Building a Capacitive Proximity Sensor using Bare Paint

by **BareConductive** on December 5, 2011

Table of Contents

Intro: Building a Capacitive Proximity Sensor using Bare Paint

In this tutorial, we are going to make a simple capacitance sensor using a pad of Bare Paint, an Arduino and a resistor. When correctly constructed, this sensor can detect the presence of a person from up to 300 mm and can work behind any non-conductive material (glass, wood, plastic, etc).

Please watch this VIDEO for a live explanation.

By the end of the tutorial you'll have a pad of Paint connected to the Arduino, sending a stream of numbers to your computer's serial port. In later tutorials we'll take a look at what you can do with these numbers. This tutorial is based on the CapSense library and code that can be found here: http://www.arduino.cc/playground/Main/CapSense . All credit for the code goes to Paul Badger for making this so easy to implement!
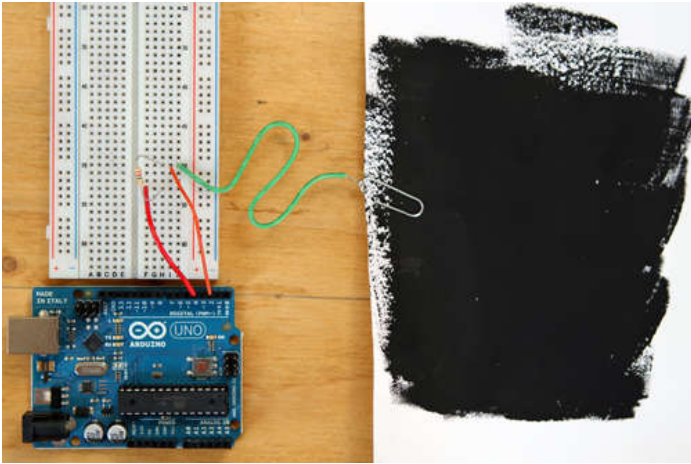
Materials

Bare Paint
Paint Brush
Paper
Arduino
USB Cable
Bread Board
Jump Wires
Resistors
Aligator Clip
Paper Clip

Components

Arduino: Sparkfun
Breadboard: Sparkfun
Jump Wires: Sparkfun
USB Cable: Sparkfun
Resistors: Sparkfun

Step 1: Materials
The first step is to gather the appropriate materials, listed above
To make a capacitance sensor using Bare Paint, you'll need:

1. A jar of Bare Paint

2. Small selection of resistors, between 100k ohm and 20M ohm (specific size is not important)

3. An Arduino, preferably Uno, Duemillonove or equivalent

4. Solderless breadboard and some jump wires

5. Some general prototyping materials to connect the pad of Paint to the Arduino, like wire, alligator clips etc

6. USB cable to connect the Arduino to you computer
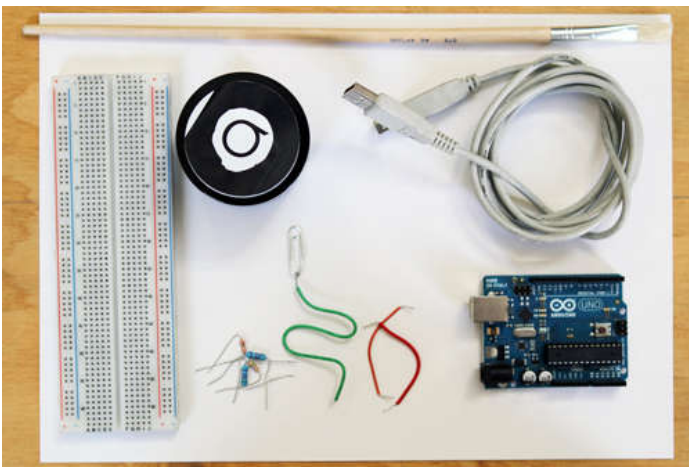
Step 2: Sensor Pad and Circuit

Now that you have your materials together, its time to start building the sensor and the circuit. First you need to paint a pad of Bare Paint onto any non conductive surface. In the video above we've just painted a piece of paper (as seen in the image below). The shape isn't important and the size should be around 100mm x 100mm (4" x 4"). The size of the pad can impact the sensitivity of the sensor, so try a few different sizes if you want and see how this changes the way that the sensor reacts. Leave your sensor pad to dry.

The next step is to build a simple circuit (pictured in image below) containing a resistor between pin 2 (orange wire connecting to the breadboard) and pin 4 (red wire connecting to the breadboard) of the Arduino board. Pin 4 is referred to as the "Sensor Pin" and Pin 2 is referred to as the "Receive Pin". An easy way to create this circuit is via a solder less breadboard.
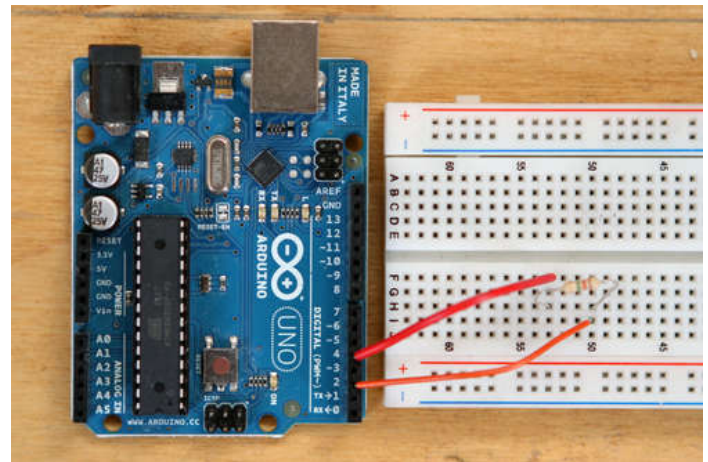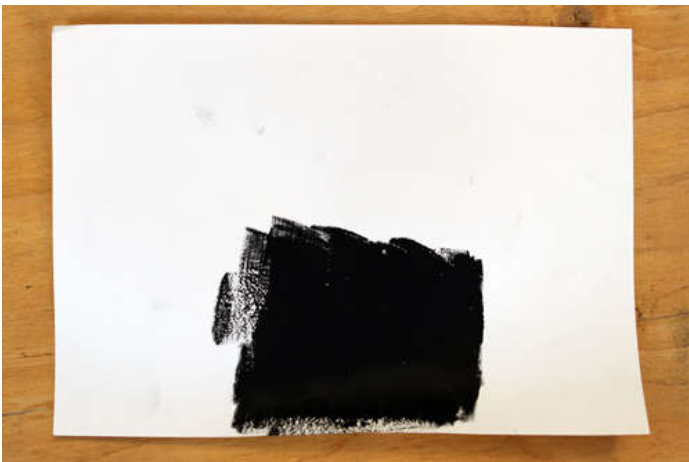
Step 3: Connecting to Bare Paint

Once you have your circuit built you can connect it to the pad of Paint. Take a look at our connecting tutorial for cool ways to connect to the Paint, but for now, we're just going to use a paperclip soldered to a wire. Whatever the connection type is, the Paint needs to be connected electrically to pin 2 on the Arduino. In the image below, you can see that the green wire is connected to the same row on the breadboard as the orange wire. For further explanation, includinging a wiring diagram, check out the Arduino playground here .
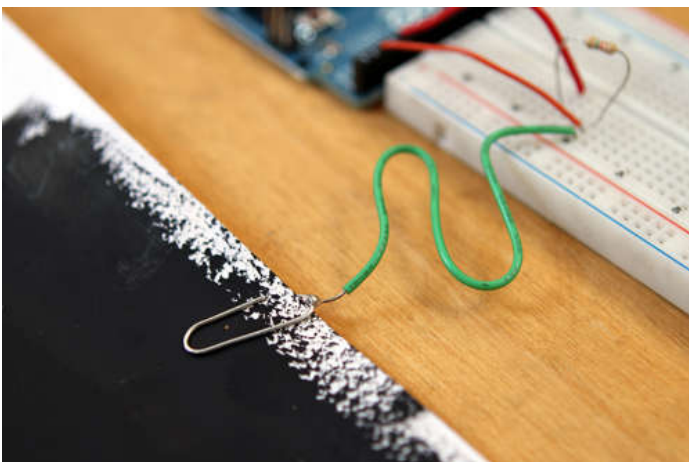
Step 4: Setting up the Arduino

Great! Now all of the physical work is done and we're off to the code. Make sure that you've downloaded and installed the Arduino programming environment. If you haven't, you can find it here . Next, you'll want to download and install the CapSense library. When you unzip the files make sure to put the folder titled "CapSense" into Documents/Arduino/libraries as shown here.

You will know that you've installed the library correctly when upon restarting Arduino you are able to select it from the Sketch / Import Library… menu, as seen here.

Importing

Now that the library is installed, copy and paste the code at the bottom of this tutorial into a new arduino sketch. Load the code onto your Arduino. For more information about loading software onto the Arduino the Arduino site here .

Copy and paste the code below into a new Arduino sketch.

```
#include <CapSense.h>
/*
* CapitiveSense Library Demo Sketch
* Paul Badger 2008
* Slightly adapted by Bare Conductive 2011
* Uses a high value resistor e.g. 10 megohm between send pin and receive pin
* Resistor effects sensitivity, experiment with values, 50 kilohm - 50 megohm. Larger resistor values yield larger sensor values.
* Receive pin is the sensor pin - try different amounts of Bare Paint
* Best results are obtained if sensor foil and wire is covered with an insulator such as paper or plastic sheet
*/


CapSense cs_4_2 = CapSense(4,2); // 10 megohm resistor between pins 4 & 2, pin 2 is sensor pin, add Bare Paint
// CapSense cs_4_5 = CapSense(4,5); // OPTIONAL: for sensor 2, 10 megohm resistor between pins 4 & 6, pin 6 is sensor pin, add Bare Paint
// CapSense cs_4_8 = CapSense(4,8); // OPTIONAL: for sensor 3, 10 megohm resistor between pins 4 & 8, pin 8 is sensor pin, add Bare Paint

void setup()
{
cs_4_2.set_CS_AutocaL_Millis(0xFFFFFFFF); // turn off autocalibrate on channel 1 - just as an example
Serial.begin(9600);
}

void loop()

{
long start = millis();
long total1 = cs_4_2.capSense(30);
// long total2 = cs_4_5.capSense(30); // OPTIONAL for sensor 2
// long total3 = cs_4_8.capSense(30); // OPTIONAL for sensor 3

// Serial.print(millis() - start); // OPTIONAL: check on performance in milliseconds
// Serial.print("t"); // OPTIONAL: tab character for debug windown spacing

Serial.println(total1); // OPTIONAL: To use additional sensors,change Serial.println to Serial.print for proper window spacing
//Serial.print("t"); // OPTIONAL: tab character for window spacing for sensor output 2
//Serial.print(total2); // OPTIONAL: print sensor output 2
//Serial.print("t"); // OPTIONAL: tab character character for sensor output 3
//Serial.println(total3); // print sensor output 3

delay(10); // arbitrary delay to limit data to serial port
}
```
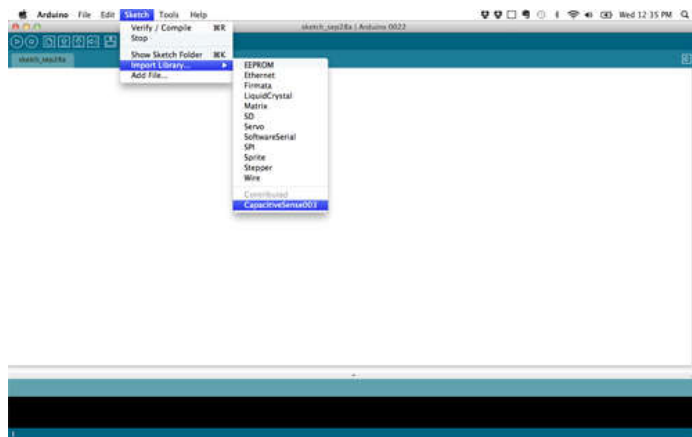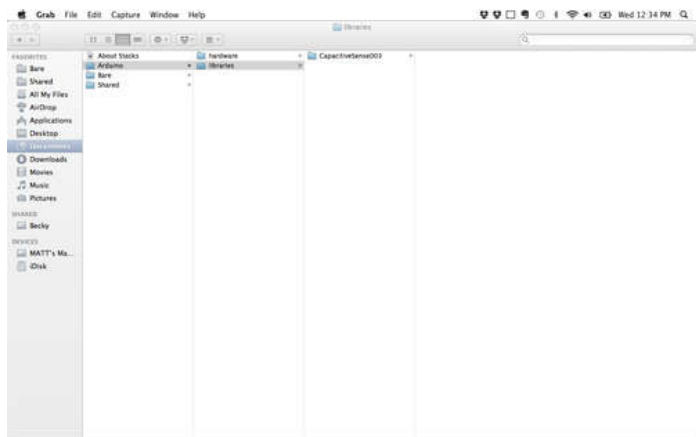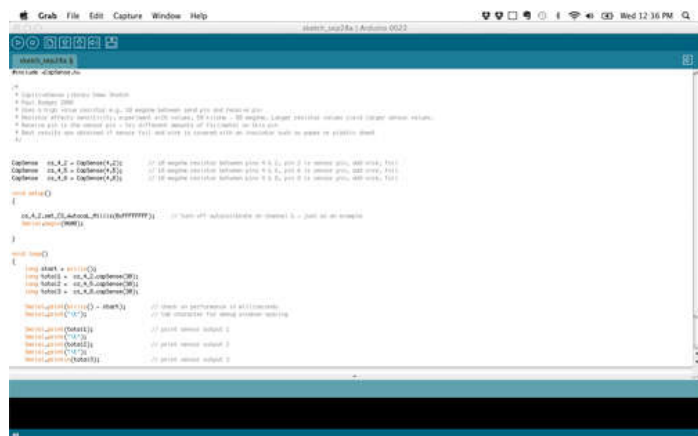
Step 5: Testing

Now we are ready to test your sensor! Make sure that your computer is plugged into the wall, or the Arduino is connected to ground as this improves the stability of the sensor. To check the output of the sensor, open the Serial monitor in the Arduino programming environment (make sure the monitor is set to 9600 baud as that is what is specified in the code). If it is working correctly, you will see a stream of numbers like in the image below. Moving your hand closer and farther from the pad of Paint should change the outputted number. Try swapping out the resistor and test how it changes the sensitivity. Larger resistors tend to make the sensor more sensitive and as such it will work from farther away.

Hey, it works! As Bibi moves her hand up and down the numbers on the serial port monitor change. It's now time to start experimenting. Try swaping the resistor, the pad of Paint or try covering the sensor with another non conductive material. This type of sensor can work on a lot of different surfaces and materials. In later tutorials we'll look at how we can use this sensor to control lights and sound. We want to see what you do with thie tutorial, so email us at community@bareconductive.com.
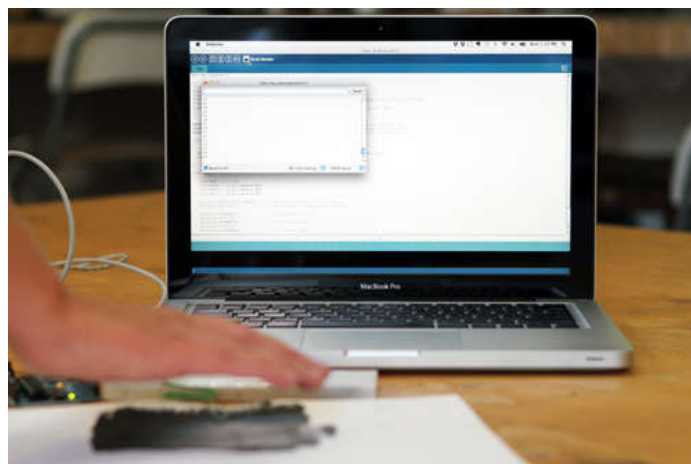
Step 6: General Links

General Links

Arduino http://www.arduino.cc/
CapSense http://www.arduino.cc/playground/Main/CapSense
Great article about Capacitive sensing from Lion Precision http://www.capacitive-sensing.com/capacitive-sensor-theory.html
Wikipedia article about Capacitive Sensing http://en.wikipedia.org/wiki/Capacitive_sensing

Related Instructables

Making a Bare Paint Potentiometer (video) by BareConductive
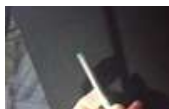

Make Conductive Rubber: Transparent stylus-iPod/iPhone by mikey77

Make a Glowing Halloween Card by BareConductive

Using Bare Paint to make paper Halloween Masks by BareConductive

Hard Tip iPad/Capacitive Stylus (made from X-acto knife) (Photos) by variances

Making a Holiday Card using Bare Paint by BareConductive