

WiReSens Toolkit: An Open-source Platform towards Accessible Wireless Tactile Sensing

Devin Murphy

ECE

University of Washington
Seattle, Washington, USA
murphydevinf@gmail.com

Junyi Zhu

EECS

University of Michigan
Ann Arbor, Michigan, USA
zhujunyi@umich.edu

Akshay Gadre

ECE

University of Washington
Seattle, Washington, USA
gadre@uw.edu

Antonio Torralba

CSAIL

MIT

Cambridge, Massachusetts, USA
torralba@mit.edu

Paul Pu Liang

Media Lab and EECS

MIT

Cambridge, Massachusetts, USA
ppliang@mit.edu

Wojciech Matusik

CSAIL

MIT

Cambridge, Massachusetts, USA
wojciech@csail.mit.edu

Yiyue Luo

ECE

University of Washington
Seattle, Washington, USA
yiyueluo@uw.edu

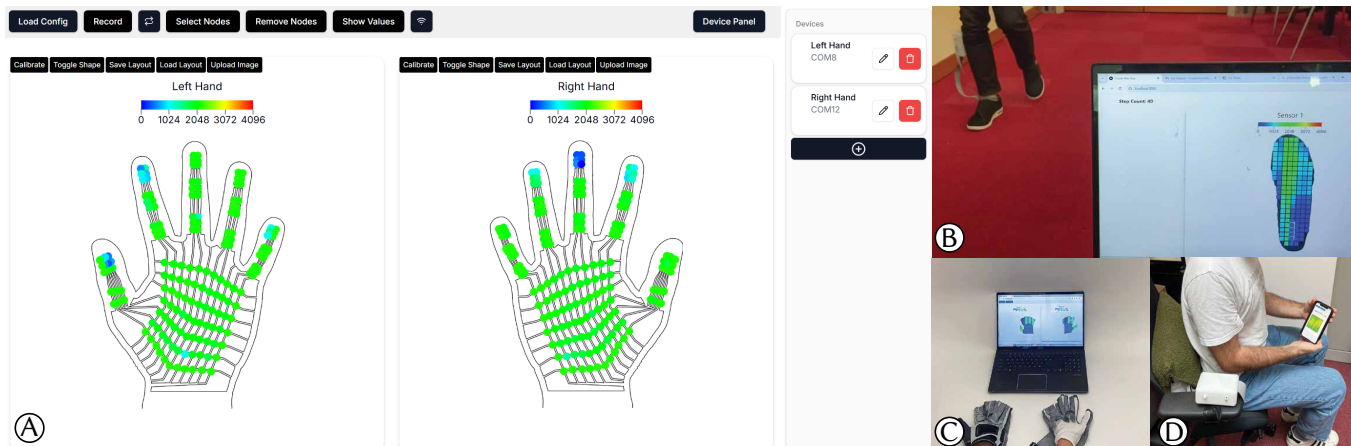


Figure 1: The WiReSens Toolkit bridges the gap between fabricating resistive tactile sensors and deploying scalable touch-interactive systems. Combining open-source hardware and software uniquely designed for portability, adaptivity, and efficiency, it interfaces with a web-based GUI that enables novices in embedded sensing to rapidly prototype across diverse users, form factors, and application domains.

Abstract

Past research has widely explored the design and fabrication of resistive matrix-based tactile sensors for creating touch-sensitive devices. However, real-world deployment of resistive tactile sensing systems remains difficult for individuals with limited prior

experience in embedded sensing due to challenges of portability, adaptivity, and efficiency. We introduce the WiReSens Toolkit, an accessible, open-source platform to bridge this gap. Central to our approach is adaptive hardware for interfacing with resistive sensors and a web-based GUI that streamlines access to advanced features for building scalable tactile sensing systems, including multi-device programming and wireless visualization across three communication protocols, autocalibration for adaptive sensitivity, and intermittent data transmission for low-power use. We validated the toolkit's usability through a user study with 11 novice participants, who, on average, configured a tactile sensor with over 95% accuracy in under five minutes, calibrated sensors 10× faster than



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

TEI '26, Chicago, IL, USA

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1868-7/26/03

<https://doi.org/10.1145/3731459.3773323>

baseline methods, and showed improved sense-making of tactile data.

CCS Concepts

• **Human-centered computing** → **User interface toolkits**; • **Hardware** → **Sensor devices and platforms**.

Keywords

Resistive Touch Sensing, Toolkits, Tangible User Interfaces, Wireless Devices

ACM Reference Format:

Devin Murphy, Junyi Zhu, Akshay Gadre, Antonio Torralba, Paul Pu Liang, Wojciech Matusik, and Yiyue Luo. 2026. WiReSens Toolkit: An Open-source Platform towards Accessible Wireless Tactile Sensing. In *Twentieth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '26)*, March 08–11, 2026, Chicago, IL, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3731459.3773323>

1 Introduction

Tactile sensors endow humans and everyday objects with the ability to capture, understand, and augment tangible interactions, increasingly finding applications within cyber-physical systems [29], the Internet of Things (IoT) [2], and Human-Computer Interaction (HCI) [31, 39]. While a variety of principles have been employed to sense touch, many researchers adopt resistive matrix-based approaches [28, 47, 53] where a piezoresistive material is sandwiched between two orthogonal electrode arrays to measure applied pressure via resistance changes. This approach is appealing because it is intuitive, reduces the need for extensive wiring [27], and past research has demonstrated that it can be implemented via a variety of fabrication techniques [1, 25, 34, 53].

To realize a vision of ubiquitous tactile sensing for large-scale understanding of tangible interactions, infrastructure beyond sensor manufacturing is essential. Deploying resistive tactile sensing devices in real-world settings presently involves substantial overhead that can result in inefficient system designs. To be portable, these devices must rely on wireless communication. Each protocol presents trade-offs in range, bandwidth, power consumption, and environmental suitability [12], and developers may invest time implementing one protocol only to find that another is better suited to a new application. Covering multiple sensing surfaces further requires multi-device synchronization, and wireless communication itself is power-intensive [43], often limiting capture duration unless custom energy-saving strategies are employed. Additionally, resistive tactile sensors require careful calibration, as signal ranges can vary widely due to fabrication inconsistencies and the diversity of interaction types they are designed to capture [18, 38]. This makes it difficult to ensure consistent pressure readings across devices or users. Without a streamlined infrastructure to handle these challenges, the deployment of resistive tactile sensing systems remains tedious at best and infeasible at scale.

To address these challenges, this paper introduces the WiReSens Toolkit: an open-source platform designed specifically for novices in embedded sensing (Fig. 1). We take inspiration from sensing toolkits that simplify prototyping through domain-specific abstractions [40, 42, 59] to design a platform that provides the missing

infrastructure needed to deploy tactile sensors at scale in real-world contexts. WiReSens mediates access to the complex functionality needed for deploying resistive matrix-based sensors through a web-based graphical user-interface (GUI) that interfaces with carefully designed hardware and software. The platform is fully open-source, with usage instructions and supplementary materials available online via a dedicated website¹ as well as through GitHub² to support future updates.

The software and hardware underlying the WiReSens Toolkit is uniquely designed for accessibility, portability, adaptivity, and efficiency. The toolkit targets novices in embedded sensing, offering intuitive interfaces for configuring sensor readout and visualizing real-time, contextually relevant sensor data. Our platform also broadens the environments in which users can confidently deploy sensors by supporting multi-device communication over three wireless protocols — Wi-Fi, Bluetooth Low Energy (BLE), and ESP-NOW — interchangeable with the click of a button. In addition, it introduces an automatic sensitivity calibration method to preserve data quality when resistive matrix-based sensors are repurposed across users, form factors, or applications. Finally, the WiReSens Toolkit provides interaction-aware data transmission through intermittent sending, enabling tangible interactions to be captured over a longer duration by enabling devices to predict sensor readings instead of sending them wirelessly.

We evaluate the WiReSens Toolkit against its design goals of portability, adaptivity, and efficiency through a series of technical experiments, a user study, and deployment in diverse application scenarios. By using our toolkit to deploy up to five simultaneously transmitting tactile sensing devices each with 1024 unique sensing taxels and to interface with sensors fabricated using four established methods from prior work, we demonstrate robust multi-sender wireless communication across all three supported protocols, optimized pressure resolution during readout, and up to a 42% increase in device lifetime through interaction-aware intermittent transmission. We also conducted a user study with 11 novice users, which demonstrated that the platform has excellent usability, streamlines access to complex functionality, and helps users better interpret pressure data visualizations. We finally demonstrate the flexibility of our toolkit by using it to rapidly prototype various applications, including musical gloves, shoe soles for gait monitoring, a pillow for posture monitoring and remote control, and a smart home welcome mat.

In summary, this work contributes

- **A flexible platform for prototyping wireless resistive tactile sensing systems**, including a web-based GUI for quick sensor reconfiguration and interactive data visualization; an open-source resistive sensor readout circuit available in two sizes; and an adaptive hardware module with on-device auto-calibration to support sensitivity tuning across users, form factors, and applications. We also implement interaction-aware wireless transmission methods that reduce power consumption during periods of tactile inactivity.
- **A technical evaluation** validating the system's robustness and efficiency, demonstrating reliable multi-device wireless

¹<https://wiresens-materials.vercel.app/>

²<https://github.com/WiReSens-Toolkit>

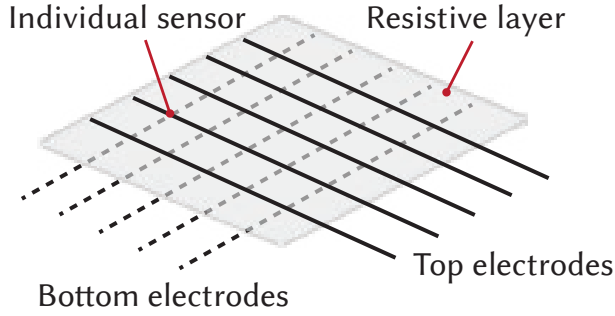


Figure 2: Typical layout of a resistive matrix-based pressure sensing form factor, with a resistive layer sandwiched between two sets of orthogonally positioned electrodes.

communication across protocols, improved pressure resolution through adaptive readout optimization, and up to a 42% increase in device lifetime with intermittent transmission.

- **A user study and application-driven validation**, showing that the toolkit supports intuitive configuration, calibration, and visualization workflows for novices. Example applications illustrate how the system enables rapid deployment of custom wireless tactile interfaces across diverse use cases.

2 Related Work

The WiReSens Toolkit is built upon the principle of resistive pressure sensing and is most similar to prior works which facilitate the creation of interactive touch-sensing systems. We first provide an overview of resistive pressure sensing before situating WiReSens Toolkit within the landscape of toolkits designed for tactile sensing.

2.1 Resistive Pressure Sensing

Past research has explored various methods for dynamic pressure sensing, with optical, capacitive, and resistive techniques being among the most widely used. Optical tactile sensors, such as Gel-Sight [57], analyze light pattern changes caused by material deformation under pressure. While offering high resolution and performance, these sensors often rely on proprietary hardware and face limitations in scalability and flexibility due to visual occlusion challenges. Capacitive sensing, on the other hand, typically involves two conductive electrodes separated by a deformable dielectric layer, where applied pressure alters capacitance [24]. However, capacitive methods often require shielding layers to mitigate stray capacitances [8, 44]. Additionally, capacitive sensing is more commonly used in a mutual [16] or swept-frequency [41] setup for binary touch detection or proximity sensing [46] rather than precise pressure measurement. This work therefore focuses on resistive-based pressure sensing.

In resistive pressure sensing, orthogonally arranged electrodes sandwiching a piezoresistive layer form a 2D sensing matrix, where pressure is measured via resistance changes at electrode intersections [5] (Fig. 2). This approach avoids the scalability problem that

is present in optical pressure sensors, as wires can easily be added or removed and spaced tighter or more spread out. When combined with digital fabrication approaches like machine knitting [53], digital embroidery [1], 3d printing [25], and flexible circuit printing [34], it has allowed resistive sensing to be incorporated in a variety of form factors like hand bags [36], carpets [28, 52], and even hair extensions [33]. We show how our toolkit is adaptable to these various fabrication approaches in this paper’s *Toolkit Evaluation* section.

Pressure signals are typically read from a resistive sensing matrix using a zero potential scanning readout circuit [9, 19], the hardware on which WiReSens Toolkit is also built (Fig. 4B). The circuit topology is well established and widely adopted due to its ability to reduce crosstalk between neighboring electrodes [9]. Although some works provide similar open-source hardware [11, 15, 17], WiReSens Toolkit supports a higher density (up to 1024 pressure sensing locations), and to our knowledge is the first open-source hardware which includes a digital potentiometer in this architecture for tuning sensitivity.

2.2 Touch Sensing Toolkits

A range of toolkits have been developed to facilitate prototyping touch sensing applications, with early works focusing primarily on capacitive sensing. CapToolKit by Wimmer et al. [55] and Midas by Savage et al. [42] were among the first to provide open-source hardware and software to streamline the prototyping of capacitive touch interfaces. Midas lowered the barrier to designing custom capacitive sensors which could be fabricated using digital manufacturing techniques such as inkjet printing, vinyl cutting, and CNC milling. These contributions allowed researchers and designers to rapidly iterate on new touch-sensing applications without needing deep expertise in circuit design or signal processing. Building on this foundation, the Multi-Touch Kit [37] further abstracted capacitive sensor processing by introducing an Arduino library for mutual capacitance-based multi-touch arrays. This allowed capacitive touch sensors to be implemented using only a standard microcontroller (MCU). While these toolkits provided valuable abstractions for capacitive touch sensing, they primarily supported binary multi-touch detection.

Support for resistive touch sensing is typically limited to specific application contexts and sensor types. The E256 project by Donneaud et al. [11] provides open-source Arduino firmware for denoised readout of resistive matrix-based eTextiles designed for musical expression. By leveraging a fabric-based piezoresistive sensor, E256 enables pressure-sensitive interaction beyond simple binary touch detection. Similar to the Multi-Touch kit, it includes Processing software for real-time visualization of sensor readings. Additionally, the 3D ViTac system by Huang et al. [17] has expanded accessibility within robotics communities by offering a hardware guide, Arduino firmware, and Python ROS libraries for parallel gripper resistive tactile sensors.

WiReSens Toolkit builds upon these prior works (Table 1) across key areas of HCI toolkit contributions as identified by Ledo et al. [22] and our design goals of portability, adaptivity, and efficiency. With built-in wireless communication support for up to five devices, each capable of sensing dynamic pressure at 1024 unique locations,

	CapToolKit [55]	Midas [42]	Multi-Touch Kit [37]	E256 [11]	3D ViTac [17]	Ours
Sensing principle	Capacitive	Capacitive	Capacitive	Resistive	Resistive	Resistive
# of sensors per device	8	9	256	256	256	1024
# of sending devices	1	1	1	1	4	5
Sensitivity calibration	✗	✗	✗	✗	✗	✓
Wireless protocol support	✗	✗	✗	✗	✗	✓
Power-saving mode	✗	✗	✗	✗	✗	✓
Visualization	Static	Configurable	Static	Static	Static	Configurable

Table 1: In contrast to existing open-source tactile sensing toolkits, WiReSens enables sensing on a larger scale (4x the number of sensors and 5x the number of devices), automatic sensitivity calibration, wireless protocol support, and power-saving operation modes. As a result, users can deploy systems with greater sensing coverage and flexibility across diverse application scenarios.

WiReSens Toolkit extends beyond individual sensors to enable the development of distributed multi-sensor interactive systems. This allows novice users to explore tactile interaction design on a larger scale. By integrating features such as automatic sensitivity calibration and power-saving modes, it simplifies access to more complex functionality, enabling users to rapidly prototype tactile sensing systems which are adaptable and sustainable in addition to functional. Finally, WiReSens is the first of these toolkits to be evaluated with first-time users, demonstrating that even those without prior experience can set up a functional tactile sensing system in 5 minutes.

3 Design Requirements

To ensure the WiReSens Toolkit would be effective for developing scalable tactile sensing systems, we outlined several key goals for its design.

Accessibility: The toolkit should be usable by novices and extensible by experts. Hardware should be easy to order and manufacture, ideally through board houses that support full assembly. Users with limited experience in embedded systems should be able to configure and deploy devices without modifying low-level firmware. At the same time, those with more technical expertise should be able to customize and extend the system.

Portability: Tactile sensing devices must be easily repositioned to capture interactions across different surfaces and environments. This means wireless communication is a necessity, especially in wearable applications where wires can restrict movement. While many sensing toolkits adopt a single communication protocol (often Bluetooth [10, 14, 59]), different protocols offer distinct advantages [12]. For example, Wi-Fi offers excellent speed and reliability indoors but is difficult to access outdoors, where Bluetooth communication is more viable. A toolkit that simplifies switching between protocols would broaden the range of environments where tactile sensing systems can be effectively implemented.

Adaptivity: A key challenge with resistive tactile sensors is that the resistance-force relationship varies across fabrication techniques, and different applications require distinct force sensitivity ranges. In these different contexts, the same readout circuit can either overamplify or underamplify signals, limiting pressure resolution [51]. Additionally, resistive sensing array data is typically

displayed statically, with each sensing node fixed in place. However, the real-world positioning of these nodes can shift when different users wear the same sensor or as conditions change over time. An adaptive visualization that accounts for these shifts could help users interpret tactile interactions more accurately and intuitively.

Efficiency: Wireless communication is power-intensive, which limits device lifetime and poses challenges for sustainable development [43, 58]. Low-power operation is therefore important to enable large-scale data collection over extended periods of time, though it can affect the computational capabilities of devices. Users should have access to methods that allow them to navigate this trade-off. Additionally, users may invest significant time and resources into developing interactive systems, so reconfiguring them for different form factors, users, and applications should be seamless, minimizing hardware and firmware modifications.

4 The WiReSens Toolkit

The WiReSens Toolkit consists of a zero-potential readout circuit for interfacing with resistive sensing arrays and a Python-backed UI for configuring sensing devices, managing data recording and playback, and visualizing pressure data. We begin with a walkthrough to illustrate the core functionalities offered by our platform followed by the implementation details behind its development.

4.1 Walkthrough

This walkthrough illustrates an example of setting up a tactile sensing system using a digitally embroidered 32×32 node resistive sensor and the WiReSens Toolkit. Although this example uses a specific fabrication method, we reiterate that the toolkit is agnostic to sensor construction and supports resistive sensors produced through a wide range of manufacturing techniques.

Hardware Setup. To begin, the user connects the horizontal and vertical electrodes of their resistive sensor to one of our zero-potential scanning readout circuits, available in large (32×32) and small (16×16) configurations (Fig. 3D). The large circuit uses ribbon cables, while the small version connects via flex cables. In this example, the user selects the large circuit and connects its ESP32 microcontroller to their laptop via USB. They then upload the Arduino-based firmware, which handles both sensor readout and wireless communication. Unlike prior toolkits, this upload is a

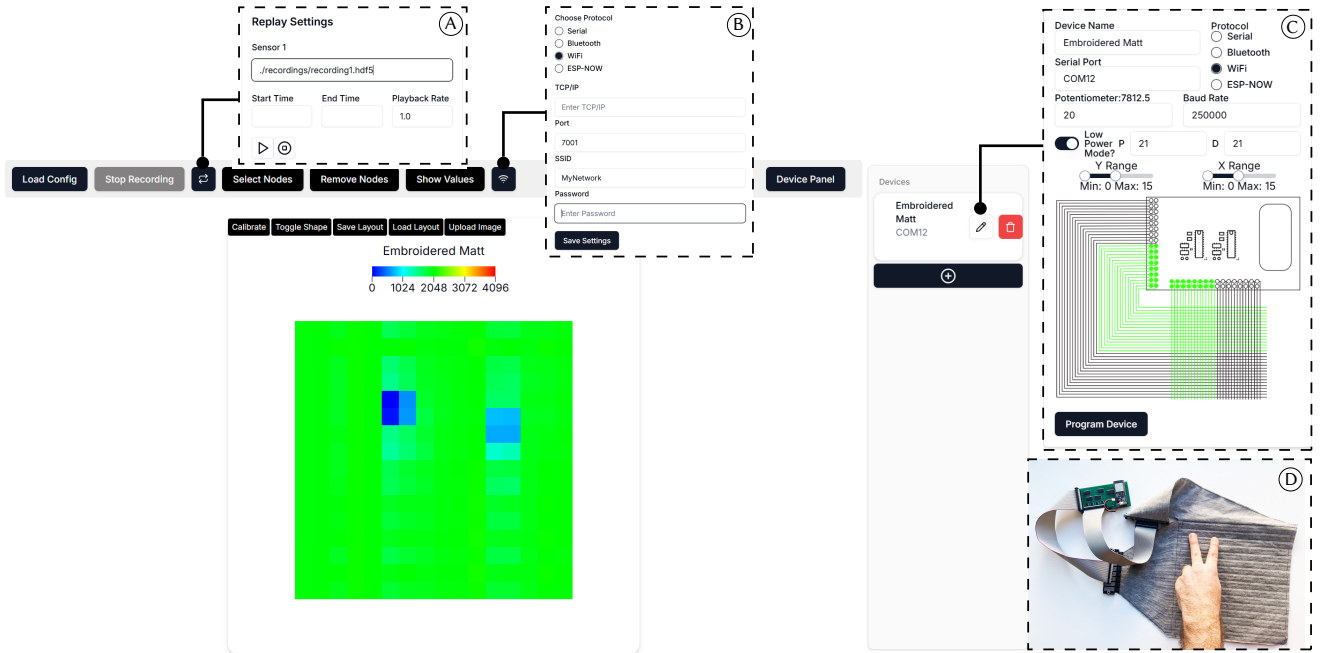


Figure 3: The WiReSens Toolkit web-based programming interface wirelessly records and displays pressure data from sensing devices (D) in real time, with panels to replay recordings (A), configure wireless protocols (B), and configure the devices themselves (C). The embroidered matt is configured to read only from the top-left quadrant and send readings via WiFi.

one-time step. After initial setup, all reconfiguration and deployment is handled through the web GUI, with no need for firmware changes. This eliminates a barrier for users without a background in embedded systems, enabling rapid iteration and adaptation across sensing scenarios.

Configuring the Readout Area. With the hardware connected, the user starts the Python backend and opens the web GUI in their browser. Through the Device Manager, they add a new device and access its configuration panel (Fig. 3C). For this application, they only need the top-left portion of the 32×32 array, so they configure a smaller 16×16 sensing region to improve the effective frame rate. Rather than requiring firmware edits or physical rewiring to modify the readout area, the interface allows intuitive selection of rows and columns, dynamically highlighting the chosen electrodes in green and providing a clear preview of the active sensing region.

Setting up Wireless Communication. WiReSens uniquely provides support for multiple wireless protocols within a single platform—USB-serial, Wi-Fi (TCP), Bluetooth Low Energy (BLE), and ESP-NOW (Espressif’s peer-to-peer protocol [13]). This flexibility lets users easily switch communication methods to best suit their application’s range, latency, or power requirements. Here, the user chooses Wi-Fi for its low latency and inputs the laptop’s IP address and network credentials through the wireless configuration panel (Fig. 3B).

Programming the Device. Once satisfied with the configuration, the user clicks “Program Device” to send the updated settings to the MCU via USB. If adjustments are needed later, the configuration can be updated directly through the web GUI, allowing

for rapid iteration. The user then unplugs the device, connects a battery, and can begin using the device for their application.

Recording, Visualization, and Playback. By clicking “Record,” the user begins data acquisition across all configured devices, using their selected communication protocols and readout areas. They’ve also written a custom method for tracking touch points in Python, which hooks into the toolkit backend and runs concurrently with recording and visualization. Real-time pressure data is displayed as a heatmap of square “sensing pixels” representing the active readout regions. With WiReSens, users can reposition sensing pixels via drag-and-drop during recording to support flexible sensor layouts, in addition to saving these layouts and uploading background images for contextual visualization. Data is stored locally in HDF5 format, and a built-in replay feature enables playback with adjustable timestamps and speed controls (Fig. 3A).

Turning on low-power mode. Aware of Wi-Fi’s energy demands, the user activates WiReSens’s low-power mode through the configuration panel (Fig. 3C). Unlike prior tactile sensing systems that transmit wireless data continuously, WiReSens includes an intermittent transmission mode that extends battery life during untethered operation, making the system more practical for long-term, mobile, or wearable deployments. The user runs a Python utility function which takes their pressure recording file as input and returns optimized intermittent transmission parameters that balance power efficiency with sensing quality for their specific application. They then input these values into the configuration panel, reconnect the MCU to their laptop using a USB cable, and click the

"Program Device" button, reconfiguring the device to operate in the low-power mode.

Sensitivity Calibration. During playback, the user notices reduced sensor responsiveness. In prior systems, addressing this would require accepting the loss of sensitivity or physically modifying the readout circuit. With WiReSens, the user simply clicks "Calibrate" to launch an automatic gain adjustment routine. Over a 60-second window, the user engages the tactile sensor in a way that reflects the range of pressures it would experience under normal operation. The system then sets an optimized amplification gain, expanding the dynamic voltage range and improving pressure sensitivity for the user's specific application.

4.2 Implementation

Here we provide further implementation details behind the toolkit's functionalities.

Web GUI and Python Library. The WiReSens Toolkit web GUI provides a visual programming interface for control of the toolkit's functionalities. It is implemented via NextJS and communicates with a locally running Python Flask backend through WebSockets. The Python backend serializes each device's data in a separate thread, where each frame is timestamped using the desktop's local Unix time and saved to an HDF5 file, ensuring synchronized data across devices. This approach ensures that data from multiple devices are consistently aligned in a common timebase, which is suitable for applications where millisecond-level synchronization is sufficient and all devices report to a single host computer [23]. It also provides a library with utility functions for analyzing sensor recordings, optimizing parameters for low power consumption, and executing custom methods on sensor data in real time.

Adaptive Readout Circuit. The WiReSens Toolkit provides two open source zero-potential readout circuit PCBs which can be manufactured and assembled using the gerber files provided for \$90.26 and \$140.92 USD, respectively, via JLCPCB (Fig. 4A). In this architecture, each electrode layer is connected to a digital multiplexer (MUX). To measure resistance at a specific node, the ESP32 controls one MUX to ground an electrode in one layer while the other MUX selects an electrode in the opposing layer for measurement. The voltage at this electrode is passed through an operational amplifier (LMV324, Texas Instruments) and digitized by a high speed, low-power successive approximation Analog to Digital Converter (ADC - AD7466, Analog Devices) at a frequency of 100 kilo-samples per second (ksps). To allow tuning of tactile sensor sensitivity and avoid the problems of ADC underutilization and op-amp output saturation, we introduce an additional op-amp (TLV9152, Texas Instruments) in inverting configuration with variable gain (Fig. 4B). The gain is controlled by a 50 k Ω digital potentiometer (MCP4018, Microchip), denoted as R_{pot} .

Wireless Communication. The toolkit firmware employs a batched transmission scheme for efficient data throughput, sending data in batches of 120 samples. All supported protocols—WiFi (TCP), BLE, and ESP-NOW—use a unified packet format: the first byte encodes the sender ID, the next two bytes specify the start index of the 120-sample segment within the sensing array, the next

240 bytes contain the sample data, and the last four bytes encode a unique packet ID. The default ESP32 transmission policies are adopted for each protocol, meaning packets are retransmitted if not acknowledged for WiFi (TCP) and ESP-NOW but not for BLE notifications [6, 49, 50]. Users can optionally configure a protocol-specific delay (in milliseconds) via the wireless settings panel, which the device observes before transmitting each packet. This delay helps stagger transmissions across devices, reducing the likelihood of packet collisions when multiple devices are active.

Low-Power Mode. To enable low-power operation modes, we employ intermittent sending, conserving power by minimizing data transmission for signals that are easy to predict or unchanging. The MCU predicts the voltage at a sensor node before reading it using the window-based time series forecasting equation:

$$v[n+1] = v[n] + \frac{1}{p} \cdot (v[n] - v[n-1]) \quad (1)$$

where $v[n]$ represents the ADC voltage reading at time step n and p is a parameter controlling how responsive the prediction is to recent shifts in voltage. The ESP32 will only send the data packet if the average error between all readings and predictions is below some user defined threshold d . If the data packet is not sent, the Python library will predict its value using the same forecasting method as the ESP32 (1), ensuring that the average error across each data packet remains below the specified threshold d . This is modeled after the method proposed by Suryavansh et al. [48], which was chosen due to its low computational complexity and memory requirements, making it ideal for implementation on a MCU.

To help select the parameter p and error threshold d , we provide a utility function which takes any tactile recording as input and performs a grid-based optimization. The optimization minimizes an objective function based on the normalized root mean square error (NRMSE) of the predicted tactile frames E and the percentage of transmitted data r :

$$\alpha * E + (1 - \alpha) * r, \quad (2)$$

where α is an adjustable parameter managing the trade-off between accuracy and transmission percentage. The script generates a visualization of the objective function, allowing users to refine and adjust the search space for further optimization.

Automatic Sensitivity Calibration. The automatic calibration sequence sets the resistance R_{pot} that best optimizes the tactile sensor's range of voltage output for a particular application. Upon start, the MCU scans the sensor array using the minimum possible value of R_{pot} for a configurable duration (defaulting to 10 seconds) and keeps track of the minimum sensor output voltages during this time. At the end of the calibration duration, the method then calculates the average of these minimum sensor output voltages V_{min} and determines the value of R_{pot} that will make the output of the opamp in the adaptive module V_{out} equal to 0 volts according to equation 3, obtained from analysis of the readout circuit topology in Fig. 4, where the denominator 264 Ω represents the smallest detectable resistance by the readout circuit (as set by $V_{ref} = 1.8V$ and $R_{g1} = 220\Omega$ in our implementation).

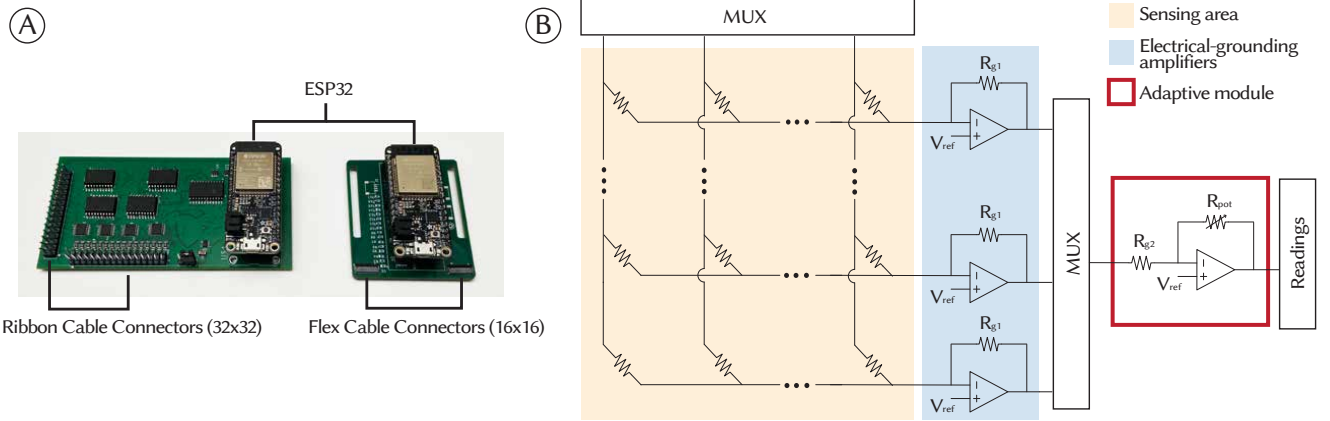


Figure 4: (A) Adaptive zero-potential readout circuit open-sourced by WiReSens Toolkit in large (left) and small (right) sizes (B) Schematic of general zero potential readout circuit (left) with additional opamp and digital potentiometer for adaptivity (red).

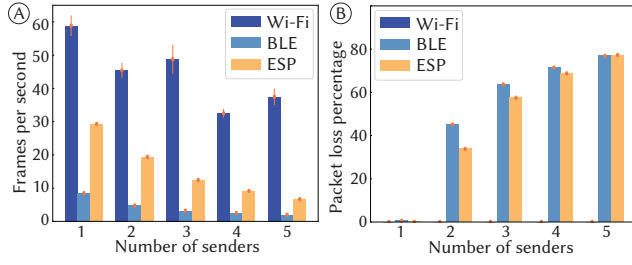


Figure 5: Average throughput (A) and average percentage of packets lost (B) per sender during multi-sender wireless communication using Wi-Fi, BLE, and ESP-NOW. Wi-Fi (up to 60 fps) supports fast, high temporal-resolution tactile interactions like tapping and stroking; ESP-NOW (up to 30 fps) is suited for moderate-speed gestures like pressing or grasping; and Bluetooth (up to 10 fps) is best for capturing slow or discrete events such as posture shifts or step detection.

$$V_{out} = V_{ref} - \frac{R_{pot}}{264\Omega} \times (V_{ref} - V_{min}). \quad (3)$$

This effectively maps the minimum pressure applied to the maximum sensor output (V_{ref} Volts) and the maximum pressure applied to the minimum sensor readout (0 Volts), ensuring that the full range of the ADC is used and pressure resolution is maximized.

5 Toolkit Evaluation

In this section we conduct a technical and user evaluation of the WiReSens Toolkit, focusing on speed and reliability of the multi-sender wireless communication, automatic calibration performance, power-reduction capabilities, and usability of the web-GUI interface.

5.1 Wireless Communication

To assess the scalability of the wireless communication abstractions in WiReSens Toolkit (including Wi-Fi, BLE, and ESP-NOW), we

programmed 1 to 5 devices to wirelessly send tactile sensing data from a 32x32 sensing matrix to a laptop running the Python backend. Each protocol and sender combination was tested for three minutes, repeated three times. Fig. 5 reports the average and standard deviation of throughput (in frames per second) and packet loss percentage per sender across all three tests. Here, one frame corresponds to approximately 2108 bytes.

Throughput generally decreases and packet loss increases as the number of senders grows across all protocols, due to limited bandwidth. However, Wi-Fi exhibited some exceptions to this trend, likely due to network variability. It consistently delivered the highest performance, achieving 58.84 fps with 0% packet loss for a single sender and 37.49 fps with no loss even with five senders. In contrast, the narrower bandwidths of BLE [6] and ESP-NOW [13] led to higher collision rates in multi-sender, high-throughput scenarios, resulting in substantial packet loss. ESP-NOW was the next most reliable protocol, with 29.30 fps and 0% packet loss for a single sender, dropping to 6.67 fps and 77% loss with five senders. BLE consistently showed the lowest performance, with 8.48 fps and 0.5% loss for one sender and just 2 fps and 76% loss with five. Accordingly, we recommend Wi-Fi for high-throughput use cases, and ESP-NOW for offline scenarios where internet connectivity is unavailable.

The WiReSens Toolkit allows users to manage the trade-off between throughput and reliability by tuning the transmission frequency via protocol-specific delays. For example, with three senders, adding a 157 ms delay to BLE reduced the frame rate to 0.75 fps per sender but kept packet loss below 1%. ESP-NOW achieved similar reliability with only a 10 ms delay, maintaining a higher throughput of 11.76 fps per sender. These results suggest that BLE and ESP-NOW are viable alternatives to Wi-Fi for scenarios where lower power consumption or extended range is prioritized, without significantly compromising data integrity.

5.2 Sensor Adaptivity

Here we evaluate the effectiveness of our calibration algorithm in two scenarios: adapting to different sensor fabrication techniques

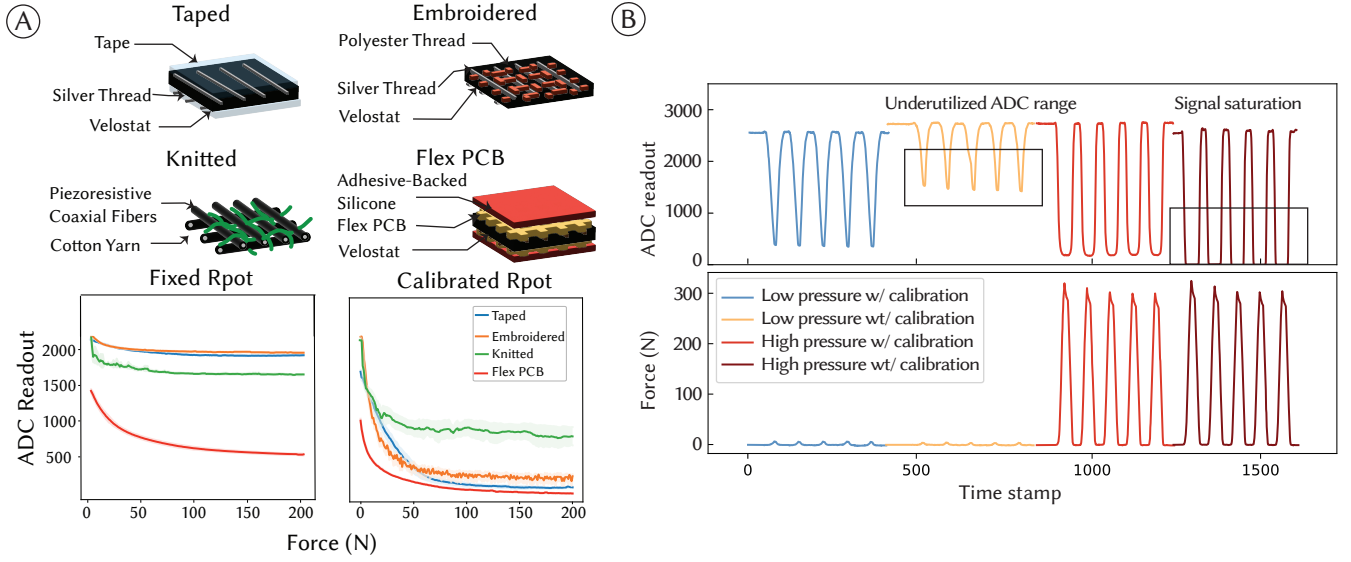


Figure 6: WiReSens Toolkit calibrates readout for different sensors and application scenarios (A) Average and standard deviation of ADC readout for four different resistive pressure sensors during cyclical force tests, before and after calibration. (B) Average ADC readout during low and high-pressure application cycles, before and after calibration. Low-pressure calibration maximizes pressure resolution (blue and yellow curves) and High-pressure calibration avoids saturation (red and brown curves).

and different application contexts. Both experiments used a mechanical tester (Shimadzu AGX-V2) to apply an adjustable normal force over an area of 9 cm^2 to a resistive sensor.

Calibrating for Different Sensors. We fabricated four resistive matrix-based pressure sensors using established methods from prior work: taping copper threads [47], digital embroidery [1], machine knitting [28], and flexible printed circuit board (FPCB) printing [34]. Further fabrication details are outlined in this paper’s supplementary materials³. For each sensor, we applied a cyclic force to a 3×4 node region at its center for five cycles, with the digital potentiometer’s resistance (R_{pot}) set to 787Ω . Fig. 6A presents the mean and standard deviation of the ADC readout within the actuated region as a function of applied force for each fabrication method. Due to the inverting topology of the op-amp, the ADC readout is inversely correlated with force, exhibiting two approximately linear regions. Notably, the taped, embroidered, and knitted sensors display a significantly smaller overall change in ADC readout across the force range when using the default resistance, thereby reducing their effective pressure resolution.

Next, we applied our auto-calibration procedure while maintaining a 200N applied force, yielding calibrated R_{pot} values of 8661Ω , 4724Ω , 9055Ω , and 1181Ω for the taped, knitted, embroidered, and FPCB sensors, respectively. Repeating the cyclic force test, we observed enhanced pressure resolution in all four sensors post-calibration, as evidenced by the increased ADC output range over the same applied force range (Fig. 6A).

Calibrating for Different Applications. We then applied low and high-pressure sequences to the same 3×4 node region of the

digitally embroidered sensor during a calibration period of 5 minutes and recorded the calibrated resistance values: $R_{\text{pot}} = 14062.5 \Omega$ for the low pressure and $R_{\text{pot}} = 8984.37 \Omega$ for the high pressure. We then repeated each pressure sequence twice: once with $R_{\text{pot}} = 14062.5 \Omega$ and once with $R_{\text{pot}} = 8984.37 \Omega$. Fig. 6B presents the applied force and average ADC readout over time for each sequence. When calibrated, the ADC readout range expanded for low-pressure scenarios, enhancing resolution, while high-pressure calibration prevented 0V saturation, allowing sensitivity up to 250 N.

5.3 Wireless Power Reduction

We first investigate how packet transmission rate affects power consumption in a tactile sensing device using the BLE protocol. We use an abstraction where the device alternates between two states: sending data (state A) or idle (state B). Then, the average current drained during device operation is calculated as

$$I_A * t_A + I_B * (1 - t_A), \quad (4)$$

where I_A and I_B is the average current draw during states A and B, respectively, and t_A is the time percentage state in state A. I_B is measured while the MCU samples but does not send pressure data, and I_A is estimated by subtracting I_B from the measured current while data is sent continuously. Using a Nordic Semiconductor Power Profiler, we log the current (mA) from a 3.3V 1200mAh battery over three minutes to obtain these averages.

We then simulate how the percentage of transmitted data changes t_A , and thus the average current draw of the device. We validate this simulation by fixing packet transmission frequency at 10 different values between 0 and 100% and recording the average current draw. Fig. 7A shows the simulated and observed trends, with results indicating that sending 1% of packets could extend the device

³<https://wiresens-materials.vercel.app/sensors/>

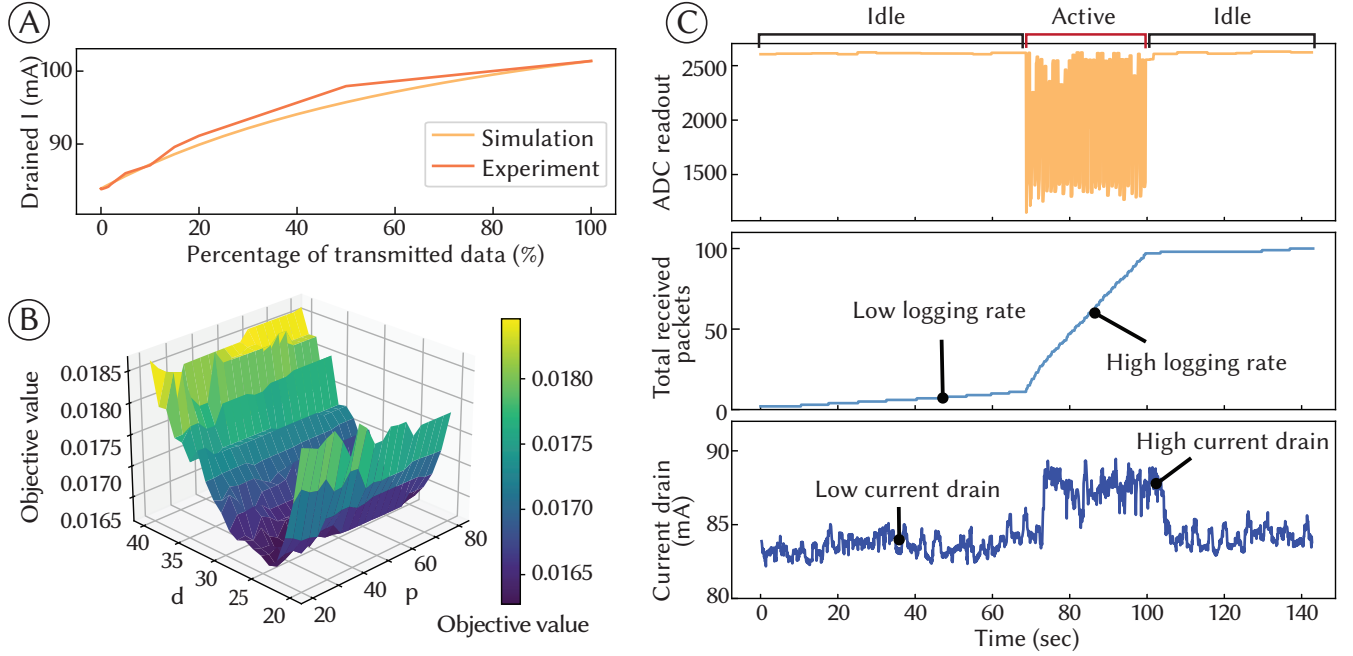


Figure 7: Characterization of intermittent sending performance. (A) Simulated and observed average current draw (mA) from a tactile sensing device as a function of the percentage of data transmitted during BLE operations. (B) Visualization of objective function used to select intermittent sending parameters. (C) Average ADC readout, received packet count, and current draw over time during periods of no applied pressure (Idle) and repeated pressure (Active). Low current drain indicates power is saved during Idle periods.

lifetime by over 20% for BLE. Using similar calculations with data collected under Wi-Fi operation, we estimate this could increase to 42 % longer battery life, as Wi-Fi operation is generally more power-intensive than BLE (we measured 152.47 mA vs 101.31 mA for BLE).

To further validate the intermittent sending performance, we use a mechanical tester to apply a series of repeated presses to a 32x32 tactile sensor during intermittent operation. Using the Python utility, we optimize the transmission parameters p and d ($p = 29$ and $d = 26$, Fig. 7B) and use them to program the device for the live pressure test. The results (Fig. 7C), indicate that when the sensor is inactive, packet transmission is minimal with low average current drain compared to when the sensor is active. The optimized values achieved packet transmission just under 5% with a NRMSE of around 0.012. With a 1200mAH battery, we estimate that this would extend the device’s lifetime by over 2 hours, while ensuring the predicted tactile frames differed from the ground truth frames by approximately only 1.2% of the full-scale range.

5.4 User Study

In our user study, we assessed whether first-time users found the WiReSens Toolkit easy to use and to what extent it simplifies prototyping resistive tactile sensing systems. Participants completed three tasks: (1) programming a new sensor, (2) calibrating a sensor, and (3) reproducing recorded pressure data. This study was approved by our institutional review board.

Participants. We recruited 11 participants (4 female, 7 male) aged 19–32 ($M = 22.64$, $SD = 3.87$) through university mailing lists and word of mouth. Participants were a mix of undergraduate and graduate engineering students. Two had prior experience with resistive pressure sensors, and none had used any part of the WiReSens Toolkit before.

Task 1: Programming a Device. To evaluate the toolkit’s learnability and usability, participants were tasked with using it to program a new device. They first received an overview of the resistive pressure sensing principle and were introduced to a 32x32 digitally embroidered sensor. The study moderator then demonstrated how to configure a device to read and record pressure data over Bluetooth using the web GUI. After the demonstration, participants could ask questions before attempting the task independently. They were asked to configure the device to read and record data from a new sensor—an 8x24 shoe sole—without guidance. They could reconfigure the device until they were confident in their setup. We recorded the time taken and the configured readout area. At the end of the study, participants completed a System Usability Scale (SUS) survey [7] to evaluate their experience.

Task 2: Calibrating a Device. Next, we assessed the extent to which our auto-calibration procedure simplifies the calibration process. The study moderator preconfigured a new device to read from a 3x2 node area of a flex PCB sensor while a constant force of 3.0 kg was applied. Participants were given background information on the sensitivity calibration challenges of resistive sensors and

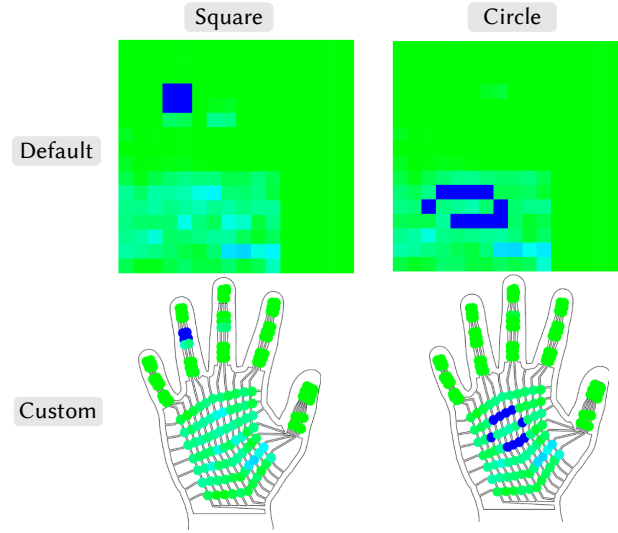


Figure 8: Study participants are asked to reproduce presses from two of the four depicted visualizations: one square press (left) and one circle press (right) from either a default (top) or custom (bottom) visualization.

were then instructed to manually adjust the digital potentiometer value through the web GUI to find the minimum amplification such that all six nodes displayed an ADC reading of zero. We recorded the time taken for this task and the user-selected value. The study moderator then asked the participant to perform the same calibration using the “Calibrate” button, which ran for a fixed duration of 10 seconds, and recorded the auto-calibrated potentiometer value.

Task 3: Interpreting Visualized Pressure Data. Finally, we examined whether the toolkit’s customized visualizations improve spatial reasoning about tactile data. Participants were shown two WiReSens Toolkit visualizations of presses on a sensing glove—one created with a circular object and one with a square object. Each visualization was presented in one of two formats: either a custom layout configured to match the glove’s shape or a default square layout (Fig. 8). The order of presentation was counterbalanced across participants. Participants were asked to reproduce the two presses using the objects and could adjust their attempt until they felt confident. We recorded the time taken and asked them to complete a survey assessing their confidence and perceived difficulty during each task on a Likert scale rating from 1 (Not Difficult/Not Confident) to 7 (Very Difficult/Very Confident).

Results. On average, participants took less than 5 minutes ($M = 4.15$, $S.D. = 1.71$) to program a new sensor, achieving an average readout area accuracy of 96% ($S.D. = 4.3\%$) on their first attempt. The average SUS rating was 82.5 ($S.D. = 9.68$), considered excellent and falling in the highest quartile of usability scores [3, 4]. Open-ended feedback further highlights the toolkit’s novice-friendly nature:

- P4: “It was very cool to work with hardware integrated with the toolkit software because it made the learning process more

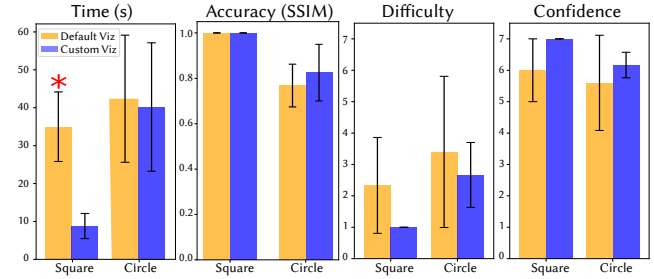


Figure 9: Mean and standard deviation of time (lower is better), accuracy (higher is better), difficulty (lower is better), and confidence (higher is better) for square and circle press tasks. Results for default (orange) and custom (blue) visualizations show custom visualization significantly improved participants’ performance in the square press task, reducing task time and perceived difficulty while boosting confidence. In the more challenging circle press task, the custom visualization also trended toward better speed, accuracy (SSIM), and confidence.

smooth and approachable. It was also beginner-friendly, and I could see myself using this for a project.”

- P9: “It was a very smooth interface that streamlines many of the common dull tasks associated with setting up a new sensor and visualizing its outputs.”
- P11: “Everything was intuitive and easily understandable.”

Participant feedback also highlighted areas for improvement in the system. Two participants suggested that higher pressure could be more intuitively represented by warmer colors and higher values in the visualization, with P4 noting, “To me, it feels more intuitive if more pressure correlates with a higher output, so the platform might be a bit easier to understand if you could somehow convert the measured output into pressure.” Several participants reported friction in the reprogramming workflow, as changing device parameters required stopping the recording, suggesting that future iterations could benefit from combining record and reprogram functions. Additionally, three participants indicated that the web GUI could provide clearer explanations of configuration parameters; P5 commented, “There were some terms/variables that I didn’t understand.” These observations reveal usability challenges that should be addressed in future work.

The toolkit expedites the tedious task of calibration. While all participants found the optimal R_{pot} value through manual calibration, this took 2.1 minutes on average ($S.D. = 1.15$). The auto-calibration always found the same user-identified optimal gain, but with a fixed calibration duration of 10 seconds, was over 10x faster. Three participants mention the auto-calibration as their favorite feature in open-ended feedback - P2 notes: “I liked the auto-calibration tool because it seemed quite tedious to find the resistance yourself”.

Finally, the WiReSens Toolkit’s custom visualization improves spatial reasoning of tactile data. We evaluated its effectiveness by analyzing participants’ task completion time, Likert scale ratings for difficulty and confidence, and accuracy of reproduced presses using the Structural Similarity Index Measure (SSIM) between binary

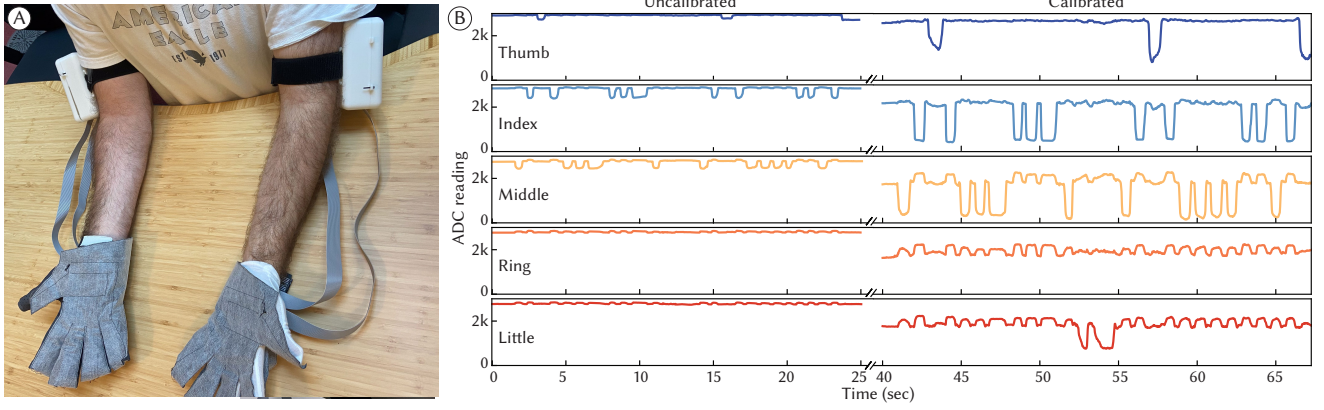


Figure 10: Wireless Musical Gloves: (A) Depiction of tactile sensing array, with readout circuit affixed to the arm via velcro straps (B) Average ADC readout in the Thumb, Index, Middle, Ring, and Little finger regions of one glove during "Mary Had a Little Lamb", before and after calibration, showing higher sensitivity after calibration.

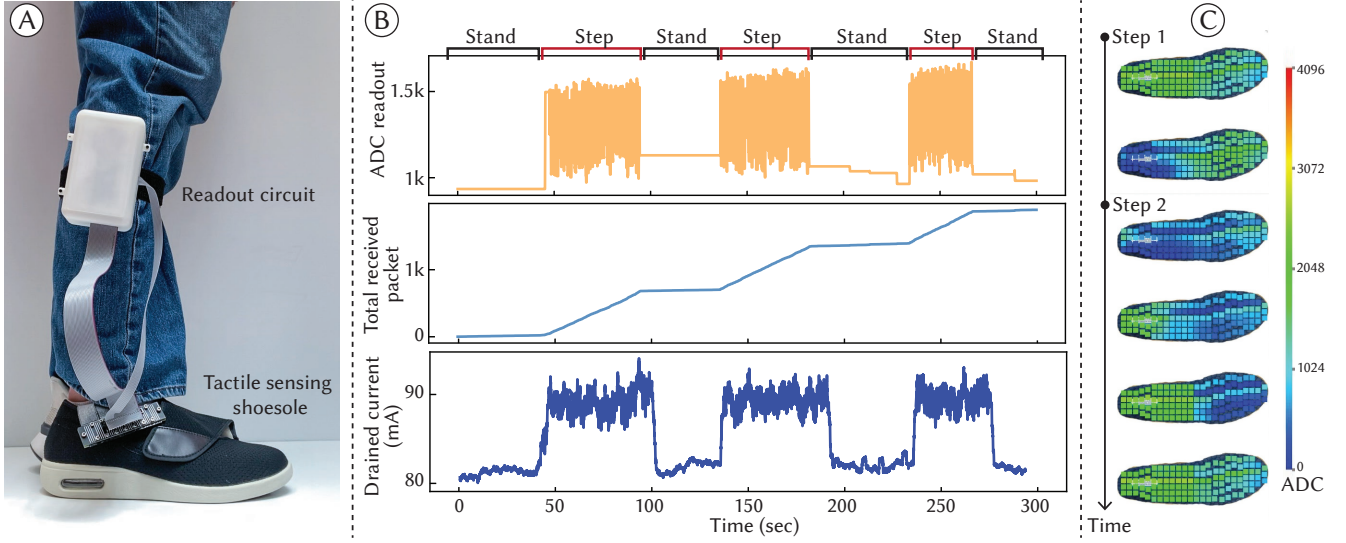


Figure 11: Gait Monitoring Shoe Sole: (A) Tactile sensing shoe sole form factor and readout circuit. (B) Power efficient operation using intermittent sending, with reduced current drain during periods of standing still, and (C) Customized visualization for gait monitoring and step tracking.

thresholded participant and ground truth presses (Fig. 9). Participants using the custom visualization in the square press task took significantly less time ($M = 8.79$, $S.D. = 3.32$, Welch's t-test $p < 0.05$) than those using the default ($M = 35.0$, $S.D. = 8.79$) and reported less difficulty ($M = 1.0$ for custom, $M = 2.33$ for default) and greater confidence ($M = 7.0$ for custom, $M = 6.0$ for default) on average. While we did not observe significant differences between visualization groups in the more challenging circle press task, participants using the custom visualization on average took less time ($M = 40.2$ for custom, $M = 42.4$ for default), reproduced presses more accurately ($M = 0.83$ for custom, $M = 0.77$ for default), and expressed higher confidence ($M = 7.0$ for custom, $M = 6.0$ for default) than those using

the default visualization. Six participants highlighted the visualization tools as their favorite feature, with P8 stating: "The layout is my favorite feature, the ability to hide and reconfigure layouts to match the shape of the object."

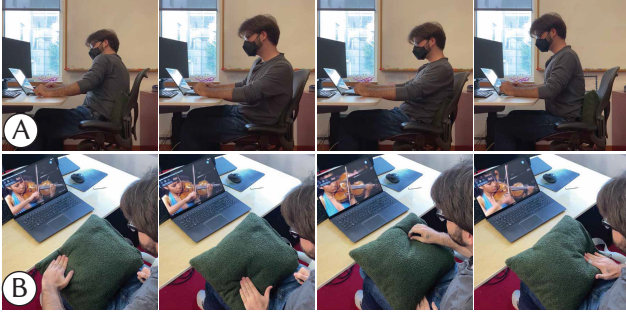
6 Example Applications

In this section, we demonstrate the utility of WiReSens Toolkit by using it to prototype a range of resistive tactile sensing systems. The tactile sensors were constructed with silver thread and Velostat, using digital embroidery [1] and direct taping [45]. Each application was prototyped in under 10 minutes.

Musical Gloves. Taking inspiration from gloves that offer new means of musical expression [35, 54], we used WiReSens Toolkit

Table 2: Applications developed using the toolkit, along with their wireless protocols, calibrated resistance ranges, and intermittent parameters (p, d).

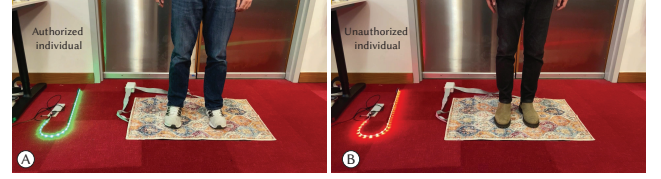
Application	Wireless Protocol	Rpot Value – Step (Resistance)	Intermittent Parameters (p, d)
Posture Monitoring Pillow	Wi-Fi	36 (14173)	(29, 26)
Remote Control Pillow	Wi-Fi	27 (10630)	—
Gait Monitoring Shoe Sole	BLE	23 (9055)	(40, 28)
Musical Gloves	Wi-Fi	27–30 (10630–11811)	—
Smart Welcome Mat	ESP-NOW	23 (9055)	—

**Figure 12: Tactile Sensing Pillow: Used for (A) Posture monitoring, from left to right: leaning right, leaning left, leaning back, sitting up straight and (B) Remote Control, from left to right: Play, Pause, Volume Up, Volume Down.**

to program two Wi-Fi-enabled tactile sensing gloves for playing a virtual piano, showcasing its potential for enhancing real-time, expressive musical performance. We recruited two users—one for the left-hand glove and one for the right-hand glove—to perform a piano duet both before and after calibration. The ADC readout in each finger region is shown in Fig. 10 for one of the users. After calibration, each user reported feeling like their glove was more responsive to their applied pressure, which can be evidenced by the greater range of ADC readout shown in Fig. 10B.

Gait Monitoring Shoe Soles. Prior research has demonstrated interest in gait monitoring as a valuable tool for understanding overall health [26, 32]. We programmed a BLE-enabled pressure monitoring shoe sole, recorded pressure data during 5 minutes of walking and standing still, and used the toolkit’s interactive visualization to replay the wirelessly recorded gait. As shown in Fig. 11, our device consumes considerably less power while the user is standing still than when they are stepping. The significant increase in packet transmission during the walking periods evidences that this is a direct result of our intermittent send algorithm ensuring efficient use of the onboard communication modules.

Other applications. Because the sensitivity of readout can adaptively change during device operation, WiReSens Toolkit enables the development of tactile sensing devices that can be repurposed for different applications. We use this feature to develop a tactile sensing pillow that can serve both as a passive recording device (higher sensitivity) for posture monitoring and as an active input interface (lower sensitivity) for remote control of media playback

**Figure 13: Smart Home Welcome Mat: IoT lamp processes tactile sensing data from a welcome mat to turn green when (A) an authorized individual enters the home and red when (B) an unauthorized individual enters the home.**

(Fig. 12). We also developed a smart home IoT welcome mat (Fig. 13) for user identification [30]. The system includes a tactile sensing mat and a separate ESP32-controlled lamp. The mat captures tactile data, which is wirelessly transmitted using ESP-NOW to another microcontroller responsible for turning the lamp on when an authorized user is standing on it. In future iterations, WiReSens Toolkit could be used to enable seamless integration with other smart home devices, such as doorbells or security cameras.

7 Discussion

In this section, we summarize the recommended authoring workflows for the toolkit based on our evaluations and discuss its primary limitations. We further outline several avenues for future works that WiReSens Toolkit’s contributions enable.

7.1 Toolkit Recommendations

Table 2 summarizes the wireless protocol, calibrated potentiometer step (R_{pot}), and intermittent parameters (p, d) used across our demonstrated applications. From this organization and our technical evaluations, several design recommendations emerge for authoring with the toolkit.

1. Identify application context and system-level design considerations. Before configuring the toolkit, developers should first consider the broader application context. Key factors include the number of devices operating simultaneously, whether the deployment occurs indoors or outdoors, the expected frequency and magnitude of interaction forces, and the required duration of continuous operation. These parameters directly inform choices of communication protocol, sensitivity calibration, and intermittent sending schedules for low-power operation.

Based on our evaluations, the toolkit can configure tactile sensing systems to record activities at up to 60 fps when WiFi is enabled, which is sufficient for capturing fast tapping and most instrument or tool interactions, though it may not fully capture high-impact forces or fine vibrations. From a sensitivity standpoint, the piezoresistive sensors tested with our toolkit perform best within the 0.3 – 50 N range. This is adequate for pressure levels up to firm grips or squeezes, but less suitable for high-force activities such as heavy lifting or jump landings.

2. Selecting a wireless protocol. We recommend choosing a wireless protocol according to the intended sensing context. From our experiments and prototypes we find:

- **Wi-Fi** is preferred for indoor systems requiring low latency or multi-device synchronization, such as our musical gloves application. The WiReSens Toolkit captured tapping gestures across two sensors at more than 30 frames per second, allowing audio feedback to feel real time.
- **ESP-NOW** is suitable for multi-device interactions in mobile or outdoor scenarios due to its independence from internet access and higher throughput compared to BLE. We find it especially useful for IoT scenarios, such as the smart welcome mat.
- **BLE** is best suited for energy-constrained or wearable devices designed for long-term continuous capture of lower-frequency signals (1–10 Hz), such as our gait monitoring shoe sole.

3. Sensitivity calibration and R_{pot} tuning. After selecting a protocol, users should run the toolkit’s auto-calibration routine while performing representative actions to set an appropriate gain resistor value. The reported R_{pot} values in Table 2 may serve as reliable starting points for similarly fabricated sensors and applications.

4. Intermittent transmission and tuning of (p, d) . For applications prioritizing energy efficiency or long recording durations, intermittent sensing can be optimized by using our provided optimization script with a recorded sample of representative activity data. Our experiments found that effective configurations typically lie within $p \in [20, 80]$ and $d \in [20, 40]$, with a weighting parameter $\alpha = 0.75$ to emphasize signal fidelity over transmission rate. Based on simulations and empirical results, we recommend applying this optimization when data collection exceeds six hours or when battery longevity is a key design constraint.

Together, these recommendations address the key design trade-offs observed in our evaluations and provide practical starting points for adapting the toolkit to new sensing contexts.

7.2 Limitations

While WiReSens Toolkit can be used in remote settings, portability may be limited by the fact that BLE and ESP-NOW communication are fundamentally lower-throughput, lower-range protocols [12] and backend support is restricted to desktop computers. This can be addressed by adding support for multiple receivers to balance the data load more effectively and implementing the backend within a mobile application. Additionally, we chose intermittent data transmission as a computationally efficient and intuitive means of power

reduction that would not hinder sensing capability, but energy savings may be limited when compared to battery-free systems [20, 21]. Future work may explore ultra-low-power wake-on-motion sensing or energy harvesting techniques to extend operational lifetime without compromising sensing fidelity. Moreover, while the ESP32 microcontroller was chosen for accessibility, it imposes constraints on sample rate, power consumption, and form factor. Using ASIC or FPGA-based sensor readout could enhance the system’s ability to detect finer-grained or fast-changing pressures, such as vibrations, while reducing power consumption and size. Finally, resistive sensors are prone to drift and are individually affected by environmental factors, such as temperature and humidity, as well as by mechanical strain. Prior studies [18, 56] have explored methods to quantify and decouple these effects. While WiReSens does not currently implement compensation for these factors, its open-source design provides a foundation for integrating future calibration techniques.

7.3 Future Work

While this work focuses on resistive sensors for tactile interactions, the WiReSens Toolkit can extend to other sensing modalities that rely on conductivity changes, such as thermistors for temperature, moisture sensors, chemiresistors for gas detection, and photoresistors for light intensity. We see this adaptability as enabling large-scale monitoring of multimodal physical phenomena beyond just touch. Additionally, future work could integrate on-device learning to dynamically optimize toolkit parameters for specific applications. For instance, in a wearable health monitoring application, the system could adjust sensor calibration in response to physiological changes like swelling or muscle tension, ensuring long-term accuracy. Finally, we envision this work advancing tactile sensing as a privacy-conscious and efficient alternative to vision-based systems, as resistive tactile sensing data is lower resolution than RGB camera data.

8 Conclusion

We introduced the WiReSens Toolkit, designed to facilitate the creation of portable, adaptive, and long-lasting resistive-matrix-based tactile sensing devices. Our implementation supports variably-sized resistive sensor array readout, wireless communication via Wi-Fi, BLE, or ESP-NOW protocols, adaptive sensitivity through auto-calibration, and power efficiency through intermittent computing techniques. Additionally, it streamlines the logging and real-time custom visualization of tactile sensing data from multiple devices. We conducted a technical evaluation of WiReSens Toolkit’s functionality, providing evidence of its ability to enable fast and robust multi-sender communication, maintain consistent sensor performance across pressure intensities, and ensure power-efficient operation. Application prototypes and a comprehensive user study further validate the interface as a usable and useful tool for developing interactive systems based on the resistive sensing principle.

We believe that WiReSens Toolkit will seamlessly integrate with ongoing advancements in digital manufacturing, wireless sensor networks, and artificial intelligence, empowering the next generation of tactile sensing technologies for deeper insights into human interactions with the physical world.

References

- [1] Roland Aigner, Andreas Pointner, Thomas Preindl, Patrick Parzer, and Michael Haller. 2020. Embroidered Resistive Pressure Sensors: A Novel Approach for Textile Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376305
- [2] Ahmed H. Anwer, Nabeel Khan, Muhammad Z. Ansari, Seung-Su Baek, Hwan Yi, Sangjin Kim, Seung-Mo Noh, and Chan Jeong. 2022. Recent Advances in Touch Sensors for Flexible Wearable Devices. *Sensors (Basel)* 22, 12 (Jun 13 2022), 4460. doi:10.3390/s22124460
- [3] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: adding an adjective rating scale. *J. Usability Studies* 4, 3 (May 2009), 114–123.
- [4] Aaron Bangor, Philip T. Kortum, and James T. Miller and. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24, 6 (2008), 574–594. arXiv:https://doi.org/10.1080/10447310802205776 doi:10.1080/10447310802205776
- [5] Bijender and Ashok Kumar. 2022. Recent progress in the fabrication and applications of flexible capacitive and resistive pressure sensors. *Sensors and Actuators A: Physical* 344 (2022), 113770. doi:10.1016/j.sna.2022.113770
- [6] Bluetooth Special Interest Group (SIG). 2014. *Bluetooth Core Specification Version 4.2*. <https://www.bluetooth.com/specifications/bluetooth-core-specification/> Accessed: 2024-08-16.
- [7] John Brooke. 1996. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland (Eds.). Taylor and Francis, London, 189–194.
- [8] Wufan Chen and Xin Yan. 2020. Progress in achieving high-performance piezoresistive and capacitive flexible pressure sensors: A review. *Journal of Materials Science & Technology* 43 (2020), 175–188. doi:10.1016/j.jmst.2019.11.010
- [9] Tommaso D'Alessio. 1999. Measurement errors in the scanning of piezoresistive sensors arrays. *Sensors and Actuators A: Physical* 72, 1 (1999), 71–76. doi:10.1016/S0924-4247(98)00204-0
- [10] Artem Dementyev, Hsin-Liu (Cindy) Kao, and Joseph A. Paradiso. 2015. SensorTape: Modular and Programmable 3D-Aware Dense Sensor Network on a Tape. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) (UIST '15). Association for Computing Machinery, New York, NY, USA, 649–658. doi:10.1145/2807442.2807507
- [11] Maurin Donneaud, Cedric Honnet, and Paul Strohmeier. 2017. Designing a Multi-Touch eTextile for Music Performances. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Aalborg University Copenhagen, Copenhagen, Denmark, 7–12. doi:10.5281/zenodo.1176151
- [12] Dania Eridani, Adian Fatchur Rochim, and Faiz Noerdiyan Cesara. 2021. Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance. In *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*. 322–328. doi:10.1109/iSemantic52711.2021.9573246
- [13] Espressif Systems. 2024. *ESP-NOW on ESP32 FAQ*. <https://docs.espressif.com/projects/esp-faq/en/latest/software-framework/protocols/lwip.html> Accessed: 2024-08-16.
- [14] Martin Feick, Cihan Biyikli, Kiran Gani, Anton Wittig, Anthony Tang, and Antonio Krüger. 2023. VoxelHap: A Toolkit for Constructing Proxies Providing Tactile and Kinesthetic Haptic Feedback in Virtual Reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 104, 13 pages. doi:10.1145/3586183.3606722
- [15] Niklas Fiedler et al. 2022. Low-cost fabrication of flexible tactile sensor arrays. *HardwareX* 12 (2022), e00372. doi:10.1016/j.ohx.2022.e00372
- [16] Tobias Grosse-Puppenthal, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 3293–3315. doi:10.1145/3025453.3025808
- [17] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 2024. 3D ViTac: Learning Fine-Grained Manipulation with Visuo-Tactile Sensing. In *Proceedings of Robotics: Conference on Robot Learning (CoRL)*.
- [18] Chao Jiang, Wenjun Xu, Yanan Li, et al. 2024. Capturing forceful interaction with deformable objects using a deep learning-powered stretchable tactile array. *Nature Communications* 15 (2024), 9513. doi:10.1038/s41467-024-53654-y
- [19] Jong-Seok Kim, Dae-Yong Kwon, and Byong-Deok Choi. 2016. High-Accuracy, Compact Scanning Method and Circuit for Resistive Sensor Arrays. *Sensors* 16, 2 (2016). doi:10.3390/s16020155
- [20] Vito Kortbeek, Abu Bakar, Stefany Cruz, Kasim Sinan Yildirim, Przemysław Pawelczak, and Josiah Hester. 2020. Bfree: Enabling battery-free sensor prototyping with python. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–39.
- [21] Christopher Kraemer, Amy Guo, Saad Ahmed, and Josiah Hester. 2022. Battery-free makecode: Accessible programming for intermittent computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–35.
- [22] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3173574.3173610
- [23] Jialin Li, Naveen Kr. Sharma, Dan R. K. Ports, and Steven D. Gribble. 2014. Tales of the Tail: Hardware, OS, and Application-level Sources of Tail Latency. In *Proceedings of the ACM Symposium on Cloud Computing* (Seattle, WA, USA) (SOCC '14). Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/2670979.2670988
- [24] Ruiqing Li, Qun Zhou, Yin Bi, Shaojie Cao, Xue Xia, Aolin Yang, Siming Li, and Xueliang Xiao. 2021. Research progress of flexible capacitive pressure sensor for sensitivity enhancement approaches. *Sensors and Actuators A: Physical* 321 (2021), 112425. doi:10.1016/j.sna.2020.112425
- [25] Jingfeng Liu, Xuan Zhang, Jintao Liu, Xingang Liu, and Chuhong Zhang. 2024. 3D Printing of Anisotropic Piezoresistive Pressure Sensors for Directional Force Perception. *Advanced Science* 11, 24 (2024), 2309607. doi:10.1002/adv.202309607
- [26] Xin Liu, Chen Zhao, Bin Zheng, Qinwei Guo, Xiaoqin Duan, Aziguli Wulamu, and Dezheng Zhang. 2021. Wearable Devices for Gait Analysis in Intelligent Healthcare. *Frontiers in Computer Science* 3 (2021). doi:10.3389/fcomp.2021.661676
- [27] Yiyue Luo, Yunzhu Li, Michael Foshey, Wan Shou, Pratyusha Sharma, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. 2021. Intelligent carpet: Inferring 3d human pose from tactile signals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11255–11265.
- [28] Yiyue Luo, Yunzhu Li, Pratyusha Sharma, Wan Shou, Kui Wu, Michael Foshey, Beichen Li, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. 2021. Learning human–environment interactions using conformal tactile textiles. *Nature Electronics* 4, 3 (2021), 193–201.
- [29] Payal S. Malvade, Atul K. Joshi, and Swati P. Madhe. 2017. IoT based monitoring of foot pressure using FSR sensor. In *2017 International Conference on Communication and Signal Processing (ICCCSP)*. 0635–0639. doi:10.1109/ICCCSP.2017.8286435
- [30] Denys J. C. Matthies, Don Samitha Elvitigala, Sachith Muthukumarana, Jochen Huber, and Suranga Nanayakkara. 2019. CapMat: A Smart Foot Mat for User Authentication. In *Proceedings of the 10th Augmented Human International Conference 2019 (Reims, France) (AH2019)*. Association for Computing Machinery, New York, NY, USA, Article 42, 2 pages. doi:10.1145/3311823.3311874
- [31] Devyani McLaren, Jian Gao, Xiulun Yin, Rúbia Reis Guerra, Preeti Vyas, Chrys Morton, Xi Laura Cang, Yizhong Chen, Yiyuan Sun, Ying Li, John David Wyndham Madden, and Karon E. MacLean. 2024. What is Affective Touch Made Of? A Soft Capacitive Sensor Array Reveals the Interplay between Shear, Normal Stress and Individuality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 52, 31 pages. doi:10.1145/3654777.3676346
- [32] Anat Mirelman, Paolo Bonato, Richard Camicioli, Terry D Ellis, Nir Giladi, Jamie L Hamilton, Chris J Hass, Jeffrey M Hausdorff, Elisa Pelosin, and Quincy J Almeida. 2019. Gait impairments in Parkinson's disease. *The Lancet Neurology* 18, 7 (2019), 697–708. doi:10.1016/S1474-4422(19)30044-4
- [33] Marie Muehlhaus, Jürgen Steimle, and Marion Koelle. 2022. Feather Hair: Interacting with Sensorized Hair in Public Settings. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference (Virtual Event, Australia) (DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1228–1242. doi:10.1145/3532106.3533527
- [34] Devin Murphy, Yichen Li, Crystal Elaine Owens, Layla Stanton, Paul Pu Liang, Yiyue Luo, Antonio Torralba, and Wojciech Matusik. 2025. Fits like a Flex-Glove: Automatic Design of Personalized FPCB-Based Tactile Sensing Gloves. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 280, 8 pages. doi:10.1145/3706599.3720147
- [35] Maggie Orth, Joshua R. Smith, E. Rehmi Post, J. A. Strickon, and Emily B. Cooper. 1998. Musical jacket. In *ACM SIGGRAPH 98 Electronic Art and Animation Catalog*. 38.
- [36] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoedauer, Martin Kaltenbrunner, Siegfried Bauer, and Michael Haller. 2018. RESI: A Highly Flexible, Pressure-Sensitive, Imperceptible Textile Interface Based on Resistive Yarns. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 745–756. doi:10.1145/3242587.3242664
- [37] Narjes Pourjafarian, Anusha Withana, Joseph A Paradiso, and Jürgen Steimle. 2019. Multi-Touch Kit: A do-it-yourself technique for capacitive multi-touch sensing using a commodity microcontroller. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1071–1083.

- [38] Zhongnan Pu, Daisuke Miyata, and Lei Jing. 2024. An Efficient Method for Calibrating Piezoresistive Pressure Sensor. In *2024 International Conference on the Challenges, Opportunities, Innovations and Applications in Electronic Textiles (E-Textiles)*. 318–323. doi:10.23919/E-Textiles63767.2024.10914233
- [39] Soonjae Pyo, Jaeyong Lee, Wondo Kim, Eunhwan Jo, and Jongbaeg Kim. 2019. Multi-Layered, Hierarchical Fabric-Based Tactile Sensors with High Sensitivity and Linearity in Ultrawide Pressure Range. *Advanced Functional Materials* 29, 35 (2019), 1902484. doi:10.1002/adfm.201902484
- [40] Blaine Rothrock, Alexander Curtiss, Juyang Bai, and Josiah Hester. 2023. Towards a Toolkit for Free Living Wearable Development. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers* (Cambridge, United Kingdom) (*UbiComp/ISWC '22 Adjunct*). Association for Computing Machinery, New York, NY, USA, 379–385. doi:10.1145/3544793.3560399
- [41] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) (*CHI '12*). Association for Computing Machinery, New York, NY, USA, 483–492. doi:10.1145/2207676.2207743
- [42] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 579–588.
- [43] Ridha Soua and Pascale Minet. 2011. A survey on energy efficient techniques in wireless sensor networks. In *2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)*. 1–9. doi:10.1109/WMNC.2011.6097244
- [44] Gidugu Lakshmi Srinivas and Sherjeel Khan. 2024. Improving Soft Capacitive Tactile Sensors: Scalable Manufacturing, Reduced Crosstalk Design, and Machine Learning. In *2024 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*. 1–4. doi:10.1109/FLEPS61194.2024.10604006
- [45] Becky Stern. 2013. Firewalker LED Sneakers: Make Velostat Step Sensors. <https://learn.adafruit.com/firewalker-led-sneakers/make-velostat-step-sensors>. Accessed: 2024-08-27.
- [46] Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. 2018. zPatch: Hybrid Resistive/Capacitive eTextile Input. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* (Stockholm, Sweden) (*TEI '18*). Association for Computing Machinery, New York, NY, USA, 188–198. doi:10.1145/3173225.3173242
- [47] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. 2019. Learning the signatures of the human grasp using a scalable tactile glove. *Nature* 569, 7758 (2019), 698–702.
- [48] Shikhar Suryavansh, Abu Benna, Chris Guest, and Somali Chaterji. 2021. A data-driven approach to increasing the lifetime of IoT sensor nodes. *Scientific Reports* 11, 1 (2021), 22459. doi:10.1038/s41598-021-01431-y
- [49] Espressif Systems. n.d.. ESP-NOW Data Send API. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html#send-esp-now-data. Accessed: 2025-11-06.
- [50] Espressif Systems. n.d.. What is the retransmission interval of TCP? <https://docs.espressif.com/projects/esp-faq/en/latest/software-framework/wifi.html#wip-what-is-the-retransmission-interval-of-tcp>. Accessed: 2025-11-06.
- [51] Fernando Vidal-Verdú, Óscar Oballe-Peinado, José A. Sánchez-Durán, Julián Castellanos-Ramos, and Rafael Navas-González. 2011. Three Realizations and Comparison of Hardware for Piezoresistive Tactile Sensors. *Sensors* 11, 3 (2011), 3249–3266. doi:10.3390/s110303249
- [52] Irmandy Wicaksono, Don Derek Haddad, and Joseph Paradiso. 2022. Tapis Magique: Machine-knitted Electronic Textile Carpet for Interactive Choreomusical Performance and Immersive Environments. In *Proceedings of the 14th Conference on Creativity and Cognition* (Venice, Italy) (*C&C '22*). Association for Computing Machinery, New York, NY, USA, 262–274. doi:10.1145/3527927.3531451
- [53] Irmandy Wicaksono, Peter G Hwang, Samir Droubi, Franny Xi Wu, Allison N Serio, Wei Yan, and Joseph A Paradiso. 2022. 3dknits: Three-dimensional digital knitting of intelligent textile sensor for activity recognition and biomechanical monitoring. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2403–2409.
- [54] Irmandy Wicaksono and Joseph Paradiso. 2020. KnittedKeyboard: Digital Knitting of Electronic Textile Musical Controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Romain Michon and Franziska Schroeder (Eds.). Birmingham City University, Birmingham, UK, 323–326. doi:10.5281/zenodo.4813391
- [55] Raphael Wimmer, Matthias Kranz, Sebastian Boring, and Albrecht Schmidt. 2007. A capacitive sensing toolkit for pervasive activity detection and recognition. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*. IEEE, 171–180.
- [56] Chen Xu, Yiran Wang, Jingyan Zhang, Ji Wan, Zehua Xiang, Zhongyi Nie, Jie Xu, Xiang Lin, Pengcheng Zhao, Yaozheng Wang, Shaotong Zhang, Jing Zhang, Chunxiu Liu, Ning Xue, Wei Zhao, and Mengdi Han. 2024. Three-dimensional micro strain gauges as flexible, modular tactile sensors for versatile integration with micro- and macroelectronics. *Science Advances* 10, 34 (2024), eadp6094. arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.adp6094 doi:10.1126/sciadv.adp6094
- [57] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. 2017. GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force. *Sensors* 17, 12 (2017). doi:10.3390/s17122762
- [58] Shulin Zhao, Prasanna Venkatesh Rengasamy, Haibo Zhang, Sandeepa Bhuyan, Nachiappan Chidambaram Nachiappan, Anand Sivasubramaniam, Mahmut Taylan Kandemir, and Chita Das. 2019. Understanding Energy Efficiency in IoT App Executions. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 742–755. doi:10.1109/ICDCS.2019.00079
- [59] Junyi Zhu, Jackson C Snowden, Joshua Verdejo, Emily Chen, Paul Zhang, Hamid Ghaednia, Joseph H Schwab, and Stefanie Mueller. 2021. EIT-kit: An electrical impedance tomography toolkit for health and motion sensing. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 400–413.