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

Devin Murphy
MIT CSAIL
Cambridge, USA
devinmur@mit.edu

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

Paul Liang
MIT Media Lab and EECS
Cambridge, USA
ppliang@mit.edu

Wojciech Matusik
MIT CSAIL
Cambridge, USA
wojciech@csail.mit.edu

Yiyue Luo
University of Washington ECE
Seattle, USA
yiyueluo@uw.edu

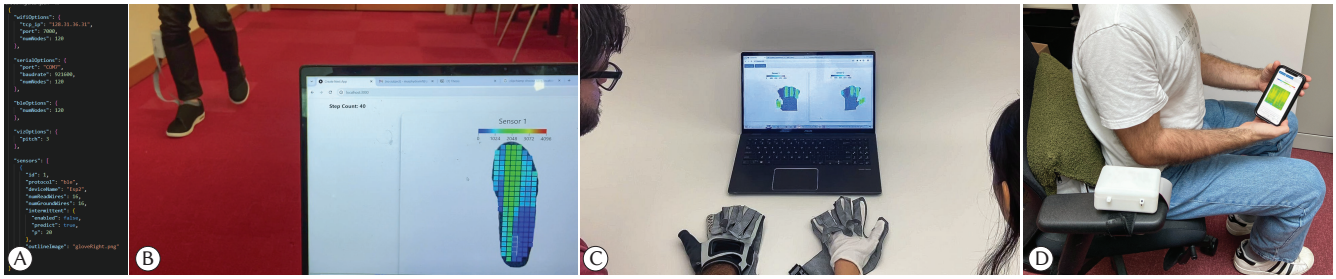


**Figure 1: WiReSens Toolkit provides open-source hardware and software to enable the development of portable, adaptive, and efficient resistive tactile sensing systems.**

## Abstract

Tactile sensors present a powerful means of capturing, analyzing, and augmenting human-environment interactions. Accelerated by advancements in design and manufacturing, resistive matrix-based sensing has emerged as a promising method for developing scalable and robust tactile sensors. However, the development of portable, adaptive, and long lasting resistive tactile sensing systems remains a challenge. To address this, we introduce WiReSens Toolkit. Our platform provides open-source hardware and software libraries to configure multi-sender, power-efficient, and adaptive wireless tactile sensing systems in as fast as ten minutes. We demonstrate our platform's flexibility by using it to prototype several applications such as musical gloves, gait monitoring shoe soles, and IoT-enabled smart home systems.

## Keywords

Tactile Sensing, Embedded Programming, accessible, open source

## 1 Introduction

Our sense of touch encodes a wealth of information about how we interact with our physical world. Tactile sensors endow both humans and everyday objects with enhanced ability to capture, understand, and augment these interactions, increasingly finding applications within cyber-physical systems and the Internet of Things (IoT) [2, 22]. Among the various existing methods for tactile sensing, recent research has adopted resistive matrix-based approaches [20, 34, 40] where a piezoresistive material is sandwiched between two orthogonal electrode arrays to measure applied pressure via resistance changes. This approach is appealing because it reduces the need for extensive wiring [19], and the sensors themselves can be fabricated using scalable manufacturing techniques like machine knitting, digital embroidery, and 3d printing [1, 40, 43].

To realize a vision of ubiquitous tactile sensing for large-scale monitoring of human-environment interactions, scalable infrastructure beyond sensor manufacturing is essential —researchers need robust tools for effective system deployment. Large scale sensing requires consideration of additional factors that make this deployment difficult: **portability**—devices must remain untethered from computational

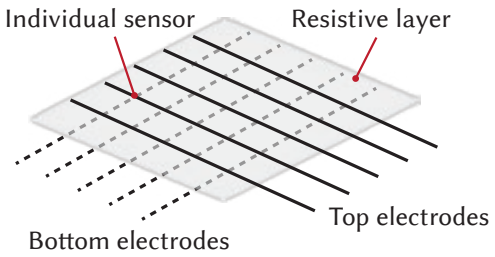


**Figure 2: Typical layout of a resistive matrix-based pressure sensing form factor, with a resistive layer sandwiched between two sets of orthogonally positioned electrodes.**

units for easy repositioning; **adaptability**—devices should be repurposable for various form factors, user needs, and application contexts; and **efficiency**—devices must operate over extended periods to capture long-term interaction patterns. As it stands, developing a tactile sensing system that meets all these requirements is technically demanding, limiting the accessibility and broader adoption of tactile sensing interfaces within a larger research community.

To adress these challenges, we introduce WiReSens Toolkit (Fig. 1). WiReSens Toolkit includes open-source [1] Arduino and Python libraries to configure an adaptive tactile sensor readout circuit via a single JSON file, serving as a flexible interface with wireless deployment. In contrast to existing open-source solutions for tactile sensing [7, 27, 28], WiReSens Toolkit's Arduino library includes abstractions for sensing array readout, wireless communication via three distinct protocols (Wi-Fi, Bluetooth Low Energy (BLE), and ESP-NOW), automatic sensitivity calibration, and energy-efficient operation modes. Additionally, our open-source Python library offers customizable interactive visualizations and methods for recording and playback of tactile sensing data from multiple devices in real-time. We offer a technical evaluation of WiReSens Toolkit which evidences its ability to provide robust multi-sender wireless communication across all three protocols, optimized pressure resolution during sensor readout, and up to a 42% increase in tactile sensing device lifetime, depending on the protocol used. We finally demonstrate the flexibility of our toolkit by using it to quickly develop (< 10 minutes each) 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:

- An open-source adaptive readout circuit and Arduino-based firmware library for programming microcontrollers to interface with resistive sensing arrays;
- A Python-based receiver library for logging and interactive visualization of tactile sensing data with a standard computer;
- A technical evaluation of the wireless communication, readout adaptivity, and power reduction capabilities of our implementation;
- A demonstration of the capabilities of WiReSens Toolkit to assist in prototyping a variety of tactile sensing applications, including musical gloves, a gait tracking shoe sole, a repurposable interactive pillow, and a welcome mat for home IoT integration.

[1] WiReSens Toolkit on GitHub

## 2 Tactile Sensing Overview

Tactile sensors are a well-established technology. They can be categorized by their employed sensing principle, with optical, capacitive, and resistive methods being among the most widely used.

Optical or vision-based tactile sensors detect touch by analyzing changes in light patterns or images caused by the deformation of a material in response to pressure. One example is the elastomer-based vision sensor GelSight [44], now commonly used for dexterous manipulation tasks in commercial robots. While these vision-based methods can be high-resolution and performant [25], they often require proprietary hardware and have a fundamental limit in their ability to scale to a large coverage area and a flexible form factor due to visual occlusion complications. Consequently, many researchers have turned their attention to capacitive and resistive-based methods for ubiquitous and wearable sensing applications.

In capacitive methods, two conductive electrodes typically sandwich an insulating dielectric layer. Applied pressure deforms the dielectric layer and subsequently increases the measured capacitance between the paired electrodes [16]. An analogous principle is used in resistive-based methods, where the electrodes sandwich a piezoresistive material instead, and the applied pressure is measured through the change in resistance [3]. In both methods, the electrodes on the top and bottom layers are commonly repeated row-wise and column-wise in an orthogonal fashion to form a two-dimensional "sensing matrix" (Fig. 2), where individual sensors are formed at the intersections of top and bottom electrodes. 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 advancements in fabrication [1, 21, 26, 40], it has allowed resistive and capacitive sensing to be incorporated in a variety of form factors like gloves [17, 34], carpets [20, 39], and pants [30].

Compared with resistive sensing, capacitive sensing can require more complex readout circuit requirements [5, 38]

| | CapToolKit [42] | Midas [29] | Multi-Touch Kit [28] | E256 [7] | Ours |
|---|---|---|---|---|---|
| **Sensing principle** | Capacitive | Capacitive | Capacitive | Resistive | Resistive |
| **# of sensors per device** | 8 | 9 | 256 | 256 | 1024 |
| **# of sending devices** | 1 | 1 | 1 | 1 | 5 |
| **Sensitivity calibration** | No | No | No | No | Yes |
| **Wireless protocol support** | None | None | None | None | Wi-Fi, BLE, Esp-Now |
| **Power-saving mode** | None | None | None | None | Yes |
| **Visualization** | Static | Configurable | Static | Static | Configurable |
| **Open-source software** | Yes | Yes | Yes | Yes | Yes |
| **Open-source hardware** | Yes | Yes | N/A | Yes | Yes |

**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.**

and shielding layers [32] in order to combat electromagnetic noise, and tools for implementing it are better explored in literature [11, 28, 29]. Therefore, we focus the scope of this work on resistive pressure sensing matrices.

## 3 Related Work

Toolkits are widely used to facilitate the creation of new interactive sensing systems. As Ledo et al. [15] note, the value of toolkits commonly lie in their ability to simplify access to complex algorithms and enable fast prototyping of software and hardware interfaces. Our literature review compares WiReSens Toolkit with tools specifically designed to ease the development of tactile sensing systems, with a focus on innovations in readout circuitry, sensor operation firmware, and data visualization software (Table 1).

***Open-source Readout Circuits.*** Pressure signals are typically read from a resistive sensing matrix using a zero potential scanning readout circuit [6, 13], the hardware on which WiReSens Toolkit is also built (Fig. 4B). In the zero-potential method, each layer of electrodes is connected to a digital multiplexer (MUX). To measure resistance at a specific sensing node in the array, one MUX grounds an electrode in one layer, while the other MUX connects an electrode and op-amp in series in the opposing layer to an Analog-to-Digital Converter (ADC). The op-amp amplifies the voltage at this electrode, which is correlated with the resistance at the intersection of the two electrodes, and the ADC digitizes this value. The circuit topology is well established and widely adopted due to its ability to reduce crosstalk between neighboring electrodes [6], and many works provide this hardware as open-source [7, 10, 12]. Compared with these works, WiReSens Toolkit is the first to provide manufacturing files for a readout circuit which both supports a density as high as 32x32 (1024 pressure sensing locations), and has a digital potentiometer for tuning sensitivity.

***Firmware Toolkits.*** Firmware toolkits often assist users by abstracting the low-level programming typically required to configure microcontroller units (MCUs) for high-fidelity, real-time wireless sensing applications. One existing open-source firmware solution for tactile sensing is the Multi-Touch kit [28], which provides an Arduino library for hardware-agnostic readout of mutual capacitance-based multi-touch arrays. While this solution does not require specialized hardware, it lacks auto-calibration features for sensitivity adjustment and built-in wireless transmission, both of which are included in WiReSens Toolkit. The work most similar to ours is the "E256" GitHub repository by Donneaud et al. [7], with open-source Arduino code for denoised readout of resistive matrix-based pressure sensing eTextiles. Similarly to the Multi-Touch kit, it does not include built-in wireless communication abstractions. Additionally, this firmware is tailored for the specific eTextile interface accompanying its release, in contrast to WiReSens Toolkit which was designed for general resistive matrix readout across different form factors.

***Visualizations.*** Visualizations play a critical role in tactile system development, aiding in debugging issues such as electrode shorting and enhancing data interpretation. The most readily available open-source visualization tools for tactile sensing are Processing software from the E256 project [7] and the Multi-Touch Kit [28]. While these tools display real-time pressure data, they lack native support for multiple devices and offer limited configuration options, which restricts their adaptability to various form factors. In contrast, the WiReSens Toolkit supports both multi-device monitoring and configurable visualizations, providing a scalable

, ,

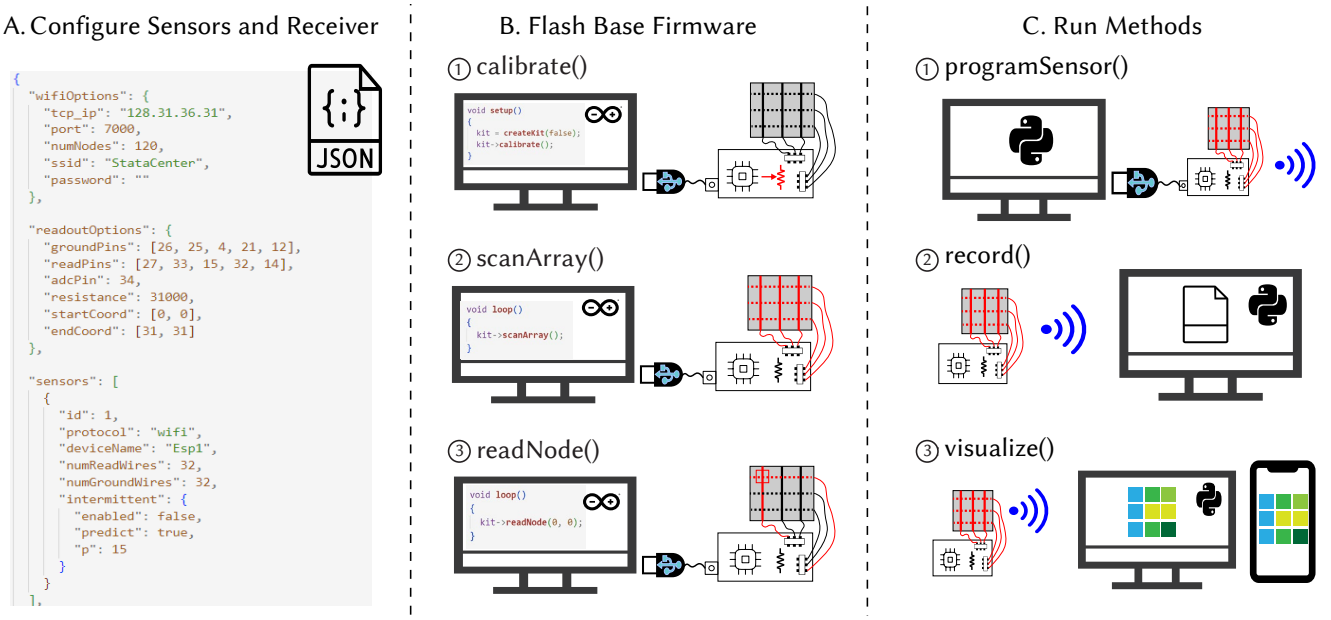


**Figure 3: Typical workflow using WiReSens Toolkit to develop tactile sensing applications. Users (A) set parameters via a JSON file, (B) flash a base firmware program to the tactile sensing device using the Arduino library to specify what on-device methods should run once configured, and (C) use the Python library to run various methods on the sensing devices and the receiver according to the JSON configuration.**

and physically intuitive representation of pressure data. Although the capacitive touch sensing toolkit Midas [29] supports custom visualizations, they require HTML, CSS, and JavaScript for implementation, adding to development time. WiReSens Toolkit simplifies this process with a built-in web interface that offers drag-and-drop functionality and JSON-based configuration right out of the box.

## 4 The WireSens Toolkit

To make the WiReSens Toolkit accessible and effective for developing scalable tactile sensing systems, we outline several key goals for its design

- **Portability:** Tactile sensing devices often must operate untethered from compute-intensive hardware so that they may be mobile and easily repositioned. This is particularly crucial in wearable applications, where wires can restrict the user's natural range of motion. Additionally, the availability and reliability of wireless protocols can vary significantly depending on the environment. For instance, while Wi-Fi offers excellent speed and reliability for indoor wireless communication, it may be difficult to access in outdoor settings. A toolkit that simplifies the process of switching between different protocols would expand the range of environments where tactile sensing can be effectively implemented.
- **Adaptivity:** Ubiquitous tactile sensing interfaces must be adaptable to various form factors, users, and application scenarios. At a base level, they need to support flexible readout sizes and configurations from the same sensor form to ensure minimal updates to hardware and firmware. Optimizing the amplification of the pressure signals for different applications is also important — if they are not amplified enough we may underutilize the full range of the ADC, and if they are amplified too much the output signal may saturate at a constant value [37]. Additionally, resistive sensing arrays are often displayed statically, with each sensing node fixed to a specific position. Without interacting directly with the sensor, it can be challenging to map visualization areas to corresponding physical regions on the interface, complicating precise mapping. This mapping may also shift when the same form factor is used by different individuals or over time. An adaptive visualization that updates in real-time to these changes would enable more accurate monitoring and assist with sensor adjustments during operation.
- **Efficiency:** Many human-environment interactions can only be captured after long-time monitoring, making the lifetime of tactile sensing devices an important consideration. However, wireless communication is very power intensive, posing a big challenge for sustainable development [31, 45]. Frequently, low-power

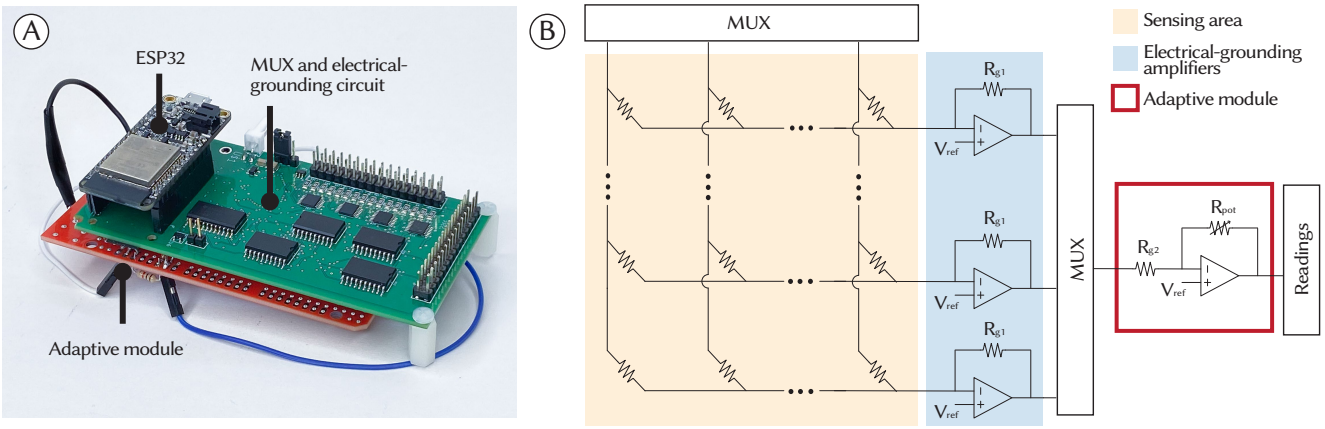


**Figure 4: (A) Hardware open-sourced by WiReSens Toolkit including an ESP32 microcontroller, zero potential readout circuit, and adaptive module and (B) Schematic of general zero potential readout circuit (left) with additional opamp and digital potentiometer for adaptivity (red).**

operation of MCU's can decrease computational capability, so users need to be able to have access to methods that allow them to navigate this trade-off. In addition, firmware that enables reuse of the same hardware with different form factors for different applications can ensure more efficient use of resources.

## 4.1 Implementation Overview

The implementation of WiReSens Toolkit comprises three main components: a JSON file for system configuration (Fig. 3A), a zero-potential readout circuit programmable via an accompanying Arduino library (Fig. 3B), and a Python library for managing sensing data processing via a local desktop computer (Fig. 3C). Using these tools, we streamline resistive tactile sensing development into three main steps. First, users specify configuration parameters such as wire layout, communication protocol settings, and visualization details via the JSON file. They then leverage the Arduino library to create and upload a base firmware program that specifies what the readout circuit will do once configured. Finally, the Python library is used to pass configuration parameters from the JSON file to the tactile sensing device, and execute methods for data logging and visualization. An overview of this main functionality is exhibited in Fig. 3.

To ensure adaptability to various sensor types, our platform does not mandate a specific fabrication approach for the resistive sensing matrix. Instead, we demonstrate the capability of our toolkit to integrate with sensors manufactured using standard techniques: high-fidelity digital embroidery [1] and low-fidelity taping of electrodes onto Velostat [33], in this paper's application section (Sec. 6).

## 4.2 JSON Configuration

Users configure their systems primarily through a JSON file which contains a variety of options for wireless communication, interactive visualization, and sensor readout. An overview of these options is available in our GitHub repository [2]. Our incorporation of JSON promotes the development of tactile sensing systems which can change with minimal effort — after a tactile sensing device has been flashed with initial base firmware (Fig.3B), users can simply update the file (Fig.3A) and pass it to the device (Fig.3C) to switch communication protocols, change the area of readout, increase sensor sensitivity, and much more.

## 4.3 Adaptive Readout Circuit

To allow tuning of tactile sensor sensitivity and avoid the problems of ADC underutilization and op-amp output saturation, we introduce an additional module between the readout MUX and the ADC in a standard zero-potential readout circuit. This module, highlighted in red in Fig. 4, incorporates an operational amplifier (TLV9152, Texas Instruments) configured with variable gain, managed by a 50 kΩ digital potentiometer (MCP4018, Microchip), which we denote as $R_{\text{pot}}$. This setup adjusts the voltage output range from $[V_{\text{ref}}, 3.3V]$ to $[0V, V_{\text{ref}}]$, inverts the relationship between applied pressure and output voltage due to the op-amp's inverting configuration, and most critically, enables adaptive sensitivity control by adjusting $R_{\text{pot}}$. The readout circuit is controlled by an ESP32 microcontroller (Adafruit Feather ESP32), and the gerber files, schematic, and bill of materials for manufacturing this circuit are available on our GitHub repository.

[2] WiReSens Toolkit on GitHub

## 4.4 Arduino Library

Our Arduino library allows users to create and upload a base firmware program to the ESP32 that specifies what it will do once configured. It contains methods for sensor readout, intermittent wireless communication, and automatic sensitivity calibration.

***Sensor Readout Methods.*** We provide two base methods for reading and transmitting pressure data from sensor arrays: *scanArray()*, and *readNode()*. The *scanArray()* method reads sensor values within an area defined by two coordinates in the form (*readWire*, *groundWire*), where *readWire* and *groundWire* are integers corresponding to the wire selected by each MUX when receiving the integer as input. *readNode()* is the more flexible sensor readout option, allowing users to specify a single coordinate in the sensing array, given in the form (*readWire*, *groundWire*). This method can be combined with basic programming logic to perform readouts in any order or pattern across the sensing array. These methods wirelessly send the obtained pressure data to a computer using the chosen communication protocol, either continuously or intermittently.

***Intermittent Sending.*** In intermittent operation, power is saved by minimizing data transmission for signals that are easy to predict or unchanging. The readout methods predict the voltage at a sensor node before reading it using the window-based time series forecasting equation:

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

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 those readings and predictions is below some user defined threshold $d$. If the data packet is not sent, the WiReSens Toolkit 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. [35], which was chosen due to its low computational complexity and memory requirements, making it ideal for implementation on a microcontroller.

To help select the parameter $p$ and error threshold $d$, we provide a script 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 $r$:

$$\alpha * E + (1 - \alpha) * r, \tag{2}$$

where $\alpha$ 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 Calibration.*** The *calibrate()* method automatically sets the resistance $R_{\text{pot}}$ that best optimizes the tactile sensor's range of voltage output for a particular application. When calibration starts, users engage the tactile sensors in a way that reflects the range of pressures under normal operation. The *calibrate()* method then scans the sensor array using a fixed resistance for a specified duration and keeps track of a percentage 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 $V_{\text{min}}$ and determines the value of $R_{\text{pot}}$ that will make the output of the opamp in the adaptive module $V_{\text{out}}$ equal to 0 volts according to equation 3, obtained from analysis of the readout circuit topology in Fig. 4.

$$V_{\text{out}} = V_{\text{ref}} - \frac{R_{\text{pot}}}{3125} \times (V_{\text{ref}} - V_{\text{min}}). \tag{3}$$

This effectively maps the minimum pressure applied to the maximum sensor output ($V_{\text{ref}}$ Volts) and the maximum pressure applied to the minimum sensor readout (0 Volts), ensuring that the full range of the analog-to-digital converter is used and pressure resolution for the user's application is maximized.

## 4.5 Python Library

Once the tactile sensing devices are flashed with base firmware, the WiReSens Toolkit Python library allows users to configure these devices by passing custom JSON parameters directly to the device. This library also manages multi-sender data reception and serialization to provide wireless data logging and interactive real-time visualization.

***Data Logging.*** The *record()* method saves the data received from each configured sensor to its own HDF5 file, a format which was chosen for its fast read/write speeds and memory efficiency when storing large amounts of data [36]. Pressure data is timestamped and saved to the file after each frame is received. We provide a utility function for reading saved files back into Python for further data processing and exploration.

***Visualization.*** The *visualize()* method initiates a real-time interactive visualization of sensor data from multiple tactile devices via a locally hosted web application, displaying 2D arrays based on each device's electrode configuration. Users can interactively select, move, and delete sensing pixels through drag-and-drop, with a "Select Nodes" button for

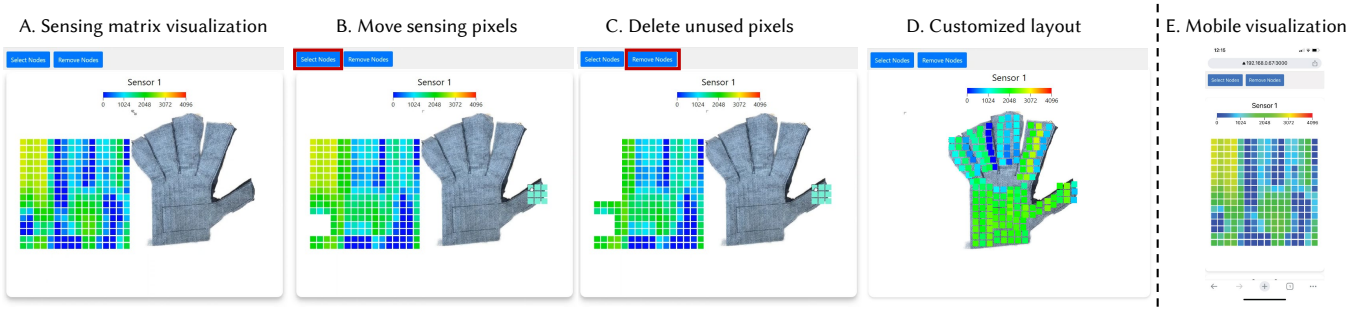


**Figure 5: An overview of the functionality provided by WiReSens Toolkit's interactive visualization, including (A) user-defined background images (B) drag and drop for moving sensing pixels (C) deletion of unused or uninformative pixels. This enables a customized layout for improved data sense-making (D). We also enable visualization on a mobile device (E).**

group selection and a "Remove Nodes" button for uninformative pixels, such as non-responsive regions in specific configurations (e.g., above the thumb in a tactile glove). The visualization includes a color bar indicating pressure levels via ADC voltage and is accessible on mobile devices connected to the same Wi-Fi network. Additionally, the *replay()* method allows users to view HDF5 recordings with controls for start and end timestamps, as well as playback speed.

## 5 Toolkit Evaluation

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

### 5.1 Wireless Communication

To assess the performance and 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 *record()* method. Each protocol and sender combination was tested for three minutes, repeated three times. Fig. 6 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 2099 bytes.

Throughput generally decreases, and packet loss increases as the number of senders grows for each protocol, due to the finite bandwidth available. However, we observed exceptions for Wi-Fi when increasing senders from 2 to 3 and 4 to 5, which we attribute to standard network speed variations.

Wi-Fi achieved the highest throughput and lowest packet loss, with over 60 fps and 0% packet loss in the best case (N=1), and over 30 fps with 0% loss in the worst case (N=4). Wi-Fi's superior performance is attributed to its use of the 5 GHz band, offering higher bandwidth and faster data rates

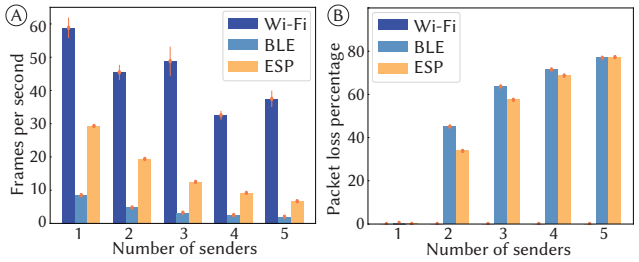


**Figure 6: Average throughput (A) and average percentage of packets lost (B) per sender during multi-sender wireless communication using Wi-Fi, BLE, and ESP-NOW.**

compared to the 2.4 GHz band used by ESP-NOW [9] and the 40 channels for BLE operation [4].

The narrower bandwidths of BLE and ESP-NOW cause more packet collisions during high-throughput multi-sender operations, contributing to increased packet loss as senders increase. An informal experiment demonstrated that WiReSens Toolkit can manage the trade-off between throughput and packet loss by altering transmission frequency. For N=3 senders, we observed a packet loss percentage under 1% with an added delay of 157 ms (dropping the throughput to 0.75 frames per second per sender) using BLE and an added delay of 10ms (dropping the throughput to 11.76 frames per second per sender) using ESP-NOW. This makes BLE or ESP-NOW viable alternatives over Wi-Fi for users prioritizing lower power consumption or extended range [8], without compromising data integrity.

### 5.2 Sensor Adaptivity

Here we characterize the impact of the adaptive module in our readout circuit and assess the effectiveness of our calibration algorithm. Both experiments used a mechanical

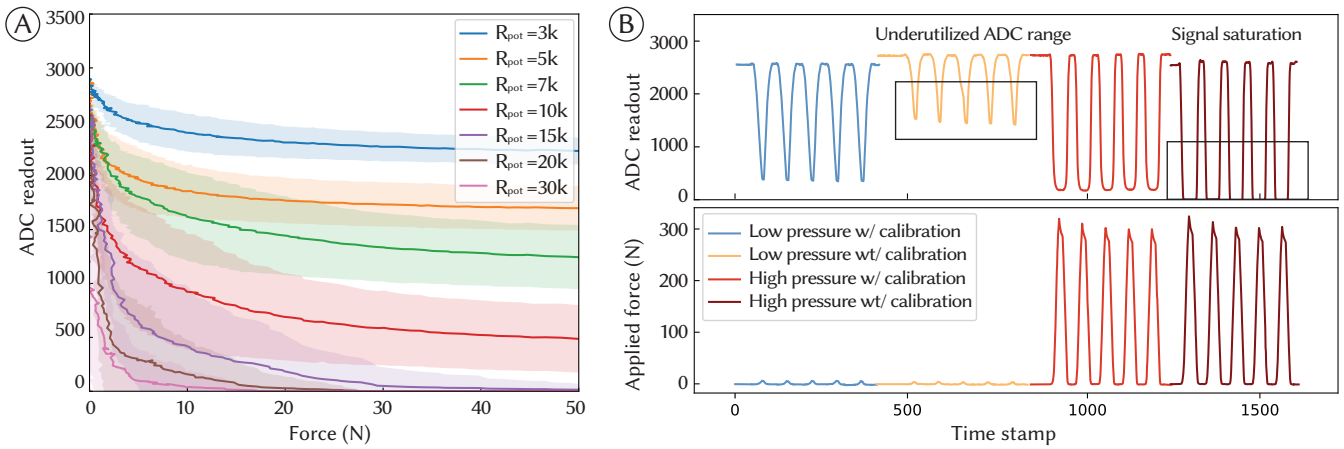


**Figure 7: Characterization of digital potentiometer and calibration. (A) Average and standard deviation of ADC readout in the region of applied force (N) for different digital potentiometer resistance values $R_{pot}$, in Ω. (B) Average ADC readout in the region of applied force before and after calibration during low and high-pressure application cycles. Low-pressure calibration maximizes pressure resolution (blue and yellow curves) and High-pressure calibration avoids saturation (red and brown curves).**

tester (Shimadzu AGX-V2) to apply an adjustable normal force to a 32x32 resistive sensor.

***Digital Potentiometer Adjustment.*** We varied force on a 4x4 node region in the sensor's center while adjusting the digital potentiometer's resistance, $R_{pot}$. We perform the pressure sweep using values for $R_{pot}$ between 3 kΩ and 30 kΩ for 5 cycles each. Fig. 7A shows the average and standard deviation of the ADC readout within the actuated region as a function of applied force for each tested value of $R_{pot}$. Due to the op-amp's inverting topology, the ADC readout is inversely correlated with force, with two approximately linear regions. Notably, the magnitude of the slope of the first linear region increases with $R_{pot}$, which due to the discrete nature of the ADC readout, enhances pressure resolution. However, higher $R_{pot}$ values also lead to output saturation at 0 for greater forces, underscoring the need for an adaptive gain calibration algorithm to manage this trade-off.

***Auto-Calibration Performance.*** To assess the performance of our auto-calibration algorithm, we applied low and high-pressure sequences to the same 4x4 node region during a calibration period of 5 minutes and recorded the calibrated resistance values: $R_{pot}$ = 14062.5Ω for the low pressure and $R_{pot}$ = 8984.37Ω for the high pressure. We then repeated each pressure sequence twice: once with $R_{pot}$ = 14062.5Ω and once with $R_{pot}$ = 8984.37Ω. Fig. 7B 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 300 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

$$I_A * t_A + I_B * (1 - t_A), \tag{4}$$

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 during idle reading, 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 and recording the average current draw. Fig. 8A shows the simulated and observed trends, with results indicating that sending 1% of packets could extend the device 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).

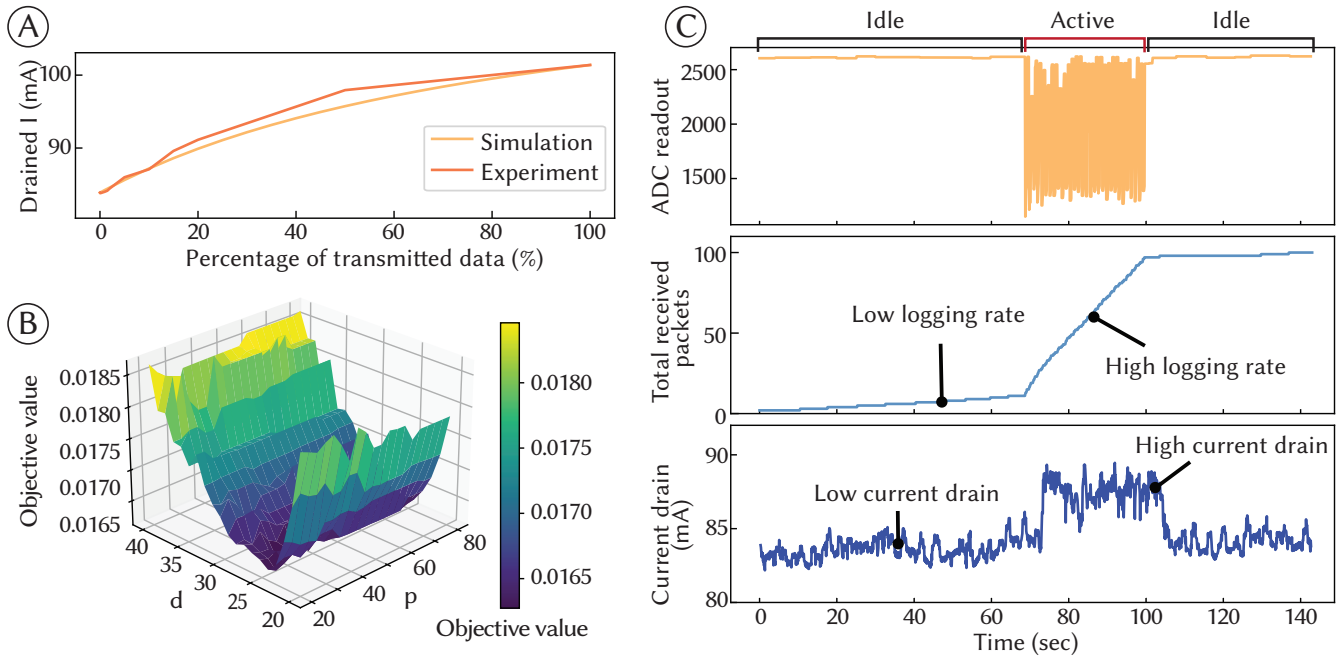


**Figure 8: Characterization of intermittent sending performance. (A) Simulated and observed average current draw (mA) from a tactile sensing device as a function of the percentage of data transmitted during BLE operations. (B) Visualization of objective function used to select intermittent sending parameters. (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.**

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 WiReSens Toolkit, we optimize the transmission parameters $p$ and $d$ ($p = 29$ and $d = 26$) and use them to program the device for the live pressure test. The results (Fig. 8B), 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.

## 6 Example Applications

In this section, we demonstrate the utility of WiReSens Toolkit by using it to prototype a range of resistive tactile sensing devices. These include musical gloves, a gait monitoring shoe sole, an interactive pillow, and a home IoT welcome mat. The tactile sensors were constructed with copper thread and Velostat, using widely accessible fabrication techniques: digital embroidery [1] for the gloves, shoe sole, and pillow, and direct taping [33] for the welcome mat. Each device was programmed in under 10 minutes (including configuring, flashing base firmware, and running methods as indicated in Fig. 3).

***Musical Gloves.*** Soft tactile sensors have been widely explored in the development of digital instruments that offer new means of musical expression [24, 41]. We used WiReSens Toolkit 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. 9 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. 9B.

***Gait Monitoring Shoe Soles.*** We spend countless hours each day on our feet, making gait monitoring a valuable tool for understanding overall health [18, 23]. To demonstrate how WiReSens Toolkit enables extended human-environment activity monitoring, we use it to program a BLE-enabled pressure monitoring shoe sole. We recorded pressure data during 5 minutes of walking and standing still, and then use the

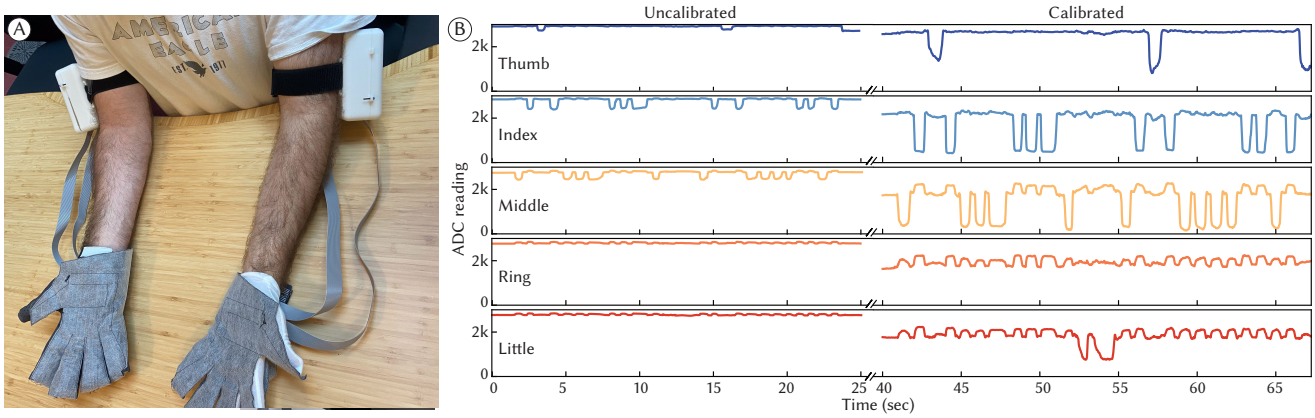


**Figure 9: Wireless Musical Gloves: (A) Depiction of tactile sensing array, with readout circuit affixed to the arm via velcro straps (B) Average ADC readout in the Thumb, Index, Middle, Ring, and Little finger regions of one glove during "Mary Had a Little Lamb", before and after calibration, showing higher sensitivity after calibration.**

toolkit's interactive visualization to replay the wirelessly recorded gait. We observe the current draw from the device's battery during periods of walking and standing still using the Nordic Semiconductor Power Profiler Kit. As shown in Fig. 10, 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 for posture monitoring and as an active input interface for remote control of media playback (Fig. 11)

The final application we developed is a smart home IoT welcome mat (Fig. 12). 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 discuss the primary limitations of WiReSens Toolkit and provide insight into how they may be addressed. We further outline several avenues for future works that WiReSens Toolkit's contributions enable.

### 7.1 Limitations

***Multi-Sender Communication.*** As shown in Figure 6, multi-sender communication using BLE and ESP-NOW protocols demonstrates noticeably slower performance when compared with Wi-Fi. While such differences are caused by fundamental differences in protocol operation, adding support for multiple receivers can help balance the data load more effectively, maintaining consistent throughput as the number of sensors scales. Additionally, we observed some variation in throughput and packet loss among individual senders during our multi-sender experiments, particularly for the BLE protocol. We believe this discrepancy arises from the laptop's automatic negotiation of the connection interval for each tactile sensing device, where the most recently connected device is allocated a shorter connection interval, allowing it to transmit more frequently than the others. To address this, we could modify our Arduino library to enable peripherals to request specific connection intervals, ideally coordinating these intervals across sending devices to ensure more balanced communication.

***Intermittent Sending.*** We've identified limitations in the accuracy and power efficiency of devices programmed with the WiReSens Toolkit during intermittent operation. The receiver's reconstruction algorithm uses packet IDs to detect missing packets and predict them in real-time, working best with ordered protocols like Wi-Fi (TCP) and Bluetooth. For

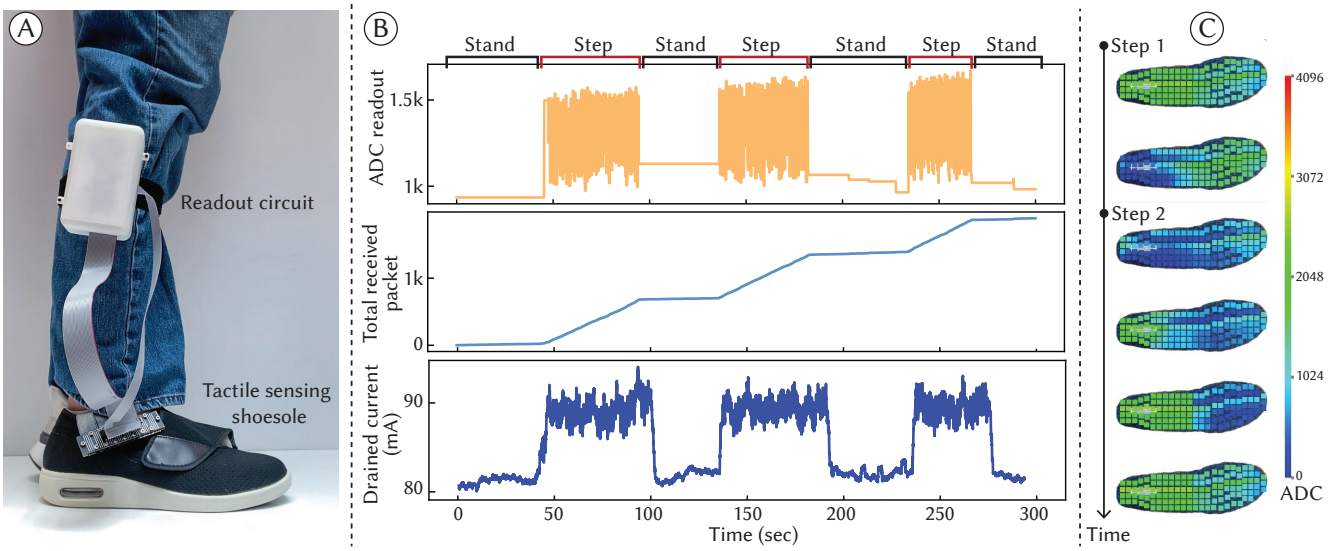


**Figure 10: Gait Monitoring Shoe Sole: (A) Tactile sensing shoe sole form factor and readout circuit. (B) Power efficient operation using intermittent sending, with reduced current drain during periods of standing still and (C) Customized visualization for gait monitoring and step tracking.**


**Figure 11: 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.**

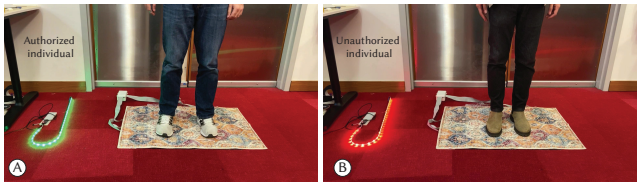


**Figure 12: Smart Home Welcome Mat: IoT lamp processes tactile sensing data from a welcome mat to turn green when (A) an authorized individual enters the home and red when (B) an unauthorized individual enters the home.**

unordered protocols, offline methods could reorder packets before applying the reconstruction algorithm, though this isn't yet implemented. During battery-powered operation, sensor readout quality declines as battery voltage drops, typically after six hours with Wi-Fi, leading to increased packet transmission and quicker battery depletion. To address this, our Arduino library may be extended with methods to monitor error thresholds, skip noise-heavy regions, reduce scan frequency, or enter sleep modes, conserving battery life.

## 7.2 Future Work

***Smaller readout circuits.*** Although improvements have been made in the fabrication of resistive tactile sensing form factors [1, 21] there is a broad need for less bulky open-source resistive matrix-based readout circuitry. While some recent works provide their circuits as open-source [7, 14], most are either relatively large or only support sensors with a smaller resolution (<16x16). Flexible PCBs present a promising medium for miniaturizing the readout circuitry and adaptive module used in this work into flexible form factors for more diverse applications.

***On-device Firmware Optimization.*** In future iterations, WiReSens Toolkit could leverage on-device learning to dynamically adjust its firmware parameters based on real-time tactile sensing data. For instance, in a wearable health monitoring device, the toolkit could automatically fine-tune sensor calibration based on changing body conditions, such as swelling or muscle tension, to maintain accurate pressure readings over time. In a smart appliance application, such

as a touch-sensitive kitchen counter, the device could learn times of the day when interaction will be most frequent, allowing it to go into sleep modes to further reduce power consumption.

***Privacy-Preserving Edge AI.*** WiReSens Toolkit could be used to develop privacy-preserving edge AI applications by enabling tactile sensing alternative to vision-based systems, which often capture identifiable information. Unlike cameras, which process visual data that may inadvertently reveal personal or environmental details, the nature of tactile data is only pressure changes and physical interactions without visual context. Combined with on-device deep learning techniques, this could allow sensitive information to remain on the device while external servers handle less identity-revealing data, enhancing privacy.

## 8 Conclusion

We introduced WiReSens Toolkit, composed of open-source hardware, an Arduino library, and a Python library designed to facilitate the creation of portable, adaptive, and long-lasting resistive-matrix-based tactile sensing devices. Our Arduino library supports variably-sized resistive sensor array readout, wireless communication via Wi-Fi, BLE, or ESP-NOW protocols, adaptive form-factor sensitivity through calibration, and power efficiency through intermittent computing techniques. Additionally, our Python receiver library streamlines the reception, 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. To illustrate WiReSens Toolkit's versatility, we use it to prototype a number of ambient and interactive sensing applications. We believe WiReSens Toolkit will seamlessly integrate with ongoing advancements in digital manufacturing, wireless sensor networks, and embedded AI, 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. https://doi.org/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. https://doi.org/10.3390/s22124460

[3] 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. https://doi.org/10.1016/j.sna.2022.113770

[4] Bluetooth Special Interest Group (SIG). 2014. *Bluetooth Core Specification Version 4.2.* https://www.bluetooth.com/specifications/bluetooth-core-specification/ Accessed: 2024-08-16.

[5] 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. https://doi.org/10.1016/j.jmst.2019.11.010

[6] Tommaso D'Alessio. 1999. Measurement errors in the scanning of piezoresistive sensors arrays. *Sensors and Actuators A: Physical* 72, 1 (1999), 71–76. https://doi.org/10.1016/S0924-4247(98)00204-0

[7] 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. https://doi.org/10.5281/zenodo.1176151

[8] 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. https://doi.org/10.1109/iSemantic52711.2021.9573246

[9] 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.

[10] Niklas Fiedler et al. 2022. Low-cost fabrication of flexible tactile sensor arrays. *HardwareX* 12 (2022), e00372. https://doi.org/10.1016/j.ohx.2022.e00372

[11] Tobias Grosse-Puppendahl, Yannick Berghoefer, Andreas Braun, Raphael Wimmer, and Arjan Kuijper. 2013. OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 152–159. https://doi.org/10.1109/PerCom.2013.6526726

[12] 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)*.

[13] 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). https://doi.org/10.3390/s16020155

[14] Shanthala Lakshminarayana, Younghun Park, Hyusim Park, and Sungyong Jung. 2022. High Density Resistive Array Readout System for Wearable Electronics. *Sensors* 22, 5 (2022). https://doi.org/10.3390/s22051878

[15] 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. https://doi.org/10.1145/3173574.3173610

[16] 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. https://doi.org/10.1016/j.sna.2020.112425

[17] Yichen Li, Yilun Du, Chao Liu, Chao Liu, Francis Williams, Michael Foshey, Benjamin Eckart, Jan Kautz, Joshua B. Tenenbaum, Antonio

Torralba, and Wojciech Matusik. 2024. Learning to Jointly Understand Visual and Tactile Signals. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=NtQqIcSbqv

[18] 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). https://doi.org/10.3389/fcomp.2021.661676

[19] 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.

[20] 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.

[21] Yiyue Luo, Kui Wu, Tomás Palacios, and Wojciech Matusik. 2021. KnitUI: Fabricating Interactive and Sensing Textiles with Machine Knitting. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 668, 12 pages. https://doi.org/10.1145/3411764.3445780

[22] 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 (ICCSP)*. 0635–0639. https://doi.org/10.1109/ICCSP.2017.8286435

[23] 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. https://doi.org/10.1016/S1474-4422(19)30044-4

[24] 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.

[25] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. 2020. OmniTact: A Multi-Directional High-Resolution Touch Sensor. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 618–624. https://doi.org/10.1109/ICRA40945.2020.9196712

[26] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoediauer, 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. https://doi.org/10.1145/3242587.3242664

[27] Paul Badger. 2016. CapacitiveSensor Library. https://www.arduino.cc/reference/en/libraries/capacitivesensor/ Accessed: 2024-08-25.

[28] 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.

[29] 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.

[30] Gurashish Singh, Alexander Nelson, Ryan Robucci, Chintan Patel, and Nilanjan Banerjee. 2015. Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 198–206. https://doi.org/10.1109/PERCOM.2015.7146529

[31] 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. https://doi.org/10.1109/WMNC.2011.6097244

[32] 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. https://doi.org/10.1109/FLEPS61194.2024.10604006

[33] 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.

[34] 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.

[35] 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. https://doi.org/10.1038/s41598-021-01431-y

[36] The HDF Group. n.d.. HDF5: Hierarchical Data Format. https://www.hdfgroup.org/ Accessed: 2024-09-03.

[37] 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. https://doi.org/10.3390/s110303249

[38] Chi Cong Vu, Seung Ju Kim, and Jongwon Kim. 2021. Flexible wearable sensors - an update in view of touch-sensing. *Science and Technology of Advanced Materials* 22, 1 (March 2021), 26–36. https://doi.org/10.1080/14686996.2020.1862629

[39] 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. https://doi.org/10.1145/3527927.3531451

[40] 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.

[41] 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. https://doi.org/10.5281/zenodo.4813391

[42] 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.

[43] 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. https://doi.org/10.1126/sciadv.adp6094 arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.adp6094

[44] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. 2017. GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force. *Sensors* 17, 12 (2017). https://doi.org/10.3390/s17122762

[45] 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. https://doi.org/10.1109/ICDCS.2019.00079