

ITP PHYSICAL COMPUTING

Servo Motor Control with an Arduino

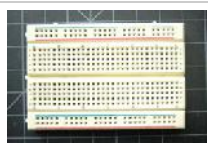
Overview

In this lab, you'll control a servomotor's position using the value returned from an analog sensor. Servos are the easiest way to start making motion with a microcontroller. Even though they don't turn 360 degrees, you can use them to create all sorts of periodic or reciprocating motions. Check out some of the mechanisms at [Rob Ives's](#) site for ideas on how to make levers, cams, and other simple machines for making motion.

(:toc Table of Contents:)

Parts

For this lab you'll need:



Solderless breadboard



22-AWG hookup wire



Arduino Microcontroller module



10Kohm resistors



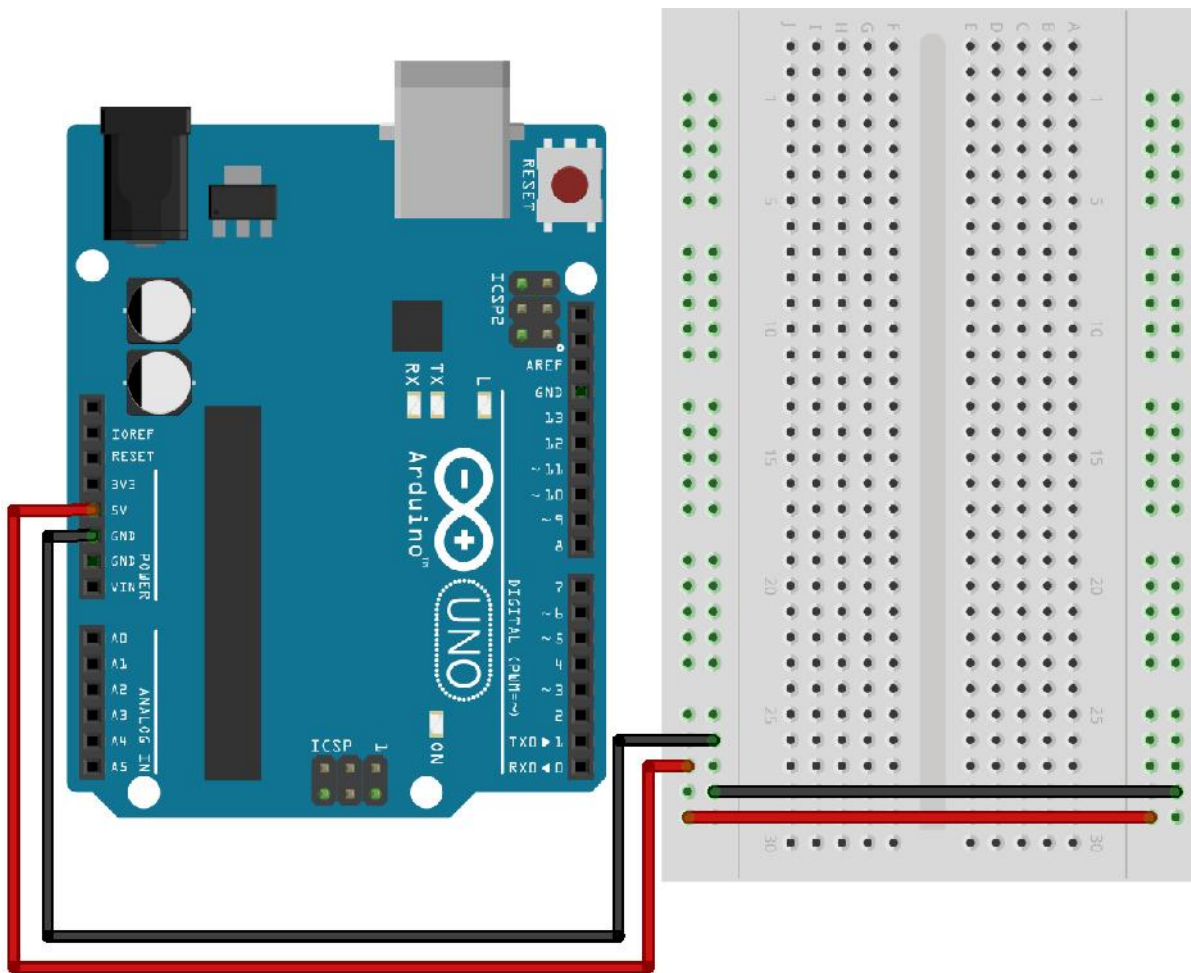
Flex sensors
(or a different
form of variable resistor)



RC Servomotor

Prepare the breadboard

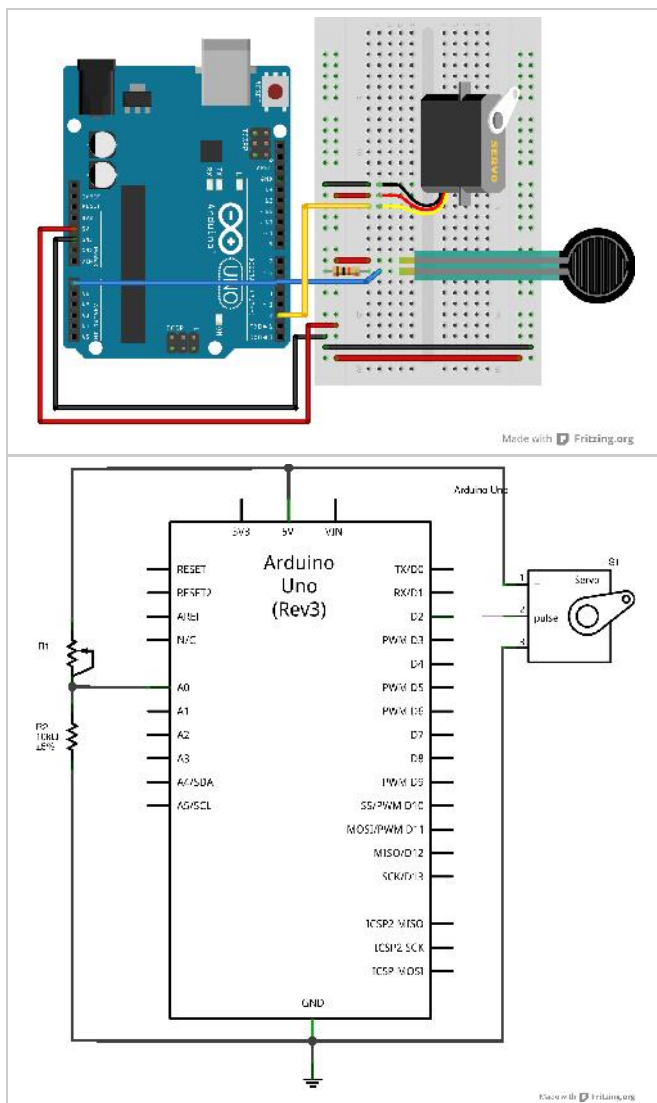
Connect power and ground on the breadboard to power and ground from the microcontroller. On the Arduino module, use the 5V and any of the ground connections:

Made with  Fritzing.org(Diagram made with [Fritzing](#) - download)

Connect an analog input sensor and a servo

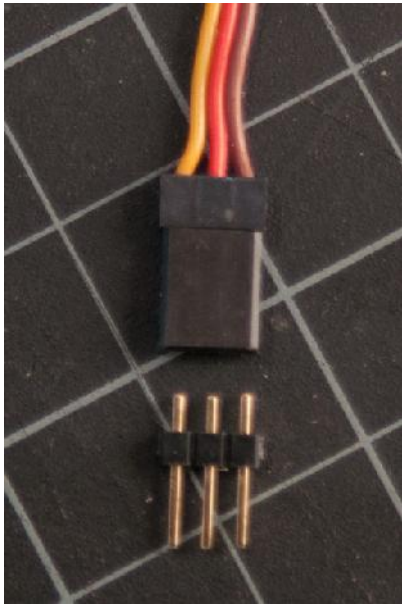
Pick any analog input and connect it to Analog pin 0 as you did in the [Analog Input and Output Lab](#). Then connect an RC servomotor to digital pin 2. The yellow wire of the servo goes to the pin, and the red and black wires go to +5V and ground, respectively.

Not all servos have the same wiring colors. For example, the Hextronik servos that come with Adafruit's ARDX kit use red for +5V, brown for ground, and mustard yellow for control.

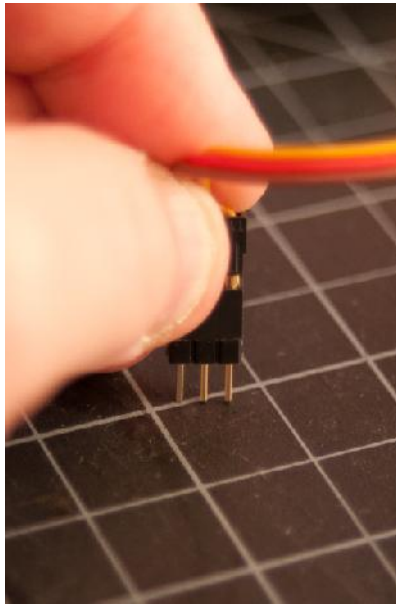


(Diagram made with [Fritzing](#))

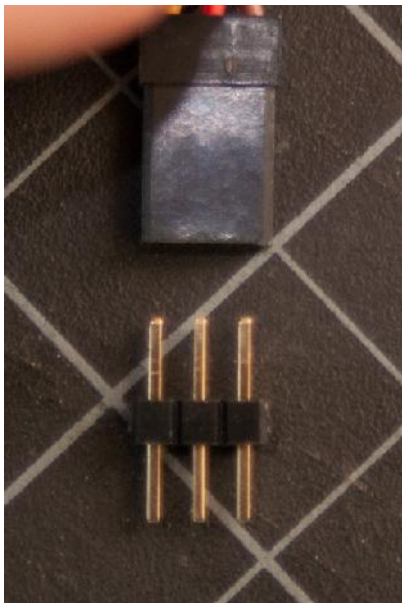
When you attach the servo, you'll need a row of three male headers to attach it to a breadboard. You may find that the pins don't stay in the servo's connector holes. Put the pins in the servo's connector, then push them down on a table gently. They will slide up inside their plastic sheaths, and fit better in your servo's connector.



header pins don't always
stay in a servo connector



Put them in and push gently
on a firm surface



Voila, your headers fit better!

Program the Microcontroller

First, find out the range of your sensor by using `analogRead()` to read the sensor and printing out the results.

Now, map the result of the analog reading to a range from 0 to 179, which is the range of the sensor in degrees. Store the mapped value in a local variable called `servoAngle`.

Finally, add the servo library at the beginning of your code, then make a variable to hold an instance of the library, and a variable for the servo's output pin. In the `setup()`, initialize your servo using `servo.attach()`. Then in your main loop, use `servoAngle` to set the servo's position.

Get Creative

This is just a suggestion for a short project. It's not a requirement for the class homework.

Servo motors give you the power to do all kinds of things.

They can be used to push a remote contro button, in a pinch:



([Project](#): Tom Igoe)

You can [play drums](#) (Project: Nick Yulman)

You can make a [rat's tail move](#) (project: Gabriela Gutiérrez)

If you've got 800 or so of them and a lot of time, you can build a [wooden mirror](#) (project: Daniel Rozin).

Come up with a project of your own that needs a little movement, and see if you can solve the problem with a servomotor.

Page last modified on September 20, 2013, at 03:35 PM