Arduino Servo Motor Control Tutorial

Garnet Hertz • SAIC • April 20th 2011

Overview

In this lab, you'll learn how to use an Arduino to control a servomotor's position using the value returned from an analog sensor, like a flex sensor, distance sensor or potentiometer. This tutorial has been adapted from Tom Igoe's Physical Computing tutorial – online at **http://itp.nyu.edu/physcomp/Labs/Servo** - visit this URL to copy the Arduino code you'll need. You can also find a similar example with a potentiometer at **http://arduino.cc/en/Tutorial/Knob**,

1. Parts

For this lab you'll need:
- Solderless breadboard
- 22-AWG hookup wire
- Arduino Microcontroller module
- Flex sensors (or any variable resistor)
- 10Kohm resistors
- RC Servomotor
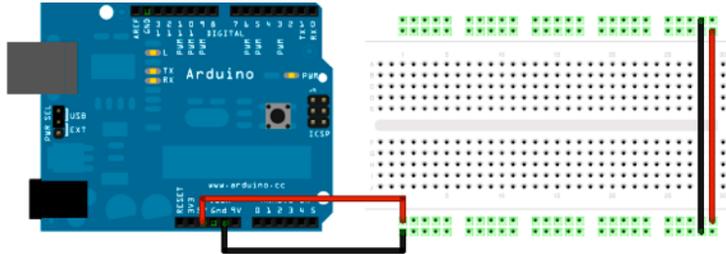
2. Prepare the breadboard

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:

3. Connect an analog input sensor and a servo

Pick any analog input and connect it to Analog pin 0. 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.
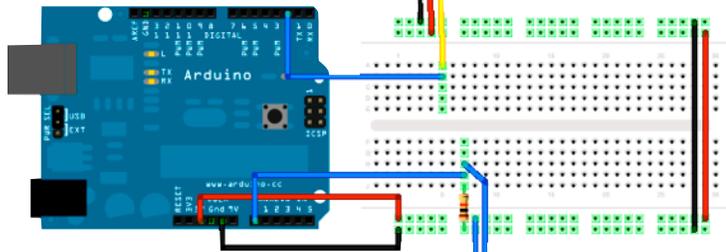
4. Program the Microcontroller

Go to **http://itp.nyu.edu/physcomp/Labs/Servo** and get the second example (Section 4.2) of how to control the motor with the Arduino Servo Library. The Servo Library provides a simple way to control a servo without having to do it "the old fashioned way" with pulse lengths (visible in Igoe's 4.1 example online). Instead, you can just say what angle you want your servo to go to and it goes there. Here's the meat of the code:

```
#include <Servo.h>      // include the servo library

Servo servoMotor;       // creates an instance of the servo object to control a servo

int analogPin = 0;      // the analog pin that the sensor is on
int analogValue = 0;    // the value returned from the analog sensor
int servoPin = 2;       // Control pin for servo motor. As of Arduino 0017, can be any pin

void setup() {
  servoMotor.attach(servoPin);        // attaches the servo on pin 2 to the servo object
}

void loop()
{
  analogValue = analogRead(analogPin);              // read the analog input (value between 0 and 1023)
  analogValue = map(analogValue, 0, 1023, 0, 179);  // map the analog value (0-1023) to the servo angle (0-179)
  servoMotor.write(analogValue);          // write the new mapped analog value to set the servo position
  delay(15);                              // waits for the servo to get there
}
```

5. Make Something

Ideas: A mechanical "Facebook status" update clock, a proximity display that indicates if someone is in your personal space, a miniature fairground swing-the-hammer-ring-the-bell strength machine, a device to enable your pet to dispense your food, a doorbell system that physically knocks on your front door, a light sensor that automatically lowers or raises your hinged clip-on sunglasses, a device that is built for display in a public space.