



NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS- 2026-43

September, 2026

**Bridging Prototype to Field: Patterns for Low-Power Wireless Sensing
Across Mobile Health and Environmental Conservation
Blaine Rothrock**

Consumer electronic devices embed a wide range of sensors, enabling them to process and transmit meaningful metrics such as activity, air quality, sleep score, and heart rate. These sensors, and the embedded systems that control them, are ubiquitous, low-power, and inexpensive. However, for a researcher exploring novel applications, consumer devices are largely useless due to limited customization and proprietary data pipelines. This means new applied projects require a significant amount of bespoke technological infrastructure, essentially starting from scratch to fill new needs. These underpinning technology subsystems, which I call "technical scaffolding," receive little academic credit and are reported inconsistently in the literature. This practice, however, constrains systems' ability to bridge the gap between prototypes and real-world deployments, thereby capping their potential impact. This dissertation provides a retrospective analysis of first-hand experience in development and deployment. I present and review the development of FaceBit, a smart face mask for respiratory sensing, and of FlexiBLE, a toolkit for rapid prototyping of embedded sensing devices, and their use in implantable bioelectronics and wearable studies. I then shift to environmental science, with a qualitative study on the use of field sensors, and finally to the development and field deployments of Makak, a system co-designed with Ojibwe Tribal Nations and organizations to conserve Manoomin (wild rice). Additionally, I provide a novel scoping review of ACM literature in embedded sensing systems to categorize how works are grounded in their applied context, what engineering details are included and made available, and how systems are evaluated. Using these sources as evidence, I present eleven development guidelines across three pillars: architectural patterns, an interoperability mechanism, and co-development practices. The

goal is to promote shared tooling that enables generalizable, field-ready research apparatus across mobile health, environmental conservation, and beyond, without having to rebuild scaffolding for each siloed development cycle.

Keywords

Embedded Systems, Wireless Healthcare, Wearables, Environmental Sensing

NORTHWESTERN UNIVERSITY

Bridging Prototype to Field: Patterns for Low-Power Wireless Sensing
Across Mobile Health and Environmental Conservation

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Computer Science

By

Blaine Rothrock

EVANSTON, ILLINOIS

September 2026

© Copyright by Blaine Rothrock, 2026

All Rights Reserved

Abstract

Consumer electronic devices embed a wide range of sensors, enabling them to process and transmit meaningful metrics such as activity, air quality, sleep score, and heart rate. These sensors, and the embedded systems that control them, are ubiquitous, low-power, and inexpensive. However, for a researcher exploring novel applications, consumer devices are largely useless due to limited customization and proprietary data pipelines. This means new applied projects require a significant amount of bespoke technological infrastructure, essentially starting from scratch to fill new needs. These underpinning technology subsystems, which I call "technical scaffolding," receive little academic credit and are reported inconsistently in the literature. This practice, however, constrains systems' ability to bridge the gap between prototypes and real-world deployments, thereby capping their potential impact. This dissertation provides a retrospective analysis of first-hand experience in development and deployment. I present and review the development of FaceBit, a smart face mask for respiratory sensing, and of FlexiBLE, a toolkit for rapid prototyping of embedded sensing devices, and their use in implantable bioelectronics and wearable studies. I then shift to environmental science, with a qualitative study on the use of field sensors, and finally to the development and field deployments of Makak, a system co-designed with Ojibwe Tribal Nations and organizations to conserve Manoomin (wild rice). Additionally, I provide a novel scoping review of ACM literature in embedded sensing systems to categorize how works are grounded in their applied context, what engineering details are included and made available, and how systems are evaluated. Using these sources as evidence, I present eleven development guidelines across three pillars: architectural patterns, an interoperability mechanism, and co-development practices. The goal is to promote shared tooling that enables generalizable, field-ready research apparatus across mobile health, environmental conservation, and beyond, without having to rebuild scaffolding for each siloed development cycle.

Acknowledgments

This dissertation is a huge collaborative effort and is not possible without the significant help and support of others. While my stubborn pursuit of better research tooling and generalization of system infrastructure is my own crusade and hill to die on, the individual project contributions and funding are part of much wider efforts than my own. Additionally, the support I have received outside of academia is equally, if not more, important. It is rare that someone gets the opportunity to pursue passions that amount to doctoral accreditation, but I have been incredibly lucky to do so, and not without great support from family, friends, and mentors. The following are specific acknowledgments of support I received:

Professor Josiah Hester has been the principal investigator of nearly all of my work and has been a great collaborator and facilitator of my efforts. He has taught me much about academic success and the importance and challenges of sustainable computing research, and the fundamentals of embedded system development. Additionally, Dr. Hester has introduced me to and welcomed my collaboration on indigenous research. The single most rewarding component of my graduate studies is my work with Ojibwe Tribal Nations and organizations on Makak, which Josiah facilitated.

Professor Nabil Alshurafa has been an outstanding and central mentor throughout my graduate career, teaching me about behavioral healthcare studies, which encompass sociotechnical study design, laboratory management, entrepreneurship, and leadership. I am extremely grateful to have had both Dr. Hester and Dr. Alshurafa as advisors of my research.

Professor Maia Jacobs was a great influence early in my PhD journey, inviting me to her lab meetings and sponsoring collaborative research involving qualitative analysis. I owe my outlook on the equal importance of qualitative assessment and technical contributions to her guidance; without it, I would have had a much more technical and siloed contribution in my dissertation.

Professor Bill Miller is a collaborator in my work with Ojibwe Tribal Nations and Organizations and brief work with The Nature Conservancy. Bill has provided mentorship in environmental conservation and my career in general.

Professor Alex Cabral has been a great component in my success as a late graduate student. Dr. Cabral led the subgroup within Dr. Hester's lab and was a great mentor and peer collaborator. Most importantly, she greatly assisted me during my final year post-prospectus in finishing this dissertation, providing weekly meetings, feedback, editing, and assistance with my job search. The outcome, this dissertation, would be greatly delayed and less polished without her help.

I give special thanks to the welcoming members of the Ojibwe communities with whom I have collaborated on the conservation of Manoomin. As a person of settler descent, my experience thus far in working alongside indigenous research has been humbling, incredibly rewarding, and a key component in shaping my outlook on computing research. Specifically, I thank members of the Great Lakes Indian Fish and Wildlife Commission, the Lac Courte Oreilles Band, the Lac du Flambeau Band of Lake Superior Chippewa, and the 1854 Treaty Authority for sharing knowledge with me and making me feel welcome as a peer researcher. I also acknowledge the Manoomin Collective, the broader group supporting the conservation of Manoomin.

I want to thank all of my peer collaborators in Ka Moamoa Lab and HabitsLab. I have learned much from working in groups on various projects. I specifically want to acknowledge my two most prominent collaborators, Alexander Curtiss and Eric Greenlee. My work would not be in its current state without their distinct help and diverse expertise.

Many family and friends have been crucial to my success in academia, and I cannot name them all here. It is a sacrifice to spend so much time and effort on a doctoral degree, and it has not come without burdens on those who support me. I specifically thank my mother,

Leigh Henderson, and my granddad, Dr. Thomas Henderson; they have both helped me emotionally and financially in various ways throughout my graduate studies. Finally, I thank my wife Julie; without her, I certainly would not have started or completed this work. She has taught me more than I could ever get out of a degree, and is, along with our son Quinn, the most important motivator in my life. She has my undying gratitude for supporting me in this endeavor.

This research is partially supported by the National Science Foundation under award numbers CNS-2145584, CNS-1850496, CNS-2032408, CNS-2038853, CNS-2030251, CNS-2107400, and IIS-2228983. I would also like to acknowledge support from the Alfred P. Sloan Foundation, VMware, Google, and Catherine M. and James E. Allchin. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other supporters.

This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-21-2-7119. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

This work was supported by ARPA-H under agreement number 140D042490003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Advanced Research Projects Agency Health or the U.S. Government.

List of Abbreviations

ACM	Association for Computing Machinery
AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
ARPA-H	Advanced Research Projects Agency for Health
BCG	Ballistocardiography
BLE	Bluetooth Low Energy
BrPM	Breaths Per Minute
CDC	Centers for Disease Control and Prevention
CoAP	Constrained Application Protocol
COTS	Commercial Off-the-Shelf
COVID-19	Coronavirus Disease 2019
CSCW	Computer-Supported Cooperative Work
DARPA	Defense Advanced Research Projects Agency
DOF	Degrees of Freedom
DOI	Digital Object Identifier
ECG/EKG	Electrocardiography/Electrocardiogram
FDA	Food and Drug Administration
FFT	Fast Fourier Transform
GATT	Generic Attribute Profile
HAL	Hardware Abstraction Layer
HATEOAS	Hypermedia as the Engine of Application State
HCI	Human–Computer Interaction
HR	Heart Rate
HTML	HyperText Markup Language
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IIR	Infinite Impulse Response
IMU	Inertial Measurement Unit
IMWUT	Interactive, Mobile, Wearable and Ubiquitous Technologies
IoT	Internet of Things
IRB	Institutional Review Board
ISWC	International Symposium on Wearable Computers
IVC	Inspiratory Vital Capacity
JITAI	Just-in-Time Adaptive Intervention
JSON	JavaScript Object Notation
LED	Light-Emitting Diode
LF-PPG	Low-Frequency Photoplethysmography
LLM	Large Language Model
LoRa	Long Range (radio)
LoRaWAN	Long Range Wide Area Network

LTE-M	Long-Term Evolution for Machines
LwM2M	Lightweight Machine-to-Machine
MCU	Microcontroller Unit
mHealth	mobile Health
ML	Machine Learning
MoU	Memorandum of Understanding
MTU	Maximum Transmission Unit
NHP	Non-Human Primate
NOAA	National Oceanic and Atmospheric Administration
NREL	National Renewable Energy Laboratory
NUS	Nordic UART Service
OS	Operating System
OSHA	Occupational Safety and Health Administration
OTA	Over-the-Air
PAR	Photosynthetically Active Radiation
PAT	Pulse Arrival Time
PCB	Printed Circuit Board
PD	Participatory Design
PDMS	Polydimethylsiloxane
PEP	Pre-Ejection Period
PPE	Personal Protective Equipment
PPG	Photoplethysmography
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PTT	Pulse Transit Time
QA/QC	Quality Assurance/Quality Control
RAM	Random-Access Memory
RE	Resilience Engineering
REST	Representational State Transfer
RMSE	Root-Mean-Square Error
RR	Respiration Rate
RSSI	Received Signal Strength Indicator
RTC	Real-Time Clock
RTOS	Real-Time Operating System
SCG	Seismocardiography
SD	Secure Digital
SDG	Sustainable Development Goal
SDK	Software Development Kit
SIGCAS	Special Interest Group on Computers and Society
SIGCHI	Special Interest Group on Computer–Human Interaction
SIROF	Sputtered Iridium Oxide Film
SoC	System-on-Chip
SoM	System-on-Module
TCP	Transmission Control Protocol
TIL	Total Inward Leakage
TVWS	Television White Space

UART	Universal Asynchronous Receiver-Transmitter
UbiComp	Ubiquitous Computing
UI	User Interface
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WHO	World Health Organization
Wi-Fi	Wireless Fidelity (IEEE 802.11)
WoT	Web of Things

*Dedicated to my wife, Julie, for her unwavering love
and support, and
to my granddad, Thomas, for teaching me at a young
age to love the act of learning.*

Contents

Abstract	3
Acknowledgments	4
List of Abbreviations	7
List of Tables	18
List of Figures	21
1 Introduction	27
1.1 Thesis & Research Questions	31
1.2 Dissertation Overview	33
1.3 Disclosures	35
2 Literature Review: Development of Embedded Sensing Systems	36
2.1 Methodology	37
2.1.1 Search	37
2.1.2 Screening	39
2.1.3 Coding	40
2.2 A Corpus of Embedded Sensing Systems	44
2.2.1 Application Type	45
2.2.2 Wide Ranging Technology	46

	12
2.3 Framing, Reuse, and Evaluation	52
2.3.1 Framing Systems in Applied Contexts	52
2.3.2 Describing System Implementation & Reuse Expectations	54
2.3.3 System Evaluation: Prototypes to Field	57
2.4 Discussion	60
2.4.1 Field Variability and the use of Artificial Intelligence	60
2.4.2 Publishing Source Materials	61
2.5 Limitations and Future Work	62
3 The Cost of Iteration in Wireless and Mobile Health: An Exploration of FaceBit .	64
3.1 Introduction	65
3.2 Related Work	67
3.3 System Design	67
3.3.1 Design Goals	67
3.3.2 Clinical Needfinding	68
3.4 System Implementation	69
3.4.1 Hardware Platform Design	70
3.4.2 Runtime System	71
3.4.3 Mask Fit Exploration	76
3.4.4 Mobile Companion Application	78
3.5 Evaluation	79
3.5.1 Performance Benchmarking	80
3.6 Retrospective	88
3.6.1 Development Cycle	89
3.6.2 FaceBit’s Evaluation	91

3.7	Towards Better Research Apparatus	92
4	A Declarative and Adaptive Platform for Wireless Personal Health Research . . .	94
4.1	Introduction	94
4.2	Related Work	97
4.2.1	Mobile Health Sensing Platforms	97
4.2.2	New Hardware, New Software Applications	99
4.2.3	Sensor Integration Middleware	100
4.2.4	Positioning FlexiBLE	101
4.3	System Requirements	101
4.3.1	Data Collection	102
4.3.2	Runtime Flexibility	103
4.3.3	Data Interoperability	103
4.4	FlexiBLE System Design & Implementation	104
4.4.1	FlexiBLE Device Specification	104
4.4.2	FlexiBLE Companion Application	106
4.4.3	Hardware and Firmware Development	108
4.5	Application: Physiohub	111
4.6	Application: Cell Oxygenation Delivery Systems	116
4.6.1	Electrocatalytic On-Site Oxygenation Platform	117
4.6.2	Hybrid Oxygenation Bioelectronics for Implanted Therapy	117
4.7	Reflections from Practice	118
4.7.1	Practitioner Context and Development Landscape	119
4.7.2	FlexiBLE's Contribution to Development Workflows	120
4.7.3	Limitations and Remaining Challenges	122
4.8	Moving Beyond FlexiBLE	124

5	Barriers and Methods of Ecological Field Sensing	126
5.1	Introduction	126
5.2	Related Work	127
5.3	Study Methodology	128
5.4	The Field Sensor Lifecycle	131
5.4.1	Data Need Finding	132
5.4.2	Technology Selection	133
5.4.3	Deployment	138
5.4.4	Maintenance	141
5.4.5	Data Analysis	143
5.5	Building Systems from Insights	147
6	Co-Designing Low-Cost Sensors for Manoomin Conservation	148
6.1	Introduction	148
6.2	Related Work	151
6.2.1	Manoomin Ecological Research & Conservation	152
6.2.2	Co-designing with Indigenous Partners	153
6.3	Requirements, Goals, & Co-Design	154
6.3.1	Deployment Goals	156
6.4	System Design	157
6.4.1	Mechanical Design	157
6.4.2	Core Electronics	159
6.4.3	Sensing Modalities	161
6.4.4	Firmware	163
6.4.5	Software	166

6.4.6	Cloud Data Collection	167
6.5	Deployment Methodology	168
6.5.1	Partners and Sites	168
6.5.2	Deployment Protocol	169
6.5.3	Post-Deployment	170
6.6	Findings	171
6.6.1	Pilot Results	171
6.6.2	Scaled Deployment	172
6.6.3	Pilot-to-Scaled Iteration	175
6.7	Retrospective	177
6.7.1	Platform Choice and Development Velocity	177
6.7.2	Cellular Infrastructure and Data Sovereignty	179
6.7.3	Energy Budgets and Field Conditions	181
6.7.4	Balancing Complexity and Deployment Scope	181
6.8	Future Directions	183
7	Learned Guidelines for the Development of Embedded Sensing Systems	184
7.1	Architectural Patterns	186
7.2	A Mechanism for Interoperability	192
7.3	Co-Development Practices	198
7.4	Towards Sustainable Systems for Long-Term Reuse	204
8	Conclusions and Future Work	206
8.1	Summary of Contributions	207
8.2	Future Work	207
8.2.1	A Knowledge Base for System Development	208

8.2.2	Framework Development	208
8.2.3	Toolkit Development	209
	References	210
A	Literature Review	241
A.1	Code Book	241
B	Protocols	247
B.1	FaceBit Clinical Interview Protocol	247
B.1.1	Questionnaire/Interview Procedure	247
B.1.2	Results of Target Metrics Section	247
B.1.3	Mask Wear	248
B.1.4	Expanded Metrics of Interest.	248
B.2	FlexiBLE Interview Protocol	249
B.2.1	Opening Checklist	249
B.2.2	Interview Guide	249
B.2.3	Closing Checklist	252
B.3	FlexiBLE Interview Codebook	252
B.4	Microclimate Sensing Field Study Protocol	256
B.4.1	Start	256
B.4.2	General Framing (required)	257
B.4.3	Field Research Specific (understanding the user flow of conducting field research)	258
B.4.4	Collecting Sensor Data Specifically	258
B.4.5	Understanding the problems in current field research technology . . .	259
B.4.6	Understand user needs during sensor data collection, specifically . . .	259

B.4.7	Understand how they analyze data	259
B.4.8	Understand how the environment could impact the field researcher in conducting their work	260
B.4.9	Understand revisits	260
B.4.10	Closing	260
B.4.11	Follow up	261
C	Additional Analysis	262
C.1	Wireless Protocols	262
C.2	Embedded Time Management Example	265
Vita		268

List of Tables

2.1	ACM Digital Library search strategy. All searches were limited to research articles in the ACM Full-Text Collection (ACM Content: DL) with an online publication date on or after 1 January 2022. <i>Records</i> is the count retrieved before screening; <i>kept</i> is the count included and coded.	39
2.2	Inclusion criteria. A paper is included only if all criteria hold.	40
2.3	Exclusion criteria and application counts across all three screening phases. Each paper may have multiple exclusions.	41
2.4	Deductive codes for <i>Application Type</i> : the target domain of each system’s contribution. Counts reflect the 124 Phase 3 included papers.	46
2.5	Representative sample of the coded corpus across the deductive matrix; one paper per row, cited by number (titles omitted). <i>Reuse</i> = Hardware/Firmware/-Analysis/Software rubric (0–4, table 2.7); PW = previous works, CoTS = off-the-shelf, — = N/A. <i>Framing</i> (table 2.6): Hyp/Lit, Hypot, Adj, SecObs, PrimObs, Co, Co+. <i>Eval</i> : Bench, Ctrl, Demo, Wkshp, TCLtd, TCFull. Multi-value cells show the top 3; the full corpus and all columns are in the data release.	51
2.6	Deductive codes for <i>Application Framing</i> : how each paper grounds the motivation and problem statement for its contribution. Counts reflect the 124 Phase 3 included papers.	53

2.7	Reusability rubric. Each paper is scored independently along four dimensions— H ardware / P latform, F irmware, A nalysis / P rocessing, and S oftware—using the levels below. Scores reflect what can be reused from the paper text <i>or</i> from the public resources it references. Not every paper is scored on every dimension (e.g., software may be <i>N/A</i>), so column totals need not equal 124.	55
2.8	Deductive codes for <i>Evaluation Type</i> . Multiple codes may apply to a single paper (e.g., benchtop validation followed by a limited-scope deployment), so the counts sum to more than the 124 included papers.	59
3.1	Power and energy consumption of the FaceBit application runtime.	80
3.2	Root mean squared error for FaceBit predicted heart rate and respiration rate compared to ground truth for all in-wild scenarios.	87
6.1	Makak sensing modalities across the two deployments. Mount indicates the physical fixture on which each sensor is attached: SS = above-water Stevenson screen, WS = water-surface fixture, A = anchor fixture, MB = integrated to mainboard enclosure.	163
6.2	Makak sensor validation against co-deployed reference instruments.	173
6.3	Major iterations from the pilot (2024) to scaled (2025) Makak deployments. Each row pairs a pilot finding with the scaled response and the observed outcome, including cases where the iteration introduced new failure modes. .	176
C.1	Per-message wire size of the same two payloads under different encodings, including the data-capable IoT application-layer protocols (LwM2M-CBOR, Sparkplug NDATA, Homie). The Health payload is a 444-sample multi-wavelength PPG buffer (four channels of integer ADC counts, anchor timestamp + 2250 μ s period); the Env payload is a single environmental reading with 0 error string(s). $\times$ floor is each encoding’s size relative to the bit-packed range-tight floor (bottom row), the theoretical minimum.	264

- C.2 Compression overlay (raw DEFLATE, zlib level 9). Generic compression recovers much of the text formats' overhead but does not reach a compact typed encoding, and adds overhead on already-small payloads. 265
- C.3 Serial-monitor excerpts from the two sketches (same workload), showing timestamp drift versus none. **Naive:** the fixed-rate column (`fixedrate_ms`) falls progressively behind real acquisition time (Δ_{fixed} grows to -2.3s); within one 20-sample batch (idx 2000–2019) a single `bundle_stamp_ms` is reused for every row; and all values are relative milliseconds from boot, never anchored. **Corrected:** each sample is stamped at acquisition, the host SYNC flips `synced` 0→1 at idx 50, `assigned_us` becomes absolute epoch microseconds, and the error stays 0. 267

List of Figures

1.1	An outline of this dissertation’s progression by chapter. I cover two domains, mobile healthcare and environmental sensing, but outline and reflect on existing work. These first-hand experiences, plus a literature covering current development methods in embedded sensing systems, are synthesized into developmental guidelines.	32
2.1	PRISMA-style screening flow, from the three ACM Digital Library searches through full-text eligibility to the 124 coded papers.	38
2.2	The web-based coding application used in the literature review.	43
2.3	Chart of ACM venues in the corpus with at least 2 representative works. There are 41 venues with a single entry.	45
2.4	Screening detail. (a) Exclusion reasons tagged at each screening pass; and (b) application type of the 124 coded papers.	47
2.5	Reusability by component	57
2.6	Distribution of evaluation methods across the corpus. Each study may have multiple methods. <i>Participatory/Workshop</i> always accompanies another method.	58
3.1	FaceBit is an energy-efficient, research-focused hardware platform that enables face masks to be retrofitted (A) to collect on-device health and safety metrics, such as heart rate, respiratory rate, and wear time (B). Notifications are sent to a phone application upon detection of state changes (C). Easy reattachment upon mask replacement (D) enables lower-burden use in difficult work environments (E).	65

3.2	1-7 Likert Scale of importance for metrics of focus. The majority of the metrics presented were deemed important by healthcare workers (to the right of the 0% mark). The length of each color represents the percentage of people who provided the corresponding response.	69
3.3	The top of the FaceBit board houses the compute, sensing, and most of the energy-harvesting circuitry. The application runs on the BMD-350 Module, which incorporates Nordic's NRF52832 (512k Flash/64k RAM). The bottom of the FaceBit board houses the energy-storage elements and the programming interface. It also includes an accessory port for adding future modules. FaceBit is shown placed in an N95 mask in (c) via a magnet (d).	70
3.4	Example air pressure signal, with mask on/off annotations, collected from FaceBit in an N95 mask in a field test.	72
3.5	Heart rate signal-processing pipeline. 3-axis gyroscope data is filtered to extract the instantaneous heart rate from the time between successive zero crossings of the signal. A standard deviation threshold is used to determine a realistic change in rate from beat to beat. Finally, heart rate is calculated as a 10-second average of valid instantaneous rates.	75
3.6	Metric signal existence plots. (a) ECG ground truth signal collected from a Polar H10 chest strap compared to a Ballistocardiography (BCG) signal extracted from the FaceBit's gyroscope. (b) Respiratory rate signal existence: Raw temperature and pressure signal from the FaceBit sensor board. (c) Mask fit exploration. Cumulative integration of the pressure signal sensed inside an N95 mask while under suction from a 12V vacuum pump. The highlighted minimum values represent a measure of mask resistance to airflow.	77

3.7	FaceBit Companion Application on iOS	(a): The FaceBit Phone Application homepage displaying general details, current wear time, respiration rate, heart rate, and temperature data from the FaceBit Sensor Board.d. Also shown is a wear-time warning indicating a mask change. (b): The mask wear time interface for tracking mask disposal. (c): FaceBit Sensor Board detail screen outlining current connection status	79
3.8	Health metric evaluation results	(a) Respiration rate (186 samples, $n = 8$): Cumulative distribution function over the evaluation compared to ground truth respiration belt. (b) Heart Rate (149 samples, $n = 8$): Cumulative distribution function over the evaluation compared to ground truth ECG chest strap.	81
3.9	Results from initial mask fit exploratory study capture using FaceBit.	(a) Signal from inspiratory vital capacity test recorded on-board FaceBit showing filtered pressure signals for each participant ($n=3$) grouped by fit. (b) Integral values from initial evaluation for participant (3) and each experiment (3) compared to each leak configuration.	84
3.10	Secondary mask fit exploratory study using an external differential pressure sensor. The integral value for five mask configurations was measured during an inspiratory vital capacity test. The percent change relative to the mask-press configuration allows us to compare individual capacities side-by-side. The trend shows that the integral value generally decreases as more leaks are introduced. Physiological differences are noted in the legend.		86

3.11	The free-living scenario "walk to lunch". FaceBit can capture accurate heart rate even in low-motion conditions despite noisy environments and environmental factors like heat and humidity. (middle) shows annotations where heart rate measurements were gathered by FaceBit, plotted over the ground truth heart rate, where each marker is the instantaneous heart rate associated with each beat. (top) Zoomed-in data-sections to highlight error. (bottom) GPS plot of the walking path with indicators of FaceBit heart rate recordings.	87
3.12	FaceBit reported respiration rate plotted over the calculated rolling average of instantaneous respiration rates from the ground truth respiration belt. The data shown is from the "walking to lunch" scenario.	88
3.13	The iterative FaceBit development cycle. The process follows two cycles: (1) development firmware is flashed to the device to collect raw sensor data, which is then analyzed to create onboard metric algorithms. (2) Application firmware is written to evaluate the algorithms along with the device function. These cycles are repeated to make progress, but they also pose limitations for evaluation in real-world environments.	89
4.1	Overview of the FlexiBLE system. Custom wearable devices equipped with sensors (e.g., PPG, accelerometer, ECG) communicate with a mobile phone application via Bluetooth Low Energy (BLE). A JSON specification file declaratively describes the device's data streams and configurable parameters, enabling the application to dynamically generate interfaces without code changes. Sensor data streams flow from the device to the phone, while experimentation parameters can be sent back to the device for runtime reconfiguration. Collected data feeds into experiments for downstream data analysis and algorithm development, which in turn inform the wearable device's firmware algorithms, creating an iterative development loop.	95

4.2	(A) An example FlexiBLE device specification in JSON format. This specification is an abbreviation of an implantable cell oxygenation device used in real studies. (B) A prototype web application for declaratively building device specifications.	106
4.3	Screenshots of the FlexiBLE iOS application. (a) Runtime parameter configuration interface with dynamically generated sliders for sensor settings such as sampling frequency and LED brightness. (b) Real-time data visualization of accelerometer data. (c) Tabular view of raw PPG sensor data with timestamps and metadata. (d) Experiment detail view showing collection metadata, GPS trace, and time markers.	109
4.4	PhysioHub system overview. (a) shows the main and auxiliary PCBs, as well as the custom case design with photoplethysmography (PPG) and electrocardiogram (ECG) configurations. (b) shows the placement on the body for our in-clinic evaluation. (c) A single participant graph of low-frequency photoplethysmography (LF-PPG) showing a clear response during the Valsalva maneuver. (d) A single R-R interval from the above participant, showing all cleaned data collected by the system, including distal and proximal multi-wavelength PPG signals, ECG, and seismocardiography (SCG), with fiducial points for each.	113
4.5	PhysioHub evaluation results: (a) time synchronization accuracy between leader and follower nodes, and (b) physiologically meaningful LF-PPG response captured during in-clinic evaluation.	115
5.1	Key factual information of the interview participants.	129
5.2	The Microclimate Sensor Lifecycle, highlighting the main high-level components of an end-to-end project, including quotes from our interview participants. .	131

6.1	Overview of the Makak system from the pilot deployment (left) and scaled deployment (right),	151
6.2	A brief timeline of co-design engagement and Makak development.	155
6.3	Companion smartphone applications used for on-site validation and configuration during Makak deployments: (a) Pilot Android companion application, showing the per-subsystem GATT view, and (b) Scaled iOS companion application, showing the command terminal used during 2025 deployments.	166
6.4	Makak deployment locations over the 2024 and 2025 seasons. Locations are approximate and shown with corresponding partner organization logos.	169
6.5	Field photographs from the Makak deployments: (a) on-site validation using the companion application (Photograph by David Cournoyer), (b) nesting observed beneath the platform during retrieval, and (c) a Makak co-deployed with a YSI EMM350 Pisces at the Emiquon Preserve near Havana, IL.	170
6.6	Sensor validation against external reference instruments. (a) Hourly mean diurnal cycle of dissolved oxygen saturation at Emiquon, vertical dotted line indicates diurnal maximum. (b) Daily change from start-of-deployment baseline at Rice Lake.	174
7.1	Developmental guidelines to evidence and research-question mapping	185

CHAPTER 1

Introduction

As you read this, there are likely more than a dozen embedded sensors within an arm's reach. Consumer devices contain a wide range of general-purpose sensors, enabling them to process and transmit data into meaningful metrics such as activity, air quality, sleep score, or heart rate. These sensors, and the embedded systems that control them, are ubiquitous, low-power, and inexpensive. However, for a researcher exploring applications beyond the intended functions, these sensors are largely useless. Mass-produced consumer products are designed for a general population, but in practice, they do not perform the same way across scenarios or for every individual. These products rarely expose raw sensor streams, and research-grade devices are expensive and only accessible via proprietary technology stacks.

In field ecology, scientists are forced to limit their research based on the capabilities of current consumer products, as one researcher states, 'proprietary systems define my research questions themselves' [1] (Chapter 5). In wireless healthcare, questioning the accuracy of existing devices requires intensive workarounds; for example, Ajmal et al. were forced to reverse-engineer existing photoplethysmography-based heart rate hardware to explore variation across skin tones and obesity, concluding that proprietary algorithms leave gaps [2]. Questioning and reporting on these gaps is difficult as data and algorithmic access remain hidden behind proprietary hardware and software. This landscape leaves researchers with two options: adapt their research to fit existing technology, or develop their own. The latter, the design of specialized embedded systems, has become a common practice in computing research, in contrast to the limitations of what is available on the market. Yet it typically requires starting from scratch, with little shared infrastructure to build on.

Embedded technology is ubiquitous and accessible; however, the ability to develop new, specialized systems that have a meaningful impact in the field is not. This dissertation studies the development methods and challenges of these systems in the context of mobile health and environmental conservation. I review the current state of development of **low-power wireless embedded sensing systems** in research settings and explore how barriers can be reduced to ultimately narrow the gap between prototype and the field.

This thesis focuses on custom embedded devices targeted for in situ deployments. These devices are commonly found in specialized health wearables and remote environmental monitors. They are internally powered, microcontroller-based, wireless, and capable of sensing their environment in some way. They may also have actuation capabilities, but this is not a requirement. I define low power as generally below the peak consumption of 500 milliwatts and normal operation under 100 milliwatts. Microprocessors with virtual memory mapping and general-purpose operating systems (e.g., Raspberry Pi and other single-board computers) are excluded. The devices perform specialized, finite functions with minimal user interaction. Examples are provided throughout this text, and a more in-depth description of the inclusion and exclusion criteria is provided in the literature review (Chapter 2).

The development of these devices requires an interdisciplinary set of engineering skills, spanning electrical, mechanical, and computational engineering. Unfortunately, little is provided in the form of best practices, methodologies, or tooling. This lack of tooling is a well-known issue in research software, which has been shown to raise concerns about reproducibility [3]. Without generalizable tooling, what Neville calls *computing apparatus* in reflecting on hypothetical software equivalents to Michael Faraday’s custom laboratory equipment [4], incremental contributions to the field require substantial effort to reconstruct supporting infrastructure. I call this supporting infrastructure **technical scaffolding**, referring to any required engineering work that does not contribute to the system’s novelty. This can include, for example, sensor drivers, data serialization, interoperability, mechanical

designs, schematics, database schemas, and other components of an end-to-end embedded system. This scaffolding is especially important when building devices for field evaluation under real-world constraints, and projects often *reinvent the wheel* with each new system.

My thesis focuses on computation perspectives: the software architectures, data management, and development practices by which embedded systems are built, deployed, and reused. However, I draw broadly from technical concepts from electrical, mechanical, and computational engineering. Mechanical and electrical engineering concerns are first-class constraints that I repeatedly confront throughout this work, but generalizing them is beyond its scope and remains an area for exploration in neighboring disciplines. I exemplify this perspective in mobile health device development in Chapters 3 and 4.

The tension between novelty and the effort required for application is not unique to this field. This gap is documented across disciplines as the *Valley of Death* [5] in biomedical research, the *research-to-practice gap* [6] in education, and the *know-do gap* or *knowledge translation* [7] in policy and public health. Stokes frames the productive middle ground as *Pasteur's Quadrant* "use-inspired basic research," that is neither purely fundamental nor purely applied [8]. The systems this work discusses sit squarely in this space; systems built to enable basic research in other fields.

Within computing, the same difficulty exists in HCI [9, 10], software engineering [11, 12], and ubiquitous computing (UbiComp) [13, 14]. Tolmie calls this mundane, often invisible, work of getting systems to run in the field *digital plumbing* and argues for tooling to support it [13]. My notion of *technical scaffolding* encompasses these components; however, it is expanded to include tooling that assists in research and development, which is *removed* in full-scale production, hence *scaffolding*. I refer to both *plumbing* and *scaffolding* throughout this work as subcategories of *system infrastructure*, which is strictly separate from individual contributions. Abowd, reflecting on twenty-five years of UbiComp, goes further, urging the field to broaden its reach across computer science [14]. The aim is to recognize and motivate

contributions that reduce the need for scaffolding through tooling and shared computing infrastructure [12, 15].

I propose that to better bridge the gap between device prototypes and real-world field deployment and achieve higher impact, we reduce the technical burden. In practice, little work produces reusable artifacts for derivative work to build from. Related, reproducibility is a well-documented problem across computer science [3, 16]. Much technical scaffolding is poorly documented or missing entirely, leading projects to duplicate efforts and cut corners in implementation to reach evaluation. Specifically, my literature review (Chapter 2) and environmental science interviews (Chapter 5) indicate that interoperability is a major, rarely reported issue and a source of friction in large-scale field deployments. *Interoperability*, in this context, refers to the different levels at which devices exchange, make available, and interpret data into meaningful information. I argue that best practices for interoperability, the basic components by which devices exchange data with hosts, networks, and downstream workflows, are critical to the reusability of embedded sensing devices.

Engineering skills are not the only component to success in applied devices. A tractable gap in the engineering landscape does not guarantee its usefulness in an applied setting [17]. Anyone can propose a technical solution, but making it practical is rare, as computer science methodologies alone cannot bridge this gap. It requires social engagement and practice of co-design [17–19]. Such methods have been shown to reduce this gap broadly, and in in-situ sensing applications [1, 20, 21], yet their use is rare and inconsistent [22–24]. In practice, integration efforts often impose hidden labor burdens on practitioners. Hidden or *invisible labor* is a term referring to undervalued labor in the workforce [25], which has been adopted in Computer Supported Cooperative Work (CSCW) and Human Computer Interaction (HCI) research on the use and adoption of technology [26, 27]. The alternative is to address device design through participatory principles, in which the technology’s beneficiaries are partners and equal contributors [28]. These types of engagements are explored as a necessary methodology

for impactful deployments. This work finds that community engagement is required for systems to enable real-world impact; careful consideration is needed to align community goals and practices and to respect the additional labor that technology integration can entail. Chapter 6 explores co-design with political and sovereignty dimensions in partnership with Ojibwe Tribal Nations and organizations.

This urgency is increasingly recognized within the computing research community itself. Communities such as the ACM Special Interest Group on Computers and Society (SIGCAS) and Computer-Human Interaction (SIGCHI)¹ have aligned around the 2030 United Nations Sustainable Development Goals (SDGs), which identify specific data gaps that must be closed to meet the targets [29, 30]. Closing those gaps depends on readily deployable systems for collecting data in the field. These are precisely the systems whose development this thesis seeks to make more tractable.

The applied computing community wants and needs better tooling to ease this barrier to entry and provide standardized methods that generalize across wireless embedded sensing system development [31, 32]. However, academic incentives do not align with artifact reuse, making such efforts unattractive [33–35]. I approach this both bottom-up, by reviewing post-COVID-19 ACM literature, and top-down, through retrospectives on first-hand experience in device and tool development. I do not, however, attempt to address the misalignment of incentives, but rather highlight the existence and provide guidelines for producing systems that focus on reuse and impact in applied science.

1.1 Thesis & Research Questions

Developmental guidelines (architectural patterns and interoperability methods) and intentional engagement practices with communities in applied domains (co-development practices) for the development and deployment of low-powered wireless sensing systems can be identified and

¹Core components of this thesis are published in venues belonging to these groups (chapters 3 to 6)

generalized, and doing so is a precondition for any shared framework that bridges prototyping and impactful field deployments. In this dissertation, I explore these generalizations across applications in mobile health and environmental conservation. I aim to address the following research questions, which are referenced throughout this dissertation.

RQ1 Application Framing: *How do embedded systems researchers ground their work for real-world applications?*

RQ2 Reusability: *What factors support or inhibit reusability in wireless sensing systems?*

RQ3 Prototype to Field: *What methodologies support the transition from technology prototype to field deployment?*

RQ4 Generalization: *What methodologies and challenges can be generalized across wireless sensing development?*

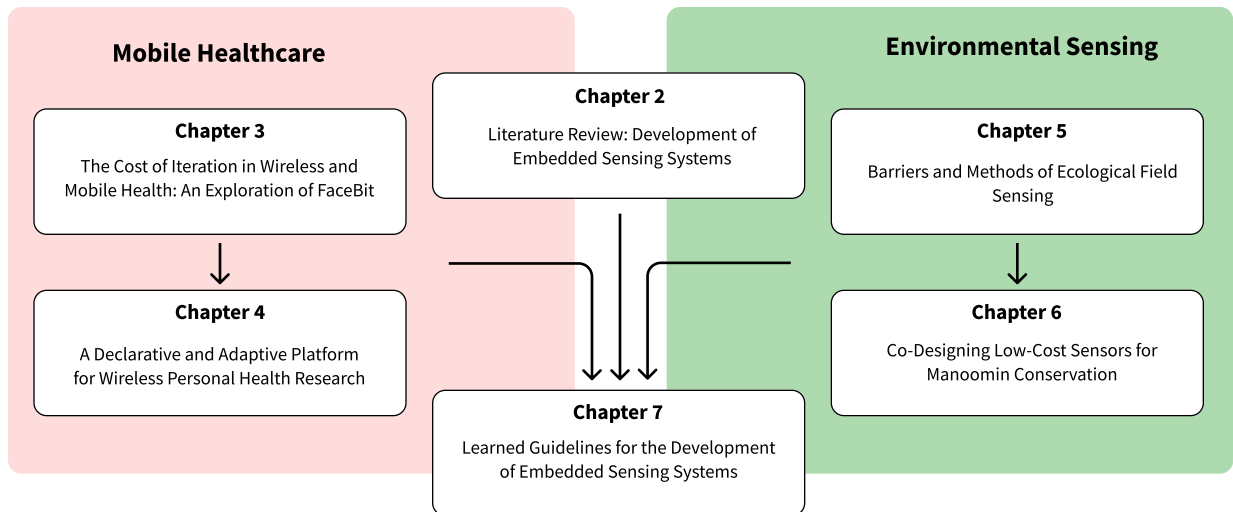


Figure 1.1: An outline of this dissertation’s progression by chapter. I cover two domains, mobile healthcare and environmental sensing, but outline and reflect on existing work. These first-hand experiences, plus a literature covering current development methods in embedded sensing systems, are synthesized into developmental guidelines.

1.2 Dissertation Overview

I address these questions by providing a comprehensive retrospective of existing case studies across the domains of mobile health and environmental conservation, and by examining broader work in the field. Throughout, I highlight common patterns and lessons learned in the development and deployment of systems, synthesizing these findings in the final chapter, in line with item RQ4. Finally, I aggregate core social and technological principles to build a roadmap for future tooling. The remainder of this dissertation is organized into the following chapters:

Chapter 2: Literature Review: Development of Embedded Sensing Systems

A review of the ACM literature post-COVID-19 pandemic, focusing on system contributions with specific applications in mobile health and ecological sensing. The review codes papers to categorize the state of application framing (RQ1), contribution reuse (RQ2), and evaluation (RQ3). I also inductively extract common challenges and methods throughout the text. This serves as a baseline of the current state of development for embedded sensing systems.

Chapter 3: The cost of iteration in wireless and mobile health

An introduction to the FaceBit smart mask system, including clinical need finding, system design and implementation, and evaluation. I then provide a retrospective outlining the project’s development methodology and the challenges we faced directly. This serves as a strong development case study that is often omitted from publications, providing first-hand insight to RQ3.

Chapter 4: A Declarative and Adaptive Platform for Wireless Personal Health Research

FlexiBLE was created in response to challenges in developing FaceBit. FlexiBLE is a proof-of-concept platform for rapid embedded sensing iteration, which includes a subset

of general features for communication and interoperability (RQ3). A detailed overview of FlexiBLE is provided, along with three case studies (RQ2). The chapter concludes with a retrospective based on three semi-structured interviews with FlexiBLE users, reflecting on strengths, weaknesses, and future directions (RQ4).

Chapter 5: Barriers and methods to field sensing

Using interview data from a previous environmental sensing survey [1], I introduce the field sensor life cycle, informed by insights from 13 participants (RQ3). A set of common challenges is presented that ecology and conservation science face.

Chapter 6: Co-designing low-cost sensors for Manoomin conservation

A presentation of Makak, a device co-designed with members of the Ojibwe Tribal Nations and organizations to conserve an Indigenous wild rice (Manoomin) (RQ1). An overview is provided of the development processes and two field deployments over two seasons (RQ3). Finally, a retrospective is conducted on the design iterations between the two deployments and on their generalizability to other ecological sensing projects and beyond (RQ4).

Chapter 7: Learned Guidelines for the Development of Embedded Sensing Systems:

A synthesis of the findings across the presented case studies and the literature, identifying developmental guidelines (architectural patterns and interoperability methods), and intentional community engagement considerations (co-development) that serve as a roadmap for future frameworks and tooling to streamline contributions to wireless embedded sensing systems.

1.3 Disclosures

Contributions of the individual projects presented in this dissertation, much of which are previously published works, are highly collaborative and cannot be attributed to me in full. In each chapter preamble and in the acknowledgments, I clearly state the previous publications, my specific contribution, and copyright when necessary. Unless otherwise stated, I am an author of all the works I include content from. Most text is reworked to fit the context of my thesis; however, some excerpts and many figures are verbatim from previous works. The overarching thesis, analysis, synthesis, and conclusions are of my own methodology.

Generative artificial intelligence (AI), such as chat-based large language models and agentic coding tools, was used throughout this work for organization, proofreading, editing, and general feedback. All assistance has originated from my own physical work, ideas, and opinions. Generative AI coding tools assisted significantly in LaTeX table formatting (not content), general LaTeX formatting, and the implementation of coding (computer *programming*, not *qualitative coding*), analysis, and tool development². In all cases, I specified the task and implementation details and verified the results; the methods, their selection, and interpretations of their output are my own.

²Section 2.1.3 describes specific generative AI uses for the literature review.

CHAPTER 2

Literature Review: Development of Embedded Sensing Systems

This chapter is a scoping review of recent embedded sensing systems from the Association for Computing Machinery (ACM) Digital Library (DL). The goal of this review is to build an empirical baseline of **how applied systems are motivated** (RQ1), **to what degree the technical contribution is reusable** (RQ2), **what methodologies are used in development and evaluation** (RQ3), and finally, **what common challenges exist across systems** (RQ4). This thesis argues that embedded sensing systems share many overlapping components across application domains; therefore, such a corpus would be useful to engineers. This review attempts to aggregate such works within the scope of this thesis.

This review is scoped to include broad applications in wireless and mobile healthcare, as well as environmental sensing, to match the first-hand development experience presented in (chapters 3 to 6). I critically examine the work’s methodology to understand *how* and *why* systems are built. To the best of my knowledge, this is the first attempt at such a study. What I discovered is that systems are framed in wide and inconsistent ways, and that implementation details are critical to reusability. Most notable is a lack of attention to detail on what I define as *technical scaffolding*. Works often name specific components, but rarely detail implementation or structure in enough detail to reuse the underlying technology. These are core topics in the synthesis chapter (sections 7.2 and 7.3).

The next section details the methodology, followed by a presentation of the findings through deductive and inductive coding mapped to the research questions. The review sets the stage

for future chapters in which I present and reflect on the first-hand development experience of similar systems. I use the details of this review as evidence in chapter 7 providing development guidelines and directions for future toolkit development. All papers and coding details are provided in a database published on Zenodo¹. I discuss potential next steps to extend this data in section 2.5.

2.1 Methodology

This is a scoping review [36] with a systematic approach to search, documentation, quantitative and qualitative coding to ensure reproducibility and facilitate future expansion. The search is scoped for both brevity and alignment with the thesis. I start with a set of research questions aligned to the overall dissertation research questions (RQ1 to RQ4), then use a 3-pass inclusion process, followed by full-text coding using my own tooling. The coding follows a deductive (pre-selected) and inductive (emergent) process. Quantitative and qualitative results are presented. The remaining subsections explain the methodology in detail; fig. 2.1 shows the 3-pass process review process.

2.1.1 Search

I completed three exploratory searches on the ACM DL. All searches targeted end-to-end embedded system contributions, with intended applications in wireless and mobile health, as well as environmental sensing. The focus is on current development methods, which change rapidly over the roughly three decades of applied wireless embedded reserach (the fields of IoT and mobile ubiquitous computing). I limited the time range for the search to recent works. Since the focus is on real-world applications and deployments, publications were limited to the period around the end of the widespread social restrictions imposed during the COVID-19 pandemic, which greatly limited in-person collaborative research. With this, the search is constrained to January 1st, 2022 (after most universities in the US returned

¹<https://doi.org/10.5281/zenodo.20617245>

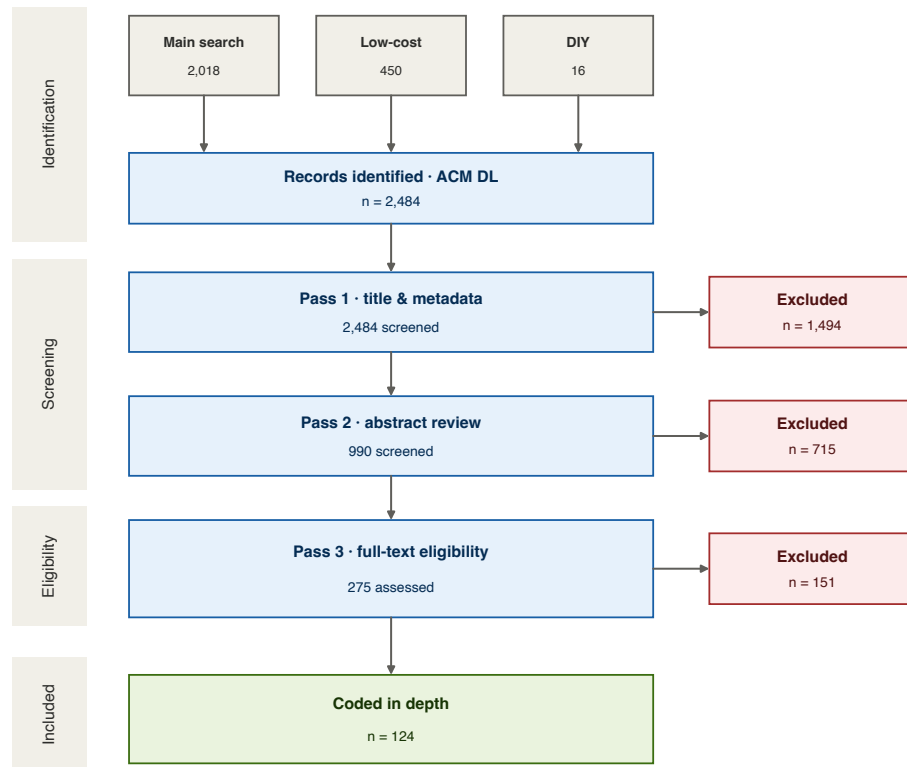


Figure 2.1: PRISMA-style screening flow, from the three ACM Digital Library searches through full-text eligibility to the 124 coded papers.

Table 2.1: ACM Digital Library search strategy. All searches were limited to research articles in the ACM Full-Text Collection (ACM Content: DL) with an online publication date on or after 1 January 2022. *Records* is the count retrieved before screening; *kept* is the count included and coded.

#	Scope	Query	Records	Kept
1	Primary	Abstract:((sens*) AND (embed* OR device* OR develop* OR hardware OR software OR firmware)) AND Abstract:((wearable* OR health*) OR (environ* OR conservation OR ecolog*)) AND Fulltext:((deployment OR eval*) OR (communit* OR design OR co-design OR collab* OR particapat* OR survey OR interview))	2,018	111
2	Low-cost	Abstract:(sens* AND low AND cost)	450	11
3	DIY	Abstract:(DIY AND Sens*)	16	2
Total records			2,484	124

to in-person courses) to December 31st, 2025 (shortly before screening started). I manually extracted BibTeX references from ACM DL, then pulled normalized metadata using the Semantic Scholar API². Papers were downloaded directly from the ACM DL using access provided by Northwestern University. Table 2.1 shows the three search queries, along with the number of papers returned for each.

2.1.2 Screening

The screening process was completed in three phases, based on predetermined inclusion (table 2.2) and exclusion (table 2.3) criteria that are scoped to the interests of this dissertation. This same criterion was used throughout all screening processes. Below are explanations of each phase in the screening:

Pass 1 The title and metadata (keywords) were used to determine exclusions. Ambiguous titles were given the benefit of the doubt (included).

Pass 2 The paper’s abstract was parsed and excluded only with a clear indication of the exclusion criteria.

²<https://www.semanticscholar.org/product/api>

Pass 3 This screening was completed alongside coding (section 2.1.3). First skimmed the text for exclusion before fully reading the text. In some cases, coding was started and then stopped if an exclusion criterion was found.

Table 2.2: Inclusion criteria. A paper is included only if all criteria hold.

Code	Title	Criterion
I1	System type	<i>Describes a wireless embedded sensing system (see Chapter 1).</i>
I2	Application domain	<i>Targets a mobile health or environmental sensing domain, or closely relates to either (see Application Type description)</i>
I3	Artifact	<i>Authors designed and/or built the artifact (custom hardware, firmware, or system integration).</i>
I4	Reviewable	<i>Provides sufficient technical detail for code development methodology, evaluation, and system composition.</i>
I5	Access	<i>Full text accessible via the ACM Digital Library; published between 2022-01-01 and 2025-12-31; peer-reviewed, “Research Article” type provided explicitly from the ACM DL search filtering.</i>

2.1.3 Coding

I coded the full text of each included paper, pairing deductive and inductive coding with the research questions, and inductively coded emergent themes. Coding drew on a codebook of 14 parent codes; the complete codebook, with sub-codes and definitions, is given in the published data set. I present the coding in a deductive matrix and anchor the findings in section 2.3; the remainder are free-form codes applied in line during reading. *Application Type* and *Application Framing* describe how systems are grounded in a target application (RQ1). *Reuseability*, with individual scoring for *hardware*, *software*, *firmware*, *data processing* captures reproducibility of the technical contributions (RQ2). *Evaluation* trace the transition from prototype to field (RQ3). Finally, cross-cutting codes *Challenges*, *Limitations*, and *Key Insights*, along with a code used for free-form notes on important developmental details,

Table 2.3: Exclusion criteria and application counts across all three screening phases. Each paper may have multiple exclusions.

Code	Title	Criterion	n
EX1	High-power compute	<i>High-power or high-dimensional signal processing (continuous image, video, audio, or RF capture; typical power ≥ 500 mW; microprocessor with MMU boundary compute, e.g., Raspberry Pi and above).</i>	270
EX2	COTS platform	<i>Consumer off-the-shelf device used as the sensing platform (smartphones, smartwatches, consumer wearables) without a custom embedded artifact.</i>	213
EX3	Out-of-scope domain	<i>Non-medical, non-environmental application (vehicles, UAVs, drones, industrial automation, industrial agriculture, VR/AR, entertainment, etc.).</i>	622
EX4	Application-agnostic	<i>Primary contribution is a novel protocol, security mechanism, or a general technique without targeted application grounding.</i>	506
EX5	No embedded artifact	<i>No system built by the authors; secondary analysis, public-dataset reuse, pure simulation, or algorithm on pre-existing data.</i>	912
EX6	Review or survey	<i>Paper is a literature review, survey, or meta-analysis.</i>	39
EX7	Insufficient detail	<i>No evaluation, no technical methodology, or contains clear errors.</i>	27
EX8	Other	<i>Case-specific exclusion documented in reviewer notes, usually tagged as useful context for the thesis but not fit for this corpus. Accompanied by an additional exclusion criterion classification.</i>	24

surface the generalizable lessons synthesized in the discussion (RQ4). The remaining code provides supporting context.

Deductive Matrix

I condense each paper into a deductive matrix for quantitative analysis. Four fields are load-bearing: Application Type, Application Framing, Evaluation Type, and a four-dimensional Reusability score (hardware, firmware, analysis, and software). I detail the rubrics in the coming section, where I discuss each field. I additionally recorded descriptive fields for context: application target, sensors, actuators, data-collection technology, microcontroller, embedded platform, interface platform, functional artifact, and external resource link. An excerpt from the selection of works, with a subset of matrix columns, is shown in table 2.5. This matrix is created using columns that map to related codes in the codebook. After the coding is complete, I manually fill in the matrix to provide an aggregate summary of the coding; `codes` and `matrix` entries are separate in the data (section 2.1.3).

Inductive

Not every code reduces to a categorical field or qualitative metric. During full-text reading, I also applied free-form codes, such as *Development Method*, *Data Collection Method*, *Challenge*, *Limitation*, *Design Consideration*, *Technology Selection*, and *Key Insight*. I also optionally include a memo to provide additional context for the annotation. These codes capture recurring, cross-cutting observations that the deductive matrix does not. Given the scoping nature of this review, I do not reduce them to a formal thematic analysis; instead, I report a focused subset of observations in the discussion (section 2.4), illustrated with representative excerpts, and defer their synthesis into generalizable developmental guidelines in chapter 7, where this coding serves as primary evidence. The complete set of emergent codes, memos, and excerpts is included in the published dataset.

Data & Tooling

Screening and coding were completed using custom tooling. This allowed freedom to develop a custom process for this review and complete control over the data. This system included a locally hosted web application and a SQLite database that tracked all details of the methodology. Figure 2.2 shows the application user-interface.

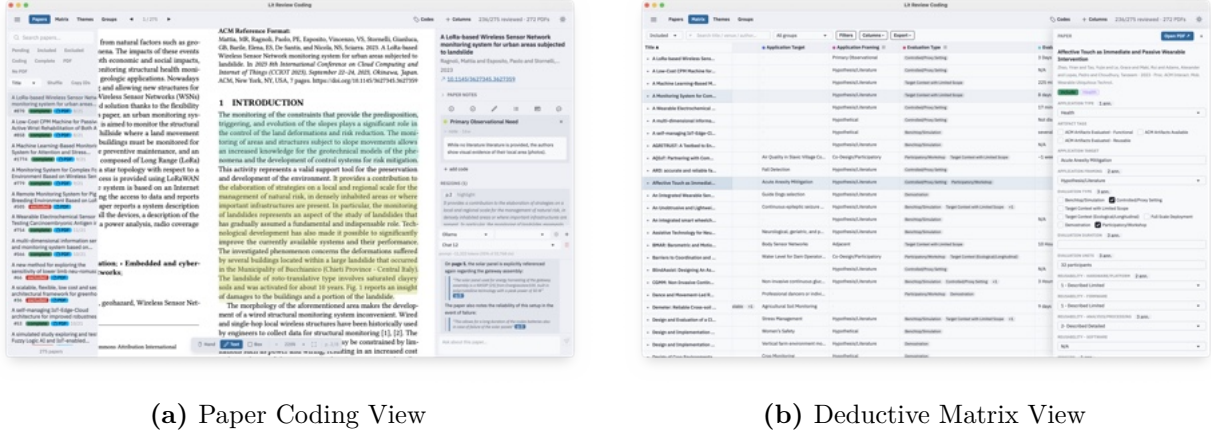


Figure 2.2: The web-based coding application used in the literature review.

I used large language models as an experiment in coding assistance, modeled on features provided in ATLAS.ti³, a common proprietary qualitative coding software suite. Before coding, for each paper included after pass 2, I exported the text and fed a series of prompts, built using inclusion and exclusion criteria and the codebook, into an open-weight model Qwen version 3.5 (Qwen3.5-35B-A3B) [37] running on a self-managed cloud server. This was used to identify information during coding and, overall, was limited in usefulness, but it was a learning experience in the possibilities of LLM use in literature reviews. It was most useful as a reference post-initial coding to identify missing features, such as mentions of deductive coding elements and the specific names of microcontrollers or programming languages. All coding was completed by hand by the author.

³<https://atlasti.com/>

The following presentation is a limited representation of the entire review, focused on the research questions. For a complete and comprehensive dataset that includes all codes, themes, and LLM input and output, see the release on Zenodo at <https://doi.org/10.5281/zenodo.20617245>. The database does not include copies of the papers, but they are referenced by DOI and include public metadata from Semantic Scholar.

2.2 A Corpus of Embedded Sensing Systems

The corpus includes a total of 124 papers after all screening passes. In this section, I present high-level findings of the data. Within the corpus, 87 papers (70%) were published in conference proceedings, and 37 (30%) in journals. There are a total of 62 venues represented in the corpus; The Journal of Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) is by far the top with 26 papers (21%). Figure 2.3 details the venues with more than one entry; 41 venues have only a single representative work.

These works span a wide range of lengths and levels of detail without formal metadata differentiation. The coding results show this variation, but page counts give us the most rudimentary metric of amount of detail; the mean page count across the corpus is 15.16 (min: 4, max: 49, median: 12). Under the ACM DL filter of "research article," I find full, long-form journal articles and short papers with little to no evaluation. To the best of my knowledge, this is a limitation of the metadata in ACM DL, and without prior knowledge of the venues, there is no clear way to filter by contribution type or level of detail. This review is an attempt to explore that variety. Short and long works alike contain useful *technical scaffolding* and details useful to anyone looking to develop similar systems; therefore, length was not an inclusion factor. I present the corpus descriptively in two parts: the applied context each system targets and the technologies from which systems are built, before turning to the research-question-aligned coding in section 2.3.

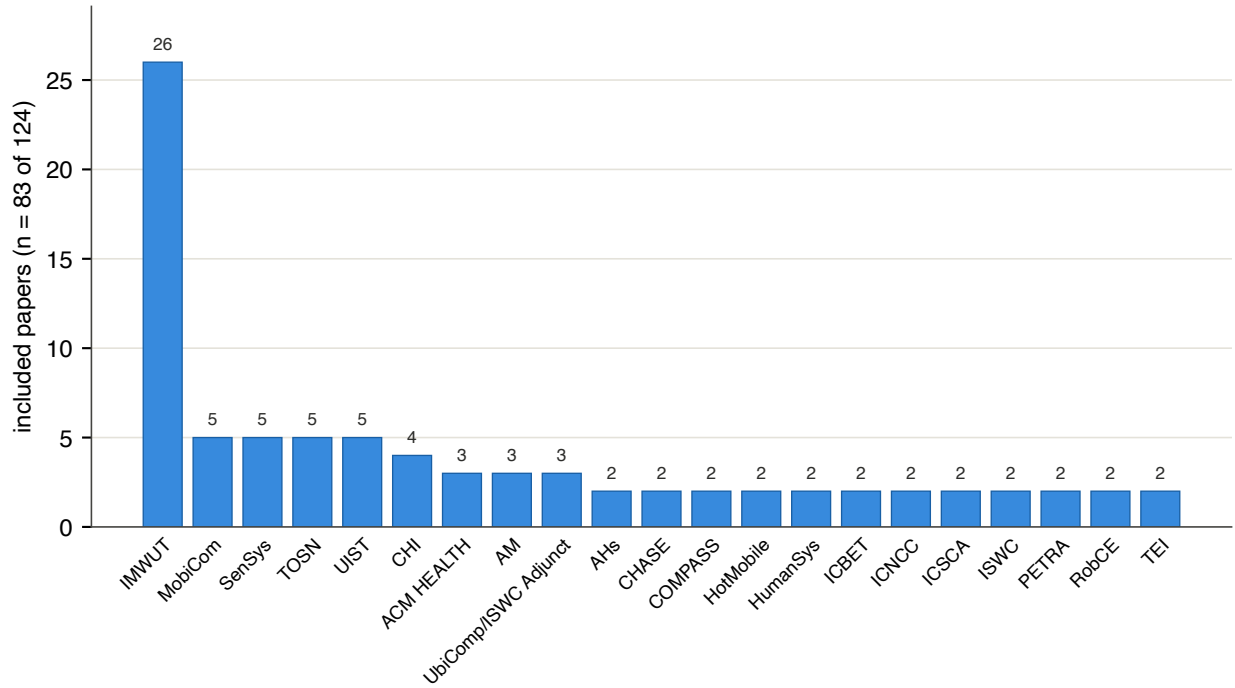


Figure 2.3: Chart of ACM venues in the corpus with at least 2 representative works. There are 41 venues with a single entry.

2.2.1 Application Type

The first coding indicates the work’s applied context, i.e., *what is the system’s intended application?*. I group the corpus by *Application Type*, which I specifically limited to complement the preceding development case studies (chapters 3, 4 and 6) and my own experience. Table 2.4 details the coding. Health is vastly more popular in the corpus than Environmental, 69% ($N=86$) and 22% ($N=27$) respectively. This is most likely due to the humanistic aspects of the fields of Ubiquitous Computing (UbiComp) and Human-Computer Interaction in ACM DL. I tried to categorize all papers into a single application bucket; one ZenseTag [38] paper truly covered both, providing an example application in health and agriculture. A few papers targeted environmental sensing for health-related applications; for example, C-Ridge [39] monitored indoor CO_2 levels in large venues, and [40] monitored domestic water consumption. I tagged these based on their sensing type (environmental-focused).

The *Other* category is reserved for devices that are adjacent to the target scope and provide interesting technical or co-development detail, but do not fully fit into either Health or Environmental. [41] demonstrates a IMU enabled wrist device which is used by performance dancers to build interactive lighting visualizations, [42] is a custom device which embeds a type of mold which becomes dormant without interactivity, and SmartDampener [43], which focuses on Tennis analytics. Application Type concerns only how the authors present the intended or envisioned applied use of the device; how these systems are grounded in their specific applications is discussed in section 2.3.1.

Table 2.4: Deductive codes for *Application Type*: the target domain of each system’s contribution. Counts reflect the 124 Phase 3 included papers.

Code	Definition	n
Health	<i>Focused on wireless and mobile healthcare; wearable health devices or adjacent. Applications are broad from activity tracking to specific specialized populations. Also includes devices focused on accessibility.</i>	86
Environmental	<i>A focused on in-situ environmental monitoring with an aspect of conservation or social good. Excludes industrial agriculture, but some agricultural devices were included along with urban/city sensing.</i>	27
Other	<i>Reserved for a limited number of papers that do not directly fall into the two other categories, but have a strong argument for placement into the corpus.</i>	11

2.2.2 Wide Ranging Technology

The corpus spans a wide range of reported technologies, yet that range resolves into a small, recurring set of building blocks. I categorize the technologies as follows: the sensors and actuators that define a device’s modality; the microcontrollers and embedded platforms on which firmware runs; the link-layer methods that move data off-device; and the interfaces through which a system is operated.

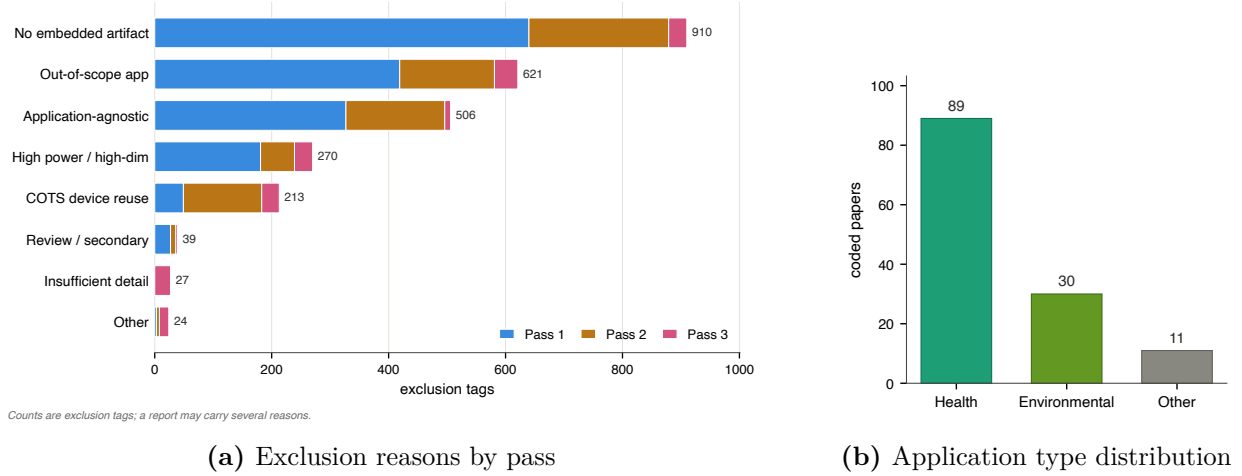


Figure 2.4: Screening detail. (a) Exclusion reasons tagged at each screening pass; and (b) application type of the 124 coded papers.

First are the types of sensors used in devices, which can be multiple per device. IMU is the most commonly reported (33%, $N=33$). Second are *custom* sensors (32, $N=26$), which are works that have specifically developed their own sensing modality, rather than using an existing integrated circuit. Examples include electrochemical analysis of sweat [44], a dental retainer sensing teeth contact [45], and a battery-free compatible impedance sensor for tree-health [46]. In these cases, the sensor itself is the contributing artifact, rather than the type of use. After IMU, temperature (19%, $N=24$), Electrical (e.g., conductance or analog signal, 12%, $N=15$), Light (12%, $N=15$), photoplethysmography (PPG, 10%, $N=13$), and air quality (particulate matter, 9%, $N=11$) round out the top 5. Unsurprisingly, these components are ubiquitous in many wireless devices. Actuation, not a requirement for the corpus like sensing, is present in 33% of works; the top types are speakers (14%, $N=17$) and LEDs (13%, $N=16$). Actuation is often found as a method of aiding sensing [47–52] or for human-interface [53–59]. Most of the corpus includes passive devices, sensing with minimal or no alteration to their environment, but a few implement control, for example in energy harvesting [46], or irrigation [60, 61]. Sensor and actuator individual part numbers are often mentioned, but are not formally tracked in the data.

The categorization of microcontrollers (MCUs) varies across the text, which isn't surprising given the branding of integrated circuits, systems-on-chip (SoCs), and physical architectures. MCUs are commonly referenced by branded models, which in turn imply specific toolchains. The coding follows suit, but note the caveats. Espressif's ESP32-series MCUs are the most common (19% $N=24$), followed by Nordic Semiconductor nRF-series (15%, $N=19$), and STMicroelectronics STM32-series (15%, $N=19$). ESP32s support both BLE and Wi-Fi, but come in different variants (S and C), each with different processors masked by the toolchain; nRF implies lower-power BLE support, and STM32 has no networking capabilities out of the box. These nuances are fundamental to implementation; however, the norm is to state the MCU family and move on. 68% of works do not report the embedded platform, the tooling on which the firmware is built, whereas 86% report the MCU.

Ten works (8%) are not built around an MCU; computing is instead hardwired or fixed-function logic. These cluster at the extreme low-power edge, or are components of a system. These include energy-harvesting and battery-free design such as a triboelectric textile [62], a soil microbial fuel cell [63], a zero-power RFID contact lens [64], custom analog sensing front-ends [65], and dedicated accelerators for intermittent operation [66].

Embedded platforms have a similar nuance: Arduino is the most reported (18%, $N=22$); however, it comes in different forms. Arduino is a company and also a development platform that spans 8-bit (Atmel) and 32-bit Cortex-M MCUs. Traditionally, Arduino is a bare-metal (no operating system) abstraction that comes with a set of libraries and an ecosystem, including an IDE and community support. Implementations of Arduino also exist for FreeRTOS, Mbed, and Zephyr RTOS; these details are almost always omitted unless code is provided. All other platforms in the corpus have occurrences of 3 or fewer. In some cases, it may be implied that Arduino is used, given its popularity, but I do not make an assumption in my coding. This and other inconsistencies in implementation reporting make reproducibility challenging (section 2.3.2).

Data transmission methods are reported in most works: 103 (83%) name an explicit transport, while the remaining 21 (17%) store data on-device, deferring the transmission to future work, or omit it entirely. BLE is the predominant method with 45 instances (36%), followed by WiFi (18%, $N=22$), wired serial (15%, $N=18$), and LoRa (8%, $N=10$). Applications often drive communication methods. LoRa, a long-range, low-power radio technology, is common in environmental deployments with no available infrastructure; all 10 occurrences are environmental applications, whereas BLE is used for short-range, is integrated into most consumer electronics, and is common in body sensor networks; 40 occurrences are health applications, 5 are other. I include instances of wired data collection used for early prototyping when wireless is the eventual intention [67–70]. Reporting, however, rarely extends beyond this physical- and link-layer choice. How data is structured, packed, and timestamped, the application-layer concerns that govern interoperability and reproducibility, are seldom described. This gap is discussed in section 2.4.

Finally, each system is tracked to see how it is operated and how its data is surfaced. Interfaces are as varied as the hardware. Mobile apps are most common (29%, $N=36$), followed by desktop applications (15%, $N=19$), web portals (15%, $N=18$), and serial or scripting consoles (15%, $N=19$) typical of research-grade prototypes. A substantial share exposes no interface at all (17%, $N=21$), being headless sensing nodes (often a system-component artifact, or collecting data internally), and some leave it undescribed (13%, $N=16$). The interface’s implementation is largely undocumented, with 40% ($N=50$) naming no platform, and Android is the most common when stated (12%, $N=15$). As with firmware, the companion software that makes a device usable is rarely reproducible, compounding the reuse gap examined in section 2.3.2.

Taken together, the census runs counter to the corpus’s surface diversity. Systems with vastly different applications are assembled from a small set of technologies: a few MCU families, a handful of sensing modalities, and a limited number of wireless technologies. It is this

recurrence across otherwise unrelated applications, rather than the surface variety, that points to a generalizable substrate beneath application-specific contributions. This motivates a corpus that an engineer could search by component rather than by domain.

Table 2.5: Representative sample of the coded corpus across the deductive matrix; one paper per row, cited by number (titles omitted). *Reuse* = Hardware/Firmware/Analysis/Software rubric (0–4, table 2.7); PW = previous works, CoTS = off-the-shelf, — = N/A. *Framing* (table 2.6): Hyp/Lit, Hypot, Adj, SecObs, PrimObs, Co, Co+. *Eval*: Bench, Ctrl, Demo, Wkshp, TCLtd, TCFull. Multi-value cells show the top 3; the full corpus and all columns are in the data release.

Ref	Type	Target	Framing	Eval	Reuse	Sensors	µC	Transport	Interface
[48]	Health	Prediabetic screening	PrimObs	TCLtd	4/3/3/PW	Light	SAMD21	WiFi	Web
[57]	Health	Dementia management	Hyp/Lit	Demo	4/4/2/3	Rotary	ESP32	BLE	Onboard, Desktop
[49]	Health	Eearable research	Hyp/Lit	Ctrl, Demo, Wkshp	4/4/4/4	IMU, PPG, Press +1	nRF	BLE	Mobile, Desktop
[59]	Health	Physical activity (child)	Co	Demo, Wkshp	2/1/1/1	IMU	ESP32	BLE	Mobile
[71]	Health	Fabric Interaction	Adj	Demo	4/4/4/—	IMU	STM32	Envis.	Desktop
[72]	Health	Diabetes Diagnosis	Hyp/Lit	TCFull	2/1/2/1	Gas	ESP32	WiFi	Web
[73]	Health	Cuffless blood pressure	Hyp/Lit	Bench, Ctrl, TCLtd +1	3/3/3/1	PPG, Temp, Force +1	nRF	BLE	Mobile
[46]	Env	Tree health	Hyp/Lit	Bench, Ctrl, TCLtd	2/2/2/0	Custom	MSP430	LoRa	Serial
[74]	Env	Juvenile Chinook salmon micro...	Co+	Bench, TCFull	2/2/2/—	Temp	MSP430	LoRa, Custom	—
[75]	Env	Soil monitoring	Hyp/Lit	Bench	4/4/1/—	Custom, Temp, Electrical	Teensy	Wired	None
[76]	Env	Soil Monitoring	Hyp/Lit	Bench, TCLtd	2/4/2/—	Soil	None	LoRa	Serial
[77]	Env	Water Level	Co	Wkshp, TCFull	1/1/2/1	US	PSoC5	Cell	Web
[41]	Other	Interactive Dance	Co	Wkshp, Demo	PW/PW/2/—	IMU, Force	—	—	Screen

2.3 Framing, Reuse, and Evaluation

I organize the three deductive classifications in the order in which a contribution is typically presented: from motivation through implementation to evaluation. **Framing** (RQ1) captures how a system is motivated for a real-world application; **Reuse** (RQ2), how fully its implementation is detailed, and how readily a peer could extend or reuse its components; and **Evaluation** (RQ3), how, and how thoroughly, a system is validated against that motivation. The data is presented for each in turn. The section remains descriptive, as these findings support the synthesis (chapter 7), which discusses how they generalize to system design.

2.3.1 Framing Systems in Applied Contexts

A work is *applied* when it specifically frames its contribution within an intended deployment context, the narrative of *why* a system should be built and why the community should care, rather than motivating it purely by advancing a technique or field. To address RQ1, the *type* of framing each work uses is coded on a scale of external engagement, from *adjacent* to *co-design/participatory in context*, as defined in table 2.6. Each work receives a single code, its *highest* level of engagement, though it may carry multiple framing annotations in the data.

The majority of works do not motivate with engagement from the target context, more than four in five (84%); they do so from previous literature (62%, $N=77$), by hypothetical assertion (16%, $N=20$), or as an adjacent use case (6%, $N=8$). Direct engagement is the exception: fewer than one in ten draw their motivation from the field. Co-design appears in 8% ($N=10$) of works, and a single paper reports *in-context* collaboration, in which an external collaborator from the target domain is included as an author [74]. I argue that in applied work, it is this engagement, not a literature gap, that most convincingly demonstrates usefulness; the corpus shows how rarely this is pursued when originating from the computing field.

Table 2.6: Deductive codes for *Application Framing*: how each paper grounds the motivation and problem statement for its contribution. Counts reflect the 124 Phase 3 included papers.

Code	Definition	n
Adjacent	<i>The target application is not primary but positioned as a secondary or add-on use case, with or without literature support.</i>	8
Hypothetical Evidence	<i>no clear link between previous works and the motivation of an artifact; framing relies on a hypothetical explanation or logical leap.</i>	20
Hypothesis / Literature	<i>Framing is drawn directly from existing literature; the need has been identified by the academic community.</i>	77
Secondary Observational	<i>Motivation is built from secondary stakeholders (e.g., clinicians consulted instead of the target patient population).</i>	4
Primary Observational	<i>Motivation is built from a formative study with target users, or from prior iterative work by the same authors.</i>	5
Co-Design / Participatory	<i>The target community is involved in co-design or participatory activities to elicit iterative feedback. Authors may be from the community members, but are not influential stakeholders.</i>	9
Co-Design / Participatory in Context	<i>Co-design where domain experts are included in authorship and make substantive contributions to the work itself.</i>	1

Framing a contribution around a literature gap is the appropriate norm for both fundamental and incremental research; a system can be rigorous and valuable regardless of its grounding, and some gaps are common knowledge and do not require additional justification. However, in specialized *applied* work specifically, external engagement is the strongest available evidence that a technology will be useful where it is deployed. I therefore read the distribution as an opportunity rather than a shortcoming. Co-design and participatory methods remain underused for demonstrating applied value. This engagement is examined first-hand in chapters 5 and 6 and synthesizes co-development guidelines in chapter 7.

2.3.2 Describing System Implementation & Reuse Expectations

The interest lies in the underlying generalizable technology that drives specific systems (RQ4). An incremental step is to categorize the *reusability* of existing systems in the literature. This coding focuses on the level of detail provided in the technical implementation of their systems. It is assumed, given the peer-review process, that adequate detail is provided on the *specifics* of individual projects, but it is well known that implementation details are secondary to novelty in the text. However, I argue that these implementation details, the *technical scaffolding*, are a meaningful component to reuse and reproducibility (RQ2). Here, *reuse* is specifically defined as the ability of a competent peer to implement a similar system, as opposed to full *reproducibility*. It's a subtle difference, and reuse certainly falls under reproducibility; however, it is less concerned with replicating results and more with applying the technology to *new* uses. Here, I generate a reuse rubric (0-4) and track it against the four technological dimensions in the text: hardware, firmware, data analysis/processing, and software. This rubric is presented in table 2.7.

Interesting patterns emerge in this categorization. A wide variety of implementation details are observed, but reporting is uneven across the system's sub-components. First, source code is rarely available across all components. In total, 24 (19%) works by supplying an external link to source material, not all follow through with promised release, see section 2.1.3. In

Table 2.7: Reusability rubric. Each paper is scored independently along four dimensions—**H**ardware / Platform, **F**irmware, **A**nalysis / Processing, and **S**oftware—using the levels below. Scores reflect what can be reused from the paper text *or* from the public resources it references. Not every paper is scored on every dimension (e.g., software may be *N/A*), so column totals need not equal 124.

Level	Label	Definition	H	F	A	S
0	Unavailable	<i>Mentioned with no implementation detail.</i>	3	12	5	27
1	Described, Limited	<i>Missing key implementation details.</i>	35	72	19	33
2	Described, Detailed	<i>Sufficiently detailed description to be reasonably reproduced from the text alone.</i>	72	17	83	7
3	Available	<i>Source files available.</i>	4	6	5	1
4	Documented	<i>Source files available and well-documented.</i>	8	5	3	1
—	N/A	<i>The dimension does not apply to the contribution.</i>	0	7	8	50
—	Previous Works	<i>Uses previous literature works.</i>	2	4	1	4
—	CoTS	<i>Uses a purchased turnkey solution.</i>	0	1	0	1

total, 11 works supply source material for hardware, 12 for firmware, 6 for data analysis, and 2 for software. It's nearly half-and-half if sources are sufficiently documented. During coding, I followed all provided links and skimmed the work to determine documentation, which is limited since I did not verify any materials in detail. When source materials are not supplied (81%), reuse must be gauged from the text alone, which is challenging; hence, the broad categorization. The initial findings supply interesting evidence on the detail that is provided, and where it is not.

Figure 2.5 shows the breakdown of each dimension by description categorization. All works supply some hardware artifact, which is an inclusion criterion. The majority of *Hardware* and *Data Analysis* descriptions are detailed or provided, 68% ($N=85$) and 74% ($N=92$) respectively. However, firmware and software have greatly reduced detail, 23% ($N=29$) and 7% ($N=9$) respectively provide enough for reuse. Software supporting data collection or device interaction is absent from 40% ($N=50$) of works. The latter makes logical sense: hardware and data analysis constitute specific contributions in applied implementation, whereas software and firmware are simply *implementation details*; therefore, a generalizable system infrastructure. When details were provided, there does not seem to be any consistent pattern in how or to what degree systems were explained; in one instance, all specific implementation details were included in an appendix [68]. The problem is that without this underlying infrastructure, system reuse becomes, at best, burdensome, as there is no source material or supporting tooling to fill the gap.

It is important to realize that sufficient detail in text alone, without a released artifact, still requires significant labor to reuse. The four dimensions, taken as a whole, support the *reproducibility* of a given work; they are split to gauge the potential for technical extraction of finer-grained system-level details that may be useful across applications. For example SmokeMon [78] a low-power thermal-imaging based smoking detecting system targeted for behavioral intervention for cessation contains a useful, custom built, infrastructure for

behavioral just-in-time intervention study management and deep technical knowledge of low-power thermal sensing. The ability to search for works with highly detailed firmware and hardware, and implement PPG [49, 73, 79] or IMU [49, 68, 71, 80] sensing can be very valuable to projects seeking a starting point based on those specific sensors, regardless of application. Currently, to the best of my knowledge, no knowledge base like this exists, and generally system infrastructure is being rebuilt from scratch repeatedly.

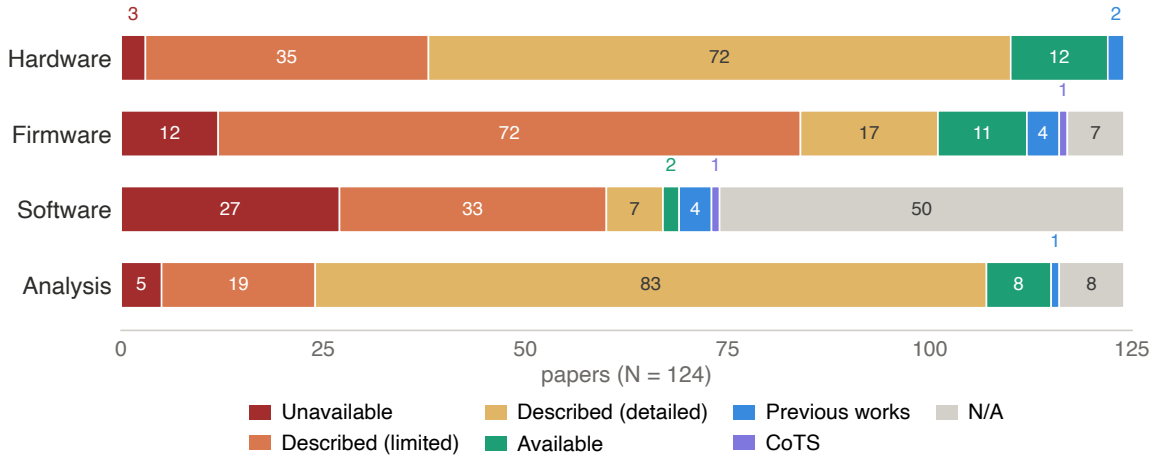


Figure 2.5: Reusability by component

2.3.3 System Evaluation: Prototypes to Field

All technical contributions in the corpus include an evaluation that supports a claim about their usefulness in a given application. The final coding categorization looks at *how* systems are evaluated and to what degree within their target context. Evaluation characterization is not singular; studies often perform multiple evaluations (46%, $N=57$), especially larger studies such as those published in IMWUT [47, 48, 50, 63, 78, 81]. *Participatory/Workshop* methods, when present, are always conducted alongside other evaluations; they include usability surveys [53, 73, 82, 83], interviews [83–85], and observational workshops [58, 73, 84]. These types of evaluations provide sociotechnical insight into the contributions, usually in terms of applied potential or how the system can be improved for future work. Given the

cross-disciplinary and technology-focused methods, code evaluation results are not tracked. Table 2.8 defines each of the evaluation categories, and fig. 2.6 visualizes the distribution.

Unsurprisingly, *Controlled/Proxy setting* evaluations are the most numerous in the corpus (48%, $N=60$). These are typical scientific evaluations where a specific function is tested under controlled conditions and where participants, if any, are not used as a target of the evaluation. For example, in mobile healthcare, recruiting peers for convenience to test the capabilities of technology rather than its intended user base [86–89], or in environmental, testing in settings where the environment can be directly controlled [90]. *Benchtop/Simulation* is also very common (40%, $N=50$), and is often included as a preliminary step before an in-wild or controlled evaluation. These are intended to validate core technology.

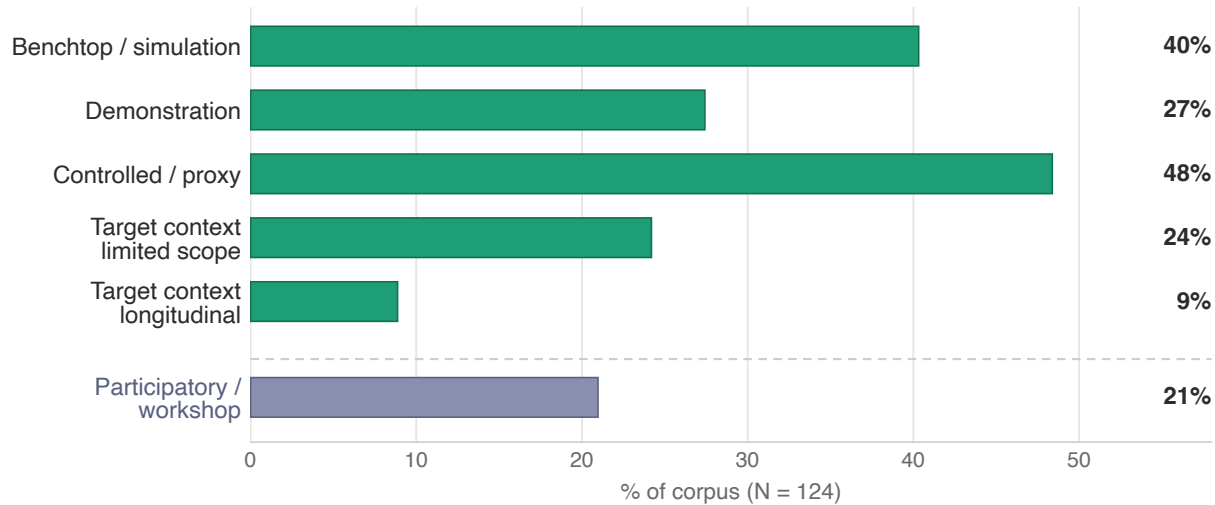


Figure 2.6: Distribution of evaluation methods across the corpus. Each study may have multiple methods. *Participatory/Workshop* always accompanies another method.

What is of more interest to the thesis, specifically RQ3, is the system’s progression from typical scientific validation to *in-wild* evaluation settings; the progression from prototype to the field. In total, 40 works include some form of evaluation in their respective application domains; 30, with limited scope and supervision; and 11 are settings that are meaningful in scope and duration, clearly going beyond a sole technical field validation (one paper provides

Table 2.8: Deductive codes for *Evaluation Type*. Multiple codes may apply to a single paper (e.g., benchtop validation followed by a limited-scope deployment), so the counts sum to more than the 124 included papers.

Code	Definition	n
Benchtop / Simulation	<i>Proof-of-concept testing in a controlled setting outside of any real-world context; targets specific components and leaves remaining implementation as future work.</i>	50
Feasibility Demonstration	<i>Showcase of a design or interface’s capabilities. Participants, if present, they are a means of demonstration rather than a target population or a metric of evaluation.</i>	34
Controlled / Proxy Setting	<i>Evaluation in a controlled setting with convenient participants or locations; assesses the technology itself, not usability in deployment.</i>	60
Target Context, Limited Scope	<i>Evaluation in the intended deployment environment but with limited duration, scale, or with heavy research team observation or assistance.</i>	30
Target Context (Ecological / Longitudinal)	<i>Full evaluation in the target context for a meaningful timeframe; still focused on the technology and its usability rather than full deployment.</i>	11
Participatory / Workshop	<i>Controlled use observed in a workshop; the goal is iterative refinement through user participation, feedback, or observation.</i>	26

both). These evaluations indicate that the system is exposed to the stochastic nature of the field, in which participants' or the environment's actions are not controlled. These types of evaluation are, obviously, more complicated but are a required stepping stone to contributions that have an impact in their target domains. An impactful deployment that fulfills a goal outside the context of computing is not tracked in this corpus, since ACM venues would not be the proper place to publish; see section 2.5.

2.4 Discussion

In this section, two of the emergent themes are explored based on the coding. However, the key findings are reserved for the synthesis chapter, where corpus coding serves as evidence for the proposed developmental guidelines (chapter 7).

2.4.1 Field Variability and the use of Artificial Intelligence

Many studies highlight the challenges and limitations of sensing variation in uncontrolled environments (e.g., the field). Initial device evaluations (benchtop, though controlled/proxy) evaluate technology under predictable, controlled conditions, often with additional external data for comparison (ground truth). However, when uncontrollable conditions are introduced, systems must adapt to noise and be hardened for field use. Health studies often mention physical differences in participant traits and filtering noise from movements in regular use [41, 45, 53, 57, 58, 64, 65, 68, 89, 91–93]. In both health and environmental application methods of calibration are also mentioned as ways to handle these scenarios [76, 94]. Studies are often too limited in scope to address all variations required for adaptation to full-scale deployments, leaving this to future work; another example of the *Valley of Death* in moving prototypes into applied use. One common recent method for dealing with variation is applying AI/ML to learn patterns in data.

The use of Artificial Intelligence is very common in the corpus; 73 studies (59%) report using AI to some degree, often as a contribution to the work. More work is required to understand and categorize the different types of AI use in the domain of wireless embedded sensing systems, as they include statistical ML models [54, 82, 95], deep learning for more complex tasks [93, 96–101] and in some cases Large Language Models (LLMs) or Transformer architectures [68, 83, 87, 102, 103] computed in both cloud (upstream application) and edge inference (on embedded device). By definition, AI/ML is used to generalize data in a way that heuristics fail by utilizing high-dimensional representations, addressing the challenges of individual variability [45, 81, 104], reducing the need for calibration [81], or filtering out noise [105], however collecting enough data from training in a constant challenge for studies [68]. More work is required to properly categorize and draw conclusions about the use of AI and its effect on mitigating variability, but it is a common claim in recent work.

2.4.2 Publishing Source Materials

In the corpus, 24 studies (19%) provide an external resource, e.g., supplemental material such as source files or data. For each provided link, the contents are analyzed and found to have little standardization or verification of source materials. Of the 24 studies 18 are provide links for GitHub⁴ [42, 48, 56, 57, 68, 71, 73, 75, 76, 78, 89, 106–112], three custom sites [43, 49, 83], two Zenodo⁵ [39, 113], and one Cornell University eCommons⁶ [97]. Two works [76, 107] include ACM artifact badges [114], and the rest state the contents in the text. At the time of writing, 11 sources have either no license or are not available; 11 have an open-source license (MIT, CC-BY-4.0, GPL-3.0, LGPL-2.1); and 2 have a custom license. Six instances were found of inconsistency where the text indicates a source material release that is not present in the link provided [43, 78, 83, 106, 107, 110]. This describes an inconsistency landscape of open source material, which is problematic to reuse and reproducibility (RQ2).

⁴github.com

⁵<https://zenodo.org>

⁶<https://ecommons.cornell.edu/>

Releasing source material is not required, but it is not uncommon in research. The problem is not maintaining that access after publication. ACM artifact badges are an attempt to address this problem, but are not widely adopted and, unless validated, suffer from similar consistency issues [115]. Similarly, git (e.g., GitHub) repositories can be altered after publication, making them less than ideal for publicly sharing static digital source material. GitHub is ideal for code in active development; for example, OpenEarable 2.0 [49] provides a well-maintained GitHub Organization and website for the project⁷. Zenodo, on the other hand, is specifically designed for timestamped, versioned source material and provides a Digital Object Identifier (DOI), which is well-suited for one-time releases. More work is required to better classify external artifacts and maintain better source files when open release is the goal. Sharing of source material, even in *as-is* form, can serve as starting points or reference material for new projects; doing so is not incentivized.

2.5 Limitations and Future Work

This scoping literature review provides foundational evidence on the state of wireless embedded sensing systems in mobile healthcare and environmental sensing. The purpose of this thesis is to compare and contrast my research and development experiences (chapters 3 to 6) with the broader literature. In doing this, I included a systematic methodology, however not fully formalized, to collect data about the framing and implementation details of systems that span the two domains. Given the findings of this review, I believe that such a corpus, in an expanded and more comprehensive form, would be useful to engineers seeking to develop wireless embedded sensing systems across any applied domain. However, this remains future work.

Because this is a scoping review, I acknowledge that this search is not extensive. I also found that the domain is quite difficult to search, especially when targeting multidisciplinary

⁷<https://open-earable.teco.edu/>

applications. Systems enabling research in other fields, for example, may not be clearly exposed in titles, abstracts, or metadata and spread widely across different venues and databases. More work is needed for a comprehensive study of embedded sensing systems; here, I focus on contributions presented at computing venues, specifically those published by ACM.

The corpus is limited in a few fundamental ways that limit its classification as a systematic review. First, deductive codes were created, and coding was completed, by a single author, which undoubtedly introduces bias. Additionally, I did not explore works outside of the ACM and had a limited temporal scope. When conducting initial searches, I found it difficult to find implementations in the database that match the definition of a *wireless embedded sensing system* across domains. Terms like *IoT* and *wearable* help, but unfortunately, text-based searches return a massive amount of results, making review laborious. Scanning full journals and proceedings, for example IMWUT, may be the best solution in hindsight. Finally, I did not use any snowball sampling methods in the work, which can be helpful for understanding historical derivatives and for linking works by the same authors or research labs. Understanding if a device was an iteration of previous work is useful information from a development perspective, but difficult to parse in a review. The transition from chapter 5 [1] to chapter 6 [21] is an example of this.

To make this review a contribution that stands alone outside the context of this thesis, a more systematic approach is needed, including additional collaborators for cross-coding, additional publication sources, and a wider domain search. Publishing such work in a searchable repository, a frontend application for the database I have supplied, could benefit the greater community as a reference for *how* systems are built, potentially seeding the hypothetical tooling I propose in chapter 7.

CHAPTER 3

The Cost of Iteration in Wireless and Mobile Health: An Exploration of FaceBit

This chapter introduces and reflects on the development and evaluation of FaceBit, an unobtrusive device integrated into a face mask that passively measures health metrics and the mask’s functionality. I present this work as a case study highlighting (1) the value of new modalities of wireless health sensing, and (2) the difficulties of development iteration when dealing with novel sensing. This chapter contains rewritten excerpts from *Facebit: Smart Face Masks Platform*[116], in which I am a co-author¹. I interweave FaceBits work with *retrospective* sections in which I discuss points related to my thesis, highlighting the challenges faced during the development.

This work was authored with equal contribution. I present the work in full here, but it is essential to clarify the division of labor and to indicate that the focus will be on my own areas of expertise. In this work, I led the clinical needfinding study and developed a mobile application that encompassed data collection and organization for analysis. The equal contributing co-author, a trained electrical engineer, led the design of the printed circuit board (PCB), firmware development, and development of the FaceBit heart rate algorithm. The additional data analysis, including health metrics beyond heart rate, evaluation, and overall system design, was conducted in equal proportions across all components. A few components of the work are omitted for brevity: details of the energy-harvesting capabilities, which were not used in the functional evaluation, the user-burden study, which did not involve

¹©2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
<https://doi.org/10.1145/3494991>

the physical device but rather a stand-in, and a reduced related works section. Other sections throughout are consolidated or reduced. This is presented to support my thesis narrative and to highlight the work’s contributions.

3.1 Introduction

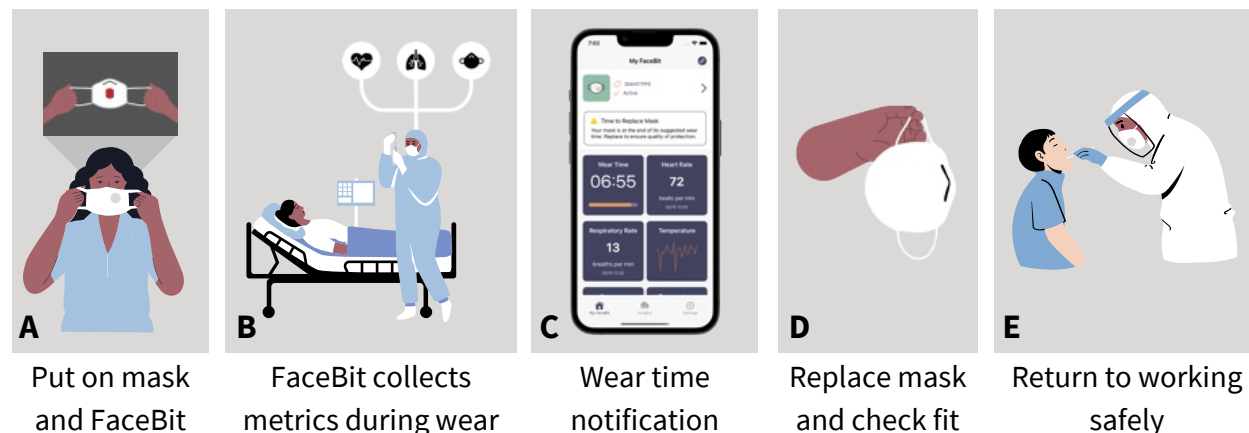


Figure 3.1: FaceBit is an energy-efficient, research-focused hardware platform that enables face masks to be retrofitted (A) to collect on-device health and safety metrics, such as heart rate, respiratory rate, and wear time (B). Notifications are sent to a phone application upon detection of state changes (C). Easy reattachment upon mask replacement (D) enables lower-burden use in difficult work environments (E).

During the COVID-19 pandemic, face masks became ubiquitous worldwide. In August 2021, the U.S. Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) recommended nearly universal masking [117], with many governments enforcing mask mandates. Due to the slow pace of global vaccination, a panel of 175 health experts warned that vaccine-resistant variants are possible, potentially prolonging the pandemic [118]. Beyond vaccination, masking is consistently one of the most effective forms of protection available [119]. Since this work was published, the effectiveness of face masks as personal protective equipment has been confirmed [120], particularly for frontline healthcare workers [121].

Despite their effectiveness, face masks have known limitations and vary in performance. Masks with particulate matter certifications, such as the N95 variant, require regulatory

oversight [122], including fit testing, which poses known logistical and equity challenges for certain populations. Fit testing requires an expensive particulate-counting device and is conducted in a controlled environment that may differ from normal wear conditions. N95 masks are also known to have reduced performance with smaller facial structures, which have been shown to disproportionately affect women, and people of Asian descent [123]. An ill-fitting mask that leaves a person exposed may go undetected for hours or days, increasing the risk of infection [124]. Fitting passive masks with active sensing has the potential to enhance wearer protection and enhance personal health sensing with new modalities.

Active-sensing face masks, "smart" face masks, could provide an early warning when health or environmental metrics indicate danger. These metrics could include respiration rate, a challenging metric in typical wearable configurations [125], air pollution, the presence of a mask leak, or the readiness of a mask for replacement. Such active sensing has been explored in more demanding PPE applications, such as firefighting, construction, and mining [126], as well as in electrical work [127]. However, masks are already burdensome to wear [128]; adding a sensing component should account for this and not introduce additional burden without increased benefit.

In this work, we present FaceBit, a prototype research platform that explores the capabilities of active-sensing face masks, specifically those targeted for frontline healthcare workers. We selected FaceBit’s capabilities, including heart rate, respiratory rate, mask wear, and mask fit, from a preliminary study conducted with clinicians. Our device is evaluated using in a series of lab and free-living experiments. An overview of the system is shown in figure 3.1. FaceBit is released open source under the GPL-3.0 license across three Github repositories²³⁴.

²iOS app: <https://github.com/ka-moamoa/facebit-ios>

³Hardware: <https://github.com/ka-moamoa/facebit-hardware>

⁴Firmware: <https://github.com/ka-moamoa/facebit-firmware>

3.2 Related Work

FaceBit sits at the intersection of energy-efficient low-power embedded systems and mobile health. While there is a wide range of work on wearables that FaceBit draws on, here, for brevity, we present related work on face-centric sensing devices.

Sensing devices worn on the ear, *earables*, are arguably the most widely used method for extracting signals from a wearer’s face and are increasingly available in consumer products [129–131]. Smart glasses have also become popular, but are primarily targeted for augmented and virtual reality, with research-focused exceptions in cardiovascular monitoring [132, 133] and metabolic parameters [134]. NeckSense is a neck-worn device targeted at chewing detection [135]. Specifically regarding face masks, researchers have instrumented them with pressure sensors to estimate wearing comfort [136] and with stickers to detect viruses such as COVID-19 [137].

3.3 System Design

FaceBit was designed as a general-purpose wearable device to explore the potential of active face-mask sensing for both health monitoring and mask function. We began with an initial set of design goals and then identified features through clinical needs-finding.

3.3.1 Design Goals

FaceBit was designed with the following high-level goals in mind:

- **Reliable and accurate health sensing.** The system should provide reliable monitoring of the wearer’s health signals and support expansion and continued exploration of new health signals.

- **Enable real time intervention.** Accurate sensing is critical; however, real-time access to wireless data to support informed decision-making or to deliver just-in-time adaptive intervention (JITAI) [138] is crucial for healthcare workers.
- **Long battery life, low bulk/weight.** Charging and battery replacement are burdensome to the user and require additional donning/doffing of the mask. Therefore, the system must provide multiday battery life, be lightweight, and unobtrusive.
- **Flexible, general purpose, research-ready, and easy to use.** Face masks are ubiquitous passive wearables that, when retrofitted with FaceBit devices, could become essential sensing platforms for future health studies in the medical and UbiComp communities. The system should fit within the mask without interfering with the wearer or affecting the mask, provide general-purpose sensing capabilities, be easily reconfigurable in software and for health metric collection, and include associated tools and infrastructure to enable streamlined data collection and interpretation.

3.3.2 Clinical Needfinding

We opted to prioritize healthcare workers for FaceBit’s functionality. However, we assume that general use by non-healthcare workers would also benefit. We conducted a study of 12 healthcare workers (four medical doctors, two nurse practitioners, and six medical assistants) at a pediatric clinic in Chicago, IL, in January 2021. We conducted a semi-structured interview that included 20 multiple-choice and free-text questions. Section B.1 provides details of the interview.

Clinicians reported rarely wearing masks before the COVID-19 pandemic; 9/12 (75%) reported never having worn one. During the pandemic, all 12 clinicians reported that, while at work, they consistently wore surgical face masks. At the time of the study, N95 shortages were widespread, which may have contributed to the use of surgical masks. They consistently wore surgical face masks at work. We asked each clinician which feature they would find most

helpful in an active-sensing face mask. Common themes included statistics on the mask’s efficiency, such as airflow, the point at which the mask becomes soiled, or the point at which it is ready for discard. Other metrics included exposure to viral infections and vital signs, such as heart and respiratory rates.

Next, we presented a list of 18 potential health metrics, compiled from health guidelines and potential face-oriented sensing modalities. These metrics can be categorized into three interest groups: Mask Information and Safety, Personal Health, and Environmental Factors. Clinicians ranked each metric on a 7-point Likert scale. The top-performing metrics, shown in fig. 3.2, were targeted during the implementation of FaceBit.

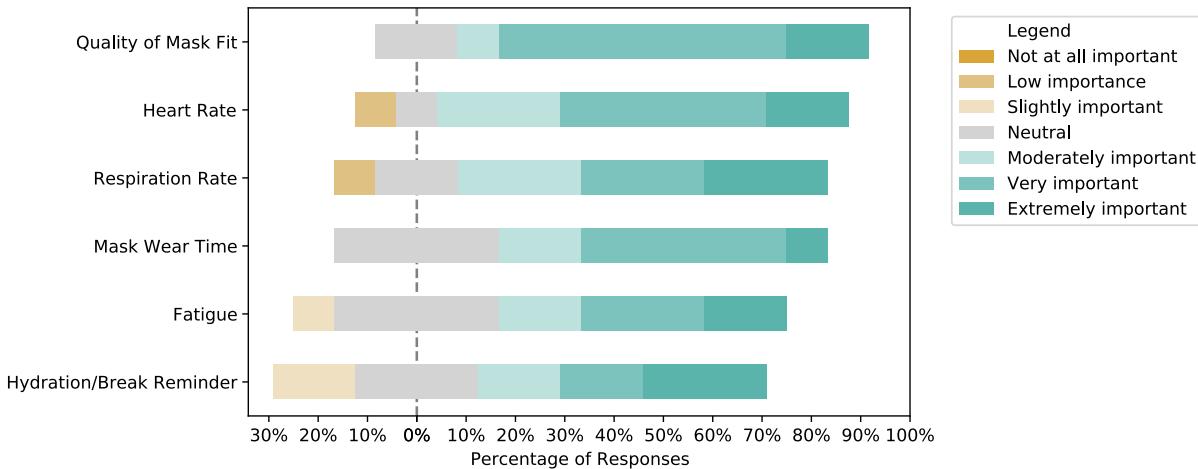


Figure 3.2: 1-7 Likert Scale of importance for metrics of focus. The majority of the metrics presented were deemed important by healthcare workers (to the right of the 0% mark). The length of each color represents the percentage of people who provided the corresponding response.

3.4 System Implementation

In this section, we detail FaceBit as an end-to-end system, including hardware; a custom printed circuit board (PCB) for in-mask attachment; firmware that collects data and computes health metrics; and software, a mobile companion application used for evaluation and as a proof-of-concept user interface.

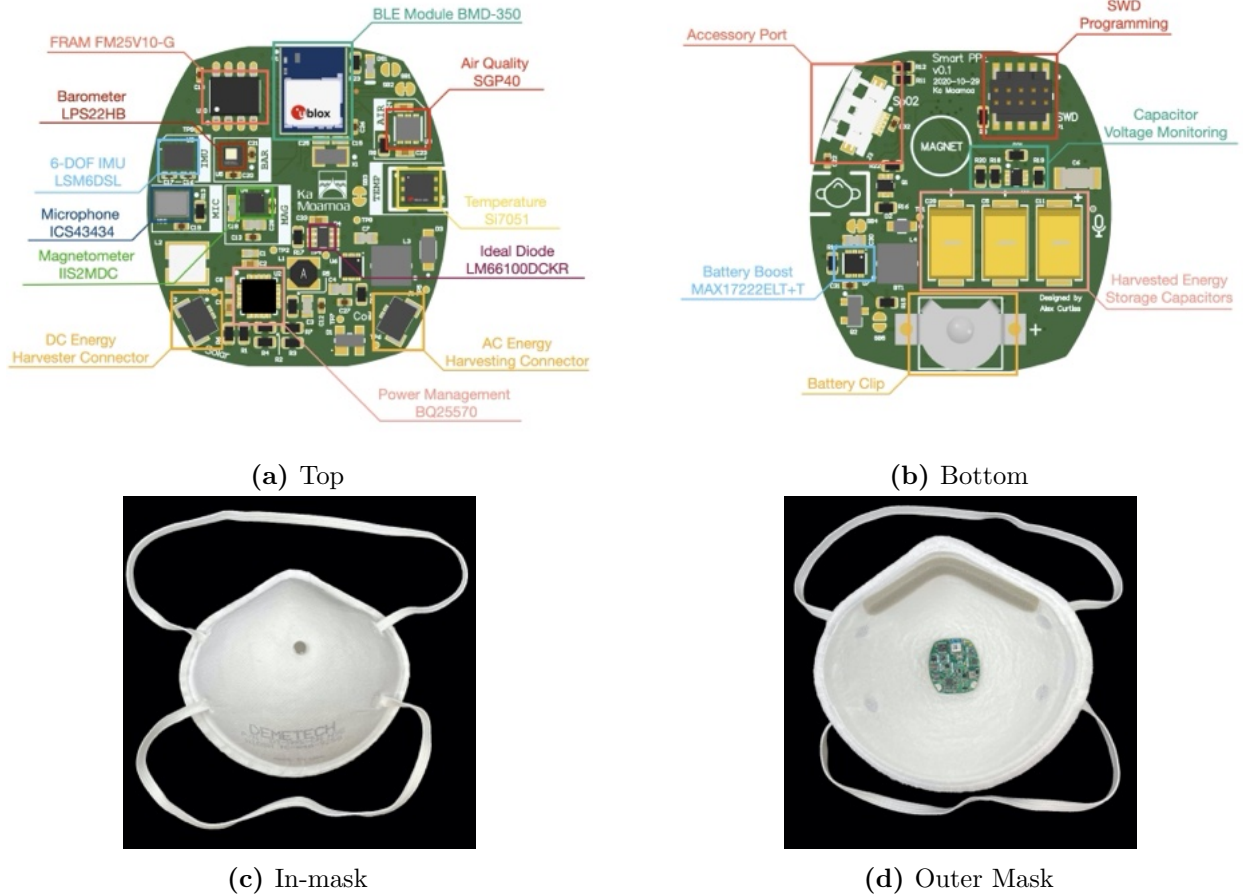


Figure 3.3: The top of the FaceBit board houses the compute, sensing, and most of the energy-harvesting circuitry. The application runs on the BMD-350 Module, which incorporates Nordic’s NRF52832 (512k Flash/64k RAM). The bottom of the FaceBit board houses the energy-storage elements and the programming interface. It also includes an accessory port for adding future modules. FaceBit is shown placed in an N95 mask in (c) via a magnet (d).

3.4.1 Hardware Platform Design

FaceBit is a two-part system. FaceBit comprises a small printed circuit board (PCB) with a microcontroller and sensor suite that attaches to any surgical, N95, or cloth face mask via a magnetic clip. Figure 3.3 shows an annotated render of the device and its placement in an N95 mask with a magnetic attachment.

In alignment with our design goals, FaceBit is small enough to fit comfortably in either an N95 or surgical mask (1.2" wide x 1.3" tall). The board features an nRF52832 SoC from Nordic Semiconductor, which is packaged with a chip antenna on the pre-certified BMD-350

BLE Module. Of the six sensors included on the PCB, only three are used for our application: the LPS22HB barometer, the LSM6DSL 6-DOF Inertial Measurement Unit (IMU), and the Si7051 temperature sensor.

FaceBit employs a flexible power architecture that supports both battery power and energy harvesting. Functional evaluation is powered by a 105 mWh primary cell via a surface-mounted battery holder. Both power sources are managed by Texas Instruments’ BQ25570 power-management integrated circuit (IC), enabling intelligent switching between them.

3.4.2 Runtime System

Unlike other works [135, 139], FaceBit’s health metrics are computed in real time on the device itself, requiring careful attention to energy consumption. This starts with our energy-efficient runtime, which eliminates the need to turn FaceBit on or off each time a mask is manually donned or doffed. While FaceBit is worn, tasks (heart rate and respiratory rate sensing and BLE broadcasting) are scheduled via a timer. The sampling rate can be adjusted to fit the application’s needs.

We focus on sensing health metrics that are technically feasible and of interest to clinicians as identified by our needfinding study. The targeted health metrics are wear time, heart rate, respiratory rate, and mask fit.

Wear Time

Wear time is determined by whether the mask is on or off. This is computed as the difference in the pressure signal over a short time window containing at least one respiratory cycle. Our key observation is that the standard deviation of a pressure signal with the mask on is substantially higher than with the mask off, owing to respiration-induced pressure fluctuations. This can be further simplified by retaining only the minimum and maximum pressure values

within a time window. This difference, in combination with a minimum threshold, enables clear visualization of state changes in the mask.

To measure the wear state, the device starts in an *idle mode*, periodically waking from sleep to check whether it is being worn. Because this task is performed every minute, it must be low-power. With this in mind, the board uses only the barometer and reads its data at 1 Hz, sleeping between measurements. It records 10 values over 10 seconds and stores the minimum and maximum. If the difference between these two exceeds 0.1 mbar, the mask is assumed to be worn. Because spurious pressure spikes can occur upon mask removal, we perform this check before each health measurement task. This wear status is aggregated to calculate a running wear time; however, to determine the wear time of a particular mask, user interaction is required to track disposal via the companion application.

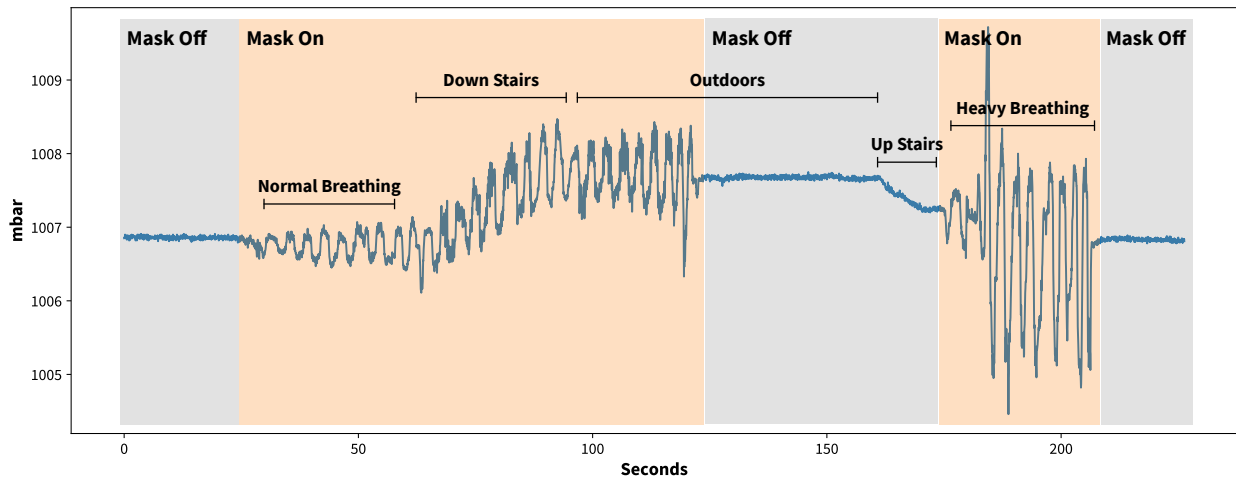


Figure 3.4: Example air pressure signal, with mask on/off annotations, collected from FaceBit in an N95 mask in a field test.

Heart Rate

Heart rate is an important physiological parameter. An elevated resting heart rate is associated with increased incidence of cardiovascular disease in both men and women [140]. Heart rate is also rated highly in our survey of healthcare workers. Unfortunately, traditional methods of heart rate detection (e.g., electrocardiogram or photoplethysmography) require continuous,

immobile skin contact, which is unlikely with an easy-to-don-and-doff mask. Therefore, we explored non-contact methods for calculating heart rate.

Ballistocardiography (BCG) is a method for assessing cardiac health whose origins date back to the turn of the 20th century [141] and were significantly refined by Starr et al. in 1939 [142]. Historical methods relied on complex mechanical setups but exploited the same basic phenomenon used in our FaceBit heart rate detection algorithm: with each heartbeat, blood is forcefully ejected from the left ventricle into the major arteries. This causes a subtle whole-body recoil that can be detected mechanically. FaceBit uses an inertial measurement unit (IMU), specifically the LSM6DSL, to sense this recoil. Each ballistocardiographic pulse consists of several high-frequency waves, which are hypothesized to arise from interactions between regions of higher and lower pressure in the ascending and descending aorta during each heartbeat [143]. The intervals and amplitudes of these individual waves in the BCG complex are thought to correlate with various cardiovascular health markers [144]. FaceBit employs a preprocessing method that leverages this characteristic structure to identify heartbeats from raw 3-axis gyroscope data. Figure 3.6a shows the recovered BCG heartbeat signal obtained from a seated participant wearing an N95 mask fitted with FaceBit, along with an ECG signal recorded using a Polar H10 chest strap.

FaceBit’s pre-processing methodology is adapted from Hernandez et al. [145]; however, we implement the algorithm entirely on the nRF52832 in C++ to run in real-time. We collect 3-axis gyroscopic data at 52 Hz, then apply a fourth-order infinite-impulse-response (IIR) band-pass filter with cutoff frequencies of 10 and 13 Hz. The passband removes the DC component and low-frequency motion artifacts (e.g., respiration), but retains the high-frequency pressure waves associated with the BCG complex, namely the I, J, K, and L waves [143]. The gyroscopic signals are then consolidated using the Euclidean norm (l^2 -norm). This normalization allows the board to be placed in any orientation, aggregating all motion. Next, the l^2 norm is fed through a smoothing, second-order IIR band-pass filter with cutoff

frequencies of 0.75 ad 2.5 Hz, corresponding to a range of 45 - 150 beats per minute (bpm). The result is the recovered BCG-HR signal in mdeg/sec.

Next, we introduce the FaceBeat algorithm, which extracts heart rates in real time from the recovered signal. This is necessary because the signal is easily corrupted by motion artifacts (e.g., walking, head movement). Moreover, the heavy filtering stage often produces signals that resemble valid heart rates even when none are present. The FaceBeat algorithm addresses these issues to identify true heart rate signals. In free-living scenarios, FaceBeat detects true signals during moments of rest, providing periodic heart rate readings without body contact. To accomplish this, we first detect and store the timestamps of descending zero-crosses in the recovered signal (mdeg/sec). When five consecutive timestamps are collected, the instantaneous heart rate is calculated by 1) measuring the consecutive elements, yielding $\frac{[seconds]}{[beat]}$ 2) taking the reciprocal of this number to obtain $\frac{[beats]}{[second]}$, and 3) multiplying the reciprocal by 60, thereby arriving at $\frac{[beats]}{[minute]}$. We later added a fourth step to the filter process because BCG waveforms are typically less than 100 mdeg/sec in amplitude; therefore, we did not count peaks >150 mdeg/sec as candidates. This results in four instantaneous heart rates from the previous five suspected beat signals.

To assess validity, we rely on the inherent physiological predictability of heart rate and interbeat interval (similar to the Pan-Tompkins algorithm for QRS detection [146]) to evaluate and calculate the standard deviation of the interbeat intervals, and if they fall below a certain threshold, we record their average as the valid heart rate. This threshold was a parameter we manually optimized during evaluation; we initially set it to 20 beats/min, then lowered it to 7 beats/min. Both the BCG amplitude filtering parameters and the inter-beat threshold remain open to refinement, requiring further investigation or real-time calibration to optimize across more scenarios and physiological variations.

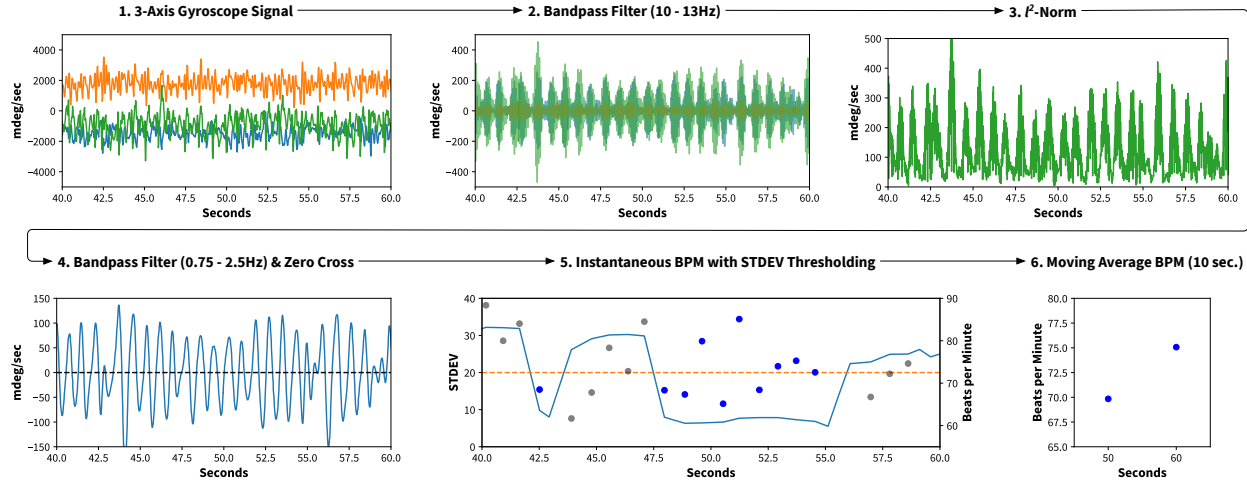


Figure 3.5: Heart rate signal-processing pipeline. 3-axis gyroscope data is filtered to extract the instantaneous heart rate from the time between successive zero crossings of the signal. A standard deviation threshold is used to determine a realistic change in rate from beat to beat. Finally, heart rate is calculated as a 10-second average of valid instantaneous rates.

Respiration Rate

Face masks provide direct access to the wearer's respiratory system, thereby enabling a low-burden (no additional burden added to the face mask) measurement of Respiration Rate. In an N95 or equivalent mask with some sealing capability, normal breathing results in a pressure drop across the filter material, which is easily detected. However, loose-fitting surgical or cloth masks, which were popular during the COVID-19 pandemic, especially among the general population, do not generate a strong pressure signal due to a lack of seal with the face. Fortunately, breathing also produces pronounced temperature fluctuations with a distinct periodicity that is highly conserved across mask types. The thermal mass of the FaceBit board, combined with the insulation properties of still air in the mask, amplifies the effect of moving air in the mask, resulting in strong breath signals in temperature readings. For this reason, we chose temperature as the measurement to derive respiration rate. However, there may be greater benefits to switching methods based on the mask or to multimodal sensing, depending on the scenario (e.g., when ambient temperatures vary and match or

exceed exhalation temperatures [147]). Figure 3.6b compares temperature and pressure signals in the mask.

The respiration rate signal processing pipeline begins by collecting temperature readings from the Si7051 temperature sensor at 10 Hz. Next, the signal is filtered with a second-order band-pass filter with cutoff frequencies of 0.067 to 1 Hz and 4 to 60 breaths per min (BrPM). Next, the timestamps of any descending zero-crosses are buffered. At the end of the sampling window (30 seconds, primarily chosen for energy consumption), the respiration rate is calculated from the average zero-cross intervals. Any interval faster than 60 BrPM or slower than 4 BrPM was removed, likely caused by speech or other disordered breathing. Additionally, we utilized a temperature threshold (0.04 °C) to reduce the classification of filtering artifacts.

3.4.3 Mask Fit Exploration

Mask fit was identified as the most important metric in our needfinding study. However, it is well known that mask fit is a problem in practice, where physiological differences such as facial hair, face size, and even gender play a role [148]. In the United States, regulated methods for mask fit are required annually in certain professions, necessitating expensive, bulky equipment that may not capture daily variations [149]. Therefore, real-time mask fit is a difficult metric to quantify and requires careful validation. For example, a single false positive in mask fit poses a significant threat to the wearer, unlike a single heart rate or respiration rate reading. Therefore, FaceBit’s mask fit is presented as an exploration study, rather than a real-time methodology. Further work is required to substantiate the real-world practice of a mask fit score. Additionally, mask fit is only measurable with regulated masks that have seals, e.g., N95, N99, etc. Unless otherwise mentioned, all our mask fit validations used N95 masks.

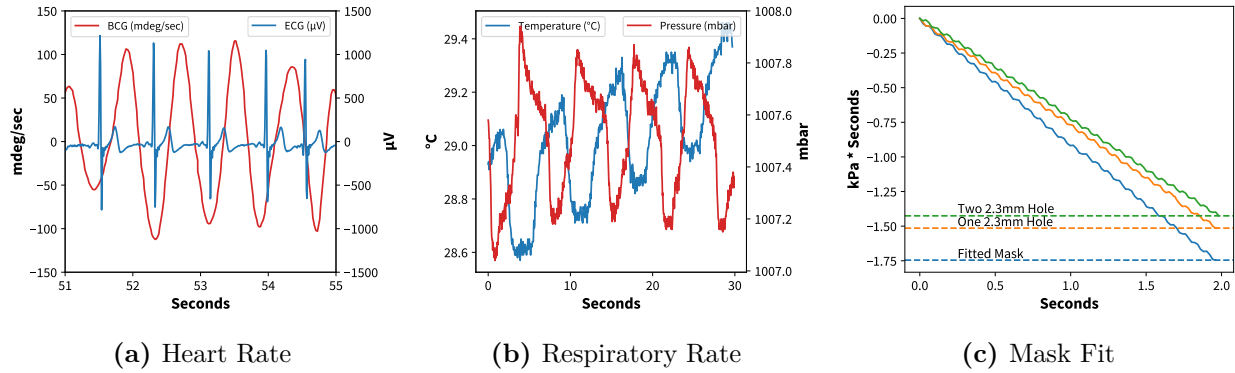


Figure 3.6: Metric signal existence plots. **(a)** ECG ground truth signal collected from a Polar H10 chest strap compared to a Ballistocardiography (BCG) signal extracted from the FaceBit’s gyroscope. **(b)** Respiratory rate signal existence: Raw temperature and pressure signal from the FaceBit sensor board. **(c)** Mask fit exploration. Cumulative integration of the pressure signal sensed inside an N95 mask while under suction from a 12V vacuum pump. The highlighted minimum values represent a measure of mask resistance to airflow.

We verify the signal using a vacuum pump attached to an added hole in the mask. These initial controlled experiences showed a promising correlation between leakage and the integral of the pressure signal while a mask was under suction via the vacuum pump (Figure 3.6c). A well-fitting mask offers greater airflow resistance, so the slope of the integral can serve as a straightforward indicator of mask fit. However, measuring this signal in the absence of a constant (e.g., the vacuum pump) poses a challenge. Therefore, we propose a method that uses the user’s lung capacity to maintain a constant air volume through the mask. This method, if deemed viable, would not provide a real-time fit metric but would allow the user to achieve a fit without additional equipment by performing a breathing exercise.

To perform this exercise, wearers exhale as much air as they can from their lungs, reaching their residual volume, and then hold their breath for five seconds. This breath-hold allows equilibrium within the mask to be recorded as a baseline. The wearer then inhales as deeply as possible at a normal rate until their lungs are as full as possible, then holds their breath for five seconds to obtain another equilibrium baseline reading. This exercise is known as an inspiratory vital capacity (IVC) test, in which vital capacity is the static lung volume

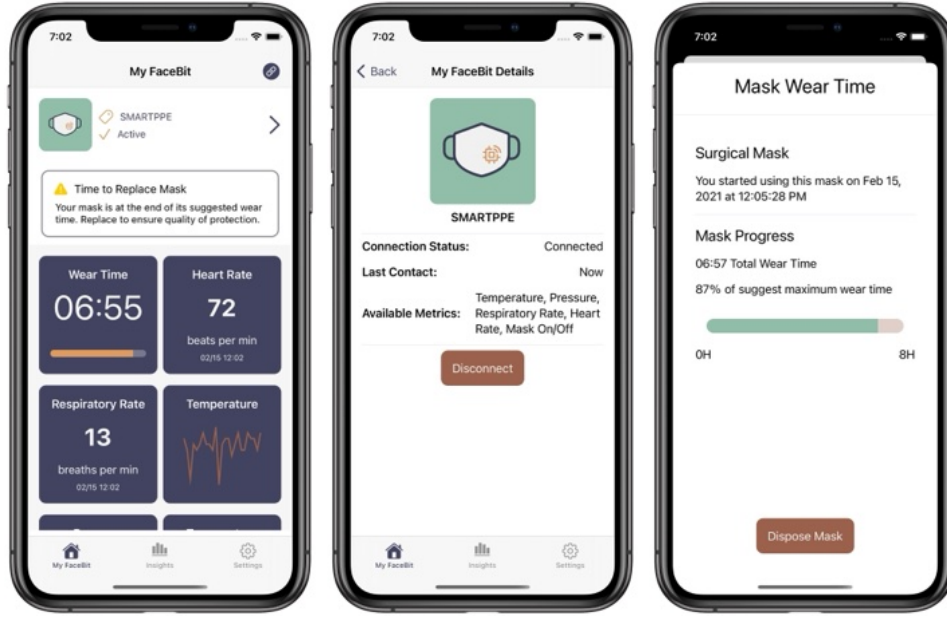
that changes only slowly with age (though it can be affected in the short term by posture or disease).

During the IVC test, pressure is collected from the LPS22HB at 50 Hz. Before starting the test, an ambient pressure (atmospheric pressure) is collected and subtracted from the test signal (normalization). The area under the curve during the IVC test [Pa*s] is proportional to the total volume of air inhaled with mask resistance [$kg^{-1}meter^4second$] as the proportionality constant. Because we do not know the total volume of inhaled air, which is necessary to compute mask resistance, we ask the wearer to calibrate the device by pressing the mask against their face to ensure a proper seal. This calibration is needed only once per wearer and per mask type. Future mask fit scores are calculated as a percentage of press-fit calibration.

The Occupational Safety and Health Administration (OSHA), requires a fit factor greater than 100 (less than 1% total inward leakage) during the quantitative fit testing procedure. Rengasmy et al. studied the effect of leaf size on particle penetration in N95 masks, and found that a mask with two leaks of 3mm diameter each allowed a total inward leakage (TIL) of $10.67\% \pm 4.6\%$ (ratio of particle concentration inside the mask to outside the mask) [150]. A mask with no leaks allows $0.31\% \pm 0.4\%$. Therefore, our goal is to detect leaks smaller than a 3-mm-diameter hole; however, the minimum allowable leak size remains unclear in the literature.

3.4.4 Mobile Companion Application

The FaceBit Companion Application is an iOS and macOS application that serves as a research tool and prototype user interface for a Smart PPR platform. All evaluations utilize the FaceBit Companion Application for real-time data collection. The application is developed for iOS and made available on macOS through Mac Catalyst. The application communicates with FaceBit via Bluetooth Low Energy (BLE) using a custom generic attribute (GATT) server. The application supports raw time-series data collection for debugging, as well as



(a) Home Screen (b) FaceBit Device Details (c) Mask Wear Screen

Figure 3.7: FaceBit Companion Application on iOS (a): The FaceBit Phone Application homepage displaying general details, current wear time, respiration rate, heart rate, and temperature data from the FaceBit Sensor Board. (b): The mask wear time interface for tracking mask disposal. (c): FaceBit Sensor Board detail screen outlining current connection status

periodic computed metrics for general-use deployments. All data is stored locally in a SQLite database.

3.5 Evaluation

We conducted a series of evaluations to validate FaceBit’s functionality. The goal of our evaluation is to measure FaceBit’s capabilities, test its robustness and performance across diverse situations, identify areas for improvement, and identify performance factors. Specifically, we performed experiments covering in-lab validation of health and performance metrics, performance in noisy free-living scenarios, the impact of mask location and type, and the impact of physiological differences on signal quality.

Table 3.1: Power and energy consumption of the FaceBit application runtime.

	Avg. Power [μ W]	Duration [s]	Energy [mJ]
Idle/Sleep	48	–	–
Mask On/Off Detection	678	10	6.8
Respiratory Rate Measurement	1,300.5	30	38.9
BCG Measurement	7,320	10	73.1
BLE Broadcast (Typical)	12,360	3.7	45.4

3.5.1 Performance Benchmarking

For FaceBit to operate reasonably without placing an undue burden on the wearer, the device must be power-efficient, ideally lasting multiple mask replacements (e.g., multiple days). We performed benchtop validation to estimate the battery’s duration. For this validation, we assumed 8 hrs/day of wear and a measurement duty cycle of 5 minutes (measurement rate). Mask on/off is checked every two minutes, and one BLE broadcast (a significant power draw) every five minutes. Table 3.1 details the power at different firmware states. We set the total battery capacity to 75% (75.6 μ Wh, 272 mJ) to account for converter efficiency and temperature variations. Using these estimates, the device can last 11 days. We believe there is further room for power optimization and assume that an 11-day battery lifetime is reasonable for user burden; evaluation of burden remains future work.

Controlled environment validation of Health Metrics

We conducted a controlled environment validation study to assess the accuracy of the heart and respiratory rate metrics calculated in real time on the device ($n = 8$). These metrics were evaluated in unison, with FaceBit wirelessly transmitting both metrics to the companion application during a set of activities performed in sequence by each participant. The evaluation included nine graduate and undergraduate students (4 male), however one participant was excluded due to a ground-truth device error.

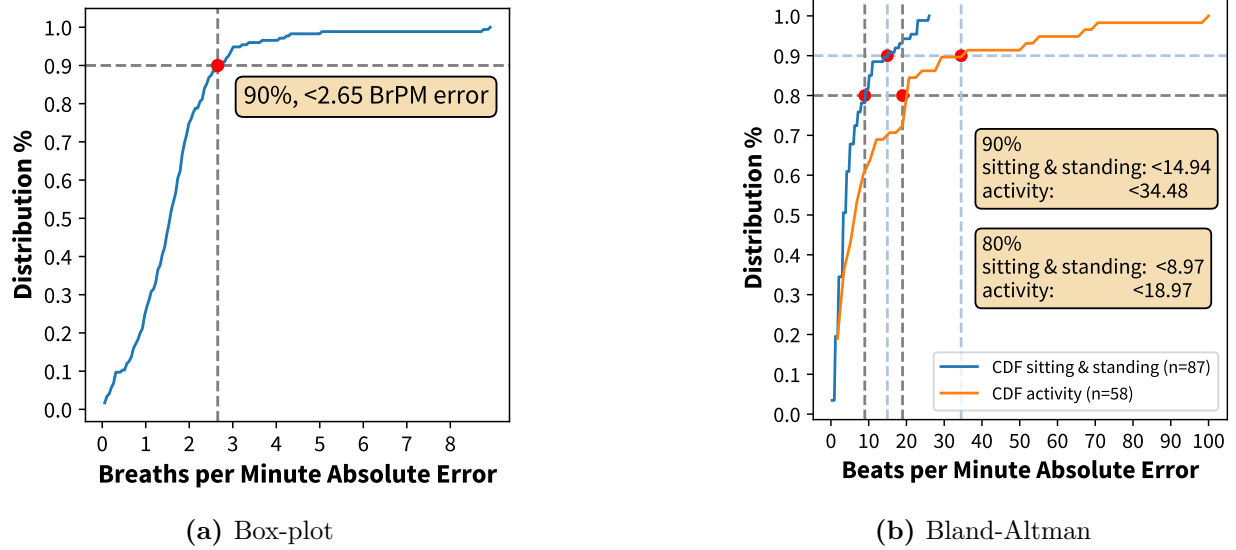


Figure 3.8: Health metric evaluation results (a) Respiration rate (186 samples, $n = 8$): Cumulative distribution function over the evaluation compared to ground truth respiration belt. (b) Heart Rate (149 samples, $n = 8$): Cumulative distribution function over the evaluation compared to ground truth ECG chest strap.

All participants wore an N95 respiratory (Demetech, cup-style) with an attached FaceBit device running the application firmware. Ground truth participants also wore a Polar heart rate monitor chest strap (H10), and a NeuLog respiration belt with a USB data logger (NUL-236, USB-200). Each participant was evaluated for 25-minutes, excluding setup and transition times. Participants performed a three-phase structure of activities: 10 minutes of sitting, 2 rounds of 1-minute squats followed by 4 minutes of sitting, and 5 minutes of standing. While seated and standing, participants watched a documentary playing on a laptop. Evaluation with six participants took place outside on a University campus, and three participants in a lab. The evaluation followed a strict protocol for consistency and sanitation, including adherence to COVID-19 guidelines.

The respiration rate (RR) evaluation yielded 186 samples across 8 participants. FaceBit, on average, reported RR once every 55 seconds with a mean difference from the ground truth of 1.06 BrPM. FaceBit recorded 90% of recorded samples within ± 2.65 BrPM (see figure 3.8a). Note that manual calculation of RR falls within 1 BrPM error by definition, counting the

number of chest raises over one minute. The NueLog ground truth device is a cumbersome air bladder that is fitted around the participant’s abdomen, recording raw pressure changes in the bladder as arbitrary units. RR ground truth was calculated using peak finding after smoothing the signal using a Savitzky-Golay filter with a window of 5 seconds and a minimal peak distance of 1 second. From this signal, an instantaneous RR was produced from the peak distance and averaged to compare with FaceBit’s 30-second average.

After reviewing the data, we found some anomalies in the ground truth data that may account for our inaccuracies, including peaks in the signal that are not breaths caused by events such as throat clearing, coughing, talking (which we tried to limit), or shifting weight. Additionally, the NeuLog software limited data collection to 5 Hz, which could cause shifted peaks, especially in higher-frequency breathing. Finally, these results identified a firmware bug that caused some respiratory rate calculations to overflow on the sensor board. Most occurrences were easily identified in post-processing, as they fell below our minimum rate (4 BrPM), some outliers remain in the resulting data as they cannot be completely ruled out.

Heart rate (HR) evaluation produced 149 samples over 8 participants. The ground truth Polar H10 chest straps were fitted around each participant’s sternum and adjusted to the recommended tightness. The Polar reported heart rate every second and streamed ECG data at 130.14 Hz. FaceBit attempted to measure heart rate once a minute. FaceBit reported a null value (unable to calculate heart rate) for 14 or 163 total sampling periods. Participants 3 and 8 accounted for the majority of these null reports, indicating a potential need for individual calibration.

Over the course of the evaluation, the mean error between FaceBit and the Polar chest strap was -5.87 bpm with a standard deviation of 14.91 bpm. As expected, the heart rates reported during activities of low movement (sitting and standing) were more accurate than periods of high activity (squats). 68% of measurements during sitting and standing were within 5 bpm of the ground truth, whereas only 43% during squats showed an increased error rate

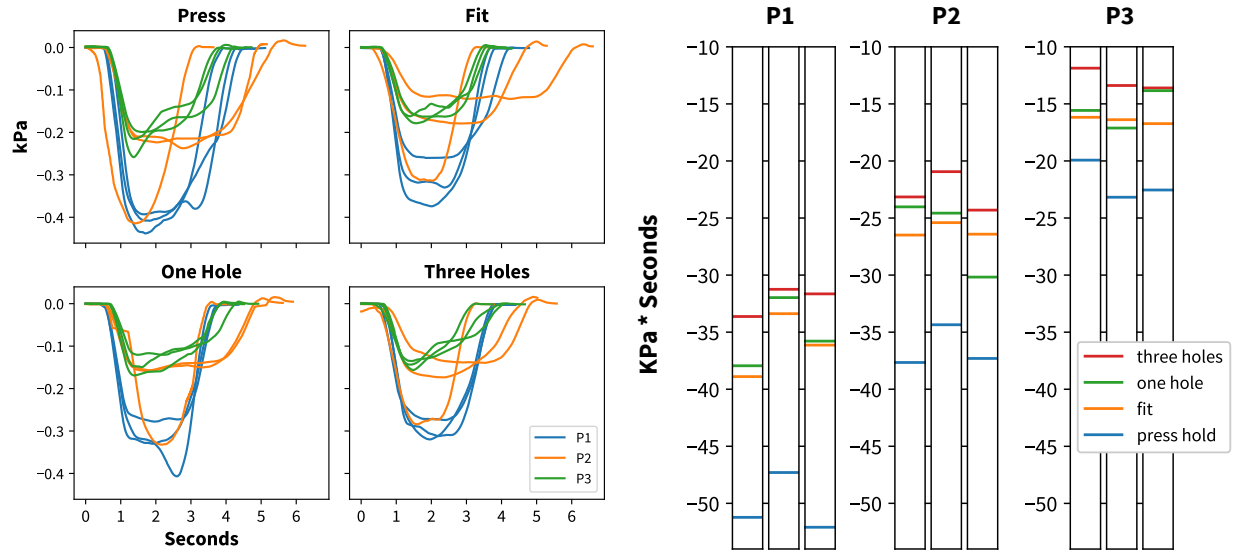
with higher heart rates. The accuracy is represented in figure 3.8b. These results indicate a need for enhanced rejection of motion artifacts to account for some extreme outliers. Some errors may be attributed to internal smoothing in the Polar chest strap, because its heart rate calculation algorithm is proprietary.

Even after removing outliers from the analysis, FaceBit consistently reports heart rates lower than the ground truth (-3.71 bpm, removing data outside the ± 1.96 standard deviation range). An observation from developmental testing of our algorithm was that individual heartbeat detection was more successful during exhalation. Respiration rate is known to affect hemodynamics: inhalation decreases arterial pressure and increases heart rate through a series of physiological changes [151]. This drop in arterial pressure during inhalation may reduce the amplitude of the BCG waveform enough to prevent detection, which explains the bias towards lower heart rates. More experimentation is needed to test this hypothesis.

Mask Leak Detection

We conducted two feasibility studies towards a method for measuring mask fit using FaceBit. These tests use raw pressure data streamed from a debugging version of the firmware. Both studies used an N95 mask modified with three plastic bulkhead fittings (2.3 mm diameter, 4.2 mm²) attached with a short length silicone tubing with a removable end-cap to simulate leaks. During each experiment, participants performed a self-guided IVC test: they exhaled to their lung capacity, held their breath for 5 seconds, inhaled to their total lung capacity, and then held their breath again for 5 seconds.

Our first study was exploratory ($n = 3$) to gather initial data to tweak our methods and assess signal outside our development team. Participants were given a modified mask equipped with FaceBit and asked to perform three mask fit configurations: self-mask fit, where users donned the mask following the CDC seal-check guidelines; mask press, where participants cupped their hands over the mask and pressed it to their face; one-hole open; and three-hole-open. We



(a) Inspiratory Vital Capacity Test on FaceBit (b) Integral Values of Vital Capacity Test on FaceBit

Figure 3.9: Results from initial mask fit exploratory study capture using FaceBit. (a) Signal from inspiratory vital capacity test recorded on-board FaceBit showing filtered pressure signals for each participant ($n=3$) grouped by fit. (b) Integral values from initial evaluation for participant (3) and each experiment (3) compared to each leak configuration.

conducted three trials per participant, randomizing the order of mask configurations in each trial. We analyzed the wirelessly collected FaceBit signals and computed the integral of the inspiratory vital capacity signal, measuring the area under the curve for each configuration. Each participant's mask-press value was larger than that for other configurations, and 7 out of 9 (77.7%) showed larger values with self-fit than with a single-hole mask. The area under the curve was smallest in the three-hole configuration in all trials. These results, while not comprehensive enough to be confounding, proved enough to continue the methodology.

We performed a follow-up study ($n = 10$) with a similar setup, adding a commercial-off-the-shelf analog differential pressure sensor module to measure external pressure. An additional hole was made in the mask, which connected to the differential pressure sensor via a silicon tube. Using the differential pressure sensor gave the advantage of automatically subtracting atmospheric pressure from the measurement. Participants performed the same test over five mask configurations: mask fit, mask press, one, two, and three holes. Three trials were

conducted per participant, with the order of mask configurations randomized. Between each trail the participant donned and doffed the mask. Before the experiment, all participants were presented with a CDC video on donning and doffing an N95 mask and performing a user seal check.

We perform statistical analysis of the participants’ results. We first combined each participant’s trials, averaging the IVC integral for each mask configuration. Then, we performed a repeated-measures analysis of variance (ANOVA) test across the five mask configurations. The data violated the sphericity assumption; the Greenhouse-Geisser correction was applied. This test shows a significant difference between mask configurations ($p = 6.57e - 5$). We continued with a T-test for each pair of mask configurations. We found that the differences in configurations with one, two, and three holes were not significant compared to each other, suggesting a threshold leak size in our methodology. Our results are visualized in Figure 3.10. Our results suggest our method is valid for detecting mask leakage, but more trials are needed to determine the minimum leak size we can confidently detect. Additionally, we require more testing to show the feasibility of leak detection with FaceBit alone. Nonetheless, these results are promising, as they show that pressure used in conjunction with an easily performed lung capacity exercise may pave the way for in-situ mask fit.

Free-living Study

As an initial test of FaceBit’s robustness and performance in real-world wear, we conducted a small-scale study in which three participants wore N95 masks equipped with FaceBit and performed daily activities. This experiment was performed last and included methodology changes to improve evaluation. First, we used a different respiration rate ground-truth sensor, a Vernier Respiration Belt, because the NeuLog was incapable of maintaining accurate measurements while moving. Second, instead of relying on the calculated heart rate from the Polar chest strap, we calculated heart rate from the sensor’s raw ECG signal using the popular open-source library HeartPy [152]. This removed any potential filtering issues in Polar’s

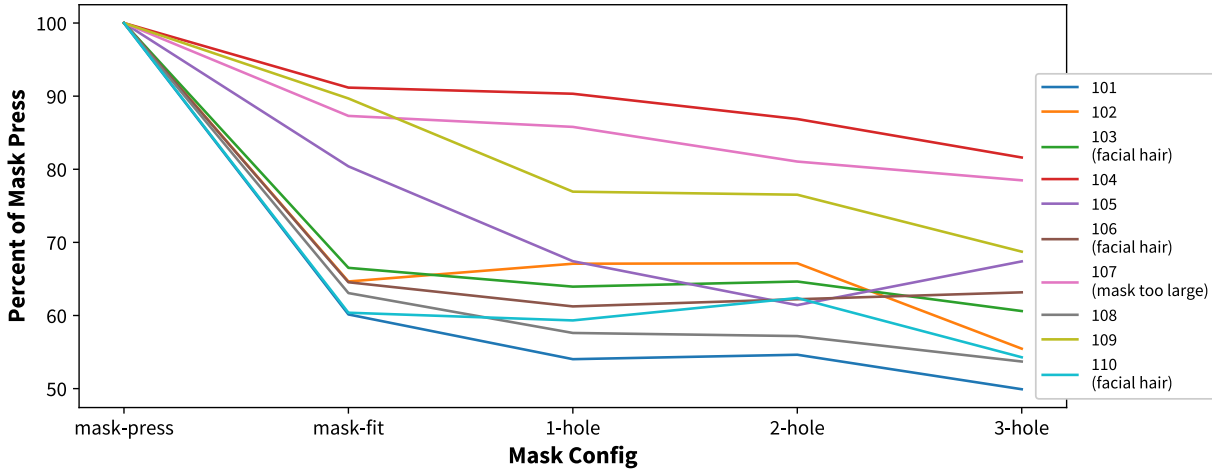


Figure 3.10: Secondary mask fit exploratory study using an external differential pressure sensor. The integral value for five mask configurations was measured during an inspiratory vital capacity test. The percent change relative to the mask-press configuration allows us to compare individual capacities side-by-side. The trend shows that the integral value generally decreases as more leaks are introduced. Physiological differences are noted in the legend.

proprietary algorithm. Third, we modified the FaceBeat algorithm to more aggressively reject time windows with significant motion, thereby reducing the number of samples collected in the hope of obtaining more accurate readings. Finally, we also modified the respiratory rate algorithm to allow measurement of high respiration rates (30-60 BrPM) to better reject oscillations that may result from our filters.

With three participants, we conducted four scenarios. In each, the participant wore FaceBit and was accompanied by a study coordinator who managed the phone application to ensure continuous data collection. The scenarios ranged from 13 to 27 minutes in length and were captured in the summer with heat in some cases exceeding 32 degrees Celsius and high humidity (>70%). The scenarios included walking and picking up lunch, riding on a public train, riding in a car in an urban environment, and working at a desk (captured as a baseline). Each scenario was performed once. One participant performed both "ride in train" and "ride in car" scenarios.

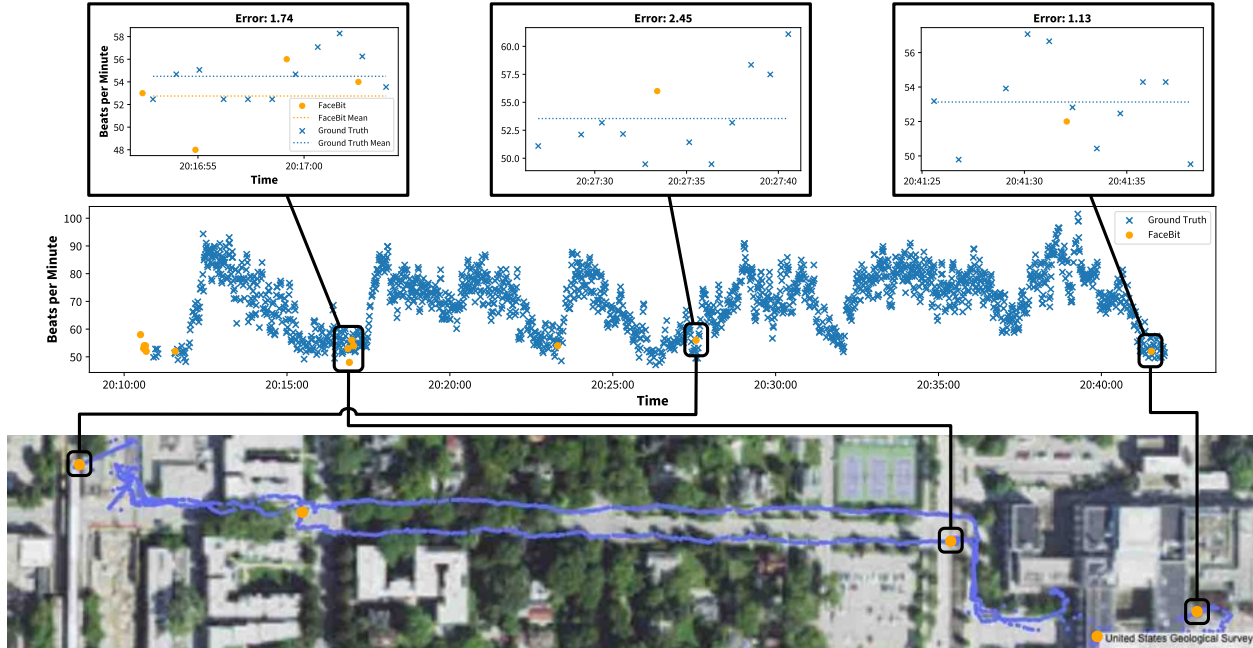


Figure 3.11: The free-living scenario "walk to lunch". FaceBit can capture accurate heart rate even in low-motion conditions despite noisy environments and environmental factors like heat and humidity. **(middle)** shows annotations where heart rate measurements were gathered by FaceBit, plotted over the ground truth heart rate, where each marker is the instantaneous heart rate associated with each beat. **(top)** Zoomed-in data-sections to highlight error. **(bottom)** GPS plot of the walking path with indicators of FaceBit heart rate recordings.

Table 3.2: Root mean squared error for FaceBit predicted heart rate and respiration rate compared to ground truth for all in-wild scenarios.

Scenario	HR (RMSE)	RR (RMSE)	# HR Samp.	# RR Samp.	Avg. HR FB	Avg. RR FB	Duration
At computer	2.52	1.48	49	18	70.22	11.49	00:27:04
Lunch	1.55	2.97	14	26	53.50	14.29	00:27:43
Driving	1.40	3.28	13	19	64.15	18.93	00:20:08
Train	2.9	3.47	2	13	71.50	21.62	00:13:47
Overall	2.09	2.80	78	76	64.84	16.58	01:28:42

Overall, FaceBit produced root mean-squared errors (RMSEs) of 2.09 bpm and 2.8 BrPM across all scenarios. This shows promise in FaceBit's ability to deliver accurate real-time readings and filter out noisy signals. We were particularly encouraged that our improvements to the FaceBeat algorithm resulted in no obvious heart rate outliers, as we saw in our controlled studies. However, there are also areas in need of improvement, mainly in balancing the need to throw out high motion periods for accuracy, and the obvious need to gather heart rate readings at times of high activity. Heart rate was captured during stillness across all

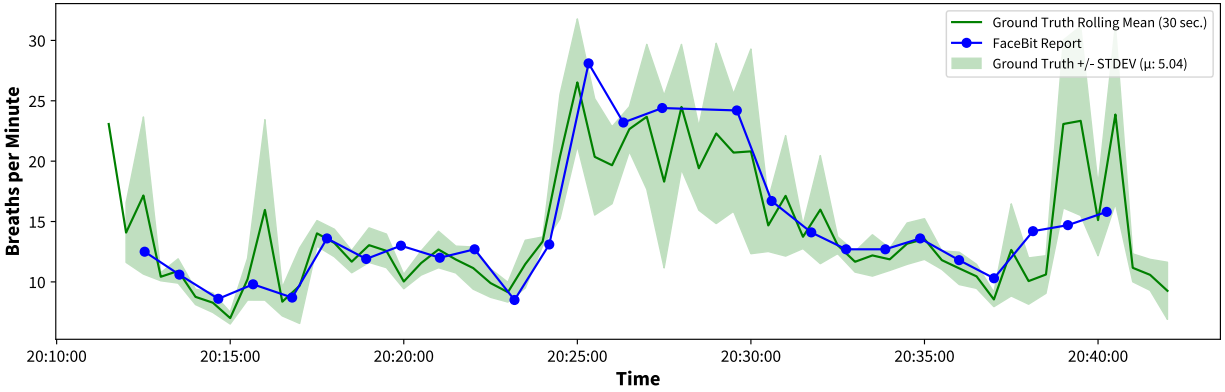


Figure 3.12: FaceBit reported respiration rate plotted over the calculated rolling average of instantaneous respiration rates from the ground truth respiration belt. The data shown is from the "walking to lunch" scenario.

scenarios, with nearly all attempts in the "sitting at desk" scenario successfully reported and only during low motion in others (train stops, stoplights, crosswalks, etc.).

3.6 Retrospective

Here, we briefly explore lessons drawn from development and evaluation of FaceBit to identify challenges and generalizations for similar systems (RQ4). FaceBit is attempting to compute metrics using a new device at a new location on the body. This requires the following development process: first, raw signal data must be collected in situ to characterize variations in face signals; then algorithms must be developed to extract metrics from those signals; and finally, those algorithms must be implemented in embedded firmware and evaluated against ground truths and real-world scenarios. FaceBit developed this project ad hoc to meet the specific needs of the project, but in hindsight, this process is not unique and is required for any new sensing system. While FaceBit's source materials are released as open source, they are not particularly *reusable* without the development team's knowledge or for anything outside the face mask sensing domain; the technical scaffolding we used is embedded in the application and is bespoke (RQ2). In chapter 4 we expand on these lessons to build a system for rapid prototyping of similar wearables without completely new infrastructure.

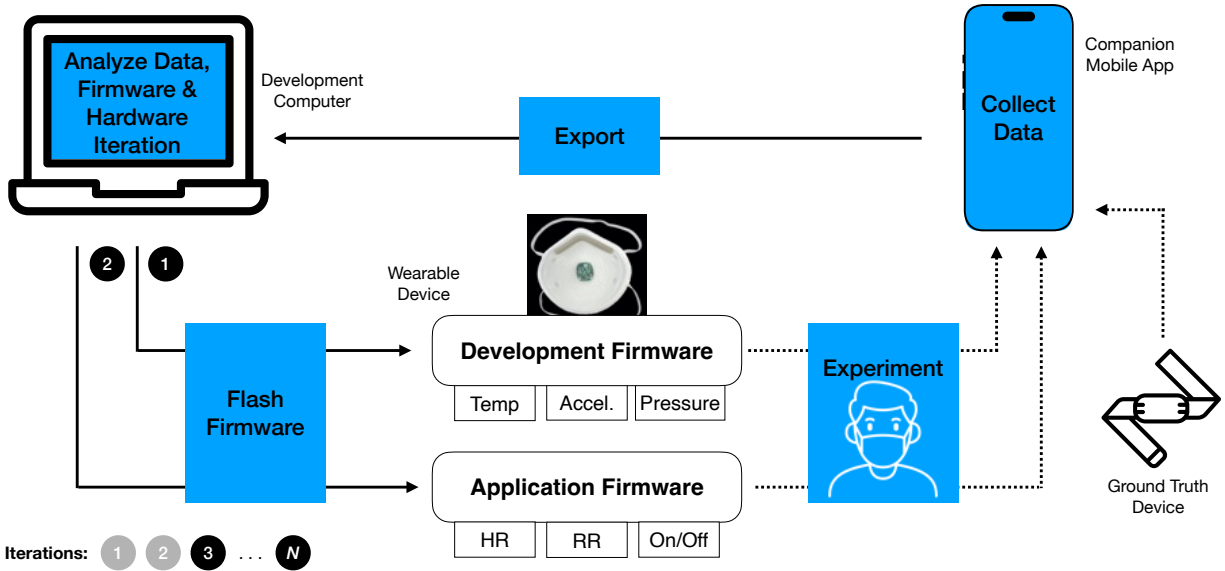


Figure 3.13: The iterative FaceBit development cycle. The process follows two cycles: (1) development firmware is flashed to the device to collect raw sensor data, which is then analyzed to create onboard metric algorithms. (2) Application firmware is written to evaluate the algorithms along with the device function. These cycles are repeated to make progress, but they also pose limitations for evaluation in real-world environments.

3.6.1 Development Cycle

FaceBit was uniquely challenging because, aside from sensor validation testing, we often needed to collect data from participants. There are no good simulations or datasets for inside a face mask. While our algorithms were not developed from scratch, we relied on well-established physiological signal processing; we still required custom validation and collaboration, given the variability across our sensors and unique positioning. We managed this using the development process outlined in fig. 3.13. We maintained two firmware instances: one for real-world *application* and another for high-frequency raw data collection in *development*. The data collected from the development firmware was analyzed and informed the computation on the application firmware over many iterations finally arriving a state we felt we could evaluate. This required flashing to update parameters during experiments limiting our testing.

Both firmware versions relied on Bluetooth Low Energy (BLE) for data collection, with data transmitted wirelessly to the companion application. Wireless data collection is a necessity, as with most wearable devices; testing and validation are extremely limited with wired systems and are only used for proof of concept [153]. FaceBit, given its placement inside a mask, was limited not just by free movement, but also by the mask seal; we opted not to do any validation with wired components, as it would interfere too much with wear. Data was stored locally in SQLite, with the companion application analyzed in Python on personal computers. Ground truth data was collected externally, depending on the device, and analyzed alongside our data when necessary during development.

Challenges and Limitations

The development cycle for FaceBit was fragile, prone to human error and data bias, and slow. Fragility and human error came from bespoke software developed by a small team with a wide scope. Any changes to the firmware required additional changes to the companion app, usually accompanied by debugging with multiple team members. Data bias was easily introduced with self-testing, repetitive actions, and homogeneous environments. Each iteration required flashing the FaceBit board with new firmware, running an experiment wirelessly, then extracting the data, a slow process compared to typical software development life cycles. We attribute these cumbersome limitations to the unfortunate need to build our own research processes and apparatus, which inevitably added little to the overall project.

Reliable data collection was a substantial bottleneck in FaceBit’s development, and at a functionality level, not unique to our application. The data values were unique, but how we collected them with custom BLE implementations was not. Building a system to do this specifically for FaceBit is cumbersome and unnecessary. Chapter 4 aims to address these generalizable components, technical scaffolding, in a reusable system for any wearable development.

3.6.2 FaceBit’s Evaluation

Challenges in evaluation were an extension of the limitations encountered in development (section 3.6) and also introduced unique components, including the integration of external data and the design and implementation of experimental protocols. Once again, much of the infrastructure needed to support evaluation was a one-off implementation; however, it was not unique to the development of FaceBit specifically.

Ground Truth Integration

When measuring heart and respiratory rate, we had to integrate ground-truth data with FaceBit’s data, ensuring accurate time-stamping for beat-by-beat or breath-by-breath comparison. This type of integration is a common challenge, as we will see in later case studies (section 5.4), and it falls into a broader discussion of interoperability. Many commercially off-the-shelf (CoTS) products in the IoT space provide their own systems for user interaction and data collection, offering cross-modality integration only with their own products, a phenomenon known as vendor lock-in [154]. Luckily in the case of FaceBit, Polar, the maker of the chest strap we used for heart rate ground truth, provided an SDK to collect data, including raw data⁵. This is unfortunately a rare instance in consumer products, but makes the Polar H10 chest strap a useful research tool [155] especially with raw collection which can remove bias from internal proprietary algorithms [156]. To collect heart rate data, we developed a separate, generic iOS application that streamed heart rate and ECG data from the chest strap.

The NeuLog and Veriner respiration rate products we used both used separate collection methods via their own distributed applications. The NeuLog respiration belt⁶ required a wired connection and was generally noisy (as discussed above), and therefore was not used in in-the-wild evaluation. Vernier, which had a more accessible mechanical respiration rate

⁵<https://github.com/polarofficial/polar-ble-sdk/>

⁶<https://neulog.com/respiration-monitor-belt/>

monitor that used force measurement via a chest strap⁷ rather than pressure from an air bladder, provided a proprietary desktop and a mobile application. This allowed us to collect data on the fly, but it was still cumbersome to switch among three applications. Given the complexity of data collection, in the free-living evaluation, data management was handled by an accompanying team member to maintain active connections to three sources.

To manage time synchronization across all devices, we added simple functionality to the companion application that timestamps the start of events. This was especially important for the NeuLog device, which only records relative timestamps (e.g., starting at 0 and counting up), so we had to manually start recording on the NeuLog and, at the same time, mark the time in the companion app. Luckily, heart rate and respiratory rate are relatively low-frequency and low-variance events, meaning a slight offset in the signals did not result in large errors.

Interoperability remains a major challenge in wearable development, limiting not only evaluation but also multimodal sensing (e.g., aggregating data from different locations or devices) [154, 157]. Building systems that promote collaborative and open research workflows is crucial for advancing the field of applied embedded sensing systems. We attempted this mentally in FaceBit by releasing our hardware designs and code as open source, but simply releasing is not enough to make true contributions to research toolkits. We need resources to actively engage with the development community, provide clear documentation, and actively seek feedback to improve our tools and make them more accessible to researchers.

3.7 Towards Better Research Apparatus

In reflecting on FaceBit’s work, we can clearly see technical overlaps in the embedded system’s underlying infrastructure, which can be decoupled from FaceBit’s specific function. The technical scaffolding identified here includes raw data collection, wireless communication

⁷<https://www.vernier.com/product/go-direct-respiration-belt/>

protocols, runtime parameterization, time-series processing, and ground-truth integration. Components of this scaffolding are prospects for external generalized tooling that is separate from specific applied applications. Chen et al. mentions in a 2021 interview study regarding human activity recognition devices that data collection is the 'hardest part of [graduate students'] research' [158]. FaceBit reflects this challenge, with its implementation reinventing a large portion of the underlying infrastructure to support its novel functionality, which is unfortunately common practice.

Academia, unfortunately, does not incentivize this type of toolkit development [159, 160]. This leaves each system with bespoke infrastructure, with detrimental technical debt, or what Hassan et al. refers to as "Reproducibility Debt" [161]. It is safe to conclude that FaceBit falls into the large category of abandoned research artifacts [12]. While it was presented as a research platform and released as open source, its physical and digital artifacts, to the best of my knowledge, have not been used in any iterative research. There are many factors beyond releasing code and design that go into implementing functional, impactful research tools. In the remainder of this dissertation, I examine not only the technical characteristics of producing tooling that supports device-agnostic embedded systems research, but also what is required to make such tools accessible and to promote contributions valued by its application domain.

CHAPTER 4

A Declarative and Adaptive Platform for Wireless Personal Health Research

In this chapter, I introduce FlexiBLE, a generalizable platform for data collection and interaction with wireless sensors via Bluetooth Low Energy. The repeated engineering overhead observed in FaceBit (Chapter 3) is a systematic challenge in the early-stage development of wireless sensing systems. FlexiBLE aims to address this challenge by providing a platform to streamline the development workflow. FlexiBLE was originally published as a workshop paper at the ACM International Symposium on Wearable Computers (ISWC) 2022 [162]¹, and has since been used as the underlying data collection and interface for other projects [163, 164]. This chapter presents FlexiBLE’s full lifecycle, from design and development to use across two application domains and practitioner evaluation via semi-structured interviews. FlexiBLE demonstrates the generalization of technical scaffolding (RQ4) and supports reusability by decoupling the novel application from the system infrastructure (RQ2).

4.1 Introduction

Wearable devices that track our everyday lives are becoming commonplace. However, collecting sensor data from wearables is challenging and creates significant engineering overhead [158], even when merely reproducing well-established methods. Many consumer products have broad sensing capabilities, but proprietary hardware and software significantly limit their customization or integration into research pipelines. Producing a system to collect high-

¹©2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
<https://doi.org/10.1145/3544793.3560399>

frequency sensor data in free-living situations to monitor metrics such as new health-related biomarkers is particularly challenging. Doing so requires developing new hardware, testing it from scratch, and building firmware and software to support accurate data collection in the wild. This is a daunting task and can lead to long development runways for new contributions to wearable systems. I argue that this overhead is a contributing factor in systems often failing to progress beyond the prototyping stage.

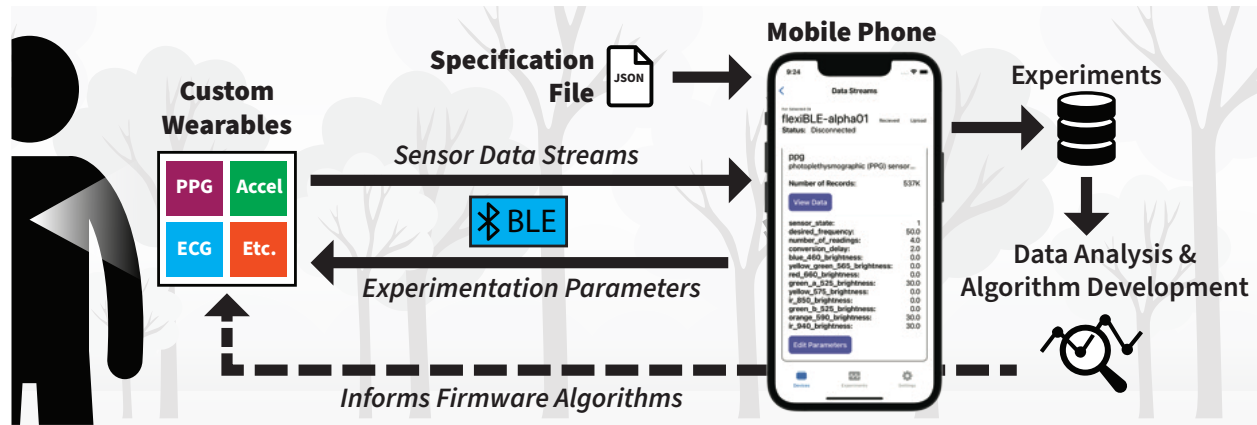


Figure 4.1: Overview of the FlexiBLE system. Custom wearable devices equipped with sensors (e.g., PPG, accelerometer, ECG) communicate with a mobile phone application via Bluetooth Low Energy (BLE). A JSON specification file declaratively describes the device’s data streams and configurable parameters, enabling the application to dynamically generate interfaces without code changes. Sensor data streams flow from the device to the phone, while experimentation parameters can be sent back to the device for runtime reconfiguration. Collected data feeds into experiments for downstream data analysis and algorithm development, which in turn inform the wearable device’s firmware algorithms, creating an iterative development loop.

Novel embedded contributions are often evaluated in simulations, benchtop, or controlled environments to validate their core functionality (section 2.3.3). However, when dealing with applied contributions in real-world settings, such as wearable computing, a system must be robust enough to operate with minimal burden. For example, assume the development of a new algorithm for gait analysis, a highly researched metric in wearable sensing; a systems research conducted in labs, but problems arise when transferring the work to free-living systems [165]. Devices must be reasonably unobtrusive while accounting for user activity, which introduces noise into sensing data. Thus real-world experimentation slows development

teams’ iterations, leading to projects never leaving the lab or being limited to isolated user studies.

I introduce FlexiBLE, a platform for a dynamic embedded sensing system interface designed for rapid prototyping of new systems in wearable healthcare. FlexiBLE serves as a case study on building reusable development infrastructure for early-stage wireless embedded sensing to move beyond bespoke implementations and towards generalizable tooling. FlexiBLE’s features include high-frequency data collection using a mobile phone or computer companion application (iOS or macOS), runtime parameterization, real-time storage and visualization, and near real-time visualization via InfluxDB². FlexiBLE is designed to be agnostic to the underlying hardware platform, allowing engineers to describe their systems via a configuration file that the application reads to produce dynamic interfaces and data structures on the fly. The system uses a timestamp-based data transmission protocol built on top of the Bluetooth Low Energy (BLE) Generic Attribute Profile (GATT) server.

The FlexiBLE iOS application is distributed via an invitation or a public link using Apple TestFlight³, the standard platform for testing applications before release to the Apple App Store. The current major version of FlexiBLE, 0.3, which is no longer in development but is in use at the time of writing, has 230 installations and 1,734 sessions, according to TestFlight analytics via App Store Connect, Apple’s online portal for app release management.

The remainder of this chapter introduces related work in this space, including similar toolkits, embedded sensing system development tools, and methodologies. I then provide a detailed overview of the FlexiBLE system and its components. I then detail several projects in which FlexiBLE has been used, including implantable cell oxygenation delivery systems and Physiohub. Finally, I reflect on FlexiBLE’s contributions and limitations through practitioner interviews.

²<https://www.influxdata.com/>

³<https://developer.apple.com/testflight/>

4.2 Related Work

Over the past fifteen years, health and behavioral research has expanded with new methods of mobile and ubiquitous computing. Wearables, peripheral devices that are worn by users to track localized physiology, have been greatly influenced by the maturation of Bluetooth Low Energy (BLE) as a communication standard for general-purpose "smartphones". This area remains an active research focus and a growing consumer market, overlapping with the Internet of Things (IoT) field. In this section, I review related work on toolkits for wearable healthcare development. We observe a persistent gap in the literature on systems specifically designed for the custom development of novel sensing modalities, and a recurring problem of ongoing support and abandonment.

4.2.1 Mobile Health Sensing Platforms

Kumar et al. conducted a systematic review of 37 mobile health sensing platforms (28 peer-reviewed, 9 non-peer-reviewed) [166]. Their analysis revealed several patterns: many works were limited in their demonstration or evaluation, yet they claimed generalizability to the field at large. For example, most frameworks were built for Android, most required additional development to support new sensors (if supported at all), and many collected data only from internal smartphone sensors. Finally, very few supported bi-directional communication. This work also categorized stakeholders into three groups: researchers, developers, and end users, noting a lack of systems that support all three. This review serves as a baseline for the historical context of mHealth toolkit development.

Smartphones have played a significant role in the historical development of wearable technology. Around the 2008-2012 timeframe, general-purpose smartphones with application development frameworks became widely available, enabling a natural progression toward leveraging their computing power and proximity to users for healthcare research. Platforms collected passive sensor data, prompted users with surveys, and collected enabled config-

uration for generalization across studies [167–169]. AWARE (2015) is a culmination of these internal smartphone toolkits, an instrumentation platform for capturing, inferring, and generating context on Android devices [170]. It provides reusable middleware that abstracts sensor retrieval into a higher-level context, enabling researchers to build custom experiments without writing code. Additionally, it includes management and data storage via a web server, enabling use in studies with thousands of participants. This architecture remains the baseline for large-scale smartphone-based behavioral studies.

Beiwe, another early yet long-lived tool, introduced full parameterization of studies via exportable JSON configuration files that specify study parameters [171]. These configurations include which phone sensors to use and how often to sample them. This tool, developed at Harvard, has been used in numerous clinical studies and is a prominent instrument for *digital phenotyping*, the practice of capturing behavioral patterns through smartphone use. Beiwe demonstrates the value of declarative configuration for reproducible research, focusing on behavioral data rather than, for example, biosignals related to psychological phenomena. However, experiment parameterization for clinical research applications is not unique; other platforms provide some form of customization, including: RADAR-base [172], Sensus [173], Ohmage [174], Lifestreams [175], MobiSens [176], and TigerAware [177]. Still, customization is limited to their respective features, providing little to no support for external sensing or prototyping of new sensing modalities. This creates a gap in behavioral research platforms and early-stage wearable sensing research.

In fully featured wireless and mobile health (mHealth) research, two platforms illustrate how custom sensing is integrated into participatory studies. mCerebrum (later MD2K, now minimally supported) has served 2,250 participants over 106K person-days, providing an end-to-end pipeline that ingests data from multiple BLE/ANT+ sensors, computes real-time biomarkers, and delivers context-aware interventions [178]. The Copenhagen Research Platform (CARP) Mobile Sensing framework (CAMS) [179] adopts a more modular approach,

providing a Flutter-based library that researchers can integrate into their own apps for new studies and devices, as well as study management features. Both platforms demonstrate a clinical need for generalizable in-situ data collection tools and the significant effort required to build and maintain them. Still, both are targeted at late-stage or production-grade devices with large participant studies, requiring manual coding to integrate new sensors. When working with early- to mid-stage prototypes, research teams often rely on their own methods for data collection and analysis, which is time-consuming [158].

Here, we focus only on platforms that emerge from research; other proprietary platforms exist in the market, particularly as they move toward formal clinical trials and FDA approvals. Regardless, a gap remains in tooling for the early development of new wireless sensing systems, which require extensive iteration, development, and debugging. Currently, integrating new wireless devices into existing platforms requires writing custom, platform-specific tooling in parallel with hardware development.

4.2.2 New Hardware, New Software Applications

It is standard practice in consumer technology for every device, or vendor, to have its own software toolkit to connect to its own devices; there’s no *universal interface*, and for good reason: each device is unique, and it’s beneficial to control the system end-to-end. Additionally, the productionization of consumer devices warrants the effort to build customized interfaces. When prototyping small-scale systems, building entire software suites is a burdensome task; however, this is a default in the literature: a new hardware system spawns the development of a new *upstream application* (mobile phone application, desktop application, cloud server, or ad-hoc tooling for data collection). Our own systems are subject to this; FaceBit developed its own companion app [116]. Open-source hardware platforms such as OpenEarable 2.0 [180], Lumos [181], and Walking Wizard [101] each have their own mobile application software solutions.

Outside of research, a few options exist to streamline data collection or device interaction, most of which are IoT platforms that require an internet connection to access. ParticleIO, which is used in a later example (chapter 6), is an entire infrastructure solution supporting data collection and parameter syncing over cellular networks. Many similar internet-enabled platforms exist. In open-source, there are, for example, ThingsBoard⁴, Adafruit IO⁵, and DeviceHub⁶, as well as many enterprise options from Amazon Web Services, Microsoft Azure, and Google Cloud. For BLE, there are fewer options outside of pure BLE scanning and GATT parsing applications. BLNYK⁷, however, is an exception, offering BLE-to-cloud capabilities with declarative interface programming, but it is expensive to scale and has a limited ecosystem.

4.2.3 Sensor Integration Middleware

A parallel research thread exists focused on a middleware problem: how to connect heterogeneous external sensors to mobile devices through reusable abstraction layers. This is more closely related to the problems we faced in our existing research. We need standardized, generalizable methodologies for sensor data collection.

Currently, BLE prevails as the standard for short-range communication in consumer wearable devices and is therefore a strong option for research prototypes in the same form factor. However, the need for generalizable methods predates the widespread adoption of BLE. Dandelion and MobiCon explored query-based and context-driven approaches to coordinating sensors on mobile devices [182, 183]. Healthopia proposed a declarative query abstraction for sensor data, focusing on reducing the complications of accurate data collection in new peripherals [184]. Open Data Kit (ODK) is a practical example of this middleware philosophy for field-deployed systems [185, 186]. This system targeted low-cost Android and

⁴<https://thingsboard.io/>

⁵<https://io.adafruit.com/>

⁶<https://devicehive.com/>

⁷<https://www.blynk.io/>

decomposed monolithic sensing applications into modular components. These contributions handled low-level concepts such as threading, buffering, and connection management, whereas drivers could be implemented using only sensor-specific commands. Finally, HealthOS [187], proposed in 2012, introduced JSON-based device abstractions that could semi-automatically generate adapter code for new sensors. This decoupled the application from device-specific protocols through a declarative description layer and a modular architecture for adapters (device interfaces) and pipelines (data transformations). The technical contributions of these works are no longer relevant to modern systems; however, their established principle that sensor integration should be mediated by an abstraction layer rather than handled through monolithic, device-coupled software aligns well with the issues commonly faced in early research prototypes.

4.2.4 Positioning FlexiBLE

To the best of our knowledge, no solution exists that focuses on application interfaces for prototyping new wearable systems. FlexiBLE is motivated by difficulties encountered in developing FaceBit, and others have expressed a need for better data collection methods [31]. Our platform is designed specifically for rapid prototyping, reducing feedback loops in the complex real-world evaluations required for wearable devices, and providing structured data management for research analysis.

4.3 System Requirements

The goal of the FlexiBLE system was two-fold: first, to alleviate some of the developmental challenges faced by FaceBit [116] (Chapter 3), and second, to create a tool that enables new methods for similar projects. In summary, much of the technical scaffolding developed for FaceBit’s development, e.g., sensor data collection and data processing, was not unique to FaceBit’s functionality; however, it required additional engineering to decouple for use in

other applications. Our goal was to generalize a core set of features to enable the reuse of a system built using FlexiBLE. This section outlines the platform’s main goals and motivation.

4.3.1 Data Collection

The most crucial component of wearable prototyping is data integrity. An early requirement in all projects is that data be collected and well-situated for analysis. To quantify this, we set a general goal that a tool should be able to collect data from multiple devices (<5), each with multiple sensors (3), at frequencies of up to 500 Hz (aggregate per device). These metrics are high for standard wearable functionality but are necessary for an early prototype to collect raw data from peripheral sensors.

Additionally, data collection must be conducted with minimal resource consumption to ensure usability in the wild over extended periods. The specific time limit is left undefined, as it depends on the use case and the device’s position. A general rule of thumb is at least a few hours of use at maximum utilization, or at least a full day (12+ hours) with runtime optimizations and reduced collection rates.

Managing free-living experiments in real time requires wireless data collection (e.g., BLE, WiFi, or Cellular). The vast majority of wearables use BLE and a mobile-phone gateway (a dedicated application). Local data storage is feasible with onboard non-volatile memory or SD cards, but it can slow iteration cycles because data cannot be viewed in real time. Therefore, we consider wireless communication a requirement for developing efficient development processes.

Finally, time is a critical concept in data collection. Most, if not all, data collected is in the form of a *timeseries*, a series of data points indexed by a timestamp. These timestamps can be relative. However, it is often useful to use an absolute timestamp for external reference. This can be challenging on embedded systems, especially when operating at millisecond- or microsecond-precision. Our use-cases require absolute timekeeping for multi-

device and ground-truth synchronization. Again, precision is use-case dependent, but we target millisecond precision.

4.3.2 Runtime Flexibility

By definition, embedded systems have limited interfaces, especially when designing for low-power, wearable form factors. During the prototyping or testing phases of development, many parameters often need adjustment, or processes need to be initiated on command. With current methods, as experienced with FaceBit (Chapter 3) and discussed in related work (Chapter 2), updating parameters requires flashing the device, tethering early experiments to the lab, or designing bespoke interfaces (companion applications). Practitioner interviews confirmed this friction, with users noting that re-flashing alters the device’s interaction with its environment (Section 4.7). Each takes time and is ill-suited to the rapid pace of development. Our system aims to serve as a *universal companion app* across devices, allowing firmware developers to define interactions without requiring application updates. This requires a declarative abstraction to runtime interfaces and an adaptive architecture similar to some of the early middleware research tools mentioned in section 4.2 [182–187]. The goal is to reduce the requirement on upstream software (systems *above* the firmware in the wearable system stack).

4.3.3 Data Interoperability

The ability to easily extract data in human-readable formats widely supported across systems is important for post-processing and integration with upstream systems. When working with embedded systems and wireless protocols, data streams are often compacted to save space and increase relative throughput. These transmission structures require external knowledge to be deciphered and may be prone to errors when parsed across systems and programming languages. For a robust system, data should not only be efficiently transmitted, but also stored in non-ambiguous formats that support interoperable data types. This amounts to a

system that provides data (sensors, metadata, and parameters) in common formats and with sufficient context to avoid requiring domain knowledge of its source.

4.4 FlexiBLE System Design & Implementation

Motivated by our experiences with FaceBit, the tools available at the time, and our refined requirements, we present the implementation details of the FlexiBLE platform. Named for its *flexibility* in wearable prototyping and its use of the *BLE* for communication. The platform includes three components targeted at early prototyping of wearable systems for in-the-wild use; we detail each here.

4.4.1 FlexiBLE Device Specification

The platform centers around an external data structure that defines a peripheral’s capabilities; we call this the device specification. It’s a JavaScript Object Notation (JSON) configuration file, formatted according to JSON Schema⁸, and is shared across the end-to-end FlexiBLE platform. At a high level, this file is generated by a firmware engineer, outlining all the sensors and parameterization, and then shared for import into the companion application. While the firmware need not explicitly follow the specification, it serves as a human-readable description of the device and includes data-parsing parameters.

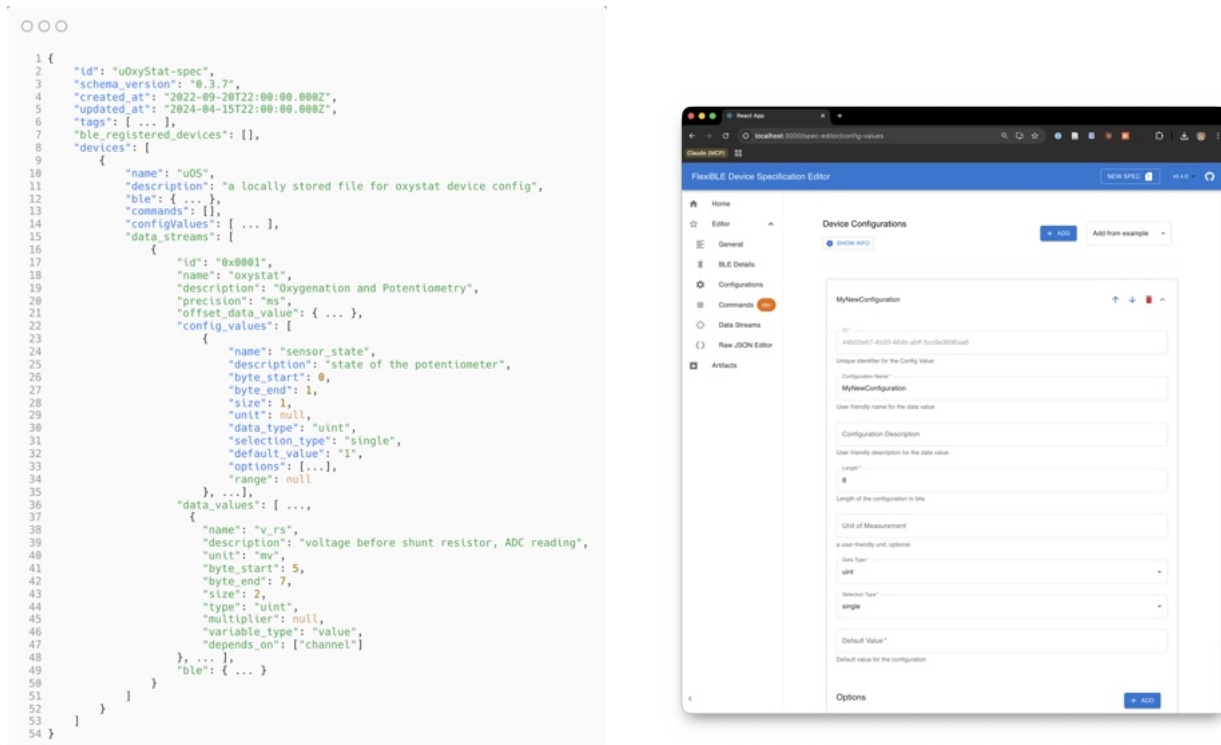
The specification includes high-level metadata about the device, including its name, description, BLE details, and a list of supported sensors. It also includes detailed configuration for each sensor, including data formats. The specification is structured hierarchically, with top-level metadata and nested sensor details, referred to as *data streams*. An example of the structure is shown in Figure 4.2a. At the top level, the specification is versioned and assigned a unique identifier, after which specific devices can be listed. Each device has a name, description, BLE details, commands, configuration values, and a list of data streams. Commands are

⁸<https://json-schema.org/>

invocations of procedures on the device, such as software resets or the execution of a specific firmware function, for example automated calibration. Commands are request-response pairs and can optionally return values. Configuration values include individual variables such as a toggle, a numeric value or range, a selection or multiple-choice, and text. There are configuration values at the device level (global configurations) and at the data-stream level for a specific sensor functionality. Finally, the configuration values structure includes fields for data packing, such as byte offset, size, and data type, as well as human-readable details such as name, description, and unit.

Data streams are lists of data to be collected by the device, usually mapping to an individual sensor, but can also map to a multi-modal onboard processing pipeline. Data streams assume a time-series format and include details on time precision and offsets (the time between readings; see Section 4.4.3 and 4.4.2 for implementation details). Data streams contain a set of configuration values specific to the function, with a single required enable toggle assumed (turning the sensor on/off). The data value arrays contain details about the collected data, with packet-packing details similar to those in the configuration values.

Specification files are loaded into the companion application and ingested for rendering dynamic user interfaces and data schemas. This enabled firmware updates that introduced new parameters and data without requiring upstream application updates. In practice, the device specification showed clear benefits but also posed challenges, particularly with manual JSON editing. We discuss this further in Section 4.7. We explored a web-based specification builder (Figure 4.2b), which avoids errors and generates firmware code for the BLE interface. This was never fully tested but demonstrates the feasibility of a more user-friendly specification-editing workflow.



(a) An example device specification in JSON format. (b) The FlexiBLE specification builder interface.

Figure 4.2: (A) An example FlexiBLE device specification in JSON format. This specification is an abbreviation of an implantable cell oxygenation device used in real studies. (B) A prototype web application for declaratively building device specifications.

4.4.2 FlexiBLE Companion Application

The FlexiBLE companion application is an iOS application that provides a dynamic user interface and database schema for interacting with any conforming device. The application is written in Swift⁹ and uses the CoreBluetooth framework for BLE communication, Apple's native library for interfacing with onboard BLE-enabled radio hardware on Apple products. The application also runs on macOS Catalyst, a feature that allows iOS/iPadOS applications to run natively on macOS with minimal code changes. Developing FlexiBLE in the Apple ecosystem was chosen purely for convenience, given my industry experience. However, regardless of the OS selection, cross-platform BLE API support remains a challenge in embedded development. On iOS, the application allows testing in the wild, FlexiBLE's

⁹<https://swift.org>

original goal, and on macOS, it provides a desktop interface for in-lab experimentation (see Section 4.6).

The FlexiBLE application requires a device specification to be provided on first run. Once loaded, the application runtime displays an interface based on the specification. *Dynamic* refers to the conditional rendering and data-driven functionality provided by the device specification data structure. Once a specification is loaded, the application scans for nearby devices within the specification, allows connections, and presents a device details view populated with information from the specification. Configurations can be updated globally or per data stream, and data are visualized in real time. Additionally, the application supports auto-connection to previously discovered devices and automatically loads previously stored configuration states upon connection. Configurations are presented via a widget-based UI tailored to the data type. Figure 4.3a shows a series of numerical range slides for several configurations of a multi-wavelength Photoplethysmography (PPG) sensor.

Upon a successful BLE connection, the application performs a handshake to configure the device for long-term operation. This includes first negotiating the maximum transmission unit (MTU) with the device and other BLE-specific connection parameters. Next, the GATT server is discovered, and all data characteristics are subscribed to for real-time data streaming (see Section 4.4.3 for BLE Generic Attribute Profile structure). Then, the reference time is written to the device to convert system uptime to calendar time (specifically the Unix Epoch). Finally, all configurations are read and written if configured to do so (a setting to write defaults or last values on connect). At this point, the application considers a device "connected," and the firmware can freely send data over BLE, and the user can interact with the device in the user interface.

All data is stored locally on the device in a SQLite database. SQLite was chosen for its interoperability and broad support across systems and languages. There is no Apple-specific

dependency on the data storage; however, we use the open-source GRDB library¹⁰, which provides a Swift interface to SQLite. A core SQLite schema is initialized on first run and stores application and device metadata. When a device specification is loaded, new tables are created to store transactional device details relating to data streams. This enables statically typed representations of data parsed in real time from the device. Figure 4.3b and 4.3c show data stored in the application. All data is stored in the database with context (metadata) persisted; this supports interoperability. The device specification or a deep understanding of FlexiBLE is not required to use the database for post-processing. As a secondary requirement for specific experiments, we implemented a cloud upload feature in FlexiBLE to send time-series data to an InfluxDB¹¹ database via the Telegraf¹² ingestion engine. This allowed fully remote monitoring, specifically helpful for experiments in a vivarium (see Section 4.6)

The application also provides a user interface for time-boxing experiments and for taking notes. Experiments are tagged with start and end times, which facilitates post-processing filtering. Time markers are single time points that can be added to the timeline, for example, to report an event of interest. For convenience, when an experiment is ongoing, the application can optionally track GPS from the device’s internal sensor (iPhone/iPad only). An example experiment is shown in Figure 4.3d.

4.4.3 Hardware and Firmware Development

FlexiBLE is designed to be agnostic to hardware and firmware, except for the requirement of BLE, and treats the firmware as the end-to-end system’s *source of truth*. This means that FlexiBLE expects firmware changes and provides a no-code path to accommodate. This is done by conforming to a specific set of BLE attributes and using the device specification file, as mentioned earlier.

¹⁰<https://github.com/groue/GRDB.swift>

¹¹<https://www.influxdata.com/>

¹²<https://www.influxdata.com/time-series-platform/telegraf/>

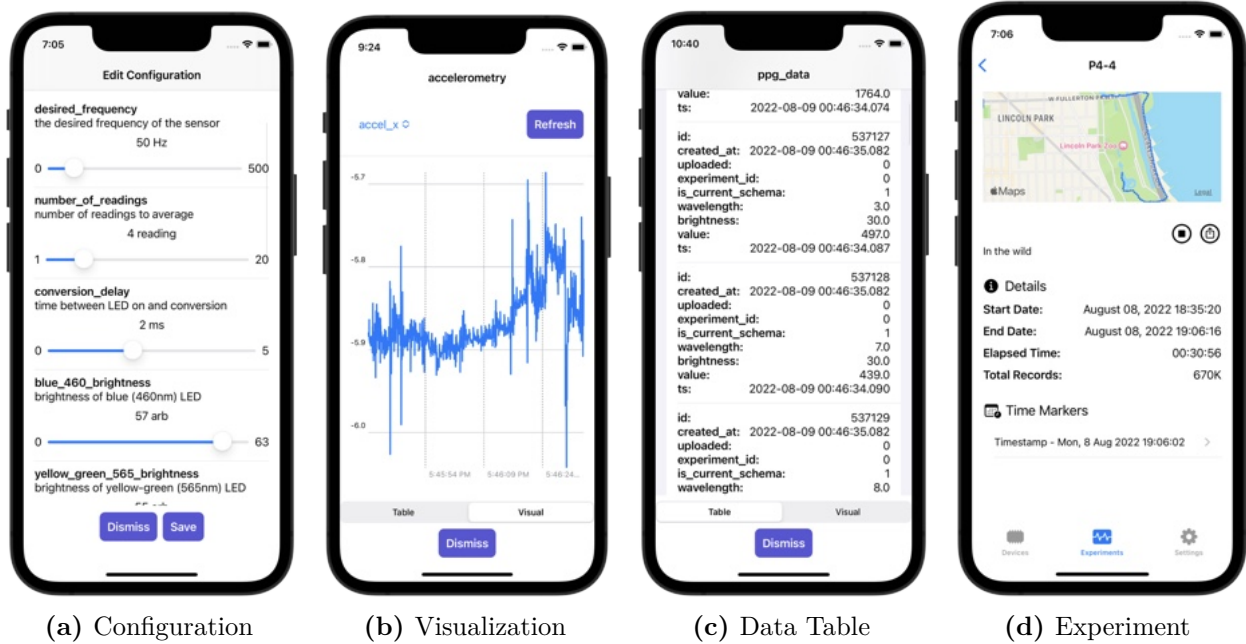


Figure 4.3: Screenshots of the FlexiBLE iOS application. (a) Runtime parameter configuration interface with dynamically generated sliders for sensor settings such as sampling frequency and LED brightness. (b) Real-time data visualization of accelerometer data. (c) Tabular view of raw PPG sensor data with timestamps and metadata. (d) Experiment detail view showing collection metadata, GPS trace, and time markers.

Any microcontroller system-on-a-chip (SoC) that supports BLE can be used as a FlexiBLE device. The BLE specification and Generic Attribute (GATT) profile structure are used to identify and structure data packets exchanged between a central device (FlexiBLE companion app) and a peripheral device (embedded device). Additionally, any BLE device can define a GATT profile (host), and after a connection is established, the other device (client) can retrieve information from that profile. GATT Profiles are structured with services (groupings of characteristics), characteristics (data storage), and descriptors (that describe characteristics), each of which is defined using a single attribute table with 16- or 128-bit UUIDs. 16-bit globally defined assigned numbers are used to identify registered services and characteristics (heart rate, calendar time, etc.) as defined by the BLE specification¹³. Developers may define their own attributes using 128-bit UUIDs for arbitrary data. Characteristics have permissions for read, write, notify, and indicate. Notify and indicate are asynchronous updates that may

¹³<https://www.bluetooth.com/specifications/assigned-numbers/>

be sent to the central during a connection, while read and write are direct requests. Descriptor attributes are designed to *describe* a characteristic; they are seldom used in practice, and uncommon or custom descriptors lack broad support in Bluetooth client stacks. Thus, we opted not to use them, preferring a separate configuration file.

The supported firmware requires a set of custom GATT services and characteristics. First, a primary service, referred to as the *Info Service*, hosts universal characteristics for all FlexiBLE-enabled devices and is included in the BLE advertising packet to identify FlexiBLE devices. Currently, the Info Service requires a characteristic that holds a reference timestamp, mapping the embedded system's internal timekeeping (uptime or wall-clock via RTC) to absolute time. This is performed by the companion application upon connection, writing the current Unix Epoch timestamp in milliseconds to the characteristic. On write, the firmware records the epoch time from the companion app and uses it as a reference point for system uptime. A limitation of BLE is that there will be variable delays when writing the reference time, resulting in a shift in the data timestamp. This offset should fall within the tens to hundreds of milliseconds (depending on connection interval and latency configurations) and is a single shift per reference time update, which is acceptable in most use cases.

Sensor data collection is managed through *Data Streams*, which map to separate GATT services with two characteristics, data and configuration. The configuration characteristic has read/write permissions and contains a developer-defined byte array that stores runtime parameters of the data stream. The packing of the byte array is determined by the configuration file's ordering and data types. In addition, the Info Service contains a separate configuration parameter for global parameters, following the same format. At write time, this characteristic is parsed as runtime parameters referenced in the firmware logic. The data characteristic has notify permissions and is a buffer array of sensor values, again defined by the ordering and data types in the configuration file. Each data packet has a header containing an anchor timestamp (the number of milliseconds since the reference time), multiple sensor-data

sub-packets, and offset times separating the sub-packets. For example, an IMU sensor is represented by three accelerometer readings along the x-, y-, and z-axes. A data packet would include a single anchor timestamp, followed by sensor values, then an offset for the next set of sensor values, and so on, repeating until the packet size (defined by the BLE maximum transmission unit) is filled.

Conforming to this BLE structure and producing a specification file allows the companion application to interact with the firmware. The firmware itself is agnostic. For example, we have used Arduino (BlueFruit), Zephyr RTOS (Nordic SDK), and Silicon Labs Gecko SDK to develop FlexiBLE devices.

4.5 Application: Physiohub

PhysioHub was originally designed as an external wearable to monitor biometrics and control an implantable device that produced hormones to regulate circadian rhythm, as part of the Defense Advanced Research Projects Agency (DARPA) Advanced Acclimation and Protection Tool for Environmental Readiness (ADAPTER) program. The device was deployed in early non-human primate (NHP) studies but was not used long-term and was subsequently repurposed for research on wearable capabilities for multimodal cardiovascular sensing. PhysioHub was built using FlexiBLE, demonstrating the system’s utility in high-demand wearable applications. PhysioHub is a strong, demanding use case for the platform, but it also exposed protocol-level limitations, which are discussed further in Section 4.7. In this section, I first introduce the system and its implementation, and then summarize the results. At the time of writing, PhysioHub is accepted for publication [163]¹⁴

¹⁴© 2026 IEEE

System Overview

Multi-node wireless body sensor networks for cardiovascular sensing require temporal precision that existing platforms cannot provide in typical free-living wearable configurations. Multimodal insights from emerging noninvasive cardiovascular metrics, such as Pulse Arrival Time (PAT), Pulse Transit Time (PTT), Pre-Ejection Period (PEP), and Low-Frequency photoplethysmography (LF-PPG), rely on signals from multiple body locations, captured with sub-millisecond synchronization. Current solutions require wires to connect distributed sensors or proprietary base stations for multi-node connectivity [188–191]. To the best of our knowledge, no device exists that enables multi-node wireless exploration of noninvasive cardiovascular sensing in free-living wearable systems. PhysioHub was developed to explore the feasibility of wireless multimodal cardiovascular sensing in free-living wearable configurations. We evaluate PhysioHub in a series of benchtop evaluations and a 15-person user study.

PhysioHub consists of multiple wearable nodes. Each node has a main and an auxiliary custom-printed circuit board (PCB), a rechargeable LiPo battery (200 mAh), and a 3D-printed enclosure (Formlabs BioMed Black). The main PCB hosts a full-featured 2.4 GHz wireless module (u-blox NINA-B302, based on Nordic Semiconductor’s nRF52832 system-on-chip), power management circuitry, and a variety of onboard sensors, including the inertial measurement unit (IMU) used in this work. The auxiliary PCB is connected to the main PCB via a low-profile board-to-board connector exposing communication buses, GPIO, and analog inputs. This work utilizes two auxiliary boards. The first is a two-lead ECG signal-conditioning block (Analog Devices AD8233). The second is a custom multi-wavelength PPG sensor that was developed over three generations. The custom PPG configuration uses Analog Devices’ ADPD4101 multimodal sensor front end, four photodiodes, and eight LEDs spanning four wavelengths: green (525 nm), orange (610 nm), red (660 nm), and near-IR (940 nm). Future applications can develop new accessory boards using this form factor. Figure 4.4 shows the main and accessory boards and custom case design for each configuration.

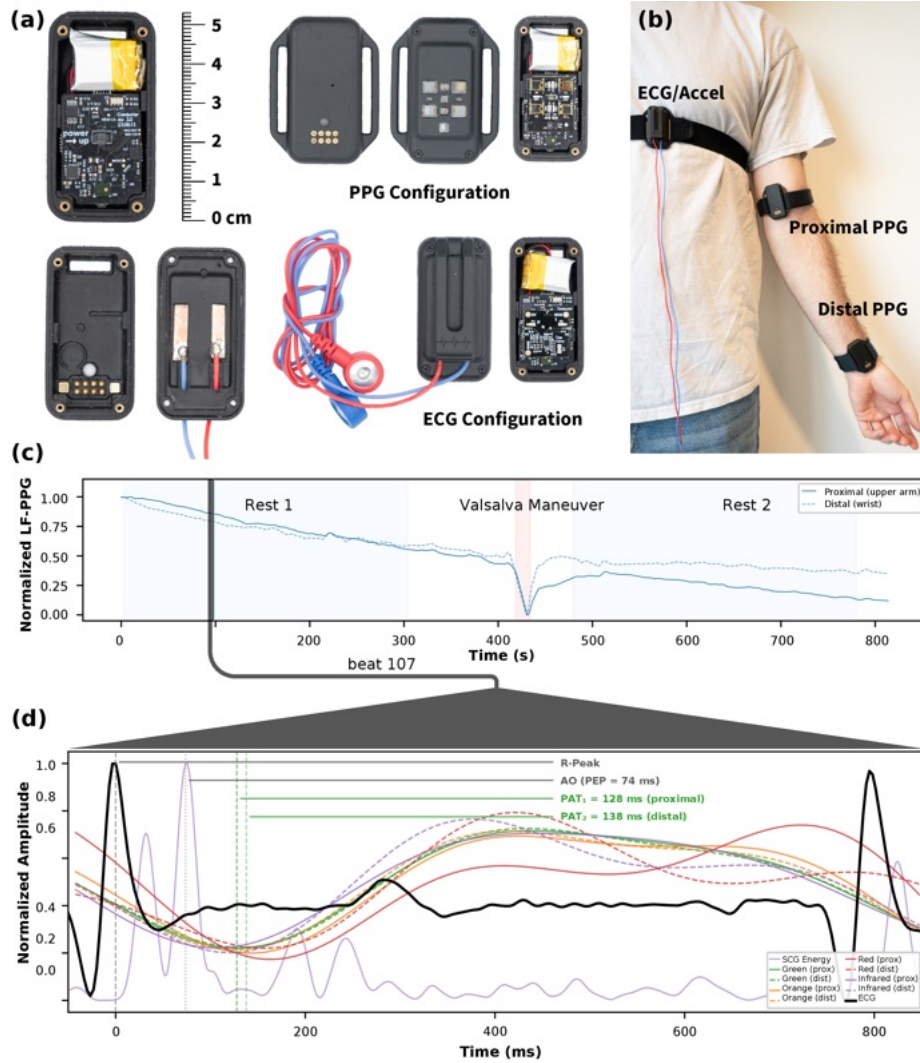


Figure 4.4: PhysioHub system overview. (a) shows the main and auxiliary PCBs, as well as the custom case design with photoplethysmography (PPG) and electrocardiogram (ECG) configurations. (b) shows the placement on the body for our in-clinic evaluation. (c) A single participant graph of low-frequency photoplethysmography (LF-PPG) showing a clear response during the Valsalva maneuver. (d) A single R-R interval from the above participant, showing all cleaned data collected by the system, including distal and proximal multi-wavelength PPG signals, ECG, and seismocardiography (SCG), with fiducial points for each.

Each node runs firmware based on the Zephyr RTOS¹⁵, which manages sensor drivers and data collection over BLE. Each node is assigned a peripheral BLE role and connects to a central mobile application based on the FlexiBLE companion application, but is slightly modified to support PhysioHub-specific features and the user interface. FlexiBLE’s device

¹⁵<https://www.zephyrproject.org/>

specification, including data streams and configuration, was used for data collection and runtime interaction. Configurations include PPG wavelength selection, sensor frequency, and other low-level sensor parameters.

BLE cannot support time synchronization with the precision required to capture sub-heartbeat features, so we rely on a new method in which nodes communicate to synchronize their clocks internally. One node is designated as the *leader* and broadcasts a synchronization signal to all other *follower* nodes using Nordic Semiconductor’s RADIO HAL API¹⁶. For this use case, FlexiBLE was utilized to designate different devices as *leaders* and *followers* while connected to the companion app. This assignment triggered the firmware to start the node-to-node clock synchronization procedure. A prerequisite is that BLE time synchronization has been established, anchoring device uptime to absolute time. The initial time anchor includes a variable offset based on BLE packet intervals and latency; however, this is acceptable since it is a one-time shift in the time series. The node-to-node sync keeps the system clocks in sync.

Benchtop Evaluations

The ability to synchronize time across devices is a novel feature of Physiohub, enabled by FlexiBLE’s infrastructure. We set up an experiment to test the accuracy of time synchronization between nodes by connecting a GPIO pin on each node to a signal generator producing a 10 Hz square wave and enabling an edge-triggered interrupt. We then define a FlexiBLE data stream that reports the polarity of each edge with microsecond precision. We analyzed reporting from data streamed to the mobile app with a single leader and two follower nodes. The results showed an average leader period of $50,002.48\mu s$ ($50,000\mu s$ expected), with a standard deviation of $16.58\mu s$, demonstrating the system’s direct noise. The following captured 99.42% and 99.18% of the edges, with mean errors of $5.38\mu s$ and $7.79\mu s$, respectively. Errors are quantized to multiples of $30.5\mu s$ due to the resolution of the system’s real-time clock (RTC). This is visualized in Figure 4.5a, showing only a few outliers $> \pm 200\mu s$. Additionally,

¹⁶ <https://docs.nordicsemi.com/bundle/ncs-2.5.3/page/nrfx/drivers/radio/hal.html>

Allan deviation decreased monotonically from $\sim 2100\mu s$ ($\tau = 50ms$) to $< 0.2\mu s$ ($\tau = 27min$), following a τ^{-1} slope consistent with white phase noise, confirming no systematic drift.

Additionally, we performed benchtop evaluations to validate each sensor used in cardiovascular sensing, ECG, multi-wavelength PPG, and accelerometer. This was done with similar setups, comparing measurements from a digital oscilloscope (RIGOL DHO900S). All reported favorable responses and residual timing errors passed a Shapiro-Wilk normality test (ECG: $p = 0.44$, PPG: $p = 0.26$, Acceleration: $p = 0.90$) with negligible temporal drift, confirming that the observed errors are stochastic.

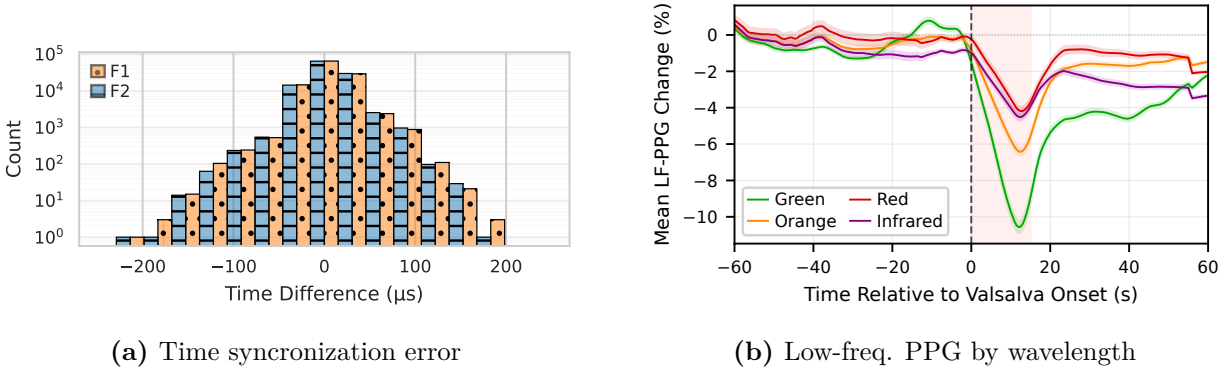


Figure 4.5: PhysioHub evaluation results: (a) time synchronization accuracy between leader and follower nodes, and (b) physiologically meaningful LF-PPG response captured during in-clinic evaluation.

In-Clinic Evaluation

We conducted an in-clinic evaluation with 15 participants (Northwestern University IRB protocol STU00221013) to demonstrate Physiohubs' capabilities. Each participant wore three nodes: one on the abdomen to record a one-lead ECG (250 Hz) and acceleration (416 Hz), and two on the upper arm and wrist to record 4-wavelength PPG (75 Hz). Participants performed a 15-second Valsalva maneuver, followed by 5-minute rest periods. Across all participants, the system transmitted 127 MB of data (7.46M records over 153 minutes), with data reliability (count of samples over the expected count) exceeding 98.5% across all modalities and an average BLE bandwidth of $14.12 \pm 1.83kB/s$. Post-collection signal

processing yielded 15,105 detected heartbeats, of which 75.8% produced simultaneous valid PPG onset and SCG aortic opening detections, enabling beat-level decomposition of PAT into PEP and PTT. Beat-level correlations confirmed that PEP and PTT capture independent cardiovascular information ($\bar{r} = 0.21$ between PAT and PEP; $r(\text{PAT}, \text{PTT}) = 0.77$), consistent with expected physiological behavior.

To evaluate whether the system captures physiologically meaningful dynamics, we examined the LF-PPG response during the Valsalva maneuver (Figure 4.5b). Across all four upper-arm wavelengths, a consistent decrease in LF-PPG amplitude was observed during the strain phase, reflecting reduced venous return and diminished peripheral blood volume. Following the release, LF-PPG recovered toward the rest baseline. Individually, 12 of 15 participants showed a clear decrease in amplitude, and a Wilcoxon signed-rank test confirmed a statistically significant reduction across all wavelengths ($p < 0.01$, $|d| > 0.75$). These results demonstrate that Physiohub can reliably capture multimodal cardiovascular signals, identify well-characterized hemodynamic responses, and enable the useful application of FlexiBLE to new devices (RQ2).

4.6 Application: Cell Oxygenation Delivery Systems

In projects funded by both the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research Projects Agency for Health (ARPA-H), FlexiBLE enabled wireless control of subcutaneously implanted devices for cell oxygenation delivery systems. FlexiBLE provides real-time biomedical research data and in vivo device parametrization, a capability not achievable with transitional early-stage biomedical research systems. This fundamentally changes the experimental methodology for early-stage implantables, enabling a wide range of rodent implantations and offering benefits over wired experiments that require full sedation, limited movement, or data extraction from onboard flash storage. The latter requires post-surgery extraction, poses a risk of data loss due to electrical damage, and offers no

in-experiment adjustment. We discuss these implications further in Section 4.7 and provide an overview of two specification applications, summarizing the work of two publications [164, 192].

4.6.1 Electrocatalytic On-Site Oxygenation Platform

Implantable cell therapies, where engineered cells are transplanted to produce therapeutics in situ, hold promise for treating diseases such as diabetes, cancer, and autoimmune disorders. One challenge is that these systems deliver oxygen to cells in subcutaneous implantation environments with limited oxygen tension (39 mmHg). At the high cell densities required for clinical relevance ($60\text{K}+$ cells/ mm^3), oxygen diffusion from the host vasculature is insufficient, leading to cell death within days. To address this, a multi-institutional collaboration developed an electrocatalytic oxygenation platform (ecO2) that generates oxygen directly at the implant via water electrolysis using sputtered iridium oxide film (SIROF) catalysts. SIROF enables efficient oxygen evolution at neutral pH within a safe voltage window (1.6-1.9V) without producing toxic byproducts. In vitro, exO2 sustained ARPE-19 cell viability at 83% over 21 days under hypoxic conditions, compared to near-total cell death without oxygenation, while also maintaining therapeutic peptide (leptin) secretion.

To validate ecO2 in vivo, devices were implanted subcutaneously in rats and operated by a remote potentiostat connection via BLE to the FlexiBLE companion application running on macOS. FlexiBLE enabled researchers to wirelessly adjust the oxygenation voltage, measurement frequency, and duty cycle in real time during experiments, without physical access to the implant.

4.6.2 Hybrid Oxygenation Bioelectronics for Implanted Therapy

Building on the results of ecO2, the platform evolved into HOBIT (Hybrid Oxygenation Bioelectronics for Implanted Therapy), a fully implantable, wireless device. HOBIT packages the ecO2 catalytic chip, the remote potentiostat, and a medical-grade battery into a hermeti-

cally sealed, biocompatible PDMS enclosure. The FlexiBLE application again served as the external control and data-collection interface, running on macOS in the lab and uploading data to InfluxDB. A 31-day in-vivo experiment in rats demonstrated HOBIT’s capabilities by co-encapsulating three engineered cell types that produce therapeutics with varying half-lives. Oxygenated devices maintained 64.6% cell viability, compared with 19.2% in controls. A separate longevity study showed sustained serum levels through day 28 in oxygenated devices, while controls dropped to baseline levels.

Across both studies, FlexiBLE provided the critical interface for runtime parameterization of oxygenation voltage and real-time collection of potentiostat current measurements for multi-week in-vivo studies. This application demonstrates the need for FlexiBLE’s feature set beyond our initial use case of wearable sensing, serving as infrastructure for implantable bioelectronics devices where physical access to the hardware is not possible after implantation. However, because FlexiBLE was not intentionally designed for this application, adapting it to this environment was challenging.

4.7 Reflections from Practice

To conclude the development of FlexiBLE, which is no longer actively supported, I conducted three semi-structured interviews with users and developers. The goals were to explore the usefulness of FlexiBLE’s two primary use cases and synthesize what worked well and what requires improvement in a future iteration of the tool. I conducted these interviews over Zoom and automatically transcribed them using Otter AI via Zoom. An interview guide was used to structure the questions and the flow of the interviews, but was not followed verbatim. The study was reviewed by Northwestern University’s IRB and deemed exempt (STU: 00224556).

Interviews were coded using a pre-defined codebook (deductive coding) aligned with the study’s goals. This included 3 themes: participant context, development process, and

FlexiBLE Experience. 12-codes were applied across 98 entries from the three participants. The codebook and interview guides are available in Appendix B.1.

All participants had a direct relationship to FlexiBLE or to research in which FlexiBLE was used; I have worked directly with each participant. Participant 1 (FP1) is a master's student tasked with rewriting the data-collection functionality of FlexiBLE to improve signal quality on Linux. They used FlexiBLE as a starting point and gained a deep understanding of the tool's architecture and capabilities. Participant 2 (FP2) is a Ph.D. student and practitioner who is a direct contributor to FlexiBLE's design from the perspective of embedded systems and hardware integration. Participant 3 (FP3) is another Ph.D. student studying biomedical engineering and is an end-user of FlexiBLE. They used FlexiBLE extensively for HOBIT work, both through the companion application and by analyzing system data.

4.7.1 Practitioner Context and Development Landscape

Each participant comes from a different background and has different perspectives on FlexiBLE. FP1 and FP2 are developers using FlexiBLE for new development efforts, while FP3 is an end user using FlexiBLE for a real-world application not related to the toolkit itself. FP1 is a newcomer to wireless embedded system development and is extending FlexiBLE's functionality to enable exploration of the signal quality of existing peripherals outside the Apple ecosystem. FP2 is a hardware and firmware engineer building new devices with FlexiBLE. FP3 uses FlexiBLE for data collection and device control in biomedical in vivo studies. FP3 has a formal background in electrical engineering with some industry experience, but was not involved in the development of FlexiBLE.

The participants' experiences range widely. FP1's previous experience in embedded systems was limited to "academic goals" where they explored core embedded system concepts and wired communication protocols. FP2 brings both industry experience and research experience in PCB and firmware development. FP3 works exclusively with implantable devices for

early-stage research. FP2 highlights the demands of custom hardware in cutting-edge research tasks, noting that 'we try and use off-the-shelf as often as we can, because making PCBs and custom firmware is a huge pain, it's risky, et cetera, et cetera.' They state the most common constraint is size: 'off-the-shelf stuff just is not made to be super tiny, and most of the devices I've made had to be, for one reason or another.'

FP2 also mentions the challenges in sensing in early-stage work, "where there are a lot of unknowns", meaning you must sense more than what an end-product would. FP3 echoes this requirement: "data gathered from [a device], eventually, a user of the system data wouldn't see it all, ideally, it's all handled onboard the embedded system," explaining how, in early-stage devices, all data is collected with the vision that eventually it would be processed onboard once the environment and capabilities are understood. This mirrors the algorithm development challenges faced by FaceBit (Section 3.6), where extensive data collection is required for iterative development. This landscape complements our personal experience and the literature. We now examine how FlexiBLE addresses these challenges and where it falls short in practice.

4.7.2 FlexiBLE's Contribution to Development Workflows

Each participant was asked to define FlexiBLE in their own words and to describe in detail how they used the platform, thereby demonstrating that FlexiBLE can serve different roles. FP1 used FlexiBLE as 'a reference for the Linux BLE communication app' where the 'whole flow is basically pretty much similar to FlexiBLE, like, discovering, connecting, configuring, and keeping them updated if we change anything remotely'. Essentially, FP1 was developing a FlexiBLE client for data collection on Linux, enabling exploration of different antennas. FP2 used FlexiBLE as 'protocol' for their custom device development that 'served a lot of purposes' from debugging to 'interfaces that I could hand off to stakeholders who would be using the devices in the field'. FP3 was one of the stakeholders FP2 mentioned, using

FlexiBLE's interface and sensor data in their research, describing FlexiBLE as 'a streamlined interface to talk to wireless embedded devices'.

Typical workflows involve developing one-off tooling (section 2.3.2). FP3 expressed limitations of bench-top equipment, such as running experiments from an oscilloscope, "a repeated cycle measurement with a lot of bench-top equipment, you're just limited on what knobs you can turn." FP2 describes a traditional workflow in which "iterat[ing] on the hundreds or thousands of different settings that are in the [firmware]" requires recompiling and flashing the device, which "changes how the device interacts with the physical environment, or you can't move around like you naturally would." FP1, given limited experience, had difficulty describing a hypothetical workflow with FlexiBLE and instead reverted to examples from coursework.

FlexiBLE alleviates these pain points by providing a customizable, low-code interface which enables '10 experiments in the time where I used to only be able to do 1' (FP2), and in the case of in-vivo studies, instead of having to wait until it was over, to get some of that feedback and know how we were doing' (FP3). These features completely altered the typical workflow, allowing FP3 to 'reduce the number of animals [we] used because of [real-time feedback]', and kicking off hardware iterations while current in-vivo studies were ongoing. FlexiBLE helped both end-users and developers, enabling a confident handoff between them: 'I knew that I could give the researchers just the switches and ranges that they needed' (FP2). FlexiBLE provides features that do not currently exist in these types of workflows, 'I don't know what I would have done, like, to give them that ability. It would have been a one-off thing, it would have been probably pretty clunky' (FP2).

FlexiBLE's strengths were stability, time savings, and the specification-driven approach to firmware updates. For FP1, FlexiBLE provided a "reliable benchmark for me, every time I ran into issues on Linux, I just kept comparing my results with FlexiBLE to differentiate between the firmware limitations." FP2 "loved the JSON because I could use it to generate new user interfaces without [application developer] input." Finally, FP3 praises the ease of

use in onboarding, stating, "It was pretty intuitive for me to take the magnet off the device, and then it showed up pretty easily in the app, and, you know, it paired itself; that part was not tedious at all." It is clear that FlexiBLE provided benefits to the use cases examined here, but participants also identified significant limitations, which we discuss in the next section.

4.7.3 Limitations and Remaining Challenges

A known limitation of FlexiBLE, as reflected in the interviews, is its dependence on Apple's ecosystem (iOS and macOS), specifically the CoreBluetooth framework. FP1 was tasked with essentially porting the underlying features of the FlexiBLE companion app to Linux to better explore the Rx/Tx characteristics of different antenna and amplifier hardware. FP2 reiterates this frustration: "If we wanted to use Bluetooth features that Mac didn't support, we were just out of luck." This is a common issue with BLE; many applications on an operating system compete for BLE access to a single hardware resource, and strict governors are in place to keep everything operational, including key user-experience devices such as keyboards, mice, and headphones. Writing cross-platform BLE code is challenging and requires custom implementations on different platforms. User interfaces are a similar problem with cross-platform development. While solutions exist, such as the SimpleBLE project¹⁷ and btleplug¹⁸ for BLE, or utilizing web-based UIs (Tauri¹⁹, Electron²⁰), they have their own limitations. Unfortunately, no simple solution exists here. Cross-platform BLE support is limited at both the application and operating system levels, despite BLE's ubiquity. Each platform has its own API to BLE (Apple: CoreBluetooth, Linux: BlueZ, Windows: Windows.Devices.Bluetooth, Android: android.bluetooth.le). This is an engineering and resources challenge, but undoubtedly possible. FP2 agrees that BLE is the best available choice; "in use, it's all Bluetooth, all Bluetooth, all the way"..

¹⁷<https://www.simpleble.org/>

¹⁸<https://github.com/deviceplug/btleplug>

¹⁹<https://tauri.app/>

²⁰<https://www.electronjs.org/>

The JSON device specification, while beneficial for firmware updates, caused practical headaches, primarily due to manual edits and limited error handling. FP2 attributes the difficulties to the lack of "type checking", "endianness" inconsistencies, errors in the companion application's error handling, and documentation. Type checking is a common problem with JSON in general. JSON Schema provides built-in structure and type checking, but it is not widely used. We experimented with a better editing experience with a device specification builder UI (Figure 4.2b), but it has not been deployed past proof-of-concept. There needs to be a better process for the handoff of specifications between the firmware and the companion application, including versioning. This is again an engineering problem, solvable with better tooling, but given the current issues, JSON as a transport format may not be the best choice. Protocol buffers (protobuf²¹) or similar may be a better choice for a more structured, type-checked alternative that can also be used in the BLE transport layer. It's clear that a declarative approach to this handoff is beneficial, but alternative implementation details to enhance the experience should be explored.

FlexiBLE faced key functionality limitations at two levels. At the protocol level, FlexiBLE underwent minimal optimization and was constrained by BLE's inherent limitations. PhysioHub, for example, required higher-precision timestamp resolution than BLE can support due to variability in transmission timing. The BLE packet structure was not as efficient as it could be: 'The way that the messages themselves were structured wasn't the most efficient' (FP2). Additionally, BLE security was never addressed in the system, mainly because it was not a requirement in early-stage development. At the application level, the UI had some quirks that participants mentioned. All three requested improved data plotting, and FP1 and FP2 requested debug logging from the peripheral device or BLE transmissions. FP2 and FP3 both reported issues with communication propagation and device connections, including slow connection intervals, which were actually asynchronous UI errors rather than issues with

²¹<https://github.com/protocolbuffers/protobuf>

BLE itself. The former represents inherent constraints of the BLE-based system; the latter reflects areas of engineering improvement.

FP2 raised a strong argument against FlexiBLE’s generalizability: ‘we sometimes had to shoehorn things into working with FlexiBLE, because it was designed to be so general purpose’; this refers to the need to quickly add features for specific use cases. This is a limitation of general-purpose systems, where one-off features cannot be integrated into the system architecture early without, usually, significant redesign. This makes it difficult to add a device-specific feature. Navigating this challenge requires a balance between flexibility and specificity, with a more tailored approach often necessary for certain use cases. This is where extensibility workflows could come into play: for example, using the general-purpose FlexiBLE companion app (client) for simple projects, while retaining the ability to port and build upon FlexiBLE features in a custom client. This is similar to what was accomplished with PhysioHub: rather than extending FlexiBLE itself, we built features such as custom time synchronization (leader and follower roles) into a separate, device-specific client that used a large portion of the companion app’s codebase.

FlexiBLE provides clear workflow changes, arguably benefiting the overall development process. However, more work is required to quantify this claim. Overall, the system’s limitations are not due to implementation failures but rather findings that can guide future iterations on similar platforms such as FlexiBLE. In chapter 7, we will explore these findings and synthesize developmental guidelines for future systems.

4.8 Moving Beyond FlexiBLE

This chapter introduced FlexiBLE, a declarative, adaptive platform for embedded system development and evaluation. Since its development, it has enabled research through PhysioHub and exO2/HOBIT, two unique projects that demonstrate its value in real-world applications (RQ2 and RQ4). Through our participant interviews, the systems have reported

a 10x reduction in firmware parameterization time and enabled the use of fewer animals in in vivo experiments. Despite its success, FlexiBLE was still written in a silo, mostly maintained by a single developer, me. This is perhaps FlexiBLE's greatest limitation. This reflects a broader challenge across this dissertation; technically capable systems fail not from engineering shortcomings but from a lack of standards, reproducibility, and stakeholder engagement. FlexiBLE is no longer being actively developed and will likely serve only as a legacy proof of concept, but I hope future work will build on its ideas. For success in future platforms, long-term sustainability and community involvement are key and arguably more important than technical innovation.

CHAPTER 5

Barriers and Methods of Ecological Field Sensing

The material in this chapter pulls from two published works in which I am a lead or co-contributing author: *Community-Driven Mobile and Ubiquitous Computing* [20]¹, “*The Devil You Know*”: *Barriers and Opportunities for Co-Designing Microclimate Sensors, A Case Study of Manoomin* [193]². We use this material to explore the use of ecological field sensors from the perspectives of practitioners. The chapter uses data from our interview study [193]; codes are revisited to highlight our thesis.

5.1 Introduction

Communities worldwide face significant challenges in preserving their environment, driven by the urgency of climate change, a global crisis that causes extreme weather and disrupts ecosystems. Unfortunately, the impacts of these events are inequitable. Vulnerable, low-income, and marginalized communities have borne the brunt of climate change, and often do not have the infrastructure or capacity to address gaps. Communities have increasingly turned to mobile and ubiquitous technology to bridge gaps in capacity and resources to support their health and safety. For example, communities in Chatham County, Georgia, have deployed smart sea level sensors to provide early warning signs for evacuations and baseline data to address the long-term effects of sea level rise [194]. Native American Tribes in the Western Great Lakes continually confront accelerating climate-driven degradation of coastal wetlands, which destroys Manoomin (wild rice) and the ecosystems it supports. These tribes have deployed local sensor networks, coupled with remote sensing, to understand ecosystem

¹©2023 held by the owner/author(s). <https://doi.org/10.1145/3631588.3631590>

²licensed under CC-BY 4.0. ©2024 held by the owner/author(s). <https://doi.org/10.1145/3685695>

change and initiate short- and long-term interventions [195]. The current capabilities and accessibility of wireless sensing systems and supporting technologies are ubiquitous, but there are sociotechnical challenges to their proper and impactful implementation.

In this chapter, I shift the focus from wireless and mobile health to microclimate sensing. With sensing, both domains share similar technological challenges, including power constraints, wireless connectivity, and in-situ deployment requirements. However, there are also key differences that were not addressed in our previous mobile health applications, including remote and harsh environments, long-range wireless communication, and sociotechnical challenges such as multi-stakeholder partnerships and data sovereignty. While distinct, these challenges complement the design consideration of low-power wireless sensing as a whole. We first review related work on ecological microclimate monitoring. We then summarize findings from an interview study, presenting a field sensing lifecycle which highlights the challenges and methodologies in current ecological deployments.

5.2 Related Work

Over the last three decades, wireless sensor networks have offered an alternative to manual data collection. Many prior works have presented such systems, including deployments to monitor seabird nesting [196], tracking wildlife [197], measuring volcanic activity [198], wildfire detection [199], and agriculture productivity [200]. Notable surveys on agricultural sensor networks [201], water monitoring [202], air quality [203], and low-cost devices [204] describe how technological advancements reduce costs, thereby increasing capabilities and applicability. A 2021 survey of 248 "conservation technology users and developers" ranked networked sensors highly for their potential to advance conservation but low for their current performance [205], highlighting a gap in their implementation.

Urban environmental sensors have shown promise in large-scale deployments. Researchers deployed 150 nodes in the city of Chicago to gain fine-grained insights into urban environments

and to understand the capabilities of large, interconnected deployments of sensing nodes with high spatial and temporal resolution [206, 207]. In a subsequent Chicago-based project, Eclipse, which deployed low-cost air quality sensors, HCI researchers showed that without an equal balance among stakeholders, the city, the community, and researchers, deployments fail to produce credible and actionable data [208]. Urban sensing is a reliable testbed for studying networked sensing and community partnerships. However, remote ecological sensing inherits the same sociotechnical challenges while also introducing infrastructure and access constraints.

As we discussed in our literature review (section 2.3), the academic literature on low-power wireless ecological sensing is sparser than that for health applications, particularly in end-to-end evaluation through field deployments. This work is an effort to fill that gap by providing accounts of technology use in the field for environmental conservation. We focus on the technology’s usability and outcomes rather than the technical advancements themselves. In contrast to our literature review, many of the devices are commercial-off-the-shelf products.

5.3 Study Methodology

In this section, we present a 13-participant interview study with individuals working in environmental science, including graduate researchers, representatives of non-profit organizations, and staff from federal institutions. These interviews were originally conducted as a preliminary survey to develop a system for Manoomin conservation (Chapter 6).

In this study, we sought to understand the broad role embedded sensing systems play in environmental science from the people who use and integrate them in the field. The goal was to establish a primary source of field data to develop new sensors, specifically for Manoomin conservation, but also for broader applications. To this end, we conducted 11 semi-structured interviews with a diverse group of 13 experts, who use or integrate environmental sensing technology, as described in fig. 5.1.

	Area of Work/Research	Organization Type	Role	Years in Role	Types of Community Partnerships	Involved in Field Work	Lifecycle Stages**	Sensing Modalities	Collection Methods	Goals & Outcomes of Data Collection/Analysis
P1	Ecology (Fish) & Hydrology	Federal government	Graduate Student	1	N/A	Yes		Temperature & Depth Sensors, Cameras	Manual from sensor local storage, device upload	Water level & stream flow
P2	Ecology (Fish) & Hydrology	Federal government	Field Scientist	25+	N/A	Yes		Fish Tagging, Camera, Hydrological Sensing: temperature, depth, pressure, rain	Manual from sensor local storage, device upload	Public datasets and publications for hydrology
P3	Ecology (Primate)	Academic research & non-Profit	Graduate Student, Community outreach coordinator	5	Indigenous communities, local government, corporate partners	Yes		Audio Logging, environmental DNA, geospatial	Manual from sensor local storage	Species identification, species distribution
P4	Water Toxicology	academic research	Principal Investigator	2	Industry and local government	Yes		Water samples, state of the art mass spectrometer (in lab)	Manual collection from sensor local storage	Water contamination - what contaminants are affecting the microbiome
P5	Water Level and Flooding	Academic research & community outreach	Principal Investigator	Did not disclose	Local government, local businesses and residents	Yes		Ultrasonic for water surface monitoring, commercially available LoRaWAN gateways	Real time streaming via LoRaWAN to WiFi or cellular-enabled powered hubs	Water level for emergency monitoring (end-goal) - low cost, low maintenance, easy installation urban water sensor
P6	Ecology (Amphibian)	Academic research	Graduate Student	4	Local government, local NGOs	Yes		Temperature, humidity, soil moisture, wind speed, radiation, cameras, audio logging, satellite images	Manual collection from sensor local storage	Microclimate modeling for climate change prediction
P7	Conservation (Birds)	Charity organization & academic research	Researcher	<1	N/A	Yes, in prior role		Satellite images (hyperspectral), acoustics, cameras, environmental DNA, LiDAR, manual point counts	Manual observation or collection from sensor local storage (prior role)	Regional biodiversity indicators driving legislation for conservation
P8*	Conservation (Wildlife)	Tribal government	Manager: Wild Rice Program	30+	Tribal government, academic	Yes		Water & air quality, wildlife health, species distribution	Manual collection from sensor local storage	Wildlife management decisions, public health decisions
P9*	Climate Change Adaptation Planning	Tribal government	Manager: Environmental Response Division	Did not disclose	Tribal government, academic	Yes		Water & air quality, wildlife health, species distribution	Manual collection from sensor local storage	Wildlife management decisions, public health decisions
P10*	Conservation (Plants)	Tribal government	Manager of specific ecology program	30+	Tribal government, academic	Yes		Water & air quality, wildlife health, species distribution,	Manual collection from sensor local storage	Wildlife management decisions, public health decisions
P11	Machine Learning Application for Conservation Efforts	Academic & industry research	Researcher	~8	Academic and non-profit organizations	No		Wide variety (data aggregation from multiple projects)	N/A	Wildlife management decisions, public health decisions
P12	Public Data Archival	Federal government	Manager & Principal Investigator	10+	Indirectly via government funded projects	No		Wide variety (data aggregation from multiple projects)	N/A	Public datasets and informatics processes
P13	Conservation (Wildlife)	Non-Profit charity organization	Research Program Manager	~5	Conservation non-profits, companies, and researchers	No		N/A	N/A	State of Conservation Technology Survey

*participants interviewed together.
 **participant discussed first hand experience within this lifecycle stage during their interview.

Lifecycle Stages	Data Need Finding	Technology Selection	Deployment	Maintenance	Data Analysis
------------------	-------------------	----------------------	------------	-------------	---------------

Figure 5.1: Key factual information of the interview participants.

Our interviews focused on both individual experiences and the organizations' or research goals. We designed an interview guide (section B.4) to encourage open conversation while guiding the overall flow of the interviews. We did not address every question for every participant; it was at the interviewers' discretion to skip those that did not pertain to the particular context or to ask more detailed questions in areas of interest. The interview guide includes two main sections. At a high level, we aimed to capture the current state of participants' field research,

including their sensing goals, the technologies they use, and the challenges they encounter in the field, followed by an open-ended discussion of what they would like to accomplish with technology in their work.

Figure 5.1 provides a high-level account of our interview participants. 10 of 13 have direct involvement in environmental fieldwork in their current roles, and the others either have previous experience or indirect involvement. For participants who spent time in the field, we asked for specific details about their experiences, including the equipment they used, how sensors were configured or validated, how often they visited deployment sites, and the methods used for data collection and processing. For those with limited access to the field, typically in management or orchestration roles, we sought higher-level opinions based on prior or secondhand experience with field sensors, their data, and the toolkits and systems that aid data collection and analysis. Specifically, P11, P12, and P13 serve roles involving data aggregation and project management. Both perspectives are valuable to this work, as they provide insights into low-level integration and high-level data aggregation.

This study was reviewed by the Institutional Review Boards of Northwestern University and the Georgia Institute of Technology, both of which deemed the research to be exempt and low risk. The research team consisted of three graduate students and two principal investigators, all of whom are authors of the work published in the ACM Journal on Computing and Sustainable Societies [193]. All participants were recruited through email. Each interview was conducted over video conference, recorded, and transcribed by a team member not included in the interview. To analyze the interview data, we first performed a ground-up inductive coding process by categorizing the transcripts into stand-alone ideas and grouping them into emergent themes. After themes were identified, we performed a top-down deductive coding pass by iteratively reassigning the original stand-alone quotes into one, multiple, or none of these themes. For this thesis, the codes were reviewed and regrouped according to their relevance to the thesis and research questions.

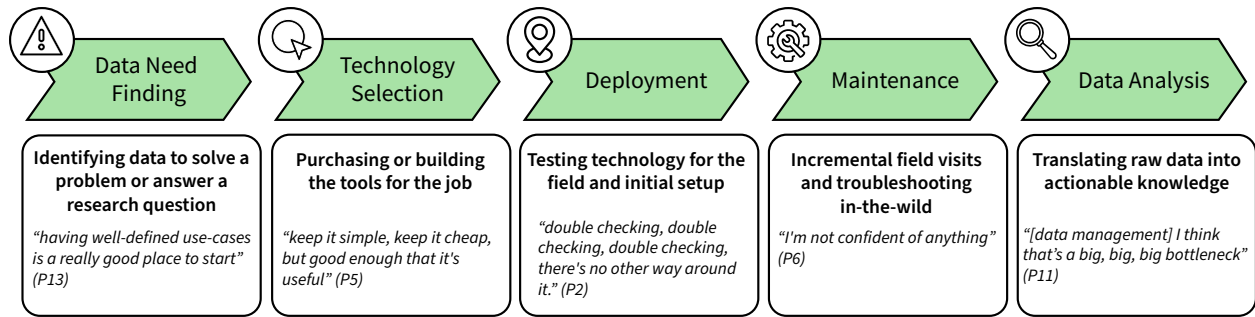


Figure 5.2: The Microclimate Sensor Lifecycle, highlighting the main high-level components of an end-to-end project, including quotes from our interview participants.

5.4 The Field Sensor Lifecycle

We identified distinct themes throughout the lifecycle of ecological field sensor use. In this section, we present these themes as an end-to-end lifecycle of in-situ field sensors, with supporting perspectives from those with firsthand experience at each step. We present these steps as a synthesis of experience from across disciplines, with a focus on their relevance to those interested in building new systems for field deployments. The lifecycle itself exemplifies the current methodology for wireless sensor-supported field ecology, and at each step, we discuss the challenges our participants face. Figure 5.2 gives an overview of the five steps in the lifecycle.

We define field sensors as in situ devices deployed in remote locations that do not rely on wired infrastructure and measure their local environment. These devices can collect data locally, requiring manual periodic or post-deployment extraction, or transmit data over wireless communication protocols (e.g., cellular, LoRaWAN). Generally, environmental sensors can accommodate higher power profiles than, for example, wearable sensors, but they still fall under the "low-power" category because they do not require an external power source. In our interviews, we also encountered the use of *remote sensor systems*, which make observational measurements, such as drones, manual point counting, and satellite imaging. Given the research focus of this thesis, we focus on in-situ stationary sensors; however, there may be

some overlap. This is discussed further in the inclusion/exclusion criteria for the systematic literature review (Section 2), which generally follows the categorization of systems here.

5.4.1 Data Need Finding

Data are often needed to support an urgent claim and to drive the allocation of resources to effect a change. P13, a product manager who aims to "improve development and adoption of tools for more effective conservation", emphasizes that "having well-defined use cases is a really good place to start" and highlights the importance of building a narrative around how data can make a difference. We found three examples of this in our interviews. P5 builds sensors for flood monitoring in low-income coastal communities to warn of structural damage to bridges. P1 and P2 study fish migrations, collecting hydrological data on small tributaries to correlate metrics to migratory patterns. Finally, P3 uses audio to track the presence of an endangered lemur species in an African microclimate. In these scenarios, data are used either as a direct indication of action (e.g., P5's flood warnings) or to build a narrative about the cause-and-effect of changes in the environment, migratory patterns, or species' presence, which can later be used to argue for change. Either way, an in situ monitor is a key tool for driving these goals.

However, in other cases, data collection is more exploratory and less directed toward a specific cause, adopting a broad approach to capture emergent trends, for example, those arising from a hypothesis. P7's work involves compiling "composite indices that aggregate biodiversity"; they are specifically interested in smaller regions where national-scale climate data cannot accurately capture ecological diversity. P8, P9, and P10, who were interviewed together, are concerned about recent ecological patterns, such as the collapse of a fish hatchery or the decline in the harvest of local wild plants. The cause of these problems is unknown but has been hypothesized, and it requires a broader collection of data points to build a qualitative narrative that can be presented to policymakers or used to request grant funding. Having data supporting a hypothesis is critical for resource allocation; however, collecting such data

can be particularly difficult for small organizations or underrepresented communities. P4 leverages its access and expertise in hydrological spectral analysis to assist local governments and community groups in identifying contaminants in the water supply, which can then be used to address the source, often industrial byproducts subject to higher-level regulations, e.g., state or federal.

In ecological sensing, identifying the *reason* to collect data is not the challenge; understanding the constraints of data collection can be. P2 hints at this: "anytime they always want more, so we have 15 minute [intervals] for where [a sensor] reads depth, if we could get every minute that would be even better" where *they* refers to superiors or stakeholders. P2 goes on to say that it's important to know one's "constraints" (or capabilities); this is where the challenges lie. The data needed is less about specific data points and more about goal alignment, which leads to impact. From a technical perspective, adding another sensor or increasing the frequency of an existing sensor may be well within the system's capabilities, but what is often overlooked is the cost in terms of reliability, longevity, maintenance, and timelines. These are the constraints P2 refers to, and when dealing with sensitive problems in resource-constrained communities, this becomes a sociotechnical problem rather than a core-technical one. In the following sections, we explore how data requirements shape the design of wireless sensing systems for ecology.

5.4.2 Technology Selection

The most frequently cited factor in selecting sensing technology is cost. The majority of our participants used commercial off-the-shelf (COTS) devices (P1, P2, P3, P6, P8, P9, P10) ranging from hobbyist to industrial-grade products. In some cases, community-supported open-source sensing products that emerged to fill a niche in the market were used (P3, P5, P6). Costs vary widely, ranging from a few hundred to "a quarter of a million dollars" for permanent installations, as described by P12, who manages data archival for a federal government agency. Purchase or deployment costs are not the only barrier; labor and maintenance costs are also

significant factors. P1 states that a specific water-quality sensor costs "\$15,000 to run every year". Given the constraints of a particular project, there is little room for trial and error when selecting devices. P13, who has a broad perspective from collaboration with various ecological organizations, attributes this to the nature of research grant funding structures. Projects have limited timeframes and strict, predetermined budgets, leaving field ecologists stuck with what they initially selected, resulting in an early trade-off analysis before all factors are known and affecting overall success.

Participants often preferred relatively low-cost mid-range products when selecting general-purpose sensing equipment. P1, P2, P6, and P9 mentioned using Onset's HOBO product line³. P6 also mentioned Lascar⁴, Tomst⁵, and Campbell Scientific⁶ as other common companies with wide-ranging products. Participants appreciate product quality assurance, software integration, and technical support, but note that these come at a cost. P6, a researcher of amphibian microclimates, notes the limitations of their work: "If you're using HOBO Pros, you'll be able to purchase five at most, which is not really scalable". This requires some to become creative with their planning or simply limit their scope to what can be afforded.

We found cases in which devices were repurposed outside their intended market. *camera traps* or *trail cams* are a strong example of this, which are a common choice for general sensing needs. Imaging provides a rich data source that can be processed to extract a wide variety of metrics. Automated, wireless cameras are accessible, easy to deploy, and easier to maintain than specialized equipment. Camera traps are generally marketed for hunting or wildlife tracking and are mass-produced, therefore inexpensive. P1, P2, P6, P7, P8, P9, P10, and P13 mention the use of these devices in some capacity. P1 discussed in detail the trade-offs of different capabilities, including on-device storage, the cost of automated cloud storage, timekeeping, and the ability to retrofit devices to meet specific needs. Cameras are a

³<https://www.onsetcomp.com/>

⁴<https://lascarelectronics.com/>

⁵<https://tomst.com/usa/>

⁶<https://www.campbellsci.com/>

convenient general-purpose data source; however, they introduce downstream issues, such as security and theft, which we will discuss in later lifecycle stages.

When a product meets the needs of a particular use case, it can lead to *vendor lock-in*, as participants indirectly describe. Vendor lock-in is a form of proprietary marketing in which companies intentionally or unintentionally create tension in a device's interoperability [209]. This incentivizes consumers to purchase products from the same brand and is a concern in applied IoT [154]. At first, when P6 begins by praising the capabilities of Onset HOBOT products, "[Onset] makes it fairly easy to have both my code [and] my software for curating the data work with my collaborators and work across our sites", also noting the convenience of cross-platform software for Onset apps. However, when P6 was asked whether proprietary software for device data collection and analysis limited their choice of tools, they responded, "Yes, so much so that it defines my research questions themselves." This is an epistemic issue that fundamentally limits the research.

The use of proprietary systems, especially when heavily dependent on a single vendor, can lead to further issues. P11, from a higher-level research management perspective, urges organizations to develop sustainability plans for their reliance on proprietary products; for example, if a company goes out of business and support is discontinued, or if a product is discontinued for marketing reasons. This independence might motivate a community-built solution, but unfortunately, there is a fundamental misalignment between industry and academia, with funding avenues that do not bridge the gap. P11 explains, "maintaining a tool that's being broadly used is very difficult to fund through grants", which leaves well-positioned research tools hard to maintain.

Regardless of funding constraints, some projects occupy the middle ground between commercial products and fully custom systems. These community-supported projects are built out of necessity and then gain widespread use within their respective communities. Our

participants mention several such projects, including AudioMoth⁷, Swift⁸, and FieldKit⁹. Community-funded and open projects are a more stable option than *in-house* development and provide better long-term technological resilience than proprietary (closed-source) products. Open source community-supported projects are great options when they fill a need, but can come with a trade-off, as P13 mentions, "having things be open source makes them super customizable, and then it also means that it's not a simple [as an off-the-shelf solution]". Customization leads to more "technical skill or training to use". In the end, if a tool doesn't exist, a commercial, community, or other organization must either settle for a reduced scope or build its own solutions.

Custom system development is less common in field ecology than, for example, mobile health; however, it does exist in practice. Through our grant partnerships, we learned about Spudnik¹⁰, an example of custom single-use sensing development. This Arduino-powered device, built at the Great Lakes Indian Fish and Wildlife Commission¹¹ (GLIFWC), measures air pressure, rainfall, water and air temperature, and water level. It was reported that Spudnik worked well, costing around \$200, but took about 40 hours to assemble and test, resulting in only a single deployment. Our participants also reported building their own internal devices. P1 used a Raspberry Pi Camera¹² with a "cellular modem, a big battery, and a solar panel" which uploads images every 15 minutes. While cheap, they express frustration in maintenance: "we have to baby it a little bit, [...] because the cellular and battery can go down." Tasks like these put burden on the field scientist, as P13 emphasizes: "conservationists are expected to do the work of like seven different professions, to be our own data scientists, solder our tools together, develop our own technologies." This is where system building requires the help of the computer science community: "we need expertise from other disciplines, there's interest

⁷<https://www.openacousticdevices.info/audiomoth>

⁸<https://www.birds.cornell.edu/ccb/swift/>

⁹<https://www.fieldkit.org/>

¹⁰<https://github.com/colemanjj/Spudnik-07>

¹¹<https://glifwc.org/>

¹²<https://www.raspberrypi.com/documentation/accessories/camera.html>

and excitement from you [computer science] guys and from people in the big tech industry who are really excited to contribute their skills. While the willingness to collaborate exists, in practice, the goals and incentives can differ."

P5 led the development of its coastal flood monitoring system, designed for simplicity, power efficiency, and low cost. However, to scale and have an impact, the system requires the support of "real administrators, not run by students that are graduating, [...] a production environment with all the expertise on security, certificates, firewall management and all those things, and a real patching routine, et cetera." What differs in the architecture of these systems from that you see in a computing lab setting is a focus on functionality over technical achievement. In this scenario, where existing technology is implemented in a new way, the impact is the novelty. P5 claims "keep it simple, keep it cheap, but good enough that [the system] is useful". In computing academia, the incentives differ; simplicity and stability are not strong motivators, and the technology itself typically needs to be novel. When working with novel technology, challenges arise when adapting to the field. P8 tells of firsthand experience of adapting a novel lake bed spring and preferential flow sensor, "Now it got a lot of buzz, so there were all kinds of folks willing to help out. Grad students were volunteering, people were doing free work, because that academic type evaluation hadn't debuted in an applied science arena yet." P8 then goes on to explain the refinements that are required in translating for application, "you figured out, oh, here's what the problems are in the applied arena, here's what the solutions are. So the next time we do it, it would be a much different game."

The costs and risks of adopting a new system can lead practitioners to prioritize convenience. Field scientists are burdened with learning new technologies in addition to their daily responsibilities. P13 illustrates, "it's a huge pain in the butt to switch systems and learn something new, [field ecologists] don't have [the] time." A tool needs to be demonstrably better to justify the commitment to adopt. P13 calls this a "hump you have to get over" in

order to replace the "shitty thing you're already using, [that] you already know how to use it, like the devil, you know, or whatever." The *Devil* here refers to a tool that is, to some extent, functional but not ideal. Above all else, the technical tools must fit within the capacity of those involved downstream in the lifecycle. Understanding the trade-offs and hidden labor is crucial to success and requires broad exposure to the applied domain.

5.4.3 Deployment

Deployment includes all steps required to install a device in the field. Tasks can include configuration, validation testing, and physical installation. We separate this step from maintenance in Section 5.4.4 because unique activities during deployment require additional commitment. Deployment is where reproducibility concerns (item RQ2) and development workflows (item RQ3) materialize in practice. Our participants identified this step as a pain point, as successful installations require extensive experience, often gained through trial and error, and knowledge of the tools and environment. P2 emphasizes that the only way to reduce deployment errors is repetition: "It's a lot of double checking, double checking, double checking, there's no other way around it."

For ecological sensing, travel to locations of interest can be burdensome and time-consuming. In some cases, our participants need to plan expeditions and have limited access in between trips. Access to sites can range from a few hours travel time to a month-long excursion. P3 and P6's research requires intercontinental travel and multi-day expeditions, which greatly limit the amount of equipment they can bring and make deployment errors extremely costly. P3 illustrates: "working our way down through all of those sites took like two months." However, even shorter distances take significant time. P1's camera traps are located "probably two hours [driving] away," and visiting all deployed cameras takes "two to three days, depending on how many I get done and what the weather is like." No matter the effort, field visits require planning around people's schedules and other responsibilities, ultimately costing money. Limiting and optimizing deployments reduces unscheduled visits and decreases labor

costs; for example, cutting deployment effort could allow the deployment of more devices, thereby enabling broader coverage and greater data.

Field scientists often pilot deployments to test a system's feasibility before installing it in the wild. These trials are particularly important when using new or adapted systems, but they are also useful for verifying baseline functionality. P13 states that wildlife monitoring devices are "often [used on low-risk species] like cattle to deploy a tracker first" before moving to target species. P1 emphasizes the need for pilot studies to evaluate device battery life, even when specifications are reported, noting a margin of error: "I will usually drop back a week or so from the shortest period the batteries work." There are sufficient unknowns and inconsistencies, even with off-the-shelf devices, that our participants do not feel confident deploying them without validating each one.

During deployment, a different type of validation is typically conducted, which we refer to as in-situ validation. This is done before physically leaving the deployment site and requires ensuring that everything is working as expected. Configuration issues can be very detrimental. P3 explains how "sometimes [configurations] just don't record", referring to recording times and sampling rates, "like, [the devices] were supposed to run from 5 a.m. to 8 p.m., but they ran continuously", which is very frustrating and avoidable with proper interfaces, "lots of super, super fun stuff." In most cases, in-situ validation requires additional equipment to communicate with the device or to collect ground-truth measurements for comparison or calibration. P1 details running a dummy fish tag through a deployed gate to verify that the fish-tracking system is working properly. P3 brings speakers and microphones to the field site to test the quality of audio recorder deployments.

It is common for systems to require on-site calibration and configuration. The preferred method for this interaction is companion mobile apps, which many systems offer; for example Onset's products, which allow local wireless communication over BLE and other deployment conveniences. P6 highlights: "You're able to program [Onset devices] in advance and then to

save battery, you can launch them right before deployment based upon like a magnet that activates it." Other systems require computer software and an internet connection, which requires additional equipment or makes in-situ validation impossible. Smartphones and laptops are ubiquitous; however, we found that some purchase lower-cost devices specifically for fieldwork to avoid damaging their primary devices. P2 recounts how they once dropped their phone in the river and now use an older one for fieldwork.

Another serious concern is device security. Systems are often left in the wild for long periods, and theft and damage from people, weather, or wildlife are common, as noted by P1, P2, P3, P4, P5, P6, and P7. Consequently, the deployment methodology is critical to the longevity of data collection. P1 points to a specific experience in which a trail camera was installed on a bridge, "in a security box [that] we Python lock to a tree", only later to be damaged and stolen by a passerby whose actions were caught on camera. P6 includes signage in the deployment locations' native language, acknowledging the type of research and providing details on the permissions they obtained to use the land, as "effectively a plea not to steal our equipment." Collectively, participants expressed a desire for security in the systems they deploy, whether through mechanical techniques, discrete or camouflaged enclosures, remote tracking capabilities, or a combination of these methods.

Successful deployments rely heavily on experiential knowledge that is rarely captured in publications or system documentation, making reproducibility challenging when scaling similar systems across organizations or disciplines. Success also involves much more than technical feasibility and relies on the robustness of systems in real-world environments. Academic projects often prioritize the former over the latter (section 2.3.3), a shift that may require community partnerships and time for systems to be truly tested in the wild.

5.4.4 Maintenance

Once deployed, devices require periodic visits for repairs, data collection, battery replacements, or a minimum retrieval. We call this step maintenance, which is the sustained burden of field sensing. This burden shapes what is viable in the long term and dictates sensing longevity. Scheduled maintenance is often required throughout deployments, but events of interest or sensor failures necessitate additional field trips, which can be costly and, in some cases, impossible to conduct.

Most participants adhere to a set maintenance schedule due to system technical limitations, primarily battery life and storage capacity: "probably the biggest issue we've had would be data storage and battery" (P2). P1 expresses frustration with the requirement for intermediary visits even with simple sensors: "I wish I could just take a temperature logger, put it out for six months, let it go for six months, come back and get it, I wish I didn't have to baby these so much." Challenge with devices requires P1 to visit the sensors "once a month," an arbitrary safety measure intended to mitigate potential data loss. Storage is seemingly a more controllable problem; we saw specific complaints regarding misconfigurations, as P1 again emphasizes, "you can have a setting wrong", which causes "the batteries [to] run down a lot quicker, and the SD card can fill up." P7 reported that a memory card was not programmed properly, necessitating another round trip to the sensor location to resolve the issue. These issues can be resolved through clear configuration workflows and interfaces.

Batteries, unfortunately, are unreliable for various reasons. Manufacturing can introduce inconsistencies without reliable sourcing, and the relationship between the battery's chemistry, environment, and usage patterns can strongly affect individual drain cycles and overall lifetime [210]. P5, whose system relies on real-time water-level monitoring, expresses frustration with batteries: "Batteries have been a huge pain." P5 details their initial selection of alkaline batteries, which have a high cost-to-watt ratio, and then switched to lithium, both of which have inconsistent failure rates in practice, which they attribute to "quality control" issues after

trying "a couple of different vendors and manufacturers." Additionally, batteries also exhibit temperature-related inconsistencies, with seasonal effects significantly affecting their longevity, as noted by both P2 and P5. To reduce failures, P5 started "deploying two [higher-capacity] batteries at a time instead of four because if I deploy four, there's a better chance one of them is gonna start out [bad]." Batteries are perhaps the squeaky wheel in system failures, but in reality, many components of sensing systems are sourced from different suppliers with varying quality assurance practices, which greatly affects longevity and reproducibility in the field. A faulty sensor, for example, may not render a system inoperable, whereas a battery will, making it a significant bottleneck and maintenance burden. System designs should prioritize bottleneck status (RQ3).

In general, participants agree that any information about a device between visits is useful, e.g., via networked communication, even if the data is limited. Network capabilities vary widely based on location. As mentioned in the previous section, P3 and P6 travel long distances for deployments with no network connectivity options, whereas P5 has near-instant access via LoRAWAN with internet gateways. P7, who has previous experience with bird counting, weighs the difference between manual counting and remote audio recorders: "Despite all the difficulties associated with going into the forest and counting birds, at least you come home at the end of the morning, and you can look at your piece of paper and you're like, I've got this data [...]. With the recorders, we were putting them out, and we would come back a week later and assuming that they hadn't been drowned or burned, then you collect the card, and you put it in, and you find out the battery died after two days." We find that in most cases of commercial sensors, networked connections were all or nothing, meaning either data was streaming completely, or nothing at all, but participants expressed the need for "middle ground" (P1) or "health check type things to figure out if we do need to send someone to go hike 10 kilometers" (P11). These "health checks" would be beneficial where high-bandwidth data transmission is not possible. In the end, everyone wants a "set it and forget it system"

with a "renewable energy source", and real-time data (P3), but in practice, it seems infeasible at scale.

Not all maintenance issues are technical. The physical environment introduces unpredictable failure modes that are only learned through experience. It becomes a constant trade-off between the technology and the environment; the longer the technology lasts, the more time the environment has to take hold. P5 adds that when batteries are working as expected, bugs nesting in the enclosure become the bottleneck: "only sort of real issue that I haven't solved yet is bugs, [...] so even if the battery lasts six or 12 months, we have to go clean it more often." Other examples include cyclones destroying trees on which the sensors were attached (P6), primates ripping off antennas (P3), damage from wildfires and flooding, being eaten by cows (P7), or simply being lost. Sensor loss and damage are always detrimental to research, but even more so without wireless connectivity, as field scientists are unaware of an issue until they return, sometimes months later, and may never know what happened.

Every visit to a deployment site, whether near or far, incurs time and monetary costs. Reducing visits, or at least keeping them predictable, broadens the capability of data collection. However, in practice, field ecologists are "not confident of anything". P6 describes "flying across the planet in hopes and prayers that whatever sensors you deployed last year are still there and are still functioning as you intended them to." Given the variability of the technology and the environment, unplanned maintenance is unavoidable, but understanding limitations and trade-offs can go a long way toward successful deployments. However, even well-maintained systems can exhibit data gaps, drift, or corrupted segments that only become apparent during data analysis.

5.4.5 Data Analysis

Translating data from sensing systems at a high level means making it useful to a broader audience, the final step in our life cycle. Data analysis does not always occur at the end

of a deployment; it can also be performed in real time, ideally with some automation. Analysis specifics, such as methods, algorithms, visualizations, etc., are use-case-specific but typically involve the general steps of validation, data cleaning, aggregation, and some form of publication. Data analysis or management can be a primary role (P4, P11, P12) or part of a field scientist's duties (P1, P2, P3, P5, P6, P7, P8, P9, P10). Either way, the goal is to make raw data actionable, but even in the best-case scenarios with well-defined collection modalities, interpreting the data can be a significant challenge.

Participants describe formatting and cleaning sensor data as tedious, especially for deployments involving a mixture of devices. P12 finds that overall "interoperability is a real challenge" when managing large, diverse datasets, with P11 resonating, "there is no way to fix the problem that different sensors have different formats, someone needs to have the skills [to integrate the data]." Interoperability problems tend to be specific, sometimes relating to proprietary formats and vendor lock-in (Section 5.4.2).

A common issue is time synchronization. P2 discusses issues with "time drift on cameras" and issues with initial configuration and daylight savings time. P1 explains being "absolutely terrified" of losing data when editing EXIF image metadata in the terminal to rectify timestamps. These processes could be automated, but in practice, this type of work is performed not by data scientists but by those who collect and care about the impact of the data. When these processes are completed manually, perhaps once a season, they are error-prone and require additional labor. These manual cleaning pipelines and heterogeneous data formats undermine reproducibility (item RQ2).

Beyond formatting, data must be checked through quality assurance/quality control (QA/QC) processes and, in many cases, prepared for public release. This includes ensuring that all data collected and cleaned matches the expected, known patterns; for example, diurnal patterns, weather, or data from other devices. This can, unfortunately, greatly delay the identification of nuanced sensor issues until the QA steps: "You don't really find out until you check after,

it's a super fun surprise" (P3). P11 shares an example that "if all your temperature sensors are accurate within five degrees instead of sub-degree," your downstream analysis is affected, and quality suffers. Finally, P2 expresses the labor involved in prepping data for public release, a requirement for their organization: "[preparation] that's almost like doing a paper in itself," noting that "the whole point is somebody can go in, grab your data, recreate your project, or your analysis."

In many cases, the sheer volume of data collected cannot be processed manually within a reasonable timeframe. P13 explains, "We have all this data, but it's often too late and too little, and we don't know how to process or analyze it." They later explain that "[data] doesn't end up feeding into management decision-making for a really long time or never." This shows that timely processing can be critical to making data collection worthwhile. P7 echoes a similar challenge with audio data: "Then you've collected a phenomenal amount of data, terabytes worth of data, ... [processing the data] is quite challenging, you certainly can't listen to it all." The functional impact of a system is only as good as the usefulness of its data; over-saturation can yield functionality comparable to that of a failed system. Even when the processes are clear, the resources may be insufficient to handle the volume of data. P7 quantifies this, stating that "[it] took [an assistant] four times the length of the recording to correctly annotate all the birds," and for a "massive tropical dawn choruses in a forest, it might be 10 times that." This is unsustainable and may necessitate reverting to manual observations.

Machine learning is often positioned as a solution for vast quantities of data; however, in practice, it introduces its own challenges. P11 works on machine learning solutions for ecological data but is skeptical of its current practical use and claims that data management is the biggest issue across organizations, stating, "Machine learning models are 10% of the problem, you have to have some way to get the data to your model." P11 also states that it is "a very dangerous game" to make claims about model accuracy with limited ecological data,

explaining that "usually the data sets we get from ecologists [come] with labels [that] are about 10% wrong," due to human error, which then propagates errors into the models. Additionally, these models are often custom, and development is expensive. P13 expresses frustration in that it's cheaper to hire an intern to manually categorize images than to hire an engineer to automate the process with machine learning. Manual processes may be bottlenecks in the analysis, but with resources already stretched thin and the goal of producing usable data, there may be no other option.

Not all data can and should be shared. Privacy is a significant concern when using camera traps on public land, where data are widely published by federal agencies. P1 and P2 address a major concern about images depicting individuals, including children. Thus, they are looking into machine learning solutions to automate the filtering, as it is too laborious with "thousands of photos." This creates additional work that yields no research results, increasing the field workers' workload. This labor should be a strong consideration when using high-fidelity data, such as camera and audio data.

Data Sovereignty is an additional concern for independent and sovereign entities, such as Tribal (Indigenous) Nations, that wish to keep their data internal, a practice that conflicts with many funding rules. This will be discussed more in Chapter 6. This is most concerning with data collected on the reservation but not reviewed by the Tribal Government: "there's no reason [for external entities] to share with us before you share with the public, so we're viewed as the public, and not as a sovereign nation" (P8). This makes collaboration difficult, as state agencies' data-sharing protocols require the public release of data. Land rights should be considered early in the process, when collecting and releasing data, as even if the intention is good, the data could be used to support actions that do not reflect the interests of underrepresented communities.

The life cycle reveals that data analysis is not simply a post hoc step but shapes and is shaped by all preceding stages. Issues, complexities, and shortcuts introduced upstream can increase

labor or reduce downstream impact, especially early in the technology selection process, where, for example, a camera may seem like an easy solution but requires significant manual labor in post-processing. Having a clear grasp of the project's end goals and expectations for the data's impact is important.

5.5 Building Systems from Insights

The goal of this interview study was to provide our team with a solid understanding of the state of ecological field sensors from those with firsthand experience with either the systems themselves or the data they produce. Our research team's primary expertise is in the technology itself (computer science, computer engineering, electrical engineering, data science); therefore, we lack the nuances of ecological field science at scale. The insights obtained here, along with our technical backgrounds, informed the development of a sensing system specifically tailored for Manoomin conservation guided by our Tribal partners, which is the subject of Chapter 6.

Similarities can be extracted and compared with challenges encountered in the development of wireless and mobile health systems, and generalized beyond that. Collaboration within the domain to which a technology is applied is mandatory for designing around its limitations. However, it is also the practitioner's responsibility to leverage cross-disciplinary knowledge when possible. We discuss the potential generalization of these findings in more detail in chapter 7.

CHAPTER 6

Co-Designing Low-Cost Sensors for Manoomin Conservation

In this chapter, we introduce and reflect on a sensing system designed for the conservation of *Manoomin*, the Ojibwe word for an Indigenous species of wild rice that grows in the Western Great Lakes region of North America. This chapter predominantly contains details from two published works in which I am the lead author and co-author, respectively: *Makak: Co-designing Environmental Sensors to Protect Manoomin (Wild Rice)*¹ [21], and *Buoy Meets World: Aquatic Sensing Insights from 169 Days of Low-Cost Wetland Buoy Deployments* [211]. The chapter traces the developmental iteration through two subsequent deployments of the Makak sensor, interweaving retrospectives that relate to my broader thesis. The work is in collaboration with our Tribal partners and related organizations as part of the STRONG Manoomin Collective².

6.1 Introduction

Wild rice is culturally, nutritionally, and ecologically significant for Native peoples in the Great Lakes region of North America. It has been harvested for thousands of years from the small lakes and slow-flowing streams in the wetlands surrounding the Great Lakes [212]. The Ojibwe (also known as Anishinaabeg, Ojibway, Ojibwa, and Chippewa) refer to this crop as Manoomin, translating to 'good berry' in Anishinaabemowin [213]. Moving forward, we refer to wild rice as Manoomin, since our partnerships are with Ojibwe communities; however, we

¹©2025 held by the owner/author(s), published under CC-BY-ND 4.0. <https://doi.org/10.1145/3715335.3735460>

²<https://www.manoom.in/>

acknowledge its significance for other Indigenous peoples, including the Dakotas, who call it Psíj.

The abundance of Manoomin has declined by nearly a third in the last 100 years [214], with more recent estimates of loss at 6-7% per year [215]. Once occupying most of Eastern North America [216], Manoomin is now mostly limited to Minnesota, northern Wisconsin, and Michigan in the United States and limited areas in Canada. The conservation and restoration of Manoomin have become critical challenges for the Ojibwe people and their allies. Manoomin is among the most culturally significant and vulnerable species in the Tribal Climate Impact Assessment [217, 218]. Generations of stewardship by Tribal National Resource Managers and Knowledge-Holders reveal a host of perturbations contributing to Manoomin’s decline, including pollutants such as mining-derived sulfate [219].

There are many active conservation efforts for Manoomin, led by intertribal organizations and Tribal Nation natural resource departments. However, the sparse distribution of Manoomin across hundreds to thousands of water bodies makes sensing challenging. Research-grade and industrial-grade equipment is expensive, and physical data collection across large areas is not feasible. Our partners have indicated a need for equipment that enables low-cost hyperlocal sensing, which is the goal of this work.

The collaborative approach (co-design, co-development) used in this work incorporates Western scientific methodologies with Indigenous methods and ways of knowing [220]. Knowledge about Manoomin, which informed the technical design, was shared with non-indigenous team members at gatherings supported by Ojibwe communities, often in person and using oral knowledge-sharing traditions. Through these relationships, we engaged with Elders and community members on observations, ambitions, and goals related to Manoomin and environmental policy. The work presented here is a compilation of our collaborative efforts, while also respecting oral traditions and Tribal sovereignty. For example, we do not share the specific locations of sensor deployments, publish data, or recount specific conversations

unless explicitly authorized to do so. Moving forward in this thesis, I limit the discussion to technical abstractions or generalizations outside the context of Indigenous research, specifically Ojibwe research. However, more information can be found in our published works, including co-authorship from the Ojibwe community [21].

The focus of this work is the development and deployment of a wireless sensing system for Manoomin. This system, *Makak* (Anishinaabemowin: container or basket), is designed to be low-cost and to address the widespread distribution of water bodies that are otherwise inaccessible for continuous data collection. To address where Chapter 5 distilled a five-stage field-sensor lifecycle from interview data, this chapter grounds each stage in a deployed system. It incorporates key considerations from our participants regarding Manoomin conservation. These connections are highlighted throughout this chapter. In addition, this chapter contributes to the broader thesis with the following:

- Co-design process with Ojibwe partners and intertribal organizations, grounding sensor design in community-articulated needs and Indigenous knowledge (RQ1).
- Two hardware iterations across two field deployments (29 devices, 8 partner organizations), documenting technical details including design decisions, tradeoffs, and remaining challenges (RQ3).
- Validation of low-cost sensor accuracy in remote wetland environments, demonstrating that hyperlocal wide-distribution coverage is a viable alternative to research-grade single-site monitoring (RQ3).
- Companion mobile interface for in-situ deployment and inspection, responding to interview-derived pain points from Chapter 5 (RQ3).
- Design recommendations for low-cost environmental sensing systems that contribute to the synthesis in chapter 7 (RQ4).

Figure 6.1: Overview of the Makak system from the pilot deployment (left) and scaled deployment (right),

First, we briefly present related work on Manoomin and co-design. We then discuss our methodologies and partnership details. Throughout this chapter, we refer to two Makak deployments: the *pilot* deployment in the summer of 2024 and the *scaled* deployment in the summer of 2025. We combine sections to cover both deployments, including system design and implementation, deployment methodologies, and results; however, we primarily focus on the most recent scaled iteration. We conclude the chapter with a retrospective discussion of how this effort generalizes to the broader thesis and to future work in this area.

6.2 Related Work

This work focuses on two research areas. First, Manoomin science relates to the specifics of North American wild rice (genus *Zizania*), its environment, and its decline. Second, our co-design methodologies, informed by the broad participatory design-to-asset-based design literature and the HCI literature focused on ecological technologies and Indigenous sovereignty, influenced our work. We omit works on wireless ecological sensing, as it is covered in depth in Chapter 5.

6.2.1 Manoomin Ecological Research & Conservation

Despite extensive research highlighting the importance of Manoomin to Indigenous Tribes, few documented examples of efforts are available [221]. The primary text for understanding Manoomin and its lifecycle is likely written by David et al. [219], led by an ecologist who worked for decades at the Great Lakes Indian Fish and Wildlife Commission (GLIFWC) and provided expertise on the topic. Most Manoomin research is published as technical reports by various agencies that inform conservation efforts. Hosterman et al. [222] describe and characterize the Lake Superior Manoomin ecosystem and its intersection with culture in a collaborative effort across many tribes and other organizations. Other contributions include workshop reports and summaries, in which participants reviewed recent data collections across government agencies and defined action items and needs [223]. Additionally, other mechanisms include descriptions of Manoomin datasets and data campaigns, which are available on the National Oceanic and Atmospheric Administration’s (NOAA) Digital Coast website for use by Tribes and Tribal-affiliated organizations [224]. Beyond these efforts, significant work describing how Manoomin interacts with the surrounding environment, development, and climate change is underway [225, 226]. Each of these works informs our study by providing context for our sensor design regarding the Manoomin ecosystem.

Another body of scientific research about Manoomin focuses on its cultivation and stems from prior extractive research practices [227]. Many recent initiatives to monitor Manoomin for conservation utilize remote sensing techniques [228], laboratory analyses of physical samples [229], and costly hardware [230], making it difficult to gather continuous, hyperlocalized data from hundreds of Manoomin waterbodies across large geographic areas. Additionally, previously deployed sensing devices were typically located on shorelines near Manoomin beds rather than within the beds themselves [230], thereby limiting data granularity and insight. Finally, a number of social-scientific studies have examined Manoomin harvesters (Tribal

and non-Tribal) to monitor harvesting practices over time [231]. These community surveys provide broader information on challenges related to Manoomin and inform our work.

6.2.2 Co-designing with Indigenous Partners

As with these prior efforts, we drew inspiration from asset-based design [232, 233], recognizing and celebrating the unique local knowledge and resources of our partner communities. The asset-based design mindset directs both the processes of design, deployment, data collection, and analysis, as well as the outcomes towards usable, impactful levers (i.e., policy briefs, resource management strategies, Tribal priorities).

This work shares common themes and approaches with technology design work in low- and middle-income countries. NkhukuProbe [234] developed sensing systems to support poultry farming in Malawi, which shares similarities to our methodology, including partner interviews, asset assessments, pilot deployments, and retrospective analyses to iteratively refine a useful tool. Similarly, [235] and [236] explore the challenges and opportunities of involving local partners—in these cases, farmers—in technology production and decision-making. Other related forms of asset-based design fall under the broader category of participatory design but focus on technology development for underrepresented communities.

Finally, we draw heavily on prior research conducted by those who worked closely with Tribal partners to develop technological and data-driven solutions. For example, the Full Circle Framework is highly relevant to our work in both approach and associated experience [237]. This work was conducted in northern New Mexico to extend last-mile Internet connectivity by deploying television white space (TVWS) wireless technologies. The goal was to provide Internet access to member communities, which included four sovereign Native Nations and several rural and semi-urban municipalities and counties. The Full Circle Framework is a “feminist participatory approach for discerning digital inequities through system design in communities with relatively autonomous cultural values and governance practices, including

Tribal sovereignty” [237]. The authors noted similar observations to ours that projects “move at the speed of trust” [238] and that projects must integrate into Tribal government workflows, be responsive to the needs and responsibilities of overworked partners, and be deliberate about data and privacy.

Similarly, our work is grounded in prior research on *Indigenous data governance* [239] and *Indigenous data sovereignty* [240, 241], principles crafted by Indigenous communities and non-Indigenous allies. These principles have been primarily developed to address the historical exploitation of Indigenous data [241] and to ensure that Indigenous communities can protect sacred data, which is often at odds with research initiatives for open data [242]. Thus, researchers, organizations, and Indigenous communities have collaborated to develop principles such as CARE (Collective Benefit, Authority to Control, Responsibility, and Ethics) [242] to provide data sovereignty and governance for Indigenous Peoples. We draw deeply from these and other principles in our work and continue to reflect on how our experiences may contribute to CARE and other principles and frameworks in the future.

Each of these prior works, despite the varied technological interventions, informs our approach. We differentiate our work from these by our specific technological intervention (buoy-based environmental sensors) and by our co-design of the sensing hardware, deployment locations, and data visualizations.

6.3 Requirements, Goals, & Co-Design

Acquiring continuous, targeted, and quantitative data that explain the decline of Manoomin is a key component of Tribal Nations advancing policy change with state and federal government entities, who are co-managers of ceded territories. Until recently, many settler government institutions have failed to recognize Tribal co-management and sovereignty, despite legally binding treaties that guarantee such protections. These data-informed policies are integral to

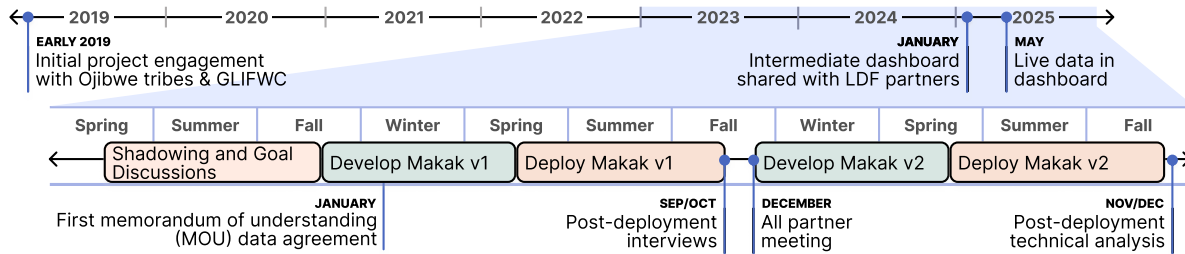


Figure 6.2: A brief timeline of co-design engagement and Makak development.

protecting Tribal lands from the lasting negative impacts of external human and environmental factors and to asserting sovereignty.

Although commercial-grade sensors provide high-accuracy data to support Manoomin research, Tribal sovereignty, and conservation management, their cost can be in the tens of thousands of dollars, which is prohibitive for most Tribal communities. The cost is particularly high given the spatial density required for Manoomin data collection, given the thousands of lakes and rivers in the study area. Motivated by community needs, our system explicitly targets low-cost, lower-quality, and widespread distribution of Manoomin sensing to supplement existing efforts and provide wider distribution of sensing.

We followed a co-design process for our system using a hybrid of Design Thinking [243] and Asset-Based design [244] approaches, centering on users' knowledge and experience to design a field system. The target user group comprises natural resource managers and other decision-makers within Tribal leadership structures. We spent the Spring and Summer of 2023 working with partners to understand their goals, which led to the system requirements. This knowledge and requirements gathering process was twofold: (1) conducting qualitative interviews to understand broader technological challenges in field ecology (Chapter 5); and (2) engaging with Manoomin knowledge holders through shadowing site-based work, talking with natural resource department leaders, and attending conferences. A targeted discussion of this hypothetical system was held at a Tribal Wild Rice Committee Meeting to engage the community in an open dialogue. The 16-person audience comprised staff from Tribal natural

resource departments, Tribal elders, and Wild Rice Chiefs. This discussion converged on a system to collect data on phenomena known to partners that diminish wild rice populations but have not yet been widely quantified. Specific phenomena include:

- Brown spot disease is related to humidity, precipitation, and abnormal wind gusts
- Herbivory, for example, from geese and swans that eat immature rice stalks, and abnormal migration patterns
- Invasive species:³ for hybrid cattails, American Lotus, Carp, Rice worms, and other forms of macrophyte and fauna.
- Algae blooms that can be caused by agricultural nutrient runoff
- Large boat wakes during the floating leaf lifecycle stage
- Water level and flow rate, which can change rapidly due to flooding and droughts, landscape modification such as beaver dams, and industrial activity such as cranberry farming;

Partners were interested in a sensor to measure these phenomena, which could inform mitigation strategies. In our initial system design, we targeted multi-modal environmental parameters which related to these phenomena of interest. In addition, partners provided input on the system’s aesthetics and data retrieval methods. This knowledge sharing was crucial to our understanding of the threats to Manoomin and current conservation efforts.

6.3.1 Deployment Goals

To promote an iterative design process, our pilot deployment incorporated a subset of the sensing modalities our partners desired. This included air temperature, water temperature,

³We utilize the term *invasive species* as it is present in Western scientific texts, but note that many of our Indigenous partners prefer to use the term “visiting relatives”

humidity, water depth, and inertial disturbances. In our second deployment, we iterated on the previous capabilities and added specialized water-quality metrics, including dissolved oxygen and conductivity. Our goals for these deployments focused on how well low-cost sensors fared in real-world settings to understand the feasibility of long-term deployments, a metric that is lacking in the literature. Our partners defined low-cost as under \$1,000; however, during development and these early deployments, the cost was funded by federal grants from the National Science Foundation⁴.

6.4 System Design

This section outlines the technical details of the Makak system. There were significant redesigns from the pilot deployment to our scaled deployment. We primarily focus on the latter, the current stable system design, and briefly reference the earlier designs to highlight technical decisions and lessons learned.

6.4.1 Mechanical Design

The pilot deployment was assembled from various off-the-shelf components: a square metal pegboard kept afloat by four plastic boat fenders, a waterproof general-purpose case housing the mainboard and electronics, and a separate Stevenson screen [245] for the above-water sensors (air temperature, humidity, atmospheric pressure). A chain attached the buoy to a mushroom boat anchor, which carried a custom 3D-printed housing with the depth-pressure sensor set in epoxy resin. Figure 6.1 shows the pilot design.

The pilot form factor demonstrated the feasibility of the buoy design but revealed several shortcomings across the eight retrieved devices. The pegboard-and-fender platform provided a large, flat surface and an open cavity beneath the pegboard, which invited wildlife interaction. Three devices had cables chewed or severed; we found evidence of nesting beneath at least one

⁴NSF Coastlines and Peoples Award: #2209226, NSF Smart and Connected Communities Award: #2233912

platform (suspected muskrats), and many of the devices were covered in bird and muskrat droppings. Additionally, the two-enclosure approach required multiple cable passages, each of which could be a potential point of water intrusion. The pegboard made field assembly and anchor attachment cumbersome, and the absence of on-board energy harvesting limited deployment duration to a single battery's lifetime. Finally, the sourcing of the mechanical components was relatively expensive due to a piecemeal bill of materials.

The scaled deployment redesign addresses problems with the pilot design by adopting a vertically integrated housing model (Figure 6.1). The platform is built around a floating PVC pipe (SCH 40, 2 in. diameter) frame that provides buoyancy while maintaining a smaller profile than the pegboard-and-fender setup. The buoyancy was reduced to sit lower in the water and provide less surface area for wildlife nesting and perching. A single IP67 waterproof junction box (QILIPSU, 28 cm $\times$ 20 cm $\times$ 13 cm) sits atop the PVC frame, attached with galvanized metal supports, and contains the core electronics and cable passage through waterproof cable glands. A custom 3D-printed Stevenson Screen mounts on top of the case and houses the above-water sensors (air, temperature, atmospheric pressure, and ambient light) and the solar panels. This design used fewer one-off purchased materials, with many available at local hardware stores or 3D-printed on common fused deposition modeling (FDM) printers.

The same anchor and attachment were used during the pilot deployment; however, in some cases, alternative anchors were used based on the partner's knowledge of the water body. A similar 3D-printed depth sensor, set in epoxy, is attached to the anchor, with a larger housing that also includes an ambient light sensor. An additional water-quality sensor is deployed in the water, slightly below the surface, using a 3D-printed jig.

Three 0.6 W solar panels are mounted on the Stevenson Screen printed at a 33° angle, oriented 120° apart in azimuth. Two design constraints shaped this configuration. First, the buoy is unmoored in rotation. The azimuthal distribution of the three panels ensures that at least

one is reasonably illuminated, regardless of the buoy’s drift. Second, our pilot retrievals show that large flat horizontal surfaces in wetland environments rapidly accumulate biofouling from birds and mammals, significantly degrading solar output. The tilt and clustering of the panels on a raised structure minimize the flat, accessible area and encourage rain runoff. The three panels are wired in series and routed internally into the junction box. The specific tilt angle was selected using the National Renewable Energy Laboratory (NREL) PVWatts calculator⁵.

In both iterations, the mechanical design reflects a core tension that recurs throughout this chapter. The buoy must be rugged enough to survive a full growing season, without maintenance, yet cheap and simple enough that partner organizations can assemble, deploy, and retrieve it without specialized equipment or training. The 2025 design did not fully resolve this tension, but provided an experience-driven iteration with better sourcing and potential scalability.

6.4.2 Core Electronics

The core electronics stack went through a complete redesign between the two deployments. Both were built around commercial-off-the-shelf IoT platforms that provided mainboards with unique capabilities. The switch was mainly motivated by the convenience of developing toolchains, cellular support, and community support.

Mainboard

The pilot Makak was built around Hardwario’s Chester-M⁶, an industrial IoT platform that integrates an nRF52833 SoC with LoRa and cellular modems onboard, power management, serial buses, and IMU and temperature sensors. The scaled deployment moved to a ParticleIO M-SOM module⁷, featuring the Realtek RTL8721DM SoC with an M.2 edge connector. The

⁵<https://pvwatts.nrel.gov/>

⁶<https://www.hardwario.com/chester>

⁷<https://docs.particle.io/reference/datasheets/m-series/msom-datasheet/>

M-SOM is mounted on ParticleIO’s Muon carrier board⁸, which integrates an LTE-M and LoRa modem along with an on-board power management system capable of solar charging. The full rationale for the switch, covering cost, SDK maturity, and cloud-feature tradeoffs, is discussed in Section 6.7.1.

The Muon carrier board features a 40-pin GPIO. We designed a custom PCB that integrates with the Muon’s breakout sensor interfaces for I2C², analog, and one-wire, an independently powered real-time clock (RTC), and an SD card for local data buffering. Sensor readings are written to the SD card between cellular uploads, protecting against data loss during extended connectivity outages common in our remote deployment locations.

Power

The pilot study used a single 7700 mAh 3.6 V Lithium Thionyl Chloride (Li-SOCl₂) "D"-sized primary cell. This cell is standard on the Chester-M platform and is selected for its wide operating temperature range (−55 to +85°C). Li-SOCl₂ is not rechargeable, so pilot devices were expected to operate on a single battery for the entire deployment window. While resilient, the Li-SOCl₂ batteries are relatively expensive, running \$25 each. Chester-M boards also include four 5 F/5.4 V supercapacitors to accommodate the high internal impedance of the Li-SOCl₂ cells.

The scaled system was configured with a 3250 mAh rechargeable LiPo cell and three 0.6 W solar panels. Panel output is conditioned per-panel via low-dropout diodes that feed the Muon’s on-board charge manage; this is to regulate the current into the battery. Since our deployments are during the summer months, the temperature ranges are less extreme and sufficient for LiPo batteries. In addition, LiPo batteries are cheaper and easier to source. The Muon power management requires a battery with temperature control (3-wire). We estimated the worst-case monthly energy budget of 18 Wh, computed with Particle’s battery

⁸<https://docs.particle.io/reference/datasheets/m-series/muon-datasheet/>

calculator⁹ against NREL’s PVWatts estimate of 10.2 Wh per panel for our latitude and tilt under a worst-case orientation. The three-panel configuration provided a nominal $\approx 5.2\times$ margin on paper, deemed sufficient to accommodate shading, biofouling, and cloud cover. This margin proved to be insufficient in practice for most devices. The failure analysis is presented in Section 6.7.3.

Connectivity

Both deployments use LTE-M as the primary uplink, a common technology for IoT with a low-power profile and a wide range. Both mainboards also support LoRaWAN and BLE. LoRaWAN is envisioned as a fallback to LTE-M, but was not deployed in either season because it requires additional gateway infrastructure; it remains an option for future sites with unstable LTE-M coverage. The coverage analysis of the scaled deployment, including the implications for Tribal lands, is presented in Section 6.7.2. BLE is used in both deployments as localized connectivity to companion mobile applications (Section 6.4.5)

6.4.3 Sensing Modalities

The sensing payload was selected through a co-design process described in Section 6.3, prioritizing modalities that partners identified as relevant to Manoomin health but lacking hyperlocal, continuous observation. This includes microclimatic conditions in the rice bed, changes in water level, thermal stratification, water-quality indicators, and external disturbances. The pilot targeted a conservative subset of these to validate the platform as a whole. The scale deployment extended the payload to ten modalities by adding water quality and ambient light.

⁹<https://docs.particle.io/tools/developer-tools/battery-calculator/>

Pilot (2024)

The pilot carried two external sensors and an on-board IMU. A Bosch BME280 mounted on the Stevenson Screen above the water measures air temperature, barometric pressure, and relative humidity. A TE Connectivity MS5803 mounted on the anchor measures the underwater temperature and the absolute water pressure. Relative water depth was derived as the difference between submerged and above-water pressure readings, using the BME280 air pressure as a dynamic reference. The Chester-M's built-in IMU recorded acceleration events on the floating platform, with the strongest frequency components extracted via an on-device Fast Fourier transform as a proxy for wave action and boat wake.

Scaled (2025)

The 2025 payload retained the pilot's sensors, including the pressure-pair differential water-level approach. It extended its capabilities by adding water-quality sensors in a mount sitting just below the water line and two ambient-light sensors.

The 3D printed Stevenson screen housed the BME280 (air temperature, barometric pressure, and relative humidity) and a Vishay Semiconductors VEML6030 ambient light sensor. The light sensor served as a low-cost proxy for incoming solar radiation.

The water-surface jig carries two Atlas Scientific probes interfaced through their EZO-family boards: an EC-kit-0.1 electrical conductivity sensor and an ENV-20-DOX dissolved oxygen sensor. In addition, a DS18B20 temperature probe was included to calibrate the surface water temperature and conductivity sensors. These three modalities were added at the partner's request following the pilot, targeting water-quality signals relevant to algae blooms, agricultural runoff, and thermal conditions at the canopy depth of early-season Manoomin.

The anchor fixture carries an MS5803 (water pressure and water temperature at depth) and a second VEML6030 (submerged light), enabling both the differential water-level metric

and a two-point vertical characterization of temperature and light attenuation through the water column. The data lead us to hypothesize that the VEML6030 may serve as a low-cost proxy for photosynthetically active radiation (PAR) and water clarity (turbidity), similar to a Secchi disk. However, validation of this hypothesis remains future work.

The IMU-based wave/wake sensing included in the pilot was used during scale deployment with an off-board IMU; it was not utilized. The pilot data FFT-derived signal proved too noisy to cleanly attribute to a specific physical event without controlled ground-truth data, which remained out of scope for the scaled deployment.

Table 6.1: Makak sensing modalities across the two deployments. Mount indicates the physical fixture on which each sensor is attached: **SS** = above-water Stevenson screen, **WS** = water-surface fixture, **A** = anchor fixture, **MB** = integrated to mainboard enclosure.

Modality	Sensor	Bus	Mount	Year
Air temperature	BME280	I ² C	SS	'24, '25
Air pressure	BME280	I ² C	SS	'24, '25
Relative humidity	BME280	I ² C	SS	'24, '25
Ambient light (air)	VEML6030	I ² C	SS	'25
Water temp. (surface)	DS18B20	OneWire	WS	'25
Conductivity	EC-Kit-0.1	UART	WS	'25
Dissolved oxygen	ENV-20-DOX	UART	WS	'25
Water pressure	MS5803-14BA	I ² C	A	'24, '25
Water temp. (depth)	MS5803-14BA	I ² C	A	'24, '25
Ambient light (sub.)	VEML6030	I ² C	A	'25
Motion (wave)	LIS2DH12	I ² C	MB	'24

6.4.4 Firmware

Both deployment implementations used similar firmware architectures, including a statement machine with periodic sensor readings and cellular uplink connections. However, both platforms were built with fundamentally different SDKs, toolchains, and embedded operating systems.

Pilot System

The pilot firmware was written in embedded C++ on the Zephyr RTOS using Hardwario’s Chester SDK. It scheduled periodic environmental sampling, accelerometer-triggered motion-frequency capture (for attempted wave detection), and 30-minute LTE-M uploads, with on-device buffering to mitigate connectivity gaps. Motion detection used an on-device FFT, retaining only the strongest frequency components per window to keep uplink payloads small. The water depth was computed differentially in post-processing. The pilot firmware also exposed a set of custom BLE GATT *data streams* in accordance with the FlexiBLE specification (Chapter 4), enabling the companion Android application to read live sensor data and update configurations on-site. The On-site verification of the sensors was a pain point identified in our preliminary interviews (Chapter 5).

Scaled System

The 2025 firmware was rewritten to support the ParticleIO platform. Particle provides its own runtime system based on the FreeRTOS¹⁰ kernel, Particle Device OS (v.6.2.1). Our build was targeted for the Particle M-SoM (M404) module on the Muon carrier board (with a parallel build variant for the M2 carrier used during development due to availability).

The core application is structured as an explicit state machine with five primary states: WAKE, IDLE, EXECUTE, SLEEP, and BLE. An additional FIRMWARE state was used to handle over-the-air (OTA) updates. From any state, pressing a button transitions the device into a bounded 10-minute BLE peripheral mode for on-site diagnostics and configuration, as in the pilot device.

The WAKE initializes peripherals, synchronizes the independently powered real-time clock (AB1805) with cloud time on the first cellular connection, and loads the active configuration and schedule. The IDLE state runs a time- and battery-aware scheduler that decides whether

¹⁰<https://www.freertos.org/>

to queue sensor measurements, cloud connections, cloud publishing, or GPS fix tasks. Tasks are dispatched by a priority-ordered task manager in EXECUTE, and once the queue drains, the devices enter SLEEP, which invokes an ultra-low-power mode with GPIO wake. Default intervals were 60-second environmental sampling and 120-second cellular publication windows, which were useful during deployment but were subsequently reduced. When the battery charge drops below 20%, the scheduler extends the sleep duration to the configured maximum, preserving capacity and allowing the battery to charge. These runtime intervals and thresholds were updated onsite via BLE or remotely via ParticleIO's cloud portal, using a feature called Ledger.

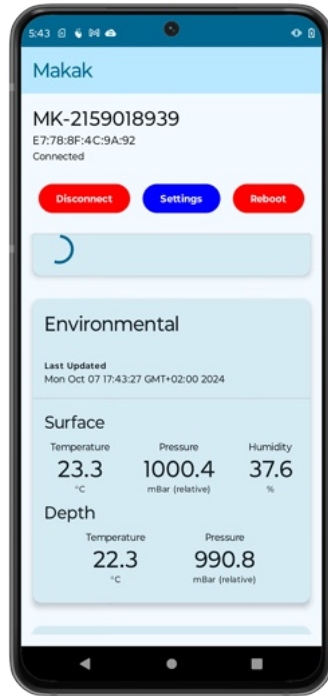
We group the sensors into an I²C multiplexer [PCA9547], with one channel dedicated to the Stevenson screen and water-surface sensors and a second to the submerged sensors mounted on the anchor. Atlas Scientific conductivity and dissolved oxygen probes are driven over UART via their EZO command set, while the DS18B20 water-temperature probe runs over a one-wire bus alongside the multiplexer. The power of the secondary sensor is controlled via dedicated GPIOs, allowing the entire sensor tree to be power-cycled on each wake.

Environmental readings are serialized as JSON and written to a separate file per sample on the SD card under `/env/`, with filenames keyed by timestamp. The publish task scans the directory on each cellular session, reads each file into a buffer, publishes via Particle's cloud abstraction layer, and deletes the source file only after the publish resolves successfully. The scheme tolerates multi-day cellular outages bounded only by the SD capacity.

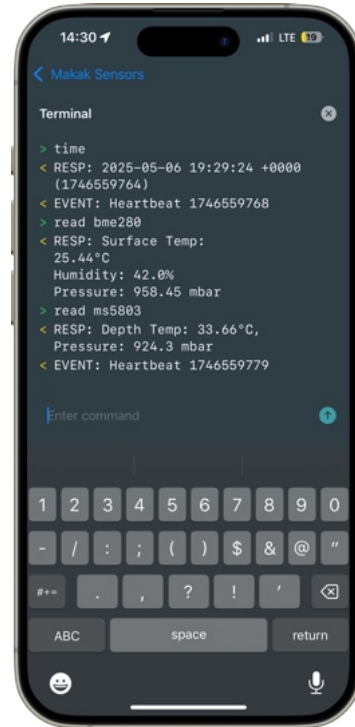
Runtime configuration is stored in a Particle Ledger that synchronizes bidirectionally between device and cloud on each connection, so remote parameter updates (sampling intervals, thresholds, feature flags) propagate without requiring an active connection. Firmware updates follow the default Particle Cloud OTA pattern with a bounded upload timeout. A BLE diagnostic GATT service based on the Nordic UART Service (NUS) profile exposes

a bidirectional text channel for on-site commands and live status, mirroring the pilot’s service-per-subsystem model in a single UART-style channel.

6.4.5 Software



(a) Pilot Android app.



(b) Scaled iOS app.

Figure 6.3: Companion smartphone applications used for on-site validation and configuration during Makak deployments: (a) Pilot Android companion application, showing the per-subsystem GATT view, and (b) Scaled iOS companion application, showing the command terminal used during 2025 deployments.

A simple companion smartphone application accompanied each firmware generation, used exclusively for on-site validation and configuration during deployment, maintenance, and retrieval visits. The role of these applications is to let a field scientist confirm, in a few minutes and without cellular reception, that a newly anchored device is sampling sensors correctly, reporting plausible values, and potentially calibrating sensors to a specific environment. This was done using BLE and by enabling a special firmware mode triggered by pressing a button on the mainboard/case. This was a direct ask and need identified in our preliminary

interviews (Chapter 5). LTE-M operates on a lower-bandwidth, wider-range configuration of the cellular network; the Makak device may be able to establish a reliable connection, whereas mobile phones will not, for example, reliably load web pages. In practice, these applications served as utility tools for our development team and as proofs of concept for future deployments.

The pilot Android application exposed each Makak subsystem (environment, wave, LTE status) as a separate BLE GATT service, mirroring the firmware’s per-subsystem architecture. The scaled iOS application unified all telemetry into a single Nordic UART Service (NUS) channel, wrapped in a structured message protocol that the app decodes into sensor panels, cellular diagnostics, a calibration interface, and a raw terminal for ad hoc commands. The shift from Android to iOS and from per-subsystem to single-channel BLE was driven more by convenience than by lessons learned. Our partners generally preferred Android as the target OS, but tight timelines and resource constraints in the scaled deployment, along with the development team’s stronger familiarity with iOS, led us to a simple solution not intended for external or long-term use. Both applications drew inspiration from or reused code from FlexiBLE (Chapter 4), thereby reducing development time. Figure 6.3 shows representative views of each application.

6.4.6 Cloud Data Collection

When LTE-M connections were established, all data was uploaded to cloud storage. For the pilot, Hardwario provided access to their beta cloud system, which allowed us to export data for analysis. For the scaled deployment, we used ParticleIO’s integrated service, which provided tools for fleet management, including data collection, configuration updates, and OTA firmware updates. ParticleIO did not store sensor data; instead, it passed it to the Noondawind platform [246], which stored it in Azure Cosmos DB and then displayed it in an ArcGIS-powered web portal for partners to view.

6.5 Deployment Methodology

In this section, we review the specific deployment details of both the pilot and scaled deployment. This includes collaboration with partners, as all deployments were accompanied and selected by partners.

6.5.1 Partners and Sites

The pilot deployment placed eight Makaks with four partner organizations: the Great Lakes Indian Fish and Wildlife Commission (GLIFWC), the 1854 Treaty Authority (1854TA¹¹), the Lac du Flambeau Band of Lake Superior Chippewa Indians (LDF, Anishinaabemowin: Waaswaaganing), and the Lac Courte Oreilles Band of Lake Superior Ojibwe (LCO, Anishinaabemowin: Odaawaa-zaaga'iganiing). The scaled deployment (2025) extended this to 21 Makaks adding 4 more partner organizations (8 total): the Bad River Tribe (BRT, Anishinaabemowin: Mashkiiziibii), the Wisconsin Department of Natural Resources (DNR), the Lake Superior National Estuarine Research Reserve (NERR), and The Nature Conservancy's Emiquon Preserve (NCEP). Deployments occurred from late April through early June, depending on weather and partner availability. The sites covered roughly 100 km² of wetland streams and lakes in northern Wisconsin and Minnesota, as shown in Figure 6.4. NCEP was an exception, located in central Illinois and unrelated to the Manoomin and Ojibwe partnership. This location was included due to an existing partnership, similar conservation efforts, and a shallow wetland environment with access to rich existing data sources for ground-truth comparison.

Site selection was partner-driven in both deployments. Specific locations and, in several cases, small alterations, such as anchoring method, visibility indicators, and wildlife deterrents, were selected by or in collaboration with partners. Decisions were made primarily based

¹¹abbreviations are unofficial

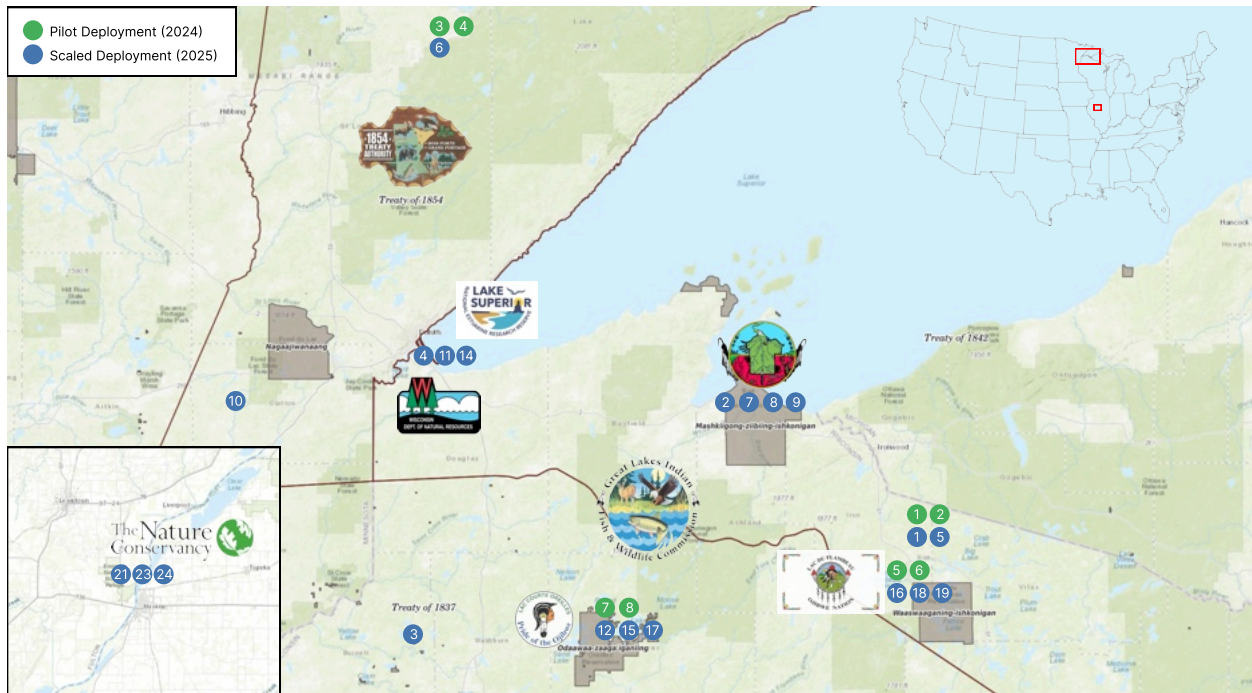


Figure 6.4: Makak deployment locations over the 2024 and 2025 seasons. Locations are approximate and shown with corresponding partner organization logos.

on water bodies of interest and, in cases requiring maintenance, on ease of access. During deployment, locations were adjusted for sunlight exposure and cellular connectivity.

6.5.2 Deployment Protocol

Before each deployment, we met with management staff at the respective organizations and gave a brief presentation on Makak, its sensing modalities, and the day's deployment plan. A representative accompanied the research team to each deployment site and provided water transportation when necessary. On each visit, we offered *Asemaa* (tobacco), as is customary in Ojibwe traditions of reciprocity with *Nabi* (water) [247].

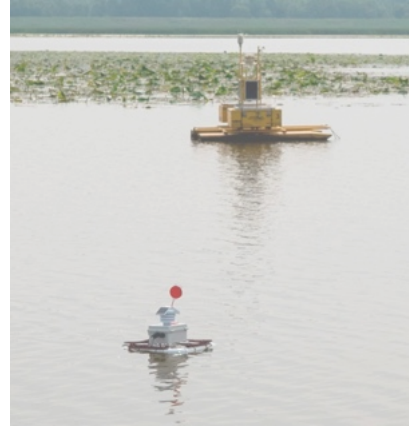
With the deployment of each Makak, we recorded initial sensor readings, network status, hardware identifiers, GPS coordinates, and ground truth water depth using a measuring stick or chain, along with unstructured field notes and photos (with approval). The length of the platform-to-anchor chain was adjusted to the current and the maximum seasonal water



(a) Onsite configuration



(b) Wildlife nesting



(c) Co-deployment with YSI Pisces

Figure 6.5: Field photographs from the Makak deployments: (a) on-site validation using the companion application (Photograph by David Cournoyer), (b) nesting observed beneath the platform during retrieval, and (c) a Makak co-deployed with a YSI EMM350 Pisces at the Emiquon Preserve near Havana, IL.

level reported by partners. Before leaving the site, we verified BLE connectivity and the live sensor values via the companion application. In the scaled deployment, which supported remote configuration updates, devices were initially configured with frequency sampling and cloud upload intervals of 1 minute (samples) and 5 minutes (uploads). Once we returned with computer access, we reduced the frequency.

6.5.3 Post-Deployment

The retrieval process was similar to the deployment. We met with partners, traveled to locations, and removed devices from the water. We recorded unstructured notes on the device state, location, and other details that we found helpful, along with photos. All devices were retrieved in September or October, depending on travel and scheduling, except for two devices in the scaled study: one was pulled early due to a malfunction and water intrusion, and another could not be found.

Each pilot retrieval visit concluded with an IRB-approved, on-site, semi-structured interview with one or more partner representatives, for a total of four interviews, each lasting 30 to 90

minutes, depending on the discussion. Each session includes an informal walkthrough of the pilot goals and data visualizations of data from the specific organization’s deployments. We then prompted partners on successes, challenges, and anticipated directions. The interviews were recorded with consent, transcribed using Otter AI via Zoom, and analyzed for themes that directly informed the scaled iteration.

One partner also proposed a cross-partner group meeting, an approach the team had not planned. We held this session online in December 2024 with four members of the research team and one to three representatives from each of the four pilot partners. The meeting allowed partners to echo and cross-reference themes from individual interviews, surface suggestions, and included a presentation of the proposed 2025 system, thus collecting collaborative feedback. The most consistent concern was data access and interpretability, which directly motivated the creation of a prototype dashboard for Makak data and informed the design of Noondawind, a centralized web portal for Indigenous data and environmental policy [246].

6.6 Findings

6.6.1 Pilot Results

The pilot validated the buoy concept and the approach to differential-pressure water levels while surfacing mechanical, power, and connectivity concerns that shaped the scaled design (item RQ3). Of the eight deployed devices, six collected usable environmental data. Makak 5 and 6, both on Tribal land where vendor coverage maps proved inaccurate, never established LTE-M connectivity, and were left in place for mechanical validation. Makak 8, also on the reservation, is only intermittently connected. Makaks 2 and 3 stopped recording after 18 days for reasons we could not determine in the field, but both shared water bodies with other reporting devices (Makaks 1 and 4, respectively), so we chose not to intervene midseason. During an unrelated planned trip in August, we located and replaced the battery on Makak

1, extending its deployment by an additional 72 days and providing an informal benchmark for single-cell SOCl_2 longevity in our duty cycle.

All eight buoys were retrieved. Three devices suffered cable damage consistent with wildlife interference. Partners attributed this primarily to muskrats, which aligned with nesting and excrement observed under several platforms (see Figure 6.5b). Bird droppings on most devices suggested that the flat platform was being used as a perch. Minnesota DNR staff independently attached pink ribbons and flags to the 1854TA devices to improve visibility to boaters, indicating that our deployed form factor was not sufficiently visible on open water.

Given the remote locations, data validation was difficult. For reporting devices, sensors track local weather patterns and diurnal cycles well. Two sets of devices (Makak 1-2 and 3-4) were located on the same water bodies. Both co-deployed pairs showed high correlation in surface and depth pressure ($p < 10^{-3}$). We compared Makak 4 with a reference water-level sensor deployed by 1854TA (Global Water Instrumentation WL16U-015), which showed a significant correlation ($p < 10^{-5}$, $r = 0.88$). Across all reporting devices, surface and depth temperature readings also correlated well ($p < 10^{-4}$), showing general weather trends. Humidity was less promising; the sensor often reached 100% relative humidity. We hypothesize that this is due to condensation in the Stevenson Screen.

6.6.2 Scaled Deployment

Across 21 devices over 2,951 combined sensor-days, 53.8% of days returned sensor data and 39.0% established cellular connectivity. All but one device was recovered; the latter was lost in a highly trafficked public waterway. Energy proved to be a common issue with the devices, with nearly all reducing their reporting rate or discontinuing data collection entirely before retrieval. Eight devices suffered water damage, four RTC failures, one failed due to poor condition, and eight operated throughout the season.

Table 6.2: Makak sensor validation against co-deployed reference instruments.

Makak	Metric	Quality ^a	N	Coverage	r_w	r_d
21	Air Pressure	++	722	23%	0.99	0.99
	Air Temp	++	722	23%	0.98	0.94
	Conductivity	o	1035	33%	0.39	0.69
	Dissolved O ₂	+	1166	37%	0.67	0.54
	Light	++	572	18%	0.90	0.95
	Rel Humidity	++	722	23%	0.95	0.95
	Water Temp	++	1216	39%	0.86	0.81
24	Air Pressure	++	762	24%	0.99	0.99
	Air Temp	++	762	24%	0.99	0.97
	Conductivity	–	1558	50%	0.24	0.21
	Dissolved O ₂	++	1609	51%	0.51	0.87
	Light	++	642	20%	0.92	0.97
	Rel Humidity	o	762	24%	0.37	0.98
	Water Temp	++	1678	53%	0.93	0.86
06	Water Level	++	142 ^b	97% ^b	0.85	n/a

^a Quality is the joint rating of windowed (r_w) and diurnal (r_d) Pearson correlation against the reference instrument: ++ good ($r_w, r_d \geq 0.8$); + moderate (≥ 0.5); o weak (≥ 0.3); – poor (< 0.3).

^b Daily aggregates rather than the higher-frequency cadence used for the other rows.

Sensor Accuracy

At the Emiquon location (The Nature Conservancy), we compared several sensors against a YSI EMM350 Pisces equipped with a Sonde (Figure 6.5c), co-deployed near Makaks 21 and 24 (Makak 23 was removed early due to water ingress). Air pressure, air temperature, light, and water temperature showed strong windowed and diurnal correlations on both Makaks (r_w and r_d at or above 0.8). Dissolved oxygen was comparable on Makak 24 but only moderate on Makak 21, which we attribute to Makak 21 being embedded in vegetation and exhibiting hyper-saturation. The diurnal peaks match the YSI sensor, indicating consistent and expected patterns (See Figure 6.6a). Relative humidity showed strong windowed correlation overall, but a weak signal on Makak 24, consistent with Stevenson-screen condensation saturating the sensor. Unfortunately, conductivity showed poor correlation, which we attribute to our placement just below the surface. The conductivity varies sharply with vertical position

and water flow, and the reference instruments were positioned near the lake or river bottom. However, we cannot rule out sensor or calibration issues.

As with the pilot deployment, Makak 6 was co-deployed with a WL16U water-level sensor, which is managed by the 1854 Treaty Authority. Figure 6.6b shows the average over 24 hours (the water level collected from WL16U once a day), tracked as a change in centimeters from the initial (baseline) water level. This shows consistent tracking; however, Makak can only track relative values, whereas WL16U reports absolute water-level elevation.

Overall, our low-cost system performed well given the limited ground-truth data from multi-month wetland deployments. Our goal is to provide trend data rather than high-quality data, enabling low-cost distribution to a broad audience. However, more work is required to remediate problematic sensors (e.g., conductivity), extend validation, and develop protocols for new sensing modalities. At the time of writing, we are planning the next steps of the Makak system including technical documentation that will allow continued development, education, and expanded use-cases.

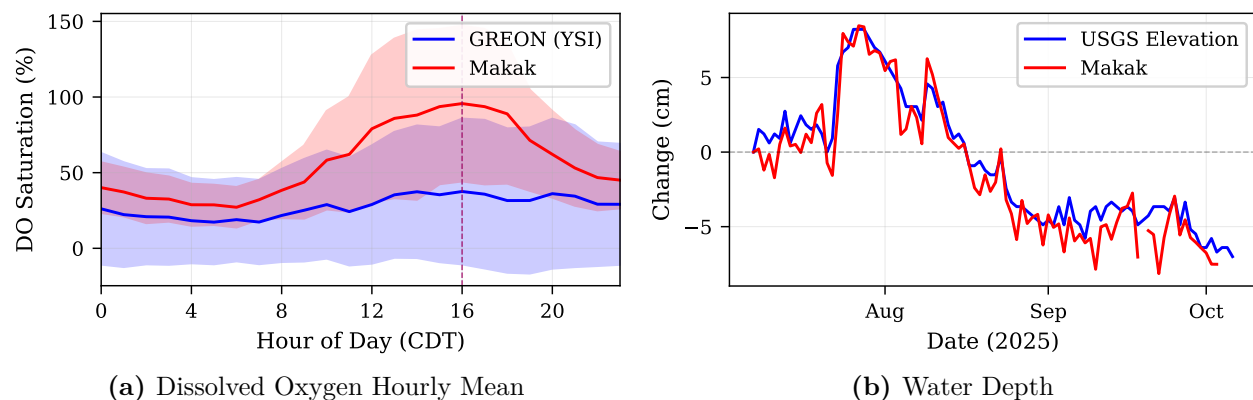


Figure 6.6: Sensor validation against external reference instruments. (a) Hourly mean diurnal cycle of dissolved oxygen saturation at Emiquon, vertical dotted line indicates diurnal maximum. (b) Daily change from start-of-deployment baseline at Rice Lake.

6.6.3 Pilot-to-Scaled Iteration

The pilot and scaled deployments together offer a longitudinal view of how a low-cost wetland sensing system iterates from prototype towards sustainable field use. Table 6.3 synthesizes the major changes from 2024 to 2025, pairing each pilot finding with a scaled response and observed result. Two relevant patterns emerge for RQ3. First, iterations that resolve pilot pain points but introduce new failure modes: firmware complexity that enabled remote remediation while introducing unexpected sensor failures, and the move to solar charging that lifted the deployment-duration ceiling, but left most devices undercharged. Second, iteration driven by team capacity rather than principled design, specifically the development of companion applications. Both patterns are characteristic of how lab-to-field development unfolds under academic timelines and small teams. The next section expands on the iterations that warrant deeper consideration for future generations and, more broadly, for low-cost embedded sensing.

Table 6.3: Major iterations from the pilot (2024) to scaled (2025) Makak deployments. Each row pairs a pilot finding with the scaled response and the observed outcome, including cases where the iteration introduced new failure modes.

Theme	Pilot finding (2024)	Scaled response (2025)	Outcome
Mechanical	Pegboard-and-fender platform with an open cavity beneath invited wildlife (chewed cables, nesting, droppings); two-enclosure design exposed multiple cable pass-throughs to water ingress.	Vertical PVC frame with smaller profile sitting lower on the water; single IP67 junction box with waterproof glands consolidating all electronics.	Less expensive and more accessible bill of materials; however, it requires assembly, no found wildlife disturbances, and less buoyancy, which may increase the chance of water ingress.
Mechanical	Off-the-shelf Stevenson Screen.	Custom 3D printed Stevenson Screen with integrated window for light sensor and solar panel mounting.	Less expensive at scale, but requires FMA printer and assembly time. Better integration and flexibility. Introduced a water leak in a few devices; redesign possible.
Power	Battery-only operation (single Li-SOCl ₂ primary cell); limited deployment time, and expensive (\$25).	3250 mAh rechargeable LiPo paired with three 0.6 W solar panels in a tilted azimuthal arrangement; nominal 5.2× energy margin on paper.	Solar output insufficient for 17 of 21 devices; daylight-hour bias developed within 2–5 weeks depending on canopy cover (Section 6.7.3).
Mainboard	Hardwario Chester-M (~\$250/unit) with limited cloud features but flexible firmware control and no vendor-cloud lock-in; SDK friction during development.	Particle Muon (~\$60/unit hardware + \$300/mo cellular subscription); mature SDK with OTA, Ledger config sync, and fleet management.	Per-unit cost lowered and development time decreased, but introduced cellular-subscription lock-in (Sections 6.7.1 and 6.7.2).
Sensing	Four modalities: air/water temperature and pressure, relative humidity, on-board IMU; on-device IMU FFT used as a proxy for wave and boat-wake events.	Ten modalities at partner request: water conductivity, dissolved oxygen, surface water temperature, two ambient light sensors (above water and submerged); IMU carried but not used.	Most modalities validated, with flagged exceptions (Section 6.6.2). FFT-based wave detection abandoned: signal too noisy without controlled ground truth.
Firmware	Fixed-cadence firmware on Zephyr RTOS using Hardwario’s Chester SDK; no remote configuration; per-subsystem GATT services.	Battery-aware adaptive scheduler on ParticleIO Device OS; OTA firmware updates and remote runtime configuration via Particle Ledger; unified Nordic UART (NUS) BLE channel.	Adaptive scheduling extended uptime under low solar; OTA enabled in-field bug remediation. Increased firmware complexity also introduced new failure modes Section 6.7.4).
Companion app	Android companion app with per-subsystem GATT, mirroring firmware architecture; partner-preferred operating system.	iOS companion app with single NUS channel and structured message protocol.	Iteration driven by team availability and timeline rather than a principled architectural lesson; both apps treated as proofs-of-concept and not validated for in-field usability with domain experts.

6.7 Retrospective

This section reflects on the scaled deployment to surface lessons that generalize beyond Makak. We highlight four tensions we encountered and the principles they suggest for similar systems.

6.7.1 Platform Choice and Development Velocity

The transition from Hardwario to ParticleIO during deployments illustrates a recurring inflection point for small teams building low-cost embedded systems. The trade-off between flexibility and development velocity often favors proprietary platforms when development resources are a binding constraint. Three factors drove our specific switch: cost, SDK maturity, and turnkey cloud features, each of which addresses a bottleneck in our team.

Cost and hardware accessibility were the primary drivers of the motivation to explore other options. Hardwario is a smaller company, where each development device costs \$250 and requires a direct invoice from the company, while the cloud service was only \$12/year/device under a promotional agreement. The ParticleIO hardware was much cheaper at around \$70; however, it required a more expensive LTE-M subscription (minimum \$300/month supporting up to 100 devices). For 30 units (the number purchased for the scaled deployment), we estimated Hardwario at \$7,500 and ParticleIO at \$3,000, including cloud subscriptions. ParticleIO's hardware was cheaper, easier to obtain, and faster.

Firmware development tensions were also a pain point for the pilot. Hardwario was built on ZephyrRTOS and provided an SDK and example firmware projects, in addition to plug-and-play firmware for the LTE-M modem. In practice, implementation required sufficient debugging and produced friction. The SDK was in active development by the Hardwario team, requiring frequent updates and direct contact. For example, an issue with switching between internal and external antennas in the LTE-modem firmware, which was not resolved

prior to deployment, required systematic debugging by our team. ParticleIO was not without its problems; we encountered some due to our use of the new MUON platform. The build tools and the connectivity abstraction layers were more turnkey and battle-tested, with a well-established support community¹². In practice, this allowed us to focus less on debugging connectivity and data transport and more on our sensors and low-power runtime.

Lastly, the cloud features and roadmap were more substantial with ParticleIO than Hardwario. ParticleIO supported LTE-M OTA firmware updates, runtime configuration syncing, network diagnostics, fleet management, and data integration pipelines, for example, Slack updates and AzureIoT integration, all out of the box. Hardwario's cloud interface was promising, but a work in progress, and it required more effort on our part to use and integrate. The tension is that ParticleIO's cloud platform is non-optional; all data must travel through their system, which can be considered a risk to Tribal sovereignty, whereas Hardwario's offers flexibility.

Ultimately, we traded complete flexibility for reduced complexity in networking firmware. However, both provide different types of flexibility: ParticleIO's deviceOS is open source, as is the HardwareIO ZephyrRTOS SDK (ZephyrRTOS itself is open source). ParticleIO's hardware is closed source including cellular provisioning, but provides a M.2 edge interface to build completely custom carrier boards. Hardwario's hardware is less flexible, but it supports daughter boards, extensive GPIO, fully open flashing for onboard Nordic modules, and the ability to use your own SIM card.

In an ideal world, Hardwario's flexibility would, on paper, be better aligned with Makak's philosophy, but given our small team, we were able to achieve greater functionality with ParticleIO's proprietary features at a lower cost. Each option has its tradeoffs, and we had to make a hard cutoff decision early in the planning between the pilot and scaled deployments, a common pivotal point in hardware development cycles (Section 5.4.2). ParticleIO provided

¹²<https://community.particle.io/>

a *proprietary Goldilocks zone* for our specific constraints but carries a sovereignty risk for our Tribal partners, as discussed in the next section and, more broadly, in chapter 7.

6.7.2 Cellular Infrastructure and Data Sovereignty

Cellular infrastructure in our deployment regions forms a chain of constraints that compresses the design space for small, tribally controlled embedded sensing. First, commercial coverage maps overestimate the actual reach of LTE-M on tribal and rural lands. Second, Verizon, a large network provider in the United States, requires specialized licensing for IoT roaming access, which significantly limits small deployments. Finally, proprietary cloud systems discussed in the previous section pose tradeoffs and risks to sovereignty. Here we discuss each and generalize for future projects.

Commercial coverage maps were poor predictors of actual connectivity, consistent with previous reports of overestimating LTE-M coverage on tribal lands. Of the 19 devices that ever had power for connection attempts, 15 connected on more than 90% of days they powered on during the first month, 4 between 50% and 90%, and 2 never connected at all. Most devices connect preferentially to AT&T towers, which are more abundant in the deployment region. From our analysis, the most generalizable finding is that distance matters: among devices that never connected, none had a tower within 4 km, and nearly every device with a tower within the range 4 km connected reliably throughout its first month.

The misrepresentation of coverage maps is a known problem in rural and Tribal areas [248]. CellWatch, an application developed by a group at the Georgia Institute of Technology, uses crowdsourcing to test practical cellular network coverage and automates reporting of discrepancies to the Federal Communications Commission (FCC)¹³ [249].

In the United States, a general IoT connection requires roaming across the three primary network providers: AT&T, T-Mobile, and Verizon. IoT providers often offer roaming SIM

¹³<https://play.google.com/store/apps/details?id=edu.gatech.cc.cellwatch>

cards to connect to the best available network; however, Verizon is often excluded due to licensing limitations. Therefore, both our carriers (ParticleIO and Onomondo¹⁴ for Hardwario) primarily used AT&T, with T-Mobile as a backup; however, some of our Tribal partners reported that Verizon had the best signal quality (others reported AT&T). ParticleIO and other providers do offer Verizon access, but require a contract engagement. ParticleIO estimated a \$10,000 minimum yearly contract for Verizon access. This may be feasible for an established large-scale deployment, but it is out of reach for our deployments. This further intensifies the challenges discussed in the previous section, in which small, tribally controlled deployments become difficult under enterprise pricing models, thereby affecting data sovereignty.

The carrier landscape narrows the set of platform options for small teams. ParticleIO does a good job of only serving as middleware for cellular connectivity. It does not provide permanent data persistence, requiring storage to be migrated to other services (Azure in our case). However, its pricing model requires a minimum commitment of \$300/month for up to 100 devices. This is a commonality that benefits centralized products and limits small deployments, for example, those managed by an individual Tribal Nation. As with all products and as noted by one of our participants in Section 5.4.2, there is always a risk that functionality and the agreement will change. For example, in January 2026, ParticleIO was acquired by Digi International; while nothing notable has changed yet, this poses risks to its existing services.

In practice, cellular infrastructure serves as a gatekeeper for our system’s capabilities. The carriers, coverage maps, and pricing structures that determine which devices can reach the network also determine which communities can build sensing systems. Alternative networking options, such as LoRaWAN, may alleviate this constraint but require additional development

¹⁴<https://onomondo.com/>

effort. The final chapter (chapter 7) treats infrastructure dependence as a first-class constraint rather than merely an operational concern.

6.7.3 Energy Budgets and Field Conditions

Our solar-harvesting configuration failed to deliver sufficient power during deployment on most devices. Our estimated energy budget was sufficient; aggregating to a metric (monthly Watt-hours) masked the actual operational constraint (an instantaneous voltage threshold under partial shading). Of the 21 devices, 17 produced insufficient solar output. Our design budgeted the total monthly energy against NREL's PVWatts estimate but did not account for voltage and current distributions. Outdoor testing under partial cloud cover yielded approximately 3.45 V from the panel configuration, below the 4.4 V minimum required by the charging circuit to deliver any current to the battery. Nearly every device reporting for more than four weeks developed a clear daylight-hour bias, with readings clustering around midday and disappearing overnight. Devices under dense tree canopy developed the bias within the first two weeks, compared with five or more weeks for devices in open water.

This can be remedied in future designs by using series-parallel wiring that preserves voltage under partial shading, along with additional validation testing. Series-parallel wiring requires a mechanical adjustment to support an even number of panels. To generalize these findings, we need to ensure that our estimates include an ample buffer to account for the operating environment's effects on the system. Our solar panels were subject to biofouling, canopy shading, and variations in weather and seasonal sun intensity, yet our estimate assumed sufficient peak sunlight to charge the battery.

6.7.4 Balancing Complexity and Deployment Scope

Increasing subsystem complexity and deployment scope simultaneously is a delicate balance, especially for small teams. Each new mechanism becomes a potential propagation path for failures elsewhere, and a larger deployment makes attribution at the individual unit

level proportionally more difficult. In hindsight, the scaled deployment added too much functionality and increased the number of devices by 2.5x.

Most subsystem failures in the scaled deployment were independent of the data, reflecting the isolation we designed into the firmware and hardware. However, a pair of correlations stood out: erroneous changes in sampling rate and failures of the water-surface temperature sensor (DS18B20) co-occurred within six days in 14 of 21 pairs of events, and the sensor eventually returned null or erroneous zeros in at least 67% of devices. We cannot conclusively establish the causal direction, but we suspect that the fragility of the one-wire protocol, combined with a power-supply issue with the RTC (AM1805), is responsible. It is a known issue with AM1805 (noted in the datasheet under "Oscillator Failure Detection") in which insufficient voltage can cause the device to appear nominal over I²C, but the oscillator does not advance the clock, so the same time persists. Not accounting for this caused errors in our scheduling system for sensor readings and sleep cycles. We assume that high-frequency reads did not allow sufficient conversion time for the DS18B20, resulting in erroneous values.

Detecting and attributing failure paths in remote deployments depends on the cloud platform infrastructure we adopt. ParticleIO's cloud system, with OTA firmware updates and configuration syncing, was helpful for debugging field errors. For example, we remediated a bug in which deleting environmental data from the SD card caused continuous duplicate uploads. However, others, such as the one mentioned above, were less straightforward and required analysis to identify. In hindsight, given our team size and limited field support, we overstepped our bounds by adding features to a new hardware platform, introducing more problems than we could adequately manage. Chapter 7 discusses this trade-off in more detail, but ultimately, issues such as these hinder progress toward a self-sufficient system for our partners and require more work to stabilize.

6.8 Future Directions

Given the nature of academic funding and graduate research timelines, Makak's future is uncertain. We are exploring avenues to make our insights and technologies more accessible to a wider audience. From buoy mechanics to low-cost water quality monitoring, We hope that Makak can serve as a foundation for new Manoomin conservation efforts with our partner organizations or expand to diverse water quality monitoring applications. As I transition to a teaching career, I plan to use Makak as a baseline for a curriculum in interdisciplinary embedded system design, or to support undergraduate research projects that extend the system.

CHAPTER 7

Learned Guidelines for the Development of Embedded Sensing Systems

This chapter presents a set of generalizable guidelines for developing low-power wireless embedded sensing systems in applied domains, contributing to RQ4. We present three pillars, two directly supporting application-agnostic technical scaffolding in wireless embedded sensing system development, **Architectural Patterns** and **a Method for Interoperability**, and a third which focuses on multidisciplinary collaborations, **Co-Development Practices**. In total, we present eleven components which we support with evidence from previous chapters. This work presents the following contributions:

- **Identification and naming** of recurring patterns, both technical scaffolding of system development and practices that ground systems in the real-world, drawn from first-hand development and the corpus (RQ4).
- **Components demonstrated in practice:** technical scaffolding validated in deployed systems, and co-development practices enacted in partnerships.
- **Components derived or extended from retrospective analysis** of our own work, where a limitation exposed the need.
- **Consolidation** into a coherent component catalog with explicit evidence mapping (fig. 7.1), position for applied embedded sensing.

Figure 7.1 maps each component to evidence from previous chapters. Evidence can be *positive*, a demonstration in prior work, or *negative*, a learned failure that shaped the component. Each

component is presented systematically: *Problem* (why it exists), *Finding* (the generalized insight), *Evidence* (grounding across our systems and literature, including where it fell short), *Enables* (what tooling or practice it unlocks). All components target the generalization goal of RQ4, but we included support for item RQ1, item RQ2, and item RQ3 throughout.

Component	Our systems					FS Study	Lit. Review	Research questions		
	FaceBit	FlexiBLE			Makak			RQ1	RQ2	RQ3
		FlexiBLE	Cell Oxy.	PhysioHub						
Architectural patterns										
Duty-cycled adaptive operation	○		●	●	●		○		✓	✓
Observability		●○	●○	●	●					✓
Parameterization	○	●	●	●	●	●○	○			✓
Real-time interfaces	●○	●	●	●	●○	●○	●			✓
A mechanism for interoperability										
Serializable transport envelope		●○			○	○	○		✓	
Self-description	○	●	●	●	○	○			✓	✓
Time		●		●	●	○			✓	
Co-development practices										
Application grounding	○		●		●		●○	✓		
Computational expertise as responsibility			●		●	●				✓
Reciprocity					●	●		✓		
Data stewardship		●	●		●	●				✓

Figure 7.1: Mapping of each guideline to its supporting evidence and the research questions it addresses. ● positive, built and demonstrated or supported by the literature; ○ negative, a problem exposed in practice or a gap in the literature; a cell may carry both. ✓ the component addresses the research question; all components address RQ4, omitted here.

The ecosystem of low-power wireless embedded sensing systems is too heterogeneous and fast-moving for tooling to lead the design and practice in applied science. Abstractions must outlast implementation. In the following sections, we present guidelines intended for technical enablers developing embedded sensing systems. We do not recommend or assume specific implementation details to keep these guidelines relevant beyond specific technical stacks. These guidelines focus first on computational components of development, then shift towards

sociotechnical themes. The goal is to provide insight for future system development and to motivate the development of formalized methodology and tooling that implements these guidelines in a generalized way for systems across broad applications.

These guidelines are not considered comprehensive, but come from direct learnings presented in the previous chapters. Most notably, we focus on computational infrastructure in the technical guidelines. There are undoubtedly electrical and mechanical engineering findings which could be synthesized; however, future analysis remains for those respective fields (chapter 1). The remainder of this chapter systematically presents each of the eleven guidelines under three distinct pillars. We conclude with a discussion of how these guidelines, together, can in practice lead to sustainable and reusable systems.

7.1 Architectural Patterns

Architectural patterns are system-level designs that dictate the high-level structure of wireless embedded sensing systems. These patterns are recurring, generalizable technical scaffolding we have identified as necessities for broad system function. Academic incentives award system infrastructure that is quick and lean; a path of least resistance to a proof-of-concept. Developing for impact, however, requires a level of *field resilience* — not the full technical hardening practiced in safety-critical applications [250–252], but a middle ground that enables real-world sensing to produce informative, actionable data.

As framing, we use Resilience Engineering (RE), a model for the development of complex adaptive systems [253], to guide our high-level components. RE was originally intended for safety-critical domains but has also been applied in lighter contexts [254, 255]. We use its components as a guide for iterative system development targeting field use. This model expects a resilient system to be capable of *learning*, *responding*, *monitoring*, and *anticipating*. The model is cross-cutting: the four patterns that follow are the scaffolding that supports systems which are *resilient*. We do not argue that the list is complete; we present each

component with justification, argue for its benefits in development and deployment, and identify generalizability for future tooling implementation.

Duty Cycled Adaptive Operation

Stateful firmware that is robust to power-cycles and adapts to its environment.

Problem. Embedded platforms provide varying scheduling mechanisms, superloops, cooperative schedules, or RTOS threads, but application state is left to the engineer to specify or sidestep. In practice, this leads to bespoke application state-machines or mismanaged scheduling, leaving the firmware inefficient, error-prone, or unpredictable. For internally powered devices, the state must also survive power cycles and adapt to a changing environment: batteries die, brownouts occur, and conditions shift outside anticipated bounds.

Finding. An application state machine is implicit in every embedded firmware; only its complexity, robustness, and reporting vary. Across our case studies and corpus, individual states and transition logic are application-specific, but the underlying capabilities generalize. State should persist across power cycles, for example, recovery management with watchdogs and external clocks. Additionally, conditional state transitions benefit from the sensed environment, for example, by adjusting sensing frequency during periods of consistency or by performing specific actions in response to an event.

Evidence. Makak conditioned its runtime on environmental and energy constraints: interrupt-driven motion capture, a low-power state below 20% battery, and persistent state across hours-long sleep cycles (chapter 6). There were also issues where a more robust application state would have helped handle failures more effectively (section 6.7). FaceBit checked mask wear at each wake cycle to conserve energy when not worn. However, its development encountered friction due to two firmware versions that could have been adapted to an adaptable state machine (section 3.6.1). PhysioHub, using FlexiBLE’s parameterization, managed state externally, allowing users to toggle sensor functions at runtime and enabling our experimen-

tation (section 4.5). Finally, our corpus generally lacks firmware descriptions and reusability: only 14% of papers provide detailed logic, and there is no consistency in reporting methods. Examples within the corpus for adaptive behavior exist, including changing behavior based on sensing anomalies [39, 256],

Enables. Standardizing this pattern enables tooling that makes application state explicit and declarative, replacing per-project re-implementation. An explicit, persisted state machine is the structure through which a system *responds* to its environment and *anticipates* known failure modes. Tooling can serve as a mechanism for achieving exhaustive transition coverage, detecting unreachable states, and generalizing persistence across power cycles, none of which is possible while the state remains implicit in the control flow. A consistent structure can help in documentation and reuse (RQ2) and reduce unknown states in the field (RQ3). Additionally, tooling can assist with visualizing state, enabling consistent reporting and comparison across applications.

Observability

System operation as first-class data; capture verbosely early, reduce as needed.

Problem. Wire debugging protocols are the ultimate development tool. They allow halting the microcontroller, inspecting and mutating memory, and verbose logging via print statements and traces. Wire protocols, however, tether devices to the lab and are unavailable in the field. In early system development, evaluation, and pilot deployments, details of system operation (e.g., state, errors, battery discharge, connectivity) are as important as the sensing data itself. Reporting this data is labor outside the firmware’s primary function and consumes already-constrained resources (power, bandwidth). Enterprise solutions exist (e.g., Memfault¹) but are priced beyond small research deployments and require proprietary tooling.

¹<https://memfault.com/>

Finding. System operation should be treated as first-class data alongside sensing data. The system should capture verbosely early, reducing as constraints demand: reduction is cheap (log levels, report intervals), but data never captured is unrecoverable. FlexiBLE framed its local database as a *lab notebook*, which generalizes the data storage mechanism by timestamping and recording every transaction, as one would in a ledger during experimentation. Most of the data will never be read; the point is that it exists when something goes wrong or in analyzing system performance.

Evidence. FlexiBLE implemented this component, which became helpful in its implementation, but lacked metrics. Both the cell oxygenator and PhysioHub utilized diagnostic data for analysis (sections 4.5 and 4.6). The cell oxygenator, however, experienced connection issues due to its subcutaneous deployment and, unfortunately, signal strength was not recorded, which would have been helpful in hindsight. This led to additional evaluations specifically for signal strength (section 4.7). Makak reported cellular connection data, which drove useful analysis of quality in remote locations (section 6.7.2). However, additional error logging could have potentially assisted in sensor failure analysis (section 6.7.4). In the corpus, we see that 68% ($N=84$) of coded papers evaluate only in lab-bound settings (benchtop, controlled, or demonstration), never the target context. It's unclear how often field observability is practiced; we assume it occurs rarely, with few documented examples of field logging [257], and no standardization for reporting.

Enables. Tooling that builds in abstraction for common operational reporting over wireless channels can turn this concept into useful data for transiting and iterating in the field (RQ3). This enables iteration on deployments without physical access, directly serving the *monitor* and *learn* capabilities of resilient systems. Reporting of data relies on the interoperability mechanism for transport (section 7.2): operational data rides the same envelope as sensing data, and lossless forward compatibility means even unparsed diagnostics are preserved.

Parameterization

System control variables as an interface.

Problem. Research firmware is dense with control variables, for example, sampling rates, thresholds, sensor configurations, and feature flags. Many of these begin as development-derived guesses and change with the development cycle. When hardcoded, these values require recompilation and reflashing to change, tethering devices to the bench. Interview participant FP2 (section 4.7) captures the cost: iterating on "the hundreds or thousands of different settings that are in the [firmware]" by re-flashing, this "changes how the device interacts with the physical environment, [and] you can't move around like you naturally would." Some IoT platforms, such as ParticleIO's ledger (section 6.4), provide this capability, but there is no consistency in the literature.

Finding. Exposing control variables as an interface fulfills the *input* half of the feedback loop in development and deployment (*output* is the next section). The parameterization method generalizes, but the parameters themselves are application-specific. Exposing, updating, validating, and persisting runtime parameters are application-agnostic technical scaffolding. Parameters also feed the conditional transitions of the application state machine (section 7.1).

Evidence. FaceBit maintained two firmware versions and a read-only interface (section 3.6.1). Rigidity in FaceBit's system abilities motivated FlexiBLE (chapter 4). FlexiBLE implemented parameterization alongside data collection (section 4.4.1). Retrospective interviews showed benefit in the development cycle (section 4.7). In practice, the cell oxygenator in vivo interface allows adjustment, reducing the number of animals required for experimentation and shortening development cycles (section 4.7), a feature uncommon in traditional early-stage biomedical systems. PhysioHub's operation depended on parameterization; firmware functionality was manual, guided by experimenters toggling sensors and control values

(section 4.5). Finally, Makak used parameters to adjust the frequency of sensor data collection in the field (section 6.6.2).

Enables. Parameterization requires a bi-directional communication channel from the device and some upstream software interface. The type of parameterization is application-specific, but the mechanism of reading and updating is generalizable. Having remote access to parameters enables responsive, adaptive firmware capabilities and true iteration in real-world environments (RQ3), helping with early prototyping to production deployments. Parameterization is a primitive for interoperability; updates are the canonical state-invocation that the mechanism must support (section 7.2).

Real-Time Interfaces

Feedback channels to a human; local and remote access.

Problem. Our systems relay data over wireless channels, but how that data is presented and used locally and remotely is inconsistent. Sometimes both remote and local interactions are required, as in the case of Makak, where cellular networks are sparse and cannot be used on-site. Without interfaces, confirming that devices are operating, configured, calibrated, and reporting plausible values is guesswork. Building interfaces, such as companion apps, onboard control, or remote cloud solutions, is substantial engineering work orthogonal to the research contribution, so prototypes skip them or improvise.

Finding. A real-time feedback channel to a human actor, independent or integrated with long-term persistence and data processing, is the *output* half of the feedback loop that parameterization opens. The justification and level of access differ by application. For example, field ecology needs deployment-time validation and remote monitoring, but the required technical scaffolding is shared across system implementations.

Evidence. On-site validation is a recurring theme in the environmental field-sensor interviews (chapter 5) and is what we termed *in-situ validation* (section 5.4.3). The preference is companion apps that use BLE onsite and provide clear validation of sensors and configuration. Makak implemented apps that were a critical component of the deployment but were not validated for usability with partners (section 6.4.5). FaceBit streamed raw, high-frequency data signals to gain insight into the development of sensor function (section 3.6). FlexiBLE generalized this through its declarative interface, allowing device specifications to drive what data is collected and how it is visualized. This provided real-time feedback on cell oxygenation in vivo experiments, where waiting until post-experiment surgical extraction to obtain data is the norm: participant FP3 stated "the FlexiBLE app let me control all of these simultaneously [...] I could always look and see in real time what the current was for all 18 of [the implanted devices]" (section 4.7). In the corpus, 29% ($N=36$) of papers report companion apps, while 32.1% ($N=34$) report no interface at all or leave it unspecified.

Enables. A real-time interface provides assurance during development and deployment (RQ3), but building the right interfaces for the application is key. Some applications can suffice with cloud-based only, while others need multiple (Makak). Real-time feedback reduces development cycles and reduces the burden of testing devices in the field. This completes the *input-output* loop with parameterization. Building general interfaces, modeling on FlexiBLE, where a single app can serve many devices, decouples firmware development from upstream app development. Section 7.2 explores how self-describing interfaces enable decoupling.

7.2 A Mechanism for Interoperability

The architectural patterns in the previous section assume data moves reliably between system layers. That movement is often considered an implementation detail and omitted from the literature. How data is structured, parsed, and ingested goes unreported, and it's assumed that serialization happens ad hoc. In our experience, especially with high sensor frequency

and precise timing requirements chapter 3 and section 4.5, the transport implementation can be a major effort and a source of hidden errors. Without proper documentation of this process, systems cannot be verified (RQ2). Solutions exist, but only at the individual layers of individual network stacks, and are inconsistent in practice (section C.1); none traverse local and networked protocols with a single, consistent structure.

A thin, simple *spanning layer* enables interoperability at scale [258]. The hourglass model of the Internet Protocol Suite is where heterogeneous applications interoperate by understanding only the narrow protocol between them. This precisely defines our goal: separating applications from physical devices through a mechanism that is transport-agnostic, self-describing, embedded-friendly, and extensible. The design is modeled primarily on FlexiBLE (chapter 4) and its retrospective lessons.

The three components that follow are not individually novel; they are well-established practices in networking and distributed systems, yet underused and underrepresented in applied embedded sensing research. This absence reflects the misaligned incentives and project structures in academic research technology, such as limited budgets, high turnover, tight timelines, and small teams. The contribution identifies these components as generalizable implementation opportunities that can enable shared technical scaffolding across projects.

Serializable Transport Envelope

Discriminator-based parsing with lossless forward compatibility.

Problem. The application layer of embedded sensing systems has no standardized, network-agnostic message structure. Parsing logic couples each firmware to its specific upstream application, so every device-application pairing reinvents message framing on an ad hoc basis. Network stacks are heterogeneous, including BLE, LoRa, cellular/WiFi, serial, and each protocol offers different primitives at different layers, so a structure tied to any one wire does not generalize. In our corpus, the transport *medium* is present in the large majority (83%,

$N=103$) of works, whereas the message *structure* is essentially never described, suggesting bespoke reinvention, hidden parsing logic, and unverifiable data pipelines (RQ2).

Finding. A top-level serializable envelope that is self-identifying via a discriminator and self-delimiting (by size or termination sequence) is all that is required to initiate layered parsing on any wire. The only assumption is a stable protocol supplying transmission and reception guarantees. This envelope is an established network practice, not a novelty: layered demultiplexing [259] and framing are already present in BLE HCI, UART, and 802.11. However, it is not standardized or practiced at the application level, i.e., the logic above the network stack. This structure supports *lossless forward compatibility*, meaning if an identifying header of the envelope exists, it can be stored regardless of whether the parsing logic can interpret its contents. If a parser encounters a substructure it does not recognize, it is preserved as raw text under an unknown delimiter; when the parsing logic is updated, it can be retroactively consumed. This follows Postel’s Robustness Principle from the original Transmission Control Protocol (TCP) documentation: "be liberal in what you accept" [260].

Evidence. FlexiBLE implemented a standard structure that allowed the firmware to send self-defined sensors and configuration values (section 4.4.1), but did not implement an encapsulation package. Packet parsing is dependent on BLE (section 4.7.3). FaceBit and Makak’s local data transmission, via their respective companion apps, followed what we typically see in the literature: one-off, solution-specific communication from the device to the application layer (sections 3.6.1 and 6.4.5). Makak’s cellular uploads to ParticleIO’s cloud served as a lossless ingestion buffer, where all data was collected as JSON structures but not parsed. The data was then forwarded to the Noondawind database for parsing (section 6.4). Our envelope extends FlexiBLE’s structure, modeled on Wireshark’s dissector API. Wireshark is an open-source network debugging tool widely used for packet inspection across many protocols. Wireshark’s dissectors conditionally walk packet structures based on discriminators and store unparsed remainders as unknown byte strings [261].

Enables. Following a wire-agnostic envelope permits writing a single parser that traverses heterogeneous stacks from embedded to cloud. Lossless forward compatibility solves problems with outdated parsing logic and new structure, as long as the envelope remains consistent, supporting flexible *observability* (section 7.1). This envelope serves as the substrate for the rest of the mechanism: typed payloads, metadata, time, and invocation are all contained within a single interoperable envelope.

Self-Description

Decoupling the firmware from the application layer.

Problem. End-to-end sensing systems are closed loops by default: a one-to-one mapping of device to client application, with parsing knowledge baked into both sides. Every change to the firmware interface requires a coordinated change to the application, and every new device requires developing yet another companion app. Consumer technology deliberately creates closed loops; vendor lock-in is data gatekeeping. FitBit, Oura, and many other wearable devices transmit over BLE but require cloud accounts. Tuya, a smart home system, encrypts local traffic and actively counters community workarounds. Research systems inherit the same coupling without the business rationale, and few tools or training exist for general-purpose collection [31]. Self-describing standards exist, including W3C WoT [262], OMA LwM2M, CoRE Link Format [263], and BLE GATT, which are valid options but come with trade-offs, as briefly discussed in section C.1.

Finding. Self-description is the decoupling primitive, long-proven by internet practices such as Representational State Transfer (REST) and Hypermedia as the engine of application state (HATEOAS) [264]: type a URL into a browser, and an interface appears without prior knowledge of the site. Self-description in this context is a mechanism that describes its capabilities, allowing a generic application to render and parse it in a similar way. Realizing this, over-constrained links roll up four core requirements:

Compact typed representation. Each value is carried in the smallest number of bytes its type and range allow (section C.1); the description must not cost the bandwidth it occupies.

Metadata out-of-band. Inline payloads carry coded references; a shared configuration or registry resolves them into a human-readable context. This benefits both constrained transport and the descriptive interfaces required by the system.

Stateful invocation. The ability to perform requests and responses with progress and context. For example, reading a parameter, then updating it, invoking a one-time calibration procedure, or performing time synchronization. These will maintain a directional discriminator (request or response), a sequence identifier, and any intermediate context needed for the specific task.

Extensibility through restraint. To support flexibility and specific functionality per application, the protocol must allow new structures to be created, for example, by exposing parameters and invokable actions with application-specific details. Unknown structures are persisted (lossless forward compatibility), allowing post-hoc recoverable functionality at application parsing layers. This addresses over-specification.

Evidence. FlexiBLE is a working demonstration of self-description that has shown benefits to the development process (section 4.7) and provided lessons for improvement (section 4.7.3). FlexiBLE’s device specification (JSON) allowed changes to a device’s firmware with the application adapting without updates. Compact payloads sustained 14.12 kB/s over BLE in PhysioHub (section 4.5), and time synchronization and parameter updates exercised stateful invocation in practice (section 4.6). FaceBit and each Makak deployment used its own one-off app, exactly the problem this component addresses (sections 3.4.4 and 6.4.5). Its cloud side uses CoAP [265], a self-describing standard, but through ParticleIO’s proprietary implementation (section 6.7.1).

Enables. Polished, application-specific companion apps are well-suited to specific workflows and productization, but building an app for every prototype is tedious and can be reduced with standardized tooling. Self-description can enable a one-to-many application that serves a

generalizable user interface (section 7.1) for any device. This will greatly reduce the software scaffolding required to test a device in the field or with users (RQ3) and promote reuse without requiring specialized software maintenance (RQ2). Additionally, the firmware can modify its functionality freely without waiting for app updates, thereby shortening development cycles.

Time

Defining, maintaining, and representing an authoritative time source.

Problem. Sensor data is inherently temporal: most is analyzed as a time series, and even non-series records, such as reference readings, calibrations, or triggered events, carry a timestamp. Ambiguity arises when wall-clock time, human-readable calendar time, is conflated with time as a unit of measurement. Intermittent, non-internet-connected systems make synchronization a challenge. Temporal fidelity requirements (precision, resolution, and accuracy) vary widely across applications and can adversely affect downstream analysis. As with data transport detail, time representation is largely omitted from the literature, yet in our experience, it is a substantial engineering burden and a source of error.

Finding In computer systems, including embedded systems, time must be stored as a count unit anchored to an authoritative source and with a known fidelity. This source can be relative (e.g., system uptime) but often requires absolute (epoch-anchored, wall-clock-referenced) values. Both authority and fidelity vary legitimately across applications, so an interoperable representation must carry them explicitly rather than assume them.

Evidence. We designed and demonstrated this model in FlexiBLE and extended it with PhysioHub. FlexiBLE timestamps every reading on-device with up to microsecond precision (configurable), using a synchronization procedure that links device uptime to calendar time (section 4.4.3); this enabled PhysioHub’s sub-millisecond node synchronization extension (section 4.5). FlexiBLE timekeeping was motivated by firmware-level variation observed during FaceBit’s development. We recreate an example of this based on findings from the

literature, as detailed in section C.2. Makak establishes the opposite end: minute-level fidelity sufficed, because seasonal clock drift did not matter given the hourly intervals between sensor readings (section 6.4). In the literature, TempMesh documents an "unknown delay" that misaligned field timestamps; remediation only recovered $\sim 40\%$ of affected records, leaving $\sim 12\%$ of all data misaligned [266]. We hypothesize that many such errors go unnoticed.

Enables. Standardization of timekeeping tooling, which enables flexible authority and structured representation, relieves a major burden on data structure and interoperability, and potentially removes hidden errors when timing is assumed (section C.2). Explicit fidelity gives downstream analysis the guarantees it currently assumes, and synchronization itself becomes a stateful action of the mechanism (section 7.2).

7.3 Co-Development Practices

The previous sections addressed technical scaffolding; here, we turn to sociotechnical guidelines that involve collaboration and determine the pathway of systems to the field. We argue for appropriate and timely engagement with those rooted in the application domain. From the technical implementers' perspective, the literature refers to these collaborators as *stakeholders*, *domain experts*, or *partners* who should contribute to development conceptually, technically, and in integration. The previous section framed a thin spanning layer for communication networks; here, we consider computer science itself as an increasingly common *spanning layer* in applied research, in which implementation enables interdisciplinary information sharing. Realizing that role depends on a sociotechnical practice that is often absent even from heavily applied computing contributions [20].

We focus on system applications that fall within *Pasteur's Quadrant* (chapter 1) [8], drawing on research that shifts the emphasis from technology to impact or demonstrated potential. We have argued that such work should be judged not by the novelty of its components but

by its integration and design for sustained impact [20]. Co-development is the tool to close that gap.

These practices are not unique to embedded sensing, but the breadth of work gives them practical weight. Many projects span the full arc from engineering a new device to evaluating it in a real-world setting; an ambitious, inherently multidisciplinary undertaking. That breadth creates both an opportunity and a responsibility to collaborate early with external partners across end-to-end development.

Drawing on our case studies, which offer contrasting perspectives on development collaboration in embedded sensing, and our literature corpus (chapter 2), we present four practices. We use *co-development* rather than the more common HCI term *co-design*, reflecting our components' more technical target. The practices follow the project life cycle, mirroring section 5.4: from grounding and engagement through partnership to sustainability for long-term use and impact.

Application Grounding

Motivating the technology solution.

Problem. Many projects claim real-world impact potential without the engagement to back it. We define *application grounding* in chapter 2 as a categorization of how academic works motivate applied system development; answering the question *why should this be built?*. Academic incentives favor convenient motivation (a path of least resistance), referred to as the Streetlight Effect [267–269]. It is more common to build on prior motivation or a gap in the literature than to engage in participation as motivation for technical intervention (section 2.3).

Finding The type of grounding varies widely, as we find in the corpus table 2.6. Each type maps to appropriate claims in the contribution; however, participatory engagement allows a

claim to have real-world impact, building the narrative for *fulfilling a need* rather than *filling a technological gap*. This sociotechnical aspect of early engagement enables the shaping of technology design to the real world, rather than proving the function of said technology. For systems to be co-developed, their reason to exist needs to be conceived through collaboration, a foundation of asset-based design [233, 244].

Evidence. In our corpus, 78% ($N=97$) of papers ground their motivation in the literature or in hypothetical scenarios, while 15% ($N=19$) report any form of external participation (chapter 2). Most notable end-to-end engagements include hyperlocalized air quality monitoring in a Slavic community [84], control of dams using water level sensing [77], and the design of a wearable cape to promote activity in South African Children [59]. These examples use participatory motivation leading to deployments within their respective communities. Our own systems show different levels of grounding methodology. FaceBit’s clinical needfinding was a single, non-iterative touchpoint, not a partnership (section 3.3.2); the cell oxygenators inverted the relationship, our team served as enablers for biomedical research (section 4.6); PhysioHub is a technical demonstration with light domain motivation (section 4.5); Makak grounded an end-to-end co-design and development process, including a prior qualitative study (chapters 5 and 6)

Enables. Grounding through engagement a precondition to co-development. A true path to real-world use starts with a social need, rather than a technological innovation. Both are valid contributions; however, to claim impact, we argue co-development enables a stronger connection to the applied application.

Computational Expertise as Responsibility

Managing authoritative knowledge and skill.

Problem. Multidisciplinary engagements are often presented as two roles, *domain partners* and *technical enablers*; however, in reality, expertise is more nuanced. It is the responsibility

of the technical expert to translate the domain needs into applied technical solutions; to do so, they must treat a domain expert’s knowledge as authoritative and set any technical bias aside [270]. It’s the responsibility of the technical enabler to fully understand the design space. However, the skills and experience of technical academic teams vary widely [15, 271] and have high turnover, a risk that is amplified when ongoing research depends on the artifact [272, 273]. [274] talks specifically to the mismatch in academic timelines and partner goals. When favoring an interesting or familiar technology over an existing or duller solution, technical bias can be disingenuous to the collaboration if it is not transparent to all parties [17].

Finding. Expertise flows in multiple directions; serving as the technical authority means being prepared, filling knowledge gaps when the application demands new background research, and holding technical preferences loosely [270]. This is where tooling that absorbs technical scaffolding plays a sociotechnical role. Less redundant infrastructure means resources can be spent on the bridging knowledge each collaboration actually requires rather than on rebuilding plumbing.

Evidence. In chapter 5, a qualitative study was designed to address our knowledge gap regarding the use of environmental field sensors. The Field Sensor Lifecycle (section 5.4) became the design vocabulary we carried into the development of Makak (chapter 6). Partners, several of them interview participants, defined the sensing modalities we targeted based on knowledge of Manoomin growth cycles and observed concerns. By contrast, FlexiBLE had a technical bias toward iOS/macOS, rooted in the team’s industry expertise. This led to fast, robust software development but forced the biomedical labs using the system to buy Apple hardware (section 4.7.3).

Enables. This framing shifts the technology design from a technology contribution to its impact on the target context. This is an entry position for reciprocity (section 7.3), where relationships align around mutually beneficial work. This enables an ethical lens for

scaffolding-reducing tooling, not just development efficiency. Less time spent on redundant development allows effort to be spent on domain specifics.

Reciprocity

Aligning goals in partnerships.

Problem. Every multidisciplinary collaboration is a means to an end for each party, or in some cases, a means to an end for one and a burden for another. Without deliberate reciprocity, the exchange skews. Imbalanced power dynamics are well documented, with the extreme providing no benefits to the target context, i.e., *helicopter research* [28]. However, the subtle failure is more common: claiming a benefit and never delivering or measuring it, the overselling that application grounding warns against (section 7.3). Collaboration is rarely free, even if it is unpaid. Efforts often impose a hidden cost, the undervalued, invisible work (chapter 1 and section 5.4) [25, 27, 275], that partners absorb on top of their own responsibilities. Makak costs our partners no money, but it costs time, site access, and attention.

Finding. Goals should be transparent and at least partially aligned, and the giving should be made part of the routine rather than an afterthought. Reciprocity is bidirectional and cross-disciplinary; credit and benefit flow both ways. However, hidden labor cannot be fully eliminated. The goal is to utilize system architecture to reduce where possible, and expose and respect what remains. Technical scaffolding does the work here; less bespoke integration means less labor imposed on partners in real-world use. However, social labor is not a tooling problem; the obligation is to acknowledge and respect it, not to engineer it away.

Evidence. In the Manoomin conservation work, publication was an explicit, jointly held goal; it was expected by the grant and tied to graduate students' timelines. Collaboration on publications at computing venues has been a learning process that is still ongoing, but we have sought community input and co-authorship while being respectful of the additional

time commitment to partners. We also enacted reciprocity by respecting and participating in our partners’ practices and protocols, rather than relying exclusively on Western scientific practices (section 6.5.2). The collaborative engagements with the cell oxygenator (section 4.6) were of a different nature; we served as technical enablers, facilitating new research in biomedical settings. Our corpus, being exclusive to ACM, is limited (section 2.5) and contains only computing contributions through a practical-motivation lens; targeting multidisciplinary venues is a option, but can have different effects on academic reward [276]. Generally, incentives for research run counter to reciprocal collaborations [277].

Enables. True partnerships with transparency and equitable contributions are a precondition for the data and sustainability practices to follow. Working toward common goals while understanding individual obligations builds durable partnerships.

Data Stewardship

Partners own and can access their data.

Problem. Data is the product and central focus of our systems; however, mismanagement can lead to harmful uses of the data collected from the community, intentional or unintentional. This is where a failure in reciprocity can lead to harmful impact. Data can be used for external benefit, providing nothing of value to the target context (e.g., helicopter research), or it can be exposed in ways that cause harm, a real concern in Indigenous partnership ([21, 239]). The technical stack can compound this risk by gatekeeping data (section 7.2), and the infrastructure itself can compromise control.

Finding. *Data Stewardship* is the practice of building a pathway for responsible data use that aligns all parties. Partners, those intended to use the system, own their data. Options range from Data Sovereignty, CARE principles, and legally binding MoUs [240, 242] to lighter agreements on ownership, access, and acceptable use by any partner. Data stewardship is important, but it is not typical engagement across a wide range of multidisciplinary projects.

In practice, stewardship depends on the technical pillars: well-designed, documented, and interoperable systems make data accessible to partners and downstream workflows, rather than being trapped behind bespoke context.

Evidence. The STRONG Manoomin Collective (chapter 6) formalized commitments through MoUs and operates under CARE/FAIR-derived sovereignty protocols adapted from regional collaborations [237, 278–280]. Partners of Makak hold complete ownership of the data; however, the Collective manages it with permission [246]. Careful consideration is given to system components for data transport, including cloud storage and proprietary software (section 6.7.1) and no data was shared without prior permission. FlexiBLE stores all data locally in a self-descriptive SQLite database, easily accessible with any data science tooling and used for analysis by the cell oxygenator teams. In field interviews, data sovereignty surfaced independently as a concern for sovereign entities managing their own environmental data (section 5.4.5). Finally, the corpus is sparse in examples of co-development (section 2.3.3), and even sparser in descriptions of data management (section 2.3.2).

Enables. Collaborative engagements, including early discussions and agreements on data management, promote ethical research and should not be an afterthought. This is the trust that can sustain partnerships beyond a single project and bridge reciprocity to sustainable reuse. Additionally, this reframes interoperability as an ethical mechanism, not only an efficiency one: self-description and observability are what make data available and interpretable by the people it concerns.

7.4 Towards Sustainable Systems for Long-Term Reuse

This chapter’s contribution is not any single component; it’s how the three pillars reinforce one another into systems designed to survive beyond the prototype. Interoperability is the transport mechanism through which observability and parameterization are practiced. Device self-description and observability make data available and usable to partners, reframing

interoperability as an ethical mechanism rather than just for components' functionality. Finally, reducing the labor required to build system infrastructure (technical scaffolding) by developing reusable tooling and methods for these components frees resources to focus on application-specific and collaborative efforts, which foster long-term relationships and, ideally, lead to more impactful technical solutions.

When these guidelines are practiced together, the goal is for systems to be cleanly handed off to a community, used as a solid starting point for an iterative project with similar technology for a new application. The combined outlook of the system design presented in these guidelines positions initiatives for sustainability, with the capability to be maintained and outlived by the timeline of academic contributions. However, in this thesis, we present evidence-supported findings without formalization or validation into a complete framework. The practice of these guidelines and the development of tooling that supports them are future work.

CHAPTER 8

Conclusions and Future Work

This thesis takes significant steps toward the development of low-power wireless embedded systems, exploring methods to make these systems more tractable and to narrow the gap between a working prototype and a system that produces meaningful data in the field. We argue that the effort spent rebuilding *technical scaffolding*, the infrastructure that constitutes a research contribution in itself, is a primary barrier to impact, and that academic incentives rarely reward reducing it. To investigate this, we paired a review of ACM literature with retrospectives on first-hand system development across two domains: mobile health and environmental sensing.

We presented four case studies. FaceBit (chapter 3), a smart face mask for respiratory health sensing, exposed the cost of iteration in wireless and mobile health and motivated the work that followed. FlexiBLE (chapter 4), a declarative platform for rapid wearable development, responded to those costs and was exercised in deployment of encapsulated cell therapies [164, 192] (section 4.6) and PhysioHub (section 4.5), and evaluated through interviews with users and co-developers (section 4.7). Turning to environmental sensing, our analysis of field-sensor interviews (chapter 5) introduced the field-sensor life cycle and its recurring barriers; Makak (chapter 6) grounded an end-to-end, multi-season co-design with Ojibwe Tribal Nations and organizations for the conservation of Manoomin. Interpreting these systems together and against the corpus of ACM literature (chapter 2), we synthesized eleven guidelines for embedded sensing development (chapter 7).

These guidelines are supported by evidence, but are not yet a formalized or validated framework. The remainder of this chapter summarizes the dissertation’s contributions and outlines the work that remains.

8.1 Summary of Contributions

Beyond the individual contributions of the systems presented in chapters 3 to 6, each published or developed in its own right, this dissertation makes three cross-cutting contributions that follow from reading these systems together and against the literature.

- A coded corpus of post-COVID-19 ACM literature (chapter 2) characterizing how wireless embedded sensing systems are motivated (RQ1), how they are built and what technical details are reported (RQ2), and how they are evaluated relative to their intended domain (RQ3).
- A set of retrospective analyses connecting prior work to the thesis: the development cycle of FaceBit (section 3.6.1), an interview study on the use and development of FlexiBLE (section 4.7), an expanded examination of the coding data from our environmental field sensor study (section 5.4), and an analysis of the technical iterations between two deployments of Makak (section 6.7).
- Eleven guidelines across three pillars—architectural patterns, a mechanism for interoperability, and co-development practices—that generalize across wireless embedded sensing systems are grounded in evidence from both first-hand development and the corpus (chapter 7).

8.2 Future Work

In this dissertation, we have taken a retrospective look at the literature and our own work to synthesize a component catalog of system design for low-power wireless embedded systems.

Future work is required to formalize and validate a development framework and implement its capabilities in tooling for generalized use. In addition, we see a benefit in expanding the literature review as a reference for engineering building systems across domain applications.

8.2.1 A Knowledge Base for System Development

Our literature review (chapter 2) is, to our knowledge, among the first to categorize the cross-disciplinary development of wireless embedded sensing systems, but it lacks systematic methodology (section 2.5). Expanding the corpus beyond the ACM, broadening it past the health and environmental domains, and adding more robust cross-coding would make its methodology fully systematic. Published as an interactive database, the corpus could then serve as a reference for engineers, sharing development knowledge across domains and academic venues.

8.2.2 Framework Development

The eleven guidelines presented in chapter 7 form an evidence-grounded catalog, not yet a formalized framework. The relationships among components are only partially articulated, and the set has not been validated beyond the systems and corpus from which it was derived. Formalizing this required following existing methods of framework development. The taxonomy-development method of Nickerson et al. [281], revised for software engineering by Usman et al. [282], provides a systematic iterative procedure with explicit ending conditions for deriving and structuring such components rigorously rather than *ad hoc* as we have done here. Design science research offers the complementary frame for positioning and evaluating the result as a design artifact [283, 284]. Two structural steps remain: defining and extending inter-component relationships, dependencies, composition, and trade-offs, extending the evidence mapping of fig. 7.1, and organizing the components along a development life cycle. Validation would then proceed through expert review with practitioners and partners and through application with systems beyond our own. This is a long-term endeavor and would

require a strong relationship with practitioners across embedded systems development and application-specific domains.

8.2.3 Toolkit Development

Toolkit development must be preceded by the identification of design components, and this thesis serves as that precursor. We have identified gaps where toolkit research can serve the broader community, and we provide a starting point with our implementation work. FlexiBLE has already enabled new research in wireless and mobile health, and this dissertation details both its benefits and its frictions in practice (sections 4.7 and 4.7.3), providing grounding for new work. One path generalizes FlexiBLE’s conceptual design into a platform free of its current constraints, removing the dependence on the Apple software ecosystem and on BLE as the sole transport, but realizing an application-layer transport protocol from the interoperability components of section 7.2. The remaining step is formal validation of the toolkit’s effect on the development cycle. Ledo et al. [33] categorize four strategies for evaluating HCI toolkit research: demonstration, usage, technical performance, and heuristics. FlexiBLE has been demonstrated and partially evaluated for technical performance (e.g., sustained throughput in PhysioHub, section 4.5), and our interviews provide heuristic insight (section 4.7). A usage evaluation, studying how embedded-systems developers build with the platform, and how well the systems they build serve their target application, remains as future work. We hold this agenda loosely, however: a toolkit proves its worth by being used, not by evaluation, arguing its potential. That tension is the crux of the incentive misalignment this thesis documents (chapter 1); building and sustaining a cross-disciplinary community of engineers and practitioners around such a system is a greater challenge than demonstrating it.

References

- [1] E. Greenlee, B. Rothrock, H. Kim, E. Zegura, and J. Hester. ““The Devil You Know”: Barriers and Opportunities for Co-Designing Microclimate Sensors, A Case Study of Manoomi”. In: *ACM Journal on Computing and Sustainable Societies* 2.3 (2024), pp. 1–30.
- [2] Ajmal, T. Boonya-Ananta, A. J. Rodriguez, V. Du Le, and J. C. Ramella-Roman. “Monte Carlo analysis of optical heart rate sensors in commercial wearables: the effect of skin tone and obesity on the photoplethysmography (PPG) signal”. In: *Biomedical optics express* 12.12 (2021), pp. 7445–7457.
- [3] Z. Hassan, C. Treude, M. Norrish, G. Williams, and A. Potanin. “Characterising reproducibility debt in scientific software: A systematic literature review”. In: *Journal of Systems and Software* 222 (2025), p. 112327.
- [4] G. V. Neville-Neil. “Can’t We Have Nice Things?” In: *Communications of the ACM* 68.8 (2025), pp. 24–25.
- [5] C. Frank, C. Sink, L. Mynatt, R. Rogers, and A. Rappazzo. “Surviving the “valley of death”: A comparative analysis”. In: *The Journal of Technology Transfer* 21.1 (1996), pp. 61–69.
- [6] C. R. Greenwood and M. Abbott. “The research to practice gap in special education”. In: *Teacher Education and Special Education* 24.4 (2001), pp. 276–289.
- [7] G. Bennett and N. Jessani. *The knowledge translation toolkit: Bridging the know-do gap: A resource for researchers*. IDRC, 2011.
- [8] D. E. Stokes. *Pasteur’s quadrant: Basic science and technological innovation*. Bloomsbury Publishing USA, 2011.
- [9] T. Isenberg. “Personal experiences of providing and using research prototypes”. In: *Proceedings of VisGap 2022-The Gap between Visualization Research and Visualization Software*. Eurographics Association. 2022, pp. 17–22.
- [10] T. R. Hansen, J. E. Bardram, and M. Soegaard. “Moving out of the lab: Deploying pervasive technologies in a hospital”. In: *IEEE Pervasive Computing* 5.3 (2006), pp. 24–31.
- [11] M. Shaw. “Writing good software engineering research papers”. In: *25th International Conference on Software Engineering, 2003. Proceedings*. IEEE. 2003, pp. 726–736.

- [12] G. Neville-Neil. “Can’t We Have Nice Things?” en. In: *Queue* 23.2 (May 2025), pp. 11–17. ISSN: 1542-7730, 1542-7749. DOI: 10.1145/3733697. URL: <https://dl.acm.org/doi/10.1145/3733697> (visited on 08/26/2025).
- [13] P. Tolmie, A. Crabtree, S. Egglestone, J. Humble, C. Greenhalgh, and T. Rodden. “Digital plumbing: the mundane work of deploying UbiComp in the home”. In: *Personal and Ubiquitous Computing* 14.3 (2010), pp. 181–196.
- [14] G. D. Abowd. “What next, ubicomp? Celebrating an intellectual disappearing act”. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. 2012, pp. 31–40.
- [15] J. C. Carver, N. Weber, K. Ram, S. Gesing, and D. Katz. “A survey of the state of the practice for research software in the United States”. In: *PeerJ Computer Science* 8 (2022). DOI: 10.7717/peerj-cs.963.
- [16] H. Kleinwaks, A. Batchelor, and T. H. Bradley. “Technical debt in systems engineering—A systematic literature review”. In: *Systems Engineering* 26.5 (2023), pp. 675–687.
- [17] E. P. Baumer and M. S. Silberman. “When the implication is not to design (technology)”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 2271–2274.
- [18] P. Dourish and S. D. Mainwaring. “UbiComp’s colonial impulse”. In: *Proceedings of the 2012 ACM conference on ubiquitous computing*. 2012, pp. 133–142.
- [19] L. Irani, J. Vertesi, P. Dourish, K. Philip, and R. E. Grinter. “Postcolonial computing: a lens on design and development”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 1311–1320.
- [20] B. Rothrock, E. Greenlee, and J. Hester. “Community-Driven Mobile and Ubiquitous Computing”. In: *GetMobile: Mobile Comp. and Comm.* 27.3 (Nov. 2023), pp. 5–9. ISSN: 2375-0529. DOI: 10.1145/3631588.3631590. URL: <https://dl.acm.org/doi/10.1145/3631588.3631590> (visited on 08/28/2024).
- [21] B. Rothrock, E. Greenlee, Y. Sangar, J. McKenna, W. Graveen, K. Hanson, M. Lewis, K. Smith, M. Falck, B. Byrne, D. Vogt, K. R. M. Suisseea, E. Zegura, J. Hester, and A. Cabral. “Makak: Co-designing Environmental Sensors to Protect Manoomin (Wild Rice)”. en. In: (2025).
- [22] A. Kilfoy, T. Hsu, C. Stockton-Powdrell, P. Whelan, C. H. Chu, and L. Jibb. “An umbrella review on how digital health intervention co-design is conducted and described”. In: *NPJ Digital Medicine* 7 (2024). DOI: 10.1038/s41746-024-01385-1.

- [23] P. Slattery, A. K. Saeri, and P. Bragge. “Research co-design in health: a rapid overview of reviews”. In: *Health Research Policy and Systems* 18 (2020). DOI: 10.1186/s12961-020-0528-9.
- [24] S. Merkel and A. Kucharski. “Participatory Design in Gerontechnology: A Systematic Literature Review”. In: *The Gerontologist* 59 (2019), e16–e25. DOI: 10.1093/geront/gny034.
- [25] A. K. Daniels. “Invisible work”. In: *Social problems* 34.5 (1987), pp. 403–415.
- [26] S. L. Star and A. Strauss. “Layers of silence, arenas of voice: The ecology of visible and invisible work”. In: *Computer supported cooperative work (CSCW)* 8.1 (1999), pp. 9–30.
- [27] M. Ulloa, B. Rothrock, F. S. Ahmad, and M. Jacobs. “Invisible clinical labor driving the successful integration of AI in healthcare”. In: *Frontiers in Computer Science* 4 (2022), p. 1045704.
- [28] F. Adame. “Meaningful collaborations can end ‘helicopter research’”. In: *Nature* 10 (2021).
- [29] OECD. *Mind the SDG Data Gaps: Insights from the OECD SDGs Hub*. Tech. rep. 20. Paris: OECD Publishing, 2025. DOI: 10.1787/9aff09e2-en. URL: <https://doi.org/10.1787/9aff09e2-en>.
- [30] Independent Expert Advisory Group on a Data Revolution for Sustainable Development. *A World That Counts: Mobilising the Data Revolution for Sustainable Development*. Report to the UN Secretary-General. New York: United Nations, Nov. 2014. URL: <https://www.undatarevolution.org/report/>.
- [31] W. Chen, S. Lin, E. Thompson, and J. Stankovic. “Sensecollect: We need efficient ways to collect on-body sensor-based human activity data!” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–27.
- [32] F. Oliveira, D. G. Costa, F. Assis, and I. Silva. “Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning”. In: *Internet of Things* 26 (2024), p. 101153.
- [33] D. Ledo, S. Houben, J. Vermeulen, N. Marquardt, L. Oehlberg, and S. Greenberg. “Evaluation strategies for HCI toolkit research”. In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–17.
- [34] M. Nebeling. “Playing the Tricky Game of Toolkits Research”. In: *workshop on HCI. Tools at CHI*. 2017.
- [35] J. S. Trueblood, D. B. Allison, S. M. Field, A. Fishbach, S. D. Gaillard, G. Gigerenzer, W. R. Holmes, S. Lewandowsky, D. Matzke, M. C. Murphy, et al. “The misalignment of

- incentives in academic publishing and implications for journal reform”. In: *Proceedings of the National Academy of Sciences* 122.5 (2025), e2401231121.
- [36] H. Arksey and L. O’malley. “Scoping studies: towards a methodological framework”. In: *International journal of social research methodology* 8.1 (2005), pp. 19–32.
 - [37] Qwen Team. *Qwen3.5: Towards Native Multimodal Agents*. Feb. 2026. URL: <https://qwen.ai/blog?id=qwen3.5>.
 - [38] N. Bhat, A. Gupta, I. Bansal, H. Govindarajan, and D. Bharadia. “ZenseTag: An RFID assisted Twin-Tag Single Antenna COTS Sensor Interface”. In: *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. SenSys ’24. Hangzhou, China: Association for Computing Machinery, 2024, pp. 336–350. ISBN: 9798400706974. DOI: 10.1145/3666025.3699342. URL: <https://doi-org.turing.library.northwestern.edu/10.1145/3666025.3699342>.
 - [39] Y. Sun, Y. Liu, Z. Wang, X. Qu, D. Zheng, and X. Chen. “C-RIDGE: Indoor CO2 Data Collection System for Large Venues Based on prior Knowledge”. In: *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. SenSys ’22. Boston, Massachusetts: Association for Computing Machinery, 2023, pp. 1077–1082. ISBN: 9781450398862. DOI: 10.1145/3560905.3567769. URL: <https://doi.org/10.1145/3560905.3567769>.
 - [40] C.-W. Yau, S. Jewsakul, M.-H. Luk, A. P. Y. Lee, Y.-H. Chan, E. C. H. Ngai, P. W. T. Pong, K.-S. Lui, and J. Liu. “NB-IoT Coverage and Sensor Node Connectivity in Dense Urban Environments: An Empirical Study”. In: *ACM Trans. Sen. Netw.* 18.3 (Sept. 2022). ISSN: 1550-4859. DOI: 10.1145/3536424. URL: <https://doi.org/10.1145/3536424>.
 - [41] R. Otterbein, E. Jochum, D. Overholt, S. Bai, and A. Dalsgaard. “Dance and Movement-Led Research for Designing and Evaluating Wearable Human-Computer Interfaces”. In: *Proceedings of the 8th International Conference on Movement and Computing*. MOCO ’22. Chicago, IL, USA: Association for Computing Machinery, 2022. ISBN: 9781450387163. DOI: 10.1145/3537972.3537984. URL: <https://doi.org/10.1145/3537972.3537984>.
 - [42] J. Lu and P. Lopes. “Integrating Living Organisms in Devices to Implement Care-based Interactions”. In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST ’22. Bend, OR, USA: Association for Computing Machinery, 2022. ISBN: 9781450393201. DOI: 10.1145/3526113.3545629. URL: <https://doi.org/10.1145/3526113.3545629>.
 - [43] R. Liu, T. Lu, S. Yuan, H. Zhou, and M. Gowda. “SmartDampener: An Open Source Platform for Sport Analytics in Tennis”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8.3 (Sept. 2024). DOI: 10.1145/3678507. URL: <https://doi.org/10.1145/3678507>.

- [44] X. Cao, Y. Gao, W. Ding, and C. Li*. “A Wearable Electrochemical Sensor for Testing Carcinoembryonic Antigen in Sweat”. In: *Proceedings of the 2023 4th International Symposium on Artificial Intelligence for Medicine Science*. ISAIMS ’23. Chengdu, China: Association for Computing Machinery, 2024, pp. 722–727. ISBN: 9798400708138. DOI: 10.1145/3644116.3644237. URL: <https://doi.org/10.1145/3644116.3644237>.
- [45] Z. Hu, A. Radmehr, Y. Zhang, S. Pan, and P. Nguyen. “IOTeeth: Intra-Oral Teeth Sensing System for Dental Occlusal Diseases Recognition”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8.1 (Mar. 2024). DOI: 10.1145/3643516. URL: <https://doi.org/10.1145/3643516>.
- [46] T. Dang, T. Tran, K. Nguyen, T. Pham, N. Pham, T. Vu, and P. Nguyen. “IoTree: a battery-free wearable system with biocompatible sensors for continuous tree health monitoring”. In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. MobiCom ’22. Sydney, NSW, Australia: Association for Computing Machinery, 2022, pp. 352–366. ISBN: 9781450391818. DOI: 10.1145/3495243.3567652. URL: <https://doi.org/10.1145/3495243.3567652>.
- [47] C. Li, J. Peng, Q. Hu, L. Chen, Y. Huang, S. Wu, W. Cheng, and Q. Zhang. “PracticalBP: Continuous Cuffless Blood Pressure Monitoring with Only One Record for Calibration”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.3 (Sept. 2025). DOI: 10.1145/3749486. URL: <https://doi.org/10.1145/3749486>.
- [48] A. Watson, C. Kendell, A. Lingamoorthy, I. Lee, and J. Weimer. “Lumos: An Open-Source Device for Wearable Spectroscopy Research”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.4 (Jan. 2023). DOI: 10.1145/3569502. URL: <https://doi.org/10.1145/3569502>.
- [49] T. Röddiger, M. Küttner, P. Lepold, T. King, D. Moschina, O. Bagge, J. A. Paradiso, C. Clarke, and M. Beigl. “OpenEarable 2.0: Open-Source Earphone Platform for Physiological Ear Sensing”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.1 (Mar. 2025). DOI: 10.1145/3712069. URL: <https://doi.org/10.1145/3712069>.
- [50] S. Mahmud, K. Li, G. Hu, H. Chen, R. Jin, R. Zhang, F. Guimbretière, and C. Zhang. “PoseSonic: 3D Upper Body Pose Estimation Through Egocentric Acoustic Sensing on Smartglasses”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7.3 (Sept. 2023). DOI: 10.1145/3610895. URL: <https://doi.org/10.1145/3610895>.
- [51] J. Wang, Y. Feng, G. Kumbhar, G. Wang, Q. Yan, Q. Jin, R. C. Ferrier, J. Xiong, and T. Li. “SoilCares: Towards Low-cost Soil Macronutrients and Moisture Monitoring Using RF-VNIR Sensing”. In: *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*. MOBISYS ’24. Minato-ku, Tokyo, Japan: Association for Computing Machinery, 2024, pp. 196–209. ISBN: 9798400705816. DOI: 10.1145/3643832.3661868. URL: <https://doi.org/10.1145/3643832.3661868>.

- [52] Z. Yin, N. T. Nguyen, A. Zuniga, M. Liyanage, P. Nurmi, and H. Flores. “SpikEy: Preventing Drink Spiking using Wearables”. In: *Proceedings of the 27th International Conference on Multimodal Interaction*. ICMI ’25. Association for Computing Machinery, 2025, pp. 78–86. ISBN: 9798400714993. DOI: 10.1145/3716553.3750782. URL: <https://doi.org/10.1145/3716553.3750782>.
- [53] K. Y. Choi, N. ElHaouij, J. Lee, R. W. Picard, and H. Ishii. “Design and Evaluation of a Clippable and Personalizable Pneumatic-haptic Feedback Device for Breathing Guidance”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.1 (Mar. 2022). DOI: 10.1145/3517234. URL: <https://doi.org/10.1145/3517234>.
- [54] Z. Yin, M. Liyanage, A.-R. Ottun, S. Paul, A. Zuniga, P. Nurmi, and H. Flores. “HIPPO: Pervasive Hand-Grip Estimation from Everyday Interactions”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.4 (Jan. 2023). DOI: 10.1145/3570344. URL: <https://doi.org/10.1145/3570344>.
- [55] M. D. Pabiania, C. S. Salvador, C. S. P. Palupit, B. N. D. Barba, and A. S. D. Yu. “Hands-Free Electrolarynx with Pitch Control”. In: *Proceedings of the 2024 14th International Conference on Biomedical Engineering and Technology*. ICBET ’24. Seoul, Republic of Korea: Association for Computing Machinery, 2024, pp. 187–190. ISBN: 9798400717628. DOI: 10.1145/3678935.3678979. URL: <https://doi.org/10.1145/3678935.3678979>.
- [56] V. Shen, X. Yang, C. Harrison, and Y. Zhang. “Hapt-Aids: Self-Powered, On-Body Haptics for Activity Monitoring”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.3 (Sept. 2025). DOI: 10.1145/3749468. URL: <https://doi.org/10.1145/3749468>.
- [57] J. Christensen, S. Kauenhofen, J. D. Loehr, J. Lang, S. Peacock, and J. J. Nicol. “MMM Duet System: New accessible musical technology for people living with dementia”. In: *Proceedings of the 18th International Audio Mostly Conference*. AM ’23. Edinburgh, United Kingdom: Association for Computing Machinery, 2023, pp. 38–44. ISBN: 9798400708183. DOI: 10.1145/3616195.3616205. URL: <https://doi.org/10.1145/3616195.3616205>.
- [58] Y. Jiang, J. Kleinau, T. M. Eckroth, E. Hoggan, S. Mueller, and M. Wessely. “MouthIO: Fabricating Customizable Oral User Interfaces with Integrated Sensing and Actuation”. In: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. UIST ’24. Pittsburgh, PA, USA: Association for Computing Machinery, 2024. ISBN: 9798400706288. DOI: 10.1145/3654777.3676443. URL: <https://doi.org/10.1145/3654777.3676443>.
- [59] S. C. Till, T. Wilson, M. Dewlok, N. Verdezoto, and M. Densmore. “Move With Me: Co-designing a Tangible User Interface To Promote Physical Activity Among Rural South African Children”. In: *Proceedings of the 12th International Conference on Communities & Technologies*. C&T ’25. Association for Computing Machinery,

- 2025, pp. 173–187. ISBN: 9798400715211. DOI: 10.1145/3742800.3742825. URL: <https://doi.org/10.1145/3742800.3742825>.
- [60] X. Ding and W. Du. “Optimizing Irrigation Efficiency using Deep Reinforcement Learning in the Field”. In: *ACM Trans. Sen. Netw.* 20.4 (July 2024). ISSN: 1550-4859. DOI: 10.1145/3662182. URL: <https://doi.org/10.1145/3662182>.
 - [61] Z. Hou, Z. Xiong, Y. He, Z. Xiong, H. Pan, C. Wang, H. Yang, and H. Tang. “Design and Implementation of an Intelligent Monitoring System for Vertical Farming Based on STM32F103RCT6”. In: *Proceedings of the 2025 International Conference on Smart Agriculture and Artificial Intelligence*. SAAI ’25. Association for Computing Machinery, 2025, pp. 184–191. ISBN: 9798400715907. DOI: 10.1145/3767624.3767651. URL: <https://doi.org/10.1145/3767624.3767651>.
 - [62] R. Batra, N. Pourjafarian, S. Chang, M. Tsai, J. Revelo, and C. H.-L. Kao. “texTENG: Fabricating Wearable Textile-Based Triboelectric Nanogenerators”. In: *Proceedings of the Augmented Humans International Conference 2025*. AHs ’25. Association for Computing Machinery, 2025, pp. 124–138. ISBN: 9798400715662. DOI: 10.1145/3745900.3746071. URL: <https://doi.org/10.1145/3745900.3746071>.
 - [63] B. Yen, L. Jaliff, L. Gutierrez, P. Sahinidis, S. Bernstein, J. Madden, S. Taylor, C. Josephson, P. Pannuto, W. Shuai, G. Wells, N. Arora, and J. Hester. “Soil-Powered Computing: The Engineer’s Guide to Practical Soil Microbial Fuel Cell Design”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7.4 (Jan. 2024). DOI: 10.1145/3631410. URL: <https://doi.org/10.1145/3631410>.
 - [64] L. Li, Y. Xie, J. Xiong, Z. Hou, Y. Zhang, Q. We, F. Wang, D. Fang, and X. Chen. “SmartLens: sensing eye activities using zero-power contact lens”. In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. MobiCom ’22. Sydney, NSW, Australia: Association for Computing Machinery, 2022, pp. 473–486. ISBN: 9781450391818. DOI: 10.1145/3495243.3560532. URL: <https://doi.org/10.1145/3495243.3560532>.
 - [65] H. Pan, Y. Wang, J. Liu, R. Ma, L. Qiu, Y.-C. Chen, G. Xue, and J. Ren. “CGMM: Non-Invasive Continuous Glucose Monitoring in Wearables Using Metasurfaces”. In: *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*. ACM MOBICOM ’25. Kerry Hotel, Hong Kong, Hong Kong, China: Association for Computing Machinery, 2025, pp. 283–298. ISBN: 9798400711299. DOI: 10.1145/3680207.3723474. URL: <https://doi.org/10.1145/3680207.3723474>.
 - [66] J. Chen, S.-W. Jun, A. Mundra, and J. Ta. “Xyloni: Very Low Power Neural Network Accelerator for Intermittent Remote Visual Detection of Wildfire and Beyond”. In: *Proceedings of the 29th ACM/IEEE International Symposium on Low Power Electronics and Design*. ISLPED ’24. Newport Beach, CA, USA: Association for Computing Machinery, 2024, pp. 1–6. ISBN: 9798400706882. DOI: 10.1145/3665314.3670810. URL: <https://doi.org/10.1145/3665314.3670810>.

- [67] C. Wu, Z. Yue, Q. Cao, T. Fang, F. Fei, and A. Song. “Development of a Six-Channel Wearable Electromyography Sensor for hand movement intention recognition”. In: *Proceedings of the 2024 4th International Conference on Robotics and Control Engineering*. RobCE '24. Edinburgh, United Kingdom: Association for Computing Machinery, 2024, pp. 50–56. ISBN: 9798400716782. DOI: 10.1145/3674746.3674755. URL: <https://doi.org/10.1145/3674746.3674755>.
- [68] B. Yang, X. Zhang, J. Zhang, Z. Huang, Q. Lu, J. Zhang, H. Hu, and Q. Zhang. “KneeGuard: A Calibration-free Wearable Monitoring System for Knee Osteoarthritis Gait Re-training via Effortless Wearing”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8.4 (Nov. 2024). DOI: 10.1145/3699768. URL: <https://doi.org/10.1145/3699768>.
- [69] Y. Wu, L. A. G. Villalobos, Z. Yang, G. T. Croisdale, Ç. Karataş, and J. Liu. “SmarCyPad: A Smart Seat Pad for Cycling Fitness Tracking Leveraging Low-cost Conductive Fabric Sensors”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7.3 (Sept. 2023). DOI: 10.1145/3610927. URL: <https://doi.org/10.1145/3610927>.
- [70] T. C. Yu, R. Arakawa, J. McCann, and M. Goel. “uKnit: A Position-Aware Reconfigurable Machine-Knitted Wearable for Gestural Interaction and Passive Sensing using Electrical Impedance Tomography”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: 10.1145/3544548.3580692. URL: <https://doi.org/10.1145/3544548.3580692>.
- [71] C. Honnet, W. Babatain, Y. Luo, O. Kilic Afsar, C. Bensahel, S. Nicita, Y. Zhu, A. Danieleescu, N. Gershenfeld, and J. Paradiso. “FiberCircuits: A Miniaturization Framework To Manufacture Fibers That Embed Integrated Circuits”. In: *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. UIST '25. Association for Computing Machinery, 2025. ISBN: 9798400720376. DOI: 10.1145/3746059.3747802. URL: <https://doi.org/10.1145/3746059.3747802>.
- [72] R. Sharma, R. Kapur, A. Bhavsar, V. Dutt, B. Kumar, and V. Kanwar. “DiaBreath: A Low-Cost, Non-Invasive Diabetes Monitor via Breath”. In: *ACM Trans. Comput. Healthcare* (Sept. 2025). Just Accepted. DOI: 10.1145/3768157. URL: <https://doi.org/10.1145/3768157>.
- [73] Z. Li, T. Lu, R. Liu, S. Yuan, J. Gosalia, C. Bowlus, D. N. Proctor, J. A. Pawelczyk, Y. Jin, and M. Gowda. “FuSenseRing: An Open-Source Platform for Robust Cuffless Blood Pressure Monitoring via Multimodal Sensor Fusion and Temperature-Adaptive Attention”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.4 (Dec. 2025). DOI: 10.1145/3770689. URL: <https://doi.org/10.1145/3770689>.
- [74] S. G. Burman, J. Gao, G. B. Pasternack, N. A. Fangue, P. Cadrett, E. Campbell, and D. Ghosal. “TempMesh – A Flexible Wireless Sensor Network for Monitoring River

- Temperatures”. In: *ACM Trans. Sen. Netw.* 19.1 (Dec. 2022). ISSN: 1550-4859. DOI: 10.1145/3542697. URL: <https://doi.org/10.1145/3542697>.
- [75] J. Madden, G. Marcano, S. Taylor, P. Pannuto, and C. Josephson. “Hardware to Enable Large-Scale Deployment and Observation of Soil Microbial Fuel Cells”. In: *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. SenSys ’22. Boston, Massachusetts: Association for Computing Machinery, 2023, pp. 906–912. ISBN: 9781450398862. DOI: 10.1145/3560905.3568110. URL: <https://doi.org/10.1145/3560905.3568110>.
 - [76] Y. Ren, W. Sun, J. Du, H. Zeng, Y. Dong, M. Zhang, S. Chen, Y. Liu, T. Li, and Z. Cao. “Demeter: Reliable Cross-soil LPWAN with Low-cost Signal Polarization Alignment”. In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. ACM MobiCom ’24. Washington D.C., DC, USA: Association for Computing Machinery, 2024, pp. 230–245. ISBN: 9798400704895. DOI: 10.1145/3636534.3649358. URL: <https://doi.org/10.1145/3636534.3649358>.
 - [77] M. M. Tobias, M. McLain, N. J. Webster, and B. Kerkez. “Barriers to Coordination and the Role of Technology in Community-Managed Watersheds”. In: *Proceedings of the 2025 ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. COMPASS ’25. Association for Computing Machinery, 2025, pp. 245–260. ISBN: 9798400714849. DOI: 10.1145/3715335.3735469. URL: <https://doi.org/10.1145/3715335.3735469>.
 - [78] R. Alharbi, S. Shahi, S. Cruz, L. Li, S. Sen, M. Pedram, C. Romano, J. Hester, A. K. Katsaggelos, and N. Alshurafa. “SmokeMon: Unobtrusive Extraction of Smoking Topography Using Wearable Energy-Efficient Thermal”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.4 (Jan. 2023). DOI: 10.1145/3569460. URL: <https://doi.org/10.1145/3569460>.
 - [79] A. N. Balaji, A. Ferlini, F. Kawsar, and A. Montanari. “Stereo-BP: Non-Invasive Blood Pressure Sensing with Earables”. In: *Proceedings of the 24th International Workshop on Mobile Computing Systems and Applications*. HotMobile ’23. Newport Beach, California: Association for Computing Machinery, 2023, pp. 96–102. ISBN: 9798400700170. DOI: 10.1145/3572864.3580341. URL: <https://doi.org/10.1145/3572864.3580341>.
 - [80] M. Meier and C. Holz. “BMAR: Barometric and Motion-based Alignment and Refinement for Offline Signal Synchronization across Devices”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7.2 (June 2023). DOI: 10.1145/3596268. URL: <https://doi.org/10.1145/3596268>.
 - [81] G. Zhu, D. Zhao, Y. Song, Z. Wang, A. Zhou, H. Ma, X. Meng, J. Wang, C. Shao, and Y. Tang. “RingBP: Towards Continuous, Comfortable, and Generalized Blood Pressure Monitoring Using a Smart Ring”. In: *Proc. ACM Interact. Mob. Wearable*

- Ubiquitous Technol.* 9.2 (June 2025). DOI: 10.1145/3731600. URL: <https://doi.org/10.1145/3731600>.
- [82] A. Aziz, N. Pham, N. Vora, C. Reynolds, J. Lehnen, P. Venkatesh, Z. Yao, J. Harvey, T. Vu, K. Ding, and P. Nguyen. “An Unobtrusive and Lightweight Ear-worn System for Continuous Epileptic Seizure Detection”. In: *ACM Trans. Comput. Healthcare* 6.1 (Jan. 2025). DOI: 10.1145/3703164. URL: <https://doi.org/10.1145/3703164>.
 - [83] H. Zhou, T. Lu, Y. Liu, S. Zhang, R. Liu, and M. Gowda. “One Ring to Rule Them All: An Open Source Smartring Platform for Finger Motion Analytics and Healthcare Applications”. In: *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*. IoTDI '23. San Antonio, TX, USA: Association for Computing Machinery, 2023, pp. 27–38. ISBN: 9798400700378. DOI: 10.1145/3576842.3582382. URL: <https://doi.org/10.1145/3576842.3582382>.
 - [84] B. Krupp, J. Fagert, C. Fan, I. Zhang, and D. Hawkins. “AQIoT: Partnering with Community Organizations to Provide Fine-Grained Air Quality Data”. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Spatial Intelligence for Smart and Connected Communities*. SpatialConnect '25. The Graduate Hotel Minneapolis, Minneapolis, MN, USA: Association for Computing Machinery, 2025, pp. 25–34. ISBN: 9798400721878. DOI: 10.1145/3764924.3770896. URL: <https://doi.org/10.1145/3764924.3770896>.
 - [85] Y. Zhao, Y. Tao, G. Le, R. Maki, A. Adams, P. Lopes, and T. Choudhury. “Affective Touch as Immediate and Passive Wearable Intervention”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.4 (Jan. 2023). DOI: 10.1145/3569484. URL: <https://doi.org/10.1145/3569484>.
 - [86] V. Durairajah, S. Gobee, and S. C. S. Xuan. “A Low-Cost CPM Machine for Passive and Active Wrist Rehabilitation of Both Arms”. In: *Proceedings of the 2023 13th International Conference on Biomedical Engineering and Technology*. ICBET '23. Tokyo, Japan: Association for Computing Machinery, 2023, pp. 80–87. ISBN: 9798400707438. DOI: 10.1145/3620679.3620692. URL: <https://doi.org/10.1145/3620679.3620692>.
 - [87] K. Mahmoodi, Y. Xie, T. Gemicioglu, C.-J. Lee, J. Kim, and C. Zhang. “EchoForce: Continuous Grip Force Estimation from Skin Deformation Using Active Acoustic Sensing on a Wristband”. In: *Proceedings of the 2025 ACM International Symposium on Wearable Computers*. ISWC '25. Espoo, Finland: Association for Computing Machinery, 2025, pp. 53–60. ISBN: 9798400714818. DOI: 10.1145/3715071.3750405. URL: <https://doi.org/10.1145/3715071.3750405>.
 - [88] M. Liu, S. Suh, B. Zhou, A. Gruenerbl, and P. Lukowicz. “Smart-Badge: A wearable badge with multi-modal sensors for kitchen activity recognition”. In: *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*. Ubi-

- Comp/ISWC '22 Adjunct. Cambridge, United Kingdom: Association for Computing Machinery, 2023, pp. 356–363. ISBN: 9781450394239. DOI: 10.1145/3544793.3560391. URL: <https://doi.org/10.1145/3544793.3560391>.
- [89] Q. Lin, S. Peng, Y. Wu, J. Liu, H. Jia, W. Hu, M. Hassan, A. Seneviratne, and C. H. Wang. “Subject-adaptive Loose-fitting Smart Garment Platform for Human Activity Recognition”. In: *ACM Trans. Sen. Netw.* 19.4 (May 2023). ISSN: 1550-4859. DOI: 10.1145/3584986. URL: <https://doi.org/10.1145/3584986>.
- [90] M. Ragnoli, P. Esposito, V. Stornelli, G. Barile, E. D. Santis, and N. Sciarra. “A LoRa-based Wireless Sensor Network monitoring system for urban areas subjected to landslide”. In: *Proceedings of the 2023 8th International Conference on Cloud Computing and Internet of Things*. CCIOT '23. Okinawa, Japan: Association for Computing Machinery, 2023, pp. 91–97. ISBN: 9798400708046. DOI: 10.1145/3627345.3627359. URL: <https://doi.org/10.1145/3627345.3627359>.
- [91] Q. (Xue, D. Nissanka, J. T. Yan, R. Wang, S. Patel, and V. Iyer. “PPG Earring: Wireless Smart Earring for Heart Health Monitoring”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713856. URL: <https://doi.org/10.1145/3706598.3713856>.
- [92] Y. Li, Y. Zhou, C. Shen, and R. Stewart. “E-textile Sleeve with Graphene Strain Sensors for Arm Gesture Classification of Mid-Air Interactions”. In: *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '24. Cork, Ireland: Association for Computing Machinery, 2024. ISBN: 9798400704024. DOI: 10.1145/3623509.3633374. URL: <https://doi.org/10.1145/3623509.3633374>.
- [93] R. Zheng, J. Fang, Y. Yao, X. Gao, C. Zuo, S. Guo, and Y. Luo. “FIP: Endowing Robust Motion Capture on Daily Garment by Fusing Flex and Inertial Sensors”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3714140. URL: <https://doi.org/10.1145/3706598.3714140>.
- [94] K. Shintani, H. Rizk, and H. Yamaguchi. “Eco-Friendly Sensing for Human Activity Recognition”. In: *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*. UbiComp/ISWC '23 Adjunct. Cancun, Quintana Roo, Mexico: Association for Computing Machinery, 2023, pp. 492–498. ISBN: 9798400702006. DOI: 10.1145/3594739.3610744. URL: <https://doi.org/10.1145/3594739.3610744>.
- [95] L. Zhang, Q. Wang, H. Chen, J. Bao, J. Xu, and D. Li. “ARD: accurate and reliable fall detection with using a single wearable inertial sensor”. In: *Proceedings of the 1st ACM Workshop on Mobile and Wireless Sensing for Smart Healthcare*. MWSSH '22. Sydney, NSW, Australia: Association for Computing Machinery, 2022, pp. 13–18. ISBN:

9781450395205. DOI: 10.1145/3556551.3561189. URL: <https://doi.org/10.1145/3556551.3561189>.
- [96] D. Wang, J. Weng, Y. Zou, and K. Wu. “EmoTracer: A Wearable Physiological and Psychological Monitoring System With Multi-modal Sensors”. In: *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*. UbiComp/ISWC ’22 Adjunct. Cambridge, United Kingdom: Association for Computing Machinery, 2023, pp. 444–449. ISBN: 9781450394239. DOI: 10.1145/3544793.3560409. URL: <https://doi.org/10.1145/3544793.3560409>.
 - [97] C.-J. Lee, J. Li, T. C. Yu, R. Zhang, V. Gunda, F. Guimbretière, and C. Zhang. “Grab-n-Go: On-the-Go Microgesture Recognition with Objects in Hand”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.3 (Sept. 2025). DOI: 10.1145/3749469. URL: <https://doi.org/10.1145/3749469>.
 - [98] D. Geißler, H. Bello, E. Zahn, E. Woop, B. Zhou, P. Lukowicz, and J. Karolus. “Head ’n Shoulder: Gesture-Driven Biking Through Capacitive Sensing Garments to Innovate Hands-Free Interaction”. In: *Proc. ACM Hum.-Comput. Interact.* 8.MHCI (Sept. 2024). DOI: 10.1145/3676510. URL: <https://doi.org/10.1145/3676510>.
 - [99] Y. Dai, J. Wu, Y. Fan, J. Wang, J. Niu, F. Gu, and S. Shen. “MSEva: A Musculoskeletal Rehabilitation Evaluation System Based on EMG Signals”. In: *ACM Trans. Sen. Netw.* 19.1 (Dec. 2022). ISSN: 1550-4859. DOI: 10.1145/3522739. URL: <https://doi.org/10.1145/3522739>.
 - [100] B. Saha, R. Samanta, S. K. Ghosh, and R. B. Roy. “TinyML-Driven On-Device Personalized Human Activity Recognition and Auto-Deployment to Smart Bands”. In: *Proceedings of the Third International Conference on AI-ML Systems*. AIML-Systems ’23. Bangalore, India: Association for Computing Machinery, 2024. ISBN: 9798400716492. DOI: 10.1145/3639856.3639859. URL: <https://doi.org/10.1145/3639856.3639859>.
 - [101] T. L. Goh and L.-S. Peh. “WalkingWizard—A Truly Wearable EEG Headset for Everyday Use”. In: *ACM Trans. Comput. Healthcare* 5.2 (Apr. 2024). DOI: 10.1145/3648106. URL: <https://doi.org/10.1145/3648106>.
 - [102] H. Hiraki and J. Rekimoto. “MaskClip: Detachable Clip-on Piezoelectric Sensing of Mask Surface Vibrations for Real-time Noise-Robust Speech Input”. In: *Proceedings of the Augmented Humans International Conference 2025*. AHs ’25. Association for Computing Machinery, 2025, pp. 333–344. ISBN: 9798400715662. DOI: 10.1145/3745900.3746088. URL: <https://doi.org/10.1145/3745900.3746088>.
 - [103] Y. Gao, W. Zhang, J. Ren, R. Zheng, Y. Jin, D. Wu, L. Shu, X. Xu, and Z. Jin. “PressInPose: Integrating Pressure and Inertial Sensors for Full-Body Pose Estimation

- in Activities”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8.4 (Nov. 2024). DOI: 10.1145/3699773. URL: <https://doi.org/10.1145/3699773>.
- [104] Q. Shao, J. Chen, H. M. C. Leung, M. Zhao, R. Xu, J. Liu, L. M. D. García, X. Jiang, M.-P. St-Onge, and X. Zhou. “My Pillow Knows My Sleep: Sleep Monitoring with Computational Fabrics in the Pillowcase”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.3 (Sept. 2025). DOI: 10.1145/3749550. URL: <https://doi.org/10.1145/3749550>.
- [105] H. Zhou, T. Lu, Y. Liu, S. Zhang, and M. Gowda. “Learning on the Rings: Self-Supervised 3D Finger Motion Tracking Using Wearable Sensors”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.2 (July 2022). DOI: 10.1145/3534587. URL: <https://doi.org/10.1145/3534587>.
- [106] G. J. Fernandes, J. Zheng, M. Pedram, C. Romano, F. Shahabi, B. Rothrock, T. Cohen, H. Zhu, T. S. Butani, J. Hester, A. K. Katsaggelos, and N. Alshurafa. “Habit-Sense: A Privacy-Aware, AI-Enhanced Multimodal Wearable Platform for mHealth Applications”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8.3 (Sept. 2024). DOI: 10.1145/3678591. URL: <https://doi.org/10.1145/3678591>.
- [107] J. Chan, A. Glenn, M. Itani, L. R. Mancl, and E. R. Gallagher. “No Ear Left Behind: Wireless Earbuds for Low-Cost Hearing Screening”. In: *GetMobile: Mobile Comp. and Comm.* 28.3 (Oct. 2024), pp. 20–24. ISSN: 2375-0529. DOI: 10.1145/3701701.3701709. URL: <https://doi.org/10.1145/3701701.3701709>.
- [108] L. Zhang, Q. Zhao, R. Wang, S. Bian, O. Gungor, F. Ponzina, and T. S. Rosing. “Offload Rethinking by Cloud Assistance for Efficient Environmental Sound Recognition on LPWANs”. In: *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: Association for Computing Machinery, 2025, pp. 437–448. ISBN: 9798400714795. URL: <https://doi.org/10.1145/3715014.3722078>.
- [109] N. Pham, H. Jia, M. Tran, T. Dinh, N. Bui, Y. Kwon, D. Ma, P. Nguyen, C. Mascolo, and T. Vu. “PROS: an efficient pattern-driven compressive sensing framework for low-power biopotential-based wearables with on-chip intelligence”. In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. MobiCom ’22. Sydney, NSW, Australia: Association for Computing Machinery, 2022, pp. 661–675. ISBN: 9781450391818. DOI: 10.1145/3495243.3560533. URL: <https://doi.org/10.1145/3495243.3560533>.
- [110] H. Brombacher, R. Van Koningsbruggen, S. Vos, and S. Houben. “SensorBricks: a Collaborative Tangible Sensor Toolkit to Support the Development of Data Literacy”. In: *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’24. Cork, Ireland: Association for Computing Machinery, 2024. ISBN: 9798400704024. DOI: 10.1145/3623509.3633378. URL: <https://doi.org/10.1145/3623509.3633378>.

- [111] R. Armani, C. Qian, J. Jiang, and C. Holz. “Ultra Inertial Poser: Scalable Motion Capture and Tracking from Sparse Inertial Sensors and Ultra-Wideband Ranging”. In: *ACM SIGGRAPH 2024 Conference Papers*. SIGGRAPH ’24. Denver, CO, USA: Association for Computing Machinery, 2024. ISBN: 9798400705250. DOI: 10.1145/3641519.3657465. URL: <https://doi.org/10.1145/3641519.3657465>.
- [112] A. Waghmare, I. Chatterjee, V. Iyer, and S. Patel. “WatchLink: Enhancing Smartwatches with Sensor Add-Ons via ECG Interface”. In: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. UIST ’24. Pittsburgh, PA, USA: Association for Computing Machinery, 2024. ISBN: 9798400706288. DOI: 10.1145/3654777.3676329. URL: <https://doi.org/10.1145/3654777.3676329>.
- [113] S. Yuan, Y. Sun, S. Wang, X. Chen, Y. Ding, D. Zheng, and S. Fan. “Non-Acoustic Speech Sensing System Based on Flexible Piezoelectric”. In: *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. SenSys ’22. Boston, Massachusetts: Association for Computing Machinery, 2023, pp. 1055–1060. ISBN: 9781450398862. DOI: 10.1145/3560905.3567768. URL: <https://doi.org/10.1145/3560905.3567768>.
- [114] <https://www.acm.org/publications/policies/artifact-review-and-badging-current>. [Accessed 14-06-2026].
- [115] S. Winter, C. S. Timperley, B. Hermann, J. Cito, J. Bell, M. Hilton, and D. Beyer. “A retrospective study of one decade of artifact evaluations”. In: *Proceedings of the 30th ACM joint European software engineering conference and symposium on the foundations of software engineering*. 2022, pp. 145–156.
- [116] A. Curtiss, B. Rothrock, A. Bakar, N. Arora, J. Huang, Z. Englhardt, A.-P. Empedrado, C. Wang, S. Ahmed, Y. Zhang, N. Alshurafa, and J. Hester. “FaceBit: Smart Face Masks Platform”. en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.4 (Dec. 2021), pp. 1–44. ISSN: 2474-9567. DOI: 10.1145/3494991. URL: <https://dl.acm.org/doi/10.1145/3494991> (visited on 06/25/2024).
- [117] U. C. for Diseases Control and Prevention. *Interim Public Health Recommendations for Fully Vaccinated People*. <https://www.cdc.gov/coronavirus/2019-ncov/vaccines/fully-vaccinated-guidance.html>. Accessed: 2021-08-01. July 2021.
- [118] W. Post. ‘Act now’ on global vaccines to stop more-dangerous variants, experts warn Biden. <https://www.washingtonpost.com/health/2021/08/10/health-experts-demand-global-vaccines-pandemic/>. Accessed: 2021-08-01. Aug. 2021.
- [119] D. Pradhan, P. Biswasroy, P. K. Naik, G. Ghosh, and G. Rath. “A review of current interventions for COVID-19 prevention”. In: *Archives of medical research* 51.5 (2020), pp. 363–374.

- [120] S. Talic, S. Shah, H. Wild, D. Gasevic, A. Maharaj, Z. Ademi, X. Li, W. Xu, I. Mesa-Eguiagaray, J. Rostron, et al. “Effectiveness of public health measures in reducing the incidence of covid-19, SARS-CoV-2 transmission, and covid-19 mortality: systematic review and meta-analysis”. In: *bmj* 375 (2021).
- [121] C. R. MacIntyre, A. A. Chughtai, M. Kunasekaran, E. Tawfiq, and T. Greenhalgh. “The role of masks and respirators in preventing respiratory infections in healthcare and community settings”. In: *bmj* 388 (2025).
- [122] O. Safety and H. Administration. *1910.134 App A - Fit Testing Procedures (Mandatory). Personal Protective Equipment. Standard Number: 1910.134 App A*. 2019.
- [123] A. Regli, A. Sommerfield, and B. von Ungern-Sternberg. “The role of fit testing N95/FFP2/FFP3 masks: a narrative review”. In: *Anaesthesia* 76.1 (2021), pp. 91–100.
- [124] J. Gralton and M.-L. McLaws. “Protecting healthcare workers from pandemic influenza: N95 or surgical masks?” In: *Critical care medicine* 38.2 (2010), pp. 657–667.
- [125] M. Chu, T. Nguyen, V. Pandey, Y. Zhou, H. N. Pham, R. Bar-Yoseph, S. Radom-Aizik, R. Jain, D. M. Cooper, and M. Khine. “Respiration rate and volume measurements using wearable strain sensors”. In: *NPJ digital medicine* 2.1 (2019), p. 8.
- [126] A. Nikulin, D. Ikonnikov, and I. Dolzhikov. “Smart Personal Protective Equipment in the Coal Mining Industry”. In: *International Journal of Civil Engineering and Technology* 10.4 (2019), pp. 852–863.
- [127] S. M. Sánchez, R. C. Vara, F. J. G. Criado, S. R. González, J. P. Tejedor, and J. M. Corchado. “Smart PPE and CPE Platform for Electric Industry Workforce”. In: *International Workshop on Soft Computing Models in Industrial and Environmental Applications*. Springer. 2019, pp. 422–431.
- [128] F. Aloweni, S. Bouchoucha, A. Hutchinson, S. Ang, H. X. Toh, N. B. Suhari, R. N. B. Sunari, and S. H. Lim. “Health care workers’ experience of personal protective equipment use and associated adverse effects during the COVID-19 pandemic response in Singapore”. In: *Journal of Advanced Nursing* 78 (2022), pp. 2383–2396. DOI: 10.1111/jan.15164.
- [129] F. Kawsar, C. Min, A. Mathur, A. Montanari, U. G. Acer, and M. Van den Broeck. “eSense: Open earable platform for human sensing”. In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 2018, pp. 371–372.
- [130] K. Li, R. Zhang, B. Liang, F. Guimbretière, and C. Zhang. “Eario: A low-power acoustic sensing earable for continuously tracking detailed facial movements”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6.2 (2022), pp. 1–24.

- [131] A. Ferlini, A. Montanari, C. Min, H. Li, U. Sassi, and F. Kawsar. “In-ear PPG for vital signs”. In: *IEEE Pervasive Computing* 21.1 (2021), pp. 65–74.
- [132] C. Holz and E. J. Wang. “Glabella: Continuously Sensing Blood Pressure Behavior using an Unobtrusive Wearable Device”. en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (Sept. 2017), pp. 1–23. ISSN: 2474-9567. DOI: 10.1145/3132024. URL: <https://dl.acm.org/doi/10.1145/3132024> (visited on 06/25/2024).
- [133] J. Hernandez, D. McDuff, K. Quigley, P. Maes, and R. W. Picard. “Wearable motion-based heart rate at rest: A workplace evaluation”. In: *IEEE journal of biomedical and health informatics* 23.5 (2018), pp. 1920–1927.
- [134] J. R. Sempionatto, T. Nakagawa, A. Pavinatto, S. T. Mensah, S. Imani, P. Mercier, and J. Wang. “Eyeglasses based wireless electrolyte and metabolite sensor platform”. In: *Lab on a Chip* 17.10 (2017), pp. 1834–1842.
- [135] S. Zhang, Y. Zhao, D. T. Nguyen, R. Xu, S. Sen, J. Hester, and N. Alshurafa. “NeckSense: A Multi-Sensor Necklace for Detecting Eating Activities in Free-Living Conditions”. en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.2 (June 2020), pp. 1–26. ISSN: 2474-9567. DOI: 10.1145/3397313. URL: <https://dl.acm.org/doi/10.1145/3397313> (visited on 02/23/2023).
- [136] S. Choi, R. Park, N. Hur, and W. Kim. “Evaluation of wearing comfort of dust masks”. In: *Plos one* 15.8 (2020), e0237848.
- [137] W. Yim, D. Cheng, S. H. Patel, R. Kou, Y. S. Meng, and J. V. Jokerst. “KN95 and N95 respirators retain filtration efficiency despite a loss of dipole charge during decontamination”. In: *ACS applied materials & interfaces* 12.49 (2020), pp. 54473–54480.
- [138] D. Spruijt-Metz and W. Nilsen. “Dynamic models of behavior for just-in-time adaptive interventions”. In: *IEEE Pervasive Computing* 13.3 (2014), pp. 13–17.
- [139] D. R. Seshadri, R. T. Li, J. E. Voos, J. R. Rowbottom, C. M. Alfes, C. A. Zorman, and C. K. Drummond. “Wearable sensors for monitoring the physiological and biochemical profile of the athlete”. In: *NPJ digital medicine* 2.1 (2019), pp. 1–16.
- [140] M. T. Cooney, E. Vartiainen, T. Laakitainen, A. Juolevi, A. Dudina, and I. M. Graham. “Elevated resting heart rate is an independent risk factor for cardiovascular disease in healthy men and women”. In: *American heart journal* 159.4 (2010), pp. 612–619.
- [141] Y. Henderson. “The mass-movements of the circulation as shown by a recoil curve”. In: *American Journal of Physiology-Legacy Content* 14.3 (1905), pp. 287–298.

- [142] I. Starr, A. Rawson, H. Schroeder, and N. Joseph. “Studies on the estimation of cardiac output in man, and of abnormalities in cardiac function, from the heart’s recoil and the blood’s impacts; the ballistocardiogram”. In: *American Journal of Physiology-Legacy Content* 127.1 (1939), pp. 1–28.
- [143] C.-S. Kim, S. L. Ober, M. S. McMurtry, B. A. Finegan, O. T. Inan, R. Mukkamala, and J.-O. Hahn. “Ballistocardiogram: Mechanism and potential for unobtrusive cardiovascular health monitoring”. In: *Scientific reports* 6.1 (2016), pp. 1–6.
- [144] L. Giovangrandi, O. T. Inan, R. M. Wiard, M. Etemadi, and G. T. Kovacs. “Ballistocardiography—a method worth revisiting”. In: *2011 annual international conference of the IEEE engineering in medicine and biology society*. IEEE. 2011, pp. 4279–4282.
- [145] J. Hernandez, Y. Li, J. M. Rehg, and R. W. Picard. “Bioglass: Physiological parameter estimation using a head-mounted wearable device”. In: *2014 4th International Conference on Wireless Mobile Communication and Healthcare-Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MOBIHEALTH)*. IEEE. 2014, pp. 55–58.
- [146] J. Pan and W. J. Tompkins. “A real-time QRS detection algorithm”. In: *IEEE transactions on biomedical engineering* 3 (2007), pp. 230–236.
- [147] G. E. Carpagnano, M. P. Foschino-Barbaro, C. Crocetta, D. Lacedonia, V. Saliani, L. D. Zoppo, and P. J. Barnes. “Validation of the exhaled breath temperature measure: Reference values in healthy subjects”. In: *Chest* 151.4 (2017), pp. 855–860.
- [148] L. Suen, Y. P. Guo, S. S. Ho, C. H. Au-Yeung, and S. Lam. “Comparing mask fit and usability of traditional and nanofibre N95 filtering facepiece respirators before and after nursing procedures”. In: *Journal of Hospital Infection* 104.3 (2020), pp. 336–343.
- [149] L. Hackett, M. M. Zhang, M. Casey, J. Miller, J. Smith, C. Low, E. Aldridge, P. J. Owen, and P. Buntine. “N-95/P2 respirator compliance with fit testing recommendations and respirator satisfaction amongst hospital staff”. In: *Infection, Disease & Health* 29.3 (2024), pp. 144–151.
- [150] S. Rengasamy, B. C. Eimer, and J. Szalajda. “A quantitative assessment of the total inward leakage of NaCl aerosol representing submicron-size bioaerosol through N95 filtering facepiece respirators and surgical masks”. In: *Journal of occupational and environmental hygiene* 11.6 (2014), pp. 388–396.
- [151] R. G. Carroll. “9 - Integrated Cardiovascular Function”. In: *Elsevier’s Integrated Physiology*. Ed. by R. G. Carroll. Philadelphia: Mosby, 2007, pp. 91–98. ISBN: 978-0-323-04318-2. DOI: <https://doi.org/10.1016/B978-0-323-04318-2.50015-7>. URL: <https://www.sciencedirect.com/science/article/pii/B9780323043182500157>.

- [152] P. van Gent, H. Farah, N. Nes, and B. Arem. “Heart Rate Analysis for Human Factors: Development and Validation of an Open Source Toolkit for Noisy Naturalistic Heart Rate Data”. In: *Proceedings of the 6th HUMANIST Conference*. June 2018, pp. 173–178.
- [153] A. M. Carek, J. Conant, A. Joshi, H. Kang, and O. T. Inan. “SeismoWatch: Wearable Cuffless Blood Pressure Monitoring Using Pulse Transit Time”. en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (Sept. 2017), pp. 1–16. ISSN: 2474-9567. DOI: 10.1145/3130905. URL: <https://dl.acm.org/doi/10.1145/3130905> (visited on 02/22/2023).
- [154] M. Noura, M. Atiquzzaman, and M. Gaedke. “Interoperability in Internet of Things: Taxonomies and Open Challenges”. en. In: *Mobile Networks and Applications* 24.3 (June 2019), pp. 796–809. ISSN: 1572-8153. DOI: 10.1007/s11036-018-1089-9. URL: <https://doi.org/10.1007/s11036-018-1089-9> (visited on 02/09/2024).
- [155] T. Skála, M. Vícha, M. Rada, J. Vácha, J. Flašík, and M. Táboršký. “Feasibility of evaluation of Polar H10 chest-belt ECG in patients with a broad range of heart conditions”. In: *Cor Vasa* 64.4 (2022), pp. 411–422.
- [156] M. Schaffarczyk, B. Rogers, R. Reer, and T. Gronwald. “Validity of the polar H10 sensor for heart rate variability analysis during resting state and incremental exercise in recreational men and women”. In: *Sensors* 22.17 (2022), p. 6536.
- [157] C. Wall, Y. Celik, V. Hetherington, P. McMeekin, R. Walker, L. Graham, R. Vitorio, and A. Godfrey. “Considering and understanding developmental and deployment barriers for wearable technologies in neurosciences”. English. In: *Frontiers in Neuroscience* 18 (Mar. 2024). Publisher: Frontiers. ISSN: 1662-453X. DOI: 10.3389/fnins.2024.1379619. URL: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2024.1379619/full> (visited on 06/14/2025).
- [158] W. Chen, S. Lin, E. Thompson, and J. Stankovic. “SenseCollect: We Need Efficient Ways to Collect On-body Sensor-based Human Activity Data!” en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (Sept. 2021), pp. 1–27. ISSN: 2474-9567. DOI: 10.1145/3478119. URL: <https://dl.acm.org/doi/10.1145/3478119> (visited on 05/16/2022).
- [159] M. Nebeling. “Playing the Tricky Game of Toolkits Research”. en. In: *Workshop on HCI. Tools at CHI*. 2017.
- [160] D. Ledo, S. Houben, J. Vermeulen, N. Marquardt, L. Oehlberg, and S. Greenberg. “Evaluation Strategies for HCI Toolkit Research”. en. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–17. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173610. URL: <https://dl.acm.org/doi/10.1145/3173574.3173610> (visited on 03/06/2023).

- [161] Z. Hassan, C. Treude, M. Norrish, G. Williams, and A. Potanin. “Characterising reproducibility debt in scientific software: A systematic literature review”. en. In: *Journal of Systems and Software* 222 (Apr. 2025), p. 112327. ISSN: 01641212. DOI: 10.1016/j.jss.2024.112327. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0164121224003716> (visited on 12/16/2025).
- [162] B. Rothrock, A. Curtiss, J. Bai, and J. Hester. “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*. UbiComp/ISWC ’22 Adjunct. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 379–385. ISBN: 978-1-4503-9423-9. DOI: 10.1145/3544793.3560399. URL: <https://dl.acm.org/doi/10.1145/3544793.3560399> (visited on 02/14/2024).
- [163] B. Rothrock, A. Curtiss, T. Cohen, and J. Hester. “PhysioHub: A Wirelessly Synchronized Multi-Node Wearable System”. In: *Proceedings of the 2026 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2026.
- [164] I. Lee, A. Surendran, S. Fleury, I. Gimino, A. Curtiss, C. Fell, D. J. Shiwerski, O. Refy, B. Rothrock, S. Jo, T. Schwartzkopff, A. S. Mehta, Y. Wang, A. Sipe, S. John, X. Ji, G. Nikiforidis, A. W. Feinberg, J. Hester, D. J. Weber, O. Veiseh, J. Rivnay, and T. Cohen-Karni. “Electrocatalytic on-site oxygenation for transplanted cell-based-therapies”. en. In: *Nature Communications* 14.1 (Nov. 2023). Publisher: Nature Publishing Group, p. 7019. ISSN: 2041-1723. DOI: 10.1038/s41467-023-42697-2. URL: <https://www.nature.com/articles/s41467-023-42697-2> (visited on 12/16/2025).
- [165] Y. Hutabarat, D. Owaki, and M. Hayashibe. “Recent Advances in Quantitative Gait Analysis Using Wearable Sensors: A Review”. In: *IEEE Sensors Journal* 21.23 (Dec. 2021), pp. 26470–26487. ISSN: 1558-1748. DOI: 10.1109/JSEN.2021.3119658. URL: <https://ieeexplore.ieee.org/document/9568979/> (visited on 02/18/2026).
- [166] D. Kumar, S. Jeuris, J. E. Bardram, and N. Dragoni. “Mobile and Wearable Sensing Frameworks for mHealth Studies and Applications: A Systematic Review”. In: *ACM Transactions on Computing for Healthcare* 2.1 (Dec. 2021), 8:1–8:28. ISSN: 2691-1957. DOI: 10.1145/3422158. URL: <https://doi.org/10.1145/3422158> (visited on 02/07/2023).
- [167] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas. “EmotionSense: a mobile phones based adaptive platform for experimental social psychology research”. en. In: *Proceedings of the 12th ACM international conference on Ubiquitous computing*. Copenhagen Denmark: ACM, Sept. 2010, pp. 281–290. ISBN: 978-1-60558-843-8. DOI: 10.1145/1864349.1864393. URL: <https://dl.acm.org/doi/10.1145/1864349.1864393> (visited on 02/08/2023).

- [168] J. Hicks, N. Ramanathan, D. Kim, M. Monibi, J. Selsky, M. Hansen, and D. Estrin. “AndWellness: an open mobile system for activity and experience sampling”. en. In: *Wireless Health 2010*. San Diego California: ACM, Oct. 2010, pp. 34–43. ISBN: 978-1-60558-989-3. DOI: 10.1145/1921081.1921087. URL: <https://dl.acm.org/doi/10.1145/1921081.1921087> (visited on 02/08/2023).
- [169] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. “The Jigsaw continuous sensing engine for mobile phone applications”. en. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. Zürich Switzerland: ACM, Nov. 2010, pp. 71–84. ISBN: 978-1-4503-0344-6. DOI: 10.1145/1869983.1869992. URL: <https://dl.acm.org/doi/10.1145/1869983.1869992> (visited on 02/08/2023).
- [170] D. Ferreira, V. Kostakos, and A. K. Dey. “AWARE: Mobile Context Instrumentation Framework”. en. In: *Frontiers in ICT* 2 (Apr. 2015). ISSN: 2297-198X. DOI: 10.3389/fict.2015.00006. URL: <http://journal.frontiersin.org/article/10.3389/fict.2015.00006/abstract> (visited on 08/04/2022).
- [171] J.-P. Onnela, C. Dixon, K. Griffin, T. Jaenicke, L. Minowada, S. Esterkin, A. Siu, J. Zagorsky, and E. Jones. “Beiwe: A data collection platform for high-throughput digital phenotyping”. en. In: *Journal of Open Source Software* 6.68 (Dec. 2021), p. 3417. ISSN: 2475-9066. DOI: 10.21105/joss.03417. URL: <https://joss.theoj.org/papers/10.21105/joss.03417> (visited on 02/07/2023).
- [172] Y. Ranjan, Z. Rashid, C. Stewart, P. Conde, M. Begale, D. Verbeeck, S. Boettcher, The Hyve, R. Dobson, A. Folarin, and The RADAR-CNS Consortium. “RADAR-Base: Open Source Mobile Health Platform for Collecting, Monitoring, and Analyzing Data Using Sensors, Wearables, and Mobile Devices”. en. In: *JMIR mHealth and uHealth* 7.8 (Aug. 2019), e11734. ISSN: 2291-5222. DOI: 10.2196/11734. URL: <https://mhealth.jmir.org/2019/8/e11734/> (visited on 05/16/2022).
- [173] H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber. “Sensus: a cross-platform, general-purpose system for mobile crowdsensing in human-subject studies”. en. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Heidelberg Germany: ACM, Sept. 2016, pp. 415–426. ISBN: 978-1-4503-4461-6. DOI: 10.1145/2971648.2971711. URL: <https://dl.acm.org/doi/10.1145/2971648.2971711> (visited on 02/08/2023).
- [174] H. Tangmunarunkit, C. K. Hsieh, B. Longstaff, S. Nolen, J. Jenkins, C. Ketcham, J. Selsky, F. Alquaddoomi, D. George, J. Kang, Z. Khalapyan, J. Ooms, N. Ramanathan, and D. Estrin. “Ohmage: A General and Extensible End-to-End Participatory Sensing Platform”. en. In: *ACM Transactions on Intelligent Systems and Technology* 6.3 (May 2015), pp. 1–21. ISSN: 2157-6904, 2157-6912. DOI: 10.1145/2717318. URL: <https://dl.acm.org/doi/10.1145/2717318> (visited on 02/08/2023).
- [175] C.-K. Hsieh, H. Tangmunarunkit, F. Alquaddoomi, J. Jenkins, J. Kang, C. Ketcham, B. Longstaff, J. Selsky, B. Dawson, D. Swendeman, D. Estrin, and N. Ramanathan.

- “Lifestreams: a modular sense-making toolset for identifying important patterns from everyday life”. en. In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. Roma Italy: ACM, Nov. 2013, pp. 1–13. ISBN: 978-1-4503-2027-6. DOI: 10.1145/2517351.2517368. URL: <https://dl.acm.org/doi/10.1145/2517351.2517368> (visited on 02/08/2023).
- [176] P. Wu, J. Zhu, and J. Y. Zhang. “MobiSens: A Versatile Mobile Sensing Platform for Real-World Applications”. en. In: *Mobile Networks and Applications* 18.1 (Feb. 2013), pp. 60–80. ISSN: 1383-469X, 1572-8153. DOI: 10.1007/s11036-012-0422-y. URL: <http://link.springer.com/10.1007/s11036-012-0422-y> (visited on 02/08/2023).
- [177] W. Morrison, L. Guerdan, J. Kanugo, T. Trull, and Y. Shang. “TigerAware: An Innovative Mobile Survey and Sensor Data Collection and Analytics System”. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. June 2018, pp. 115–122. DOI: 10.1109/DSC.2018.00025.
- [178] S. M. Hossain, T. Hnat, N. Saleheen, N. J. Nasrin, J. Noor, B.-J. Ho, T. Condie, M. Srivastava, and S. Kumar. “mCerebrum: A Mobile Sensing Software Platform for Development and Validation of Digital Biomarkers and Interventions”. en. In: *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. Delft Netherlands: ACM, Nov. 2017, pp. 1–14. ISBN: 978-1-4503-5459-2. DOI: 10.1145/3131672.3131694. URL: <https://dl.acm.org/doi/10.1145/3131672.3131694> (visited on 05/16/2022).
- [179] J. E. Bardram. *The CARP Mobile Sensing Framework – A Cross-platform, Reactive, Programming Framework and Runtime Environment for Digital Phenotyping*. arXiv:2006.11904 [cs]. June 2020. URL: <http://arxiv.org/abs/2006.11904> (visited on 01/18/2023).
- [180] T. Röddiger, M. Küttner, P. Lepold, T. King, D. Moschina, O. Bagge, J. A. Paradiso, C. Clarke, and M. Beigl. “OpenEarable 2.0: Open-Source Earphone Platform for Physiological Ear Sensing”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9.1 (Mar. 2025), 16:1–16:33. DOI: 10.1145/3712069. URL: <https://dl.acm.org/doi/10.1145/3712069> (visited on 11/19/2025).
- [181] A. Watson, C. Kendell, A. Lingamoorthy, I. Lee, and J. Weimer. “Lumos: An Open-Source Device for Wearable Spectroscopy Research”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6.4 (Jan. 2023), 187:1–187:24. DOI: 10.1145/3569502. URL: <https://dl.acm.org/doi/10.1145/3569502> (visited on 11/18/2025).
- [182] F. X. Lin, A. Rahmati, and L. Zhong. “Dandelion: a framework for transparently programming phone-centered wireless body sensor applications for health”. en. In: *Wireless Health 2010*. San Diego California: ACM, Oct. 2010, pp. 74–83. ISBN: 978-1-60558-989-3. DOI: 10.1145/1921081.1921091. URL: <https://dl.acm.org/doi/10.1145/1921081.1921091> (visited on 02/08/2023).

- [183] Y. Lee, S. S. Iyengar, C. Min, Y. Ju, S. Kang, T. Park, J. Lee, Y. Rhee, and J. Song. “MobiCon: a mobile context-monitoring platform”. en. In: *Communications of the ACM* 55.3 (Mar. 2012), pp. 54–65. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/2093548.2093567. URL: <https://dl.acm.org/doi/10.1145/2093548.2093567> (visited on 02/08/2023).
- [184] C. Min, C. Yoo, Y. Lee, and J. Song. “Healthopia: towards your well-being in everyday life”. en. In: ().
- [185] R. Chaudhri, W. Brunette, B. Hemingway, and G. Borriello. “ODK sensors: an application-level sensor framework for Android devices”. en. In: *Proceedings of the 3rd ACM Symposium on Computing for Development*. Bangalore India: ACM, Jan. 2013, pp. 1–2. ISBN: 978-1-4503-1856-3. DOI: 10.1145/2442882.2442918. URL: <https://dl.acm.org/doi/10.1145/2442882.2442918> (visited on 02/08/2023).
- [186] W. Brunette, R. Sodt, R. Chaudhri, M. Goel, M. Falcone, J. Van Orden, and G. Borriello. “Open data kit sensors: a sensor integration framework for android at the application-level”. en. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. Low Wood Bay Lake District UK: ACM, June 2012, pp. 351–364. ISBN: 978-1-4503-1301-8. DOI: 10.1145/2307636.2307669. URL: <https://dl.acm.org/doi/10.1145/2307636.2307669> (visited on 02/08/2023).
- [187] J. H. Lim, A. Zhan, E. Goldschmidt, J. Ko, M. Chang, and A. Terzis. “HealthOS: a platform for pervasive health applications”. en. In: *Proceedings of the Second ACM Workshop on Mobile Systems, Applications, and Services for HealthCare*. Toronto Ontario Canada: ACM, Nov. 2012, pp. 1–6. ISBN: 978-1-4503-1764-1. DOI: 10.1145/2396276.2396281. URL: <https://dl.acm.org/doi/10.1145/2396276.2396281> (visited on 02/08/2023).
- [188] R. Tauman, M. Berall, R. Berry, T. Etzioni, N. Shrater, D. Hwang, I. Marai, P. Manthena, A. Rama, R. Spiegel, et al. “Watch-PAT is useful in the diagnosis of sleep apnea in patients with atrial fibrillation”. In: *Nature and science of sleep* (2020), pp. 1115–1121.
- [189] J. Espina, T. Falck, J. Muehlsteff, and X. Aubert. “Wireless body sensor network for continuous cuff-less blood pressure monitoring”. In: *2006 3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors*. IEEE. 2006, pp. 11–15.
- [190] A. M. Carek, J. Conant, A. Joshi, H. Kang, and O. T. Inan. “SeismoWatch: wearable cuffless blood pressure monitoring using pulse transit time”. In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1.3 (2017), pp. 1–16.
- [191] M. Di Rienzo, G. Rizzo, Z. M. Işilay, and P. Lombardi. “SeisMote: a multi-sensor wireless platform for cardiovascular monitoring in laboratory, daily life, and telemedicine”. In: *Sensors* 20.3 (2020), p. 680.

- [192] C. Wright, A. Surendran, I. Lee, J. E. Villacres, A. Ezerins, A. Curtiss, N. Brown, J. Liu, B. Rothrock, P. Rios, et al. “Design of a wireless, fully implantable platform for in-situ oxygenation of encapsulated cell therapies”. In: *Device* 4.5 (2026).
- [193] E. Greenlee, B. Rothrock, H. Kim, E. Zegura, and J. Hester. ““The Devil You Know”: Barriers and Opportunities for Co-Designing Microclimate Sensors, A Case Study of Manoomi”. en. In: *ACM Journal on Computing and Sustainable Societies* 2.3 (Sept. 2024), pp. 1–30. ISSN: 2834-5533. DOI: 10.1145/3685695. URL: <https://dl.acm.org/doi/10.1145/3685695> (visited on 06/07/2025).
- [194] K. M. Cobb, R. Clark, N. Deffley, R. Mathews, E. D. Lorenzo, L. Polepeddi, A. Chavan, T. Cone, J. Koval, A. Robel, A. Hyde, J. M. L. Ramirez, I. Tien, K. Park, M. McSorley, D. Shabaka, and M. McClain. “Research to Action Frameworks for Equitable Coastal Resilience: A Case Study from Savannah, Georgia”. In: AMS, Jan. 2022. URL: <https://ams.confex.com/ams/102ANNUAL/meetingapp.cgi/Paper/399502> (visited on 07/21/2023).
- [195] P. David, L. David, H. K. Stark, K. J. Stark, S. N.-A. Fahrlander, and J. M. Schlender. *Manoomin (Wild Rice) Seeding Guidelines*. Tech. rep. Version 1.0. Great Lakes Indian Fish and Wildlife Commission, 2019, p. 158. URL: <https://docslib.org/doc/1913211/manoomin-wild-rice-seeding-guidelines> (visited on 07/20/2023).
- [196] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. “Wireless sensor networks for habitat monitoring”. In: *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. WSNA '02. New York, NY, USA: Association for Computing Machinery, Sept. 2002, pp. 88–97. ISBN: 978-1-58113-589-3. DOI: 10.1145/570738.570751. URL: <https://dl.acm.org/doi/10.1145/570738.570751> (visited on 07/20/2023).
- [197] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet”. In: *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. ASPLOS X. New York, NY, USA: Association for Computing Machinery, Oct. 2002, pp. 96–107. ISBN: 978-1-58113-574-9. DOI: 10.1145/605397.605408. URL: <https://dl.acm.org/doi/10.1145/605397.605408> (visited on 07/20/2023).
- [198] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. “Deploying a wireless sensor network on an active volcano”. In: *IEEE internet computing* 10.2 (2006), pp. 18–25.
- [199] D. M. Doolin and N. Sitar. “Wireless sensors for wildfire monitoring”. In: *Smart structures and materials 2005: sensors and smart structures technologies for civil, mechanical, and aerospace systems*. Vol. 5765. SPIE. 2005, pp. 477–484.

- [200] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman. “{FarmBeats}: an {IoT} platform for {Data-Driven} agriculture”. In: *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. 2017, pp. 515–529.
- [201] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail. “Energy-efficient wireless sensor networks for precision agriculture: A review”. In: *Sensors* 17.8 (2017), p. 1781.
- [202] M. Pule, A. Yahya, and J. Chuma. “Wireless sensor networks: A survey on monitoring water quality”. In: *Journal of applied research and technology* 15.6 (2017), pp. 562–570.
- [203] Y. Kang, L. Aye, T. D. Ngo, and J. Zhou. “Performance evaluation of low-cost air quality sensors: A review”. In: *Science of The Total Environment* 818 (2022), p. 151769.
- [204] F. Mao, K. Khamis, S. Krause, J. Clark, and D. M. Hannah. “Low-Cost Environmental Sensor Networks: Recent Advances and Future Directions”. English. In: *Frontiers in Earth Science* 7 (Sept. 2019). Publisher: Frontiers. ISSN: 2296-6463. DOI: 10.3389/feart.2019.00221. URL: <https://www.frontiersin.org/journals/earth-science/articles/10.3389/feart.2019.00221/full> (visited on 12/18/2025).
- [205] T. Speaker, S. O’Donnell, G. Wittemyer, B. Bruyere, C. Loucks, A. Dancer, M. Carter, E. Fegraus, J. Palmer, E. Warren, et al. “A global community-sourced assessment of the state of conservation technology”. In: *Conservation Biology* 36.3 (2022), e13871.
- [206] C. E. Catlett, P. H. Beckman, R. Sankaran, and K. K. Galvin. “Array of things: a scientific research instrument in the public way: platform design and early lessons learned”. In: *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*. SCOPE ’17. New York, NY, USA: Association for Computing Machinery, Apr. 2017, pp. 26–33. ISBN: 978-1-4503-4989-5. DOI: 10.1145/3063386.3063771. URL: <https://dl.acm.org/doi/10.1145/3063386.3063771> (visited on 06/25/2025).
- [207] C. Catlett, P. Beckman, N. Ferrier, M. E. Papka, R. Sankaran, J. Solin, V. Taylor, D. Pancoast, and D. Reed. “Hands-on computer science: the array of things experimental urban instrument”. In: *Computing in Science & Engineering* 24.1 (2022), pp. 57–63.
- [208] M. I. G. Daep, A. Cabral, T. M. Werner, R. Mansour, C. Catlett, A. Roseway, C. Needham, N. Udeagbala, and S. Counts. “The “Three-Legged Stool”: Designing for Equitable City, Community, and Research Partnerships in Urban Environmental Sensing”. en. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–19. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3581289. URL: <https://dl.acm.org/doi/10.1145/3544548.3581289> (visited on 06/07/2025).

- [209] C. Shapiro and H. R. Varian. *Information rules: A strategic guide to the network economy*. Harvard Business Press, 1999.
- [210] A. Sørensen, H. Wang, M. J. Remy, N. Kjettrup, R. B. Sørensen, J. J. Nielsen, P. Popovski, and G. C. Madueño. “Modeling and experimental validation for battery lifetime estimation in nb-iot and lte-m”. In: *IEEE Internet of Things Journal* 9.12 (2022), pp. 9804–9819.
- [211] E. Greenlee, B. Rothrock, C. Atchison, C. McNerney, E. Zegura, J. Hester, and A. Cabral. “Buoy Meets World: Aquatic Sensing Insights from 169 Days of Low-cost Wetland Buoy Deployments”. In: *Proceedings of the 2026 International Conference on ICT for Sustainability (ICT4S)*. Bern, Switzerland: IEEE, June 2026.
- [212] J. K. Huber. *Palynological investigations related to archaeological sites and the expansion of wild rice (Zizania aquatica L.) in northeast Minnesota*. University of Minnesota, 2001.
- [213] E. Benton-Banai. *The Mishomis book: The voice of the Ojibway*. University of Minnesota Press, 2010. ISBN: 9780816673827.
- [214] A. D. Drewes and J. Silbernagel. “Uncovering the spatial dynamics of wild rice lakes, harvesters and management across Great Lakes landscapes for shared regional conservation”. In: *Ecological Modelling* 229 (2012), pp. 97–107.
- [215] M. Nyblade, D. J. Larkin, D. Vogt, R. Croll, G.-H. C. Ng, J. Graveen, K. Hansen, H. Panci, B. Byrne, B. Panek, et al. “CLIMATE CHANGE CONTRIBUTES TO LOSS OF WILD RICE (MANOOMIN/PSIN), THREATENING INDIGENOUS LIFEWAYS”. In: *First We Must Consider Manoomin/Psin: Impacts of Climate and Land Cover Change on* (2023), p. 47.
- [216] A. E. Jenks. *The wild rice gatherers of the upper lakes: a study in American primitive economics*. Vol. 19. US Government Printing Office, 1901.
- [217] M. Stults, S. Petersen, J. Bell, W. Baule, E. Nasser, E. Gibbons, and M. Fougerat. *Climate Change Vulnerability Assessment and Adaptation Plan: 1854 Ceded Territory Including the Bois Forte, Fond du Lac, and Grand Portage Reservations*. Tech. rep. Duluth, MN: 1854 Ceded Territory, 2016, p. 146.
- [218] H. Panci, M. Montano, A. Shultz, T. Bartnick, and K. Stone. *Climate Change Vulnerability Assessment, Version 1*. Tech. rep. Great Lakes Indian Fish and Wildlife Commission (GLIFWC), 2018, p. 30.
- [219] P. David et al. “Manoomin”. In: *Great Lakes Indian Fish and Wildlife Commission 1* (2019). Version 1.0, pp. 1–158.

- [220] J. Smith. “Rights of Nature and Tribal Sovereignty: Protecting Natural Communities, Wild Rice, and Salmon in the United States”. In: *Maryland Journal of International Law* 38.1 (2023), p. 11.
- [221] S. Raman. *Culture and conservation thrive as Great Lakes tribes bring back native wild rice — news.mongabay.com*. <https://news.mongabay.com/2024/03/culture-and-conservation-thrive-as-great-lakes-tribes-bring-back-native-wild-rice/>. 2024.
- [222] H. R. Hosterman, K. Ritter, N. Schuldt, D. Vogt, D. M. Erickson, O. L. Griot, E. Johnston, K. Schmidt, E. Ravindran, R. D. LaBine, et al. “Lake Superior Manoomin cultural and ecosystem characterization study”. In: *Ecology and Society* 28.3 (2023).
- [223] K. R. Prihoda, A. Frie, A. Schrank, and D. R. Schreiner. “Workshop Summary: Habitat Management in the St. Louis River Estuary”. In: (2022).
- [224] N. Oceanic and O. f. C. M. Atmospheric Administration. *Restoring and Monitoring Lake Superior Coastal Wetland Manoomin*. <https://coast.noaa.gov/digitalcoast/stories/manoomin.html>. 2018.
- [225] L. McGilp, C. Castell-Miller, M. Haas, R. Millas, and J. Kimball. “Northern Wild Rice (*Zizania palustris* L.) breeding, genetics, and conservation”. In: *Crop Science* 63.4 (2023), pp. 1904–1933.
- [226] M. Nyblade. “First We Must Consider Manoomin/Psin: Impacts of Climate and Land Cover Change on Wild Rice”. PhD thesis. University of Minnesota, 2023.
- [227] K. S. Whitney. “Manoomin: The taming of wild rice in the Great Lakes Region”. In: *Arcadia* (2015).
- [228] M. Banner. “Enbridge Line 3 Impact on Wild Rice Lakes in Minnesota Using GIS and Remote Sensing”. In: *Voices of Indigenuity* (2023).
- [229] A. K. Waheed. “Manoomin (wild rice) and environmental change at a significant river system of the Lac du Flambeau Band of Lake Superior Chippewa”. MA thesis. University of Minnesota, 2021.
- [230] J. Gurneau, C. O’Brien, V. A. Rivera, A. Curtis, M. O’Connell, A. Cottrell, N. Tillison, J. Gilbert, J. Coleman, R. Sankaran, et al. “In Situ Sensing of Manoomin (Wild Rice) Habitat to Create a Voice for Plant Relatives”. In: *AGU Fall Meeting Abstracts*. Vol. 2022. 2022, ED42A–02.
- [231] M. A. Davenport, B. Duever, A. Kadrie, E. Green, J. Peplinski, L. Matson, M. Dockry, and G.-H. C. Ng. *Protecting the Harvest: A survey of Minnesota state-permitted (non-tribal) wild rice harvesters*. 2020.

- [232] A. Cho, R. G. Herrera, L. Chaidez, and A. Uriostegui. "The " Comadre" Project: An Asset-Based Design Approach to Connecting Low-Income Latinx Families to Out-of-School Learning Opportunities". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–14.
- [233] M. Wong-Villacres, C. DiSalvo, N. Kumar, and B. DiSalvo. "Culture in action: Unpacking capacities to inform assets-based design". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14.
- [234] G. Hope Chidziwisano, A. Mariakakis, S. Wyche, V. Mafeni, and E. Gideon Banda. "Nkhukuprobe: Using a sensor-based technology probe to support poultry farming activities in malawi". In: *Proceedings of the 4th ACM SIGCAS Conference on Computing and Sustainable Societies*. 2021, pp. 275–287.
- [235] T. Wilson, H. Mthoko, Y. Adnan, and S. Till. "The Devil is in the deployment: Lessons learned while deploying an AI and IoT-enabled hydroponics grow tent with rural subsistence farmers in South Africa". In: *Proceedings of the 7th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. 2024, pp. 318–329.
- [236] E. Oduor, P. Waweru, J. Lenchner, and C. Neustaedter. "Practices and technology needs of a network of farmers in Tharaka Nithi, Kenya". In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–11.
- [237] M. E. Duarte, M. Vigil-Hayes, E. Zegura, E. Belding, I. Masara, and J. C. Nevarez. "As a squash plant grows: social textures of sparse internet connectivity in rural and tribal communities". In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 28.3 (2021), pp. 1–16.
- [238] S. M. Covey, S. R. Covey, and R. R. Merrill. *The speed of trust: The one thing that changes everything*. Simon and schuster, 2008.
- [239] S. R. Carroll, D. Rodriguez-Lonebear, and A. Martinez. "Indigenous data governance: strategies from United States native nations". In: *Data science journal* 18 (2019), p. 31.
- [240] T. Kukutai, S. Russo Carroll, and M. Walter. "Indigenous data sovereignty". In: (2020).
- [241] M. Walter, T. Kukutai, S. R. Carroll, and D. Rodriguez-Lonebear. *Indigenous data sovereignty and policy*. Taylor & Francis, 2021.
- [242] S. R. Carroll, I. Garba, O. L. Figueroa-Rodríguez, J. Holbrook, R. Lovett, S. Materechera, M. Parsons, K. Raseroka, D. Rodriguez-Lonebear, R. Rowe, et al. "The CARE principles for indigenous data governance". In: *Open Scholarship Press Curated Volumes: Policy* (2023).

- [243] D. Schallmo, C. A. Williams, and K. Lang. “An integrated design thinking approach-literature review, basic principles and roadmap for design thinking”. In: *ISPIM innovation symposium*. The International Society for Professional Innovation Management (ISPIM). 2018, pp. 1–18.
- [244] M. Wong-Villacres, A. Gautam, W. Roldan, L. Pei, J. Dickinson, A. Ismail, B. DiSalvo, N. Kumar, T. Clegg, S. Erete, et al. “From needs to strengths: Operationalizing an assets-based design of technology”. In: *Companion Publication of the 2020 Conference on Computer Supported Cooperative Work and Social Computing*. 2020, pp. 527–535.
- [245] T. C. Stevenson. “New description of box for holding thermometers”. In: *J. Scottish Meteorol. Soc* 1 (1864), p. 122.
- [246] J. A. McKenna, G. Buraglia, J. Kolakaluri, R. Baker-Ramos, S. J. Carter, J. Graveen, J. Gilbert, J. Rasmussen, B. Byrne, D. Vogt, et al. “Noondawind: Co-Designed Dashboard for Indigenous Data Access and Environmental Policy Implementation”. In: *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. 2026, pp. 1–25.
- [247] J. Lavallee. “Asemaa Nitam (Tobacco First): Indigenous Healing Practices within the Ojibwe Tradition”. PhD thesis. University of Toronto (Canada), 2022.
- [248] T. Mangla, E. H. Showalter, V. Adarsh, K. Jones, M. Vigil-Hayes, E. Belding-Royer, and E. Zegura. “A tale of three datasets”. In: *Communications of the ACM* 65 (2022), pp. 67–74. DOI: 10.1145/3462672.
- [249] B. Palacios Abad, E. Belding, M. Vigil-Hayes, and E. Zegura. “Note: Towards community-empowered network data action”. In: *Proceedings of the 5th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. 2022, pp. 585–588.
- [250] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. “Basic concepts and taxonomy of dependable and secure computing”. In: *IEEE transactions on dependable and secure computing* 1.1 (2004), pp. 11–33.
- [251] L. Lanzieri, G. Martino, G. Fey, H. Schlarb, T. C. Schmidt, and M. Wählich. “A review of techniques for ageing detection and monitoring on embedded systems”. In: *ACM computing surveys* 57.1 (2024), pp. 1–34.
- [252] M. A. Solouki, S. Angizi, and M. Violante. “Dependability in embedded systems: a survey of fault tolerance methods and software-based mitigation techniques”. In: *IEEE Access* 12 (2024), pp. 180939–180967.
- [253] E. Hollnagel. *Resilience engineering in practice: A guidebook*. Ashgate Publishing, Ltd., 2013.

- [254] A. LimoncelliThomas et al. “Resilience engineering: learning to embrace failure”. In: *Queue* (2012).
- [255] E. A. de Carvalho, J. O. Gomes, A. Jatobá, M. F. da Silva, and P. V. R. de Carvalho. “Employing resilience engineering in eliciting software requirements for complex systems: experiments with the functional resonance analysis method (FRAM)”. In: *Cognition, Technology & Work* 23.1 (2021), pp. 65–83.
- [256] J. Ning, R. Wang, and J. Zhou. “Research on a Mobile Hydrological Detection System Based on BDS”. In: *Proceedings of the 3rd International Conference on Electronic Information Technology and Smart Agriculture*. ICEITSA ’23. Sanya, China: Association for Computing Machinery, 2024, pp. 45–50. ISBN: 9798400716775. DOI: 10.1145/3641343.3641351. URL: <https://doi.org/10.1145/3641343.3641351>.
- [257] B. Krupp, J. Fagert, C. Fan, I. Zhang, and D. Hawkins. “AQIoT: Partnering with Community Organizations to Provide Fine-Grained Air Quality Data”. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Spatial Intelligence for Smart and Connected Communities*. 2025, pp. 25–34.
- [258] D. D. Clark. *Interoperation, open interfaces, and protocol architecture*. Tech. rep. mimeo, Laboratory of Computer Science, Massachusetts Institute of Technology, 1995.
- [259] D. Clark. “The design philosophy of the DARPA Internet protocols”. In: *Symposium proceedings on Communications architectures and protocols*. 1988, pp. 106–114.
- [260] J. Postel. *Transmission control protocol*. Tech. rep. The Defense Advanced Research Projects Agency, 1981.
- [261] *Chapter 9: Packet Dissection — wireshark.org*. https://www.wireshark.org/docs/wsdg_html_chunked/{C}hapter{D}issection.html. [Accessed 30-05-2026].
- [262] *Web of Things (WoT) Thing Description 1.1*. <https://www.w3.org/TR/wot-thing-description11/>. [Accessed 28-05-2026].
- [263] Z. Shelby. *Constrained RESTful Environments (CoRE) Link Format*. RFC 6690. Aug. 2012. DOI: 10.17487/RFC6690. URL: <https://www.rfc-editor.org/info/rfc6690>.
- [264] R. T. Fielding. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [265] Z. Shelby, K. Hartke, and C. Bormann. *The constrained application protocol (CoAP)*. Tech. rep. 2014.
- [266] S. G. Burman, J. Gao, G. B. Pasternack, N. A. Fangue, P. Cadrett, E. Campbell, and D. Ghosal. “Tempmesh—a flexible wireless sensor network for monitoring river temperatures”. In: *ACM Transactions on Sensor Networks* 19.1 (2022), pp. 1–28.

- [267] C. S. Hendrix. “The streetlight effect in climate change research on Africa”. In: *Global Environmental Change* 43 (2017), pp. 137–147.
- [268] M. Battaglia and M. A. Atkinson. “The streetlight effect in type 1 diabetes”. In: *Diabetes* 64.4 (2015), pp. 1081–1090.
- [269] J. Hoelzemann, G. Manso, A. Nagaraj, and M. Tranchero. *The streetlight effect in data-driven exploration*. Tech. rep. National Bureau of Economic Research, 2024.
- [270] A.-T. Tran, G. Guo, J. Taylor, K. A. Chan, E. L. Raymond, and C. DiSalvo. “Situating Datasets: Making Public Eviction Data Actionable for Housing Justice”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–16.
- [271] J. E. Hannay, C. MacLeod, J. Singer, H. P. Langtangen, D. Pfahl, and G. Wilson. “How do scientists develop and use scientific software?” In: *2009 ICSE workshop on software engineering for computational science and engineering*. Ieee. 2009, pp. 1–8.
- [272] F. Goth, R. Alves, M. Braun, L. J. Castro, G. Chourdakis, S. Christ, J. Cohen, S. Druskat, F. Erxleben, J.-N. Grad, et al. “Foundational competencies and responsibilities of a research software engineer: Current state and suggestions for future directions”. In: *F1000Research* 13 (2025), p. 1429.
- [273] J. Hajek, M. Rashid, M. Sevil, A. Cinar, P. A. Alvarez Fernandez, and D. Jain. “The necessity of interdisciplinary software development for building viable research platforms: case study in automated drug delivery in diabetes”. In: *Proceedings of the 21st Annual Conference on Information Technology Education*. 2020, pp. 390–396.
- [274] M. I. Daepf, A. Cabral, T. M. Werner, R. Mansour, C. Catlett, A. Roseway, C. Needham, N. Udeagbala, and S. Counts. “The “Three-Legged Stool”: Designing for Equitable City, Community, and Research Partnerships in Urban Environmental Sensing”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–19.
- [275] S. L. Star and A. Strauss. “Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work”. en. In: *Computer Supported Cooperative Work (CSCW)* 8.1-2 (Mar. 1999), pp. 9–30. ISSN: 0925-9724, 1573-7551. DOI: 10.1023/A:1008651105359. URL: <http://link.springer.com/10.1023/A:1008651105359> (visited on 11/04/2022).
- [276] E. Leahey, C. M. Beckman, and T. L. Stanko. “Prominent but less productive: The impact of interdisciplinarity on scientists’ research”. In: *Administrative Science Quarterly* 62.1 (2017), pp. 105–139.
- [277] M. Bell and N. Lewis Jr. “Universities claim to value community-engaged scholarship: So why do they discourage it?” In: *Public Understanding of Science* 32.3 (2023), pp. 304–321.

- [278] Northwestern Engineering. *Northwestern Partners with Indigenous Scientists to Conserve Great Lakes Wetlands*. McCormick School of Engineering, Northwestern University. Aug. 2022. URL: <https://www.mccormick.northwestern.edu/news/articles/2022/08/northwestern-partners-with-indigenous-scientists-to-serve-great-lakes-wetlands/> (visited on 05/01/2026).
- [279] S. R. Carroll, E. Herczog, M. Hudson, K. Russell, and S. Stall. “Operationalizing the CARE and FAIR Principles for Indigenous data futures”. In: *Scientific data* 8.1 (2021), p. 108.
- [280] L. Matson, G.-H. C. Ng, M. Dockry, M. Nyblade, H. J. King, M. Bellcourt, J. Bloomquist, P. Bunting, E. Chapman, D. Dalbotten, et al. “Transforming research and relationships through collaborative tribal-university partnerships on Manoomin (wild rice)”. In: *Environmental Science & Policy* 115 (2021), pp. 108–115.
- [281] R. C. Nickerson, U. Varshney, and J. Muntermann. “A method for taxonomy development and its application in information systems”. In: *European Journal of Information Systems* 22.3 (2013), pp. 336–359. DOI: 10.1057/ejis.2012.26.
- [282] M. Usman, R. Britto, J. Börstler, and E. Mendes. “Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method”. In: *Information and Software Technology* 85 (2017), pp. 43–59. DOI: 10.1016/j.infsof.2017.01.006.
- [283] A. Hevner and S. Chatterjee. *Design Research in Information Systems: Theory and Practice*. en. Vol. 22. Integrated Series in Information Systems. Boston, MA: Springer US, 2010. ISBN: 978-1-4419-5652-1 978-1-4419-5653-8. DOI: 10.1007/978-1-4419-5653-8. URL: <https://link.springer.com/10.1007/978-1-4419-5653-8> (visited on 01/28/2024).
- [284] S. Gregor and A. R. Hevner. “Positioning and presenting design science research for maximum impact”. In: *MIS Quarterly* 37.2 (2013), pp. 337–355. DOI: 10.25300/MISQ/2013/37.2.01.

APPENDIX A

Literature Review

A.1 Code Book

Code	Definition
Application Type	The targeted application of the paper. We are looking at papers in the space of healthcare (wireless and mobile health) and wireless ecological sensing (in situ).
Environmental	Field Sensors-like targets
Health	wireless and mobile healthcare (wearables and the like)
Other	anything else.
Application Framing	How do the authors frame the work’s application?
Adjacent	When the target application we care about is not the primary one. Essentially an add-on to another use. This can include literature references or not.
Hypothetical Evidence	Used when no references are given, but a hypothetical explanation is used.
Hypothesis/Literature	The authors pull from existing literature to frame. i.e., other works identify the need.
Secondary Observational Need	Motivation is built from secondary stakeholders. For example, if targeting users with a medical condition, doctors are consulted rather than the target users.

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
Primary Observational Need	A formative study with target users to elicit motivation or requirements. This could also include iterative work, such as a previous paper by the same authors.
Co-Design/Participatory	The target population/community is involved in co-design or participatory work to elicit iterative feedback or contribution. Authors may be members of the target community or population, but do not constitute influential stakeholders.
Co-Design/Participatory in Context	Similar to co-design/participatory, but domain experts are included in authorship and make a significant contribution to the work itself.
Evaluation Method	Used to categorize the context of the system's evaluation.
None	No formal evaluation is performed.
Participatory/Workshop	Demonstration of controlled use in a workshop setting. The goal is to iteratively build on the solution rather than directly test its capabilities. Can be similar to a demonstration, but requires some participation, use observation, feedback collection, etc., from users.
Benchtop/Simulation	Testing is conducted as a proof-of-concept in a controlled setting outside of any real-world context. Typically target specific component(s) of the system, introduce a new technology or methodology, and leave the remainder of the implementation as future work.
Controlled/Proxy Setting	Evaluation performed in a controlled setting with convenient participants or at a convenient location. Includes only eval on the technology itself, in which location and/or participation do not evaluate usability

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
Feasibility Demonstration	Evaluation in the form of a demonstration. This often showcases the capabilities of a design or system interface. While the demonstration might include a participant, they are usually not a target, but simply a means of demonstration.
Target Context with Limited Scope	Evaluation environment is the same as the intended deployment environment; however, they have limited duration, scale, or require heavy reserach team observation or assistance. Scale can vary widely; however, the number of devices, deployment time, or degree of independence should be sufficient to demonstrate real-world use.
Target Context (Ecological/Longitudinal)	Full evaluation in the target context for a meaningful timeframe (varies on application). This evaluation still focuses on technology and its usability rather than a full-scale deployment
Full Scale	A system is used in the field with a meaningful impact on the target context. Assumes a prior technology evaluation and proof of successfully filling a need.
Hardware	A description of hardware. Like a sensor or microcontroller.
Microcontroller	
Sensor	
Comm. Module	
Development Kit	
Mechanicals	
Power	
Actuator	
Adjacent/COTS	

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
PCB Layout/Wiring	
Interface	hardware-specific interfaces, like LEDs, buttons, and screens.
Cost	
Software/Firmware	A description of software or firmware, or a link to the source.
Logic	
Frameworks	
Operating System	operating system or bare metal environment
Post Processing/Analysis	
Data Storage/Transmission	
Programming Language	
Interface	
System Design/Architecture	
Cost	
Development Methods	A description of the development process
Simulations/Testing Tools	development-specific tooling, such as power profilers, oscilloscopes, and debugging hardware or software.
Applications/Tools	Application and tool used in the development process (not custom-built). For example, MATLAB, Xcode, cloud services, and Adafacent hardware like gateways.
Hardware/Manufacturing	How something was made. For example, PCB design pipeline, or custom manufacturing techniques or methods.
Sourcing	how materials were purchased or acquired.

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
Custom Methods	A unique or novel method created for the specific workflow.
Custom Tooling	Something built specifically for this work, but has reuse capabilities or some generalizability and is an artifact (software application, hardware jig, etc.)
Existing Theoretical Methodology	
Algorithms	
ML/AI	
Iterative Prototyping	
Challenges	Challenges in any step of the process
Limitations	Limitations identify at any step of the process
Technology Selection	Evidence as to why a specific technology was selected
Artifact Contribution	The purpose of the work.
Hardware	
Software/Tooling	
Data	
Insights	
System	
Algorithm	
Design Consideration	

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
Key Insight	This code is used to tag annotations for later use. Basically, it indicates an unclassified inductive theme or a point that I feel may be a candidate for a call-out in my writing.
Resources	external material (code, designs, data, etc.)
Data Collection Methods	

APPENDIX B

Protocols

B.1 FaceBit Clinical Interview Protocol

B.1.1 Questionnaire/Interview Procedure

We conducted the study in 2 sessions, onsite at a local office. The Questionnaire consisted of 3 parts, with interactions with the interviewers at the beginning to introduce the survey, in the middle to give an overview of the FaceBit platform, and at the end to discuss and conclude. While the participant was working on the questionnaire, the interviewer(s) left the room until the participant had completed a section.

B.1.2 Results of Target Metrics Section

Heart rate. After conducting the study in the clinic to determine the most critical measures, we found it challenging to identify clear winners without identifying the subject of the measure (i.e., the clinician or patient). When asked which metrics were most critical to the patient (scores ≥ 5), responses show the following to be most critical: oxygen saturation, respiration rate, heart-rate, and fever detection. Clinicians on the other hand were also interested in measures of heart-rate and fever detection, however were also interested in knowing their quality of mask fit, oxygen saturation levels and whether the mask was soiled (needed replacement) or not.

B.1.3 Mask Wear

Before the pandemic, 3 out of 12 (25%) participants noted occasionally wearing a face mask, usually while seeing patients suspected of a viral infection or when the clinician was ill. 4 participants reported wearing N95 masks at some point during the COVID-19 pandemic. However, all participants wore surgical or cloth masks during the study and rarely reported requiring an N95. Overall, we found that all clinicians rely on face masks as their primary personal protective equipment during the COVID-19 pandemic. We should note that given the apparently lower attack rate of SARS-CoV-2 amongst children, those requirements for a pediatric setting may differ from that of other medical settings, and the fact that the COVID-19 pandemic caused a national shortage of PPE equipment may also have played a part in the limited use of N95 masks in this setting.

B.1.4 Expanded Metrics of Interest.

- **Mask Information and Safety:** The first and primary goal of a face mask is protection. Therefore, metrics related to the protective ability of a mask are sensible.
- **Personal Health:** A face mask's proximity to the respiratory system provides rich, untapped sensing opportunities for personal health monitoring. We can leverage this to provide real-time monitoring of physiological signals. Measuring these values over time can allow us to infer second-order metrics such as fatigue, and when to suggest a break or hydration.
- **Environmental Factors:** It is also possible to use wearable sensors to monitor environmental health factors. The proximity to the respiratory system makes face masks an ideal site for studying air quality and composition.

B.2 FlexiBLE Interview Protocol

B.2.1 Opening Checklist

*Note: These steps are **required** and are to be followed verbatim at the beginning of the interview.*

- Interviewer Intro
 - I will ask you to introduce yourself in a moment.
- Can I record this Interview? You may ask me at any time to stop or pause the recording. This recording will be transcribed using Zoom via Otter.ai which is managed by Northwestern University’s enterprise account.
- < start recording >
- Introduction to research
- Have you had a chance to review the consent form? Do you have any questions?
- < verbal consent >
- If you would like to stop, or take a break at any time, please let me know. You may revoke your consent to participate at any time, in which case I will stop and delete this recording.

B.2.2 Interview Guide

Note: These questions serve as a guide for the interviewer. While we want to make sure we cover specific themes, the interview may take a natural conversational flow. Our goal is to dive into the participants’ experiences and uncover new themes.

General Questions

- Please introduce yourself
 - What is your profession, or focus in school
 - Educational background
 - Anything else you would like to share
- For the record, what is your relationship with me, the interviewer?
- Briefly explain how you utilize or would like to utilize wireless low-powered embedded sensors.
 - Provide a definition if necessary.
- In what context do you interact with these devices?
 - Do you develop them? (software, firmware, hardware engineering?)
 - Do you integrate them?
 - Do you deploy and do research with them?
 - Etc. ...
- What is your research and role, and how do these devices fulfill a need?
- What type of sensors and/or actuators do you work with?
- How do you interface with the embedded devices you work with?
 - Mobile application, desktop application, cellular, etc.
 - How is this experience?

- What works, and what can be better.

FlexiBLE Platform Use

Note: These questions are only intended for those who have used an internal platform called FlexiBLE.

- Are you familiar with the FlexiBLE platform?
- In your own words, what is the functionality of FlexiBLE.
- How have you used the FlexiBLE platform in your research?
 - What purpose did it serve?
- What gaps, if any, does FlexiBLE fill.
- What worked well about using FlexiBLE?
- What didn't work well with FlexiBLE?
- If you had a magic wand, what would you change about the system?
 - There can be multiple things.

Wireless Embedded System Deployment

- Walk me through a deployment example from your experiences
 - Who are the stakeholders?
 - What is the application?
- What does validation and integration testing look like?
 - Or what would it look like?

- How are bugs/issues tracked and triaged?
- How are devices upgraded?
- What tools/toolkits have you used for deployment?
 - What works well?
 - What challenges remain?

B.2.3 Closing Checklist

*Note: These steps are **required** and are to be followed verbatim at the end of the interview.*

- I appreciate the time you took for this interview
- You have provided valuable information for my research. Would you be available by email or another interview if more questions come up?
- **Referrals:** We are interested in speaking with others in similar roles and with similar experience. Do you have anybody you are willing to pass along to us?
 - If so, we will email them about our research and ask them to volunteer for this interview study.
- Please do not hesitate to reach out to me if you have questions or curiosities in the future.

B.3 FlexiBLE Interview Codebook

Code	Definition
1. Participant Context	
CONTEXT-ROLE	Participant’s professional role, discipline, and how they relate to embedded systems work. <i>Include:</i> Descriptions of job title, academic focus, or disciplinary background. Statements about how they encounter embedded systems in their work. <i>Exclude:</i> General biographical info unrelated to their embedded systems work. Personally identifiable information.
CONTEXT-USE	Types of sensors, actuators, embedded devices the participant has worked with and to what capacity. <i>Include:</i> Mentions of specific devices, sensor types, or roles. <i>Exclude:</i> Hypothetical use.
2. Development Process	
DEV-HW-SELECTION	How participants select hardware components (microcontrollers, sensors, wireless protocols). <i>Include:</i> Discussion of selection tradeoffs, criteria, and constraints. <i>Exclude:</i> Hindsight into selection.
DEV-PLATFORM-TOOLS	Development platforms, operating systems, toolchains, IDEs, and environments used in embedded development. <i>Include:</i> Named tools, descriptions of development environment, preferences or frustration with specific tools. <i>Exclude:</i> Post-processing tools, generalizations, FlexiBLE.
DEV-DATA-COLLECTION	How participants collect, store, and move data from their embedded systems. <i>Include:</i> Descriptions of data pathways. <i>Exclude:</i> Analysis, reserach findings, use of FlexiBLE.

continued on next page

Code	Definition
------	------------

(continued from previous page)

DEV-CHALLENGES	Difficulties, frustrations, and barriers encountered during embedded system development. <i>Include:</i> Explicit statements of difficulty, pain points, workarounds, or processes that took longer than expected in development. <i>Exclude:</i> Challenges specific to FlexiBLE, deployment challenges.
----------------	--

3. FlexiBLE Experience

FLEX-PURPOSE	Participant's understanding of what FlexiBLE is and what it is meant to do. <i>Include:</i> Descriptions of FlexiBLE's functionality, purpose, or role in their own words. <i>Exclude:</i> Descriptions of other toolkits, or descriptions for text.
FLEX-USE-CASE	How the participant actually used FlexiBLE in their research or project. <i>Include:</i> Specific projects, studies, or takeaways where flexiBLE was applied, and what it was used for concretely. <i>Exclude:</i> Hypothetical or aspirational uses.
FLEX-GAPS-FILLED	Needs or gaps that FlexiBLE address which were previously unmet. <i>Include:</i> Comparisons to prior workflows, statements about what was possible with FlexiBLE that wasn't before, time/effort saved. <i>Exclude:</i> General praise with specificity.
FLEX-LIMITATIONS	Aspects of FlexiBLE that did not work well or caused friction. <i>Include:</i> Negative assessment of specific features, bugs, UX issues, missing capabilities or frustrations. <i>Exclude:</i> Limitations of embedded systems in general.

continued on next page

Code	Definition
<i>(continued from previous page)</i>	
FLEX-STRENGTHS	Aspects of FlexiBLE that worked well for the participant. <i>Include:</i> Positive assessments of specific features, design decisions, or experiences, things they would keep. <i>Exclude:</i> General positive sentiment without reference to specific functionality.
FLEX-IMPROVEMENTS	Desired changes, new features, or modifications the participant would make to FlexiBLE. <i>Include:</i> Forward-looking suggestions, magic wand responses, feature requests, design changes, etc. <i>Exclude:</i> Complaints about current functionality without a suggested direction.
4. Deployment and Integration	
DEPLOY-CONTEXT	The deployment setting, stakeholders, and application domain. <i>Include:</i> Descriptions of where devices were deployed, who is involved, and the application area. <i>Exclude:</i> Prototyping/benchtop details.
DEPLOY-MAINTENANCE	How bugs are tracked, devices are upgraded and systems are maintained post-deployment. <i>Include:</i> Bug tracking processes, OTA updates, firmware versioning, device swaps, and field maintenance visits. <i>Exclude:</i> Initial development phase debugging.
DEPLOY-CHALLENGES	Difficulties specific to deploying and maintaining embedded systems in the field. <i>Include:</i> Statements about real-world deployment friction: connectivity issues, user error, environmental factors, logistics. <i>Exclude:</i> Development-phase challenges.

continued on next page

Code	Definition
------	------------

(continued from previous page)

5. Collaboration & Knowledge Transfer

COLLAB- ONBOARDING	How participants onboard others to their systems, tools, and workflows. <i>Include:</i> Training processes, documentation practices, mentoring, knowledge transfer to new team members or collaborators. <i>Exclude:</i> Self-directed learning.
COLLAB- INTERDISCIPLINARY	Interactions, tensions, or dynamics arising from working across disciplines on embedded systems. <i>Include:</i> Communication challenges with non-engineers, differing expectations, bridging technical and domain expertise. <i>Exclude:</i> Within-dicipline collaboration.

6. Emergent

EMERGENT-{}	Open placeholder for inductively derived codes generated during analysis (no a priori definition).
-------------	--

B.4 Microclimate Sensing Field Study Protocol

B.4.1 Start

- Can I record? <start recording>
- <Intro to this Research>
- <Before we get started>
 - Have you received and had a chance to review the consent form?
 - Do you have any questions regarding the consent form?

- Do you wish to participate in this research?
- <Once the participant agrees>: Also, I want to mention that if you want to stop at any point, take a break, or if I ask you a question that you do not wish to answer, please feel free just to let me know, and that is completely fine. Ask participants if they have any questions before beginning.
- <If it is someone we've talked to before>: I know we've talked before, but I may ask some questions that we've already discussed, and I might phrase some of my questions in an intentionally vague way.

B.4.2 General Framing (required)

- Tell me about your organization and it's high-level goals/mission
- Tell me about your role
- Paint a picture of what data collection looks like at your organization
 - What data is important to your organization?
 - How does your organization support community needs?
- What community and personal partnerships, if any, do you rely on for your work?
- What technology do you use in your role?
- What works well with the technology you are currently using?
 - Is there something you particularly like about <technology x>?
- What are some challenges you face in your role collecting and managing environmental data

- guiding: what limits your sensing capabilities?
- If you had a magic wand, what data would you want to see [as it relates to the organization or environment] e.g., unlimited resources and technology

B.4.3 Field Research Specific (understanding the user flow of conducting field research)

- What is your role in conducting field research? (If actually going into the field, continue on this section. Else skip to next section)
- Describe a typical field research example you've been involved in. How long did it take, where were you, what tools did you use, etc.
- What technology do you use in the field (that you bring back with you or that you leave behind)? How do you set up and interact with this technology?
- How do you prepare for field research?

B.4.4 Collecting Sensor Data Specifically

- How do you usually collect sensor data from remote locations for further analysis?
- What devices do you usually carry with you? E.g., camera, fitbit, headphones, cellphones, GoPro, etc? How much does it weigh? How do you carry it?
- How often do you go into the field to collect data?
- How long does it take for one data collection trip? How long would you be at the spot to collect sensor data?
- How much data does each sensor generate?

B.4.5 Understanding the problems in current field research technology

- What are your biggest frustrations conducting field research?
- What is the user interface like for the technology you use in the field?
- Does your field technology ever not work as intended, either in the moment or when you check the data later? How often does your field technology fail to collect the data you were trying to collect?
- If your field technology ever fails to work as intended, why do you think it failed? (User error, poor design, batteries died, rough environment, etc.)
- What do you think is the most difficult or frustrating part during sensor data collection, and why? How do you address those challenges?

B.4.6 Understand user needs during sensor data collection, specifically

- How do you pull the data from the sensor?
- How and when do you check the data collected? E.g., ensuring the data validity / quality.
- Are you satisfied with the amount of time it takes from when technology captures data in the field to when you have access to it?
- Imagine a system that helps you collect sensor data; how would you like to interact with such a system? How often? With what kind of interface?

B.4.7 Understand how they analyze data

- How do you use the data you collect from the field? What systems do you enter it into and what depends on this data?

- Where are the data stored? (e.g., cloud service or on premises)
- What tools or systems are used for data analysis?
- What decisions does this data drive?

B.4.8 Understand how the environment could impact the field researcher in conducting their work

- How often do you have internet or cell phone connection while in the field? What is the quality of this connection?
- If you leave sensors behind in the field, how do you decide where to set it up?

B.4.9 Understand revisits

- Have you ever physically revisited leave-behind sensors to collect stored data? Could you walk us through the entire workflow?
- Did you use any tools in this process? How did you use them?*
- How often do you revisit the sensors/technology you deployed in the field? What is motivating the frequency of these visits?

B.4.10 Closing

- Demographic questions: I'm going to ask you a few demographic questions, but feel free to not answer any of them.
 - What is your age?
 - What is your race?
 - What is your gender?

- I appreciate the time you took for this interview.
- You have provided a lot of valuable feedback. Would you be available for a follow-up interview if more questions come up?
- <Referrals>: We are interested in speaking with others in similar roles and experiences in sensing for environmental conservation, do you have anybody you are willing to pass on?

B.4.11 Follow up

- For the Visa Gift Card, we need to collect a US address to register it. Can you please provide that for us (Email, street address, city, state, zip code)
- Send them a gift card
- Update interview tracking spreadsheet
- Follow-up email asking for additional connections and add those to the interview tracking spreadsheet

```

packet HEALTH {
    timestamp : uint32 // UNIX epoch
    period    : uint16 // sampling period in microseconds
    count     : uint16 // samples per channel (N)
    green     : uint32[N] // PPG channel, raw ADC counts
    orange    : uint32[N]
    red       : uint32[N]
    infrared  : uint32[N]
}

```

Listing C.1: Stored fields and data types for a hypothetical health sensing data collection packet

APPENDIX C

Additional Analysis

C.1 Wireless Protocols

Many methods exist for wirelessly transporting data, each with different layers of abstraction. In this section, we present a packet-size comparison across a set of application-level data transport protocols. This is a supplemental demonstration for section 7.2, which provides a brief overview of the requirements and size differences among existing solutions. To demonstrate this, we build two packets: a health packet, roughly modeled after PhysioHub (section 4.5), and an environmental packet, roughly modeled after Makak (chapter 6). Listing C.1 and listing C.2 show pseudo-representations of the data structure.

Table C.1 shows the same mocked packet for each example using different protocols. We see a wide variation in sizing. The larger fully descriptive protocols, such as SensML and JSON in Web IoT format, are large but completely independent. Hybrid approaches, including LwM2M-CBOR, SparkPlug, and Homie, require an external one-time-cost context to understand

```

packet ENV {
    timestamp      : uint32    // UNIX epoch
    temperature    : float32   // degrees Celsius
    atm_pressure   : uint16    // mbar
    water_pressure : uint16    // mbar
    battery        : uint8     // percent (0-100)
    state          : enum {IDLE, SAMPLING, UPLOADING, ERROR}
    errors         : string[]  // variable-length list, may be empty
}

```

Listing C.2: Stored fields and data types for a hypothetical environmental sensing data collection packet

the packet and greatly reduce the overall size. The floor packet size, before compression, is either bit-packed or a raw struct representation, but requires very careful parsing on both sides of the transmission. FlexiBLE (chapter 3) used a custom bit-packing schema, a device specification registry, and a contextual BLE GATT service to obtain near-floor individual packet sizes. We do not recommend a particular solution, but provide this comparison as a reference.

Together, the columns of table C.1 trace the tradeoffs between payload size, self-description, and transport-independence. The self-describing text formats (JSON, SenML) carry readable metadata in each packet sub-component create significant overhead. On the other end of the spectrum raw structs and bit-packed messages strip field identity entirely, requiring each endpoint to know the specifics of parsing. The application protocols that stay both small and self-describing (LwM2M, Sparkplug, Homie), by binding to a single transport. No row at once is near the floor, independent of the wire, and self-describing. The three properties are individual well served and jointly unmet. Flexible reached near-floor sizes only by pairing custom bit-packing with a device-specification registry and a BLE GATT service to supply the missing context (section 4.4.1). The mechanism in section 7.2 takes this gap as its starting point: rather than endorse a protocol, it specifies what such a mechanism would need, a wire-agnostic envelope, compact typed payloads, and out-of-band metadata that restores self-description without paying for it on every packet.

Table C.1: Per-message wire size of the same two payloads under different encodings, including the data-capable IoT application-layer protocols (LwM2M-CBOR, Sparkplug NDATA, Homie). The Health payload is a 444-sample multi-wavelength PPG buffer (four channels of integer ADC counts, anchor timestamp + 2250 μ s period); the Env payload is a single environmental reading with 0 error string(s). $\times$ floor is each encoding’s size relative to the bit-packed range-tight floor (bottom row), the theoretical minimum.

Encoding	Class	Wire-agn.	Self-desc.	Health (PPG)		Env	
				Bytes	$\times$ floor	Bytes	$\times$ floor
JSON	Text	Yes	Inline	12,482	2.81	83	6.92
SenML/JSON	Text	Yes	Part.	68,354	15.37	208	17.33
MessagePack	Binary	Yes	Struct.	8,914	2.00	57	4.75
CBOR	Binary	Yes	Struct.	8,914	2.00	58	4.83
SenML/CBOR	Binary	Yes	Part.	35,980	8.09	92	7.67
Protobuf	Binary	Yes	Schema	5,349	1.20	21	1.75
Raw struct (fixed)	Binary	Yes	Schema	7,112	1.60	15	1.25
LwM2M-CBOR	Binary	Partial	Registry	13,127	2.95	68	5.67
Sparkplug NDATA	Binary	No	Birth	7,163	1.61	51	4.25
Homie (data)	Text	No	Topic	12,606	2.83	179	14.92
Bit-packed (range-tight)	Binary	Yes	Schema	4,447	1.00	12	1.00

Wire-agn. (wire-agnostic) — whether the encoding is independent of the transport. **Yes:** the bytes ride over any link (HTTP, CoAP, MQTT, BLE, raw radio). **Partial:** defined for several transports but coupled to one ecosystem (LwM2M-CBOR rides the CoAP-centric LwM2M stack). **No:** bound to a specific transport (Sparkplug and Homie are MQTT-only). **Self-desc.** — how a receiver recovers each field’s identity and meaning. **Inline:** field names and structure travel in the payload, human-readable. **Struct.:** binary but type-tagged, so a generic decoder recovers the structure without a schema; names still inline. **Part.:** values and units are inline, but deeper meaning resolves through an external registry. **Registry:** fields are integer object IDs defined in an external registry. **Birth:** compact alias-keyed updates are resolved by a one-time self-describing birth message sent earlier in the session. **Topic:** semantics live in the MQTT topic hierarchy plus a retained `$description` (Homie). **Schema:** nothing on the wire identifies fields — both ends share an out-of-band schema or fixed layout (Protobuf, raw struct, bit-packed).

Compression is independent level for payload size, but is usually managed below the application-layer. table C.2 shows the effect of a general DEFLATE compression: it trims the verbose text formats but inflates already-small packets, and complements a compact encoding rather than replacing one.

Table C.2: Compression overlay (raw DEFLATE, zlib level 9). Generic compression recovers much of the text formats’ overhead but does not reach a compact typed encoding, and adds overhead on already-small payloads.

Encoding	Health (PPG)		Env	
	Raw	+zlib	Raw	+zlib
JSON	12,482	4,794	83	83
SenML/JSON	68,354	10,947	208	122
Homie (data)	12,606	4,880	179	120
CBOR	8,914	4,489	58	64
Raw struct (fixed)	7,112	3,845	15	23

C.2 Embedded Time Management Example

In this demonstration, we create two firmware timing examples, a *naïve* and *correct* approach. The naïve example is modeled from two code releases we found in the literature review (chapter 2). More work is required to know if these timing issues affect the overall result, and that is not the point of the demonstration, therefore we do not name those works and the code here is rewritten with similar patterns. We present the two examples in pseudocode in listings C.3 and C.4 and in public projects on Wokwi projects: <https://wokwi.com/projects/467137275079824385> (naïve), <https://wokwi.com/projects/467177695703095297> (corrected). Wokwi is a online ESP32 simulator. Our example are written for Arduino on the ESP32 microcontoller, but the logic is not specific to any platform. We detail each below:

Naïve : This is a simple firmware with superloop control logic. It *claims* to sample it’s sensors at 100Hz, trusting a trailing `delay(100)` to set the cadence. In reality the loop period is defined by the delay *plus* the time it takes to do the work within the loop. With this logic the small amount of time it takes to perform the logic within the

```

loop:
delay(10 ms)                # BUG 1: real period = delay + work
read 3 sensors              #          (work time is unaccounted)
sample.time = index * 10 ms # BUG 1: assumed fixed rate, not
    measured
buffer += sample
on buffer full (20 samples):
stamp = millis()            # BUG 2: ONE stamp for the whole batch
emit every row with stamp   # BUG 3: relative ms from boot, no epoch

```

Listing C.3: Naive timekeeping (pseudocode): an assumed fixed rate, one transmit-time stamp per batch, and relative time with no epoch anchor. Runnable sketch: `scripts/embedded_time_demo/naive/naive.ino` (Wokwi).

```

loop:
next += 10 ms                # FIX 3: monotonic deadline
wait until micros() >= next  #          (jitter can't pile up)
if first sync:               # FIX 2: anchor uptime to a real epoch
anchor = (micros(), epoch)
t = micros()                  # FIX 1: stamp at acquisition, per
    sample
read 3 sensors
assigned = epoch + (t-anchor) #          absolute time once synced
emit (t, assigned, synced)

```

Listing C.4: Corrected timekeeping (pseudocode), per the framework's Time component: a per-sample acquisition stamp, an authoritative epoch anchor, and a monotonic cadence.

loops steadily slides the period behind real time. It timestamps each sample from its index, buffers 20 samples, and at flush time takes a single system uptime reading that it attaches to whole batch. This logic provides only one timestamps per batch even though they were acquired over 200 ms. Additionally the timestamps are relative to the systems clock, time since boot, and not anchored to wall-time and therefore cannot be aligned across boot cycles. Table C.3 shows the drift that occurs in this cycle as run from the Wokwi project. In examples we found, timestamps are not recorded and sensor period is assumed based on the logic, hiding this error entirely.

Corrected : This examples presents an identical workload with timekeeping management.

The loops wakes on a running monotonic deadline to avoid jitter accumulation into the

sample period. Every sample is timestamped at acquisition time rather than once per batch. On the first connection it accepts an authoritative epoch from the host, recording a reference to system uptime so the system can report absolute time. Before that sync occurs the system reports a relative time. The payoff is per-sample timestamps that remain exact even when the cadence hiccups, a sample that lands late after a stall is stamped correctly. Table C.3 shows a corrected run from Wokwi, where time stays accurate.

Table C.3: Serial-monitor excerpts from the two sketches (same workload), showing timestamp drift versus none. **Naive:** the fixed-rate column (`fixedrate_ms`) falls progressively behind real acquisition time (Δ_{fixed} grows to -2.3s); within one 20-sample batch (idx 2000–2019) a single `bundle_stamp_ms` is reused for every row; and all values are relative milliseconds from boot, never anchored. **Corrected:** each sample is stamped at acquisition, the host SYNC flips `synced` 0→1 at idx 50, `assigned_us` becomes absolute epoch microseconds, and the error stays 0.

Naive idx,t_true_us,fixedrate_ms,bundle_stamp_ms				
idx	t_{true} (ms)	fixedrate_ms	bundle_stamp_ms	Δ_{fixed} (ms)
0	0.0	0	196	0.0
1,000	10,400.0	10,000	10,596	−400.0
3,000	31,200.0	30,000	31,396	−1,200.0
5,999	62,336.8	59,990	62,390	−2,346.8
<i>one batch (idx 2000–2019): 4 of 20 rows — bundle_stamp_ms identical, acquired up to ~200 ms apart</i>				
2,000	20,800.0	20,000	20,996	−800.0
2,001	20,810.2	20,010	20,996	−800.2
2,002	20,820.3	20,020	20,996	−800.3
2,019	20,992.8	20,190	20,996	−802.8
Corrected idx,t_true_us,assigned_us,synced				
idx	t_{acq} (ms)	assigned_us	synced	Δ (ms)
0	10.0	10,000	0	—
49	500.0	500,000	0	—
50	510.0	1,718,000,000,000,000	1	0.000
51	520.0	1,718,000,000,010,000	1	0.000
3,000	30,053.2	1,718,000,029,543,150	1	0.000
5,999	60,000.0	1,718,000,059,490,000	1	0.000

$t_{\text{true}}/t_{\text{acq}}$ is an oracle acquisition time a real device would not log; it is shown only to make the error visible. $\Delta_{\text{fixed}} = \text{fixedrate_ms} - t_{\text{true}}$ is the naive fixed-rate error. The corrected Δ is `assigned_us` minus the epoch-anchored oracle; “—” marks pre-sync rows, which make no calendar claim (`synced`=0).

Vita

Education

- 2026 **Ph.D. in Computer Science**, Northwestern University
Advisors: Dr. Josiah Hester and Dr. Nabil Alshurafa
- 2020 **M.S. in Artificial Intelligence**, Northwestern University
- 2013 **B.S. in Informatics**, Indiana University Bloomington

Professional Experience

- 2026 – Present **Purdue University** – Assistant Professor of Practice in Computer Science
 A teaching-track faculty member based on the Purdue Indianapolis campus focused on the core undergraduate computer science program.
- 2023 – Present **Winnemac Labs** – Co-Founder and Software Engineer
 A consultancy focused on developing end-to-end embedded systems to aid research. Focus on software for embedded systems, including interfaces, mobile applications, data pipelines, and cloud services.
- 2020 **NASA** – Data Science Intern
 Designed a comprehensive relational model to manage a portfolio of funded early-stage technology projects, enhancing process efficiency. Innovated in virtual reality technology by creating applications to visualize complex data and integrating machine learning methods, facilitating over 40 hours of agency-wide interviews on artificial intelligence and data science.

- 2018 – 2020 **Nordstrom Trunk Club** – Senior Software Engineer
- Revamped the customer-facing mobile app, reducing technical debt and prioritizing a mobile-first approach to optimize the most profitable sales application. Created a modular data management system and integrated key business features, including a self-checkout process, stylist messaging, and accessibility. Implemented the ‘Style Swipes’ feature, leading to over 65 million interactions in three months, improving customer engagement, and driving valuable sales data.
- 2013 – 2018 **Wipro Appirio** – Senior Consultant
- Led international projects on the Digital Studios team, specializing in data-driven integration of enterprise mobile and web applications with Salesforce and Heroku. Served as technical architect on a \$3+ million iPad data analytics sales application project, overseeing all stages from requirements gathering to deployment. Successfully executed the second-largest Salesforce data migration to date for a large human resources firm, maintaining a complex deployment pipeline with multiple staging environments and development streams.
- 2012 **GE Healthcare** – Information Technology Leadership Program Intern
- Evaluated the security of FileVault 2 disk encryption for Mac OS X on company-administered computers. Implemented an automated installation script to ensure HIPAA compliance on Apple devices.

Teaching and Course Development

2024 – 2025

Wireless and Mobile Health

In collaboration with the Computer Science and Preventive Medicine department at Northwestern University, serving as a teaching assistant in Winter 2024 and 2025, and an instructor in Summer 2024. Provide continued support through planning and guest lectures. Focused on reworking material in signal processing, multi-modal sensing, embedded firmware development, application development, and edge machine learning. **Course evaluation:** 6/6 rating (n=10), with comments highlighting clarity, responsiveness, and enthusiasm.

2021 – 2025

We All Code

Developed a four-hour robotics course for We All Code, a nonprofit organization that provides free classes to students in Chicago aged 8–17. Developed custom firmware that allows students to control the robot via Bluetooth Low Energy using a custom Python library. The course has had 183 attendees over 20 sessions and is scheduled monthly. Additional instruction sessions include HTML, CSS, JavaScript, and Python development.

2011 – 2013

Introduction to Informatics

Lead Undergraduate Instructor, instructing a lab session twice a week and assisting with another. Responsible for giving technical lab lectures, grading assignments, and facilitating class projects. Participated in four semesters of courses. Selected as lead instructor in my second year to facilitate grading and other responsibilities among the other undergraduate instructors.

Publications

- 2026 **Blaine Rothrock***, Alexander Curtiss*, Thomas Cohen*, Josiah Hester. **PhysioHub: A Wirelessly Synchronized Multi-Node Wearable System.** *In Proceedings of the 2026 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE).* PREPRINT
- 2026 Chris Wright, Abhijith Surendran, Inkyu Lee, Juan E. Villacres, Alex Ezerins, Alexander Curtiss, Nathan Brown, Junyi Liu, **Blaine Rothrock**, Peter Rios, Daisy Lopez, Ira Joshi, Hafsa Nasir, Han Wang, Cody Fell, Anthony Davis, Jose Oberholzer, Josiah Hester, Tzahi Cohen-Karni, Jonathan Rivnay, Omid Veisesh. **Design of a Wireless, Fully Implantable Platform for In-Situ Oxygenation of Encapsulated Cell Therapies.** *Device*, 4(5), 101106.
<https://doi.org/10.1016/j.device.2026.101106>
- 2026 Eric Greenlee, **Blaine Rothrock**, Cody Atchison, Chris McNerney, Ellen Zegura, Josiah Hester, Alex Cabral. **Buoy Meets World: Aquatic Sensing Insights from 169 Days of Low-cost Wetland Buoy Deployments.** *In Proceedings of the 2026 International Conference on ICT for Sustainability (ICT4S).* PREPRINT
- 2025 **Blaine Rothrock**, Eric Greenlee, Yaman Sangar, Julia McKenna, William Graveen, Kristen Hanson, Melissa Lewis, Kathleen Smith, Miles Falck, Brandon Bryne, Darren Vogt, Kimberly R. Marion Suiseeya, Ellen Zegura, Josiah Hester, Alex Cabral. **Makak: Co-designing Environmental Sensors to Protect Manoomin (Wild Rice).** *In Proceedings of the ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies* (pp. 115–130).
<https://doi.org/10.1145/3715335.3735460>

- 2025 **Blaine Rothrock**, Eric Greenlee, Yaman Sangar, Julia McKenna, William Graveen, Kristen Hanson, Melissa Lewis, Kathleen Smith, Miles Falck, Brandon Byrne, Darren Vogt, Kimberly R. Marion Suiseeya, Ellen Zegura, Josiah Hester, Alex Cabral. 2025. **Makak: Co-designing Environmental Sensors to Protect Manoomin (Wild Rice)**. *COMPASS '25: Proceedings of the ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. (June 2025), pages 115–130.
<https://doi.org/10.1145/3715335.3735460>
- 2024 **Blaine Rothrock***, Eric Greenlee*, Hyeonwook Kim, Ellen Zegura, and Josiah Hester. 2024. **“The Devil You Know”: Barriers and Opportunities for Co-Designing Microclimate Sensors**. *ACM J. Comput. Sustain. Soc.* 2, 3, Article 39 (September 2024), 30 pages.
<https://doi.org/10.1145/3685695>
- 2024 Glenn J. Fernandes, Jiayi Zheng, Mahdi Pedram, Christopher Romano, Farzad Shahabi, **Blaine Rothrock**, Thomas Cohen, Helen Zhu, Tanmeet S. Butani, Josiah Hester, Aggelos K. Katsaggelos, and Nabil Alshurafa. 2024. **HabitSense: A Privacy-Aware, AI-Enhanced Multimodal Wearable Platform for mHealth Applications**. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 101 (September 2024), 48 pages. <https://doi.org/10.1145/3678591>
- 2023 **Blaine Rothrock**, Eric Greenlee, and Josiah Hester. 2023. **Community-Driven Mobile and Ubiquitous Computing**. *GetMobile: Mobile Comp. and Comm.* 27, 3 (September 2023), 5–9.
<https://doi.org/10.1145/3631588.3631590>

- 2023 Inkyu Lee, Abhijith Surendran, Samantha Fleury, Ian Gimino, Alexander Curtiss, Cody Fell, Daniel J. Shiwarski, Omar Refy, **Blaine Rothrock**, Seonghan Jo, Tim Schwartzkopff, Abijeet Singh Mehta, Yingqiao Wang, Adam Sipe, Sharon John, Xudong Ji, Georgios Nikiforidis, Adam W. Feinberg, Josiah Hester, Douglas J. Weber, Omid Veisesh, Jonathan Rivnay, and Tahi Cohen-Karni. 2023. **Electrocatalytic on-site oxygenation for transplanted cell-based therapies**. *Nature Communications* 14 7019 (2023). <https://doi.org/10.1038/s41467-023-42697-2>
- 2022 Mara Ulloa, **Blaine Rothrock**, Faraz S. Ahmad, and Maia Jacobs. **Invisible clinical labor driving the successful integration of AI in healthcare**. *Frontiers in Computer Science* 4 (2022): 1045704. <https://doi.org/10.3389/fcomp.2022.1045704>
- 2021 **Blaine Rothrock**, Alexander Curtiss, Juyang Bai, and Josiah Hester. **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 (UbiComp/ISWC '22 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 379–385. <https://doi.org/10.1145/3544793.3560399>
- 2021 **Blaine Rothrock***, Alexander Curtiss*, Abu Bakar, Nivedita Arora, Jason Huang, Zachary Englhardt, Aaron-Patrick Empedrado, Chixiang Wang, Saad Ahmed, Yang Zhang, Nabil Alshurafa, and Josiah Hester. 2022. **FaceBit: Smart Face Masks Platform**. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 151 (Dec 2021), 44 pages. <https://doi.org/10.1145/3494991>

*denotes equal contribution

Invited Talks, Conference Presentations, & Posters

- 2025 **Northwestern University Center for Native American and Indigenous Research Annual Research Symposium**
Poster presentation, *Makak: Co-designing Environmental Sensors to Protect Manoomin*.
- 2025 **WWA Wetland Science Conference**
Accepted presentation titled *Makak Aabajitton Ganawendang Manoomin: Guiding wild rice conservation with Anishinaabe worldview* in collaboration with the Great Lakes Indian Fish and Wildlife Commission.
- 2024 **IEEE Real Time Communications Conference & Expo**
Invited presentation titled *Makak: Community-Driven Microclimate Sensor Development for Wild Rice Conservation* in the Internet-of-Things track.
- 2024 **Northwest HCI Meeting**
Participation in the works-in-progress section facilitating discussion on the topic *Enabling Dynamic Interfaces for Better Interoperability of Embedded Systems*.
- 2024 **ACM COMPASS**
Presentation for JCSS publication “*The Devil You Know*”: *Barriers and Opportunities for Co-Designing Microclimate Sensors*.
- 2022 **ACM UbiComp/ISWC**
Presentation for IMWUT publication *FaceBit: Smart Face Masks Platform*.

- 2022 **ACM UbiComp/ISWC Adjunct**
 Presentation and participation at a hybrid workshop for Human Activity Sensing Corpus and Application.
- 2016 **Dreamforce**
 Technical presentation titled *How I Learned to Stop Worrying and Love Heroku Connect* on how to enable Salesforce.com object structures to drive mobile application functionality.

Patents

- 2025 **Method and System for a Smart Mask**
 A smart face mask device includes a housing that is designed to mount to a personal protective face mask and a memory within the housing that is configured to store sensor data. The device includes one or more sensors configured to generate the sensor data during use of the personal protective mask. The device also includes a processor operatively coupled to the memory and configured to process the sensor data to determine information regarding a user of the personal protective mask, and to transmit the information regarding the user to a remote application.

Honors & Awards

- 2025 **Peter and Adrienne Barris Outstanding Teaching Assistant**
 Northwestern University
- 2024 **Recognition of Dedicated Service**
 IEEE EMBS BSN 2024

2024	International Travel Grant The Graduate School, Northwestern University – For travel to ACM COMPASS in India
2009 – 2013	Indiana Commission for Higher Education 21st Century Scholar – Competitive merit and need-based scholarship covering full tuition for undergraduate studies

Volunteering and Service

2025 – Present	Northwestern University Computer Science Department Computer Science Research Track Program Mentor
2024 – Present	IEEE Body Sensor Networks Technical Program Committee Member Manuscript Reviewer
2022 – Present	ACM Journal on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) Manuscript Reviewer
2022 – 2025	We All Code Mentor and Instructor
2024	ACM Conference on Human Factors in Computing Systems (CHI) Manuscript Reviewer
2021	Illinois Science & Technology Collation Mentor Matching Engine Mentor
2020	ACM Empirical Methods in Natural Language Processing Manuscript Reviewer