Sensory Perception & Interaction Research Group
University of British Columbia

Create Your Own Touch Sensor System

Here we present a short guide on how to build your own pressure and location sensors using the techniques we use in the SPIN lab. The sensors are low-cost, high resolution, and relatively easy to create, inspired by the work of Hannah Perner-Wilson and Mika Satomi and their well-documented DIY wearable technology developments.

Overview

The sensors use a four-layer construction, with a sheet of conductive-striped fabric as both top and bottom layer, and piezoresistive & mesh layers sandwiched between. To create a grid pattern of location data, we place the top and bottom conductive-striped fabrics at 90 degrees (transpose) to each other. Here is a diagram outlining the construction:

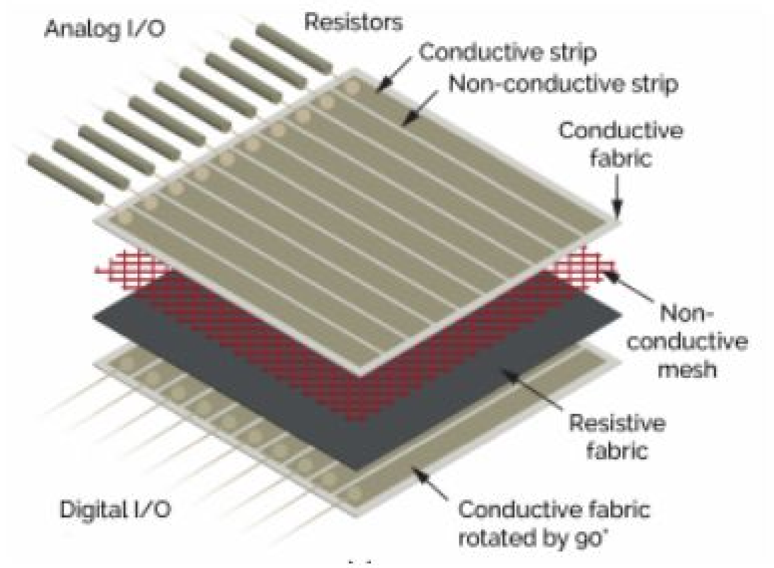


Figure A: fabric layers

Background: How the Sensor Works

The sensor functions through the piezoresistive fabric layer. As a user touches the sensor, the piezoresistive fabric layer (in the middle of the sensor) decreases its resistance value, and this deformation can be used to indicate the pressure of the touch. We derive the location of this pressure data from the two outer layers of fabric. By creating a grid pattern with the conductive stripes of these two layers, we are able to define the pressure data by x,y position, which indicates the position associated with the data. More details about the sensor can be found in the sensor design document on SPIN Lab Github, here.

Suggested Materials

Sensor Fabrics

Piezoresistive fabric (ours is 20 kilo-ohm per square fabric from Eeonyx)
Fabric with conductive stripes (ours is Eeonyx zebra-striped conductive fabric)

Mesh fabric, medium weight

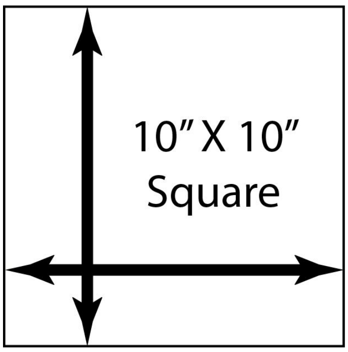
Adhesives
Masking tape
To adhere the layers together, choose either iron-on bonding tape like this or hand-sewing materials if you prefer to stitch by hand.

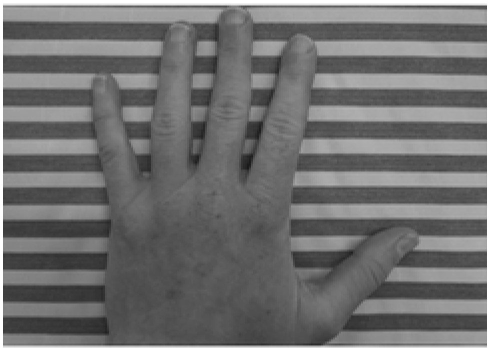
Wiring
Arduino or other microcontroller (we use the Arduino Mega)
Wires -- For a 10x10 sensor, you will need two 20-inch lengths of 10-set ribbon wires, two 10-pin jumper pieces, and four 10-wire male/male jumper wires. If you are adapting the sensor shape or dimensions, you will want to adapt this for the number of rows+columns in your sensor. See '*Wire & Debug*' section with detailed pictures on page four, below.)
Resistors (one per row/column, for a 10x10 sensor we used ten $1.8k\Omega$ resistors)
Breadboard -- enough to fit 10 wires, needs to be larger if your circuit will require more wires.

Miscellaneous
Clothing iron
Scissors
Wire strippers, cutters
Ruler, pen
Optional - fabric rotary cutter (example here.)

Steps

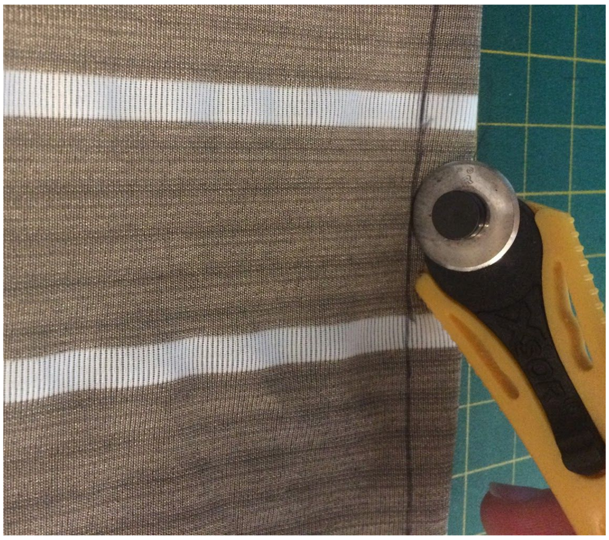
1. ***Plan & Cut***





1.1. Decide on the sensor shape and size. Since research in the SPIN lab involves gathering touch data from the human hand, we developed a 10"x10" square-shaped sensor to provide enough space for a human hand to move around and perform gestures. Your sensor's shape and size really depends on the application you have in mind. A practical consideration is any constraints you may have with hardware: you will want to compare the number of conductive stripes your fabric has, (as each stripe will become a wire of information) versus the number of pins available on your Arduino. Keep in mind that your Arduino will be accommodating a top and bottom layer of the conductive fabric, so it will need double the input pins.

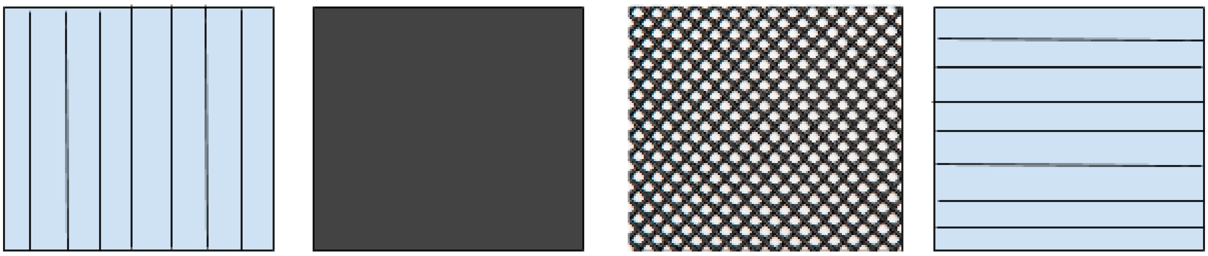
1.2. Using a ruler and pen, draw your sensor design onto each fabric layer. Be sure to leave 2-3 inches of extra fabric around each edge, which will be useful for wiring the fabric to the circuit. (More on that in *Wire & Debug*, on page 4.) If your fabrics have major creases, give them a quick iron to make them sit flat. (You can use a clothing iron with steam.)



1.3. Cut each fabric layer, along the lines you just drew. Use scissors, or a rotary fabric cutter like this one.

2. ***Layout***

Arrange each layer in a "sandwich", as follows:
Bottom: sheet of conductive-striped fabric, stripes going vertically*
Middle 1: piezoresistive fabric layer
Middle 2: mesh layer *(layers middle 1 and middle 2 can be swapped)*
Top: second conductive-striped fabric layer, stripes running horizontally*,
** Feel free to switch top and bottom layers' horizontal / vertical ordering. The main goal is that the orientation of the stripes of the top and bottom layer are at a 90-degree angle, which will give us the grid-like sensor pattern.*



Top: zebra-striped conductive fabric

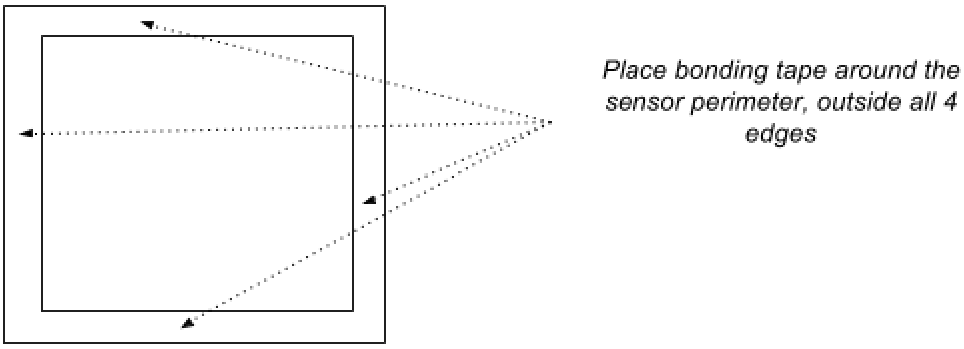
Mid 1: piezoresistive fabric

Mid 2: mesh

Bottom: zebra-striped conductive fabric

3. ***Bond / Sew***

   Now it's time to adhere each layer to one another. Based on your preference, choose either hand-stitching each layer together, or bonding them together with a simple heat-activated bonding glue available at craft stores and a clothing iron. (Caution with a sewing machine, we found that it caused the conductive yarns to get pushed through the fabric causing unexpected resistance values.)

   If you choose to bond the layers, make sure you place the tape outside the sensor boundary. If you apply the bonding glue within the sensor boundary, you are effectively killing the performance of the sensor in that location. (Because the sensor works by detecting changes in pressure, bonding it down within the boundary disables any movement to occur.)

4. ***Wire & Debug***

   To wire your sensor, you will want to create a connection for all the rows and columns you have formed, by wiring each conductive stripe on the sensor's top and bottom layers. You can complete the wiring without soldering by taking a ten-set of ribbon cable, stripping the wire ends, and piercing each wire end through one of the 10 conductive stripes of each side, near the bottom. (To secure the wire, pierce it back through the fabric to imitate a sewing action, until there is no dangling wire end.) If you like to solder, another approach you could take is to thread a conductive yarn through the conductive stripe of the fabric, and then solder the ribbon cable's wire end to the conductive thread, to join them together. Once each connection is wired, place masking tape over each connection individually to secure the wires, or add a circle of glue from a hot glue gun. (Securing the wiring is a must for touch sensors that invite a lot of frequent, high pressure interactions.) Choose one side (ie, top layer) as the rows and one side as the columns (ie, bottom layer), so you end up with two 10-wire ribbon cables connected to the sensor, one side wired horizontally and one side wired vertically. Below is a close-up view of the fabric/wire connection technique.

Figure C: How to wire ribbon cables to fabric. Notice the right side here is deliberately left bare without tape, to show the wiring. The left side of the picture shows how you would apply masking tape, to secure the connections.

Once the fabrics' conductive stripes are all wired up, you're ready to build streamlined connections to the rest of the circuit. Clip a jumper to the end of each ribbon cable, and into each jumper you can then insert the set of male/male jumper wires, which will connect to your Arduino and breadboard circuit. This makes it really easy, clean and tidy to connect those wires to the rest of the circuit you're about to build. Here is an example:

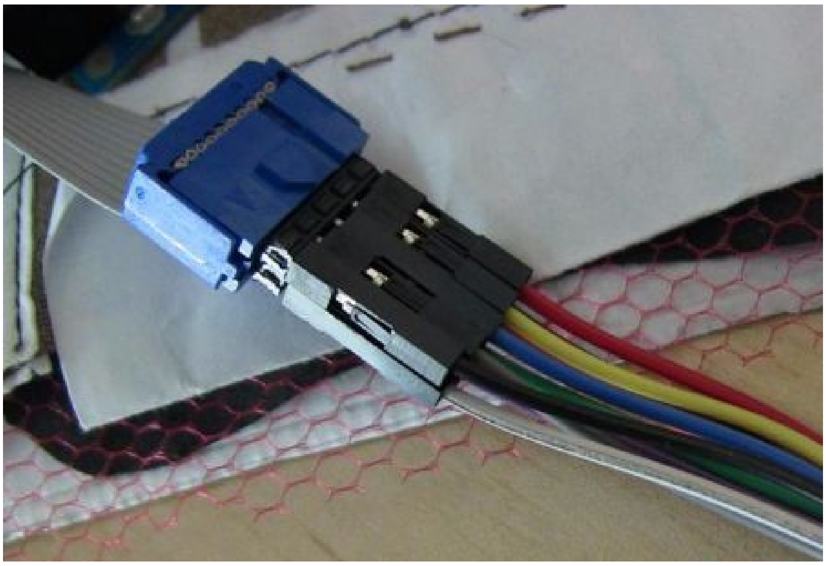


Figure D: Pictured here is the jumper connection described above, where a 10-pin jumper part is placed at the end of the ribbon cable, and male jumper cables are inserted into the jumper piece. This technique is used on the two wires, extending from the top and bottom fabric layers.

Next you will create the circuit that will input power to the sensor, collect the sensor's data, and feed it into the microcontroller for post-processing. Out of the two sides you just wired, choose one wire for ground, and one wire for analog. (Either choice works!) The analog wire will be collecting and reading the pressure data, and the other will be grounding the circuit. The circuit connections should be wired to look like this:

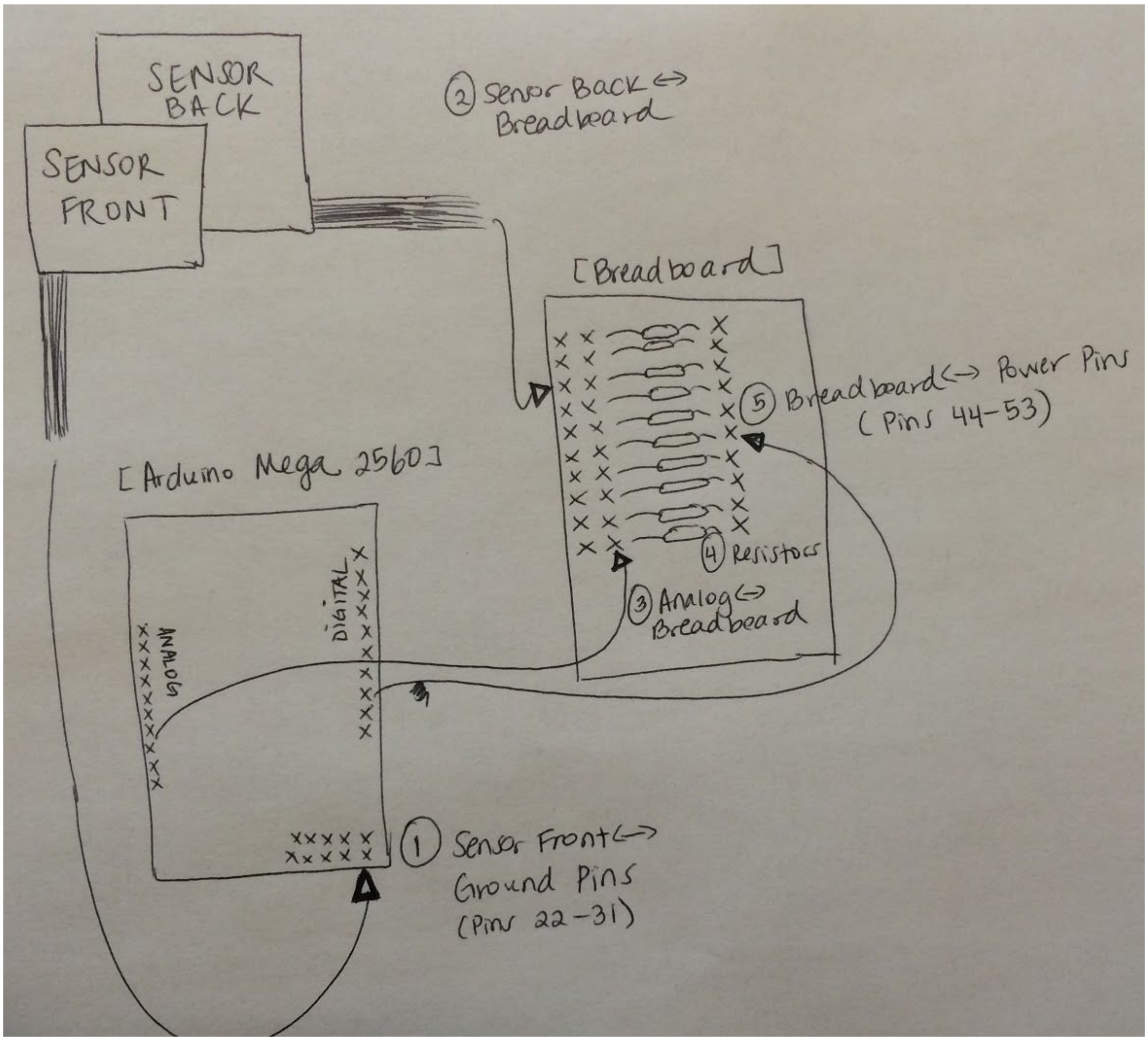


Figure E: Pictured here is a diagram of the circuit wiring. The five steps to implement the circuit are labelled accordingly in the diagram.

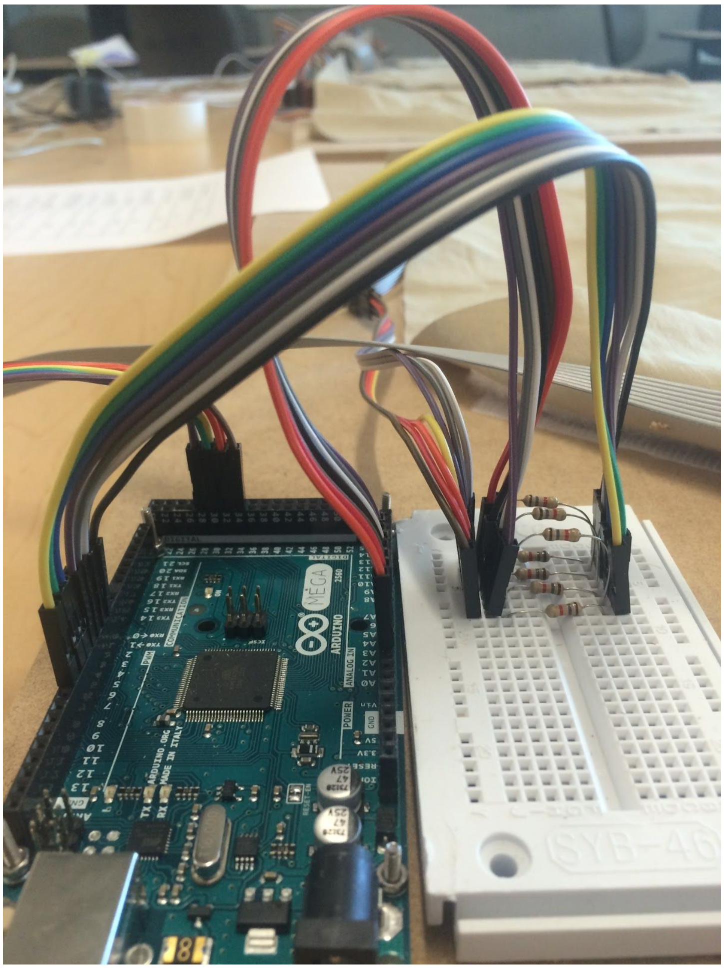


Figure F: Arduino and breadboard wiring, note the resistors' wiring with alternating high-low position so they do not touch.

Wiring tends to be the most time consuming aspect of creating this sensor, and it often involves a little bit of debugging. Just like any type of development, the best way to minimize debugging issues is to test incrementally as you go. You can create your own sensor visualization program to test the sensor as you wire it, and anytime you make a change to our circuit, you can run the program to see if something was wired incorrectly. The next section will offer some resources to get started on the sensor code.

5. ***Code***

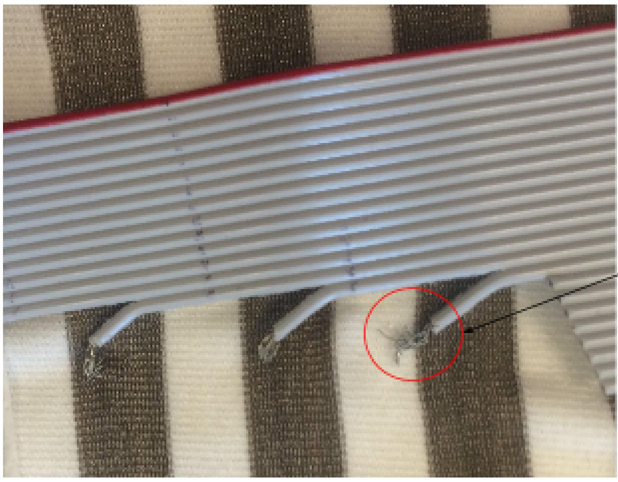
Create your own code for any project, or if you'd like to re-construct the touch sensor project, feel free to use the code developed here in SPIN. You can access the Arduino code for a 10"x10" touch sensor and a Matlab program to visualize the data on our Github page, here. (You can use the visualization program for observation during data collection, and for debugging the sensors.) As soon as your code is ready, you can upload it to your Arduino, and then you're all set to go!

Common Pitfalls

Here are some of the common mistakes, which you can now safely avoid:

1. Loose wire-fabric connections.

The problem with loose wires is that the connection is very weak, and there's a chance that the wires can accidentally extend beyond one conductive stripe, creating unpredictable results. One solution is to make sure the wire is threaded tightly through the fabric, also twisted into a single piece, and then folded into the fabric. As extra security you can apply masking tape over each connection individually, and then tape over the whole row / column.



Loose wire extending into the non-conductive stripe. This is what we want to avoid!

2. Jumper wiring errors.

It's really easy to connect the jumper wires into the wrong order, especially as the cables bend and twist, making it hard to see the ordering. One solution is to make use of the wire colouring scheme, double checking that the colour sequence matches on each side of the connection. It's also extremely helpful to test the circuit every time you make a change.

3. Resistors touching. (See Figure F for this solve.)

To avoid this you can wire resistors in an alternating staggered height format, with every even-index resistor flush to the breadboard, and every odd-index resistor at ~5cm height.

4. ***Bonding / adhering within the sensor boundary.***

If you apply the bonding glue or another adhesive within the sensor boundary, you are effectively killing the performance of the sensor in that location. To avoid this, leave an extra two inches of working fabric outside of the desired sensor area on each layer.

Conclusion

With this touch sensor system you can measure many different aspects of social touch, on any kind of substrate you desire. The flexibility and portability of the sensor lends well to draping it over anything from human skin, to soft robotics, to stiff substrates. For more information on touch sensing projects we're conducting in SPIN lab, visit our website here.