



Capacitive Touch with Conductive Fabric & Flora

Created by Becky Stern



Last updated on 2018-08-22 03:33:33 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Tools & supplies	4
Wiring the circuit	6
Code	9
Adding more inputs	14

Overview

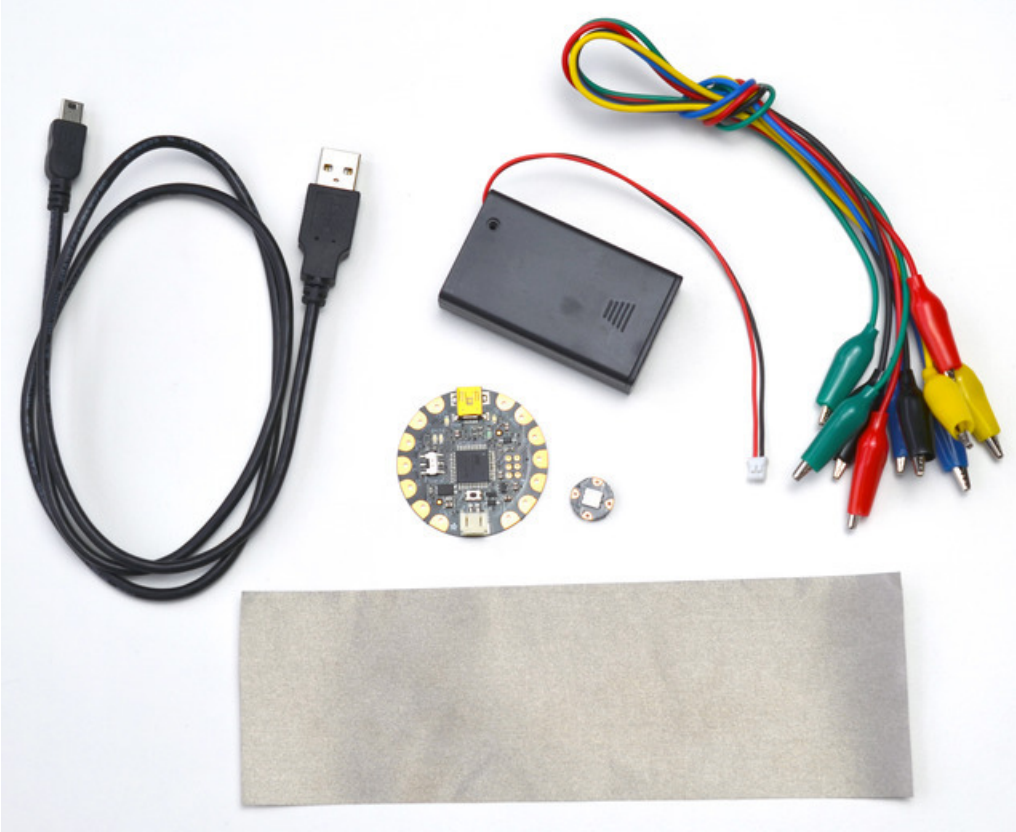
It's easy to use [Arduino's CapacitiveSensor library \(https://adafru.it/aWp\)](https://adafru.it/aWp) with Flora to make conductive fabric touch sensors! You can make soft game controllers, stereo remote pillows, interactive plush toys, or touch-sensitive clothes. If you're new to Flora, check out these tutorials first to bring you up to speed:

- [Getting started with FLORA \(https://adafru.it/aSZ\)](https://adafru.it/aSZ)
- [Flora RGB Smart Neo Pixels \(https://adafru.it/aRT\)](https://adafru.it/aRT)



Most photos in this guide by Collin Cunningham.

Tools & supplies



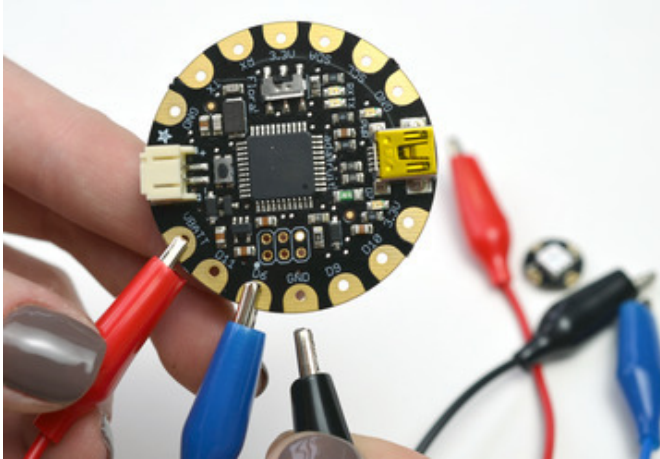
To prototype a Flora capacitive sensing circuit, you'll need the following:

- [Flora main board \(http://adafru.it/659\)](http://adafru.it/659)
- [Woven conductive fabric \(http://adafru.it/1168\)](http://adafru.it/1168)
- [Flora RGB Smart Neo Pixel \(http://adafru.it/1060\)](http://adafru.it/1060)
- [Alligator clips \(http://adafru.it/1008\)](http://adafru.it/1008)
- [Battery holder \(http://adafru.it/727\)](http://adafru.it/727)



You'll also need a few very high resistors, 50K - 50M. The resistor value affects the sensitivity of the touch sensor-- at the low end the sensor will only be activated by absolute touch, and on the high end the sensor can be activated from a distance of 12 inches or more.

Wiring the circuit



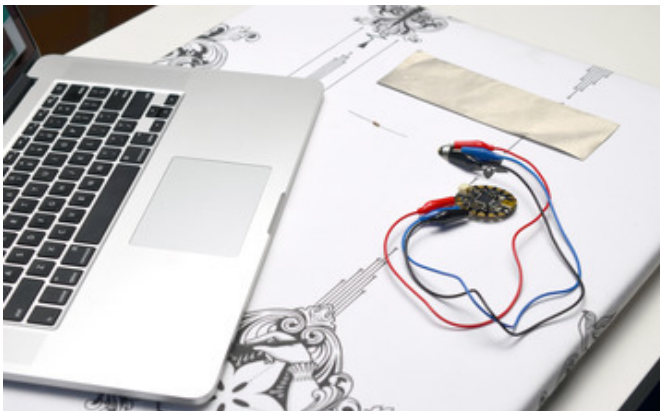
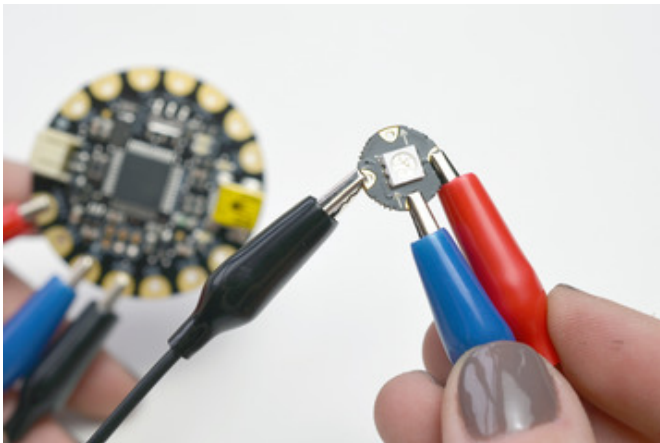
Pixel

Use a Flora RGB Pixel as an output. Later we'll code it up to change color when the sensor is touched.

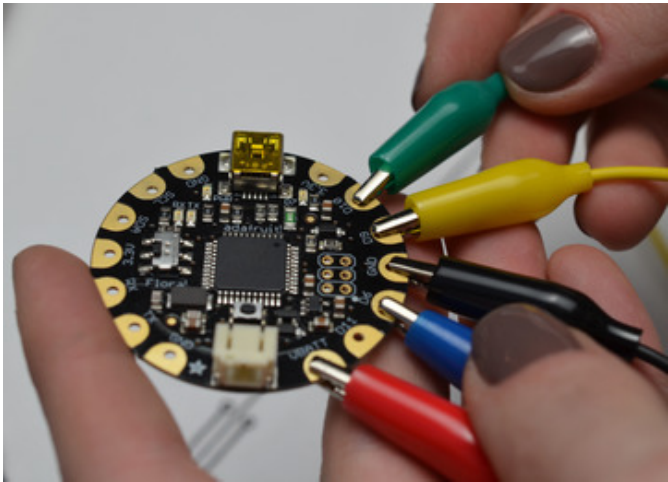
Hook up three alligator clips to VBATT, D6, and GND on your Flora.

Clip the other ends to +, inward-facing arrow, and - on the pixel board.

Read the [Flora RGB Pixels tutorial \(https://adafru.it/aRT\)](https://adafru.it/aRT) and test to make sure the pixel is working!

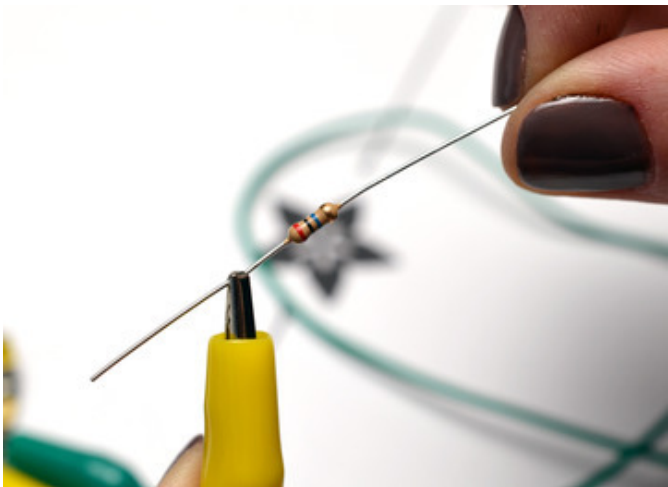


If your bench surface is metal or ESD, insulate your circuit from it with an ironing board, cutting mat, roll of paper, etc.

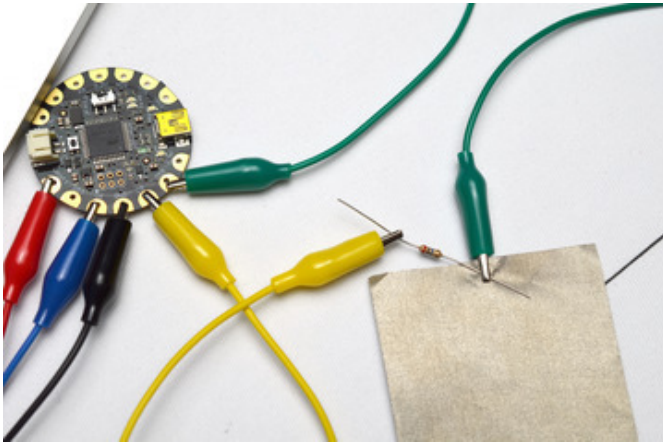


Adding fabric

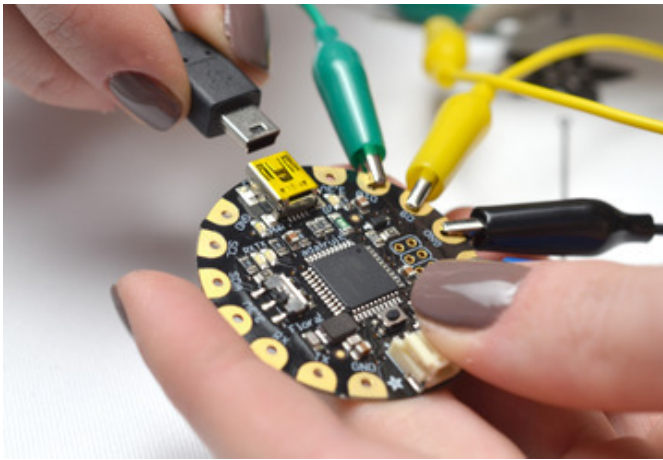
Clip two more alligator clips to pads marked D9 and D10.



Clip D9's other end to either lead on your very high resistor.

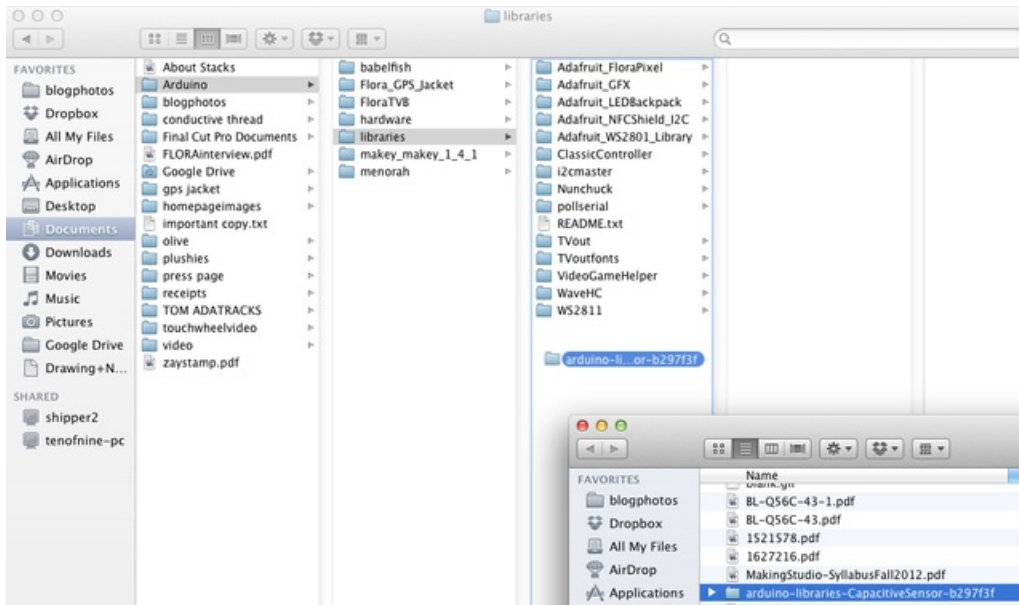


Clip D10's other end to the conductive fabric, also pinching the opposing leg of your resistor.



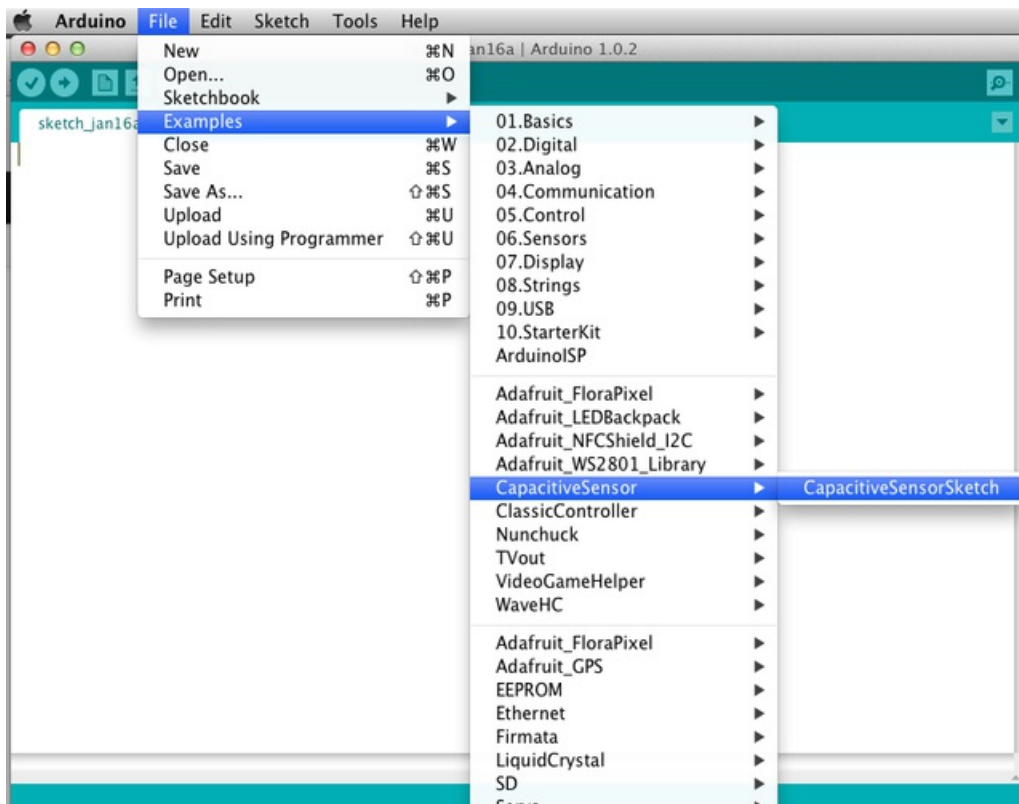
Connect your Flora to your computer with a USB cable.

Code



Download the [CapacitiveSensor library from Arduino](https://adafru.it/aWp) (<https://adafru.it/aWp>) written by Paul Badger, unzip the file and move the entire folder into your Arduino libraries folder.

Here's a video overview of how the circuit and code work, thanks to Paul Stoffregen:



Fire up the Adafruit version of the Arduino IDE ([get it here \(https://adafru.it/aVE\)](https://adafru.it/aVE)) and open the sample sketch for the CapacitiveSensor library.

Find this first section of code:

```
CapacitiveSensor cs_9_10 = CapacitiveSensor(4,2);
CapacitiveSensor cs_9_2 = CapacitiveSensor(4,5);
CapacitiveSensor cs_4_8 = CapacitiveSensor(4,8);
```

And change to reflect your current wiring, commenting out all but one sensor configuration:

```
CapacitiveSensor cs_9_10 = CapacitiveSensor(9,10);
//CapacitiveSensor cs_9_2 = CapacitiveSensor(9,2);
//CapacitiveSensor cs_4_8 = CapacitiveSensor(4,8);
```

Also find this corresponding bit of code:



```
void setup()
{
  cs_4_2.set_CS_Autocal_Millis(0xFFFFFFFF); // turn off d
  Serial.begin(9600);
}

void loop()
{
  long start = millis();
  long total1 = cs_4_2.capacitiveSensor(30);
  long total2 = cs_4_6.capacitiveSensor(30);
  long total3 = cs_4_8.capacitiveSensor(30);

  Serial.print(millis() - start); // check on perform
  Serial.print("\t"); // tab character fo

  Serial.print(total1); // print sensor out
  Serial.print("\t");
  Serial.print(total2); // print sensor out
  Serial.print("\t");
  Serial.println(total3); // print sensor out

  delay(10); // arbitrary delay
```

And update to read:

```

void loop()
{
  long start = millis();
  long total1 = cs_9_10.capacitiveSensor(30);
  //long total2 = cs_4_5.capacitiveSensor(30);
  //long total3 = cs_4_8.capacitiveSensor(30);

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

  Serial.println(total1);              // print sensor output 1
  //Serial.print("\t");
  //Serial.print(total2);              // print sensor output 2
  //Serial.print("\t");
  //Serial.println(total3);           // print sensor output 3

  delay(10);                           // arbitrary delay to limit data to serial port
}

```

Upload this code to your Flora board and open Arduino's serial monitor. Touch the conductive fabric and watch the numbers change on screen. Take note of the low end of the spike-- for me the reading always went above 4000 when I touched the fabric sensor.

Add an IF statement that sets Flora's onboard LED (D7) to light up if the sensor reading exceeds the "I touched it" threshold:

```

if (total1 > 4000){
  digitalWrite(7, HIGH);
}else{
  digitalWrite(7, LOW);
}

```

After uploading this code you should see Flora's red LED light up every time you touch the fabric. Now onto a more interesting output, the RGB pixel!

Here's the entire sketch, updated to drive a single RGB Neo Pixel. Copy it and paste it into a new sketch or replace your current sketch:

```

#include <CapacitiveSensor.h>
#include "Adafruit_FloraPixel.h"

/*
 * CapitiveSense Library Demo Sketch
 * Paul Badger 2008
 * Uses a high value resistor e.g. 10M between send pin and receive pin
 */

```

```

uses a high value resistor e.g. 10M between send pin and receive pin
* Resistor effects sensitivity, experiment with values, 50K - 50M. Larger resistor values yield larger s
* Receive pin is the sensor pin - try different amounts of foil/metal on this pin
* Modified by Becky Stern 2013 to change the color of one RGB Neo Pixel based on touch input
*/

CapacitiveSensor cs_9_10 = CapacitiveSensor(9,10); // 10M resistor between pins 4 & 2, pin 2 is
//CapacitiveSensor cs_9_2 = CapacitiveSensor(9,2); // 10M resistor between pins 4 & 6, pin 6 is
//CapacitiveSensor cs_4_8 = CapacitiveSensor(4,8); // 10M resistor between pins 4 & 8, pin 8 is
Adafruit_FloraPixel strip = Adafruit_FloraPixel(1);

void setup()
{
  cs_9_10.set_CS_Autocal_Millis(0xFFFFFFFF); // turn off autocalibrate on channel 1 - just as an exa
  Serial.begin(9600);
  strip.begin();
  strip.show();
}

void loop()
{
  long start = millis();
  long total1 = cs_9_10.capacitiveSensor(30);
  //long total2 = cs_9_2.capacitiveSensor(30);
  //long total3 = cs_4_8.capacitiveSensor(30);

  if (total1 > 4000){
    digitalWrite(7, HIGH);
    strip.setPixelColor(0, Color(0,255,0));
    strip.show();
  } else {
    strip.setPixelColor(0, Color(0,0,0));
    strip.show();
  }

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

  Serial.println(total1); // print sensor output 1
  //Serial.print("\t");
  //Serial.println(total2); // print sensor output 2
  //Serial.print("\t");
  //Serial.println(total3); // print sensor output 3

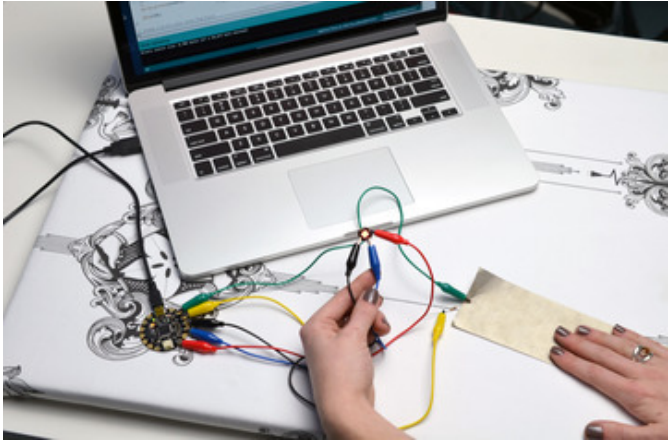
  delay(10); // arbitrary delay to limit data to serial port
}

// Create a 24 bit color value from R,G,B
RGBPixel Color(byte r, byte g, byte b)
{
  RGBPixel p;

  p.red = r;
  p.green = g;
  p.blue = b;

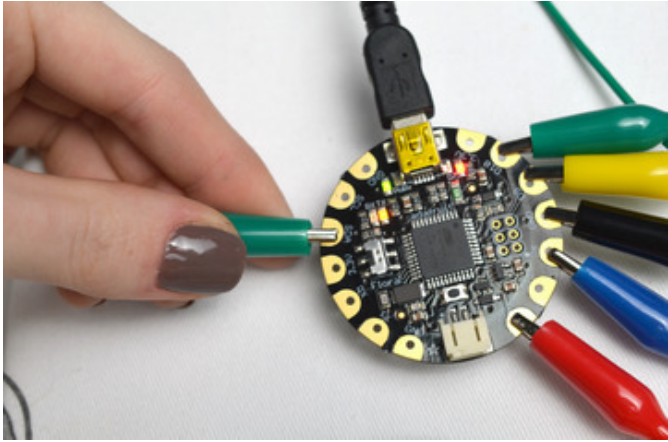
  return p;
}

```



Test your sensor by touching the conductive fabric. The pixel should light up when you are in contact with the sensor!

Adding more inputs



To add another fabric touch pad, clip another alligator clip to another digital pin. I picked SDA, which is D2 in the Arduino software.



Clip yet another alligator to the non-fabric side of the first resistor. This extends the connection from D9 to the new conductive fabric pad.



Clip SDA (D2)'s other end to the fabric and another big resistor, as shown. Now you can look for a sensor reading from the new addition.



Upload this sketch for changing between two colors with two touch pads:

```
#include <CapacitiveSensor.h>
#include "Adafruit_FloraPixel.h"

/*
 * CapitiveSense Library Demo Sketch
 * Paul Badger 2008
 * Uses a high value resistor e.g. 10M between send pin and receive pin
 * Resistor effects sensitivity, experiment with values, 50K - 50M. Larger resistor values yield larger s
 * Receive pin is the sensor pin - try different amounts of foil/metal on this pin
 * Modified by Becky Stern 2013 to change the color of one RGB Neo Pixel based on touch input
 */

CapacitiveSensor cs_9_10 = CapacitiveSensor(9,10); // 10M resistor between pins 4 & 2, pin 2 is
CapacitiveSensor cs_9_2 = CapacitiveSensor(9,2); // 10M resistor between pins 4 & 6, pin 6 is se
//CapacitiveSensor cs_4_8 = CapacitiveSensor(4,8); // 10M resistor between pins 4 & 8, pin 8 is
Adafruit_FloraPixel strip = Adafruit_FloraPixel(1);

void setup()
{
  cs_9_10.set_CS_AutocaL_Millis(0xFFFFFFFF); // turn off autocalibrate on channel 1 - just as an exa
  Serial.begin(9600);
  strip.begin();
  strip.show();
}

void loop()
{
  long start = millis();
  long total1 = cs_9_10.capacitiveSensor(30);
  long total2 = cs_9_2.capacitiveSensor(30);
  //long total3 = cs_4_8.capacitiveSensor(30);

  if (total1 > 4000){
    digitalWrite(7, HIGH);
    strip.setPixelColor(0, Color(0,255,0));
    strip.show();
  }
  if (total2 > 4000){
```

```
strip.setPixelColor(0, Color(255,0,0));
strip.show();
}

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

Serial.print(total1);                // print sensor output 1
Serial.print("\t");
Serial.println(total2);              // print sensor output 2
//Serial.print("\t");
//Serial.println(total3);           // print sensor output 3

delay(10);                           // arbitrary delay to limit data to serial port
}

// Create a 24 bit color value from R,G,B
RGBPixel Color(byte r, byte g, byte b)
{
  RGBPixel p;

  p.red = r;
  p.green = g;
  p.blue = b;

  return p;
}
```