

# Data as a Material?

Tuning a sensor

Hfg Karlsruhe, 26 May 2021



```
solemaker.js
```

```
function redraw() {
  var voronoiCells = voronoiCellsGroup.selectAll(".voronoi-cell")
    .data(voronoi_polygons(voronoi_points))
    .enter()
    .append("path")
    .attr("class","voronoi-cell")
    .call(redrawPolygon);
  voronoiCells.call(redrawPolygon);
  voronoiCells.exit().remove();
  voronoiCellsGroup.selectAll(".voronoi-cell")
    .call(redrawPolygon);

  /**
   * Convert the outline points and voronoi seeds to sole fields
   */
  // Check if outline_points is round
  if(outline_points.length > 2) {
    var sole_polys = generate_cells_without_outline_inset(outline_points, voronoi_points, voronoi, sole_design);
    sole_cells = sole_polys;
    voronoi_cells = generate_cells_without_outline_inset(outline_points, voronoi_points, voronoi, sole_design);
    // var sliced_cells = slice_polygons(sole_polys);

    var sole_poly_selection = soleCellsGroup.selectAll(".sole-cell")
      .data(sole_polys, function(d) { return d; });
    sole_poly_selection
      .enter()
      .append("path")
      .attr("class","sole-cell")
      .attr("d", function(d) { return d ? "M" + d.join("L") + "Z" : null; });

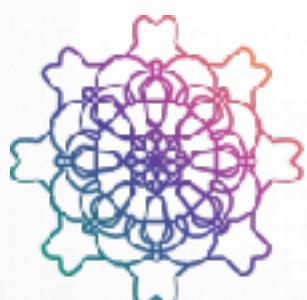
    sole_poly_selection.attr("d", function(d) { return d ? "M" + d.join("L") + "Z" : null; });
    sole_poly_selection.exit().remove();
  }

  var outline_circle = pointControlsGroup.selectAll(".outline-circle")
    .data(outline_points);
}

Line 1, Column 1
```

Tab Size: 4    JavaScript

**ArcInTexETN**  
Marie Skłodowska-Curie actions European Training Network (ETN)

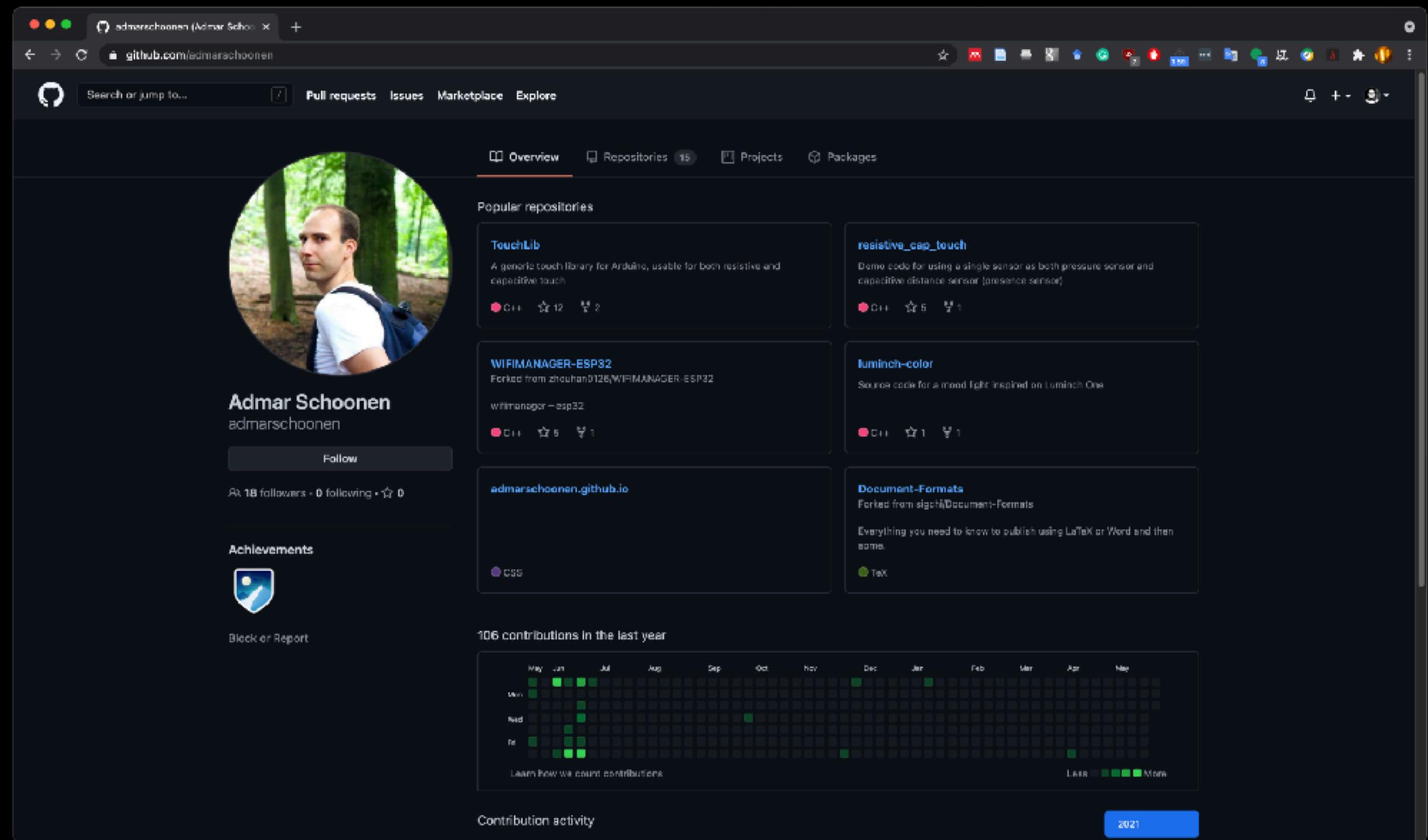


**WEAR  
sustain**



**Solemaker**

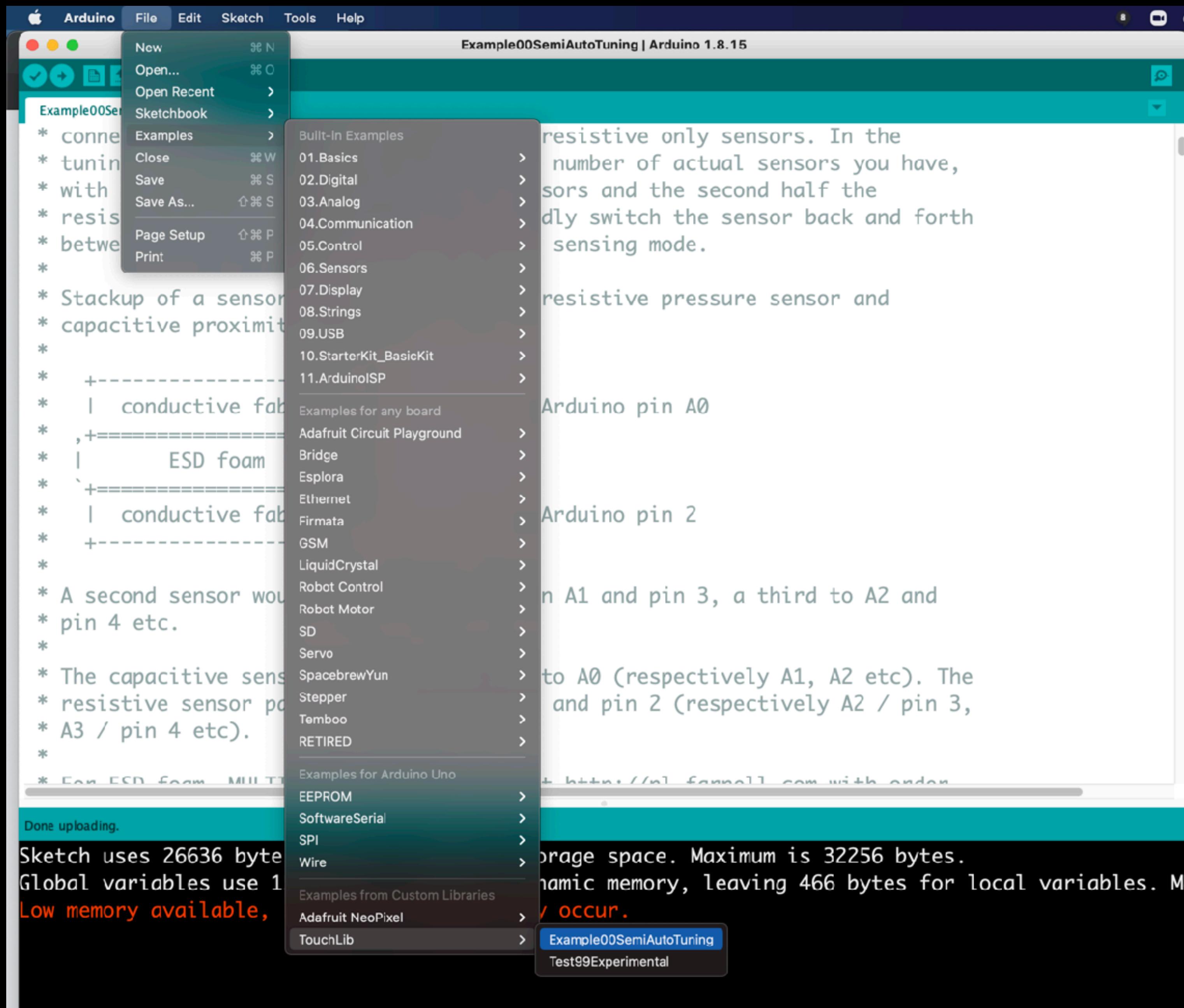
# TouchLib



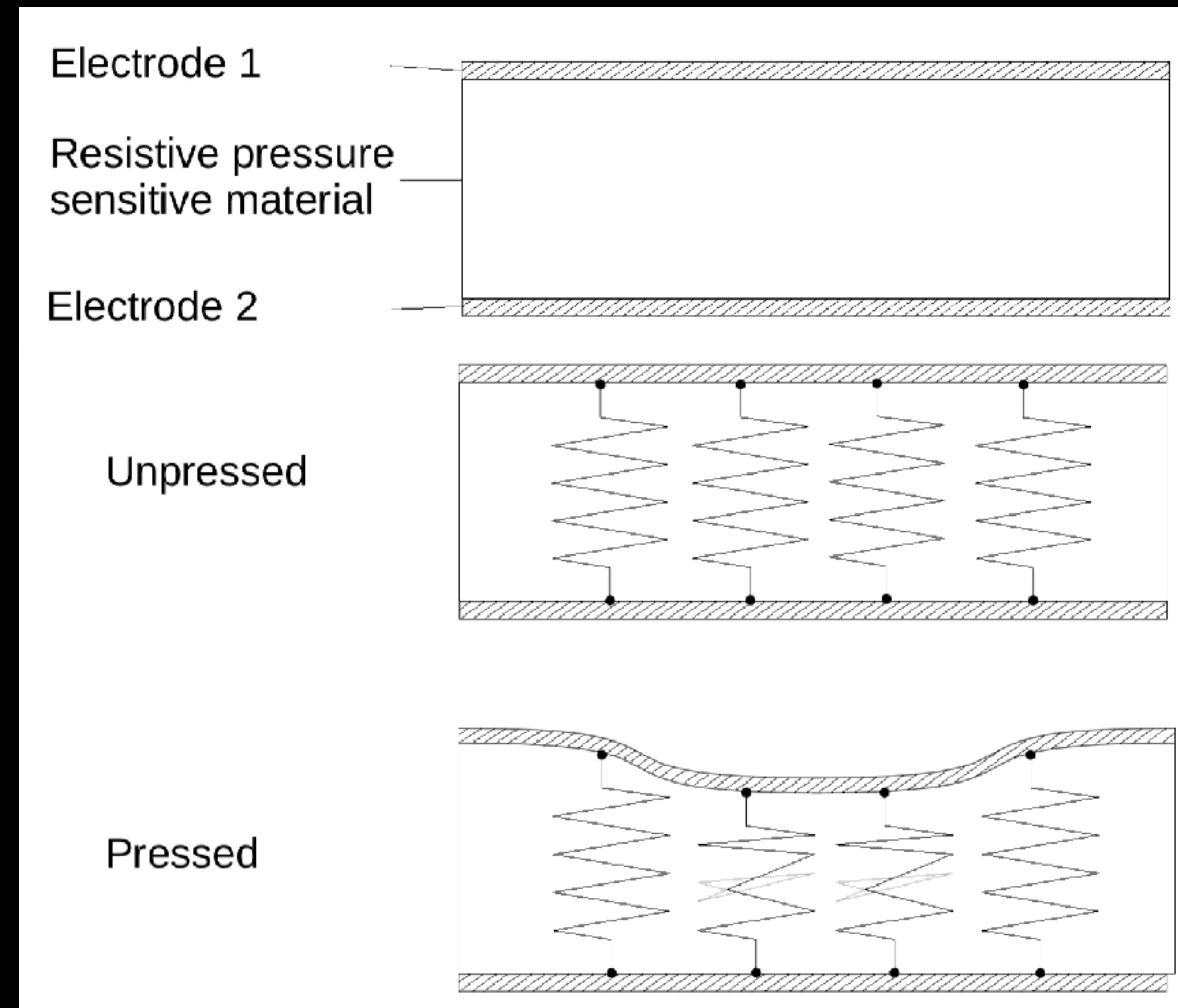
- <https://github.com/admarschoonen/TouchLib/archive/refs/tags/v0.0.14.zip>
- Unzip and install in ~/Documents/Arduino/libraries or C:\\My Documents\\Arduino\\libraries\\

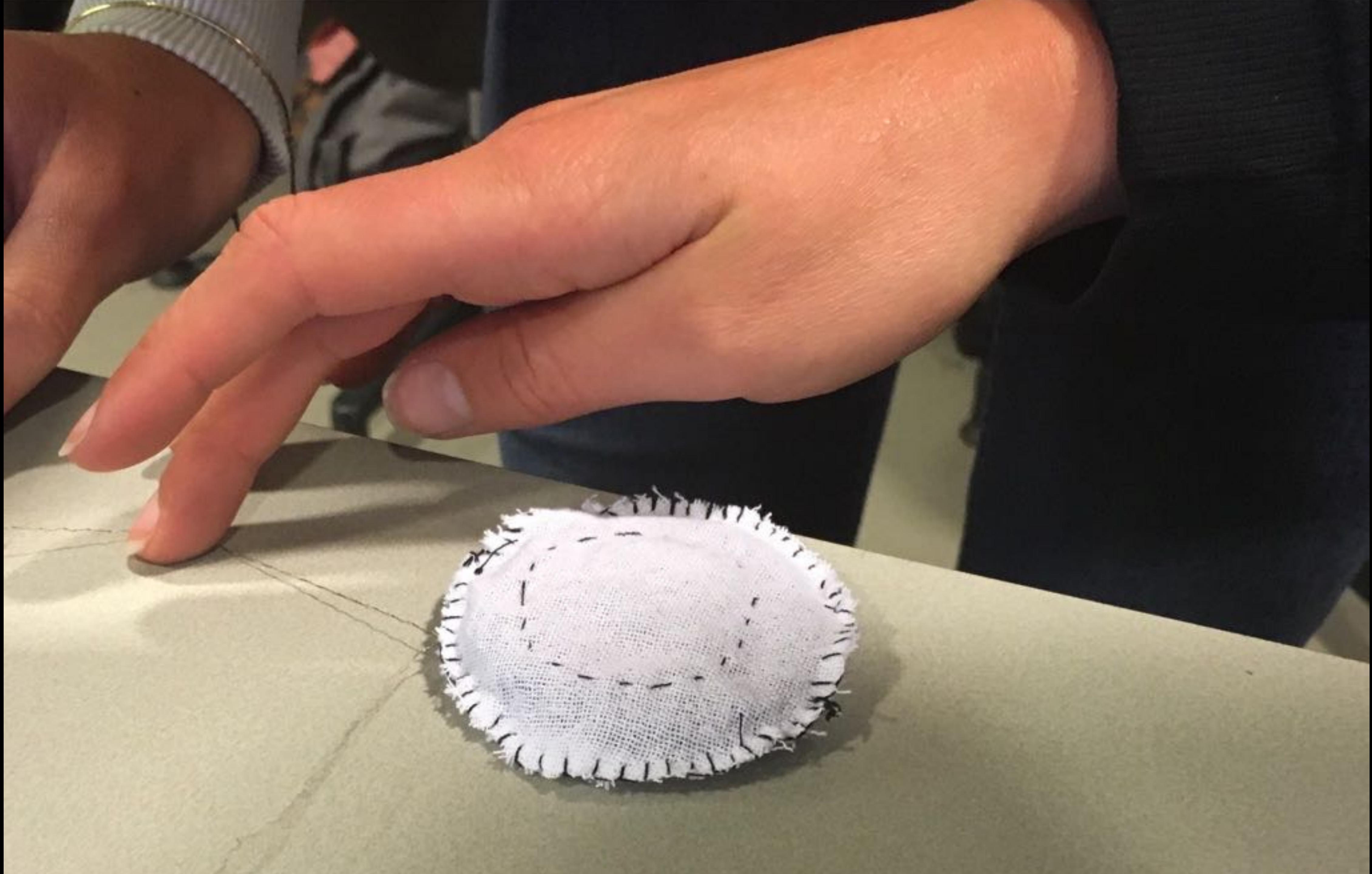
# The Plan

- Day 1 Morning:
- Day 1 Afternoon: The Data of Presence and Pressure, Start Making Pressure and Presence sensor (Textiles, Velostat, Eeyontex, Conductive and Normal Flexible Printing Filament, FilaFlex or NinjaFlex)
- Day 2 Morning: The sensor as data (TouchLIB)
- Day 2 Afternoon: Physical/Digital Natives: Making with Data, Using Code to move a 3D printer. (Processing and Spreadsheets).(3D printing, weaving,
- Day 3 Morning: Data as a blended material in physical-digital relationships. Embodied ideation session.(Aluminum shielding coated conductive material, opti-sensors)
- Day 3 Afternoon: Scaling physical-digital "Data as a material" into Ultra-Personalized Product Service Systems (Mural.com canvas)
- Day 4 Morning: The Aesthetics of the Date: Fine tuning the Interaction - Interactive Machine Learning (Second Samples of 3D printing, Weaving and Fabricating)
- Day 4 Afternoon: Symposium, Presentation of Results, and Reflections

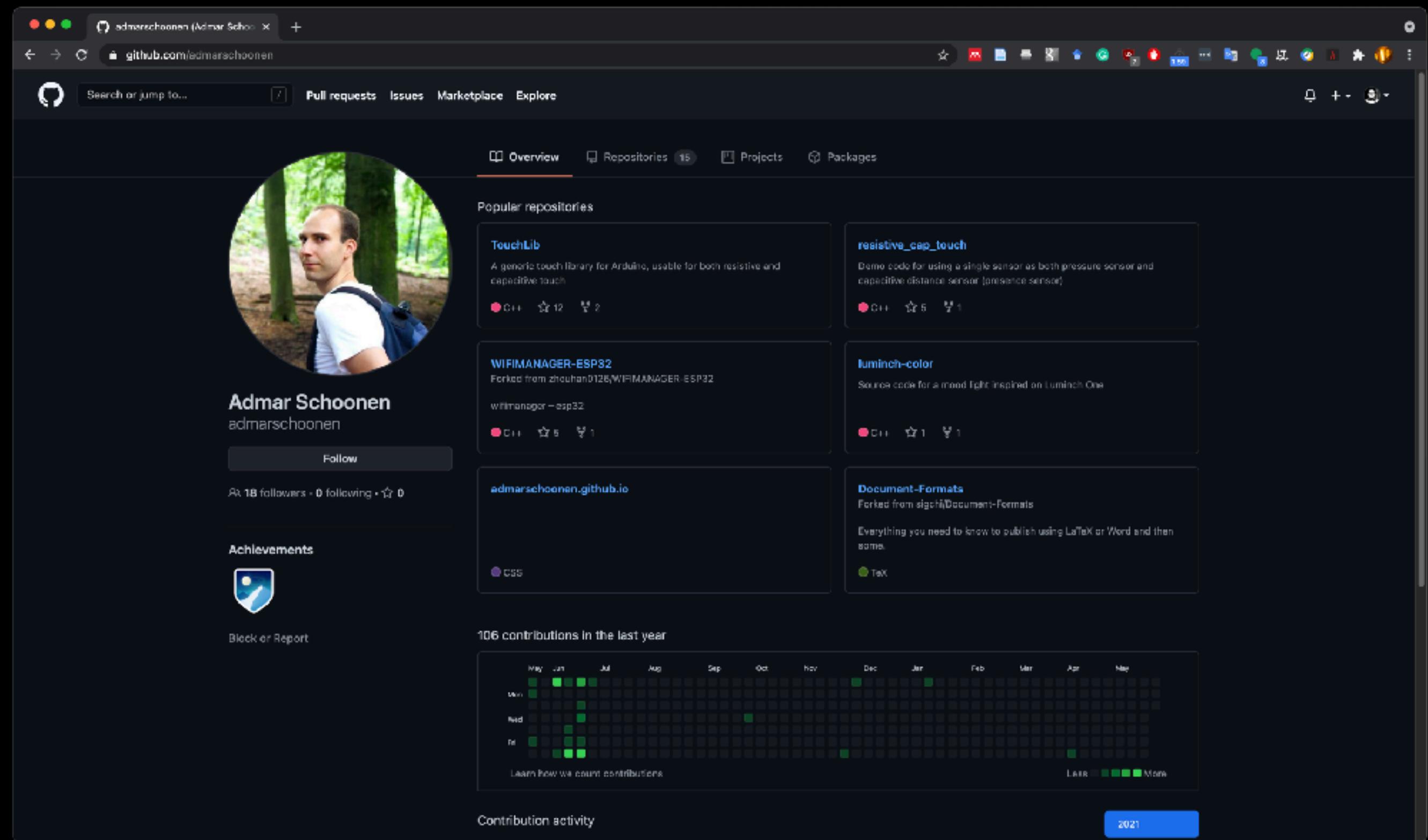


# Sandwich Sensor



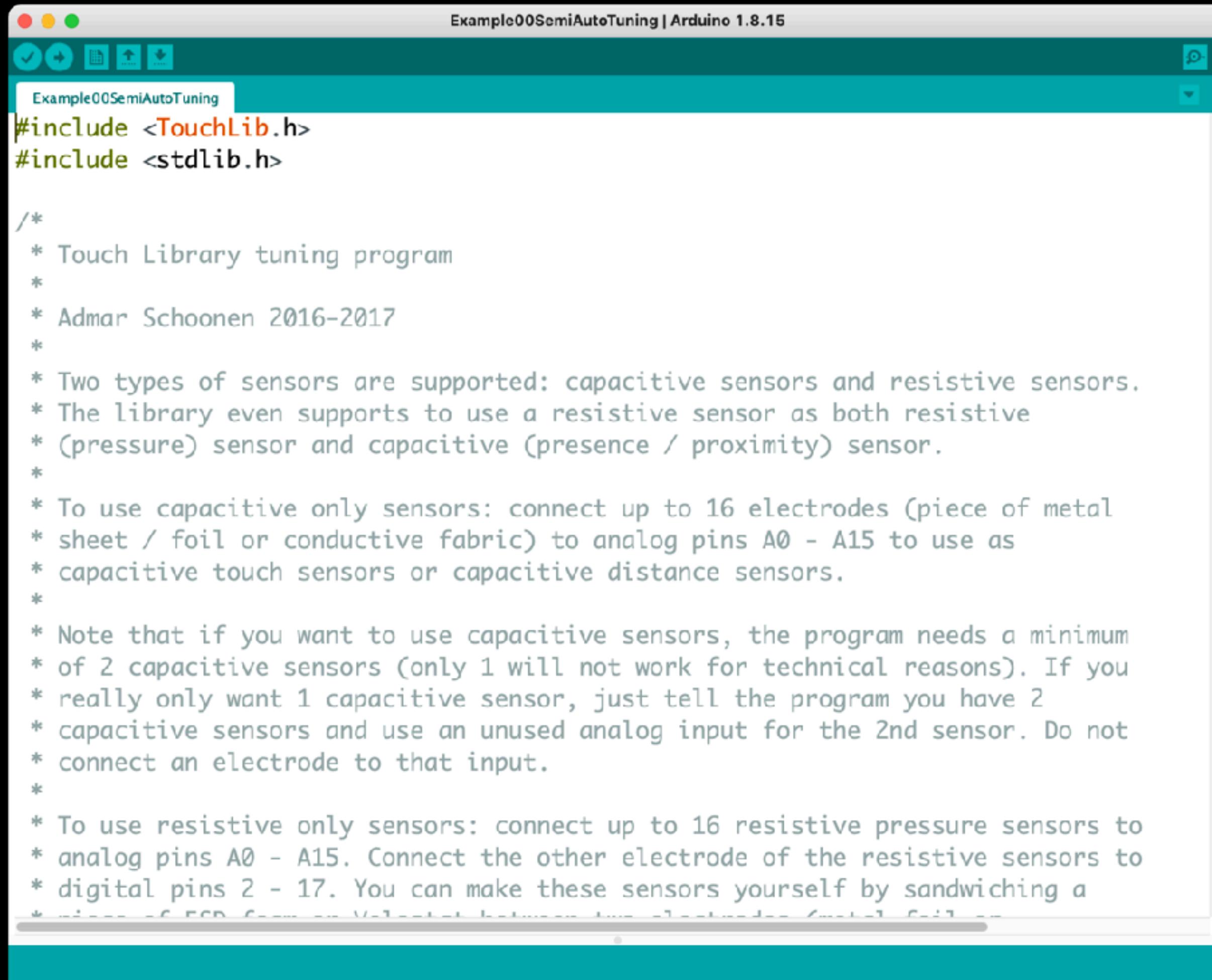


# TouchLib



- <https://github.com/admarschoonen/TouchLib/archive/refs/tags/v0.0.14.zip>
- Unzip and install in ~/Documents/Arduino/libraries or C:\\My Documents\\Arduino\\libraries\\

# Open Examples/TouchLib/ Example00SemiAutoTuning



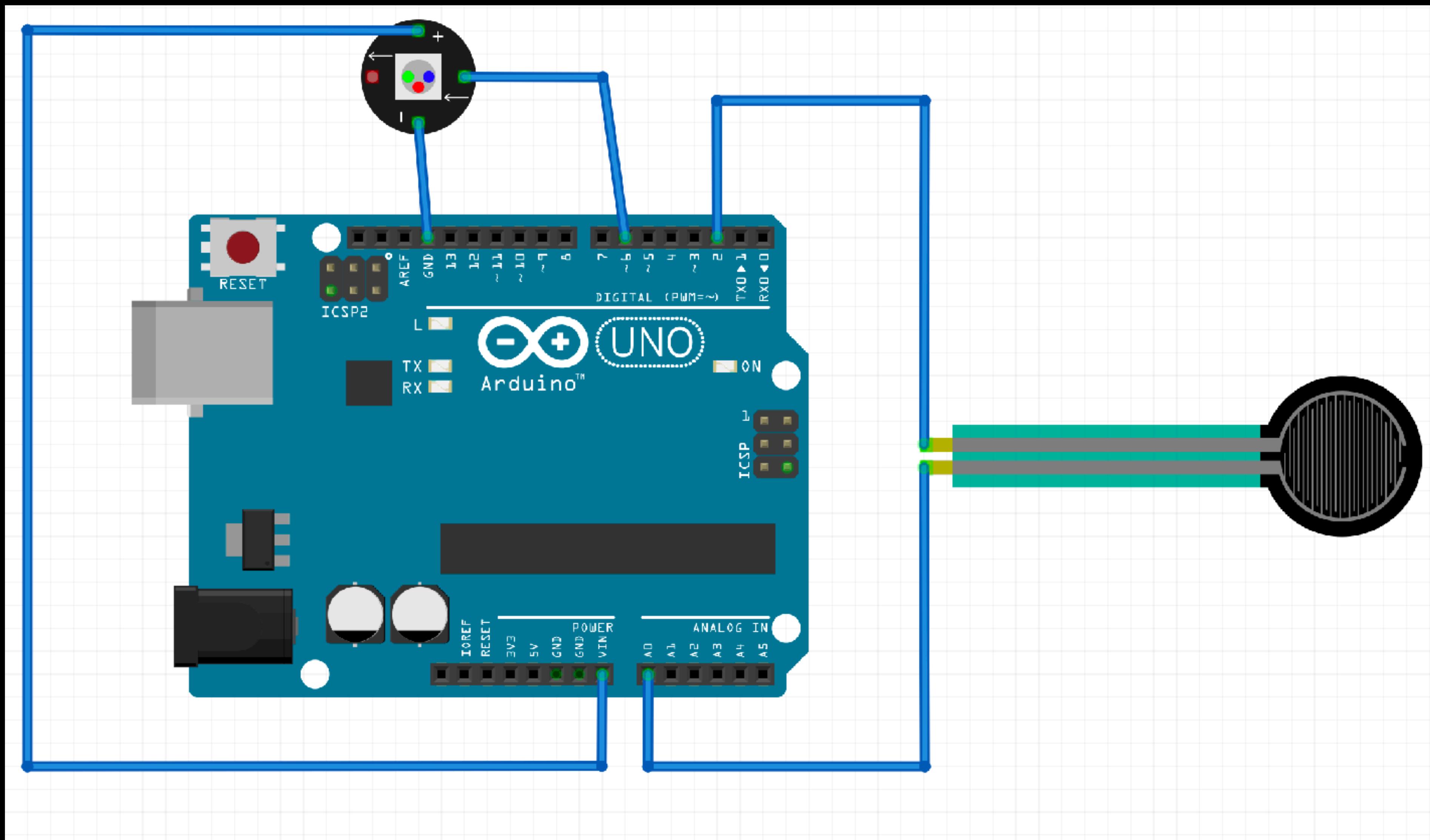
The screenshot shows the Arduino IDE interface with the title bar "Example00SemiAutoTuning | Arduino 1.8.15". The code editor contains the following C++ code:

```
#include <TouchLib.h>
#include <stdlib.h>

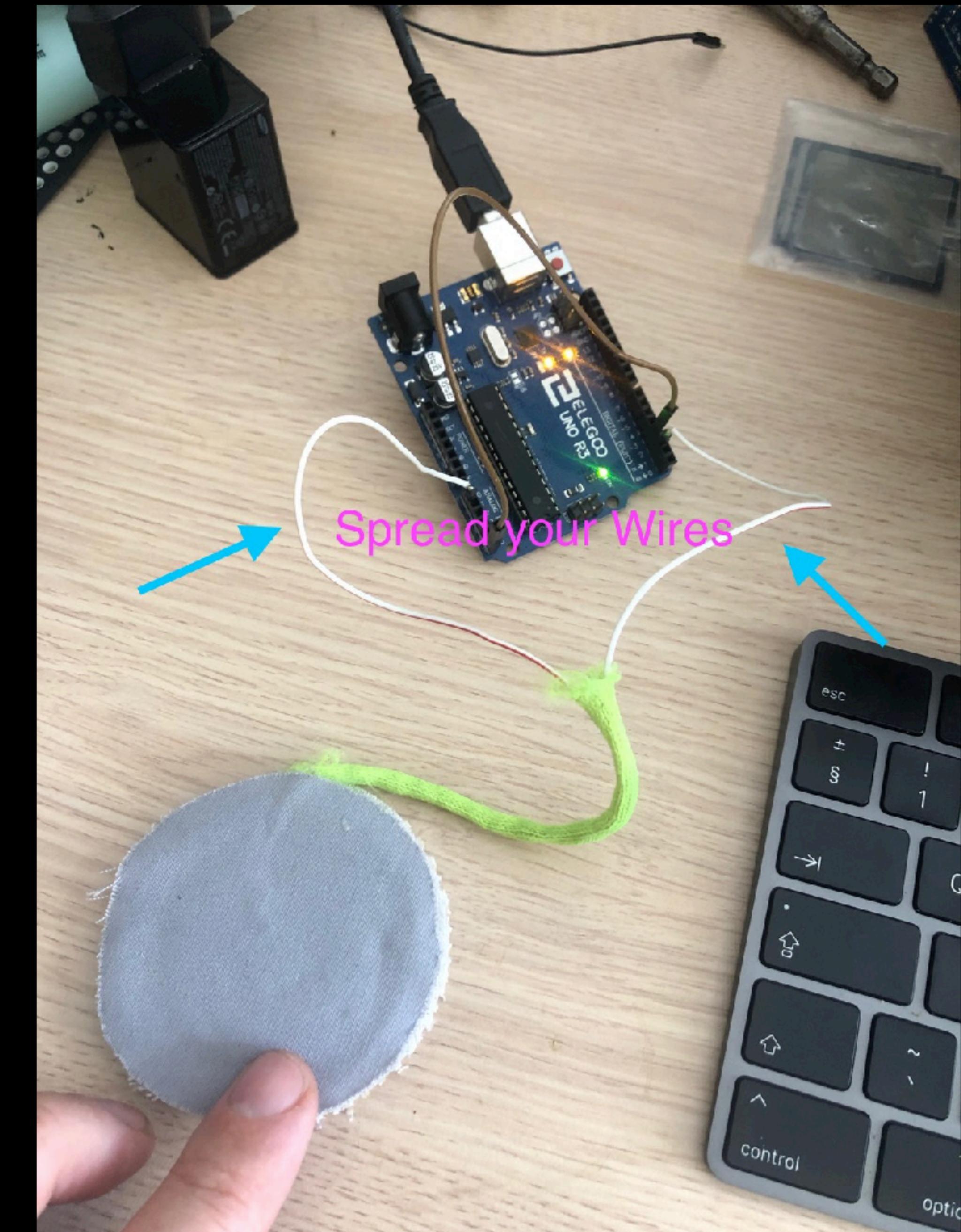
/*
 * Touch Library tuning program
 *
 * Admar Schoonen 2016-2017
 *
 * Two types of sensors are supported: capacitive sensors and resistive sensors.
 * The library even supports to use a resistive sensor as both resistive
 * (pressure) sensor and capacitive (presence / proximity) sensor.
 *
 * To use capacitive only sensors: connect up to 16 electrodes (piece of metal
 * sheet / foil or conductive fabric) to analog pins A0 - A15 to use as
 * capacitive touch sensors or capacitive distance sensors.
 *
 * Note that if you want to use capacitive sensors, the program needs a minimum
 * of 2 capacitive sensors (only 1 will not work for technical reasons). If you
 * really only want 1 capacitive sensor, just tell the program you have 2
 * capacitive sensors and use an unused analog input for the 2nd sensor. Do not
 * connect an electrode to that input.
 *
 * To use resistive only sensors: connect up to 16 resistive pressure sensors to
 * analog pins A0 - A15. Connect the other electrode of the resistive sensors to
 * digital pins 2 - 17. You can make these sensors yourself by sandwiching a
 * thin piece of metal between two pieces of insulating material.
```

- Reboot Arduino

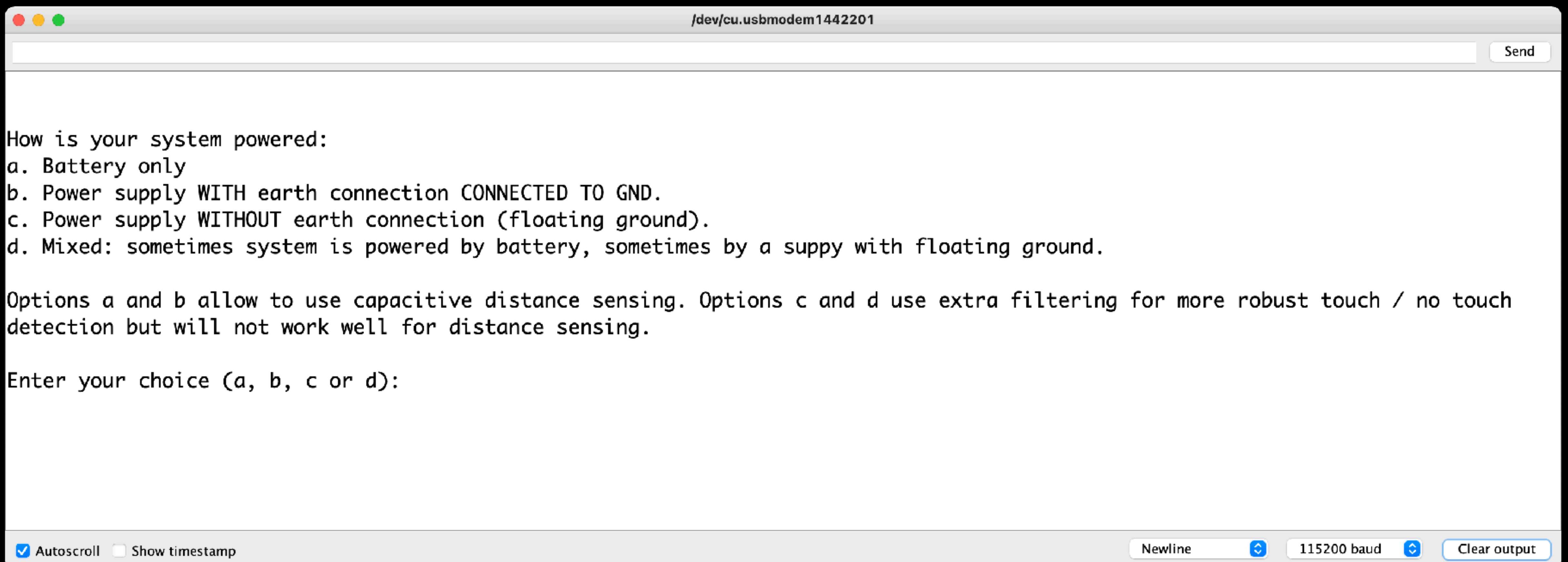
# Today's Circuit



# Spread your Wires



# Presence and Ground



# A Convenient Lie...

```
How many sensors do you want to tune? (1 - 6) 3 ←
First we need to know what type of sensors the system has and to which pins they are connected.
Is sensor 0 a capacitive sensor using CVD method or a resistive sensor using analogRead() method?
Enter c for capacitive using CVD or r for resistive: c ←
Which analog pin is sensor 0 connected to?
A0 ←
Is sensor 1 a capacitive sensor using CVD method or a resistive sensor using analogRead() method?
Enter c for capacitive using CVD or r for resistive: c ←
Which analog pin is sensor 1 connected to?
a5 ←
Is sensor 2 a capacitive sensor using CVD method or a resistive sensor using analogRead() method?
Enter c for capacitive using CVD or r for resistive: r ←
Which analog pin is sensor 2 connected to?
A0 ←
Which digital pin is the resistive sensor also connected to (used as ground pin)? 2 ←
```

Autoscroll  Show timestamp  Newline  115200 baud  Clear output

# Stand Back, Really.

```
Next step is to tune the sensors.  
Performing noise measurement for all sensors. Make sure to not touch any sensor. Enter y to start the noise measurement. y  
Noise measurement started...  
Noise measurement finished. Found the following thresholds:  
Sensor 0:  
released -> approached: 31  
approached -> released: 28  
  
Sensor 1:  
released -> approached: 335  
approached -> released: 302  
  
Sensor 2:  
released -> approached: 18  
approached -> released: 17
```

Autoscroll  Show timestamp      Newline  115200 baud  Clear output

# Sensor 0: 1 Finger, Whole Hand

```
/dev/cu.usbmodem1442201
Send

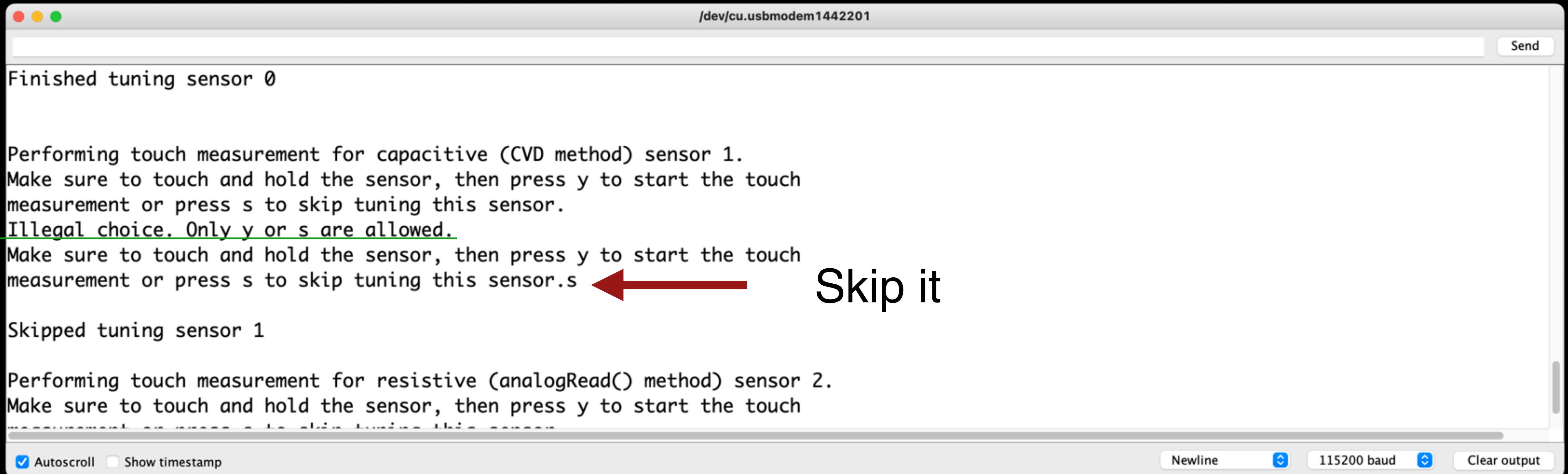
Performing touch measurement for capacitive (CVD method) sensor 0.
Make sure to touch and hold the sensor, then press y to start the touch
measurement or press s to skip tuning this sensor.
Illegal choice. Only y or s are allowed.
Make sure to touch and hold the sensor, then press y to start the touch
measurement or press s to skip tuning this sensor.y ← One Finger
Found the following thresholds:
    approached -> pressed: 127
    pressed -> approached: 114

Performing maximum range measurement for capacitive (CVD method) sensor 0. Make sure to cover and hold the sensor with as many fingers as
fit or with your whole hand, then press y to start the touch measurement.
Illegal choice. Only y is allowed. Make sure to cover and hold the sensor with as many fingers as will
fit or with your whole hand, then press y to start the touch measurement. y ← Whole Hand
Found the following maximum: 947

Finished tuning sensor 0

Autoscroll Show timestamp Newline 115200 baud Clear output
```

# Sensor 1: Forget about it, (or connect pin A5 to pin 4 for better tuning)



```
/dev/cu.usbmodem1442201
Send

Finished tuning sensor 0

Performing touch measurement for capacitive (CVD method) sensor 1.
Make sure to touch and hold the sensor, then press y to start the touch
measurement or press s to skip tuning this sensor.
Illegal choice. Only y or s are allowed.
Make sure to touch and hold the sensor, then press y to start the touch
measurement or press s to skip tuning this sensor.s ← Skip it
Skipped tuning sensor 1

Performing touch measurement for resistive (analogRead() method) sensor 2.
Make sure to touch and hold the sensor, then press y to start the touch
measurement or press s to skip tuning this sensor.

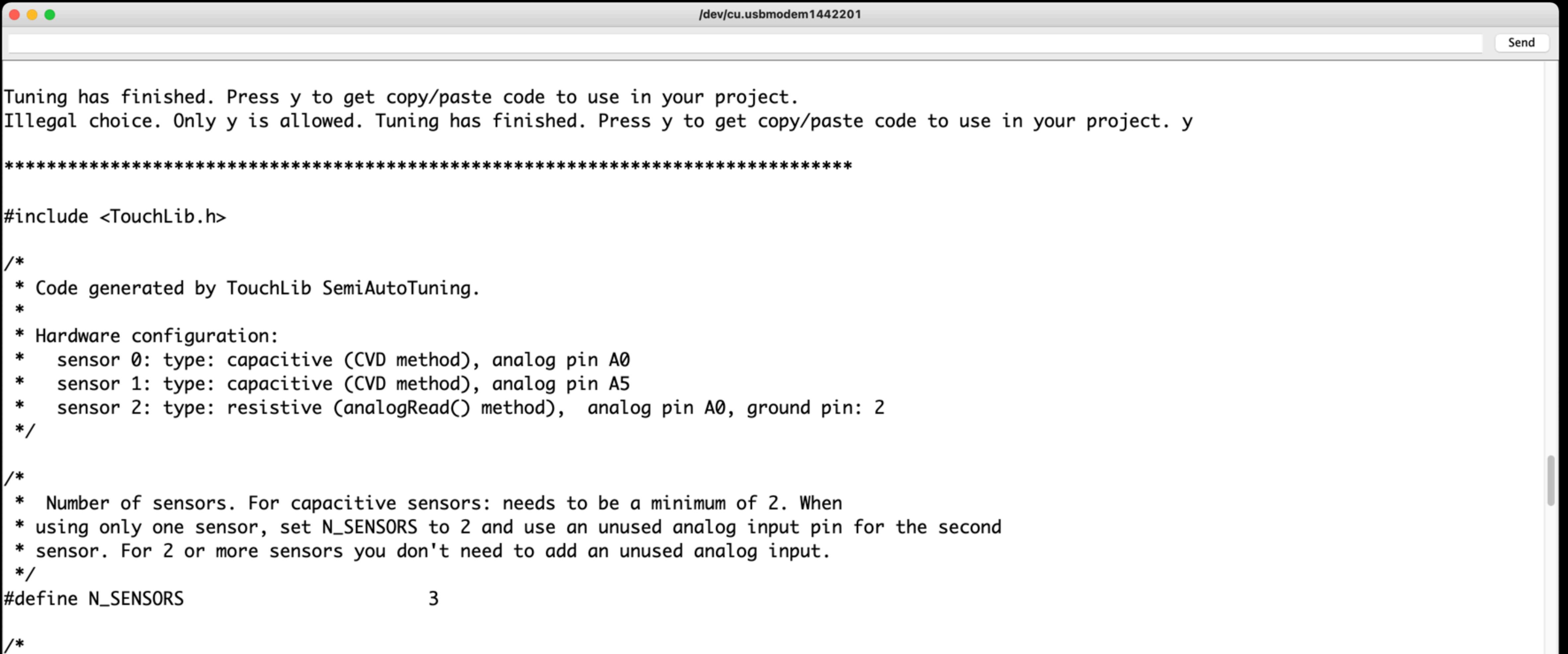
Autoscroll  Show timestamp Newline  Clear output
```

# Resistive Tuning

```
Performing touch measurement for resistive (analogRead() method) sensor 2.  
Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.  
Illegal choice. Only y or s are allowed.  
Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.y ←  
Error! Detected signal is too low: -175 < 32. Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.  
Illegal choice. Only y or s are allowed.  
Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.y  
Error! Detected signal is too low: -49 < 32. Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.  
Illegal choice. Only y or s are allowed.  
Make sure to touch and hold the sensor, then press y to start the touch  
measurement or press s to skip tuning this sensor.y  
Found the following thresholds:  
    approached -> pressed: 241  
    pressed -> approached: 217  
  
Performing maximum range measurement for resistive (analogRead() method) sensor 2. Make sure to press and hold the sensor as firmly as possible, then  
press y to start the touch measurement.  
Illegal choice. Only y is allowed. Make sure to press and hold the sensor as firmly as possible, then  
press y to start the touch measurement. y ←  
Found the following maximum: 750  
  
Finished tuning sensor 2
```

Autoscroll  Show timestamp      Newline      115200 baud      Clear output

# Copy and paste your new code into a new sketch



The screenshot shows a terminal window titled '/dev/cu.usbmodem1442201' with a 'Send' button in the top right corner. The window displays the following text:

```
Tuning has finished. Press y to get copy/paste code to use in your project.  
Illegal choice. Only y is allowed. Tuning has finished. Press y to get copy/paste code to use in your project. y  
*****  
  
#include <TouchLib.h>  
  
/*  
 * Code generated by TouchLib SemiAutoTuning.  
 *  
 * Hardware configuration:  
 *   sensor 0: type: capacitive (CVD method), analog pin A0  
 *   sensor 1: type: capacitive (CVD method), analog pin A5  
 *   sensor 2: type: resistive (analogRead() method), analog pin A0, ground pin: 2  
 */  
  
/*  
 * Number of sensors. For capacitive sensors: needs to be a minimum of 2. When  
 * using only one sensor, set N_SENSORS to 2 and use an unused analog input pin for the second  
 * sensor. For 2 or more sensors you don't need to add an unused analog input.  
 */  
#define N_SENSORS  
/*
```



# Smoothing



The screenshot shows the Arduino IDE interface with the title bar "sketch\_may25a | Arduino 1.8.15". The code editor contains the following C++ code:

```
#include <TouchLib.h>

/*
 * Code generated by TouchLib SemiAutoTuning.
 *
 * Hardware configuration:
 *   sensor 0: type: capacitive (CVD method), analog pin A0
 *   sensor 1: type: capacitive (CVD method), analog pin A5
 *   sensor 2: type: resistive (analogRead() method), analog pin A0, ground pin: 2
 */

/*
 * Number of sensors. For capacitive sensors: needs to be a minimum of 2. When
 * using only one sensor, set N_SENSORS to 2 and use an unused analog input pin for the second
 * sensor. For 2 or more sensors you don't need to add an unused analog input.
 */
#define N_SENSORS 3

/*
 * Number of measurements per sensor to take in one cycle. More measurements
 * means more noise reduction / spreading, but is also slower.
 */
#define N_MEASUREMENTS_PER_SENSOR 32 ←
```

The status bar at the bottom indicates "Done uploading." and "Sketch uses 14670 bytes (45%) of program storage space. Maximum is 32256 bytes."

# tlSensors.getState(n)

# tlSensors.getState(k)

- 3 = Released
- 5 = Approached
- 8 = Pressed
- If (tlSensors.getState(n) == 3) { //Released  
  pixels.setPixelColor(i, pixels.Color(0, 150, 0)); //greenish }  
elseif (tlSensors.getState(n) == 5) { //Approached  
  pixels.setPixelColor(i, pixels.Color(0, 0, 150)); //blueish }  
elseif (tlSensors.getState(n) == 8) { //Pressed  
  pixels.setPixelColor(i, pixels.Color(150, 0, 150)); //purple }  
else{  
  pixels.setPixelColor(i, pixels.Color(0, 0, 0)); //black }  
pixels.show(); // Send the updated pixel colors to the hardware.

# Today's Circuit

