Force Sensitive Resistor Hookup Guide

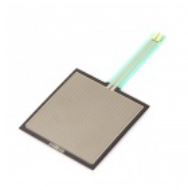
CONTRIBUTORS: *JIMB0*

♡ FAVORITE 1

Introduction

Force-sensitive resistor's (FSR) are easy-to-use sensors designed for measuring the presence and relative magnitude of localized physical pressure.

Force Sensitive Resistor 0.5"
SEN-09375
$6.96
★★★★½ 9



Force Sensitive Resistor - Square
SEN-09376
$9.96
★★★★½ 4

Force Sensitive Resistor - Small
SEN-09673
$5.96
★★★½☆ 3

Force Sensitive Resistor - Long
SEN-09674
$20.95
★★★★★ 1

The resistance of an FSR varies as the force on the sensor increases or decreases. When no pressure is being applied to the FSR, its resistance will be larger than 1MΩ. The harder you press on the sensor's head, the lower the resistance between the two terminals drops. By combining the FSR with a static resistor to create a voltage divider, you can produce a variable voltage that can be read by a microcontroller's analog-to-digital converter.

Suggested Materials

This tutorial serves as a quick primer on FSR's and demonstrates how to hook them up and use them. Beyond an FSR of your choice, the following materials are recommended:

Arduino Uno – We'll be using the Arduino's analog-to-digital converter to read in the variable resistance of the FSR. Any Arduino-compatible development platform – be it a RedBoard, Pro or Pro Mini – can substitute.

Resistor Kit – To turn the FSR's variable resistance into a readable voltage, we'll combine it with a static resistor to create a voltage divider. This resistor kit is handy for some trial-and-error testing to hone in on the most sensitive circuit possible.

Breadboard and Jumper Wires – The FSR's terminals are breadboard-compatible. We'll stick in that and the resistor, then use the jumper wires to connect from breadboard to Arduino.

SparkFun RedBoard - Programmed with Arduino
DEV-12757
$19.95
★★★★½ 108

Resistor Kit - 1/4W (500 total)
COM-10969
$7.96
★★★★½ 105

Breadboard - Self-Adhesive (White)
PRT-12002
$4.95
★★★★½ 26

Jumper Wires Standard 7" M/M - 30 AWG (30 Pack)
PRT-11026
$1.95
★★★★½ 20

Suggested Reading

Analog components, like these FSRs, are a great sensor-reading entry-point for beginners, but there are a few electronics concepts you should be familiar with. If any of these tutorial titles sound foreign to you, consider skimming through that content first.

Analog to Digital Conversion
The world is analog. Use analog to digital conversion to help digital device interpret the world.

Voltage Dividers
Turn a large voltage into a smaller one with voltage dividers.

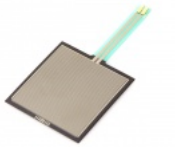
What is an Arduino?
What is this 'Arduino' thing anyway?

Analog vs. Digital
This tutorial covers the concept of analog and digital signals, as they relate to electronics.

FSR Overview

There are a variety of FSR options out there, and a few key characteristics to differentiate them: **size, shape, and sensing range**. Here's a quick overview:

Name	Shape	Sensing Area	Min Pressure	Max Pressure
 **Force Sensitive Resistor - Small (SEN-09673)**	Circular	7.62 mm dia (0.3 in)	0.1 kg (0.22 lb)	1 kg (2.2 lb)
 **Force Sensitive Resistor 0.5" (SEN-09375)**	Circular	12.7 mm dia (0.5 in)	100 g (0.22 lb)	10 kg (22.04 lb)

 **Force Sensitive Resistor - Square (SEN-09376)**	Square	44.45 x 44.45 mm (1.75 x 1.75 in)	100 g (0.22 lb)	10 kg (22.04 lb)
 **Force Sensitive Resistor - Long (SEN-09674)**	Rectangular	6.35 x 609.6 mm dia (0.25 x 24.0 in)	100 g (0.22 lb)	10 kg (22.04 lb)

Shape and Size

Most FSR's feature either a circular or rectangular sensing area. The square FSR is good for broad-area sensing, while the smaller circular sensors can provide more precision to the location being sensed.

The rectangular FSR's include a small-ish square 1.75 x 1.75" sensor and a long 0.25 x 24" strip. The rest of the sensors feature a circular sensing area.

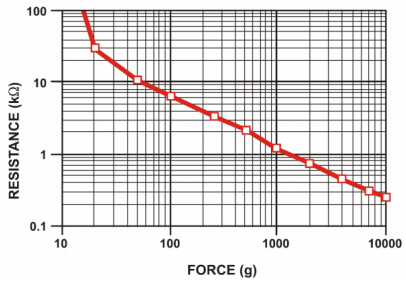
Sensing Range

Another key characteristic of the FSR is it's rated sensing range, which defines the minimum and maximum amounts of pressure that the sensor can differentiate between.

The lower the force rating, the more **sensitive** your FSR hookup has the potential to be. But! Any pressure beyond the sensor's maximum limit will be unmeasurable (and may damage the component). The small 1kg-rated FSR will provide more sensitive readings from 0 to 1kg, but won't be able to tell the difference between a 2kg and 10kg weight.

Force vs. Resistance

The graph below, figure 2 from the FSR Integration Guide, demonstrates the typical force-resistance relationship:



The relationship is generally linear from 50g and up, but note what the relationship does below 50g, and even more-so below 20g. These sensor's have a **turn-on threshold** – a force that must be present before the resistance drops to a value below 10kΩ, where the relationship becomes more linear.

> These sensors are simple to set up and great for sensing pressure, but they aren't incredibly accurate. They're useful for sensing the presence of something, and the relative magnitude of that force, but they're not all that great at measuring weight (that's what load cell's are for!).

Example Hardware Hookup

By creating a voltage divider with the FSR and another resistor, you can create a variable voltage output, which can be read by a microcontroller's ADC input.

Selecting a Static Resistor

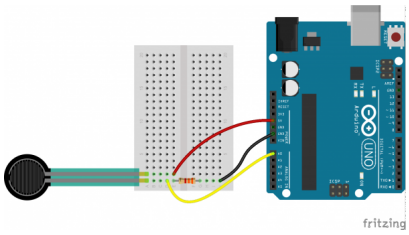
The tricky part of voltage-dividing an FSR is selecting a static resistor value to pair with it. You don't want to overpower the maximum resistance of the FSR, but you also don't want the FSR's minimum resistance to be completely overshadowed either.

It helps to know what range of force you'll be reading. If your project's force-sensing covers the broad range of the FSR (e.g. 0.1-10kg), try to pick a static resistance in the middle-range of the FSR's resistive output – something in the middle of 200-6kΩ. 3kΩ, or a common resistor like **3.3kΩ**, is a good place to start.

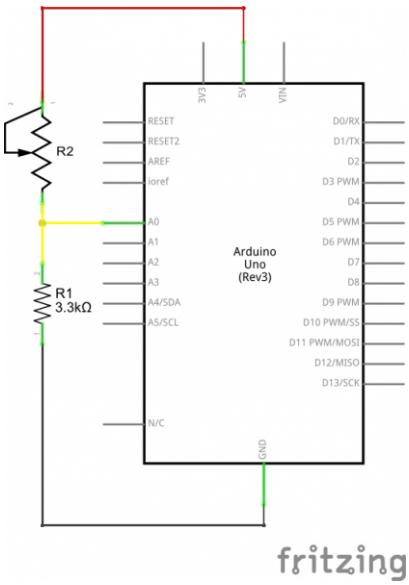
> Short on resistors? If all you have is 10kΩ resistors (looking at you Sensor Kit visitors), you can still make something close to 3k! Try putting **three 10kΩ's in parallel** to create a 3.33kΩ monster resistor. Or put three 330Ω resistors in **series** to create a 990Ω concoction, which will work pretty well too.

Example Circuit

Here's a fritzing diagram combining the FSR, 3.3kΩ resistor, three jumper wires and the Arduino:



And the schematic:



This voltage divider will cause the voltage at A0 to increase as the resistance of the FSR decreases. When the FSR is left untouched, measuring as nearly an open circuit, the voltage at A0 should be zero. If you press as hard as possible on the FSR, the voltage should increase close 5V.

Example Arduino Sketch

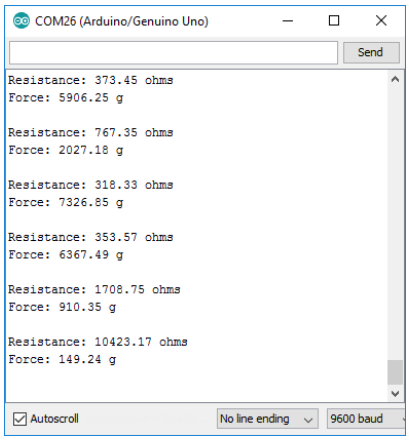
Here is a simple Arduino example based on the circuit above. Copy and paste this into your Arduino IDE, then upload!

> **Note:** This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE.
> If you have not previously installed an Arduino library, please check out our installation guide.

```
/******************************************************************************
Force_Sensitive_Resistor_Example.ino
Example sketch for SparkFun's force sensitive resistors
  (https://www.sparkfun.com/products/9375)
Jim Lindblom @ SparkFun Electronics
April 28, 2016

Create a voltage divider circuit combining an FSR with a 3.3k resistor.
- The resistor should connect from A0 to GND.
- The FSR should connect from A0 to 3.3V
As the resistance of the FSR decreases (meaning an increase in pressure), the
voltage at A0 should increase.

Development environment specifics:
Arduino 1.6.7
******************************************************************************/
const int FSR_PIN = A0; // Pin connected to FSR/resistor divider

// Measure the voltage at 5V and resistance of your 3.3k resistor, and enter
// their value's below:
const float VCC = 4.98; // Measured voltage of Ardunio 5V line
const float R_DIV = 3230.0; // Measured resistance of 3.3k resistor

void setup()
{
  Serial.begin(9600);
  pinMode(FSR_PIN, INPUT);
}

void loop()
{
  int fsrADC = analogRead(FSR_PIN);
  // If the FSR has no pressure, the resistance will be
  // near infinite. So the voltage should be near 0.
  if (fsrADC != 0) // If the analog reading is non-zero
  {
    // Use ADC reading to calculate voltage:
    float fsrV = fsrADC * VCC / 1023.0;
    // Use voltage and static resistor value to
    // calculate FSR resistance:
    float fsrR = R_DIV * (VCC / fsrV - 1.0);
    Serial.println("Resistance: " + String(fsrR) + " ohms");
    // Guesstimate force based on slopes in figure 3 of
    // FSR datasheet:
    float force;
    float fsrG = 1.0 / fsrR; // Calculate conductance
    // Break parabolic curve down into two linear slopes:
    if (fsrR <= 600)
      force = (fsrG - 0.00075) / 0.00000032639;
    else
      force =  fsrG / 0.000000642857;
    Serial.println("Force: " + String(force) + " g");
    Serial.println();

    delay(500);
  }
  else
  {
    // No pressure detected
  }
}
```

After uploading, **open your serial monitor**, and set the baud rate to 9600 bps.

If you apply pressure to the FSR, you should see resistance and estimated pressure calculations begin to appear:



Play with the circuit and see how high or low you can get the readings to be. If you have more resistors, try swapping larger or smaller values in for the 3.3kΩ to see if you can make the circuit more sensitive. Don't forget to change the value of `R_DIV` towards the top of the sketch if you do!

Resources and Going Further

Now that you've got a force-sensing Arduino circuit, what project are you going to create? If you need more FSR-related resources, be sure to check out the FSR Integration Guide, which goes in-depth on the sensor's characteristics. The guide also presents a few more complex circuits you can try hooking up to get even more sensitivity out of your FSR.

CHECK OUT THE FSR INTEGRATION GUIDE

Need some project inspiration? Want to check out some similar analog sensors? Check out some of these related tutorials:

Getting Started with Load Cells
A tutorial defining what a load cell is and how to use one.

SIK Keyboard Instrument
We can use the parts and concepts in the SparkFun Invetor's Kit to make a primitive keyboard instrument.

Sensor Kit Resource Hub
An overview of each component in the SparkFun Sensor Kit, plus links to tutorials and other resources you'll need to hook them up.

Flex Sensor Hookup Guide
An overview of the flex sensor - a bendable variable resistor. Plus, example circuits and Arduino code to get you started!