The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
Paper ID #42194

Wireless Environmental Sensing Electronics Framework Development with Successive Capstone Projects

Prof. David Burnett, Portland State University

David C. Burnett received the B.S. and M.S. degrees in Electrical Engineering from the University of Washington focusing on circuits, embedded systems, and oceanographic instrumentation. He received the Ph.D. in Electrical Engineering and Computer Sciences at the University of California Berkeley as an NDSEG Fellow focusing on crystal-free fully-integrated wireless sensor nodes. He is currently an assistant professor of electrical and computer engineering at Portland State University. Prior to his Ph.D., he was Member of Technical Staff at Sandia National Laboratories, Livermore, Visiting Lecturer at Da Nang University of Technology, Vietnam, and Electrical Engineer for an experimental ROV at McMurdo Station, Antarctica. He has served on various ACM SIGGRAPH conference committees serving as emerging technology juror and responsible for special technical projects and data networks. His research interests include crystal-free RF communication, low-power circuit design, and field-deployable sensor systems. He is a Senior Member of the IEEE.

Wireless Environmental Sensing Electronics Framework Development with Successive Capstone Projects

1. Introduction

The undergraduate program at the Portland State University Department of Electrical and Computer Engineering (PSU ECE) includes a three-quarter capstone design sequence typically taken during the student's senior year. For the last three years, and a fourth currently ongoing at time of writing, a capstone project has been sponsored by the author's Wireless Environmental Sensing Technology (WEST) Lab. The purpose of these sponsored capstones are as follows:

1. In the short term, design a wireless sensor to solve a specific problem.
2. In the long term, converge at a more general-purpose hardware and software solution set with which to design future wireless sensor nodes.
3. Year on year, build on the previous year's results and thereby provide students with a unique opportunity to see examples of good and bad documentation written by their own peers, and learn how to improve their own written technical communication in the future.
4. Involve more undergraduate students in engineering research and research-adjacent project work in the spirit of the CURE (Course-based Undergraduate Research Experience) movement, inspiring more students to continue their education through graduate school.

This third purpose was inspired by past project-based courses that also benefit from long-term improvement of an underlying framework. In that past course, the framework was integrated circuit design and students continue to learn from their predecessors how to communicate technical ideas well [1]. Here, the hardware design is at the PCB level rather at the integrated circuit level. The fourth purpose, with student-centric focus, was inspired by the department's goal to increase the number of graduate students undertaking PhD-level research. Students must first be aware of engineering research as a concept before they can consider research as a career path, hence one of the aims of this series of capstone projects.

The WEST Lab's focus includes integrated circuit design, wireless embedded systems, and sensor development, and these capstone projects align with those research topics. The lab's core wireless sensor system competencies have myriad applications, particularly in environmental sensing, and the PI is often approached by researchers and companies interested in a wireless sensor network for agriculture, harmful gas monitoring, ecological observation, continuous water quality measurement, and more. These applications require mature systems with high technology readiness level (TRL), so the most cutting-edge work is inappropriate. But there are several commercially-available systems the lab leverages, such as Analog Devices SmartMesh IP [2] these commercial devices are also fairly well-supported and can be woven together by experienced undergraduates.

The problems each year's capstone project aimed to solve are as follows:

- Year 1 (2020-2021): vertical farming startup interested in an easily-deployed growing conditions monitor that can be used at high density indoors.

- Year 2 (2021-2022): A harmful gas monitoring project in collaboration with neighboring university, research institute, and national lab to observe gas concentrations in a contaminated environment [3].
- Year 3 (2022-2023): Indoor air quality monitoring to assess ventilation to predict risk of airborne disease transmission and determine immediate impact of wildfire events on indoor air quality [4].
- Year 4 (2023-2024): Ongoing at time of writing; moisture content monitoring of biomass to ensure long-term carbon sequestration.

In these projects, the PI's past research, and the design project in his "Instrumentation and Sensing" course, a design pattern emerges and outlines a framework that can enable rapid design of wireless sensors. The framework includes the core components described below.

1.1 Sensing

The wireless devices need to take some property of the physical world and transduce it into a signal in the electrical domain. The property (analyte), the transducer, and the transduction mechanism vary, but the result typically ends up as a voltage connected to an analog-to-digital converter. Commercial sensors may include digital signal conditioners and digital interfaces. The set of sensor interfaces is generally well-understood and typically falls into SPI, I2C, and UART digital communication, or an analog voltage to be matched with an ADC's full-scale range and resolution to achieve sensitivity and dynamic range goals. Sometimes, the output signal is based on time: duty cycle, pulse widths, or pulse rates. The method to get the sensor's instantaneous value into the digital domain can vary, so any wireless sensor framework must include flexibility in this area. Fortunately, thousands of diverse commercial microcontrollers are available with support for nearly every possible sensor interface type. Microcontroller selection is typically a matter of familiarity and ability to rapidly prototype, rather than searching based on requirements specification.

1.2 Computation

In most wireless sensing systems, little actual signal processing is needed. Some systems may be severely data-constrained and record much more raw data than can be communicated through low-power radios; these systems, such as acoustic recorders, may filter, downsample, compute frequency spectra, etc., to reduce the volume of data to be transmitted down to its most salient. The growing popularity of machine learning suitable for edge computing, such as TinyML [5], is also responsible for some modest computation resident on the wireless sensor node.

Applications such as environmental sensing need only sample a single sensor at periods on the order of hours and report that information to a base station. For wireless sensor systems recording environmental analytes, computation here is limited to housekeeping: scheduling and control of sensor(s), temporary local storage, and management of the wireless communication system. Modern microcontrollers are likely to be able to perform all such tasks on a single CPU but, for implementation simplicity, the housekeeping computation tasks may be running on a different CPU than the wireless communication tasks to allow development by different teams in parallel or to enable easy bolt-on of wireless functionality. Control of sensors and storage is such a routine operation that it can even be delegated to a purpose-built digital core instead of firmware running on a general-purpose CPU [6]. This means wireless communication is often the most compute-intensive task of the sensor node, requiring precise timing to schedule queued messages to be sent during time-synchronized data exchanges such that the wireless radios can be powered off when not needed [7, 8]. The most critical feature of a wireless sensor node's computation subsystem may therefore be accurate timing, which is a feature often overlooked in wireless systems due to the assumption of a crystal oscillator resident inside each wireless node. As wireless nodes continue to miniaturize, this may not be a good assumption and low-level assumptions about network design must be reconsidered [9-11].

1.3 Communication

As mentioned in the prior section, there is a blending between computation and communication because the computation subsystem can be responsible for control of communication: ordering message queues, tracking packet delivery, powering on/off the radio, etc.

Resource sharing is one challenge, but efficient and reliable communication is typically the larger challenge in wireless sensor networks. The most popular and easy-to-use networks stacks are typically high power (WiFi), low reliability (Bluetooth LE), unsuitable for high density (LoRa), or some combination of these. Low-power, time-synchronized, and reliable (very low packet loss) wireless stacks are rare, and due to their lower popularity typically have fewer easy-to-use libraries provided which makes them more challenging to implement.

Furthermore, end users often prefer a fully abstracted-away wireless link, opting for a "wireless serial port" by which a wireless link is interacted with using a bidirectional UART. Data is clocked into the UART port on one wireless node and clocked out of the UART port on the other wireless node, with no user interaction or configuration required.

1.4 Actuation

The output of any wireless sensor system is often an afterthought. It should be more than just an appended CSV on a hard drive because the purpose of collecting these sensor samples is to learn something and take action as a result. Clear definition of that result, and the output to support it, is critical.

1.5 Power Consumption Assessment

The components in this framework are often the topic of papers and graduate theses focused on the "Internet of Things" class of systems. These implementations often miss the mark on the most important metric, reliability, by failing to achieve long battery life and thereby require frequent and time-intensive maintenance. These short-lived wireless sensor systems often go offline within a few days, severely limiting their uptake by real users.

The root cause of this issue is typically that commercial products are designed to be easy to use and functional for as wide of a customer base as possible. If the product is faster to implement, that is seen as an acceptable trade-off for requiring more batteries. These sponsored capstones are a teaching tool but also aim to result in real performance breakthroughs and enable wireless sensing in application areas where none was available before. Environmental sensing systems in particular call for long battery life to minimize periodic maintenance, which is expensive at best, or impossible due to hazardous environmental conditions at worst.

To that end, a key component in design is careful assessment of the system's per-component power consumption and the intentional spending of electrical energy to achieve the end goal. Power consumption expectations must be tabulated from datasheets, determined if acceptable, and compared with current measurements at fine time scales enabled by such instruments as the Joulescope [12].

Low power consumption can only be achieved through close attention to design, adding a moderate amount of complexity by the aggressive use of sleep modes and interrupts. Often times very simple solutions are overlooked, such as removing LED indicator lights or using transistors to electrically disconnect devices with high standby current when not in use. Time-synchronized wireless communication, rather than always-on radio as with WiFi (or even many always-on wakeup radios) is usually the key enabler of low average power for most wireless sensor nodes. Some sensors consume high current and throw the balance of power consumption towards the sensors, as can be the case with, e.g., hot-wire anemometers.

2. Methods

To date, four projects have been sponsored. The fourth is currently in progress. Each year, one team of 4-5 students has been assigned to the PI's evolving wireless sensor capstone project. Year 1 saw a team of 5, Year 2 saw a team of 4 with one additional summer student to polish the project for publication, and Year 3 saw a team of 4. Year 4, currently ongoing, has a team of 4 assigned. All of the PI's sponsored capstone projects required the following features to intentionally promote convergence on the aforementioned design framework and to achieve the learning goals regarding written technical communication:

- Device contents:
  - Reliable, low-power wireless network link
  - Microcontroller with very low current deep sleep ability
  - One or more environmental sensors
- Device evaluation:
  - Rigorous power consumption measurement at fine timescales compared with expectations developed from published datasheets
  - Analysis of correctness of sensor data. (Does the system really work?)
  - Demonstration of more than one device operating simultaneously
- Documentation of performance and functionality:
  - Data-supported argument that the result (sensor capabilities, data rate/resolution, battery lifetime, size, cost) is compatible with the intended use case
  - User guide
  - Documentation suitable for a future capstone team to build from

Capstone design periods are nominally six months in length, January through June, with an optional orientation meeting before the design period when teams are formed and assigned to projects in December. The team's goal is to be completely finished with all deliverables by June 1. The timeline starts with teams performing a literature and product search to identify methods of solving the environmental sensing problem in a way that meets performance requirements. In these capstones, the second year and beyond were asked to include last year's design in their literature search.

Each team has weekly meetings with their capstone advisor, a faculty member. When capstone projects are sponsored by industry, student teams typically meet with their project sponsor monthly or even sometimes quarterly. In the case of these environmental sensor projects, the project sponsor was another faculty member, and so sponsor meetings were also nominally weekly and combined with the advisor meeting. The advisor provides general guidance about design, schedule, backup plans, team cohesion, task assignment fairness, and so on, and the industry sponsor provides goal clarification, approves feature revisions in response to roadblocks, approves purchases, and gives short-term milestone suggestions.

Project work continues through June of a given academic year with continuous feedback given at weekly meetings. Students perform self- and peer-assessments with respect to teamwork and technical contributions. This feedback is kept private and is accessible only to involved faculty; the results serve as a signal to identify potential team cohesion issues that may require faculty intervention. At the end of the academic year, deliverables are produced and evaluated by the project advisor and sponsor to determine grades. These deliverables include a comprehensive project report along with all written code and hardware designs to form a complete set of project documentation.

3. Results

Each annual capstone project had its unique share of obstacles, but a functional demonstration was made with the provided components each time. Hardware exemplars for the three years of projects are pictured in Figs. 1-3. Each successive year saw further refinements, even as each team undertook design for a different problem. Though clear documentation was lacking, as described later, it may be that simply seeing and understanding the results of the previous year's efforts set new expectations for the next team. Self-assessment specifics must remain confidential, but generally teams performed more cohesively and successfully with each successive year. Project documentation (report, software, hardware designs, etc.) quality improved each year as well, resulting in steadily increasing grades year-on-year. We speculate that documentation and engineering residuals made the scope of the project easier to conceptualize. This in turn may have allowed student teams to better forecast challenges and more effectively plan, leading to better managed individual roles and expectations.

In the first year, the team operated under pandemic lockdown restrictions that made hardware design collaboration difficult. Nonetheless, the team was able to divide tasks and define interfaces appropriately and were the first to produce something resembling the intended framework. They were able to get SmartMesh IP running and routing sensor data, and plotting the results, though with rudimentary hardware that included breadboards.

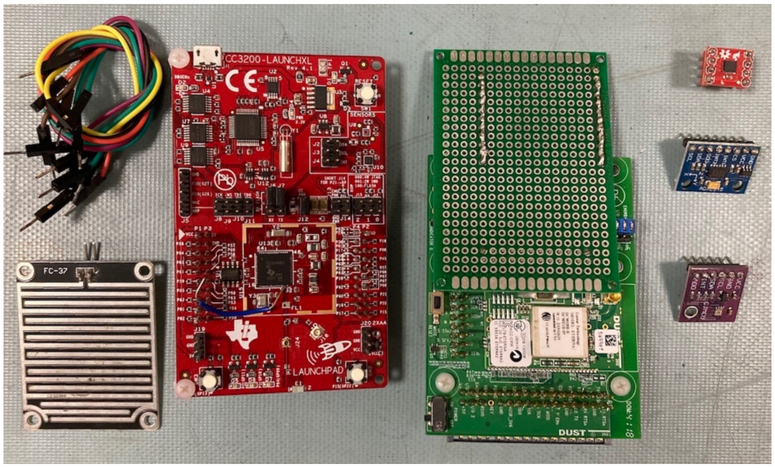

Fig. 1: Collection of hardware used to demonstrate firmware and software developed during Year 1, including moisture sensor (lower left), TI CC3200 development board (red PCB), SmartMesh IP development board with perf board soldered on (green PCB) and various accelerometer and environmental sensor breakouts (3x on right).

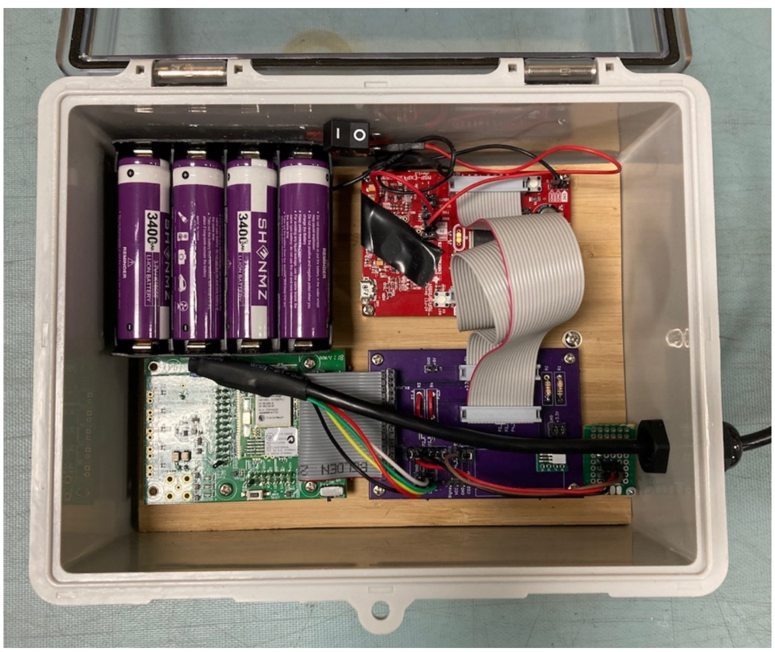

Fig. 2: Plastic housing with lid open with Year 2 sensor node electronics mounted inside. Components include 4x 18650 Li-Ion batteries, TI MSP430FR5994 development board (red PCB), SmartMesh IP development board (green PCB), and custom junction PCB to connect components (purple PCB) with temperature and humidity sensors installed (small green PCBs on top of purple PCB). Also included is a discrete MOSFET (lower right corner) to control power to the nitrous oxide sensor (not pictured, connected via black cable protruding from right side). Project is described in more detail in [3].

In the second year, the team tried to base their design off of the previous year's results so they could take the project further and include a new nitrous oxide sensor. This team learned about the value of documentation after spending several weeks trying to get the prior year's hardware running. They were eventually successful and were able to produce a few packaged devices suitable for field deployment. A custom PCB was included to provide routing between development boards. Over the summer, another undergraduate was brought on to improve stability of the microcontroller firmware and rigorously evaluate the gas sensor performance in a laboratory environment. This student also finalized a paper about the sensor system that was later published [3].

The third-year project showed maturation of the wireless sensor framework. Not only was the team able to get the previous year's code running sooner, but they added indoor air quality sensors with enough time remaining to take power consumption data and use those results to inform design. The third-year team's firmware includes a setting for battery life-driven sampling rates: instead of sampling rate dictating battery life, the target battery life is first given. Measured per-sample sensor energy costs are used to calculate sampling rate. This team also finished a publication-quality paper shortly after the end of the term [4], rather than first needing a few months of summer work to improve the device and its publication.

Fig. 3: Year 3 sensor electronics enclosed by custom laser-cut acrylic case with helpful labels attached. Components include 4x 18650 Li-ion batteries, SmartMesh IP development board (vertically-oriented with white antenna protruding in upper left corner), TI MSP430FR5994 development board (under green PCB), custom junction PCB to connect components (green PCB), Sensirion SCD41 carbon dioxide sensor (obscured by white label at bottom) and Sensirion SPS30-PM2.5 particle counter sensor (green sensor protruding from acrylic case along bottom right corner). Project is described in more detail in [4].

Project residuals in the form of code, hardware, and documentation were of mixed utility to later teams. Code was often the most useful, only because its per-function purpose was little described in the final project report intended to focus on results and user-facing capabilities for a sponsor. The code could instead be read and eventually understood. The most severe lack was in startup documentation: when writing up their project, team members almost universally forgot how far they had come. Reports and documentation could have benefited from detailed software tool (e.g., development environment) installation instructions and screenshots, exact version numbers of software used, python packages installed, connectors used for computer interface, board switch and jumper configuration differences from as-shipped configuration, etc. At final documentation time, students were already taking their development substrate for granted.

For simplicity, the "wireless serial port" model of wireless link interaction is what the teams used for the first three years by leveraging the Analog Devices SmartMesh IP product. In the fourth year, currently ongoing, the team has been tasked with performing computation and communication on a single CPU using the OpenWSN open-source wireless sensor network stack [8]. It is hoped that, in this fourth year, the project will also produce documentation and instructions for a reliable, low-power, long-lived wireless sensor system framework.

In the second year, the team encountered sampling period drift as a result of improper use of a sleep timer. Rather than programming an interrupt with period of e.g., 10 s, the team used a built-in sleep command to invoke low-power mode for 10 s. This had the result of extending sampling period to 10 s plus the awake time (code execution time), leading to a small but growing drift between the assumed clock period and the real sampling period. Built-in low-power commands are easy to use but have hidden downsides that are only evident with testing. In the fourth year, sharing the CPU between communication and housekeeping/computation tasks may prove to be a good way to avoid this sampling period drift issue. The communication stack already includes scheduled events; with that code already present, scheduling sensor sampling may be more straightforward to implement than it was previously.

In all three years, and expected for the fourth year, system actuation of these projects has converged on a status board showing continuously updated data plots. This means actuation is in the hands of the human observing the data. That is acceptable for the problem statements encountered so far, but in future projects the sensor data should be used as part of a control loop such as a PID controller with, e.g., ventilation and/or indoor air cleaning included as part of the loop. Scheduled wireless communication is expected to provide PID loop updates at a predicable rate, which should enable setting an appropriate control loop period [13].

4. Conclusion

This series of capstone projects has been highly successful. It accomplished the four stated overall goals: wireless sensors were designed to solve specific problems (with each successive year improving in quality), the projects showed and continue to show convergence to common core components (The use of MSP430 and SmartMesh IP specifically, general system design consisting of a coordinating microcontroller at the core with a dedicated wireless module treated as a peripheral, and visualization of the data in the GUI), and students have been able to see examples of past work to understand the value of clear technical writing (this is qualitative;

teams are appearing to be better organized over time after seeing past examples). And undergraduate students were certainly involved in the world of research, as described in more detail below. The series has been successful in other ways too, such as producing publication-quality results and has initiated successful engineering careers for many of the students.

In all, two research papers have resulted from these capstone projects which, in turn, include links to data repositories to enable others to build wireless sensor nodes from these projects' source code and hardware designs. Student teams showed clear improvement, year on year: after Year 1, the fundamental components were present, but a cohesive demonstration was challenging and no paper could be written about the results. In Year 2, the project functioned but needed a summer student to fix errors and co-author the publication with the PI. In Year 3, the team had a solid demonstration by the end of the academic year and the paper draft was high quality even before the PI first reviewed it. This last paper was presented at an international conference by a student team member. Two students involved in these projects have gone on to graduate school. Two more are awaiting graduate application results, and at least three students are employed in local engineering industry.

As an incidental finding, we observed each year that initial student assumptions and expectations regarding wireless networking and power minimization were severely lacking at the outset of each year's project. Reliable and low-power wireless communication is still difficult to implement easily, which may contribute to this misunderstanding. Students seem to begin the project with misapprehensions about the capability and suitability of commercially-available, rapidly-prototyped, kit-based wireless links. Students also take time to fully appreciate the impact of system-level design considerations, such as the battery life cost of a status LED or choice of a linear vs. switching voltage regulator, in deeply duty-cycled systems intended for very low average power consumption. Over the course of these capstone projects, students do successfully understand these concepts better as demonstrated by the excellent published results. Using this evolving wireless sensor node framework, there is opportunity not only to solve new environmental sensing application problems, but also to teach more nuanced understanding of the design of such systems that is otherwise difficult to grasp using today's commercially-available hardware platforms.

References

[1] D. C. Burnett, B. Kilberg, R. Zoll, O. Khan, and K. S. J. Pister, "Tapeout class: Taking students from schematic to silicon in one semester," in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), May 2018, pp. 1–5. doi: 10.1109/ISCAS.2018.8351506.

[2] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802.15.4e networks: Performance and trade-offs," in 2015 IEEE International Conference on Communications (ICC), Jun. 2015, pp. 604–609. doi: 10.1109/ICC.2015.7248388.

[3] R. Leon et al., "The Networked Nitrous Node: A Low-Power Field-Deployable COTS-Based N_2O Gas Sensor Platform," IEEE Sensors Letters, vol. 7, no. 6, pp. 1–4, Jun. 2023, doi: 10.1109/LSENS.2023.3276339.

[4] B. P. Hippe, A. A. Dezay, M. A. Garcia, M. C. Newton, J. M. Acken, and D. C. Burnett, “Multi-Node Networked Indoor Air Quality Monitor,” in 2023 IEEE SENSORS, Oct. 2023, pp. 1–4. doi: 10.1109/SENSORS56945.2023.10324899.

[5] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, “A Comprehensive Survey on TinyML,” IEEE Access, vol. 11, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.

[6] S. E. Short and D. C. Burnett, “An Analog Frontend Controller for Continuous Sensor Sampling Without CPU Interaction for Fully-Integrated Single-Chip Sensor Systems,” in 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), Oct. 2022, pp. 0426–0430. doi: 10.1109/UEMCON54665.2022.9965702.

[7] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, “A Realistic Energy Consumption Model for TSCH Networks,” IEEE Sensors Journal, vol. 14, no. 2, pp. 482–489, Feb. 2014, doi: 10.1109/JSEN.2013.2285411.

[8] T. Watteyne et al., “OpenWSN: a standards-based low-power wireless development environment,” Trans. Emerging Tel. Tech., vol. 23, no. 5, pp. 480–493, Aug. 2012, doi: 10.1002/ett.2558.

[9] T. Chang et al., “6TiSCH on SCμM: Running a Synchronized Protocol Stack without Crystals,” Sensors, vol. 20, no. 7, Art. no. 7, Jan. 2020, doi: 10.3390/s20071912.

[10] D. C. Burnett et al., “CMOS oscillators to satisfy 802.15.4 and Bluetooth LE PHY specifications without a crystal reference,” in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Jan. 2019, pp. 0218–0223. doi: 10.1109/CCWC.2019.8666473.

[11] D. C. Burnett, F. Maksimovic, B. Wheeler, O. Khan, A. M. Niknejad, and K. S. J. Pister, “Free-running 2.4GHz Ring Oscillator-Based FSK TX/RX for Ultra-Small IoT Motes,” in 2020 15th European Microwave Integrated Circuits Conference (EuMIC), Jan. 2021, pp. 101–104.

[12] “Joulescope JS110: Precision DC Energy Analyzer,” Joulescope. [Online]. Available: https://www.joulescope.com/products/ joulescope-precision-dc-energy-analyzer

[13] C. B. Schindler, T. Watteyne, X. Vilajosana, and K. S. J. Pister, “Implementation and characterization of a multi-hop 6TiSCH network for experimental feedback control of an inverted pendulum,” in 2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), May 2017, pp. 1–8. doi: 10.23919/WIOPT.2017.7959925.