WireTouch: An Open Multi-Touch Tracker based on Mutual Capacitance Sensing

Armin Wagner
Vienna, Austria
wagner@wiretouch.net

Georg Kaindl
Vienna, Austria
kaindl@wiretouch.net

ABSTRACT

We present a fully disclosed tracking pipeline based on mutual capacitance sensing. We designed both hard- and software to support and foster experimentation. Without the use of vendor-supplied cap-sense ICs and proprietary software parts, every aspect and parameter of the user interface can be modified with ease. This flexibility gives the researcher a high degree of freedom while designing novel interactive systems. The working prototype of an audio-tactile multi-touch map is presented as an exemplary application.

Author Keywords

touch-sensitive interfaces, capacitive sensor, open-source hardware, prototyping

INTRODUCTION

Capacitive multi-touch sensing technology has matured in recent decades. Today's hardware vendors supply software development kits which abstract entirely from the raw tracking data acquired. Contemporary operating systems provide additional abstraction layers. Gesture recognition frameworks ease the integration of the current de facto standard of multi-touch interaction patterns. Supported by these frameworks, software developers and interaction designers are able to focus on their craft, without needing to fully understand the underlying technology.

This separation of concerns provides the current basis for the efficient production of multi-touch appliances on an industrial scale. But from an academic perspective this strategy has several drawbacks. First, gaining a holistic understanding of the system is hampered. Relevant parameters remain hidden. Interdependencies which span across the borders of the abstraction layers cannot be studied. What lies outside the current standard repertoire of multi-touch interaction patterns cannot be explored in an unrestricted manner. Secondly, there is little support for the construction of prototypical systems that diverge from the sizes, shapes, resolutions and tracking capabilities demanded by the current state of the mass market. In the worst case, basic functionalities have to be reimplemented or even reinvented from the ground up. This extra work might exceed the resources available to the researcher.

A common approach to this problem is the use of *hacked* off-the-shelf products as parts of prototypical systems. It allows the rapid construction of proof-of-concept demonstrations. But its wider use, especially in the scientific context, is limited. Hacking proprietary products is not only problematic because of legal matters. The use of *black boxes* as parts of prototypical systems limits the reproducibility of the system to the product-life span of its proprietary components. This intransparency can severely impede community driven research. Without access to the source code, the circuit diagrams and the documentation, follow-up research becomes dependent on ongoing product availability and long-term support. Thus promising research projects can come to a sudden halt when the trend driven market turns in a different direction and abandons system critical components.

We believe HCI research could benefit from the presentation of fully disclosed user interface systems. Access to the source code and blueprints can provide a solid basis for long-term community driven research. It helps to fulfil a basic requirement of scientific research: reproducibility. It fosters the designers understanding of the interdependencies of all involved components and design trade-offs. Moreover, if the potential modification is taken into account from the beginning, a fully disclosed system can provide the plasticity needed for rapid experimental prototyping.

We therefore present WireTouch, an experimental system based on openness and flexibility. We present a modular architecture. We give a detailed description of the technology, and we provide the information needed to build a complete system. We also showcase an exemplary implementation, demonstrating the versatility of the system and its potential value as an HCI prototyping platform.

What WireTouch is not, is a representation of current industry standards. WireTouch was not designed to represent current best practices or state-of-the-art optimizations, nor was it developed to stand in as an alternative, free version of commercially available consumer products. The main idea behind WireTouch is, to put an accessible tracking pipeline into the hands of interested researchers and designers, and to thereby foster experimentation.

PRIOR WORK

The usage of mutual capacitance sensors as input devices for computing devices has a rich history – full of sudden forward leaps, but also of dead-ends and discontinuities. Early publicly accessible research conducted at CERN in the 1970s [3]

September 4, 2016, Vienna, Austria.

http://dx.doi.org/10.5281/zenodo.61461

stopped soon after the collaboration with commercial partners started to require non-disclosure agreements [15]. Research on the experimental multi-touch panel presented at CHI '85 [8] was discontinued after being confronted with a superior prototype built by Bell Labs [5]. Details of this prototype never reached the broader scientific audience. Further basic research on capacitive tracking technology continued, but mainly behind closed curtains. Academia abandoned capacitive touch sensors and started to focus on alternative tracking technologies. "Given this long history of capacitive measurement", Zimmerman et al. remarked at CHI '95, "one might wonder why EF sensing is not common in human computer interfaces" [18]. But soon after, with access to new, inexpensive electronic components and computing power, the situation was about to change. One particularly promising project that emerged was SmartSkin, a sensor architecture presented by Rekimoto at CHI '02 [11]. SmartSkin demonstrated the feasibility of a responsive, flexible and scaleable multi-touch tracker, built on the basis of basic electronic parts. Unfortunately, the published documentation lacked the level of detail necessary for a complete reproduction. Hence, public follow up research on SmartSkin remained mostly limited to the near field of Sony CSL, where it had originated.

With the release of the Apple iPhone in 2007, high end capacitive sensing technology started to flood the mass market. Black boxed multi-touch modules became an integral part of a wide variety of consumer grade products. These developments were paralleled by two other relevant trends: a growing interest into Free and Open Source Software (FOSS) and physical computing. Projects such as Wiring [2] and Arduino not only showed how open soft- and hardware, paired with adequate licences, can foster community based research. They demonstrated the utility of a prototyping platform which bridges the gap between low- and high-level approaches in experimental user interface design practices. This approach brought forth several promising attempts to leverage capacitive multi-touch sensing techniques in a similar manner [12][9]. These projects tried to extend the prototyping toolkits available to interaction designers. They attempted to expand the explorable space beyond the limits of off-the-shelf products.

These solutions, however, were still limited to *self-capacitance sensing* techniques and only supported a very small number of sensor points. For certain prototypes this might be acceptable or even preferable. In other cases, access to more sophisticated sensing methods, or a combination of several sensing modes, might be crucial. Moreover, the source code and schematics were not fully disclosed. This shortfall might not be the result of a consciousness decision of the authors against openness. It rather represents the current modus operandi of the academic HCI discourse and ecosystem – a situation that will hopefully change in the near future.[1]

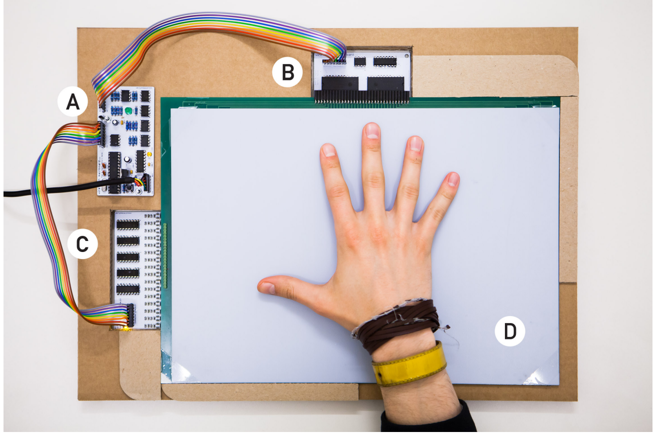


Figure 1. An exemplary WireTouch setup. A = main board; B = signal board; C = sensor board; D = sensor panel.

HARDWARE DESIGN

Basic Principles of Functioning

The physical basis of mutual capacitance sensing, as implemented in WireTouch, is an AC signal which passes from one conductor to another via an electric field. Nearby objects can alter the electric field and thereby weaken the coupled signal. This effect can be used to measure proximity.

A typical multi-touch panel designed for mutual capacitance sensing consists of a two-layered panel with vertically oriented conductive paths on one, horizontal paths on the other side and dielectric material in-between. The intersections of the paths act as capacitors which couple the AC signal. By sequentially sending signals through the vertical paths and measuring them on the horizontal side – or vice-versa – the positions of one or multiple conductive objects can be tracked. The current implementation of WireTouch supports up to 1024 measurable intersections. In contrast to grid based systems relying on *self-capacitance* sensing, *ghosting* does not pose a problem.[2]

General Design Considerations

While the actual hardware design may differ from SmartSkin in many ways, it is heavily inspired by its basic layout. The sensing system consists of five hardware parts: the main board, the signal board, the sensor board, the touch panel and a computer running the tracking software (see Figure 1). The division of the hardware into separate components enables the researcher to change parts of the system without necessarily needing to discard others. This lowers the cost of upgrades, an aspect which becomes especially relevant in community driven development where multiple prototypes have to be upgraded repeatedly. The modular setup allows to rapidly prototype, exchange and test individual parts of the circuit. During the development of WireTouch to its current form, the main board

[1] A first workshop on possible collaborations between the FLOSS and CHI communities was held in 2010[1]. More forceful attempts to integrate FLOSS into research practices can be observed in neighbouring disciplines [14][13]. For a broader discussion on this topic, see also [6].

[2] When the user touches a self-capacitance measurement grid on multiple locations at once, determining the exact number of touches and their positions can become challenging. The system will track touches where there are none (ghost touches). It is, of course, possible to build a non-grid based self-capacitance measurement based system. This can solve the ghosting problem, but will reduce the maximum resolution of the sensor panel considerably.

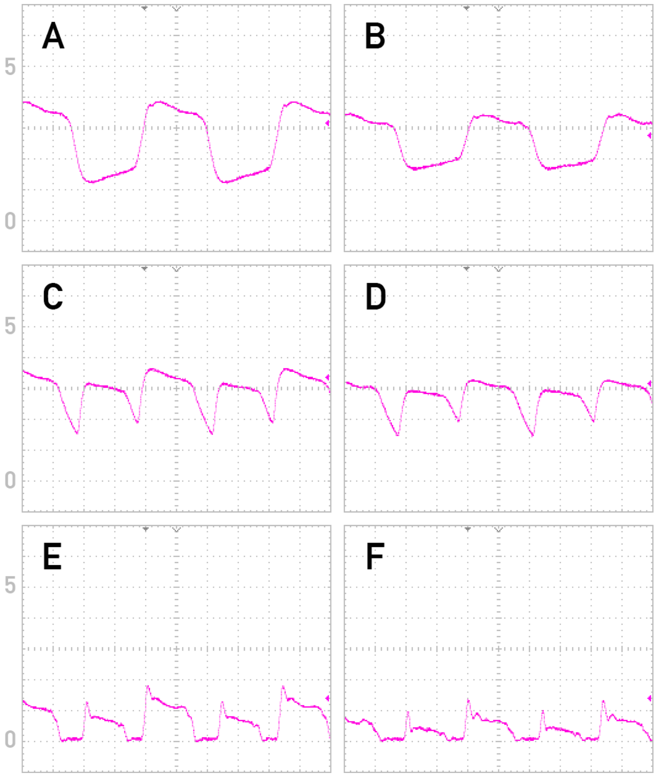


Figure 2. Here a 0 to 5V 457 kHz square wave is used to generate the reference signals. A single crosspoint is measured. The first column (A, C, E) shows the untouched state. The second column (B, D, F) shows the touched state (500ns per division). A, B = the signals after passing the sensor panel, amplified; C, D = the "folded" signals; E, F = the signals after passing the difference amplifier.

was replaced several times by a breadboard version while the basic functionality of the sensor and signal board remained unchanged.

To support the creation of custom hardware derivatives with minimal equipment, we limited the use of SMD parts to a single module. Apart from the signal board, everything can be soldered by hand. We also consciously chose an ATMEGA328P MCU as the heart of the main board to lower the entry barrier for interaction researchers which are already familiar with the popular Arduino IDE.

Circuit Design

The 8-bit AVR RISC-based microcontroller runs Arduino-compatible firmware and controls the hardware layer of the system. One of its tasks is to generate a square-wave at an adjustable frequency (132kHz – 762kHz).

This ”reference signal” is transmitted to the WireTouch signal board via a board-to-board connection cable and fed through a voltage buffer. The buffered signal is then passed through a multiplexer, which applies it consecutively to each vertical line of the sensor array. The grid lines connected to the signal board are floating traces. Through the small capacitance formed between the vertical and horizontal sensor grid lines, the signal is coupled into the WireTouch sensor board. Each horizontal line is grounded through a high-value pull-down resistor and pre-amplifies the coupled signal through a dedicated low-noise op-amp, before passing it back to the WireTouch main board after being multiplexed.

In the next step, the main board inverts the signal and uses an analog switch triggered by the reference signal to "fold" the coupled signal across the DC part (see Figure 2). The folded signal is then stretched across the MCU's supply voltage range with a difference amplifier. The stretched signal is then fed into a peak detector. The peak detector's capacitor is programmatically discharged between the measurements of consecutive intersection points by grounding it via an analog switch. The MCU's ADC finally samples the signal for a software-definable duration. A higher duration reduces the influence of stray noise. Naturally, as the sampling interval is increased, the overall frame rate of the system decreases.

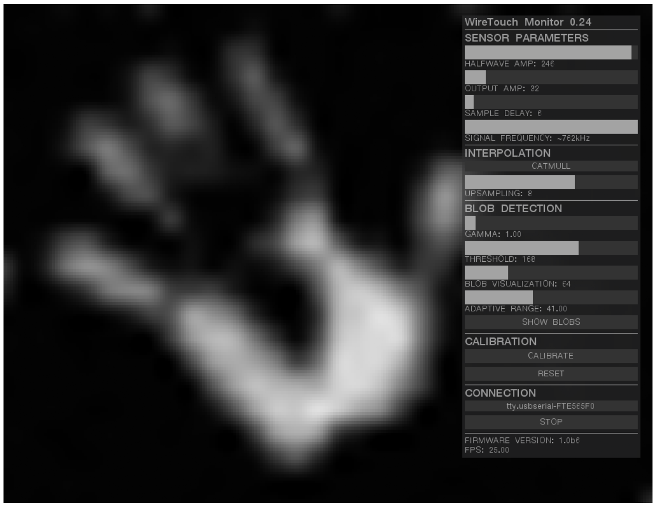
The processed signal is sampled at 10 bits of resolution. The individual measurements are sent to the host computer over the serial UART link. In order to not waste bandwidth with padding bits, the data is sent using a packed binary format. Since the serial bandwidth is comparatively limited, the data transfer doesn't employ addressing or partitioning information. Thus, the host determines the edges of a data frame, as well as individual node offsets within a frame, solely based on the total amount of taken measurements per frame. In the case of bytes being dropped from the serial ring buffer, the system needs to be re-synchronized to a known state.

Calibration

Due to small imperfection in the sensor matrix, as well as differences in impedance related to different PCB traces or cable lengths, the coupling effect exhibits a variance at each cross point. The calibration aims to adapt to such differences at the analog processing level by defining parameter offsets for the difference amplifier baseline and output stage amplification parameters.

When the calibration starts, these parameters are first reset to a default value. By repeatedly sampling the circuit output at a cross point roughly at the middle of the sensor, the difference amplifier baseline parameters successively run through a predefined range, determining the highest possible value at which the measurement still exhibits some noise and thus doesn't appear to be clipped by the amplification process. The output amplification remains at the default level during this process. Once the baseline has been determined, the signal is sampled at each cross point, but with an iteratively incremented offset to the output amplification parameter. For each point, the offset is determined at which the measurement is closest to the value determined for the sample point in the first stage of the calibration process. The offsets are stored in the firmware and get applied to the circuit whenever a given crosspoint is sampled, thus mitigating both more or less efficient couplings at any cross point.

Since individual samples can be taken in timeframes of microseconds, the calibration process can be executed efficiently. However, in order to alleviate unwanted noise effects, measurements are repeated and averaged for each cross point, as to improve the accuracy of the calibration mechanism.



HALFWAVE AMP: 246
OUTPUT AMP: 32
SAMPLE DELAY: 6
SIGNAL FREQUENCY: ~762kHz
CATMULL
UPSAMPLING: 8
GAMMA: 1.00
THRESHOLD: 168
BLOB VISUALIZATION: 64
ADAPTIVE RANGE: 41.00
tty.usbserial-FTE565F0
FIRMWARE VERSION: 1.0b6
FPS: 25.00

Figure 3. The tracking software, showing the interpolated measurements of a hand, resting on flat 32x22 sensor matrix. The sensor panel is covered with a 1.5mm thick glass plate.

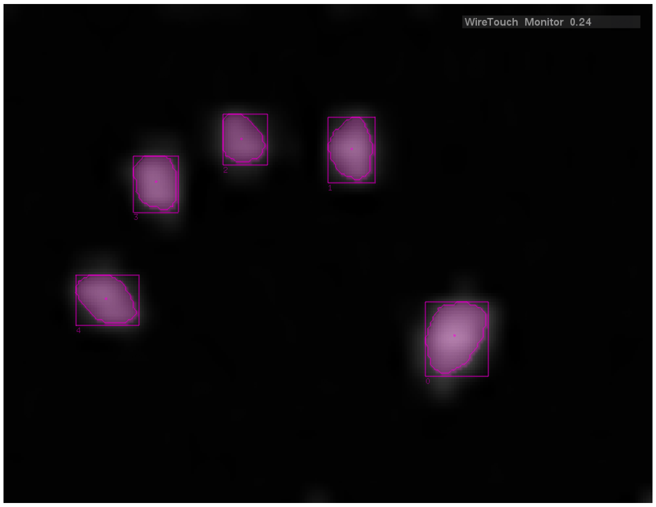


Figure 4. The software, tracking five moving fingers at 25fps.

The complete calibration process can be executed at system startup, but can also be re-run at a later time. This is especially useful for improvised sensor matrices such as glued wires or flexible surfaces, where the sensor lines might stretch or move during operation.

SOFTWARE DESIGN

Crucial parameters of the measurement circuit are adjustable by software running on the connected computer. This includes the frequency of the reference signal, the duration of the measurements, the difference amplifier's subtracted DC component and the output amplification. Altering these parameters via the GU during operation allows to manually fine-tune the system in accordance to the sensor-design and the objects to be tracked (see Figure 3). The end result of the scanning process can be examined visually on the computer screen as the raw data gets transformed into a grayscale image. One single image represents one completed scanning routine.

By treating the measurements as a stream of images, methods and implementations from the field of image processing and computer vision can be used to filter the captured data. The current version of our software applies a gamma expansion and features a set of configurable interpolation algorithms. As the image-processing happens solely on the computer, other image enhancement techniques can be added with ease.

The actual tracking of finger tips or other objects is completely offloaded to the software running on the connected computer. Due to portability and ease of development we based its implementation on openFrameworks. OpenFrameworks is a popular open source cross-platform toolkit that provides access to the OpenCV library, which includes a robust contour finder [16, 4]. Various third-party libraries are available that extend OpenCV and provide helpful services, such as blob labelization across the time axis. The current implementation of the tracking software allows the adjustment of the blob detection thresholds via the GUI. We also implemented a rudimentary TUIO server which transmits the coordinates and identification numbers of the tracked blobs to end-user applications running on the same or a networked machine [7].

EXEMPLARY APPLICATION: AN AUDIO-TACTILE MAP

Tactile graphics are common teaching tools in specialized schools for visual impaired students. Audio-tactile maps enhance tactile maps by linking pre-recorded sounds or synthetic speech to specific positions (hot spots) on the map. Basic implementations of this concept exist since the late 1980s [10]. The ability to continuously track multiple fingertips opens up new possibilities of interaction and new ways of information retrieval.

To explore this potential, we built a custom-sized sensor panel, tailored to the dimensions of the tactile maps used at a local school for visual impaired students. We designed a double-sided PCB board with 32 vertical traces connected to the signal board, and 22 horizontal traces connected to the sensor board, resulting in 768 individual sensors arranged on a 1cm spaced grid. We covered the top with a tactile map made out of A4-sized micro capsule paper (see Figure 5).

The visual feedback of the tracking software allowed us to adjust the sensitivity of the system to the properties of the touch panel and the material positioned between the panel and the user. The prototype proved capable of recognizing the touch of multiple fingers and their continuous movement on the tactile map.

We sent the coordinates of the tracked blobs to a high-level software application, which provided audio feedback whenever certain areas (hot spots) on the map were touched. We then created a tangible map of the school building and enhanced it with sounds, the students had recorded while exploring the building together with a teacher.

During our first test of the prototypical system, we observed that users occasionally brought all ten fingers and both palms into play, in order to acquire a quick overview of larger areas. In other situations, the second hand served as a static point of reference: While resting at a certain position on the map, one or several fingers of the other hand were used to examine

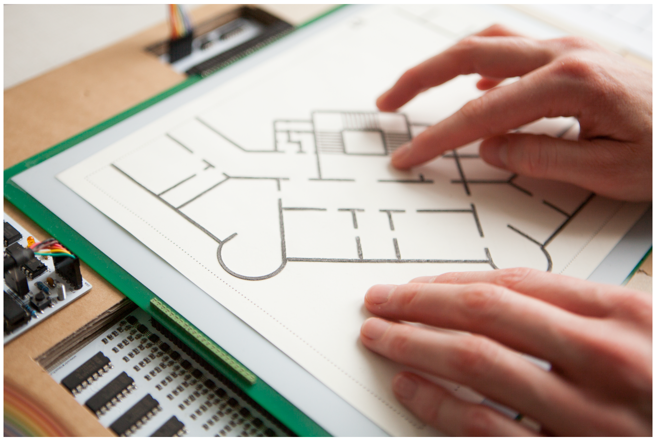


Figure 5. While moving the fingers across the tactile floor plan, the user gets auditory cues about near hot spots.

specific details or to measure distances. The continuous playback of the recorded sounds lead to a continuous stream of distracting noise. We therefore decided to decrease the amount of unintentional playbacks by changing the trigger mechanism to double-tapping. We also decided to communicate the positions of the hot spots through sonification. Now the user would get subtle acoustic cues about hot spots in reach, while exploring the tactile map. When the fingers of the user passed a hot spot, the software would generate a stereophonic sound, informing the user about the relative position by altering its volume, pitch and balance. To generate these sounds, we sent the tracking data to a real-time sound synthesis server via the TUIO protocol.

By implementing a crude palm-detection algorithm we were also able to discern different modes of the user's activity automatically. How these additional informations can be used to further enhance the interaction with the audio-tactile map – in particular during dual-handed explorative procedures – is a matter of future research.

CONCLUSION

We have presented a “full stack” open tracking device and prototyping platform based on mutual capacitance sensing. Built on top of widely available, well-documented hard- and software-components, the system can be readily adapted to different uses and projects. The firmware, the tracking client software, the schematics, as well as the PCB layouts are freely available[17].[3]

As a demonstration of the system's capabilities, we described an exemplary, viable application, which employs finger tracking on tactile surfaces designed for visual impaired users. The versatility of the system allowed us to build a tailor sized sensor matrix and adjust the measurement parameters, taking the tangible overlay into account. The openness of the tracking software enabled us to quickly alter the interactive system right after we had conducted our first informal user tests.

We hope that WireTouch will support other researchers in the exploration of new application areas. Future work may also include the simplification of the circuit and the build process, the improvement of the overall performance, and the development of general guidelines for the design of non-standard sensor panels.

In the broader scheme of things we hope to spark a discussion. We have argued that the full disclosure of user interface systems can facilitate HCI research in multiple ways. But we are also aware that a simple ”bottom up” approach might prove insufficient. If the development of open soft- and hardware should become more commonplace in our discipline, matters of publishability, funding and other structural challenges need to be addressed.

[3] The software source code and the design files have been released under the GNU General Public License v3. This, however, does not free the reader from patent law. Before building and distributing a specific hardware application, be sure to study the legal situation in your country.

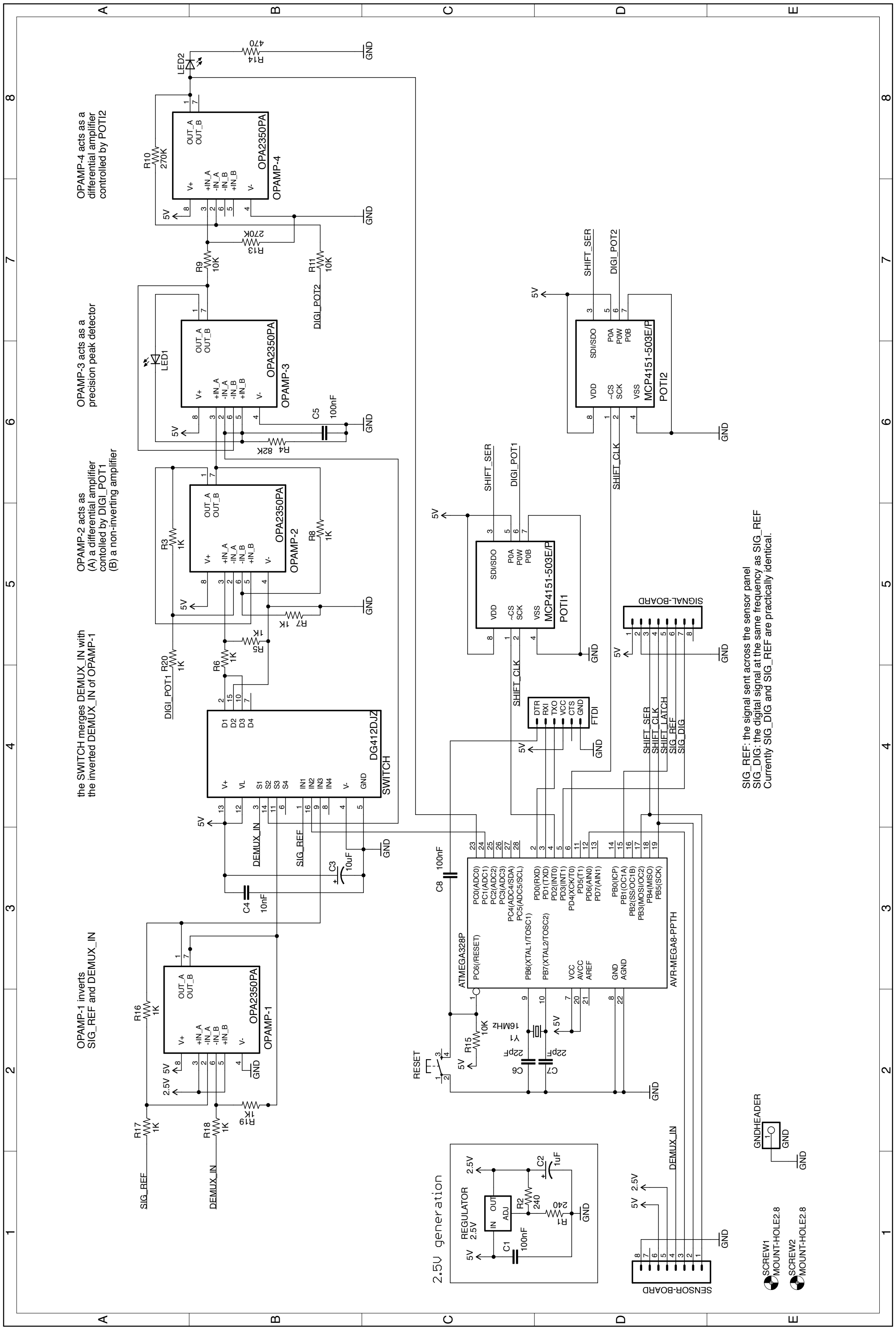

OPAMP-4 acts as a differential amplifier controlled by POTI2
OPAMP-3 acts as a precision peak detector
OPAMP-2 acts as (A) a differential amplifier contolled by DIGI_POT1 (B) a non-inverting amplifier
the SWITCH merges DEMUX_IN with the inverted DEMUX_IN of OPAMP-1
OPAMP-1 inverts SIG_REF and DEMUX_IN
SIG_REF: the signal sent across the sensor panel
SIG_DIG: the digital signal at the same frequency as SIG_REF
Currently SIG_DIG and SIG_REF are practically identical.
2.5V generation
REGULATOR 2.5V
OPA2350PA
OPAMP-1
OPAMP-2
OPAMP-3
OPAMP-4
DG412DJZ
SWITCH
ATMEGA328P
AVR-MEGA8-PPTH
MCP4151-503E/P
POTI1
POTI2
FTDI
SIGNAL-BOARD
SENSOR-BOARD
GNDHEADER
SCREW1 MOUNT-HOLE2.8
SCREW2 MOUNT-HOLE2.8
RESET
16MHz
SHIFT_SER
SHIFT_CLK
SHIFT_LATCH
SIG_REF
SIG_DIG
DIGI_POT1
DIGI_POT2
DEMUX_IN
LED1
LED2
5V
2.5V
GND
A
B
C
D
E
1
2
3
4
5
6
7
8

Figure 6. Schematic diagram of the main board, v1.3

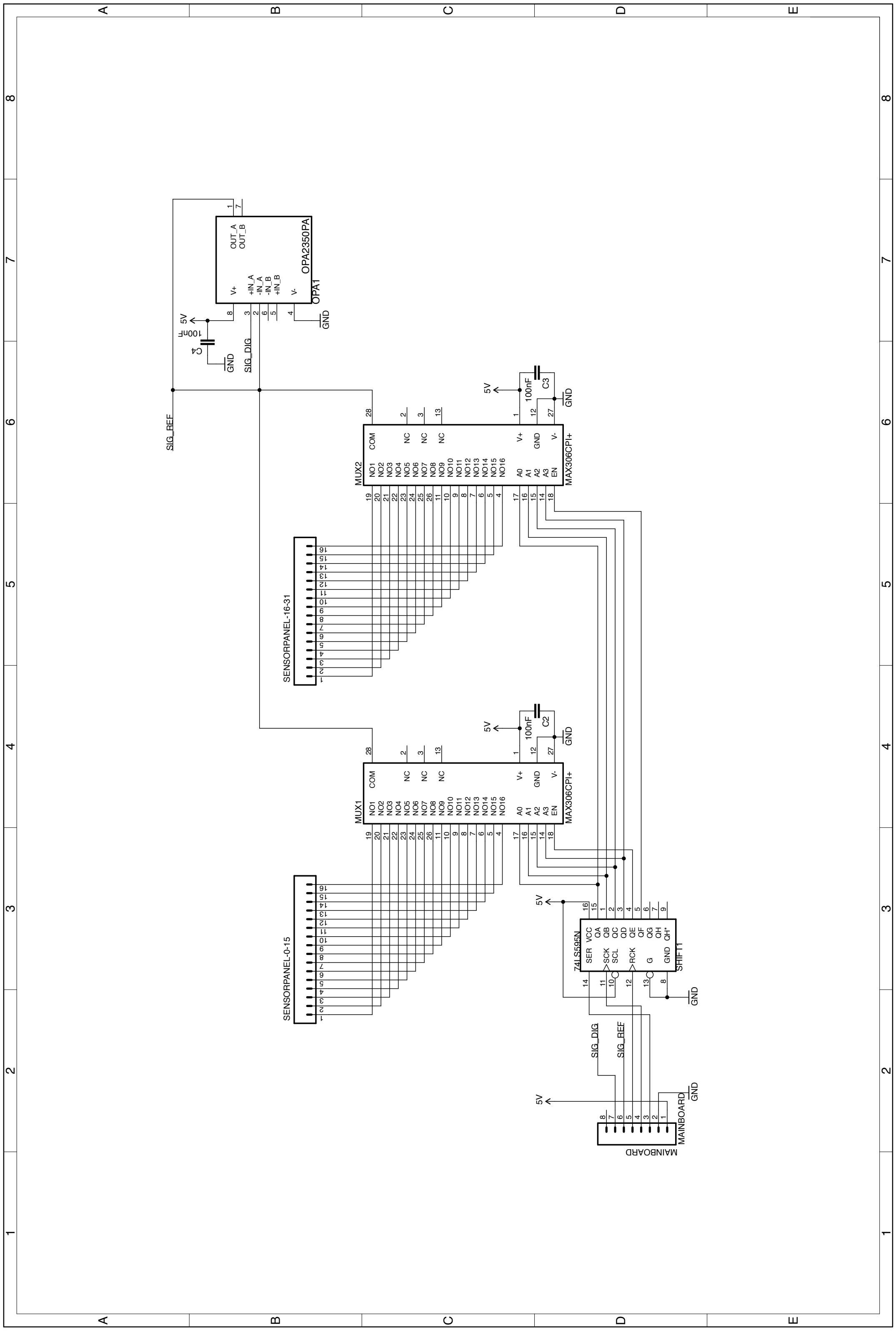


Figure 7. Schematic diagram of the signal board, v1.3

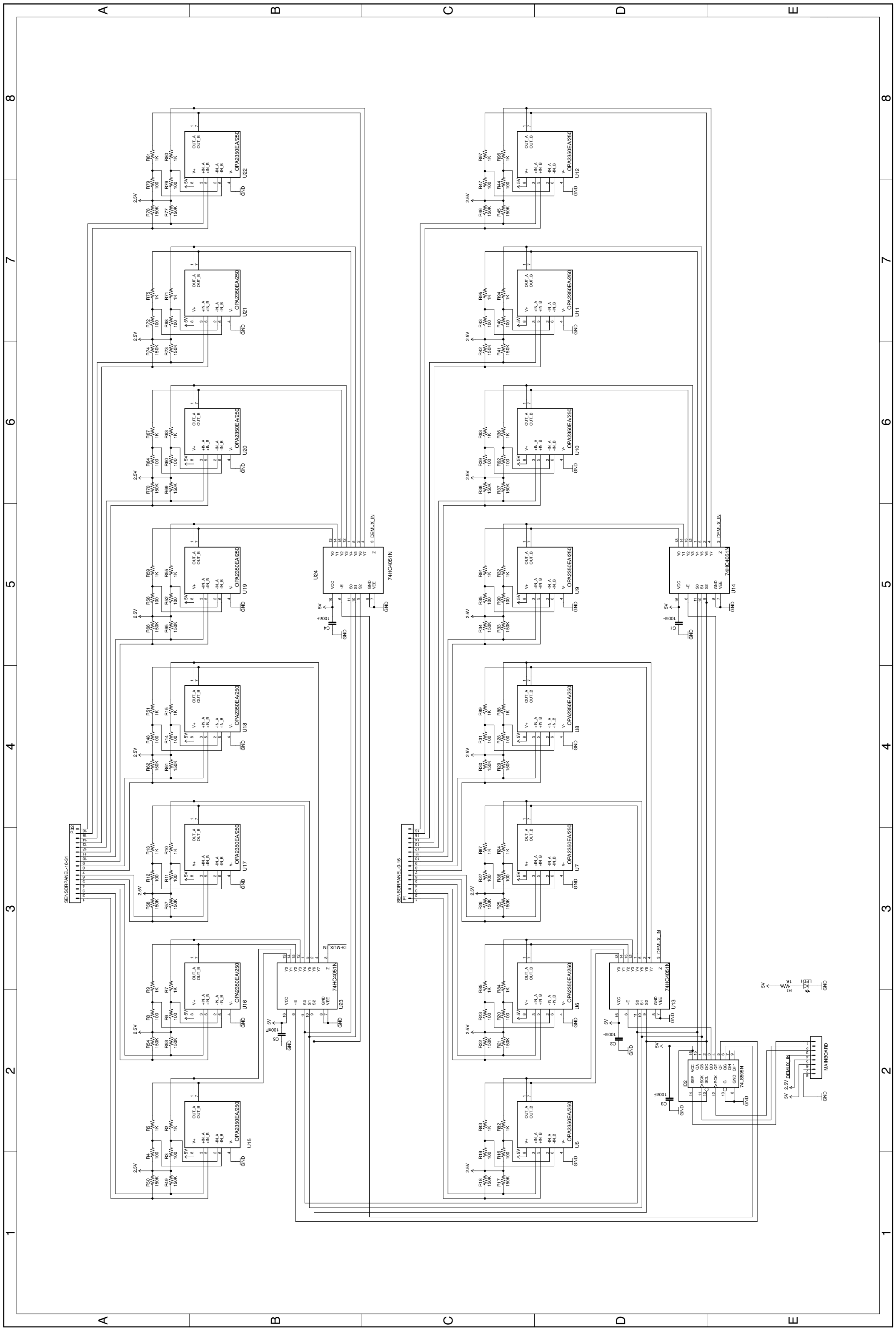


Figure 8. Schematic diagram of the sensor board, v1.3

REFERENCES

1. Bach, P. M., and Terry, M. The future of FLOSS in CHI research and practice. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, ACM (New York, NY, USA, 2010), 4473–4476.

2. Barragán, H. Wiring: Prototyping physical interaction design. Master's thesis, Interaction Design Institute Ivrea, 2004.

3. Beck, F., and Stumpe, B. Two devices for operator interaction in the central control of the new CERN accelerator. 73-6, CERN, 1973.

4. Bradski, G. The OpenCV library. *Dr. Dobb's Journal 25*, 11 (November 2000), 120, 122–125.

5. Buxton, B. A touching story: A personal perspective on the history of touch interfaces past and future. *Society for Information Display (SID) Symposium Digest of Technical Papers 41*, 1 (May 2010), 444–448.

6. Ince, D. C., Hatton, L., and Graham-Cumming, J. The case for open computer programs. *Nature*, 482 (February 2012), 485–488.

7. Kaltenbrunner, M., Bovermann, T., Bencina, R., and Costanza, E. TUIO - a protocol for table based tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)* (Vannes, France, 2005).

8. Lee, S., Buxton, W., and Smith, K. C. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1985).

9. Olberding, S., Gong, N.-W., Tiab, J., Paradiso, J. A., and Steimle, J. A cuttable multi-touch sensor. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, ACM (New York, NY, USA, 2013), 245–254.

10. Parkes, D. Nomad: an audio-tactile tool for the acquisition, use and management of spatially distributed information by visually impaired people. In *Proceedings of the second international conference on maps and graphics for visually impaired people*, A. F. Tatham and A. G. Dodds, Eds. Internation Cartographic Association Commission VII (Tactile and Low Vision Mapping) and Royal National Institute for the Blind, London, 1988, 24–29.

11. Rekimoto, J. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, ACM (New York, NY, USA, 2002), 113–120.

12. Savage, V., Zhang, X., and Hartmann, B. 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, 2012), 579–588.

13. Sonnenburg, S., Braun, M. L., Ong, C. S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Müller, K.-R., Pereira, F., Rasmussen, C. E., Rätsch, G., Schölkopf, B., Smola, A., Vincent, P., Weston, J., and Williamson, R. The need for open source software in machine learning. *J. Mach. Learn. Res. 8* (Dec. 2007), 2443–2466.

14. Stajich, J. E., and Lapp, H. Open source tools and toolkits for bioinformatics: significance, and where are we? *Briefings in Bioinformatics 7*, 3 (2006), 287–296.

15. Stumpe, B., and Sutton, C. The first capacitative touch screens at CERN. *CERN Courier 50*, 3 (April 2010).

16. Suzuki, S., and Abe, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing 30*, 1 (1985), 32–46.

17. Wagner, A., and Kaindl, G. WireTouch. `https://github.com/arminbw/wiretouch`. (Accessed: 2016-09-04).

18. Zimmerman, T. G., Smith, J. R., Paradiso, J. A., Allport, D., and Gershenfeld, N. Applying electric field sensing to human-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press/Addison-Wesley Publishing Co. (New York, NY, USA, 1995), 280–287.