Multi-Touch Kit

A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing Using a Commodity Microcontroller
-- Step-by-Step Tutorial --

Narjes Pourjafarian[1], Anusha Withana[2], Jan Dickman[1],
Joe Paradiso[3], Juergen Steimle[1]

[1]Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
[2]School of Computer Science, The University of Sydney, Sydney, Australia
[3]Responsive Environments, MIT Media Lab, Cambridge, Massachusetts, United States

Version 1.0, Date 19.10.2019

This document contains a step-by-step tutorial to build and set up a Multi-Touch sensor using the *Multi-Touch Kit* software toolkit and a commodity microcontroller.

Project Webpage (including video, link to source code, etc.):
https://hci.cs.uni-saarland.de/multi-touch-kit/

Reference:
If you use this toolkit, please reference the scientific publication as follows:
http://dx.doi.org/10.1145/3332165.3347895

Requirements

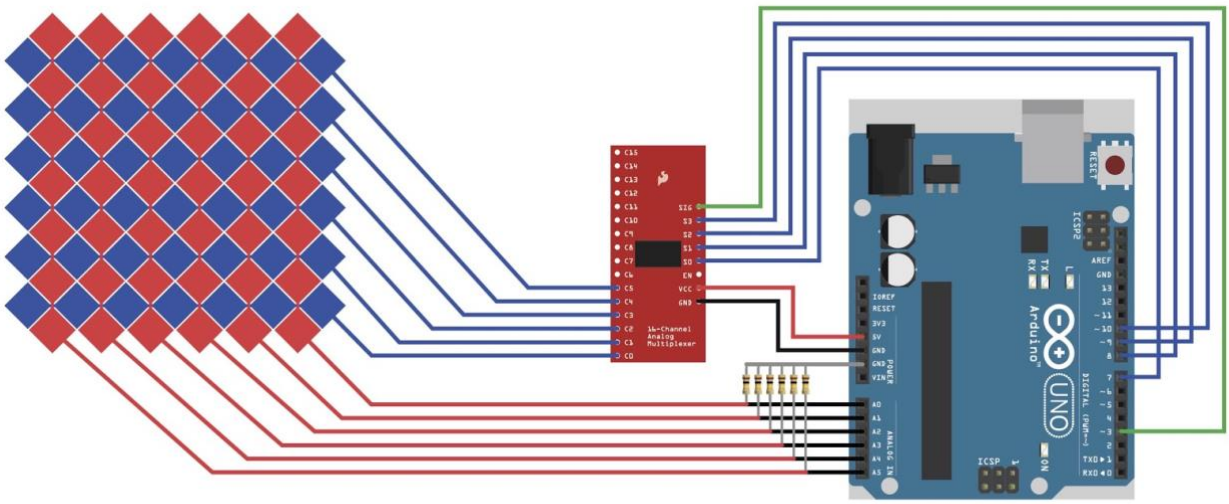
Hardware Setup
- Arduino (Mega2560, Uno, LilyPad)
- Multiplexer (e.g., CD74HC4067[1])
- Resistors (100K Ohm)

Software Setup
- Arduino IDE
- Processing 3
- *Multi-touch Kit* Processing and Arduino Library
- BlobDetection and OpenCV Libraries for Processing

Multi-Touch Sensor
- Some inexpensive materials for building the multi-touch sensor (see the section for details)

Hardware Setup

To build the circuit, see the circuit diagram below and follow the instructions:



[1] e.g., https://www.sparkfun.com/products/9056

1. Connect Arduino to the multiplexer

    a. Use a standard multiplexer (e.g., CD74HC4067, < $1) which has a bandwidth to work in the functional frequency range (above 10MHz).
    b. Connect the VCC and GND pins of the multiplexer to the 5V and GND pins of the Arduino, respectively.
    c. Connect Select pins of the Multiplexer (S0, S1, S2, and S3) to the Arduino digital pins. (e.g., Arduino Uno: S0~7, S1~8, S2~9, and S3~10, and Arduino Mega: S0~48, S1~49, S2~50, and S3~51)
    d. Connect the Signal pin (SIG) of the multiplexer to the Arduino (Arduino Uno/Lily: digital Pin 3, Arduino Mega: digital Pin 9)

2. Connect multiplexer to the touch sensor[2]

    e. Connect the output pins of the multiplexer (starting at c0) to the transmission lines of the sensor (blue lines connected to the sensor in the circuit diagram)
    f. Connect the receiver lines of the sensor to the analog pins of the Arduino (starting at A0) (red lines connected to the sensor in the circuit diagram)
    g. Connect all receiver lines to the ground of the circuit with 100k Ohm resistors on each line

Software Setup

Arduino Library

1. Download and install Arduino IDE:
   https://www.arduino.cc/en/Main/Software

2. Download and install the *Multi-Touch Kit* library from the link below:
   https://github.com/HCI-Lab-Saarland/MultiTouchKit

   (You can find more information on how to install additional Arduino Libraries in[3])
   Sketch→ Library → Add Zip Library

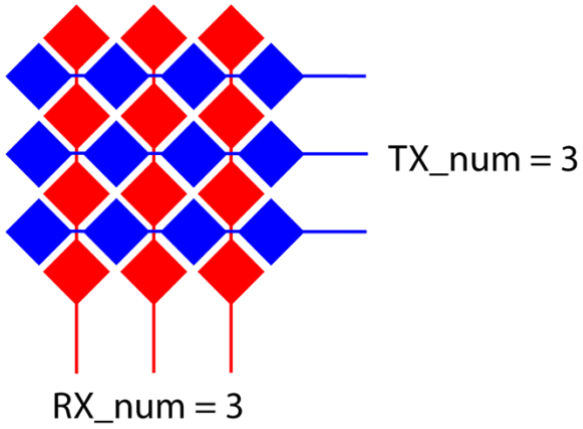
3. Open the example sketch in the library based on your Arduino hardware (Mega or UNO)
   File→ Examples → Multi-Touch Kit → Select appropriate

[2] Find more information below on how you can quickly fabricate your own multi-touch sensor

[3] https://www.arduino.cc/en/guide/libraries

4. Based on the number of the transmitter (TX) and receiver (RX) lines of the multi-touch sensor, adjust *TX_num* and *RX_num* variables

```
int RX_num = 3;
int TX_num = 3;
```



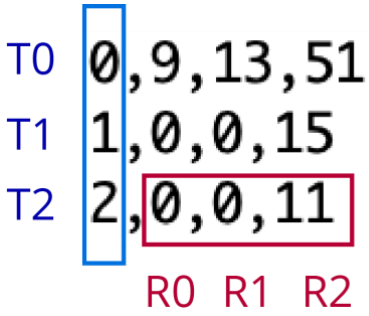
5. Select raw capacitance data or thresholded touch up/down data:
   Set the Boolean variable raw_data to define what data you want to receive from the toolkit

   a. *raw_data = true* : For access to the raw capacitance data of touch events, set raw_data to true. (In this case, you don't need to change the variable threshold.)
   b. *raw_data = false* : For access to thresholded (binary) touch down/touch up data, you need to set the raw_data to false and adjust the threshold variable based on your multi-touch sensor (See FAQ-1 below to learn how to adjust the threshold). If the value is exceeding the threshold, a touch down is reported; otherwise, touch up.

```
boolean raw_data = true;
int threshold = 30;
```

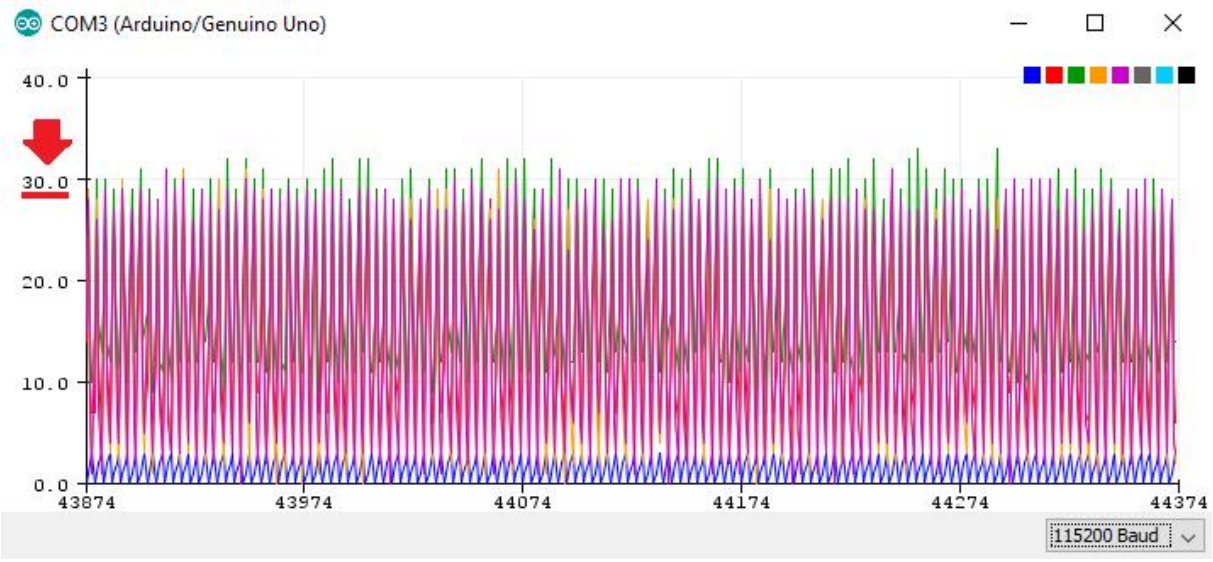
6. Upload the sketch to the Arduino microcontroller (Make sure the correct Port/Board is selected)

7. The Arduino will continuously send the values via the serial port, frame after frame. You will see several lines of data that are separated by commas. The first number of each line is the current TX number and the following numbers are the values measured at the RX lines, starting at R0. The example below shows one frame of raw data from a 3x3 sensor. You can directly use this data in your application without the Processing library. The Processing library can be optionally used and offers higher-level functionality for processing and visualizing multi-touch data.

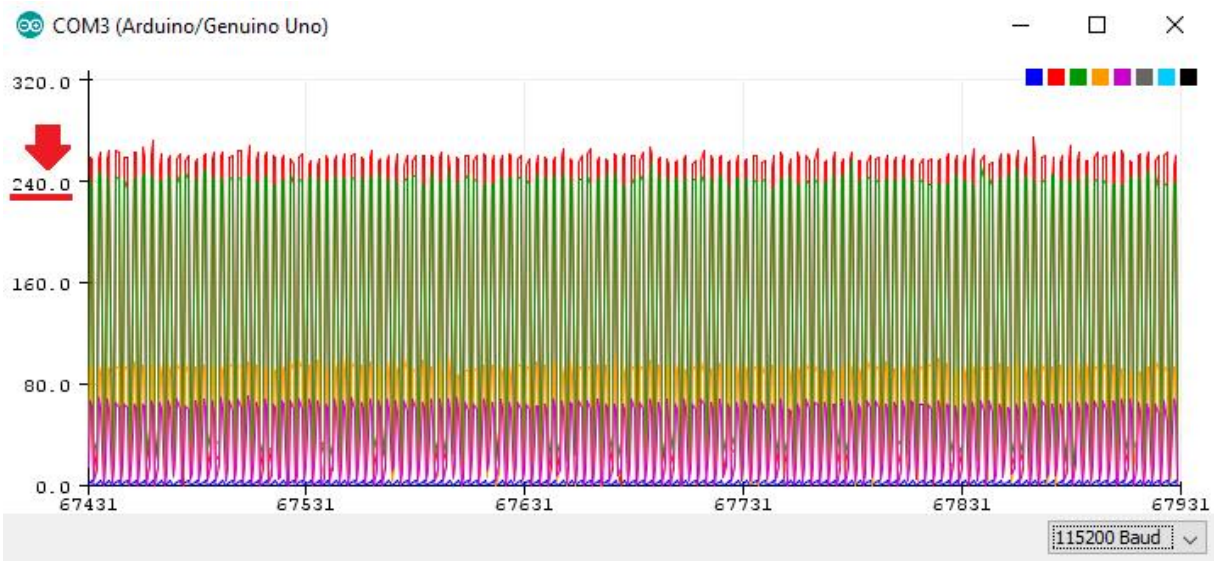


8. For testing, open the serial plotter and touch the sensor. Make sure that the serial plotter is set to the correct Baud Rate (115200 by default). You will see an output similar to this one here:

No-touch:



Touch:



Processing Library

The Processing library of *Multi-Touch Kit* offers higher-level functionality to process, record and visualize touch data.

1. Install Processing IDE:
   https://processing.org/download/

2. Install the opencv for Processing and BlobDetection Libraries (Processing > Sketch > import library > add library > filter: *type in opencv or blobdetection* > install)

3. Download the *MultiTouchKit* Processing Library and install it:
   https://github.com/HCI-Lab-Saarland/MultiTouchKitUI

4. Open the example sketch in the Library (MTK_example)

5. Based on the number of transmitter and receiver lines of the Multi-touch sensor, adjust the tx and rx variables

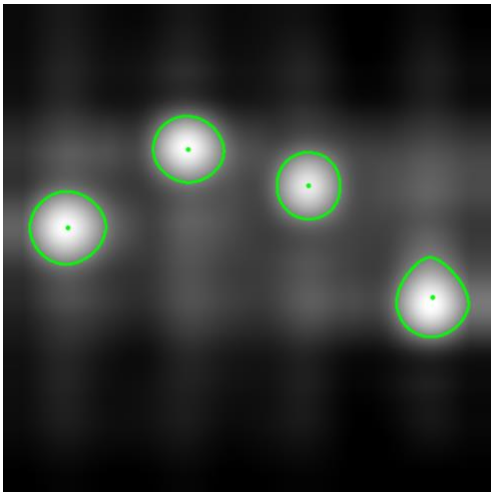
```
int tx = 16;
int rx = 16;
```

6. Set the index of serial port correctly (Check the FAQ-2 to learn how to find the right port)

7. Run the sketch.

8. A window will open which shows “Calibrating”. Please wait until this message disappears. After that, you can start using the sensor.

9. When you touch the sensor, you will see similar results as follows.



10. Follow the instructions in the example sketch in the Library to adjust your values/thresholds if necessary (remember that you have to restart the sketch so that changes in the code can affect).

Making the multi-touch sensor

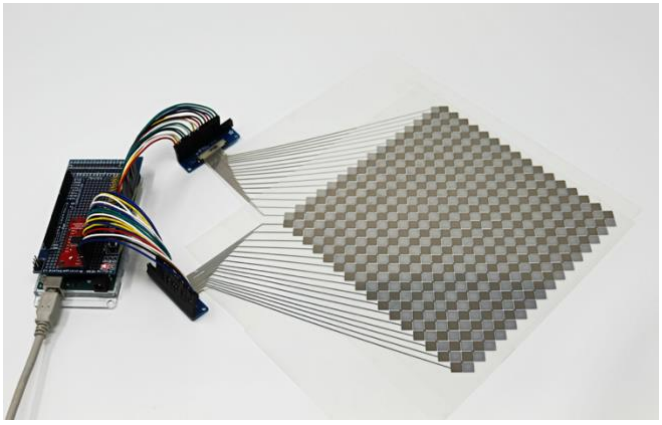
The Multi-Touch Kit libraries are compatible with sensors made using various ways. For instance, custom sensors can be inkjet-printed using a desktop printer and silver ink; textile multi-touch sensors can be fabricated using conductive yarn; flexible sensors can also be fabricated using copper sheet or a silver pen. We will provide details in this section.

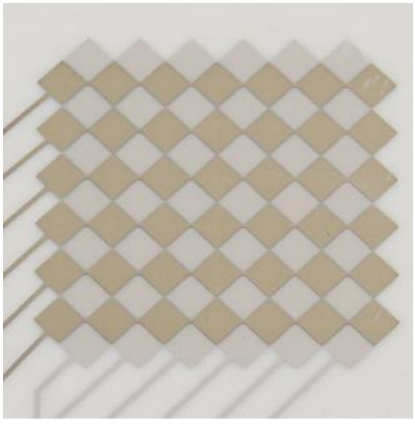
No matter how the sensor is fabricated, we recommend using the classical two-layered diamond pattern that is commonly used for mutual-capacitive touch sensing [4]. For the design of electrodes of the sensor, we included examples of SVG files for sensors with 8x8 and 16x16 electrodes at the project's GitHub page. You can directly print those files or modify the designs (e.g., different number of TX and RX lines) to make the design meet your requirements:

https://github.com/HCI-Lab-Saarland/MultiTouchKitDoc/tree/master/MultiTouchSensor

Inkjet Printed Flexible Multi-Touch Sensor on PET Film

A sensor can be printed on transparent PET film using a Canon IP100 desktop inkjet printer and conductive silver ink[5]. The TX and RX electrode layers are printed on separate PET films and then attached with a very thin layer of the adhesive film (e.g., temporary tattoo adhesive). The top surface of the sensor is insulated with a thin layer of transparent dielectric (e.g., overhead transparent film). An example sensor is shown below:

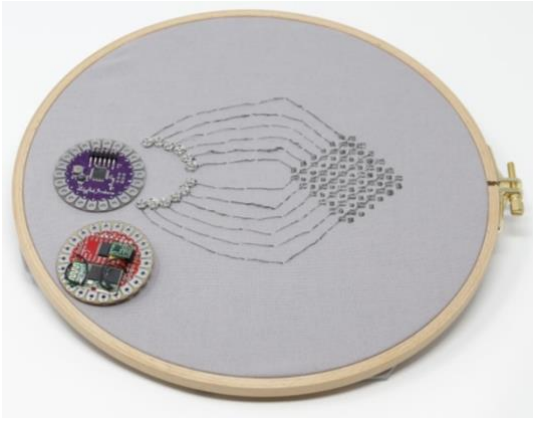


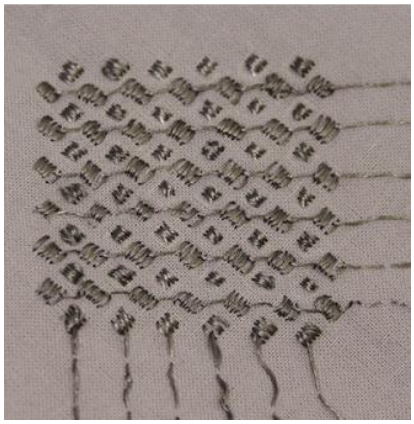


[4] http://ww1.microchip.com/downloads/en/DeviceDoc/FAQs%20-%20Sensor%20Design%20Guidelines.pdf

Textile Multi-Touch Sensor with Conductive Yarn

A textile multi-touch sensor can be realized by stitching the electrode pattern with conductive yarn (e.g., Adafruit Stainless Thin Conductive Thread). To do it manually, the sensor patterns are drawn on both sides of the fabric (TX electrodes on one side, RX electrodes on the other side), and then the patterns are stitched using conductive yarn. Pay attention that none of the electrodes should be connected except those belonging to the same RX or TX line. Connect RX electrodes on one side of the textile, and TX line on the other side.
Alternatively, it is possible to use a programmable sewing machine with conductive yarn. After stitching, the sensor is isolated with coating spray (e.g., Kontakt Chemie 74313-AA). An example of a hand-stitched sensor, for connection to an Arduino Lilypad, is shown below:

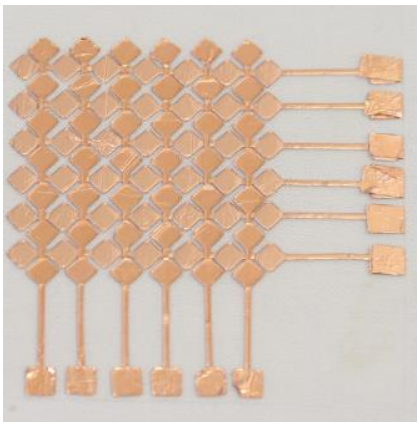




Multi-Touch Sensor made of Copper Sheet

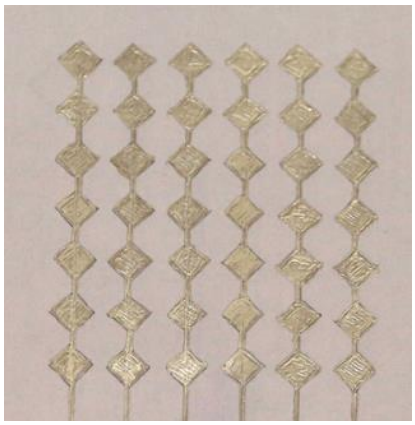
Another method is to create the electrodes out of a conductive copper sheet. You could use scissors or other tools to cut the design manually. Using a computer-controlled cutter machine, e.g., a Brother ScanNCut will be faster and more accurate. The electrodes can be put on opposite sides of a plastic sheet (e.g., overhead transparencies). An alternative method is to put down the first layer on a substrate material, then coat it with non-conductive material (e.g., non-conductive paint spray) and then put the second layer on top (this is how the sensor on the 3D printed bunny was realized).

Yet another alternative for ultra-rapid fabrication, if high sensing accuracy is not required, uses simple copper tape. Simply put strips of copper tape in a row/column matrix (make sure rows and columns are electrically isolated from each other) and use it for low-resolution sensing.



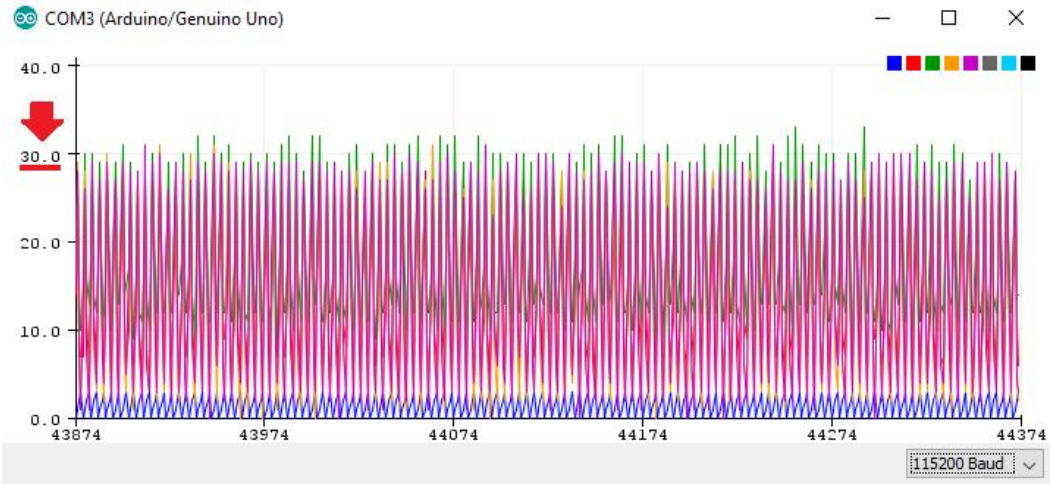
Hand-Drawn Sensor Using Silver Ink Pen

You can use a Silver Ink Pen (e.g., Circuit Scribe) to draw the electrodes on both sides of a paper sheet. You can print the pattern on the paper beforehand to get the electrodes exactly where they are supposed to be.

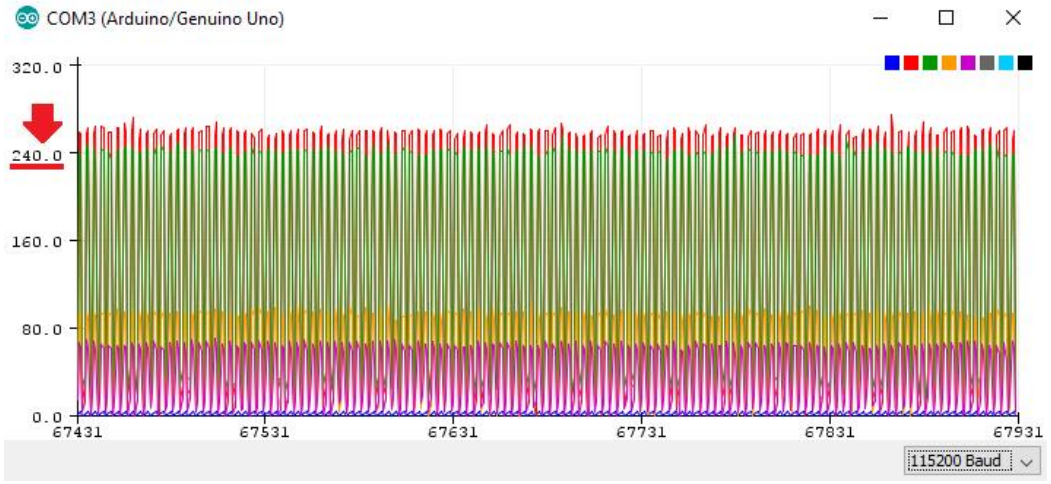


FAQ

1. *How to set the threshold to receive touch up and down events?*
   To find the right threshold, set *raw_value* = *true* (as described above). Open the Serial Plotter. Check the maximum value that you observe without touching the sensor (e.g., 30 in the plot below). Repeat the experiment, when you are touching the sensor with one finger (e.g., 240 in the second plot below). Find the difference between these two values and select a value which as around half of the difference as threshold value.



No-Touch



Touch

2. *How to find the index of serial port for Processing?*
   c. In Arduino IDE go to " Tools → Port ". Find the name of the port that your Arduino is connected to.
   d. In Processing IDE, get a list of all the available serial ports, by writing:
   println(Serial.list());
   e. Find the Arduino port among the printed ports. Use its index (starting from 0) to set "serialPort":
   int serialPort = <index>;

3. *The results are noisy?*
   Your Laptop/PC is not properly grounded, try to unplug your Laptop or try a different Computer.

4. *Too thick dielectric between the transmitter lines or on top layer?*
   The raw value will be too weak, and the blobs are not detectable.

5. *Sensor size?*
   We have tested the multi-touch sensor up to 175 x 175 mm. The optimum electrode size is 6x6 mm and 0.5 mm distance between electrodes.

6. *Sensor material?*
   Multi-Touch Kit works with highly conductive materials such as silver and copper. It does not works with less conductive material such as ITO because of the high-frequency operation of the sensor.

7. *Stylus?*
   Since the sensing approach is based on extra-body propagation of signals, it is not possible to capture input made with conductive objects.

For more information please refer to the paper and supported video[5].

[5] https://hci.cs.uni-saarland.de/multi-touch-kit/