



Paper-Based, Capacitive Touch Pads

Citation

Mazzeo, Aaron D., William B. Kalb, Lawrence Chan, Matthew G. Killian, Jean-Francis Bloch, Brian A. Mazzeo, and George M. Whitesides. 2012. Paper-Based, Capacitive Touch Pads. *Advanced Materials* 24, no. 21: 2850–2856. doi:10.1002/adma.201200137

Published Version

doi:10.1002/adma.201200137

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:34753836>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

Paper-based, Capacitive Touch Pads

Aaron D. Mazzeo¹, Will B. Kalb¹, Lawrence Chan¹, Matthew G. Killian¹,
Jean-Francois Bloch², Brian A. Mazzeo³, and George M. Whitesides^{1,4,*}

¹Department of Chemistry and Chemical Biology, Harvard University

²Institut National Polytechnique de Grenoble

³Department of Electrical and Computer Engineering, Brigham Young University

⁴Wyss Institute for Biologically Inspired Engineering at Harvard University

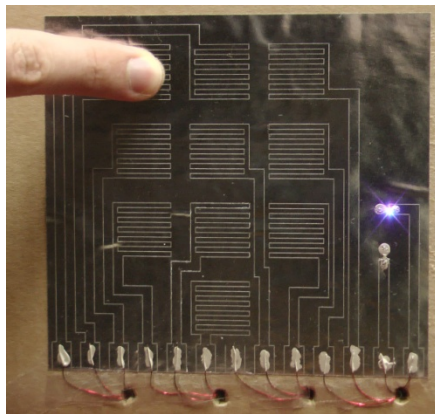
*corresponding author: gwhitesides@gmwgroup.harvard.edu

Keywords: paper, capacitance, touch pad, sensor, flexible electronics

Table of Contents:

Mazzeo et al. describe methods of patterning metallized paper to create touch pads of arrayed buttons that are sensitive to contact with both bare and gloved fingers. The paper-based keypad shown detects the change in capacitance associated with the touch of a finger to one of its buttons. Mounted to an alarmed cardboard box, the keypad requires the appropriate sequence of touches to disarm the system.

Image for Table of Contents:



This paper describes low-cost, thin, and pliable touch pads constructed from a commercially available, metallized paper commonly used as packaging material for beverages and book covers. The individual keys in the touch pads detect changes in capacitance or contact with fingers by using the effective capacitance of the human body and the electrical impedance across the tip of a finger. To create the individual keys, a laser cutter ablates lines through the film of evaporated aluminum on the metallized paper to pattern distinct, conductive regions. This work includes the experimental characterization of two types of capacitive buttons and illustrates their use with applications in a keypad with 10 individually addressable keys, a keypad that conforms to a cube, and a keypad on an alarmed cardboard box. With their easily arrayed keys, environmentally benign material, and low cost, the touch pads have the potential to contribute to future developments in disposable, flexible electronics, active, “smart” packaging, user interfaces for biomedical instrumentation, biomedical devices for the developing world, applications for monitoring animal and plant health, food and water quality, and disposable games (e.g., providers of content for consumer products).

There is no simple method of integrating buttons with structures on single-use or throw-away devices. Current commercial buttons are not thin enough, inexpensive enough, or easy enough to array seamlessly with paper-based products for disposable applications. The touch pads in this work are thin ($\sim 60\ \mu\text{m}$ in some cases), simple to array, fabricated by etching patterns into metallized paper, low-cost ($< \$0.25\ \text{m}^{-2}$ for the thin grade of metallized paper we use in this work), and lightweight (100s of g m^{-2}). The individual keys measure changes in capacitance when touched by a user, and the buttons require no physical displacement of conductive elements. Even though the individual keys on the touch pads detect changes in capacitance, the paper-based keypads are still functional when touched by fingers in nitrile gloves.

Developments in paper-based electronics include ring oscillators with organic electronics^[1], transistors^[2-4], methods for patterning conductive traces^[5-7], speakers^[8], super capacitors^[9], batteries^[10], MEMS^[11], and solar cells^[12]. Each of these developments focuses on a single technological advance that would enable new types of consumer products. Many types of new consumer products will require some form of user interface or input. In order to gather key strokes from end users with polymer-based or textile-based flexible electronics, it is possible to use changes in resistance^[13] or capacitance^[14-17]. Obviously, technology being developed commercially – Samsung, 3M, Displax, Elo TouchSystems, and other companies are pursuing technology for transparent touch-based systems – for use with glass-based displays on mobile phones or computer screens might also be applicable to future developments in one-time use, paper-based electronics. Conventional capacitive touch screens detect changes in capacitance through a piece of glass with a thickness much greater than that of the polymeric coatings on top of the metallized paper. Thus, instead of detecting changes in capacitance of 10s or 100s of pF, they use optimized electronics to detect changes closer to 1 pF. Besides a patent for capacitive buttons on posters^[18], there are no descriptions for simple methods of receiving input/key strokes on paper-based substrates.

Metallized paper is a commodity. It is easy to ablate or cut the metal film with a laser or razor blade, and it is both durable (i.e., documents can last for years) and disposable. Touch pads made from metallized paper may fill a niche between displays with touch screens and conventional, commercial keypads. The metallized paper (A-238 and A-550 from Vacumet Corporation) consists of cellulose, polymeric coatings, and evaporated aluminum (see SI Figure 1). The cellulose serves primarily as structural support for a ~10-nm thick, conductive layer of evaporated aluminum. The metallized layer without structural support would not have sufficient strength for manual handling or use. The polymer-based portions of the paper assist in adhering

the metallized layer to the cellulose and protect the thin, metallized layer from scraping or other environmental damage. The uncoated side of the metallized paper has exposed cellulose-based paper, which can absorb moisture, while the polymeric coating on the opposite side of the paper blocks moisture. Nevertheless, using the laser cutter to ablate through the polymeric coating can make the metallized paper more susceptible to moisture on the coated side. Experiments have also shown that a paper-based button exposed to moisture can exhibit a change in effective capacitance (see Supporting Information).

A button or key on a touch pad is a physical device that signals a single, binary change in state when touched by a user, and returns to its original state when no longer touched. To array buttons on metallized paper, we used a laser cutter (VersaLaser from Universal Laser Systems) to ablate lines through the aluminized layer of the paper to create discrete, conductive regions. These conductive regions served as both electrodes for the buttons and conducting traces that lead to external electronics.

Each button on the touch pads described in this work is a capacitor. The individual capacitors, or buttons, have a capacitance – the ratio of storable charge to the applied electric potential – that changes when touched with a finger. Between parallel plates, capacitance (C) scales with the inverse of the gap (Equation 1), where ϵ is the dielectric constant of the material separating the plates, ϵ_0 is the permittivity of free space (8.854×10^{-12} F/m), A is the cross-sectional area of the plates, and d is the distance between plates.

$$C = \epsilon\epsilon_0 A/d \quad (1)$$

In the case of capacitive coupling of a finger to a parallel-plate capacitor, we consider the finger to contribute electrical impedance (Z_b) (see Figure 1 A-D). This electrical impedance acts as a

“body capacitor” with an approximate capacitance of 100 pF and an approximate resistance of 1.5 k Ω ^[19]. Because of the polymer coated on the top surface of the metallized paper, the finger does not make direct, conductive contact with the top plate of the capacitor. As the finger approaches the polymeric coating above the metallized layer, the capacitance between the finger and the top plate of the button (C_g) increases. When the finger makes contact with the button, the body capacitor then makes a significant, detectable contribution to increasing the effective capacitance measured at the button.

Buttons using parallel-plate capacitors require two sheets of metallized paper. The bottom sheet (not touched with a finger) serves as a grounded electrode, while the top sheet (touched with a finger) carries an applied potential and is electrically active. To create buttons on a single sheet of metallized paper, we patterned active and grounded electrodes adjacent to each other (Figure 1 F). A finger can bridge between the active electrode and grounded electrode on a single side of metallized paper (Figure 1 G). For a single button with a single sheet of metallized

Figure 1

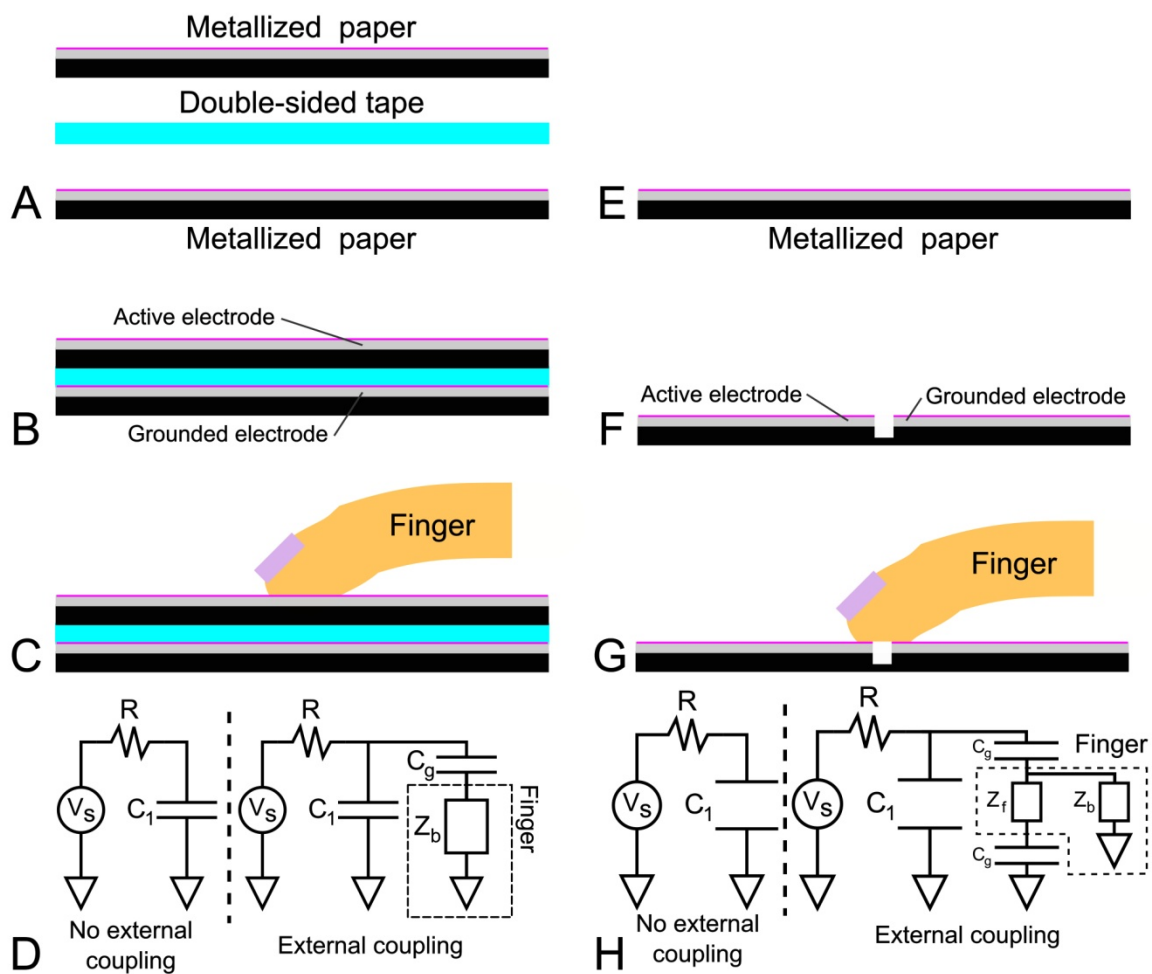


Figure 1: Schematic layouts for two types (A-D and E-H) of capacitive keys made with metallized paper. (A) Components for a two-layer key based on capacitive coupling between a finger and an active electrode. (B) Assembled capacitor from components shown in A. (C) A finger serves as an external, electrical coupling to the active electrode. (D) Description of the electrical circuits without (left) and with (right) the coupling of the finger to the active electrode. R is the value of the resistor placed in series with an untouched button with effective capacitance of C_I . The source of electric potential (V_s) applies a stepped, repeating pattern to the circuit. (E) Metallized paper used for single-layer touch pads. (F) Etched or ablated lines through the conductive portion of the metallized paper designate regions or traces of conductance. (G) A finger bridges the gap between an active electrode and a grounded electrode to cause a measurable change in capacitance. (H) Description of the electrical circuits without (left) and with (right) the coupling of the finger to the electrodes.

paper, assigning either of the electrodes to ground would not make a difference provided the areas of the electrodes are comparable. Nevertheless, a single sheet of multiple buttons benefits from having a single grounded electrode that surrounds all of the buttons without the need for individually grounded traces running to each button.

The change in electrical impedance with a finger bridging between the grounded and active electrodes is a result of the skin acting as a dielectric material and the ionic solution within or on the surface of the finger acting as a conductive material. The dominant electrical path from one metallized electrode to an adjacent electrode goes from a metallized electrode, through the insulating coating on the metallized paper, through or along the surface of the finger, through the insulating coating above the adjacent electrode, to the metallized layer on the adjacent electrode. The electrical impedance through or along the surface of the human finger (Z_f) acts in parallel with the impedance (Z_b) through the finger and the body of the person. Given our objective and microprocessor-based method of detecting on/off touches, we model the variable impedance of each key as having an effective capacitance or time constant associated with an “RC-circuit” (see SI Figure 3).

To aid our understanding of the interaction of a finger with our capacitive buttons, we have detected touches successfully with gloved human fingers and non-human finger-portions of nitrile gloves filled with aqueous solutions. Human fingers in gloves (SterlingTM nitrile powder-free examination gloves from Kimberly-Clark) caused smaller changes in capacitance than those observed with bare human fingers, but the changes in capacitance with human fingers in gloves were still measurable. The layer of rubber between the finger and the conductive layer of the metallized paper increased the distance between the finger and the electrodes (relative to a bare finger touching a button) and thus, caused a reduction in the overall measured capacitance. The finger-portion of a glove filled with an aqueous solution (deionized water in one case and

~150 mM NaCl in another) also registered responses with the system shown in SI Video 1. For both cases (glove filled with deionized water and glove filled salt water), the apparent change in capacitance was less than that with a gloved finger. The reduced change in capacitance may have been from the effective thickness of the glove being greater than that of human skin, reduced conductivity of the aqueous solution, or increased resistance in the path to electrical ground through the liquid-filled portion of the glove. In contrast to a liquid-filled glove, an air-filled glove registered almost no observable change in capacitance when pressed against a key (i.e., capacitive buttons would rarely trigger a response by lighting an LED). We attribute these spurious readings to the accumulation of surface charge on the outer surface of the glove.

Vacumet Corporation evaporates approximately 0.05 g/m^2 of aluminum onto papers of different thicknesses. Assuming a density of 2700 kg/m^3 for aluminum, the thickness of the layer of evaporated aluminum is less than 20 nm. The two grades of metallized paper used in this work have specified average thicknesses and ranges of $56 \pm 6 \text{ }\mu\text{m}$ and $137 \pm 5 \text{ }\mu\text{m}$. These specified thicknesses include paper, coatings of polymer, and evaporated aluminum (see SI Figure 1). By measuring the electrical resistance of different cut sheets of metallized paper, we estimate the average thickness and standard deviation of the evaporated aluminum to be $13 \pm 2 \text{ nm}$ (see SI Figure 2).

Figure 1 A-D show the configuration for keys based on coupling between a finger and a parallel-plate capacitor with two layers of metallized paper. To fabricate two-layered buttons, we etched the traces and cut out the overall dimensions of the sheets of metallized paper using a laser cutter. Once cut, we used double-sided tape (Indoor Carpet Tape from 3M) to hold the pieces of metallized paper together. We brushed on silver conductive adhesive (Silver Conductive Adhesive 503 from Electron Microscopy Sciences) to connect traces or tabs on the metallized paper electrically with external electronics through conductive pads or wired leads

To measure the changing effective capacitance of the buttons and demonstrate interactive applications using paper-based touch pads, we used the Arduino platform (UNO and MEGA 2560). Arduino is a microprocessor-based system with open-source hardware and software for signal processing and computation (see <http://www.arduino.cc/>). The Arduino platform has also demonstrated its utility for applications with flexible or wearable electronics^[20]. The Arduino is capable of applying a step input to a resistor and capacitor in series and measuring the time required for the potential on the capacitor to reach 2 V (see an example at <http://arduino.cc/it/Tutorial/CapacitanceMeter> and the Supporting Information for code used in SI Video 1). Using this measured time, it is possible to estimate an “RC” time constant and calculate an effective capacitance for a circuit with a known resistance (see SI Figure 3). The typical resistance for the resistor placed in series with a button was 100 k Ω or 1 M Ω . To buffer the potential across the capacitor against the impedance of the inputs leading to the Arduino, we used an operational amplifier (LM324) with unity gain. We also used a demultiplexing chip (TI CD4067BE 1:16) addressed with four binary outputs to measure the responses of 10 to 48 individual buttons with only one to three electrical inputs. The Arduino-based system and accompanying electronics required about 0.4 W, even though an estimate for the power required for an individual button of 100 pF is 250 nW (see Supporting Information for both estimates).

At this point, the electronics for detecting changes in capacitance are neither mechanically compliant nor inexpensive enough to be considered disposable after a single use. The packaged electronics might plausibly be multiple-use with only the paper-based system being disposable. It is also conceivable that the electronics could be made inexpensive enough through mass-production and conventional techniques for fabricating semiconductor-based circuits (e.g., disposable RFID tags and cell phones are representative efforts to produce low-cost, complex electronic systems). To make the required circuits flexible, DuPont™ Pyralux® or the metallized

paper itself might serve as a flexible substrate. Nevertheless, we envision applications, which might use reusable electronics for interfacing the disposable touch pads with standard protocols for communication with personal computers or commercial electronic databases (e.g., a small box with reusable electronics for sensing and wireless transfer of data might ship multiple times with distinct disposable touch pads and packaging).

To test our touch pads, we constructed single buttons (Figure 2). One button consisted of parallel plates (Figure 2 A and B), while the other button consisted of a single layer of metallized paper with etched traces to form inter-digitated electrodes, or macroscopic buttons that were easy to touch (Figure 2 D and E). We mounted the buttons to a manila folder with double-sided tape, connected the electrical leads of the buttons to an Arduino-based system, and used the serial output from the Arduino to log data on a laptop computer. Figure 2 C and F show the measured effective capacitances in a box plot – based on the time required for the potential across the capacitive button in series with a resistor of 1.01 M Ω to reach 2 V with a step input of 5 V – measured over five seconds of sampling for seven untouched buttons and seven touched buttons with an index finger from a single user. The touch button with a single sheet of metallized paper was more sensitive than the touch button with two sheets (see Supporting Information for more details). Touch buttons in a single sheet are also easier to construct than touch buttons in a two-layer configuration (i.e., no need to align and bond two layers of metallized paper to each other).

To demonstrate the functionality of the touch button in Figure 2 D, we pressed the button shown over 2000 times, and it continued to function reproducibly without exhibiting drift (i.e., it was not necessary to adjust the threshold of detection. That said, we have implemented a scheme for other touch pads with moving averages for the thresholds on the Arduino to ensure proper

Figure 2

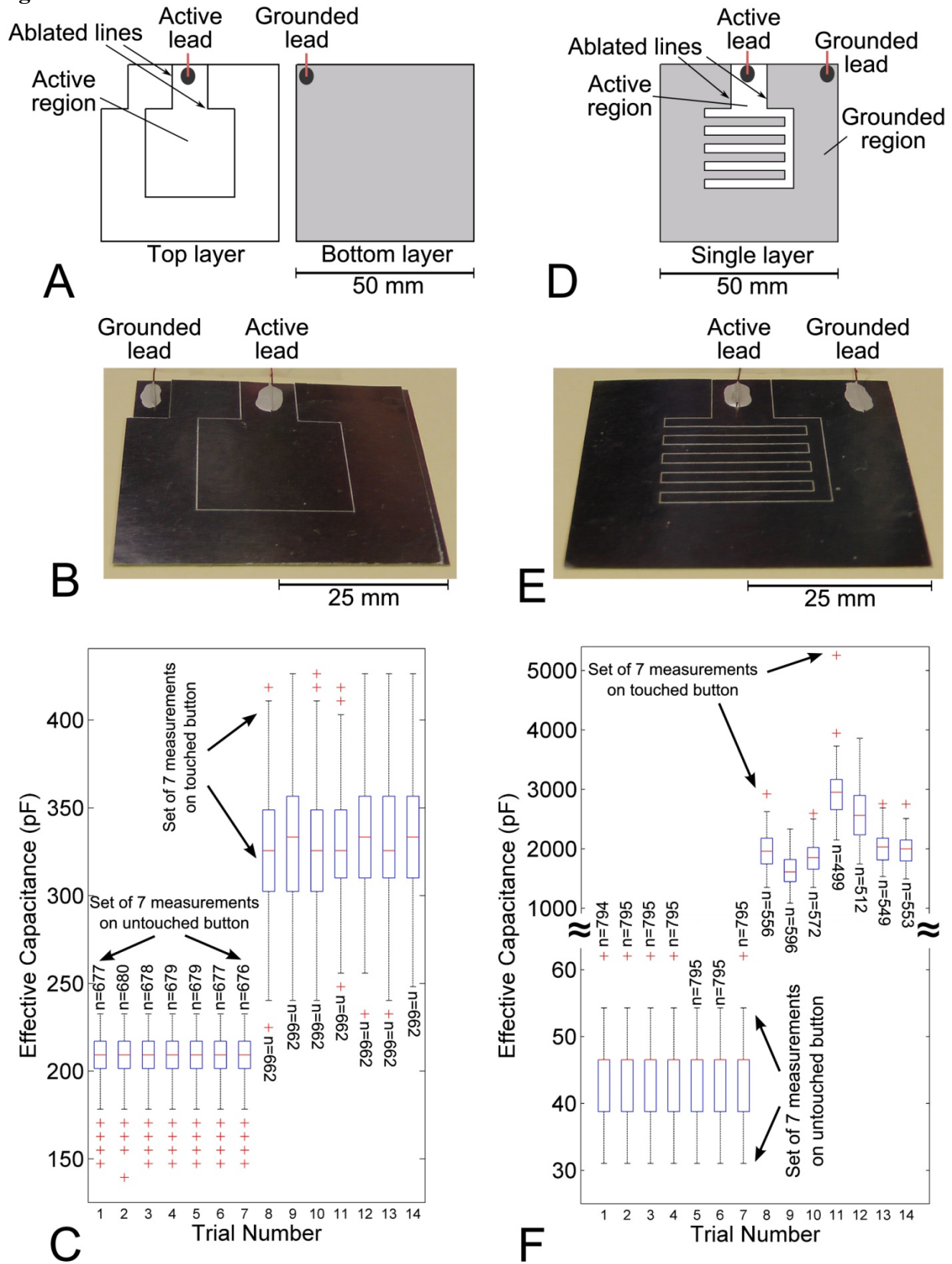


Figure 2: Capacitive buttons fabricated with metallized paper. (A) The top piece was Vacumet A-238 (thickness of 56 μm) with etched lines outlining the touched, active region, and the bottom piece (grey) was Vacumet A-550 (thickness of 137 μm) and served as an electrical ground. Manually cut pieces of double-sided tape bound the two pieces of metallized paper together. (B) Photo of the button shown in A. (C) Box plots for the effective capacitance of the button shown in B for two sets (touched with a bare finger and untouched) of seven measurements, each lasting five seconds and having more than 660 sampled points ($n > 660$). The whiskers extend 1.5 X the interquartile length dictated by the distance between the top (75th percentile) and bottom (25th percentile) of the boxes. The horizontal line in the box shows the median, and the crosses represent outliers. (D) A single piece of Vacumet A-238 etched with inter-digitated fingers forming a button. The greyed region served as electrical ground. (E) Photo of the tested button shown in D. (F) Box plot for the effective capacitance of the button shown in E for two sets (touched and untouched) of seven measurements, each lasting five seconds and having more than 490 points ($n > 490$). The attributes of the box plot are the same as those in C, except the median is at the top of the box for the untouched cases.

responses over extended periods of time (see <http://www.arduino.cc/en/Tutorial/Smoothing> and included code for SI Video 1 in the Supporting Information). We also measured the response of the button to hundreds of presses with a bare finger and a gloved finger (nitrile-based glove from Kimberly-Clark). Figure 3 shows some of the measurements taken with the Arduino-based system after the button had already received more than 1000 presses. The red crosses shown in Figure 3 A and B indicate when the Arduino-based system detected a change in the state of the button relative to a fixed threshold of an effective capacitance of 78 pF. Figure 3 C and D show the distribution of peak values in measured capacitance during each press. The touches with a finger in a gloved hand elicited a much smaller change in capacitance than that observed with a bare finger, and future improvements in electronics might improve sensitivity to gloved fingers.

To make a keypad with ten buttons (Figure 4), we ablated lines in a piece of Vacumet A-238 (thickness of 60 μm) to form conductive traces and inter-digitated regions for ten separate keys. After ablating the lines in the metallized paper, we attached the keypad to a manila folder with double-sided tape. To connect the keypad to external electronics, we attached 30-gauge wire to each of the contact pads (ten contact pads for the keys and one contact pad for the ground) with conductive paste. To address the separate keys, we used an input and five outputs (one output supplied a stepped signal to the RC circuits and the other four addressed a demultiplexer) on the Arduino. The stepped signal from the Arduino went through the same resistor for all ten keys but went through a separate capacitive region as dictated by a 1:16 demultiplexer (TI CD4067BE). Figure 4 C shows measurements taken when a gloved finger touched each key individually, touched keys “0” and “1” simultaneously, and touched keys “1” and “2” simultaneously. With another 10-button keypad mounted to a piece of plastic (see SI Video 1), more than 25 different people were all able to trigger the appropriate LEDs to respond with the touch of their bare fingers.

Figure 3

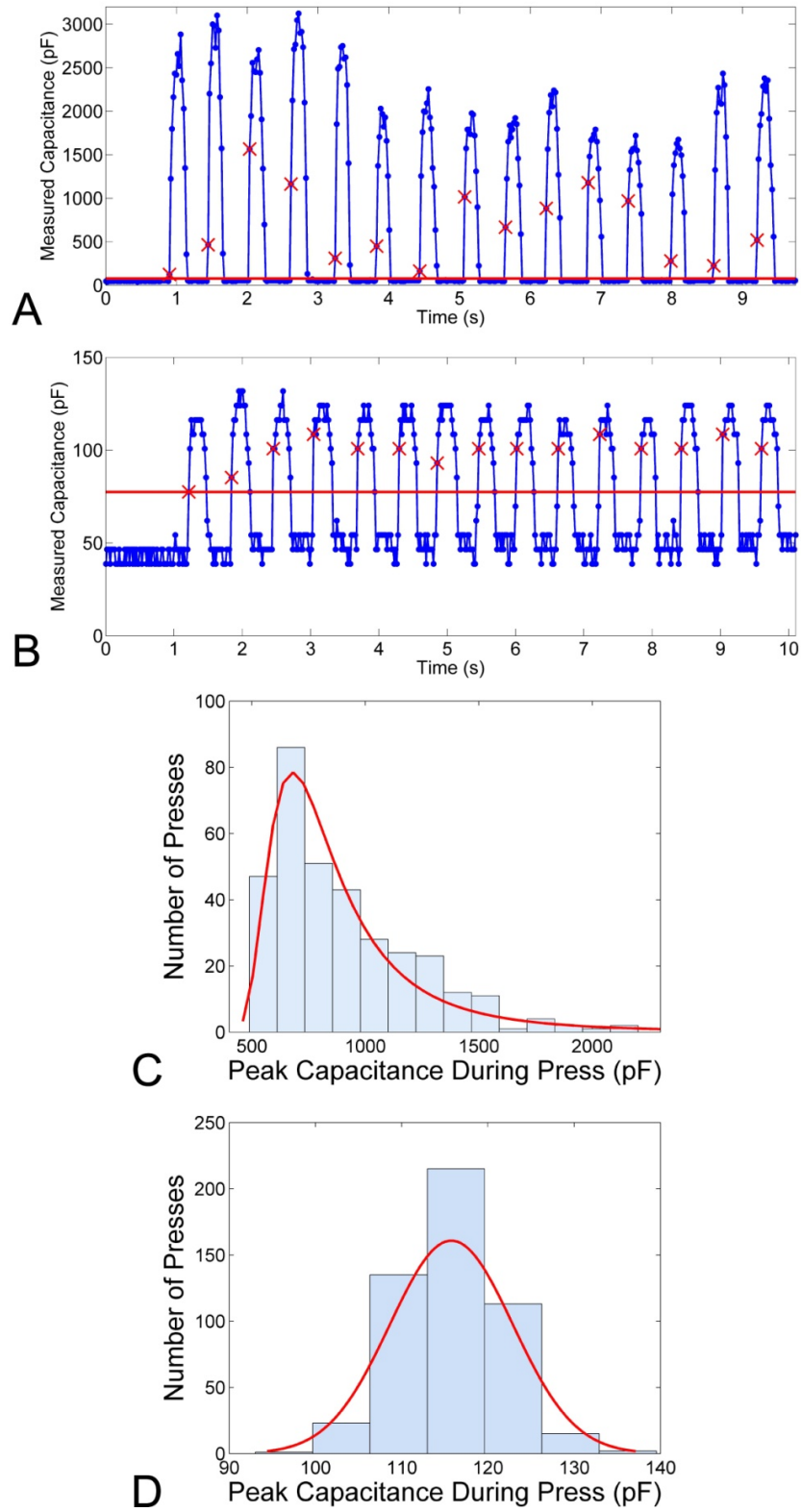


Figure 3: Measured changes in capacitance for the capacitive button shown in Figure 2 D after the button had already experienced more than 1000 touches. (A) Measurements taken with a bare finger touching the button. The ticks show when the electronic system registered a capacitance greater than or equal to the threshold of 78 pF. The red baseline shows the threshold set at 78 pF. (B) Measurements taken with a gloved finger and the same threshold (red baseline) shown in A. (C) The distribution (general extreme value) of peak capacitances measured during 335 presses of the button with a bare finger. The data had a minimum measured peak at 490 pF. (D) The distribution (normal) of peak capacitances measured during 504 presses of the button with a gloved finger. The mean was 120 ± 7 pF (± 1 standard deviation) for 504 presses, and the minimum peak measured was at 93 pF.

Figure 4

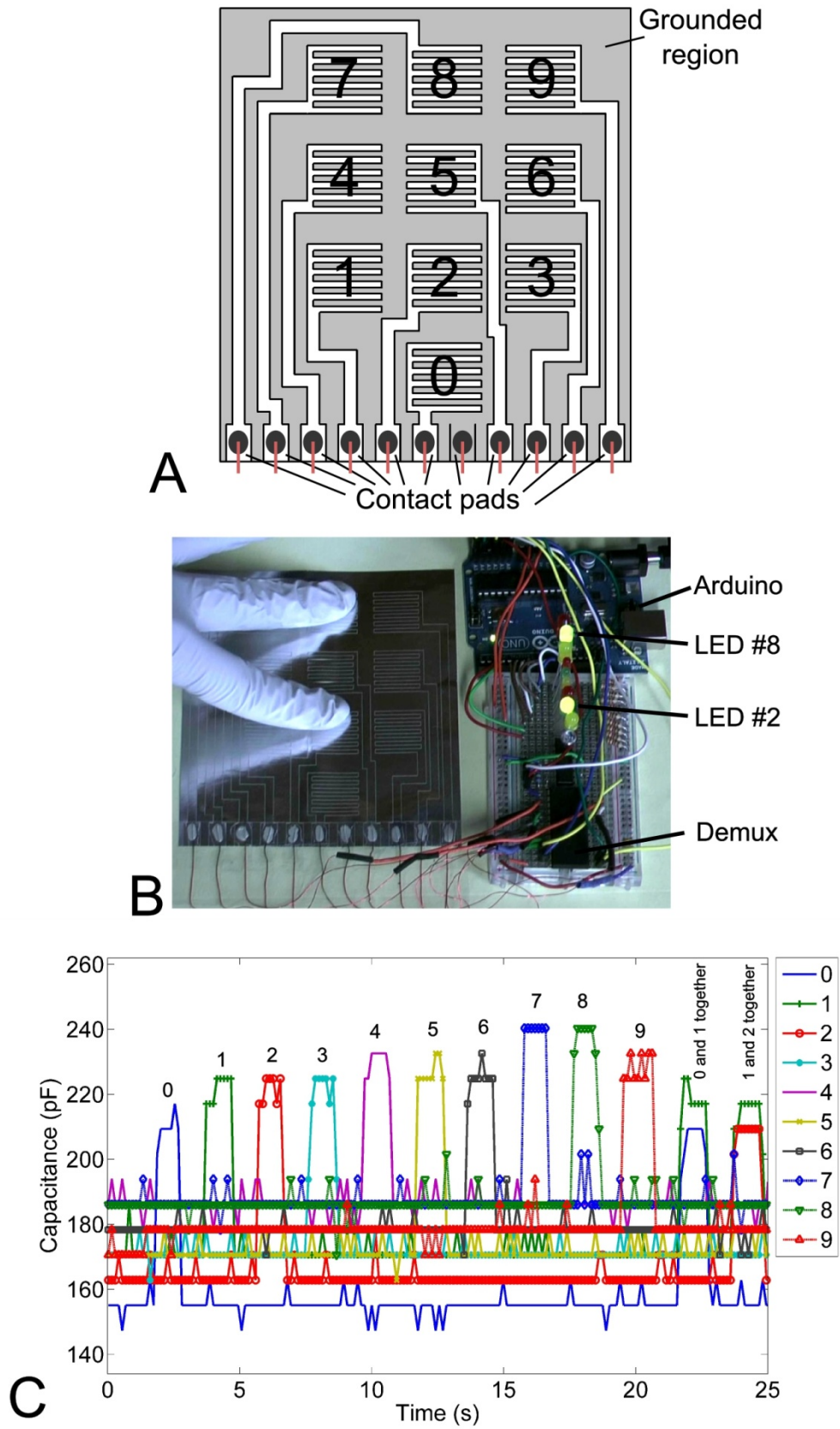


Figure 4: Touch pad with 10 keys that produced measurable changes in capacitance when touched with a bare or gloved finger. (A) The single layer (105 mm x 116 mm) layer comprises discrete keys (17 mm x 19 mm) with metallized traces leading to contact pads for connection to the Arduino-based system. (B) Photo of the completed keypad being touched with two gloved fingers on key “2” and key “8”. (C) Measured responses at each of the 10 buttons to touches with a gloved finger. The change in capacitance was greater with a bare finger than that measured with a gloved finger.

To demonstrate the fabrication of a three-dimensional keypad from a two-dimensional patterned piece of metallized paper, we ablated lines in a piece of Vacumet A-550 (thickness of 140 μm) and created traces for six individual buttons as shown in Figure 5 A. Then, we gently folded (sharp creases could cause a break in electrical conductivity) the two-dimensional pattern around a cube to form the three-dimensional keypad shown in Figure 5 B and C. Using the Arduino-based electronics and the same demultiplexer used for the keypad with 10 buttons, the three-dimensional keypad detected touches with either bare or gloved fingers and lit corresponding LEDs (see SI Video 2).

The paper-based touch pads described in this work may be useful as active packaging to prevent theft of the contents of containers. To demonstrate a possible implementation (see Figure 6 and SI Video 3), we attached a piece of metallized paper (thickness of 60 μm) with a 10-button keypad and two LEDs to a cardboard box with double-sided tape (Figure 6 A and B). The keypad served as the user interface to arm or disarm an alarm “built” into the box.

When armed, the LEDs were off. Opening the top lids on the box caused a decrease in the capacitance between the two pieces of metallized paper taped to the lids (capacitive switch). This decrease in capacitance, unlike the increases experienced by the buttons when touched with a finger, triggered the alarm, sounded a buzzer, and lit up both LEDs on the metallized paper. For purposes of demonstration, closing the lids caused the alarm to stop sounding. To disarm the alarm, the user entered a numeric code by touching the keys on the keypad. With every press of one of the keys, the blue LED would flash to provide visual feedback to the user. To demonstrate the functionality of all the keys in SI Video 3, we set the password to “0,1,2,3,4,5,6,7,8,9”. When disarmed, the electronics lit the green LED, and opening the box did not trigger the alarm. To arm the alarm from the disarmed state, a user hit any button on the keypad, and the LEDs returned to an unlit state.

Figure 5

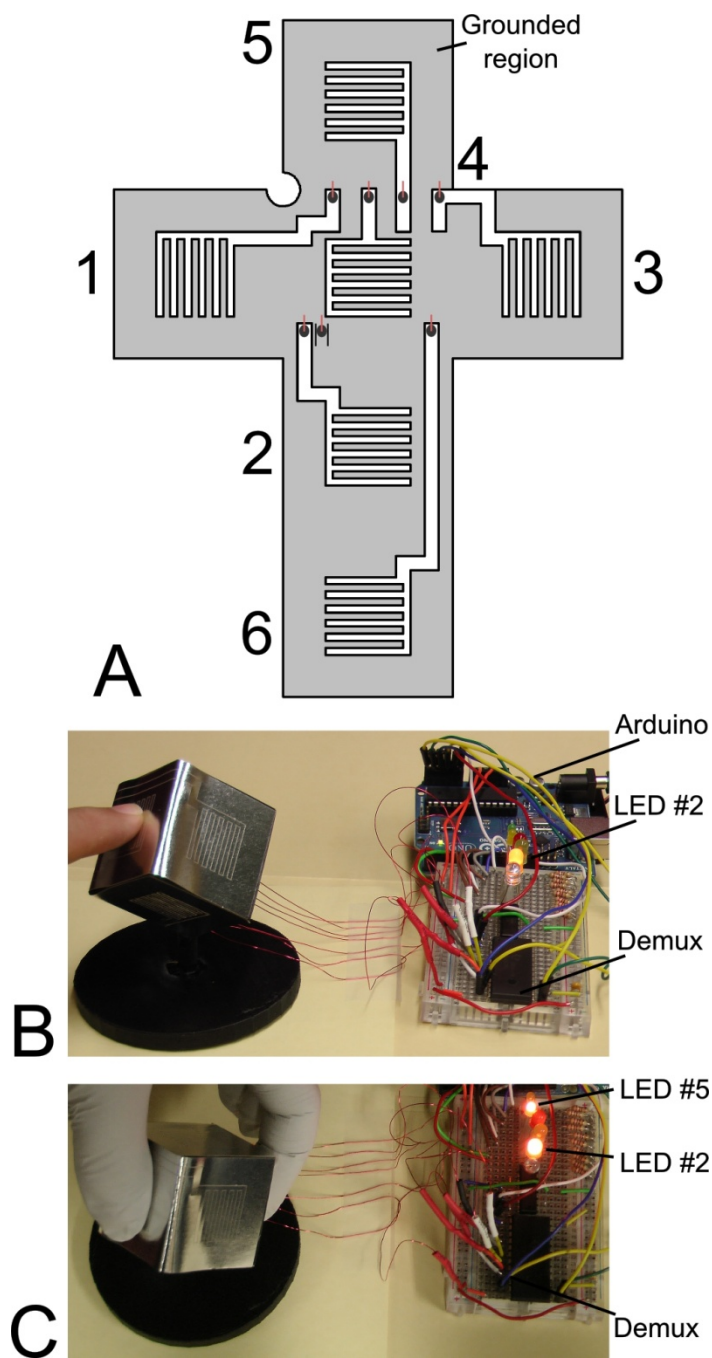


Figure 5: Paper-based touch pad fixed to a cube. (A) Layout for six buttons—one button for each face. Each edge of the cube has a length of 38 mm. (B) Image and associated output for a bare

finger touching the button on face number “2”. (C) Image and associated output for a gloved thumb and index finger making simultaneous contact with buttons number “5 and “2”.

Figure 6

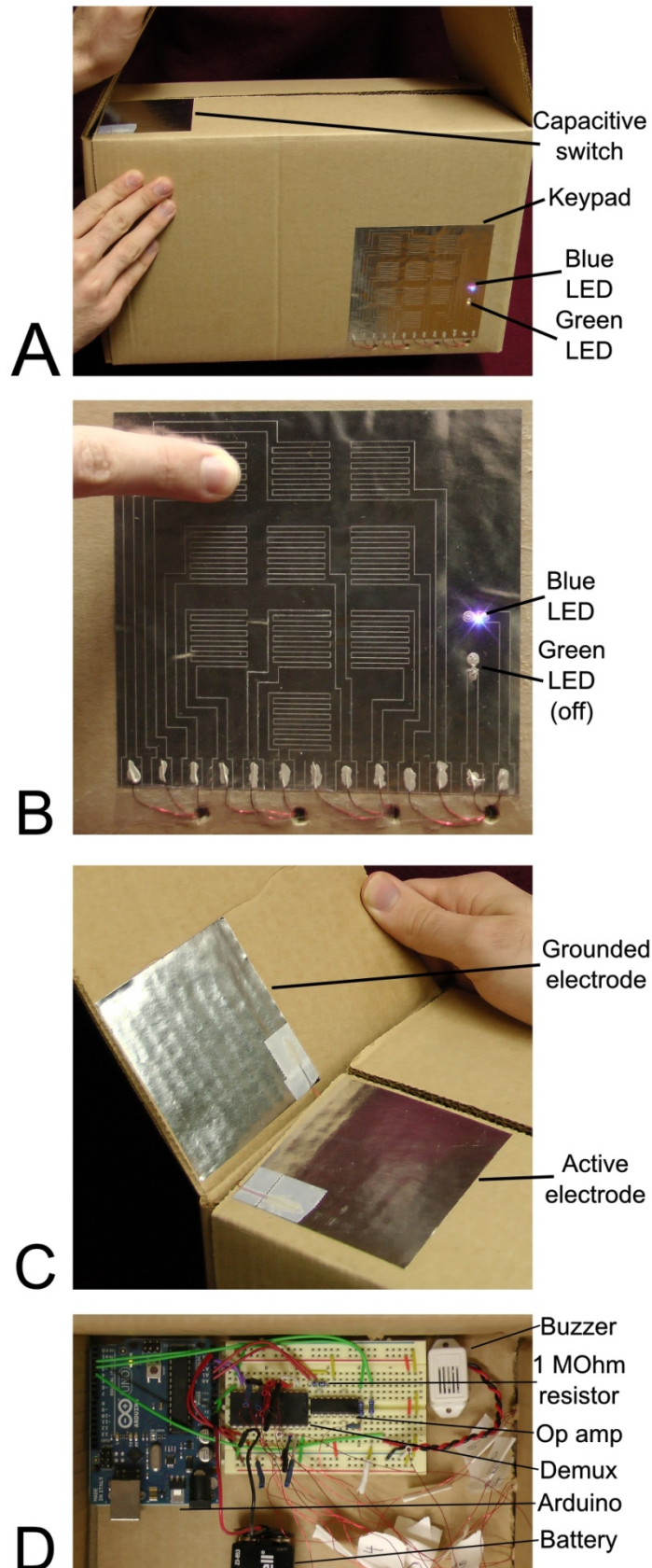


Figure 6: An alarmed cardboard box with a paper-based touch pad. (A) On an outside face of the box, the paper-based touch pad had accompanying LEDs to provide feedback to the user. Both LEDs turned on when the alarm went off. In the upper left region, there was a capacitive switch to detect whether or not the box was open. (B) The keypad and accompanying LEDs. The keypad had to receive the appropriate code to disable the alarm. The blue LED flashed whenever a button was pushed. (C) Close-up photo of the capacitive switch. (D) All the required electronics inside the box for operating the alarm. The buzzer sounded when the alarm went off.

Capacitive sensing and metallized paper provide a low-cost solution for integrating electronic interfaces with paper-based products. The paper-based keypads in this work are disposable, are as thin as 60 μm , are simple to etch in a single layer of metallized paper, and can consist of arrays of buttons. The keypads respond to both bare and gloved fingers, but performance is more reliable with bare fingers. Paper that serves as a structural support for very thin layers of metal (approximate thickness of 10 nm) has the potential to provide solutions for future developments in sensors and flexible electronics.

Acknowledgements

This work was supported by Defense Advanced Research Projects Agency (DARPA) N/MEMS S&T Fundamentals Program under grant no. N66001-1-4003 issued by the Space and Naval Warfare Systems Center Pacific (SPAWAR) to the Micro/nano Fluidics Fundamentals Focus (MF3) Center, the Gates Foundation, and DARPA Transient Electronics grant no. W911NF-11-1-0254. In addition, we would like to thank Steven A. Morin, Sam Liss, Adam A. Stokes, Jabulani R. Barber, Robert F. Shepherd, and Courtney Hilliard for helpful discussions.

References

- [1] F. Eder, H. Klauk, M. Halik, U. Zschieschang, G. Schmid, C. Dehm, *Appl. Phys. Lett.* **2004**, *84*, 2673.
- [2] R. Martins, A. Nathan, R. Barros, L. Pereira, P. Barquinha, N. Correia, R. Costa, A. Ahnood, I. Ferreira, E. Fortunato, *Adv. Mater.* **2011**, *23*, 4491.
- [3] S. Yun, S. D. Jang, G. Y. Yun, J. H. Kim, J. Kim, *Appl. Phys. Lett.* **2009**, *95*.
- [4] R. Martins, P. Barquinha, L. Pereira, N. Correia, G. Goncalves, I. Ferreira, E. Fortunato, *Appl. Phys. Lett.* **2008**, *93*.
- [5] A. Russo, B. Y. Ahn, J. J. Adams, E. B. Duoss, J. T. Bernhard, J. A. Lewis, *Adv. Mater.* **2011**, *23*, 3426.
- [6] A. C. Siegel, S. T. Phillips, M. D. Dickey, N. S. Lu, Z. G. Suo, G. M. Whitesides, *Adv. Funct. Mater.* **2010**, *20*, 28.

- [7] D.-H. Kim, Y.-S. Kim, J. Wu, Z. Liu, J. Song, H.-S. Kim, Y. Y. Huang, K.-C. Hwang, J. A. Rogers, *Adv. Mater.* **2009**, *21*, 3703.
- [8] H. Tian, T. L. Ren, D. Xie, Y. F. Wang, C. J. Zhou, T. T. Feng, D. Fu, Y. Yang, P. G. Peng, L. G. Wang, L. T. Liu, *ACS Nano* **2011**, *5*, 4878.
- [9] Z. Weng, Y. Su, D. W. Wang, F. Li, J. H. Du, H. M. Cheng, *Adv. Energy Mater.* **2011**, *1*, 917.
- [10] L. B. Hu, H. Wu, F. La Mantia, Y. A. Yang, Y. Cui, *ACS Nano* **2010**, *4*, 5843.
- [11] X. Liu, M. Mwangi, X. Li, M. O'Brien, G. M. Whitesides, *Lab on a Chip* **2011**, *11*, 2189.
- [12] M. C. Barr, J. A. Rowehl, R. R. Lunt, J. J. Xu, A. N. Wang, C. M. Boyce, S. G. Im, V. Bulovic, K. K. Gleason, *Adv. Mater.* **2011**, *23*.
- [13] R. K. Kramer, C. Majidi, R. J. Wood, "Wearable tactile keypad with stretchable artificial skin", presented at *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May, **2011**.
- [14] D. Cotton, I. M. Graz, S. P. Lacour, *Sensors Journal, IEEE* 2009, *9*, **2008**.
- [15] A. R. Deangelis, D. B. Wilson, B. A. Mazzeo, US Patent 7,578,195, **2009**.
- [16] A. R. Deangelis, B. D. Wilson, B. A. Mazzeo, US Patent 7,395,717, **2008**.
- [17] A. R. Deangelis, B. D. Wilson, B. A. Mazzeo, US Patent 7,301,351, **2007**.
- [18] R. Hagglund, T. Unander, A. From, P. Wagberg, US Patent App. 20,070/018,998, **2003**.
- [19] *ESD Fundamentals*, ESD Association, Rome, NY **2010**.
- [20] L. Buechley, M. Eisenberg, *Pervasive Computing, IEEE* **2008**, *7*, 12.

Paper-based, Capacitive Touch Pads

Aaron D. Mazzeo¹, Will B. Kalb¹, Lawrence Chan¹, Matthew G. Killian¹,

Jean-Francois Bloch², Brian A. Mazzeo³, and George M. Whitesides^{1,4,*}

¹Department of Chemistry and Chemical Biology, Harvard University

²Institut National Polytechnique de Grenoble

³Department of Electrical and Computer Engineering, Brigham Young University

⁴Wyss Institute for Biologically Inspired Engineering at Harvard University

*corresponding author: gwhitesides@gmwhgroup.harvard.edu

Supporting Information:**Characterization of Metallized Paper**

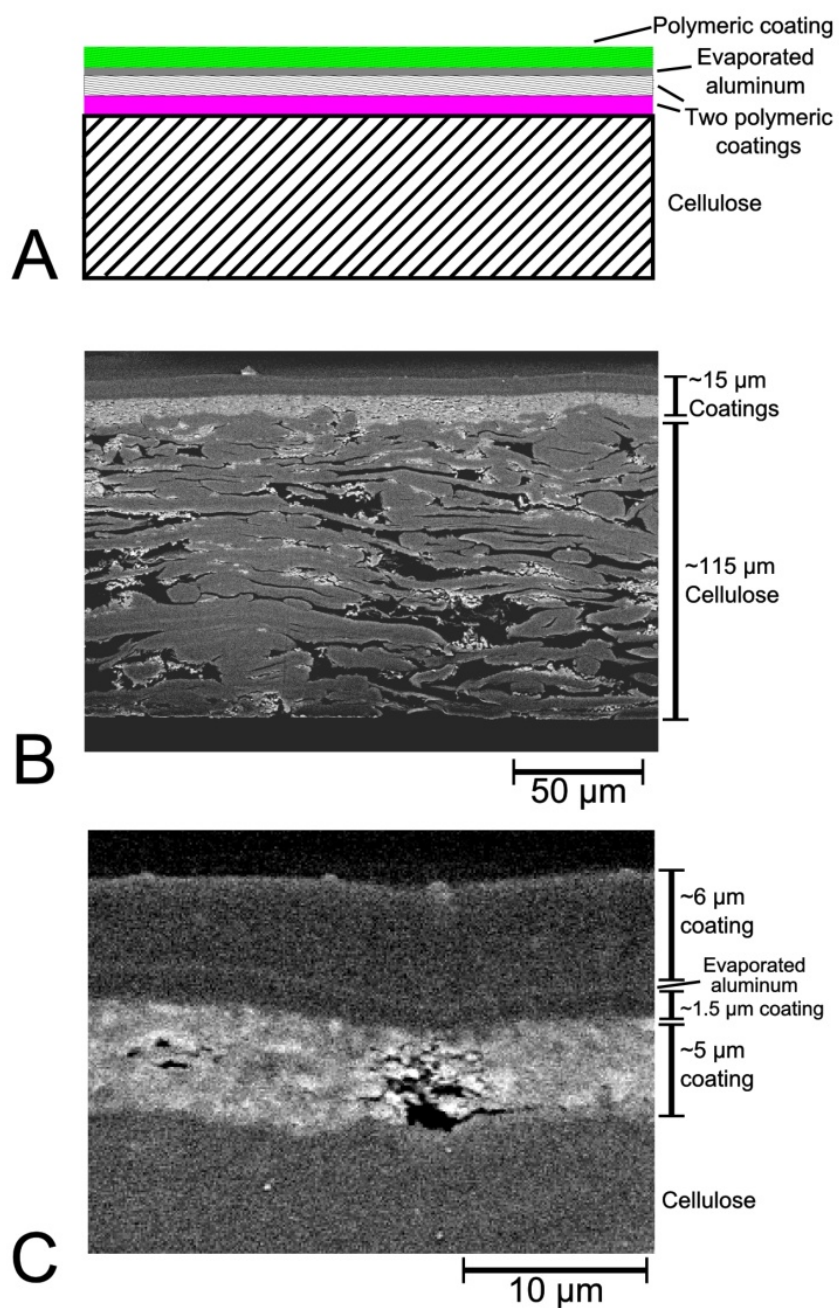
For the metallized/aluminized paper used in this work, we measured resistances along pieces with a length of 50 mm and widths ranging from 2 to 25 mm. For the widest set (width of 25 mm) of 7 separate pieces, we measured a resistance of $5 \pm 0.6 \Omega$ (mean \pm 1 standard deviation). For the thinnest set of 7 separate pieces, we measured a resistance of $58 \pm 7 \Omega$ (see SI Figure 2). Using a resistivity of $26 \text{ n}\Omega \cdot \text{m}$ for aluminum, the average calculated thickness across all 49 samples was $13 \pm 2 \text{ nm}$.

Etching and Cutting Components

The process of assembling a touch pad consisted of cutting the individual layers from metallized paper, double-sided tape, and if desired, additional paper for structural support. For this work, we used a VLS3.50 laser cutter (50-watt laser) from Universal Laser Systems with the standard 2.0-in. lens. According to its user guide, the laser cutter has x-y resolution up to 1000 dpi ($25 \mu\text{m}$), but the spot size of the beam used to ablate material is much larger and depends on the selected lens (three available options). The 2.0-in. lens we typically use has a nominal spot size of 1.8 mm, but another lens can have a spot size of approximately 0.4 mm. For the initial prototyping of buttons shown in this work, the resolution of the laser cutter is more than sufficient, and we have not attempted to minimize the size of the buttons themselves.

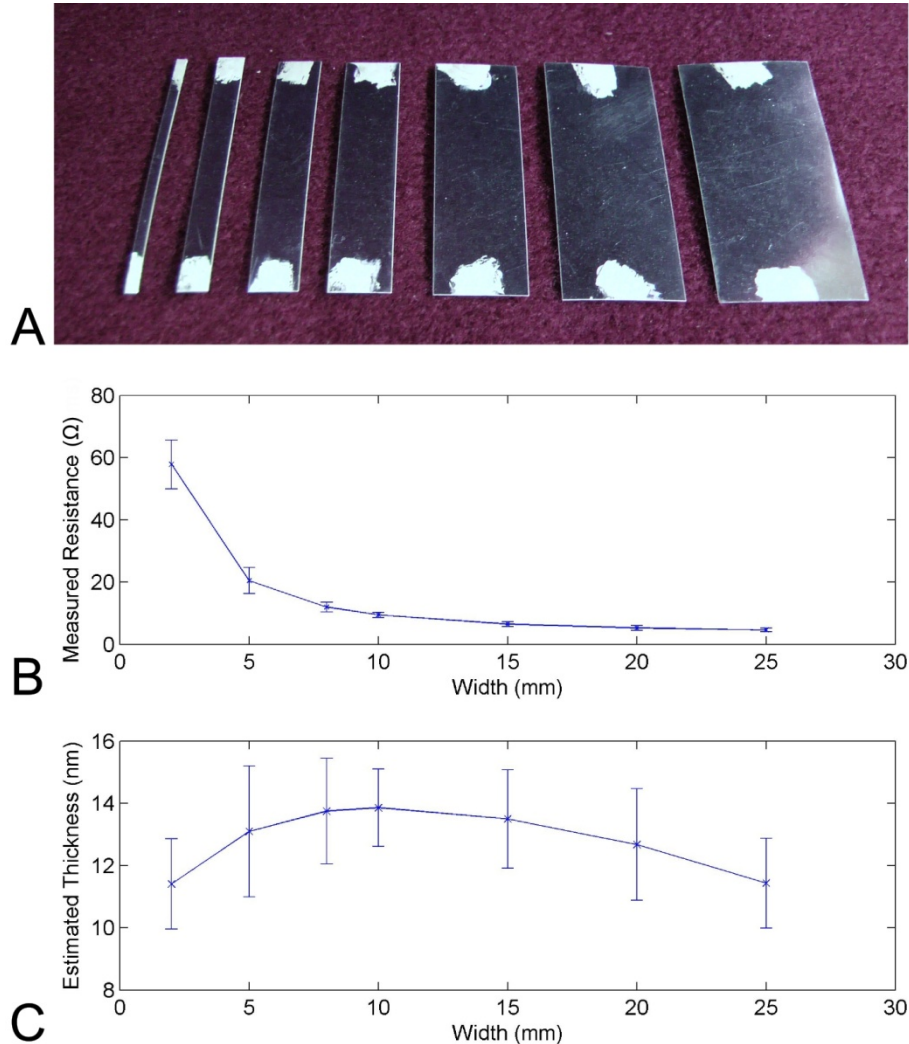
To array buttons on a single sheet of metallized paper, the laser cutter ablated lines in the thicker Vacumet A-550 metallized paper without cutting completely through the paper to form conductive traces or separate regions of conductivity on the paper. With the thin metallized paper (Vacumet A-238) and non-uniformities in the cutting power of the laser cutter, it was difficult to avoid cutting through the entire piece of paper in certain regions while still removing the aluminum. In some cases, ablation at low power left perforated lines through the paper.

SI Figure 1



SI Figure 1: Morphology of metallized paper from Vacumet Corporation. (A) The metallized paper consists of cellulose with 3 polymeric coatings and 1 layer of evaporated aluminum. (B) SEM image showing the entire thickness of a cross-sectional slice of metallized paper (Vacumet A-550). (C) SEM image showing a cross-sectional portion of Vacumet A-550. The layer of evaporated aluminum (faint line) is much thinner than the polymeric layers (see SI Figure 2).

SI Figure 2



SI Figure 2: Measurements of the resistances and estimated thicknesses of cut samples of metallized paper. (A) Image of one set of samples with a length of 50 mm and varied widths of 2, 5, 8, 10, 15, 20, and 25 mm. The spots at each end of the samples are regions of silver-based conductive paste. (B) Averaged resistances (BK Precision 5370) of seven different samples for each specified width with the error bars showing ± 1 standard deviation. (C) Estimated thicknesses for the evaporated aluminum, where the thickness is $h = \rho L / (wR)$, ρ is the resistivity of the aluminum ($26 \text{ n}\Omega \cdot \text{m}$), L is the length of the samples (50 mm), w is the width of the samples, and R is the measured resistance. The number of samples is 7 ($n=7$) at each point shown on the graph. The averaged, estimated thickness of all the samples ($n=49$) is 13 nm with a standard deviation of 2 nm.

Cutting through the thickness of the thin paper was not usually an issue for fabricating buttons when using double-sided tape to adhere the patterned metallized paper to another surface.

We cut through Vacumet A-238 (aluminized paper with a thickness of 56 μm) with the settings of 12% power, 80% speed, and 500 pulses per inch. To ablate through the conductive layer without cutting out parts completely, we used 3.5%, 80% speed, and 500 pulses per inch on the laser cutter or 3% power, 70% speed, and 500 pulses per inch. To cut through the Vacumet A-550 (thickness of 136 μm), we used settings of 15% power, 80% speed, and 500 pulses per inch. We found settings of 6% power, 80% speed, and 500 pulses per inch adequate for cutting traces in the paper without cutting completely through the paper.

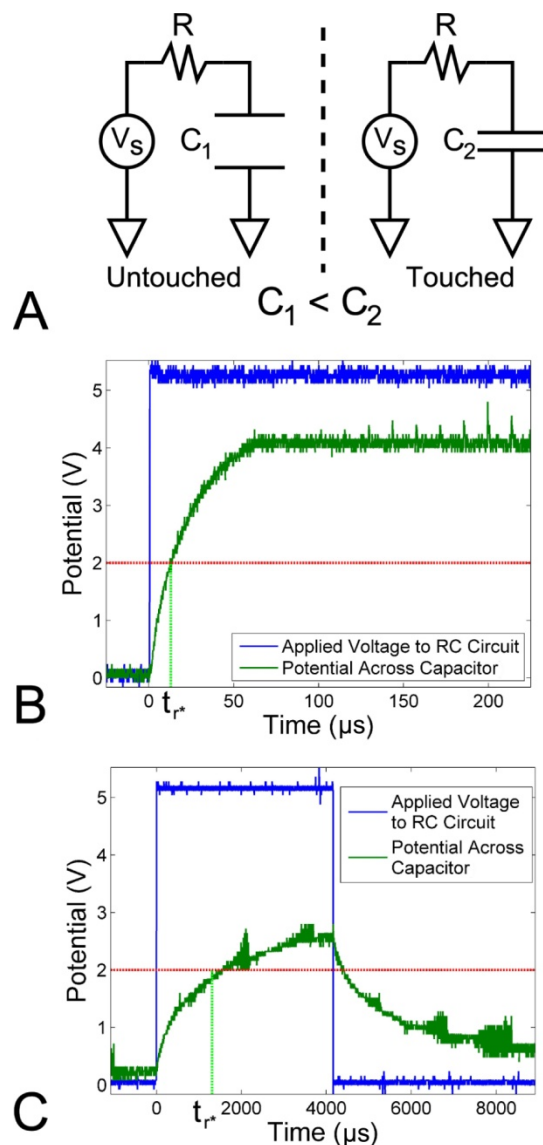
Interfacing to external electronics

We brushed on conductive silver paste (Silver Conductive Adhesive 503 from Electron Microscopy Sciences) to connect traces or tabs on the metallized paper electrically with external electronics through conductive pads or wired leads. Initially, we scraped away or dissolved a portion of the polymeric coating above the evaporated aluminum with a razor blade or 2 μL of acetone. Nevertheless, we found that the solvent in the conductive adhesive dissolved through the insulating coating sufficiently to form an electrically conductive bond.

Thresholds for registering touch

The Arduino-based systems calculated delays or elapsed times and compared them with temporal thresholds to determine changes in effective capacitance. To measure capacitance in terms of standard units (Farads), we estimated an effective time constant t_{r*} , which corresponded to the time required for the voltage to reach 2 V out of the applied 5 V. For an ideal RC circuit (SI Figure 3A), $t_{r*} = \ln(5/3)RC$ or $0.51RC$, where R and C represent the resistance and capacitance, respectively. Notably, the measured potential with an oscilloscope – not used to acquire any data shown in the main text – were not ideal (SI Figures 3B and C) possibly because

SI Figure 3



SI Figure 3: (A) The capacitance of a button (type shown in Figure 2E) increases when touched. To measure the change in capacitance of a button, we used a resistor in series with the capacitive button. (B) While applying a step in potential across the resistor-capacitor (RC) circuit, we observed the potential across the capacitive button increase with a time constant equivalent to the product of the resistance and capacitance in the circuit (data shown is from the button in Figure 2 E). The required amount of time for the capacitor to charge to 2 V (threshold for Arduino’s input to go from 0 to 1) required time t_{r^*} . For the shown measurement of the untouched button, $t_{r^*} = 13 \mu\text{s}$. (C) When touched, the capacitance across the button increased and $t_{r^*} = 1300 \mu\text{s}$.

of the complex nature of the impedance of the contact between the finger and the button, the impedance through or along the finger itself, or the finger-body-ground impedance. For Figure 4 C, the thresholds were 95 μ s for key 0, 97 μ s for key 1, 95 μ s for key 2, 102 μ s for key 3, 106 μ s for key 4, 100 μ s for key 6, 100 μ s for key 6, 106 μ s for key 7, 106 μ s for key 8, and 100 μ s for key 9. In other versions of the programmed Arduino (e.g., Figure 4 B, Figure 5, and Figure 6), we implemented an averaging routine (see <http://www.arduino.cc/en/Tutorial/Smoothing>) to adjust thresholds in real-time based on measurements of effective capacitance of the untouched keys.

Measurements on single buttons

We constructed the buttons shown in Figure 2; one button consisted of parallel plates (Figure 2 A and B), while the other button consisted of a single layer of metallized paper with etched traces (Figure 2 D and E). The material for the top layer of Figure 2 B and the entire device shown in Figure 2 E was the thin metallized paper (Vacumet A-238), while the bottom layer of Figure 2 B was the thicker metallized paper (Vacumet A-550). To test the devices, we mounted them to a manila folder with double-sided tape.

To measure the changing capacitance of the devices shown in Figure 2 C and F, we placed a resistor of 1.01 M Ω (measured with R.S.R 308B Multimeter at 21°C, 50% RH) in series with the capacitive button. Using Arduino, it was possible to calculate the required amounts of time for the potential across the capacitor to reach 2 V. Each point in the graphs is the mean of the effective capacitance of the button calculated over five seconds of sampling.

For measurements on two-layered devices (22 °C, 50 % RH), seven measurements without being touched had a mean capacitance of 206 pF and a standard deviation of 15 pF for n=4746 (n is the number of samples taken during the five seconds of sampling). Seven measurements while

touched with a bare finger had a mean capacitance of 331 pF and a standard deviation of 33 pF for $n=4634$.

For measurements on a button with a single layer of metallized paper (22 °C, 49% RH), seven measurements without being touched had a mean capacitance of 43.8 pF and a standard deviation of 4.1 pF for $n=5564$. Seven measurements while touched with a bare finger had a mean capacitance of 2120 pF and a standard deviation of 520 pF for $n=3837$.

To make an approximate comparison for the sensitivities between the two-layered button and the button made from a single layer of metallized paper, we calculated an upper limit for the capacitance of an untouched button (mean capacitance plus 3X the standard deviation of the measurements) with a lower limit for the capacitance of a touched button (mean capacitance minus 3X the standard deviation of the measurements). For the two-layered button, we found the upper limit of 251 pF ($206 \text{ pF} + 3 \times 15 \text{ pF}$) for the untouched button to be greater than the lower limit of 238 pF ($331 - 3 \times 17 \text{ pF}$) for the touched button. For the button fabricated with inter-digitated electrodes on a single piece of metallized paper, we found upper limit of 56.1 pF ($43.8 \text{ pF} + 3 \times 4.1 \text{ pF}$) for the untouched button to be less than the lower limit of 560 pF ($2120 \text{ pF} - 3 \times 520 \text{ pF}$) for the touched button. Thus, this comparison suggested that the touch pads with a single layer of metallized paper and inter-digitated electrodes/buttons were more sensitive than the buttons fabricated using two layers of metallized paper.

Consumption of Power

For one of our systems with an Arduino Uno, we measured a draw of approximately 0.4 W (50 mA of current at 8 V of potential) to all of the electronics. We also noticed an increase in draw of a few milliamps when one of the touched buttons triggered an LED (the LED was in series with a resistor of 1 k Ω and had an applied potential of 5V). A standard 9-V battery holds

about 16.2 kJ of energy (500 mAh at 9 V) and could handle a draw of 9 V at 0.4 W for approximately 11 hours.

With more efficient electronics, the touch pads could last for much longer periods of time with a standard 9-V battery. The potential across the resistor in an RC-circuit in response to a step input of potential V_s has the following relationship:

$$V_R = V_s e^{-t/\tau},$$

where t is time and τ is the time constant RC . Using the expression for power (V_R^2/R) and integrating with respect to time, the energy dissipated in the resistor in response to a step input in potential V_s is the following:

$$E_R = \frac{CV_s^2}{2},$$

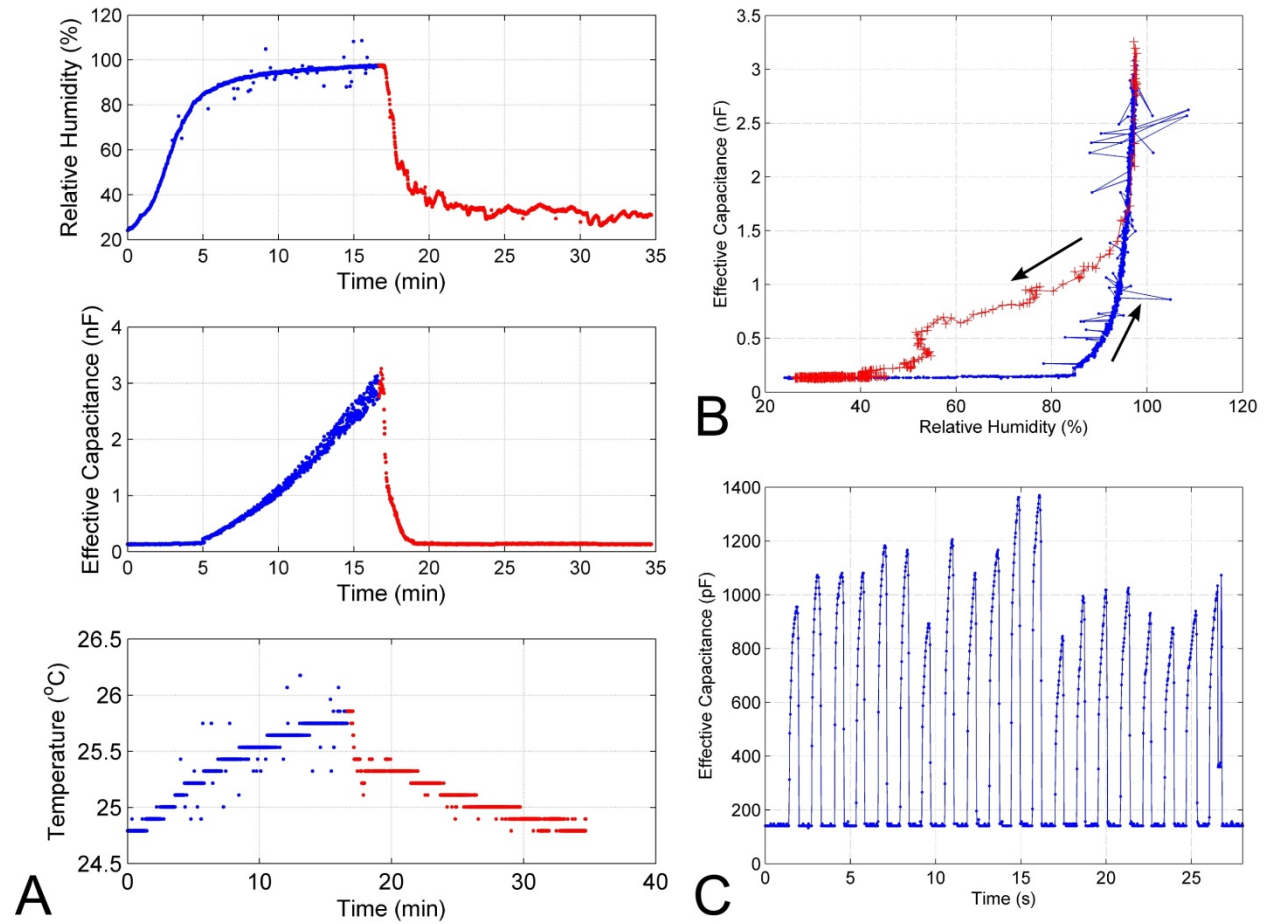
or the energy stored in the capacitor of the RC-circuit. For a resistor in series with a capacitor of 100 pF (approximate value for one of our buttons) and a step input in potential of 5 V, the energy dissipated in response to a single step input upon charging the capacitor is 1.25 nJ. Because the resistor dissipates energy for both the charging and discharging of the capacitor, the dissipated energy for a complete cycle of a square wave-like input is 2.5 nJ. If we sample the button 100 times a second, the dissipated power is 250 nW per button. With 10 buttons, we would dissipate 2.5 μ W, which is about 5 orders of magnitude less than the power consumed by the Arduino-based system. In reality, the electronics required to power the buttons will require more power than these ideal estimates. Nevertheless, there is ample room for improvement even by just reducing the magnitude of the applied potential.

Environmental Conditions

To test the performance of a paper-based touch pad subject to changes in humidity, we placed a pot of heated water ($\sim 57^\circ\text{C}$) in a clear, plastic container with the 10-button keypad shown in SI Video 1. We then used the serial output from an Arduino Uno to record the temperature with a voltage divider and a thermistor (RL0503-17.56K-96-MS from GE Sensing and Inspection Technologies), relative humidity with a breakout board from Sparkfun (HIH-4030 with a response time of 5 s from Honeywell), and the effective capacitance of button "0" on the keypad using the method described in SI Figure 3. SI Figure 4 A shows the resulting measurements plotted against time. Once the water-filled pot entered the chamber, the relative humidity rose from approximately 24% to 98% in approximately 17 minutes. After 17 minutes, we opened the container to let moisture escape, and the relative humidity within the chamber decreased to 31%. During the entire process the change in temperature within the chamber was less than 2°C . To show that the touch pads can function in cold climates, SI Figure 4 C shows the effective capacitance of a button being pressed 20 times in a cold room with a temperature of 4°C and a relative humidity of 50%.

The effective capacitance of the untouched button remained steady until the relative humidity reached approximately 85%. Then, the effective capacitance rose as shown in SI Figure 4 B along the path with the arrow pointing upwards and to the right. It is possible that the increase in effective capacitance was a result of moisture accumulating on the surface of the button or within the paper comprising the button because visible moisture accumulated on the walls of the plastic container, the effective capacitance appeared to continue to rise even after humidity was leveling off near 98%, and the effective capacitance exhibited a hysteretic behavior as the relative humidity decreased with the opening of the chamber. The hysteretic behavior

SI Figure 4



SI Figure 4: (A) Plots of the relative humidity, effective capacitance of a button, and temperature in a chamber versus time. (B) The effective capacitance of a button plotted against relative humidity for the environmental conditions in A. (C) The measured effective capacitance of a button being touched with a bare finger in a cold room with a temperature of 4 °C and relative humidity of 50%.

shown in Figure 4 B suggests the paper needed time to dry before the effective capacitance returned to its initial value. The moisture may have also served to reduce the resistance between the electrodes and cause an increase in the effective time constant.

To observe the effect of moisture on the resistance of a button, we used an LCR meter (B&K Precision Model 885) to measure the parallel capacitance and resistance of a button exposed to moisture spouting from an ultrasonic humidifier (Vicks V5100NS). Before exposure to the moisture, the measured parallel capacitance and resistance were 13 pF and 250 M Ω at a measured frequency of 1 kHz. Once exposed to moisture, the measured capacitance increased to 100 pF, while the resistance in parallel with the capacitance decreased to 700 k Ω . After shutting off the humidifier, the button dried within a few minutes, and the capacitance decreased to 15 pF with a resistance of 100 M Ω .

In addition to varying humidity as shown in SI Figure 4 A and B, we placed the same keypad in an oven to measure effective capacitance as a function of temperature. While going from 25 °C to 57 °C over a period of 40 minutes with a reduction in relative humidity from 8% to 1%, the mean effective capacitance was 141 pF with a standard deviation of 2.3 pF.

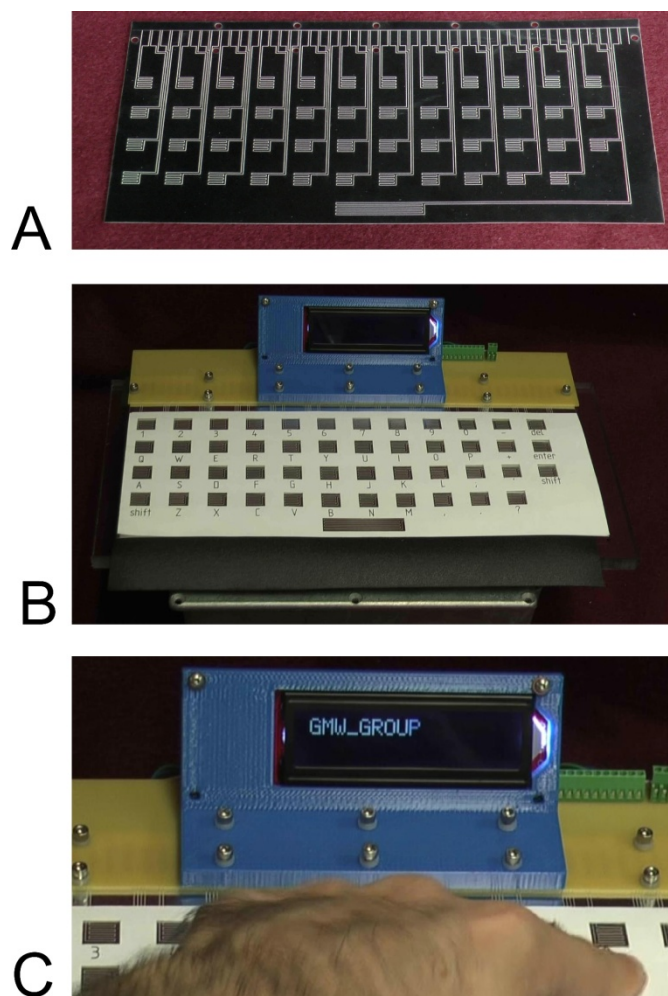
Cost

Thin metallized paper costs less than US \$0.25 per square meter, and thicker metallized paper costs less than US \$0.75 per square meter. The 10-button keypad requires between 2-3 minutes of time on the laser cutter. An estimate for the mass of conductive paste per contact is 10 mg, and the conductive paste costs US \$39 per 15 g. Thus, the conductive paste cost about US \$0.03 per button. A cheaper alternative might be carbon ink. An estimated cost of the thicker metallized paper for the 10-button keypad (0.01 m²) is US \$0.08, and an estimate cost for the

conductive paste is \$0.33 (ten buttons but 11 contacts including electrical ground). Thus, the cost of materials for the 10-button keypad is less than \$0.50.

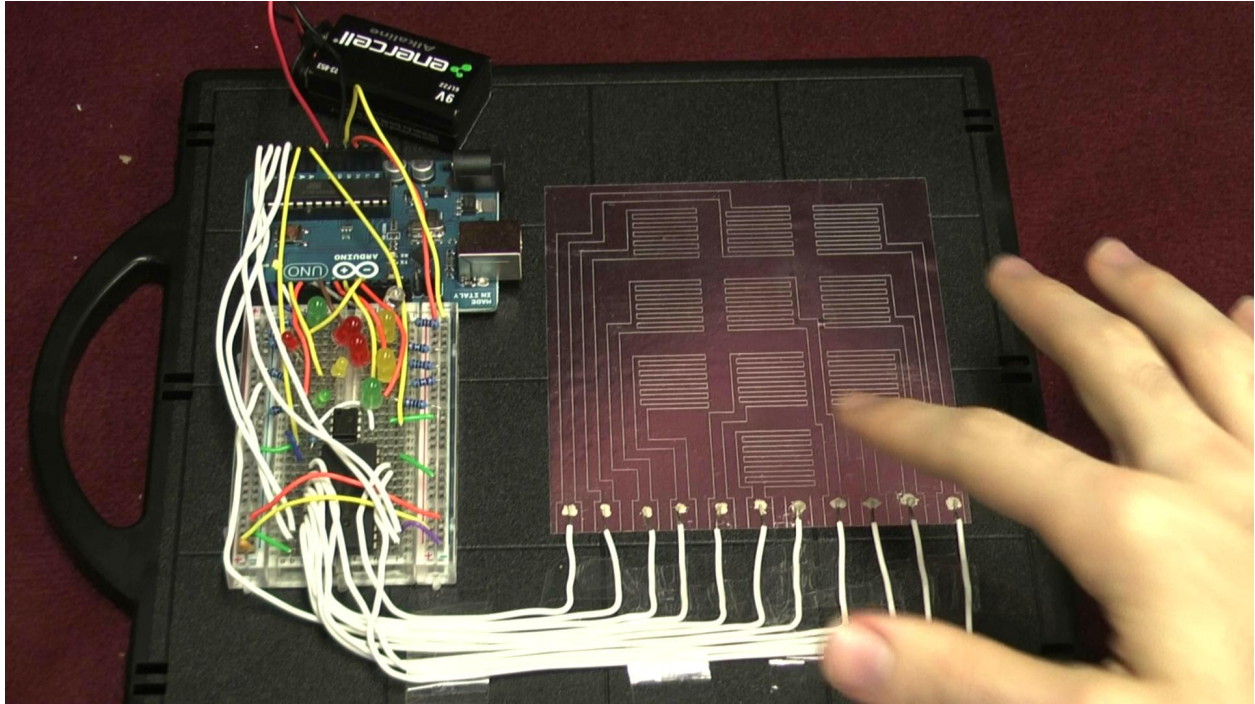
The Arduino costs about US \$25 in its pre-made form, but less user-friendly controllers could cost less. The breadboard costs about US \$7. The operational amplifier costs less than US \$3 each in small quantities. The software for programming the Arduino is open-source (free). The buzzer for the alarmed box costs less than US \$4. The total cost of the reusable electronics is less than US \$50, and the electronic components could cost much less with appropriate design for manufacturing.

SI Figure 5



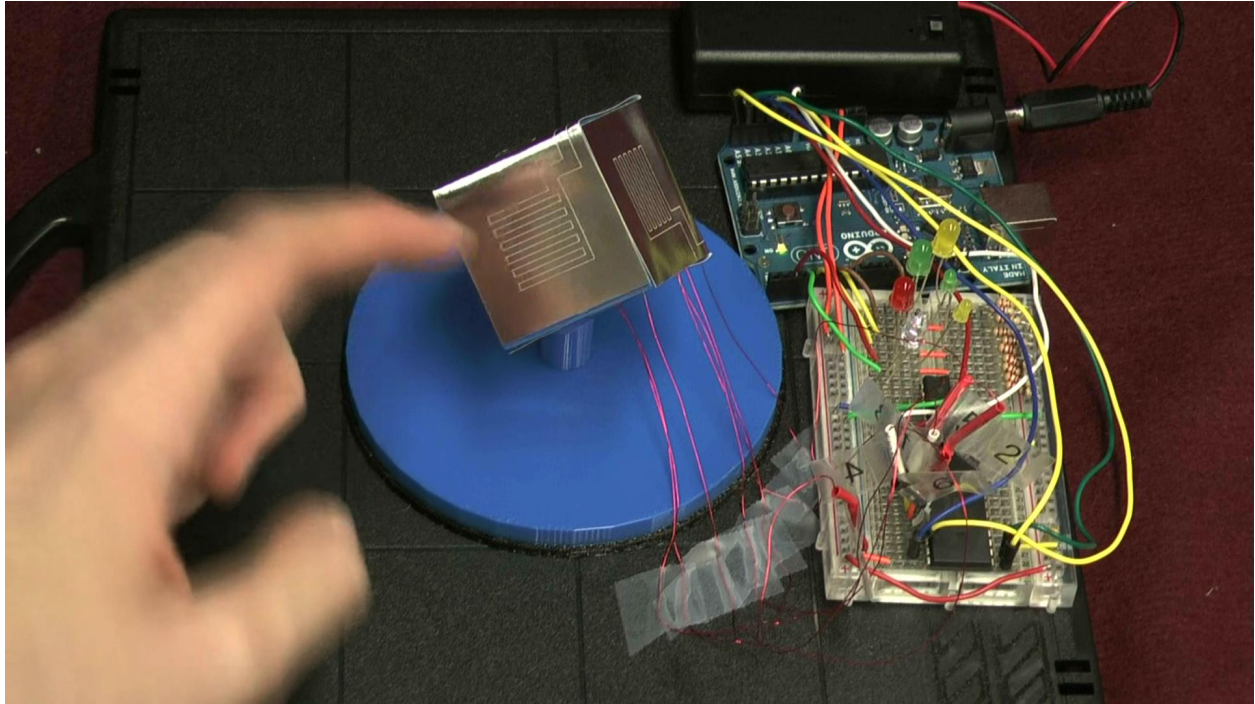
SI Figure 5: A functional QWERTY-based keyboard with 48 keys. (A) Metallized paper with etched traces and buttons of the QWERTY-based keyboard before its assembly. Individual traces (etched with the laser cutter) lead to a button for each key. The width of the keyboard was approximately 25 cm wide and 10 cm tall (not including the extra printed circuit board). (B) The assembled keyboard with a patterned piece of paper designating the location of each key and also allowing contact between the metallized finger and a bare finger. Below the metallized paper is foam, acrylic, and an additional printed circuit board with an Arduino board, multiplexers, and other appropriate electronics. (C) The user after just having finished typing “GMW Group” (see SI Video 4).

SI Video 1



SI Video 1: This video shows four different clips showing a functional 10-button keypad. The first two clips show bare fingers touching keys, while the last two clips show gloved fingers touching the keys. Each key addresses an individual LED. Simultaneous touching of multiple keys lights multiple LEDs. The “ThresholdFactor” (variable in the code included at the end of the Supporting Information) had a value of 1.12 for all four clips.

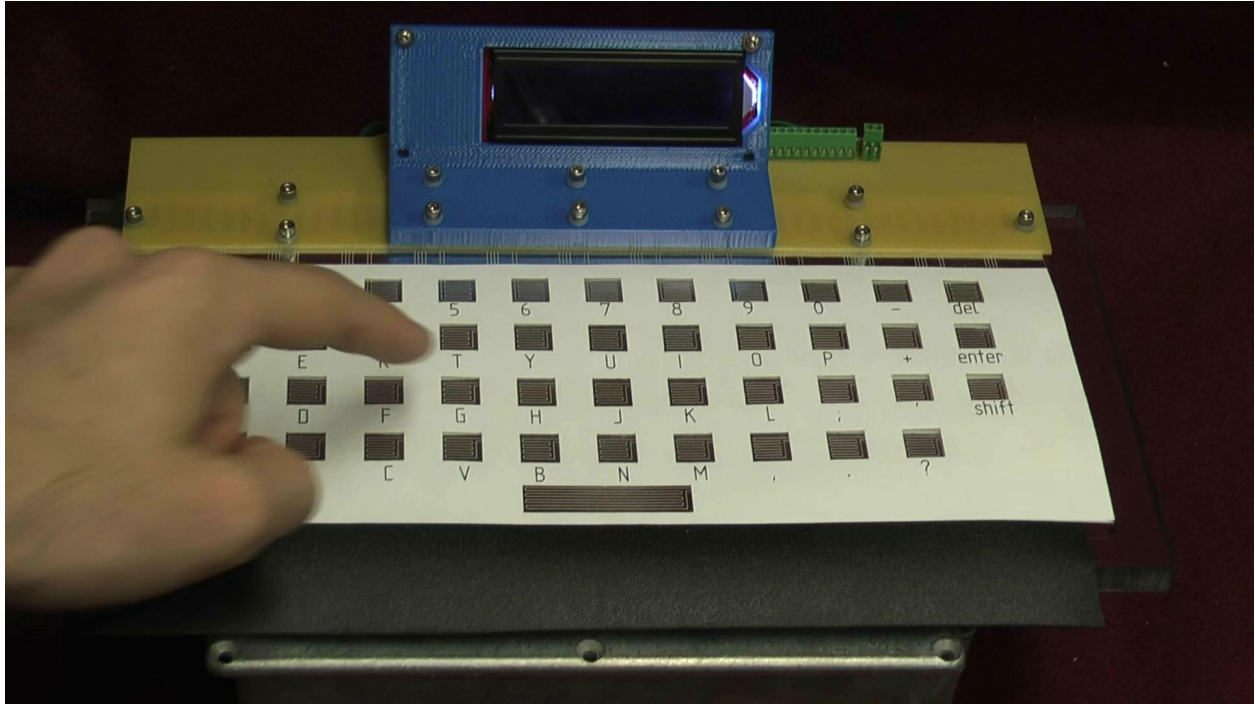
SI Video 2



SI Video 2: This video shows four different clips showing a functional 6-button keypad folded around a cube. The first two clips show bare fingers touching keys, while the last two clips show gloved fingers touching the keys. Each key/face addresses an individual LED. Simultaneous touching of multiple keys lights multiple LEDs. Within the same code used for SI Video 1, the “ThresholdFactor” (variable in the code included at the end of the Supporting Information) had a value of 1.13 for all four clips.

SI Video 3

SI Video 3: Video of the alarmed box described in Figure 6. The box is in an unarmed state initially, so that opening it does not set off an alarm. Touching the keypad then arms the box, and the alarm goes off with the opening of the lids. By typing the password into the keypad, the box returns to its unarmed state, and it is possible to open the lids again without setting off the alarm.

SI Video 4

SI Video 4: Video of the functioning keyboard described in SI Figure 5. The user types in the phrase “GMW Group” and then proceeds to demonstrate the functionality of every key.

Arduino Code for SI Video 1

```
/*Adapted from
http://arduino.cc/it/Tutorial/CapacitanceMeter and
http://www.arduino.cc/en/Tutorial/Smoothing*/
#define Output A1
#define Input1 A0
#define MultiplexA A2
#define MultiplexB A3
#define MultiplexC A4
#define MultiplexD A5
#define LED0 2
#define LED1 3
#define LED2 4
#define LED3 5
#define LED4 6
#define LED5 7
#define LED6 8
#define LED7 9
#define LED8 10
#define LED9 11

unsigned long StartTime; //Time when step output goes
high
unsigned long ElapsedTime0; //Length of time
(microseconds)
//required for RC circuit to reach 2 V.
unsigned long ElapsedTime1;
unsigned long ElapsedTime2;
unsigned long ElapsedTime3;
unsigned long ElapsedTime4;
unsigned long ElapsedTime5;
unsigned long ElapsedTime6;
unsigned long ElapsedTime7;
unsigned long ElapsedTime8;
unsigned long ElapsedTime9;

unsigned long TotalStartTime; //Time when measurments
begin

const int NumReadings=30;
const int SkipCountsForSmoothing=5;
int AverageNow=0;
long int Counter=0;

const float ThresholdFactor=1.12;
int Index0=0;
int Index1=0;
int Index2=0;
int Index3=0;
int Index4=0;
int Index5=0;
int Index6=0;
int Index7=0;
int Index8=0;
int Index9=0;

int Threshold0=95; //Starting Values for Moving Averaged
//Thresholds in microseconds
int Threshold1=97;
int Threshold2=95;
int Threshold3=102;
int Threshold4=106;
```

```
int Threshold5=100;
int Threshold6=100;
int Threshold7=106;
int Threshold8=106;
int Threshold9=100;

int Thresholds0[NumReadings]; //Array of the Moving
Averages
int Thresholds1[NumReadings]; //Array of the Moving
Averages
int Thresholds2[NumReadings]; //Array of the Moving
Averages
int Thresholds3[NumReadings]; //Array of the Moving
Averages
int Thresholds4[NumReadings]; //Array of the Moving
Averages
int Thresholds5[NumReadings]; //Array of the Moving
Averages
int Thresholds6[NumReadings]; //Array of the Moving
Averages
int Thresholds7[NumReadings]; //Array of the Moving
Averages
int Thresholds8[NumReadings]; //Array of the Moving
Averages
int Thresholds9[NumReadings]; //Array of the Moving
Averages

int ThresholdsTotal0=NumReadings*Threshold0;
//Running Total for Moving Averages
int ThresholdsTotal1=NumReadings*Threshold1;
int ThresholdsTotal2=NumReadings*Threshold2;
int ThresholdsTotal3=NumReadings*Threshold3;
int ThresholdsTotal4=NumReadings*Threshold4;
int ThresholdsTotal5=NumReadings*Threshold5;
int ThresholdsTotal6=NumReadings*Threshold6;
int ThresholdsTotal7=NumReadings*Threshold7;
int ThresholdsTotal8=NumReadings*Threshold8;
int ThresholdsTotal9=NumReadings*Threshold9;

void setup()
{
  pinMode(Output, OUTPUT);
  pinMode(Input1, INPUT);
  pinMode(MultiplexA, OUTPUT);
  pinMode(MultiplexB, OUTPUT);
  pinMode(MultiplexC, OUTPUT);
  pinMode(MultiplexD, OUTPUT);
  pinMode(LED0,OUTPUT);
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(LED4,OUTPUT);
  pinMode(LED5,OUTPUT);
  pinMode(LED6,OUTPUT);
  pinMode(LED7,OUTPUT);
  pinMode(LED8,OUTPUT);
  pinMode(LED9,OUTPUT);
  digitalWrite(MultiplexA,LOW);
  digitalWrite(MultiplexB,LOW);
  digitalWrite(MultiplexC,LOW);
  digitalWrite(MultiplexD,LOW);
  digitalWrite(LED0,LOW);
```

```
digitalWrite(LED1,LOW);
digitalWrite(LED2,LOW);
digitalWrite(LED3,LOW);
digitalWrite(LED4,LOW);
digitalWrite(LED5,LOW);
digitalWrite(LED6,LOW);
digitalWrite(LED7,LOW);
digitalWrite(LED8,LOW);
digitalWrite(LED9,LOW);
for (int ThisReading = 0; ThisReading < NumReadings;
ThisReading++){
  Thresholds0[ThisReading] = Threshold0;
  Thresholds1[ThisReading] = Threshold1;
  Thresholds2[ThisReading] = Threshold2;
  Thresholds3[ThisReading] = Threshold3;
  Thresholds4[ThisReading] = Threshold4;
  Thresholds5[ThisReading] = Threshold5;
  Thresholds6[ThisReading] = Threshold6;
  Thresholds7[ThisReading] = Threshold7;
  Thresholds8[ThisReading] = Threshold8;
  Thresholds9[ThisReading] = Threshold9;
}
Serial.begin(9600);
Serial.println("System for Measuring Capacitance ");
Serial.println("Time to reach 2 V in microseconds");
TotalStartTime=millis();
}

void loop()
{
  //For Channel 0 on multiplexer (Button 0)
  digitalWrite(MultiplexA,LOW);
  digitalWrite(MultiplexB,LOW);
  digitalWrite(MultiplexC,LOW);
  digitalWrite(MultiplexD,LOW);
  delay(1);
  digitalWrite(Output,HIGH);
  StartTime=micros();
  while(digitalRead(Input1)<1){
  }
  ElapsedTime0=micros()-StartTime;
  digitalWrite(Output,LOW);
  delay(2);
  //For Channel 1 on multiplexer (Button 1)
  digitalWrite(MultiplexA,HIGH);
  digitalWrite(MultiplexB,LOW);
  digitalWrite(MultiplexC,LOW);
  digitalWrite(MultiplexD,LOW);
  delay(1);
  digitalWrite(Output,HIGH);
  StartTime=micros();
  while(digitalRead(Input1)<1){
  }
  ElapsedTime1=micros()-StartTime;
  digitalWrite(Output,LOW);
  delay(2);
  //For Channel 2 on multiplexer (Button 2)
  digitalWrite(MultiplexA,LOW);
  digitalWrite(MultiplexB,HIGH);
  digitalWrite(MultiplexC,LOW);
  digitalWrite(MultiplexD,LOW);
  delay(1);
```

```
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime2=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 3 on multiplexer (Button 3)
digitalWrite(MultiplexA,HIGH);
digitalWrite(MultiplexB,HIGH);
digitalWrite(MultiplexC,LOW);
digitalWrite(MultiplexD,LOW);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime3=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 4 on multiplexer (Button 4)
digitalWrite(MultiplexA,LOW);
digitalWrite(MultiplexB,LOW);
digitalWrite(MultiplexC,HIGH);
digitalWrite(MultiplexD,LOW);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime4=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 5 on multiplexer (Button 5)
digitalWrite(MultiplexA,HIGH);
digitalWrite(MultiplexB,LOW);
digitalWrite(MultiplexC,HIGH);
digitalWrite(MultiplexD,LOW);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime5=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 6 on multiplexer (Button 6)
digitalWrite(MultiplexA,LOW);
digitalWrite(MultiplexB,HIGH);
digitalWrite(MultiplexC,HIGH);
digitalWrite(MultiplexD,LOW);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime6=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 7 on multiplexer (Button 7)
digitalWrite(MultiplexA,HIGH);
digitalWrite(MultiplexB,HIGH);
```



```
digitalWrite(MultiplexC,HIGH);
digitalWrite(MultiplexD,LOW);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime7=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 8 on multiplexer (Button 8)
digitalWrite(MultiplexA,LOW);
digitalWrite(MultiplexB,LOW);
digitalWrite(MultiplexC,LOW);
digitalWrite(MultiplexD,HIGH);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime8=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//For Channel 9 on multiplexer (Button 9)
digitalWrite(MultiplexA,HIGH);
digitalWrite(MultiplexB,LOW);
digitalWrite(MultiplexC,LOW);
digitalWrite(MultiplexD,HIGH);
delay(1);
digitalWrite(Output,HIGH);
StartTime=micros();
while(digitalRead(Input1)<1){
}
ElapsedTime9=micros()-StartTime;
digitalWrite(Output,LOW);
delay(2);
//Print Results
/*Serial.print(millis()-TotalStartTime);
Serial.print(" ");
Serial.print(ElapsedTime0,DEC);
Serial.print(" ");
Serial.print(ElapsedTime1,DEC);
Serial.print(" ");
Serial.print(ElapsedTime2,DEC);
Serial.print(" ");
Serial.print(ElapsedTime3,DEC);
Serial.print(" ");
Serial.print(ElapsedTime4,DEC);
Serial.print(" ");
Serial.print(ElapsedTime5,DEC);
Serial.print(" ");
Serial.print(ElapsedTime6,DEC);
Serial.print(" ");
Serial.print(ElapsedTime7,DEC);
Serial.print(" ");
Serial.print(ElapsedTime8,DEC);
Serial.print(" ");
Serial.print(ElapsedTime9,DEC);
Serial.print(" ");*/
//Turn on appropriate LED(s)
AverageNow=Counter%SkipCountsForSmoothing;
if(ElapsedTime0>Threshold0*ThresholdFactor){
```

```
digitalWrite(LED0,HIGH);
//Serial.print(" ");
//Serial.print("1");
}
else{
digitalWrite(LED0,LOW);
//Serial.print(" ");
//Serial.print("0");
if(AverageNow==0){
ThresholdsTotal0=ThresholdsTotal0-
Thresholds0[Index0];
Thresholds0[Index0]=ElapsedTime0;
ThresholdsTotal0=ThresholdsTotal0+ElapsedTime0;
Threshold0=ThresholdsTotal0/NumReadings;
Index0=Index0+1;
if(Index0>=NumReadings){
Index0=0;
}
}
}
if(ElapsedTime1>Threshold1*ThresholdFactor){
digitalWrite(LED1,HIGH);
//Serial.print(" ");
//Serial.print("1");
}
else{
digitalWrite(LED1,LOW);
//Serial.print(" ");
//Serial.print("0");
if(AverageNow==0){
ThresholdsTotal1=ThresholdsTotal1-
Thresholds1[Index1];
Thresholds1[Index1]=ElapsedTime1;
ThresholdsTotal1=ThresholdsTotal1+ElapsedTime1;
Threshold1=ThresholdsTotal1/NumReadings;
Index1=Index1+1;
if(Index1>=NumReadings){
Index1=0;
}
}
}
if(ElapsedTime2>Threshold2*ThresholdFactor){
digitalWrite(LED2,HIGH);
//Serial.print(" ");
//Serial.print("1");
}
else{
digitalWrite(LED2,LOW);
//Serial.print(" ");
//Serial.print("0");
if(AverageNow==0){
ThresholdsTotal2=ThresholdsTotal2-
Thresholds2[Index2];
Thresholds2[Index2]=ElapsedTime2;
ThresholdsTotal2=ThresholdsTotal2+ElapsedTime2;
Threshold2=ThresholdsTotal2/NumReadings;
Index2=Index2+1;
if(Index2>=NumReadings){
Index2=0;
}
}
}
}
```

```
if(ElapsedTime3>Threshold3*ThresholdFactor){
  digitalWrite(LED3,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED3,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal3=ThresholdsTotal3-
Thresholds3[Index3];
    Thresholds3[Index3]=ElapsedTime3;
    ThresholdsTotal3=ThresholdsTotal3+ElapsedTime3;
    Threshold3=ThresholdsTotal3/NumReadings;
    Index3=Index3+1;
    if(Index3>=NumReadings){
      Index3=0;
    }
  }
}
if(ElapsedTime4>Threshold4*ThresholdFactor){
  digitalWrite(LED4,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED4,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal4=ThresholdsTotal4-
Thresholds4[Index4];
    Thresholds4[Index4]=ElapsedTime4;
    ThresholdsTotal4=ThresholdsTotal4+ElapsedTime4;
    Threshold4=ThresholdsTotal4/NumReadings;
    Index4=Index4+1;
    if(Index4>=NumReadings){
      Index4=0;
    }
  }
}
if(ElapsedTime5>Threshold5*ThresholdFactor){
  digitalWrite(LED5,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED5,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal5=ThresholdsTotal5-
Thresholds5[Index5];
    Thresholds5[Index5]=ElapsedTime5;
    ThresholdsTotal5=ThresholdsTotal5+ElapsedTime5;
    Threshold5=ThresholdsTotal5/NumReadings;
    Index5=Index5+1;
    if(Index5>=NumReadings){
      Index5=0;
    }
  }
}
```

```
}
if(ElapsedTime6>Threshold6*ThresholdFactor){
  digitalWrite(LED6,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED6,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal6=ThresholdsTotal6-
Thresholds6[Index6];
    Thresholds6[Index6]=ElapsedTime6;
    ThresholdsTotal6=ThresholdsTotal6+ElapsedTime6;
    Threshold6=ThresholdsTotal6/NumReadings;
    Index6=Index6+1;
    if(Index6>=NumReadings){
      Index6=0;
    }
  }
}
if(ElapsedTime7>Threshold7*ThresholdFactor){
  digitalWrite(LED7,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED7,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal7=ThresholdsTotal7-
Thresholds7[Index7];
    Thresholds7[Index7]=ElapsedTime7;
    ThresholdsTotal7=ThresholdsTotal7+ElapsedTime7;
    Threshold7=ThresholdsTotal7/NumReadings;
    Index7=Index7+1;
    if(Index7>=NumReadings){
      Index7=0;
    }
  }
}
if(ElapsedTime8>Threshold8*ThresholdFactor){
  digitalWrite(LED8,HIGH);
  //Serial.print(" ");
  //Serial.print("1");
}
else{
  digitalWrite(LED8,LOW);
  //Serial.print(" ");
  //Serial.print("0");
  if(AverageNow==0){
    ThresholdsTotal8=ThresholdsTotal8-
Thresholds8[Index8];
    Thresholds8[Index8]=ElapsedTime8;
    ThresholdsTotal8=ThresholdsTotal8+ElapsedTime8;
    Threshold8=ThresholdsTotal8/NumReadings;
    Index8=Index8+1;
    if(Index8>=NumReadings){
      Index8=0;
    }
  }
}
```

```
    }  
  }  
  if(ElapsedTime9>Threshold9*ThresholdFactor){  
    digitalWrite(LED9,HIGH);  
    //Serial.print(" ");  
    //Serial.println("1");  
  }  
  else{  
    digitalWrite(LED9,LOW);  
    //Serial.print(" ");  
    //Serial.println("0");  
    if(AverageNow==0){  
      ThresholdsTotal9=ThresholdsTotal9-  
Thresholds9[Index9];  
      Thresholds9[Index9]=ElapsedTime9;  
      ThresholdsTotal9=ThresholdsTotal9+ElapsedTime9;  
      Threshold9=ThresholdsTotal9/NumReadings;  
      Index9=Index9+1;  
      if(Index9>=NumReadings){  
        Index9=0;  
      }  
    }  
  }  
  Counter=Counter+1;
```