

Multi-Touch Kit: A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing Using a Commodity Microcontroller

Narjes Pourjafarian¹, Anusha Withana², Joseph A. Paradiso³, Jürgen Steimle¹

¹Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

²School of Computer Science, The University of Sydney, Sydney, Australia

³Responsive Environments, MIT Media Lab, Cambridge, Massachusetts, United States

¹{pourjafarian, steimle}@cs.uni-saarland.de, ²anusha.withana@sydney.edu.au, ³joep@media.mit.edu

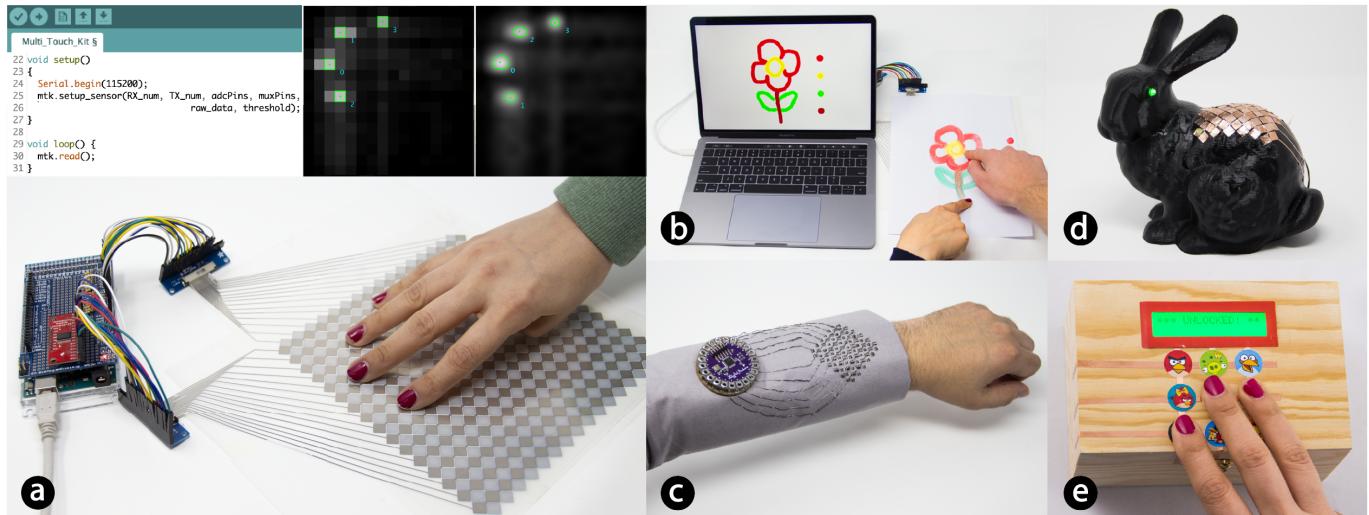


Figure 1. Multi-Touch Kit enables electronics novices to easily implement high-resolution capacitive multi-touch sensing using a commodity microcontroller (a). This supports rapid prototyping of multi-touch surfaces that are customized in dimensions, shape and materials, for applications such as paper-based interaction (b), textile multi-touch sensing with a Lilypad (c), multi-touch input on 3D printed objects (d) and everyday objects (e).

ABSTRACT

Mutual capacitance-based multi-touch sensing is now a ubiquitous and high-fidelity input technology. However, due to the complexity of electrical and signal processing requirements, it remains very challenging to create interface prototypes with custom-designed multi-touch input surfaces. In this paper, we introduce *Multi-Touch Kit*, a technique enabling electronics novices to rapidly prototype customized capacitive multi-touch sensors. In contrast to existing techniques, it works with a commodity microcontroller and open-source software and does not require any specialized hardware. Evaluation results show that our approach enables multi-touch sensors with a high spatial and temporal resolution and can accurately detect multiple simultaneous touches. A set of application examples demonstrates the versatile uses of our approach for sensors of different scales, curvature, and materials.



This work is licensed under a Creative Commons Attribution-NonCommercial International 4.0 License.

UIST '19, October 20–23, 2019, New Orleans, LA, USA.

Copyright is held by the author/owner(s).

ACM ISBN 978-1-4503-6816-2/19/10.

<http://dx.doi.org/10.1145/3332165.3347895>

CCS Concepts

- Human-centered computing → Human computer interaction (HCI);
- Hardware → Sensor devices and platforms;

Author Keywords

Multi-touch input; Capacitive touch sensing; Prototyping.

INTRODUCTION

Capacitive multi-touch is a high-fidelity input technology now common in a myriad of computing devices. The industrial standard for accurate and robust multi-touch sensing is based on mutual-capacitance sensing. Despite its popularity, prototyping interactive devices with embedded multi-touch functionality is still technically demanding. Commercial multi-touch controllers commonly require complicated firmware programming or low-level USB programming to access raw data. These controllers limit customization in terms of number and size of electrodes, available data, materials, and shapes of the sensor. Furthermore, although some multi-touch chips are inexpensive, using them in a prototype typically requires buying a costly development kit or designing a breakout board and a programmer. Therefore, using mutual-capacitance multi-touch sensing is mainly limited to industrial solutions or research labs with significant electrical engineering (EE) expertise. Consequently, this technology remains outside the

realm of typical interaction designers and makers who seek to prototype new touch-based interfaces.

In contrast, do-it-yourself electronic kits, such as Arduino, and their extensions have enabled non-experts to rapidly build functional electronic prototypes. One example is the Capacitive Sensing Library¹, which provides a simple firmware library to realize basic capacitive sensing using an Arduino without any specialized hardware. However, firmware is restricted to loading-mode sensing, a comparably simple mode of capacitive sensing not well suited to support multi-touch sensing grids. Our main objective is to contribute a solution of similar ease and simplicity that supports the considerably more complex mutual-capacitance sensing technique, while avoiding use of specialized hardware and proprietary software that thus far is required for this mode of capacitive sensing.

In this paper, we introduce *Multi-Touch Kit*, a low-cost do-it-yourself technique to enable interaction designers, makers, and electronics novices alike to rapidly create and experiment with high-resolution multi-touch sensors of custom sizes, geometries, and materials.

In contrast to existing solutions, the Multi-Touch Kit is the first technique that works with a commodity microcontroller (our implementation uses a standard Arduino) and does not require any specialized hardware. As a technical enabler, we contribute a modified multi-touch sensing scheme that leverages the human body as a transmission channel of MHz range signals through a capacitive near-field coupling mechanism. This leads to a clean signal that can be readily processed with the Arduino's built-in analog-to-digital converter, resulting in a sensing accuracy comparable to industrial multi-touch controllers. Only a standard multiplexer and resistors are required alongside the Arduino to drive and read out a touch sensor matrix.

The technique is versatile and compatible with many types of multi-touch sensor matrices, including flexible sensor films on paper or PET, sensors on textiles, and sensors on 3D printed objects. Furthermore, the technique is compatible with sensors of various scale, curvature, and electrode materials (silver, copper, conductive yarn) fabricated using conductive printing, hand-drawing with a conductive pen, cutting, or stitching.

A comprehensive firmware and software library implements the sensing scheme, enabling developers to easily read out raw capacitance images as well as tracked locations of touch points at a high frame rate.²

We present empirical evaluation results that demonstrate the technique's ability to accurately detect touch input for sensors of various sizes, materials, and curvatures down to a radius of 15 mm. To verify the practical usefulness of the technique, we used Multi-Touch Kit to implement five technical demonstrators comprising, among others, multi-touch sensors on paper-based interfaces, 3D printed objects, and textiles. These applications demonstrate the kit's ability to support high-resolution multi-touch input on sensors of up

to 175×175 mm size, on flat and curved geometries of various materials. The applications further show the Multi-Touch Kit's compatibility with different rapid prototyping techniques, ranging from low-fidelity sketching with simple copper tape or a conductive pen to high-fidelity printed sensors.

In summary, the main contributions of this paper are:

1. Multi-touch sensing using a commodity microcontroller without any special hardware, based on a modified multi-touch sensing approach utilizing frequencies in the range of tens of MHz for body channel transmission through capacitive near-field coupling.
2. A comprehensive firmware and software library for Arduino and Processing that enables electronic novices to easily control and read out multi-touch sensors with a few lines of code. The library gives real-time access to raw capacitance images and tracked touch locations.
3. Empirical results demonstrating accurate multi-touch sensing for sensors of various scales, materials, and curvatures.
4. Demonstration of practical usefulness with five implemented application examples.

RELATED WORK

Multi-Touch Kit is related to prior work on touch sensing and technical frameworks for rapid prototyping applications:

Touch Sensing

Touch is a mature input technology popularized earlier with screen-based devices [4], evolving since then into diverse application areas and scales, such as interactive spaces [7, 42], objects [32, 28] and on-body interfaces [16, 14, 27, 37, 38]. Research has also demonstrated a versatile spectrum of methods to fabricate custom touch sensors. These comprise crafting with conductive copper and gold leafs [21], silicone casting [37], inkjet printing [22], screen printing [27, 38] and 3D printing [34].

Different technologies can be used to sense touch. Optical methods such as frustrated total internal reflection (FTIR) [13] and depth cameras [14, 39] are commonly used for large-scale touch screens. Acoustic methods have also been shown in touch interactive surfaces [25] and on the body [15, 16]. Other technologies include resistive methods [17, 37], electric field sensing [43], impedance profiling [32], time-domain reflectometry [40] and electric field tomography [41].

Arguably, the most frequently used method and industry standard for sensing touch is projected capacitive sensing [11, 9, 8]. Specifically, mutual-capacitive sensing enables high-resolution sensing of multiple simultaneous touch contacts, can be embedded in small form-factor devices [11, 31] and have a low-latency [24]. Despite its popularity, prototyping multi-touch needs advanced knowledge of the underlying technology and specialized hardware. In this paper, we propose a modified approach for capacitive multi-touch sensing using only a commodity microcontroller.

¹<https://playground.arduino.cc/Main/CapacitiveSensor>

²<https://hci.cs.uni-saarland.de/multi-touch-kit/>

Prototyping Touch-Sensing Applications

With the popularity of camera-based methods for multi-touch sensing, different software frameworks were introduced to streamline prototyping and implementing touch-sensing applications. For example, camera-based systems such as ReACTIVision [20] have stimulated and enabled extensive research on multi-touch sensor surfaces. More recent work, such as the depth-camera based RoomAlive, incorporates modern methods of sensing into accessible toolkits [19].

Similarly, capacitive sensing toolkits had similar effects for capacitive touch sensing. The *CapSense library*³ is an Arduino library for loading mode capacitive touch sensing and does not require special hardware or advanced knowledge. However, it is limited by the inherent drawbacks of loading mode sensing and cannot deliver high resolution or multi-touch sensing without complex instrumentation. The *OpenCapSense* toolkit [9] is more powerful and supports several forms of capacitive sensing, including mutual capacitance sensing. However, it is designed for hover and gesture recognition, lacks the capabilities to support high-resolution touch sensor surfaces, and requires a proprietary hardware controller board. In contrast our approach works with a commodity microcontroller.

In addition to hardware-related approaches, software frameworks are available to process and classify multi-touch input for interaction. An overview is provided in [6]. Our firmware and software libraries take inspiration from this prior work and propose the first solution for rapid prototyping of capacitance-based sensing for high-resolution multi-touch input.

Sensing Modes and Commercial Touch Controllers

The simplest capacitive sensing type is loading mode. It can be easily realized by electronics novices using the Arduino CapSense library and easily adapted to custom designs. Several commercial controllers are available for loading mode sensing, such as MPR121, Microchip MTCH6102, and Analog Devices AD7142 [12]. However this technique is low resolution and restricted to single touch.

In contrast, most multi-touch sensing approaches require complex hardware. For instance, shunt mode (mutual-capacitance (mCap)) sensors measure the change of capacitance between two intersecting conductors caused by the proximity of an external conductive element such as human touch. To measure the change in capacitance, mCap controllers transmit an AC signal through one electrode (TX electrode) and observe the received AC signal at the other electrode (RX electrode). Touch sensing surfaces are realized by organizing these TX and RX electrodes into a row-column matrix. However, in these sensors, the change of mutual capacitance between touched and not touched states is typically much smaller than the stray capacitance [5]. Therefore, sophisticated measuring methods (Capacitance-to-Digital Conversion (CDC), Sigma-Delta Modulation, Successive Approximation with Single-Slope ADC [35]) are required to identify touch input with a sufficient signal-to-noise ratio. These methods require complex analog circuitry and cannot be implemented on commodity

microcontrollers such as an Arduino without specialized hardware.

Dedicated controller chips that implement these sophisticated measuring methods are commercially available. For instance, canonical examples include Microchip MTCH6301 and the Texas Instruments MSP430FR2xx family. While offering a viable path for electronics experts, these chips are clearly too complicated to program, adapt and interface with for electronics novices, typical interaction designers or makers. Other operating modes are transmit and receive modes. In these sensing modes, the coupling between the body and one of the electrodes is greater than the coupling between the body and the other electrode or between the transmitter and the receiver [12]. The same hardware as shunt mode is also used for transmit and receive modes [12].

In conclusion, commercial chips are not designed to support rapid iterations and customization of options. They usually come with specific instructions on the sensor designs (e.g., exact dimensions of electrodes). Specifically when designing custom sensors, the underlying signal models get affected in ways that require new circuit designs to accurately sense touch. We overcome these challenges with our flexible, do-it-yourself multi-touch scheme.

THE MULTI-TOUCH KIT

We introduce a sensing scheme that makes it possible to sense multi-touch on a touch sensing matrix using a commodity microcontroller without any specialized hardware. We then present the do-it-yourself hardware implementation and the Arduino and Processing libraries. Together, they enable novices to rapidly prototype custom multi-touch sensors and to access raw capacitance data or high-level multi-touch coordinates using a few lines of code.

Sensing Approach

In *Multi-Touch Kit*, we propose a modified multi-touch sensing approach that utilizes the extra-body transmission through electric field. Specifically, we leverage the fact that in the frequency ranges from 100 kHz to 40 MHz the electric field around the body behaves as a quasi-static near-field [36, 3]. In our sensing scheme, we use *projected capacitive sensing* [8] with a modified *transmit+receive mode* [29, 11]: a TX electrode transmits a signal in the MHz range; in this frequency range, the quasi-static electric field allows for strong capacitive coupling between the TX electrode, the finger, and the RX electrode [36]. Simply put, the finger can be considered a conductor that couples both electrodes [3]. Since the propagation of electrical signals in the selected frequency range (< 40 MHz) is better along the human body than through the air, a touch event yields an increment in amplitude of the received signal. This increment is significant enough to be captured by a commodity microcontroller, with a sufficient SNR for robust touch sensing. Note that this is contrasting to decrements observed in classical *shunt-mode* mCap approaches [11].

To leverage this basic principle in a touch sensor implementation, we address three aspects: (1) Investigating the *frequency response* of the touch system to select a suitable frequency, (2)

³<https://playground.arduino.cc/Main/CapacitiveSensor>

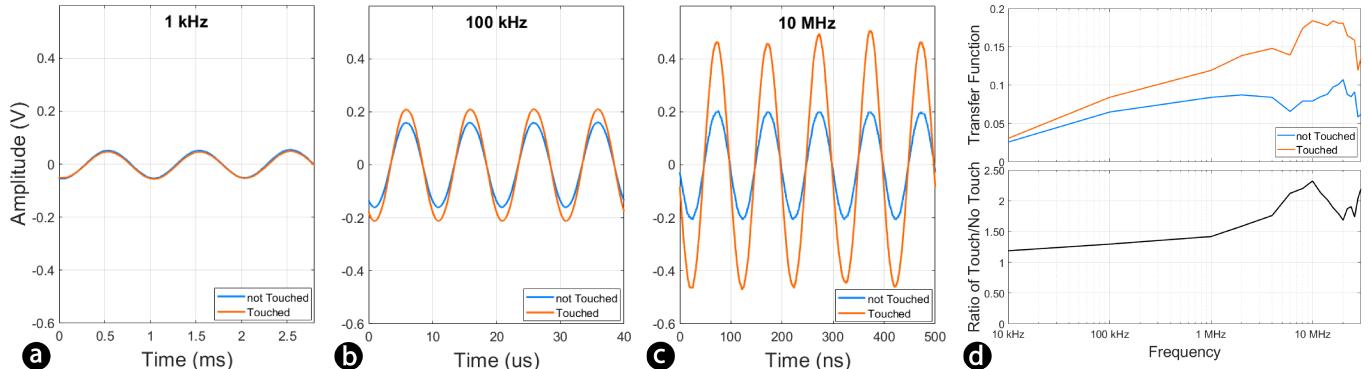


Figure 2. (a-b) At lower frequencies, the receive signal does not show considerable change in amplitude between touched and not touched states. (c) At higher frequencies (10MHz), the extra-body propagation causes a significant increase in the receive signal during touch. (d) Frequency response of the matrix multi-touch sensor for touched and not touched states. The ratio between touched and not touched states is largest at 10 MHz.

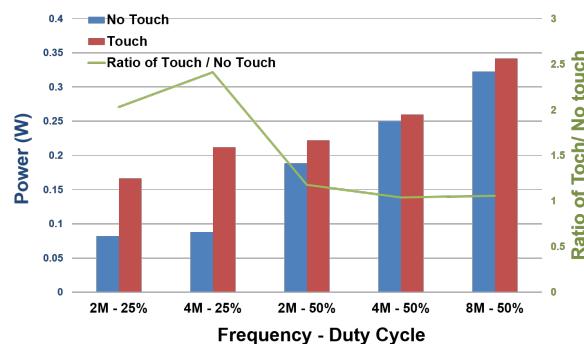


Figure 3. Total energy of receive signals in different frequencies and duty cycles. The greatest difference between the energy of touch and no touch signals belongs to the 4 MHz frequency with 25% duty cycle, which makes it a suitable choice for touch detection.

generating the effective frequency band as a transmit signal with a commodity microcontroller, and (3) accurately capturing the changes in the receive signal at touch events using its built-in analog-to-digital converter (ADC). We will now discuss each of these aspects and confirm their validity.

Frequency Response of the Touch Sensor

To systematically investigate extra-body transmission in the context of matrix-type touch sensing, we conducted an empirical study to derive the frequency response of the touch system in the frequency band of interest (< 40MHz). In this experiment, sinusoidal signals with frequencies ranging from 1 kHz to 30 MHz⁴ were produced with a function generator (Keysight 33600A) and used as the transmit signal to an 8 × 8 multi-touch sensor (6 × 6 mm diamond size and 0.5 mm distance between electrodes [26]). Received signals were measured with an oscilloscope (PICOSCOPE 6402A) under two conditions: (1) when a finger is touching the sensor intersection (Touched) and (2) when there is no touch (Not Touched). To illustrate the strong effect of frequency, Figure 2-a, b and c shows the transmit and receive signals for the two conditions at frequencies of 1 kHz, 100 kHz, and 10 MHz respectively.

The amplitude of the frequency response of the touch sensor is formulated by calculating the gain or the input to output

⁴1 kHz, 10 kHz, 100 kHz, 1 MHz, and 2...30 MHz in 2 MHz intervals, all 5 Vpp

ratio (i.e., receive signal / transmit signal) for each sinusoidal transmit signal. Figure 2-d shows the frequency response of the touch sensor at conditions Touched and Not Touched for frequencies between 10 kHz and 30 MHz. An additional plot of the ratio between Touched to Not Touched is added since this is indicative of the signal-to-noise ratio (SNR): higher the ratio, higher the expected SNR. Therefore, the peak of the ratio curve helps us to identify the optimal frequencies for the touch sensor. As shown in Figure 2-d, the difference peaks at 10 MHz implying that a transmit frequency centered around 10 MHz is the optimal choice.

Generate the Effective Transmit Signal

Generating a sinusoidal signal with a commodity microcontroller is difficult since most models do not feature a digital-to-analog converter (DAC) and hence are limited to digital output. With the fixed clock frequencies in these devices, generating a specific frequency corresponding to the peak response of the body is even more challenging. As a solution to this problem, we propose to use carefully selected patterns of periodic digital signals to generate outputs with concentrated spectral power at the target frequency band. Most commodity microcontrollers are capable of creating a wide range of pulse width modulation (PWM) signals, which are digital square waves with different duty cycles. Using Fourier analysis, these PWM signals can be represented as a collection of sinusoidal signals (harmonics) spread across a wide bandwidth. Our approach is to identify a PWM signal (both frequency and duty cycle) that has high amplitude harmonics in the peak areas of the frequency response of our multi-touch sensor.

To identify the suitable frequency and duty cycle for a PWM signal with harmonics in the optimum frequency band, we conducted an empirical study by recording the received signal (using the oscilloscope) for a set of transmit PWM signals. As inputs, we selected 2MHz, 4MHz and 8MHz PWMs with 25% and 50% duty cycles. We selected these configurations to represent the available PWM frequencies in Arduinos and to spread the power in both odd (25%) and even harmonics (50%).⁵ The output power is calculated based on Parsevals Theorem by (1) deriving the Fourier series of each received

⁵8MHz-25% was not used since Arduino Uno and Mega cannot generate this signal.

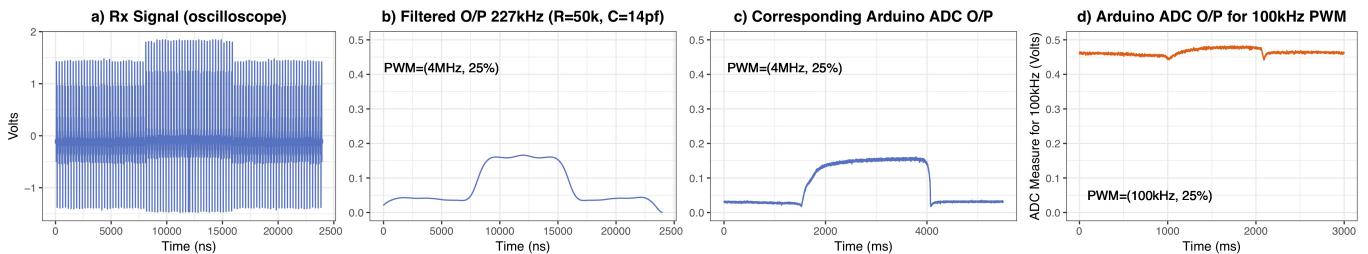


Figure 4. (a) Receive (RX) signal of the sensor showing no-touch to touch and back to no-touch transition (Transmit signal 4MHz, 25% duty cycle); (b) RX signal filtered with Arduino internal ADC RC configuration (average $R = 50\text{k}\Omega$, $C = 14\text{pF}$); (c) Output of the Arduino's ADC; (d) In contrast, a low frequency (100KHz, 25% square wave) transmit signal does not allow for robust touch sensing using the ADC.

signal to represent it as a summation of individual sinusoidal harmonics, (2) calculating the sum of squares of each coefficient divided by two, and (3) then adding the square of the DC component. We derived the output power for each input PWM configuration for the two different conditions (touched and not-touched) as shown in Figure 3. Results show that the 4MHz signal with 25% duty cycle outperforms the other PWM configurations. For illustration, Figure 5 shows the prominent harmonics of this PWM signal. They are close to our target transmit frequency of 10 MHz.

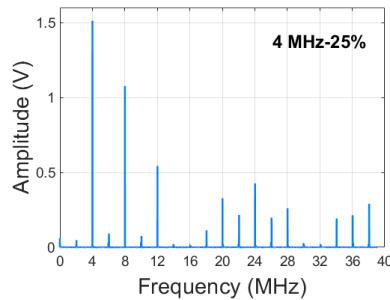


Figure 5. The prominent harmonics of 4MHz, 25% duty cycle are close to 10 MHz.

Analyzing Receive Signal

Most commodity microcontrollers lack the capability to accurately sample a signal of this high frequency. For instance, the highest sampling frequency with accurate ADC conversion for ATmega328P based microcontrollers (most Arduinos) is limited to 1MHz.

To overcome this challenge, we leverage the fact that human touch-down and touch-up events occur at a much lower frequency than the actual PWM frequency of TX and RX signals. This results in an amplitude modulated signal where the increase in amplitude due to touch contact envelopes the PWM signal as shown in Figure 4-a (captured with Picoscope, USB oscilloscope). Amplitude modulated signals can be easily recovered using a simple low-pass filter (LPF).

The internal architecture of the Arduino analog-to-digital (ADC) converter implements a low-pass filter. The ADC utilizes a sample-and-hold capacitor (14pF). This capacitor, along with the path resistance (ranging from $1\text{k}\Omega$ to $100\text{k}\Omega$ ⁶), creates an internal low-pass filter (LPF). The cut-off frequency f_c of this filter is well below the PWM frequency. Assuming a resistance value in the center of the range specified in

⁶ATmega328P data sheet Figure 23-8

the data sheet ($R = 50\text{k}\Omega$), with $C = 14\text{pF}$, the cut-off frequency is $f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 50 \times 10^3 \times 14 \times 10^{-12}} = 227.4\text{kHz}$. This LPF filters the high-frequency components of the received signal, leaving the attenuated low-frequency touch signal to be converted as the ADC values.

To demonstrate the effect of this low-pass filter, we modeled it with R's *Signals* package and applied it to the captured raw received signal. Figure 4-b shows the signal after applying the low-pass filter. It shows that the filtered signal accurately represents the touch state. Figure 4-c shows the microcontroller's ADC output (converted to Volts) for the transition from no-touch to touch to no-touch. It shows the values correspond to the low-pass filtered signal. For comparison, we also captured the ADC output with a lower frequency PWM signal (100kHz, 25% square wave). As shown in Figure 4-d, this results in a poor SNR, demonstrating the greatly superior performance of the high-frequency signal.

Hardware Implementation

Considering its popularity among the HCI and maker communities, we selected the Arduino platform as our foundation hardware unit. *Multi-Touch Kit* limits the use of external hardware to a commonly available simple multiplexer and standard resistors. It is compatible with a variety of multi-touch sensor matrices to support versatile prototyping.

Hardware Components and Interconnection

The *Multi-Touch Kit* hardware schematic is shown in Figure 6. We tested setups with Arduino Uno (ATmega 328P), Arduino Mega (ATmega 2560), and Arduino LilyPad (ATmega 328P), all very popular microcontrollers. Arduino's hardware *Timer 2*, the internal crystal oscillator of the controller, is used as the clock generator to generate a 4 MHz square wave with 25% duty cycle of 5 Vpp magnitude via pulse width modulation (PWM). Since the high-frequency PWM signal is limited to a few pins, we use a standard multiplexer (CD74HC4067, < \$1) to drive multiple transmitter lines. This multiplexer is a general-purpose component and users can freely choose their own multiplexer since it has bandwidth to work in the functional frequency range. The receiver terminals are connected to $100\text{k}\Omega$ load resistors. Voltage across load resistors is measured using the analog input pins of the Arduino. As shown in Figure 7-a, the complete setup can be easily implemented on a breadboard, even for sensor matrices of considerable size (16 × 16). Alternatively, a more compact and physically more robust setup can be realized with an Arduino Proto Shield,

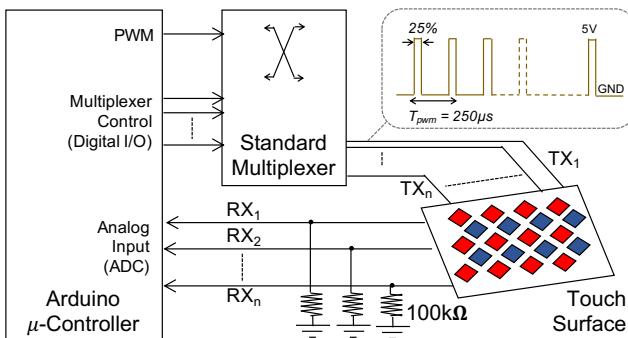


Figure 6. Multi-Touch Kit hardware schematic and interconnection

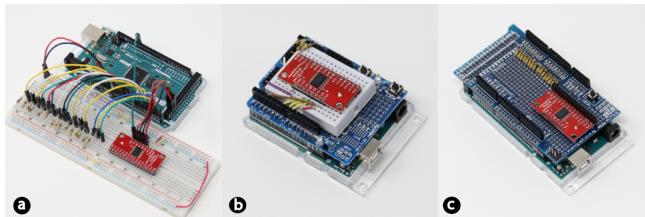


Figure 7. The simple hardware setup can be implemented on (a) a breadboard, (b) using an Arduino Proto Shield without soldering, and (c) with soldered connections.

either using an embedded mini breadboard for solderless operation (Figure 7-b), or by soldering components (Fig. 7-c). It is also possible to realize the full setup using an Arduino LilyPad with a Lilypad prototype board, connecting the Lilypad setup to the sensor with snap-in connections for ease of use in textile applications (Figure 1-c).

Fabrication of Multi-Touch Sensor Matrix

Multi-Touch Kit is compatible with established rapid prototyping techniques for fabricating sensor matrices that have been presented in the literature. These include *conductive inkjet printing* on a desktop printer [23], as demonstrated in [27], or cutting *copper foil* using a commercial vinyl cutter, as used in [33]. Sensor designs can also be hand-drawn with a *conductive pen* (Circuit Scribe), or stitched on fabric with *conductive thread* (Adafruit Stainless Thin Conductive Thread). For very rapid, low-resolution designs, it is even possible to manually apply strips of copper tape in rows and columns, as we will demonstrate in the application section. Sensors can be curved down to a radius of 15 mm.

We recommend using the classical two-layered diamond pattern that is commonly used for mutual-capacitive touch sensing [7, 26], with electrode dimensions ranging between 4 × 4 mm – 6 × 6 mm and an inter-electrode spacing of 1 mm. Our library includes reference vector graphic designs that can be directly printed or cut.

Software Implementation

We provide an Arduino firmware library and a Processing library. It offers an API for easy access to low-level raw capacitance values and high-level touch coordinates, while hiding the low-level logic of our sensing approach from the application developer.

Arduino Library

The Arduino library internally sets the responsible registers to configure the relevant frequency and duty cycle of the PWM signal. It further sets the reference voltage for the analog-to-digital converter and controls time-division multiplexed scanning of the sensor matrix internally. The library reports raw capacitance values. Alternatively, for rapid prototyping, it can report binary touch up/down states based on simple thresholding (more advanced touch blob analysis and tracking is offered in the Processing library).

Only two functions are required to be called in an Arduino program to interface with the sensor:

`setup_sensor()`: This function needs to be called only once in the `setup()` function of the Arduino program. It accepts the following arguments: sensor dimensions (the number of TX and RX lines), an array with numbers of analog-in ports connected to RX pins, an array with numbers of digital I/O pins connected to control the multiplexer, a Boolean variable defining whether raw capacitance data or touch up/down states shall be reported, and a threshold for touch down state.

`read()`: This function returns a two-dimensional array of 10 bit raw capacitance readings or binary touch up/down states corresponding to each row-column intersection. Each function call completes a full scan of the touch sensor.

With an Arduino Mega and for sensors with dimensions 4 × 4, 8 × 8, and 12 × 12, the read function on average took 1.85ms ($SD = 0.45$), 7.2ms ($SD = 0.46$) and 16.38ms ($SD = 0.49$) respectively to complete. Therefore, with a 12 × 12 sensor, the highest achievable frame rate is 60fps.

Processing Library

To convert the raw capacitance values returned by the Arduino library into high-level multi-touch information, data need to be (1) calibrated and scaled, (2) interpolated and merged and (3) blobs extracted and touch points tracked over consecutive frames. To streamline the process, we have created a software library for the frequently used open-source prototyping platform Processing⁷. It parses raw touch data sent from the Arduino through the serial port for further processing.

Calibration and Scaling

Raw data of mutual-capacitance sensors are affected by several internal and external factors of the sensor design. For instance, previous research shows the intensity of raw values varies with distance from the connecting edge of a sensor matrix [27]. Additionally, in custom designs, the custom size, length, shape and materials used for the electrodes may also affect the homogeneity of the raw values. Therefore, raw values are first calibrated and normalized.

The calibration process is done once per sensor per user, and the results are saved. It consists of two steps. First, we remove the offset values of individual cross-section characteristics by subtracting the average noise floor. Then, the developer is asked to touch a random place on the sensor surface. The reported values are saved and used to normalize and pre-scale the data before interpolation. This process can be automated.

⁷<https://processing.org/>

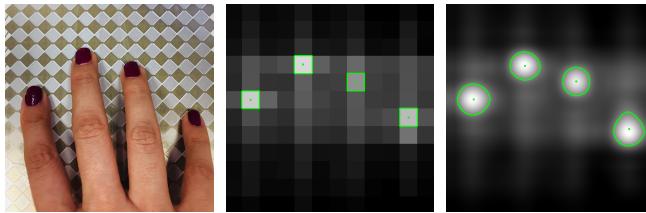


Figure 8. Detecting multi-touch input on a 12x12 sensor: (a) four fingers touching the sensor (b) raw capacitive image, (c) interpolated image with blob detection.

Interpolation and Merging

Data from the sensor are organized in arrays of integers representing each TX and RX intersection. These arrays are merged into a 2D image creating a capacitive image of the sensing surface. However, because of the limited number of electrodes, this image has a low resolution. Similar to previous research in capacitive sensing, we use bi-cubic interpolation to create an up-scaled imaged suitable for blob detection and finger tracking using image processing techniques [27].

Touch Detection and Tracking

Finally, to identify the touch locations on the surface of the sensor, we use the OpenCV⁸ library for the blob extractions from the capacitive image. The centers of these blobs are detected as touch points (Figure 8). Each touch point is labeled with a unique ID and tracked over consecutive frames.

EVALUATION

In order to verify the functionality of the *Multi-Touch Kit* as a prototyping platform, we conducted a series of technical studies. These evaluate the signal-to-noise ratio (SNR) of multi-touch input on sensors of various scale, the accuracy of touch location, the effect of curved geometries, and the effect of different electrode materials and fabrication approaches.

For all of our experiments, we used a fully mobile, battery-powered setup (similar to Figure 1a, except the data sent via Bluetooth to a PC), as this creates the most challenging grounding condition for capacitive touch sensing [10]. We used an Arduino Mega microcontroller, which sent the raw data to a PC via a Bluetooth connection. Except for evaluating different materials, all sensor samples were printed on transparent PET film using a Canon IP100 desktop inkjet printer and conductive silver ink [22]. The TX and RX electrode layers are printed on separate PET films and then attached together with a very thin layer of adhesive film. The top surface of the sensor is insulated with a thin layer of transparent dielectric. During the experiments, the sensors were placed on the surface of a wooden table.

Signal-to-Noise Ratio and Scalability

Different applications demand customized multi-touch surfaces with various sizes. The most important factor to support such customization is the sensor's ability to scale, while offering a sufficiently high signal-to-noise ratio for robust touch sensing. We conducted a pilot study to identify the most demanding test conditions and then evaluated touch input in these conditions with 10 participants.

⁸<https://opencv.org/>

Pilot Study

To identify the most demanding touch conditions for the main study, we conducted a pilot study. We used a sensor with 12×12 electrodes $6 \times 6\text{ mm}$ diamond shape [7, 26]. We followed a factorial design with four locations on the touch surface and four multi-touch cases to test. Corners were selected to represent the most challenging locations compared to the connecting edges [27]. For each corner, we tested four touch conditions: single-touch, simultaneously touching with a second finger on the same TX line, touching with a second finger on the same RX line, and touching with three fingers (on corner, TX and RX lines). In case of multi-touch input, secondary and tertiary fingers were positioned on the respective TX or RX line at the position closest to the corner while still being detected as its own touch point. We had previously identified that this is the most demanding multi-touch condition in terms of signal-to-noise ratio. As the dependent variable, we calculated the signal-to-noise ratio (SNR) of the touch input, which is the most commonly used measure to evaluate the quality of touch sensing [5]. For each condition, 5 iterations of SNR values were recorded.

The results revealed that touching the closest corner to the connecting edge for both RX and TX lines had the lowest SNR (57 , $SD = 20.0$). Furthermore, it became apparent that additional multi-touch contacts reduced the SNR compared to single-touch sensing. We further tested these touch conditions under various grounding conditions of the user (sitting with legs resting on the floor, sitting with lifting the foot, and standing on the floor) and changing the sensor position (put on table, handheld, or put on the arm with isolation layer between the sensor and the skin). These conditions did not considerably affect the SNR. The highest change we observed was 3%. In light of the high SNRs identified in our experiments, this effect is negligible.

User Study

We conducted a controlled experiment with users to more formally investigate the signal-to-noise ratio of our sensing approach and to account for the effect of body capacitance, which is known to vary across users. We recruited 10 voluntary participants (4 female, mean age 35).

We selected the most demanding conditions identified in the pilot: the sensor with 12×12 electrodes and $6 \times 6\text{ mm}$ electrode dimensions. On this sensor, we chose the location that had performed the least well in the pilot study: the corner closest to TX and RX connectors. By showing a sufficiently high SNR in this most demanding case, we will be able to show the overall robustness of the sensing approach. The sensor was placed on a table. The participant was standing.

As conditions, the study had four different touch locations, which are indicated in Figure 9-a. They comprised the most demanding single-touch, dual-touch, and triple-touch locations we had identified in the pilot. In each condition, the participant was asked to touch the sensor consecutively five times with one, two, and three fingers at the respective positions that were visually marked on the sensor, for one second with a one-second pause in-between touching. Raw capacitance data were sent to a PC through a Bluetooth connection and logged

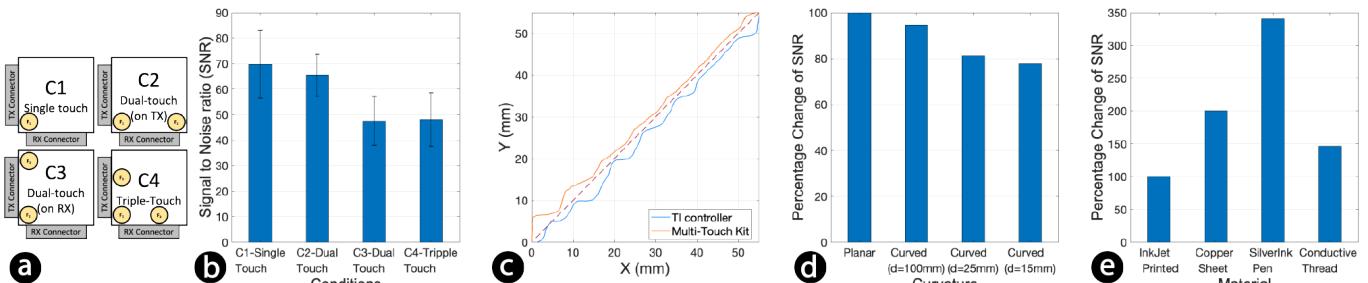


Figure 9. Evaluation results: (a) Locations tested in user study represent the most demanding locations, (b) Signal-to-noise ratio for touch input with 1, 2 and 3 fingers on a sensor with 12x12 electrodes, (c) Spatial accuracy of touch input; (d) Effect of sensor curvature on SNR; (e) Effect of electrode material on SNR.

for analysis. Figure 9-b shows the average signal-to-noise ratio for each condition and all participants. The results show that all values are well above the critical value of 15, which is required for robust touch sensing at industrial strength [5].

Spatial Accuracy

To measure the spatial accuracy of touch sensing and to compare it with the baseline of an industry-strength commercial touch controller, we recorded finger movement on a sensor and compared the interpolation results with ground truth. We selected the Texas Instruments TI MSP-CAPT-FR2633 touch controller chip for the baseline comparison. Since this controller supports a maximum of 16 I/O pins, we used a 8×8 multi-touch sensor of $55 \times 55 \text{ mm}$ size for this experiment. Following the method presented in [2], we visually marked the main diagonal axis of the sensor starting from the electrode farthest from the signal driving lines. The diagonal axis was selected, as it is to be expected that the accuracy of interpolation is lowest because of the larger distance between electrode intersections. The finger was dragged diagonally through the sensor along the marked line. This was repeated 5 times with the sensor connected to the Multi-Touch Kit and 5 times with the sensor connected to the commercial touch controller. The resulting raw data were recorded and used for interpolation and calculation of the touch locations.

The Root Mean Square Error (RMSE) of each trial was calculated. The average RMSE for Multi-touch Kit is 1.56 mm ($SD=0.17$), and for the TI controller 1.94 mm ($SD=0.20$). The results show that our toolkit has a spatial accuracy comparable to the commercial touch controller. For qualitative visual inspection, the results of the trial with the highest RMSE in either condition are depicted in Figure 9-c. The plot shows that the sensed locations of Multi-Touch Kit closely match with the ground-truth. The maximum offset is less than 3.90 mm , which is close to the natural imprecision of human touch [18].

Curvature

To evaluate the effect of curved sensor geometries, we conducted a technical evaluation with four conditions: planar and 3 curved geometries with a diameter of 100 mm , 25 mm and 15 mm each. The larger diameter reflects the typical curvature of everyday objects such as mugs, while the smallest one reflects objects such as markers or pens.

The experiment was run with a 4×4 electrodes, $30 \times 30 \text{ mm}$ sensor. The small dimension was chosen to be able to wrap

the sensor around surfaces of very small diameter. Touched and not touched events (1 s interval, 5 trials) were captured at the most demanding intersection (closest to the transition lines) and the SNR was calculated. Figure 9-d presents the percentage change of SNR with respect to the planar condition. As expected, the planar condition has the highest SNR, since the finger has maximum contact with the sensor surface. The SNR of the most curved condition was 22% lower. Considering the very high (well above 40) that we have identified above for the most demanding touch conditions (Figure 9-d), it is apparent that even a considerably larger reduction would still ensure SNR values above 15 (the required value for robust touch sensing). This demonstrates that the sensing approach is robust for curved geometries.

Materials

Finally, we investigated the effect of using different materials and fabrication methods for the physical sensor matrix: inkjet-printed with conductive silver ink, hand-painted with a silver pen, stitched with conductive yarn, and cut out of copper sheet. For each condition, we fabricated a sensor with 6×6 electrodes and $45 \times 45 \text{ mm}$ dimension. As the insulation layer, we used transparent PET film ($\sim 70 \mu\text{m}$ thickness), standard A4 office paper (80 g/m^2), embroidery fabric (Muslin, thread count of 150), and overhead PET film ($\sim 100 \mu\text{m}$ thickness), respectively.

All sensors were placed on a wooden table. Each sensor was touched 5 times with the index finger (1 s touched, 1 s released) at the most demanding location (closest intersection to the driving lines).

Figure 9-e depicts the percentage change of SNR with respect to the inkjet-printed sensor. The results show that the SNR further increases for other materials. Copper is more conductive than silver-printed electrodes. While the sheet resistance of electrodes hand-drawn on paper with a conductive pen is higher than of inkjet-printed electrodes, using paper results in a thinner dielectric layer. This fully compensates for the loss in conductivity. The textile solution with conductive thread, in turn, benefits from having the transmitter and receiver electrodes on the same side of the textile substrate. Overall, these results confirm the compatibility of our sensing approach with different materials and their fabrication techniques.

EXAMPLE APPLICATIONS

To demonstrate the functionality and versatility of the Multi-Touch Kit, we created five applications. They show use of

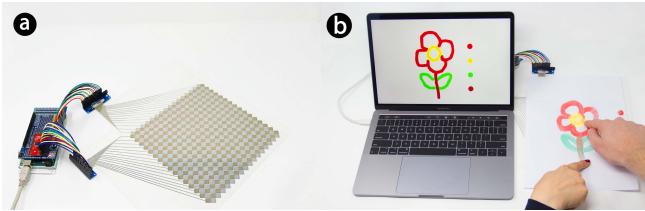


Figure 10. High-resolution interactive surface: (a) 16x16 multi-touch sensor is connected to an Arduino Mega, (b) digital capture of finger painting is visualized on a laptop.

the toolkit with different materials, substrates, scales, geometries, and fabrication methods. They range from very fast physical prototyping with copper tape to high-fidelity and high-resolution sensors that are printed, hand-drawn or embroidered.

High-Resolution Interactive Surface

To turn surfaces in the physical environment into a high-resolution input surface, we designed a customized multi-touch sensor with a 16×16 electrode matrix of $177 \times 177\text{ mm}$ size. Electrodes were printed on a desktop inkjet printer (Canon PIXMA iP100) using silver ink (Mitsubishi NBSIJ-MU01). The sensor was tethered to an Arduino Mega with an extension board containing the multiplexer (Figure 10-a). The sensor supports multi-touch input of up to 10 fingers and can be used for various high resolution and multi-touch scenarios.

As one example, we implemented an interactive finger painting application. The application uses the Arduino and Processing library to directly retrieve tracked touch coordinates. A sheet of office paper is placed on top of the sensor. The user can then create a colorful physical drawing using colors and drawing with one or multiple fingers simultaneously. A high-resolution digital copy of the painting is captured by the touch sensor and visualized in a viewer application that runs on a laptop (Figure 10-b). The color of digitally captured strokes can be set in the application. To draw a new painting, the user only needs to replace the paper while keeping the sensor sheet.

Textile Multi-Touch Sensor with Conductive Yarn

To demonstrate that Multi-touch Kit supports sensors on various materials and substrates, we created a textile multi-touch sensor (Figure 11-a). It contains a 6×6 grid of diamond-shaped electrodes that were stitched on a textile using conductive yarn (Adafruit Stainless Thin Conductive Thread). While a programmable sewing machine could have been used for this purpose, we opted for stitching by hand to confirm the functionality even for the less accurate manual fabrication approach. After stitching, we used coating spray (Kontakt Chemie 74313-AA) to isolate the transmitter line.

For textile compatibility, we used an Arduino Lilypad and a Lilypad prototype board containing the multiplexer, Bluetooth and battery. The Lilypad was connected to the textile sensor with snaps, which makes it easy to attach and detach from the garment. The setup was connected via Bluetooth to a smartphone that recognizes simple gestures.

Inspired by [30], we embedded our sensor on the sleeve of a shirt to offer direct interaction with a mobile device while the

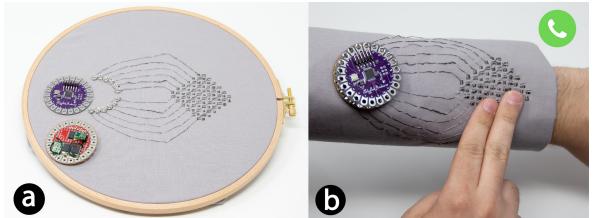


Figure 11. Textile multi-touch sensor with conductive yarn: (a) Sensor setup with Arduino Lilypad and Proto Board. (b) Tapping the sensor with two fingers accepts a phone call.

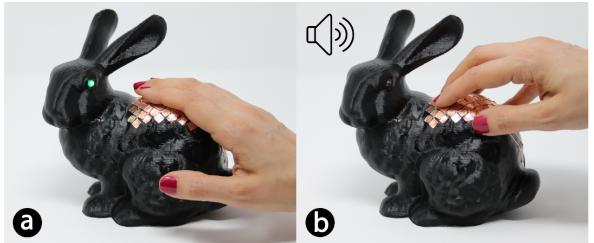


Figure 12. Multi-Touch sensor on 3D printed object: (a) cuddling the rabbit's back lights up the rabbit's eyes, (b) poking causes a purring sound.

user is on the go (Figure 11-b). Swiping with three fingers to the right or left is mapped to switching between music tracks; tapping with two fingers accepts an incoming call, while covering the sensor with the full hand rejects the call.

Multi-Touch Sensor on 3D Printed Object

Our technique is compatible with curved multi-touch sensors on 3D printed objects. We 3D printed a Stanford bunny on an FDM printer (Ultimaker S5) and turned it into a multi-touch sensitive interactive object (Figures 12 & 1-d). We covered the rabbit's curved back with a 6×6 multi-touch sensor. The sensor was made from copper sheet cut with a vinyl cutter. The transmitter and receiver layers were mutually isolated with a transparent, acrylic coating spray (Kontakt Chemie 74313-AA). We added LEDs to the rabbit's eyes and a speaker to the body. When the user cuddles the rabbit's back with the full hand, the rabbit's eyes light up. When poking it with a finger, the rabbit makes a purring sound (Figure 12-a,b).

Interactive Greeting Card with Hand-Drawn Sensor

To demonstrate that our kit works with sensors that are hand-fabricated using a conductive pen, we realized an interactive greeting card that can play music and is controlled using touch input. The greeting card contains a color image that was printed using a color desktop printer (Figure 13-a,b). We used a conductive pen (Circuit Scribe) to draw a 4×4 multi-touch sensor pattern alongside conductive lines to connect the sensor and surface-mount LEDs that we attached onto the card (Figure 13-c). The card was connected to an Arduino Uno with a Bulldog clip containing wires (Figure 13-d). We mapped swiping right or left to "turn on and off the LEDs" and tapping with two fingers to "turn on and off the music".

Rapid Prototyping with Copper Tape

The early phases of a design process commonly involve quickly exploring a large number of design alternatives at a low fidelity. Implementation time is critical here, as it would

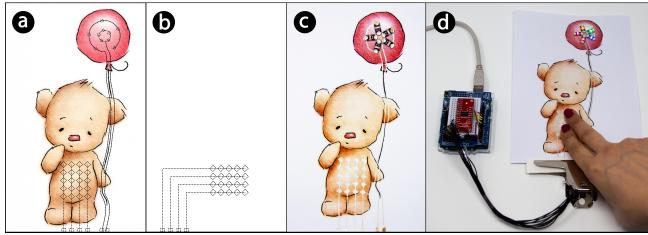


Figure 13. Interactive greeting card: (a) Front side of the greeting card with dot patterns highlighting the sensor area, (b) backside of the card, (c) fabricated card contains hand-drawn multi-touch sensor and LEDs, (d) the sensor is connected to an Arduino Uno with a clip.

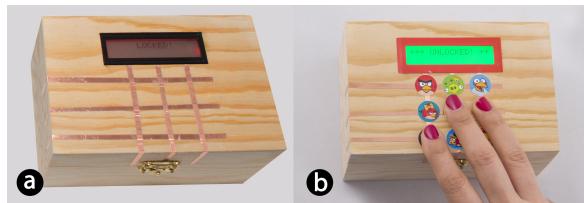


Figure 14. Wooden treasure box: (a) Rapid prototyping with copper tape on a wooden box, (b) touching three different images unlocks the box.

be prohibitive to invest significant time for every design alternative. We demonstrate use of the Multi-Touch Kit for very rapid prototyping with a handmade low-fidelity sensor made of simple copper tape. The sensor is fabricated within a minute by applying strips of copper tape to the desired input location to form a matrix of rows and columns, and then connected to an Arduino Uno, which is placed inside the box (Figure 14-a). While the effective sensing resolution is certainly lower than in our other demonstrators, as the strips do not form a dense diamond pattern, it can be sufficient in many cases of low-fidelity prototyping. For this application, we used the touch event detection of the Arduino library.

We demonstrate this with an interactive wooden treasure box (Figure 14-b). The box can be unlocked by simultaneously touching a secret combination of locations on the sensor using multiple fingers. We attached 6 strips of copper tape on the lid of the box to create a 3×3 touch sensing matrix. To help the user remember the correct locations, we added different graphical icons on top of each intersection. The box is unlocked only if the correct combination of three images is touched.

LIMITATIONS AND FUTURE WORK

Results from the technical evaluation and the successful implementation of the applications show that *Multi-Touch Kit* can accurately detect multi-touch input with sensor matrices of different scale, curvature, and materials. We experimentally validated its functionality for sensors up to 12×12 electrodes; anecdotally we can confirm its functionality for 16×16 electrodes, as we have used this larger size for the high-resolution interactive surface used in the first example application. Visual evidence of the sensing accuracy of this large sensor size is provided in the companion video. However, compared to commercial multi-touch controllers, our rapid prototyping approach is subject to several limitations:

Since the sensing approach is based on extra-body propagation of signals, it is not possible to capture input made with

conductive objects. Furthermore, due to high frequency signals, the approach is less well suited for sensors made of high-resistance materials, such as ITO. We also observed that properties of the dielectric materials used on top and between sensing electrodes have a significant effect on the SNR. We observed that thicker top insulators (more than $400 \mu\text{m}$) will render the interface unusable. This also implies that hover state is not captured by the sensor.

We never experienced any issues of RF interference during development and use of our sensors. We have further tested the interference of our sensor on nearby devices with an AM/FM radio and could not detect any noise. To test the operation of the sensor when integrated into other electronics, the sensor was placed close to an active LCD display and main power cable. We did not observe any effect on the sensor reading, nor on the operation of the other electronic devices. Our approach is compliant with the FCC regulations on equipment authorization of home-built radio frequency devices [1].

Our current prototypes are implemented with the Atmel megaAVR family of micro-controllers. Due to variations in the internal ADC and PWM implementations, other micro-controllers may have different responses. In future work, we plan to fabricate and test larger sensors and extend the hardware support for our open-source library by including other frequently used commodity platforms such as Teensy and Raspberry PI.

CONCLUSION

we have presented a technique for do-it-yourself prototyping of capacitive multi-touch sensors. By using the improved extra-body propagation of electrical signals at higher frequencies, we were able to demonstrate the possibility of implementing capacitive multi-touch sensing on commodity microcontrollers without adding any specialized hardware. Together with the Arduino firmware and Processing libraries we presented, this makes it fast and easy for novices –DIY enthusiasts, interaction designers, kids at school, etc.– to realize custom applications that comprise capacitive multi-touch sensing. Results from a technical evaluation revealed a high signal-to-noise ratio and high spatial accuracy for robust multi-touch sensing in interactive prototypes. The approach is compatible with sensors of custom scale and curvature. We have formally evaluated sizes up to 12×12 electrodes, but have shown that larger sizes of 16×16 support accurate touch sensing as well. Results from our technical studies and implemented application demonstrators further show that *Multi-Touch Kit* is compatible with sensors fabricated using multiple materials and various rapid prototyping techniques. We hope our work will enable more widespread use and experimentation with multi-touch sensing.

ACKNOWLEDGMENTS

This project received funding from the European Research Council (ERC StG InteractiveSkin 714797). We thank Jan Dickmann, Amr Gomaa, and Sebastian Ruestig for their help with the fabrication of application examples, Muhammad Hamid and Rayhanul Islam Rumel for their help with video editing and Tobias Lange and Julian Ritter for their support with measuring the interference. We also thank the reviewers for their insightful comments.

REFERENCES

- [1] 2019. Code of Federal Regulations, Title 47: Telecommunication, 15.23 Home-built devices. <http://www.ecfr.gov/>. (2019). Retrieved: 04-04-2019.
- [2] H. Akhtar and R. Kakarala. 2014. A Methodology for Evaluating Accuracy of Capacitive Touch Sensing Grid Patterns. *Journal of Display Technology* 10, 8 (Aug 2014), 672–682. DOI: <http://dx.doi.org/10.1109/JDT.2014.2312975>
- [3] J. Bae, H. Cho, K. Song, H. Lee, and H. Yoo. 2012. The Signal Transmission Mechanism on the Surface of Human Body for Body Channel Communication. *IEEE Transactions on Microwave Theory and Techniques* 60, 3 (March 2012), 582–593. DOI: <http://dx.doi.org/10.1109/TMTT.2011.2178857>
- [4] Frank Beck and Bent Stumpe. 1973. *Two devices for operator interaction in the central control of the new CERN accelerator*. Technical Report. CERN.
- [5] Burke Davison. 2010. Techniques for robust touch sensing design. *AN1334 Microchip Technology Inc* (2010), 53.
- [6] Georg Freitag Markus Wacker Dietrich Kammer, Mandy Keck. 2010. Taxonomy and overview of multi-touch frameworks: Architecture, scope and features. In *Workshop on Engineering Patterns for Multitouch Interfaces*.
- [7] Paul Dietz and Darren Leigh. 2001. DiamondTouch: A Multi-user Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. ACM, New York, NY, USA, 219–226. DOI: <http://dx.doi.org/10.1145/502348.502389>
- [8] Tony Gray. 2019. *Projected Capacitive Touch*. Springer International Publishing, Cham. DOI: <http://dx.doi.org/10.1007/978-3-319-98392-9>
- [9] T. Grosse-Puppendahl, Y. Berghofer, A. Braun, R. Wimmer, and A. Kuijper. 2013. OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 152–159. DOI: <http://dx.doi.org/10.1109/PerCom.2013.6526726>
- [10] Tobias Grosse-Puppendahl, Andreas Braun, and Xavier Dellangnol. 2016. Prototyping Capacitive Sensing Applications with OpenCapSense. *GetMobile: Mobile Comp. and Comm.* 20, 2 (Oct. 2016), 16–21. DOI: <http://dx.doi.org/10.1145/3009808.3009814>
- [11] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017a. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA. DOI: <http://dx.doi.org/10.1145/3025453.3025808>
- [12] Tobias Grosse-Puppendahl, Christian Holz, Gabe A. Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017b. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 10. DOI: <http://dx.doi.org/10.1145/3025453.3025808>
- [13] Jefferson Y. Han. 2005. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 115–118. DOI: <http://dx.doi.org/10.1145/1095034.1095054>
- [14] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011a. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 441–450. DOI: <http://dx.doi.org/10.1145/2047196.2047255>
- [15] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011b. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, New York, New York, USA, 627. DOI: <http://dx.doi.org/10.1145/2047196.2047279>
- [16] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body As an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 453–462. DOI: <http://dx.doi.org/10.1145/1753326.1753394>
- [17] David Holman, Nicholas Fellion, and Roel Vertegaal. 2014. Sensing Touch Using Resistive Graphs. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 195–198. DOI: <http://dx.doi.org/10.1145/2598510.2598552>
- [18] Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2501–2510. DOI: <http://dx.doi.org/10.1145/1978942.1979308>
- [19] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 637–644. DOI: <http://dx.doi.org/10.1145/2642918.2647383>

- [20] Martin Kaltenbrunner and Ross Bencina. 2007. reacTIVision: A Computer-vision Framework for Table-based Tangible Interaction. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI '07)*. ACM, New York, NY, USA, 69–74. DOI:<http://dx.doi.org/10.1145/1226969.1226983>
- [21] Hsin-Liu (Cindy) Kao, Christian Holz, Asta Roseway, Andres Calvo, and Chris Schmandt. 2016. DuoSkin: Rapidly Prototyping On-skin User Interfaces Using Skin-friendly Materials. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 16–23. DOI:<http://dx.doi.org/10.1145/2971763.2971777>
- [22] Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, Cheng Zhang, and Gregory D Abowd. 2013a. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 363–372.
- [23] Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. 2013b. Instant Inkjet Circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 363–372. DOI:<http://dx.doi.org/10.1145/2493432.2493486>
- [24] Darren Leigh, Clifton Forlines, Ricardo Jota, Steven Sanders, and Daniel Wigdor. 2014. High Rate, Low-latency Multi-touch Sensing with Simultaneous Orthogonal Multiplexing. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 355–364. DOI:<http://dx.doi.org/10.1145/2642918.2647353>
- [25] James J. Fitzgibbon Michael C. Brenner. 1985. Surface acoustic wave touch panel system. (22 1985). US Patent US4644100A.
- [26] Microchip. 2012. Sensor Design Guidelines. (11 January 2012). Retrieved April 4, 2019 from <http://ww1.microchip.com/downloads/en/DeviceDoc/FAQs%20-%20Sensor%20Design%20Guidelines.pdf>.
- [27] Aditya Shekhar Nittala, Anusha Withana, Narjes Pourjafarian, and Jürgen Steimle. 2018. Multi-Touch Skin: A Thin and Flexible Multi-Touch Sensor for On-Skin Input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 33, 12 pages. DOI:<http://dx.doi.org/10.1145/3173574.3173607>
- [28] Simon Olberding, Jürgen Steimle, Suranga Nanayakkara, and Pattie Maes. 2015. CloudDrops: Stamp-sized Pervasive Displays for Situated Awareness of Web-based Information. In *Proceedings of the 4th International Symposium on Pervasive Displays (PerDis '15)*. ACM, New York, NY, USA, 47–53. DOI:<http://dx.doi.org/10.1145/2757710.2757718>
- [29] Joseph A. Paradiso and Neil Gershenfeld. 1997. Musical Applications of Electric Field Sensing. *Computer Music Journal* 21, 2 (1997), 69. DOI:<http://dx.doi.org/10.2307/3681109>
- [30] Ivan Poupyrev, Nan-Wei Gong, Shihoh Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. 2016. Project Jacquard: Interactive Digital Textiles at Scale. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4216–4227. DOI:<http://dx.doi.org/10.1145/2858036.2858176>
- [31] Jun Rekimoto. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*. ACM Press, New York, New York, USA, 113. DOI:<http://dx.doi.org/10.1145/503376.503397>
- [32] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 483–492. DOI:<http://dx.doi.org/10.1145/2207676.2207743>
- [33] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: Fabricating Custom Capacitive Touch Sensors to Prototype Interactive Objects. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 579–588. DOI:<http://dx.doi.org/10.1145/2380116.2380189>
- [34] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capriate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 253–258. DOI:<http://dx.doi.org/10.1145/2807442.2807503>
- [35] Ryan Seguine. 2007. Capacitive Sensing Techniques and Considerations. Mobile Handset DesignLine. (2007). Accessed: 2018-09-20.
- [36] Virag Varga, Gergely Vakulya, Alanson Sample, and Thomas R. Gross. 2018. Enabling Interactive Infrastructure with Body Channel Communication. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 169 (Jan. 2018), 29 pages. DOI:<http://dx.doi.org/10.1145/3161180>
- [37] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In

- Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).* ACM, New York, NY, USA, 2991–3000. DOI: <http://dx.doi.org/10.1145/2702123.2702391>
- [38] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. 2017. SkinMarks : Enabling Interaction on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA. DOI: <http://dx.doi.org/10.1145/3025453.3025704>
- [39] Andrew D. Wilson. 2010. Using a Depth Camera As a Touch Sensor. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 69–72. DOI: <http://dx.doi.org/10.1145/1936652.1936665>
- [40] Raphael Wimmer and Patrick Baudisch. 2011. Modular and Deformable Touch-sensitive Surfaces Based on Time Domain Reflectometry. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 517–526. DOI: <http://dx.doi.org/10.1145/2047196.2047264>
- [41] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 1–14.
- [42] Yang Zhang, Chouchang (Jack) Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++: Room-Scale Interactive and Context-Aware Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 273, 15 pages. DOI: <http://dx.doi.org/10.1145/3173574.3173847>
- [43] Thomas G. Zimmerman, Joshua R. Smith, Joseph A. Paradiso, David Allport, and Neil Gershenfeld. 1995. Applying Electric Field Sensing to Human-computer Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 280–287. DOI: <http://dx.doi.org/10.1145/223904.223940>