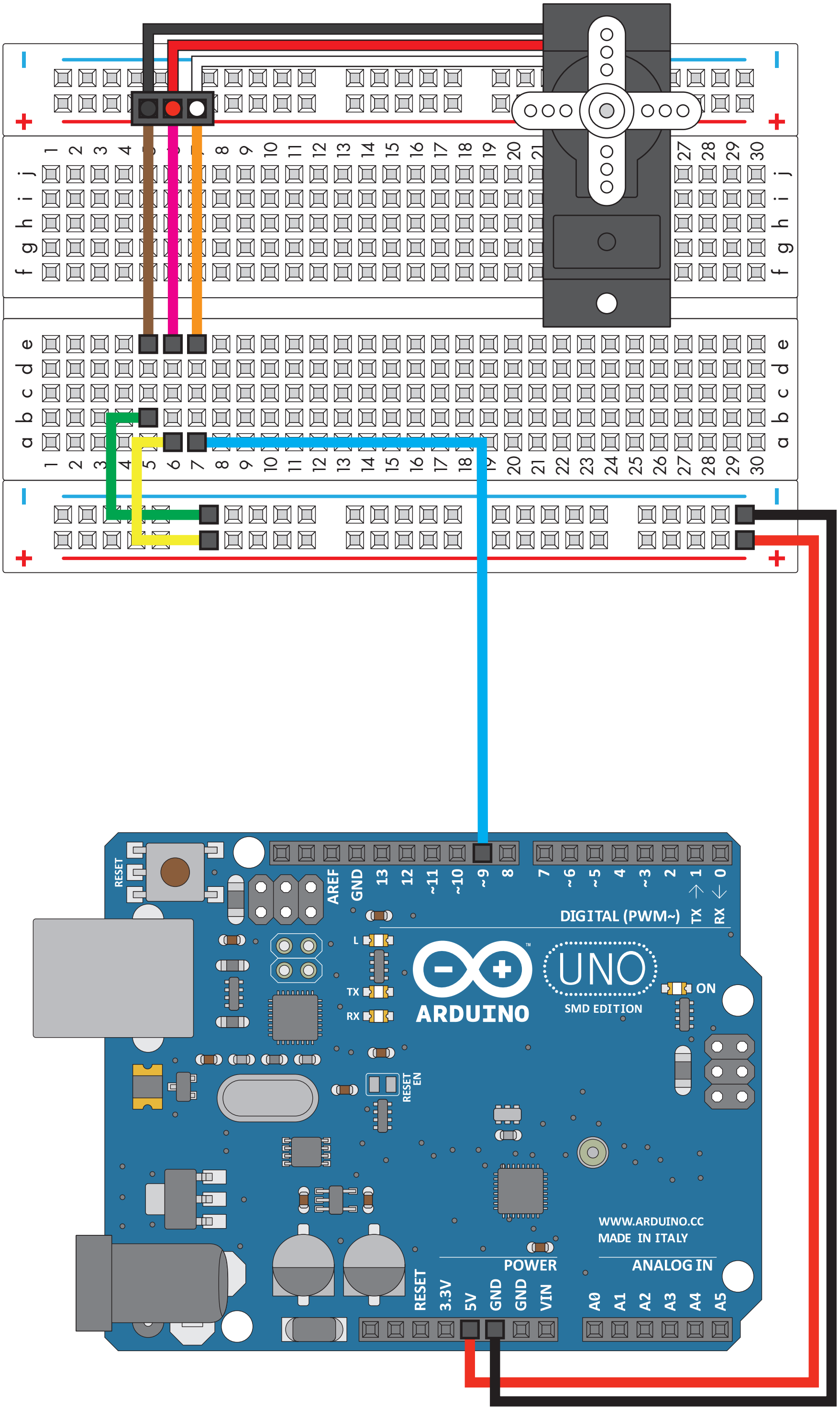
x 1

Wire



x 8





Circuit 8: A Single Servo

8

Arduino Code:

Open Arduino IDE // File > Examples > SIK Guide > Circuit # 8

Code to Note:

#include <Servo.h> ⇨ #include is a special "preprocessor" command that inserts a library (or any other file) into your sketch. You can type this command yourself, or choose an installed library from the "sketch / import library" menu.

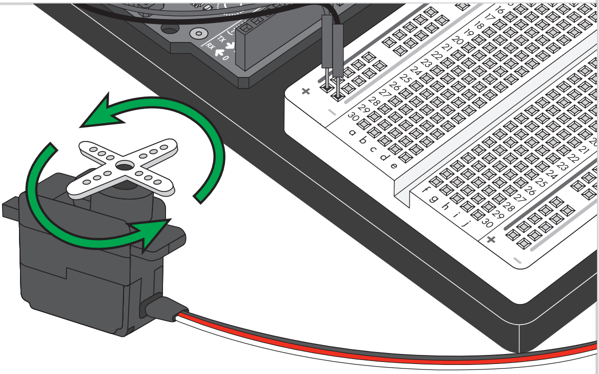
Servo servo1;

servo1.attach(9); ⇨ The servo library adds new commands that let you control a servo. To prepare the Arduino to control a servo, you must first create a Servo "object" for each servo (here we've named it "servo1"), and then "attach" it to a digital pin (here we're using pin 9).

servo1.write(180); ⇨ Servos don't spin all the way around, but they can be commanded to move to a specific position. We use the servo library's write() command to move a servo to a specified number of degrees(0 to 180). Remember that the servo requires time to move, so give it a short delay() if necessary.

What You Should See:

You should see your servo motor move to various locations at several speeds. If the motor doesn't move, check your connections and make sure you have verified and uploaded the code, or see the troubleshooting tips below.



Troubleshooting:

Servo Not Twisting
Even with colored wires it is still shockingly easy to plug a servo in backward. This might be the case.

Still Not Working
A mistake we made a time or two was simply forgetting to connect the power (red and brown wires) to +5 volts and ground.

Fits and Starts
If the servo begins moving then twitches, and there's a flashing light on your Arduino board, the power supply you are using is not quite up to the challenge. Using a wall adapter instead of USB should solve this problem.

Real World Application:

Robotic arms you might see in an assembly line or sci-fi movie probably have servos in them.