

Perpetual Sensing: Experiences with Energy-Harvesting Sensor Systems

by

Bradford Campbell

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2017

Doctoral Committee:

Associate Professor Prabal Dutta, Chair
Assistant Professor Mosharaf Chowdhury
Associate Professor Branko Kerkez
Adjunct Assistant Professor Thomas Schmid, University of Utah

Bradford Campbell

bradjc5@gmail.com

ORCID iD: 0000-0002-4103-8107

© Bradford Campbell 2017

To my parents, for their continued support throughout my education.

ACKNOWLEDGMENTS

Looking back on my Ph.D. journey, one thing that has become clear is that even if it is possible to navigate this path alone, it is probably not worth it. I have learned a tremendous amount from the people I have had the privilege of working with, including new research ideas, new technical skills, how to be successful in grad school, and much more. This starts with my advisor, Prabal Dutta, who has wisely guided me towards interesting ideas over the years while making sure that the results get written up too. He has also had patience for me when I get distracted on new projects and the trust that it will be worthwhile. Next are Pat Pannuto and Branden Ghena, whom I've spent many late nights with debating, exploring, and implementing new ideas. Without them to tell me when I'm wrong these last six years would have been much more challenging. I also want to mention Sam DeBruin, Will Huang, and Ben Kempke, who started their own grad school journeys at about the same time I did, and have helped immensely with figuring out exactly what we are doing and how to be successful. Ye-Sheng Kuo, Noah Klugman, Meghan Clark, Thomas Zachariah, Rohit Ramesh, Josh Adkins, and Neal Jackson have all been a great help to me as well, as having so many diverse talents in the same lab truly does make for a better environment and has made my work noticeably better. Then there are the past members of the lab, Aaron Schulman, Sonal Verma, Lohit Yerva, and Andrew Robinson, who helped start Lab11 at Michigan and helped me find my way in this new environment.

I want to thank my committee members, Mosharaf Chowdhury, Branko Kerkez, and Thomas Schmid, whose insights across many fields have greatly benefited my dissertation.

Part of what makes doing systems research so compelling is that collaboration can lead to bigger and more impactful work, and I must thank my collaborators who have helped make the work in this dissertation possible: Ye-Sheng Kuo, Sam DeBruin, Meghan Clark, Branden Ghena, Lohit Yerva, Pat Pannuto, Thomas Schmid, and Prabal Dutta. Without their insight, expertise, support, motivation, and hard work I would not have been able to complete this dissertation.

I also want to thank my family and my friends who may not fully understand this strange journey that has occupied my time for the last few years, but who have been supportive nonetheless. I am also very appreciative for the friends I have met through CSEG, who are willing to give a little time back to the department to make this a better experience for all of us. In particular Pat and Zakir, who have always been willing to create some chaos with me

in the hopes that we leave our corner of the University of Michigan better than we found it. And to everyone who came out on a Monday night, or got lost with me in a foreign country, thank you, I don't think I would have made it this far without you.

TABLE OF CONTENTS

Dedication	ii
Acknowledgments	iii
List of Figures	ix
List of Tables	xi
Abstract	xxii
Chapter	
1 Introduction	1
1.1 The Trillion Sensor Power Problem	1
1.2 Better Buildings: A Pressing Social Need	3
1.3 Energy-Harvesting Sensors for Buildings	4
1.4 Thesis Statement	7
1.5 Contributions of this Dissertation	7
2 Background and Related Work	9
2.1 Pervasive and Ubiquitous Sensing	9
2.2 Wireless Sensor Networks	10
2.2.1 Hardware Platforms	10
2.2.2 Wireless Mesh Networks	12
2.3 Energy-Harvesting Sensors	14
2.3.1 Outdoor Energy-Harvesting	14
2.3.2 Indoor Energy-Harvesting	15
2.4 Building Monitoring	16
2.4.1 Ambient Sensing	16
2.4.2 Occupancy Detection	17
2.4.3 Electricity Consumption	17
3 Gecko: Adding Energy-Harvesting to Wireless Sensor Networks	21
3.1 Energy-Harvesting Options	21
3.2 Indoor Lighting: Enough to Power Sensors	22
3.3 Energy Storage Options	23
3.4 Leaf Node Design: Revisiting Core System Components	25
3.4.1 Power Supply	25

3.4.2	Timekeeping	26
3.4.3	Sensor Operation	26
3.5	Optimizing for Low Energy Operation	28
3.5.1	Parallelized and Optimized Software Startup	28
3.5.2	Synchronizing Energy-Harvesting Nodes	29
3.5.3	Gecko to Low Power Mesh Communication	30
3.6	A Viable Indoor Energy-Harvesting Sensor Node	32
3.6.1	Wakeup, Sense, and Transmit Energy Consumption	32
3.6.2	Sampling Rate in Indoor Environments	32
3.7	Summary	33
4	Monjolo: Energy-Harvesting Energy Meters	34
4.1	Monjolo Design Principles: Measurement Based on Harvesting Rate	34
4.1.1	Basic Operating Principle	35
4.1.2	Monjolo Theoretical Model	35
4.1.3	Dealing with Data Loss	36
4.1.4	Moderating Packet Delivery Rates	36
4.1.5	Handling Non-linear Recharge Rates	37
4.2	Monjolo System Architecture	37
4.2.1	Hardware Platform	38
4.2.2	Sensor Operation	38
4.2.3	Data Aggregator	40
4.3	Monjolo Implementations: Sensing Energy Consumption in Buildings	41
4.3.1	Indoor Photovoltaics Monjolo	41
4.3.2	Thermal Gradients Monjolo	42
4.3.3	AC Power Meter Monjolo	44
4.4	Monjolo Evaluation	49
4.4.1	Sensing Accuracy	50
4.4.2	Activation Rate	51
4.4.3	Comparison to Theoretical Model	53
4.5	Deltaflow: Correcting for Uncalibrated Pulse Streams	53
4.5.1	Dissaggregation Methodology	54
4.5.2	Deltaflow Performance	59
4.6	Summary	61
5	Gemini: Accurate Non-Contact Power Metering by Virtualizing Voltage	62
5.1	Gemini: Decoupling Voltage and Current Measurement	63
5.1.1	Virtualizing Voltage	64
5.1.2	Energy-Harvesting	65
5.1.3	Power Meter System Operation	66
5.2	Implementation Details	67
5.2.1	Voltage Monitor	67
5.2.2	Energy-Harvesting Current Sensor and Power Meter	68
5.3	Gemini Evaluation	71
5.3.1	Accuracy: 8.7% Average Error, 2.2 W Absolute Error	71

5.3.2	Misaligned Current and Voltage Errors	73
5.3.3	Correctly Matching Current and Voltage	74
5.3.4	Design Limitation	74
5.4	Summary	75
6	Triumvi: A Truly Deployable Energy-Harvesting Power Meter	77
6.1	Triumvi: Standalone Meter with Integrated IV Sensing	78
6.1.1	Energy-Harvesting Power Supply	79
6.1.2	Multiplexing the Current Transformer	80
6.1.3	Voltage Measurement	81
6.1.4	System Operation	82
6.1.5	Three Phase Loads	83
6.1.6	Large Loads	84
6.1.7	Detailed Load Analysis	84
6.1.8	Upgrades for Proven Deployments	84
6.1.9	Network and Gateway Operation	85
6.1.10	Deployment Considerations	86
6.1.11	System Variations	86
6.2	Triumvi Implementation	87
6.2.1	Energy-Harvesting Circuit	87
6.2.2	Current Transformer Multiplexing	87
6.2.3	Calculating and Reporting Power	88
6.3	Triumvi System and Deployment Strategy	89
6.4	Triumvi Performance	90
6.4.1	Measurement and Harvesting Range	90
6.4.2	Accuracy	91
6.5	Summary	92
7	Beyond Energy: Building Up the Building Sensing Toolkit	94
7.1	Detecting Door Open and Close Events	95
7.1.1	Buzz: Vibration and Sharp Motion Detector	95
7.1.2	Buzz: 65% of Events Detected	96
7.2	Monitoring HVAC Vents	97
7.2.1	Breeze: Airflow and Air Vent Meter	97
7.2.2	Breeze: 0.125 Samples/Second	98
7.3	Monitoring Interior Lighting	99
7.3.1	BigBen: Light/Occupancy Sensor	99
7.3.2	BigBen: 40 Second Time Jitter	100
7.4	Energy-Quality Tradeoffs	101
7.5	Summary	102
8	An Integrated Modular Architecture for Energy-Harvesting Sensor Systems	104
8.1	Architectural Overview	104
8.2	Design Space of Energy-Harvesting Indoor Sensors	105
8.3	Architectural Layers and Abstractions	106

8.3.1	Energy-Harvesting Power Supplies	106
8.3.2	Triggers: Decoupling Harvesting from Computation	107
8.3.3	Sensing on a Limited Energy Budget	108
8.3.4	Data Communication or Storage	109
8.4	Architectural Evaluation	110
8.4.1	System Architecture In Related Work	110
8.5	Summary	112
9	Deployment Experiences	113
9.1	Non-Contact Energy-Harvesting Energy Metering	113
9.1.1	Installation Overhead and Deployment Speed	114
9.1.2	Dataset Efficacy	114
9.2	Monjolo Indoor Light Sensing	119
9.2.1	Installation Overhead	120
9.2.2	Sensor Performance	120
9.3	Gateway Support Infrastructure	126
9.3.1	Single-hop Networking	126
9.3.2	Gateway Software Architecture	127
9.3.3	Data Collection, Processing, and Visualization	129
10	Conclusion	131
Bibliography		134

LIST OF FIGURES

1.1	Energy-harvesting sensors for building metering and monitoring. Sensors monitor key points in buildings, including lighting, electricity consumption, door state, and ambient conditions. They leverage a shared gateway platform and cloud backend for data collection and analysis. This building-centric toolkit leverages a variety of techniques to enable perpetual sensors that can help improve buildings.	6
3.1	Energy availability from indoor lighting. (a) shows the average irradiance of four typical offices over the course of a partly sunny day (good) and a rainy day (bad). (b) shows daily irradiation across 13 sensors deployed in four offices over a one week period including sunny and rainy days. Although there is nearly a 50x difference in harvested energy across space and time, a roughly $10 \mu\text{W}/\text{cm}^2$ lower bound average radiance exists.	22
3.2	Irradiance (on a log-linear scale) over a typical week in four offices, oriented in four different directions. Each office is equipped with two to four typically wall-mounted sensors, each facing in different directions within the office, to capture different light environments. The horizontal lines correspond to the irradiance necessary for a leaf node to send a message every minute given the respective light source type. On most days, most leaf nodes have enough light to send a message once a minute, but on some days, some do not. This highlights the importance of node placement and solar cell sizing. It also illustrates the need to support a wide operating dynamic range.	23
3.3	Ragone plot of energy storage technology from established technology to research prototypes. A thin film battery [101], a Panasonic lithium-ion battery [88], an aluminum electrolytic capacitor [38], a tantalum capacitor [25], an activated carbon supercapacitor [38], a reduced graphene film micro-supercapacitor (MPG-MSC) [148], a laserscribed graphene micro-supercapacitor with fumed silica nanopowder (LSG-MSC FS-IL) [37], a fiber-based micro-asymmetric supercapacitor (ASC) [63], and a lithium-ion hybrid capacitor (LAN25) [123] are compared. Batteries excel at energy storage and offer one to two orders of magnitude better energy density per volume than other options, but have limited discharge rates. Capacitors can supply power quickly, but are limited in their capacity. Commercially available supercapacitors sit in the middle, and research prototype supercapacitors are increasing both power density and energy density.	24

3.4	The power supply stores indoor solar currents of just a few microamps and provides periodic burst of regulated voltage. The power supply exports several monitoring and control signals to allow applications to tailor their operation to the key power supply events.	26
3.5	(a) Operation of the energy harvesting power system. The capacitor voltage is seen dropping from about 2.5 V to about 1 V. The boost regulator starts up and establishes a 3.0 V supply until the core requests shutdown after it finishes its radio transmission. <i>Radio Boot</i> refers to the execution of the first instruction, enable radio regulator. (b) Stages of an optimized TinyOS cold-boot overlaid with the processor and radio current. Our design achieves cold-boot to packet transmission in less than 4 ms using all of the optimizations discussed. The net effect is a reduction in TinyOS cold boot by a factor of 69 \times in time and 14.6 \times in energy.	27
3.6	Solar cell current vs voltage, and power vs voltage, under low ($12.7 \mu\text{W}/\text{cm}^2$), medium ($53.8 \mu\text{W}/\text{cm}^2$), and high ($101.7 \mu\text{W}/\text{cm}^2$) indoor lighting levels. Although solar cell output current varies by an order of magnitude, the maximum power point varies within a small range of approximately 2.6 V to 3.0 V for (a) and 2.1 V to 2.3 V for (b), or about (10-15%). This lets us use fixed-threshold voltage trip switches for charge/discharge operation under typical indoor lighting conditions. (c) MCP1640 was chosen due to its efficiency over a large input voltage range, and because it has output disconnect and low leakage current.	28
3.7	Achieving initial synchronization. Methods for achieving initial synchronization include both asynchronous neighbor discovery and synchronous event-triggered discovery. (a) A free-running leaf transmits a packet when it has sufficient energy. A triggered leaf transmits at a well-known interval. (b) Each point represents the duty cycle of each minute. Each step in the graph corresponds to the discovery of a leaf node. Even while synchronized with five leaf nodes, the branch node's duty cycle only slightly increases from 2.45% to 2.75%. (c) A trace of synchronous event-triggered discovery is shown. Leaf (N1) and branch (N2) nodes both respond to a sudden change in light level (triggered by a zero-power light detector circuit), choose a short backoff, and transmit packets back and forth until the leaf depletes the energy in its storage capacitor.	30
3.8	Gecko sample and transmission rate for a range of indoor lighting levels and multiple solar cell types. Gecko is able to work over three decades of indoor irradiance levels. At an irradiance level of about $40 \mu\text{W}/\text{cm}^2$, Gecko is able to transmit once per minute which is a standard sample rate for an indoor ambient monitoring sensor.	33
4.1	Theoretical response rate for Monjolo.	35

4.2	Example Monjolo energy harvesting operation. A trace of the energy-harvester charging up the main reservoir capacitor, C_{store} . When the capacitor reaches 5.1 V, the microcontroller boots up. The microcontroller samples the V_{timer} line to determine whether to send a packet or not. When the node wakes up at 5.6 s there is only approximately 0.6 V on the timing capacitor, causing the node to send a packet before recharging C_{timer}	38
4.3	RPi Gateway. The gateway is an IEEE 802.15.4 packet receiver shield based on the TI CC2520 radio paired with a Raspberry Pi mini-computer.	40
4.4	Indoor lighting Monjolo. The power supply of Gecko is re-designed to be the form factor of a commercially available solar panel and the remaining Monjolo hardware components are sized to fit on a matching daughterboard. This sensor can easily and unobtrusively placed near a light.	41
4.5	Activation rate of indoor lighting Monjolo from incandescent and fluorescent lights.	42
4.6	Activation rate of Thermes at a range of pipe temperatures in 23 degrees C ambient conditions. As the temperature difference between each side of the Peltier module increases, so does the average rate of activations over the first five minutes of harvesting.	42
4.7	Thermes device attached to a pipe. Thermes is designed to be a bracelet that can easily stick around a hot water pipe.	43
4.8	Instantiations of AC Monjolo. An evaluation version allows for adjusting various capacitance and operational settings to explore how design choices affect the system. The plugload version uses an internal solid-core current transformer and can meter any plug-load. The split-core version can clip around a wire in a circuit panel to meter the entire circuit.	44
4.9	Transfer function of the current transformer used in the energy-harvesting power supply when connected to a 430Ω resistor. The power the coil can deliver varies non-linearly as the load power changes. This affects the activation interval of the sensor, causing the activation rate to be non-linear with load power.	45
4.10	The charging cycle of two AC Monjolo sensors with a half-wave rectifier. AC Monjolo #1 has a slightly higher load attached and is able to charge up slightly faster than AC Monjolo #2. However, after the discharge phase both AC Monjolo sensors start charging at the same point in time, causing their wakeup frequency to equalize with only a slight phase shift. This lack of continuous charging leads to a stair-step relationship between activation rate and load power, which can result in a loss of precision, particularly at high load powers.	48

4.11 Frequency of Monjolo activations over a range of loads. (a) shows that as the primary load's power increases from zero to 475 W, the activation rate increases from zero to 8.5 Hz. Although not linear, this monotonically increasing relationship between activation rate and load power suggests that Monjolo may be a viable approach to metering a range of AC loads. In (b), the activation rate is quantized. If the LTC3588 energy harvester is permitted to continue to supply energy to C_{store} during the discharge phase of operation, then the quantizing effect inherent in the system is substantially more pronounced. Particularly at high load powers, the steps span a nearly 50 W range, which dilutes precision and introduces $\pm 5\%$ error.	50
4.12 Accuracy of AC Monjolo's power measurements over a range of different loads. When the load is constant, AC Monjolo's error is very low at only 1%. When the load changes, AC Monjolo's relatively slow response time causes sharp instantaneous error while the data aggregator waits to receive a packet informing it that the load has changed. When the load is very small, as it is at $t = 500$ s, AC Monjolo is unable to transmit and the resulting error is quite large, although a less naive power estimation algorithm would be able to correct for this. Ignoring the few sections where AC Monjolo is drastically incorrect, AC Monjolo shows an average error of 3.7% across a randomly varying resistive load.	51
4.13 AC Monjolo performance on loads with non-unity power factors. Each figure shows the ground truth power and power factor measurement of the load, and AC Monjolo's estimate of the power. If the power factor remains constant, AC Monjolo estimates the power with a constant offset, as shown in 4.13a and 4.13b. The heat gun in 4.13c is an interesting case when the high frequency oscillations cause AC Monjolo to estimate an average value. The fluctuating power factor in 4.13d causes AC Monjolo to sharply overestimate, but it quickly recovers. While non-unity power factors cause AC Monjolo to lose accuracy, we believe the magnitude offset can be compensated for with other mechanisms.	52
4.14 Activations/second for two coils over a range of resistive loads. The first coil is a solid-core current transformer (CT) with 10 windings of the AC phase line around it. The second is a split-core CT. Below 5 W for the solid-core and below 117 W for the split-core the device is unable to harvest enough energy to activate and therefore is unable to perform any metering. Beyond that, as the load power increases, the ability for the node to harvest increases.	52
4.15 Theoretical response rate for Monjolo compared to observed response for AC Monjolo.	53
4.16 System architecture. The Deltaflow server takes in aggregate power draw measurements and the activation frequencies of energy-harvesting sensors attached to individual loads. By augmenting aggregate power measurements with sensor activation (or pulse) frequencies that are correlated with power draw, Deltaflow determines the individual energy consumption of the sensed loads. The system is able to function even if some loads remain unsensed.	54

4.17	Effect of hidden loads. The change in aggregate power at time t_0 is attributed to the load metered by Sensor 1 whose frequency changes at that time. The aggregate also changes at t_1 , but since neither sensor changes, we attribute the change at t_1 to a hidden load and ignore state C. This allows us to discard the effects of non-aliasing uninstrumented hidden loads.	57
4.18	Deltaflow performance on a realistic tree of loads. Figure 4.18a shows the hierarchy of loads and the corresponding sensors. Figure 4.18b shows the ground truth power traces for the aggregate and each load. Figure 4.18c shows the Deltaflow estimates for all five loads and an estimated aggregate for comparison purposes. Figure 4.18d shows the breakdown of overall energy usage by each load and the estimate provided by Deltaflow, as well as the percent of the total energy usage each load represents. While the error for loads (C) and (E), which change power states at similar times, is high (>40%), Deltaflow handles loads that transition independently successfully. Also, Deltaflow is able to successfully integrate streams from multiple types of pulse counting sensors.	59
5.1	Gemini overview. A single voltage monitor (V) continuously monitors the voltage channel of the AC circuit. Energy-harvesting power meters (EH) use current transformers to meter each load. To calculate true power, the EH meter requests the voltage parameters from the voltage monitor, samples the local current waveform, synthesizes the voltage waveform, computes power, and transmits the result to a receiver which forwards the data to the cloud. Maintaining timing across domains is key to the success of this approach.	63
5.2	Voltage and current waveforms of eight loads. The current waveforms vary wildly, but the voltage waveforms are consistently sinusoidal and can be represented by one or just a few Fourier coefficients. The sinusoidal nature of the voltage waveform makes it a better candidate to be virtualized, thus supporting our architectural choice to virtualize voltage.	64
5.3	Comparison of two voltage waveforms at physically disparate points in the same electrical system. Measuring the voltage at two different loads with different power draws on different circuits tapped off of the same leg of a transformer results in voltage waveforms that are nearly identical. This suggests that measuring voltage at a single watchpoint is sufficient for calculating the power draw of multiple loads.	65
5.4	Voltage and current waveforms. Ground truth current and voltage waveforms are shown above their synthesized/sampled counterparts. The voltage waveform is synthesized from the parameters sent by the voltage monitor and the current waveform is sampled from the current transformer monitoring the load. The discrete waveforms are multiplied together, point-by-point, to compute the load's power draw.	66

5.5	Sensing hardware prototypes. The voltage monitor is a plug-load form-factor meter that can plug in to an outlet near a circuit panel box to retrieve the local voltage signal. The energy-harvesting node is prototyped with a solid-core current transformer but the hardware design extends to a split-core CT without modification.	67
5.6	Transmitting AC voltage phase information. After the energy-harvesting node requests the voltage parameters, the voltage monitor responds with parameters of the voltage waveform such as Fourier coefficients, phase, and frequency. Phase is encoded as the time offset (t_0) between the response packet's SFD signal and the voltage signal's most recently rising zero-crossing. The energy-harvesting node uses its local timestamp of the SFD signal to phase align the current waveform samples to the voltage channel without requiring synchronized clocks.	68
5.7	Energy-harvesting power supply and measurement circuit. One current transformer is used for harvesting and the other for measurement. Future implementations could multiplex one current transformer for both operations.	69
5.8	Energy-harvesting operation. V_{CAP} is the voltage on the 600 μF capacitor and V_{CC} is the 3.3 V supply voltage.	69
5.9	State machine for the energy-harvesting power meter. Nodes use the timing capacitor to rate-limit sampling and determine when to compute power.	70
5.10	Accuracy over a range of resistive loads. We meter a range of loads with a unity power factor using Gemini. The x-axis shows the ground truth power draw of the load. In the bottom graph, Gemini's estimate is overlaid on the line representing an ideal device. The middle graph shows the percent error of the estimate, and the top graph shows the absolute error of the estimate. The average percent error is 1.2% and the average absolute error is 0.61 W. The error grows when the load is greater than 95 W due to ADC clipping of the current waveform. These results demonstrate that Gemini is a viable power meter for resistive loads.	72
5.11	Power metering accuracy for four real-world loads. We meter four loads with a ground truth meter and Gemini. Gemini is able to mirror ground truth well with the highest error appearing when loads either rapidly change power draw or when loads draw very little power (e.g. drawing ~1 W). For instance, because (c) has a low power draw, the average percent error is relatively high at 17.2%, but the average absolute error is only 1.0 W. The highest average absolute error is 4.9 W for (d), mostly due to the current waveform clipping after the 55 minute mark. These test cases demonstrate Gemini's viability as a power meter for real-world loads that operate within Gemini's (configurable) operating range.	72
5.12	Effect of timing offsets when calculating power. Even 1 ms of phase error can result in 50% measurement error.	74

5.13	Power calculations with different phase errors. The instantaneous power for the eight loads in Figure 5.2 is calculated with the current and voltage waveforms correctly aligned and with the current waveform shifted as would happen if the incorrect voltage leg of a transformer was used. In all eight cases the correct result is the greatest positive result, suggesting that using <code>max()</code> as a heuristic is often sufficient for correctly selecting the voltage channel . . .	75
6.1	Triumvi system overview. Meters clip to wires exiting circuit breakers and harvest energy to power themselves, detect voltage phase, sample AC current, and wirelessly transmit true power measurements. Optionally, they share charge, measure synchronously, or use an external voltage signal or power source. The nearby gateway collects the measurements and publishes them to local users, existing BMS/SCADA systems, local storage, or cloud services. This metering system, with its easy-to-install sensors that support flexible metering options, leverages the benefits of energy-harvesting operation to make effective circuit-level energy meters.	78
6.2	Triumvi system block diagram. The three main system components (harvesting, current, and voltage) use inputs from the split-core current transformer and voltage sense conductor to provide the processing core power, and scaled current and voltage signals, to compute AC power. The core's real-time clock (RTC) and low-energy memory (FRAM) can time-stamp and store data locally when the network is unavailable. An expansion header allows external voltages to be used (for the voltage channel signal or to power the meter, if higher accuracy or resolution is required).	80
6.3	Triumvi state machine. Once Triumvi has booted due to sufficient energy, it iterates through setup, measure, and reporting phases before waiting for its energy store to replenish.	82
6.4	Triumvi configuration expansion board and high current version. This configuration attachment contains three knobs for setting which panel the meter is in, and which circuit it is metering. It also includes an optional battery, a recharging circuit, and 2.5 mm and 3.5 mm jacks for charge sharing, synchronous measurements, external power, and external voltage waveform signals. The large load meter can measure 20 A to 1000 A loads.	83
6.5	Gateway hardware. The Triumvi gateway contains an Intel Edison [57] and a dedicated CC2538 [130] co-processor for receiving the 802.15.4 packets from Triumvi.	85
6.6	Current transformer multiplexing circuit. This circuit is responsible for switching the CT between harvesting and measurement operation. When EN is high, the MOSFETs connect the inputs of the CT across the burden resistor through diodes D1-D4 and enable current sampling. When EN is low, the PMOS FETs stay off, preventing any current flow through R_{BRDN} and enabling harvesting. Multiplexing the current transformer is essential for saving size and cost by requiring only a single CT.	87

6.7	Current and voltage trace of a Triumvi measurement. The voltage on the storage capacitor, PGOOD, VCC, and the current draw of the microcontroller are shown while Triumvi measures power. At the start (time equals zero) Triumvi enables the voltage measurement circuit and waits 300 ms (this initial event and waiting period are omitted from the figure but included in timing and power calculations). It then enables the current measurement circuit and waits 50 ms for the output to stabilize. Next it waits for a rising zero-crossing of the AC voltage signal before sampling the current for 16.6 ms. After sampling it computes power, forms an encrypted packet, and transmits the packet.	88
6.8	Installation time for 14 circuits. As Triumvi devices are configured with the panel ID and circuit number, and clipped around the correct circuit, they begin transmitting power measurements. Each mark represents a received packet, and the start time of each horizontal line marks when the Triumvi for that circuit was installed. Based on the start time of the last circuit, installing 13 meters took about seven minutes for a familiar, but inexperienced user. Including the time for removing and reinstalling the front panel (required for all CT-based installations) and first circuit install, we can reasonably assume a half hour to instrument a 14 circuit panel.	89
6.9	Activation rate as a function of primary load. Triumvi's measurement rate is observed for a range of loads with power factor of 1.0. Below 1.25 A, Triumvi is unable to harvest enough power to successfully measure, but Triumvi can sample at nearly 0.5 Hz with loads above 3 A, which is rate-limited to mitigate wireless channel congestion.	90
6.10	Triumvi accuracy with three real-world circuits. Triumvi is compared against ground truth for three circuits representative of three different environments where circuit panel meters may be deployed. The circuit in 6.10a contains two desktop computers, four monitors, and various lab equipment, as may be present in an office building. Even with a power factor consistently below 0.9, Triumvi is able to meter the circuit with 5.13% RMS error. 6.10b shows a circuit with LED lighting, surround sound audio, and a desktop computer, as may be present in a residential setting. Triumvi slightly under-reports power and overall RMS error is 6.47%. The third circuit, 6.10c, contains a rack of servers as may exist in a more industrial setting. The servers are a very constant load which Triumvi underestimates but otherwise tracks well, including two spikes around 170 minutes into the test. For that circuit, Triumvi displays only 3.89% RMS error. These traces demonstrate Triumvi's ability to meter real-world circuits.	91
6.11	Triumvi error with and without voltage phase information. For a range of loads with power factor of 0.85, the RMS error of Triumvi's power measurement using only RMS current and using sampled current plus voltage phase information is shown. Without any voltage information the error averages 18.5% RMS, and with phase information the average error is 2.5% RMS.	92

7.1	Vibration sensor. A zero-power piezoelectric wakeup trigger activates the sensor when it is moved. When placed on the edge of a door, the sensor can be used to detect door open and close events.	95
7.2	CDF of the interval between door open events. Plotted on a log scale x-axis is the CDF of time intervals between subsequent door opening events of a door over a month period. Also shown are the recharge times for the solar based energy-harvesting power supply in different lighting conditions. Sensors in rooms with natural light would be able to detect most door open events, and even in moderately lit rooms at least 65% of door open events would be detected.	96
7.3	Door open events across a 24 hour period. Buzz was deployed on the door of a lab with moderate lighting and no exterior windows. Each vertical line represents a door open event as detected by a ground truth sensor and the energy-harvesting Buzz sensor. The red dashed lines in the ground truth row are door open events that Buzz did not detect. Buzz successfully detected 66 out of the total 100 door open events. This aligns well with our expected detection rate from Figure 7.2.	96
7.4	Breeze airflow sensor. The solar energy harvester is placed near a light source, such as a window, and a MEMs airflow sensor is placed near an air vent. When the harvester has sufficient energy it samples the airflow sensor and reports its readings. Decoupling the harvesting from the measurement point enables both faster sampling and a smaller sensor.	97
7.5	Airflow metered over time. A Breeze sensor is placed near an air vent as the vent cycles on and off and changes its flow volume. The airflow sensor is uncalibrated and therefore we don't include y-axis units. While noisy, the data from Breeze are sufficient to determine when air is blowing from the vent and to determine when the flow changes.	98
7.6	Light based occupancy sensor. The energy-harvesting sensor detects state changes in the measured light by detecting when the sensor transitions between different harvesting rates. For example, in a room with only a single source of light, the sensor can detect when the light turns on by observing when the sensor transitions from not harvesting to harvesting. With more ambient light in the room this transition may not be as stark, but there will still be a detectable change in harvesting rate. Timestamping to determine the rate of harvesting is done with a real-time clock.	99
7.7	Light state over time. BigBen monitors the on/off state of a light fixture for 24 hours. It detects change events well and shows only a 40 s offset error from ground truth, likely due in part to jitter errors when initializing the RTC.	101

7.8	(a) shows a CDF of detected door events for a range of rechargeable energy storage capacities. We assume 25 μ W of harvesting between the hours of noon and 2 AM, no harvesting between 2 AM and noon, 3 μ W of leakage, and 1 mJ per door event detection. We then analyze the door events from Figure 7.2 to determine how many events Buzz could detect with varying amounts of onboard energy storage. (b) shows how many door events could be detected or how long the sensor could last without harvesting and still detect the next door event for a range of energy storage capacities.	102
8.1	Energy-harvesting node architecture and interfaces between physical components. The physical interface is effectively a shared bus as the various layers (power supply, controller, and sensor) stack and share the same signals. Not every layer must use every signal, however. For example, the power supply has no use for the generic GPIO pins, but the sensor modules often use them for ADC readings or a digital bus interface.	105
8.2	Composing the components in this architecture can help realize new sensors for different applications. A sensing system is composed of a harvesting frontend matched with a control and wireless board plus any additional needed sensors. The created systems can then be used to monitor different aspects of a building.	110
9.1	Multiple generations of Triumvi meters installed inside of circuit panels by simply clipping them around the conductor to be monitored.	114
9.2	Triumvis charging before configuration. The Triumvi storage tray can charge up to ten Triumvis, and it is also a convenient jig for setting the panel and circuit knobs before deploying inside of a panel.	115
9.3	Circuit-level power breakdown for a house over the course of one day. The time axis is in GMT/UTC.	116
9.4	Energy breakdown by circuit for a house over one day. With Triumvi, the power measurements can be used to disaggregate energy based on each circuit. 117	117
9.5	Number of measurements recorded for each circuit. Three circuits reported no measurements, and circuit #13 reported very few (1281).	117
9.6	Observed measurement rate of the Triumvis on each circuit. Shown are the minimum, first quartile, median, third quartile, and maximum of the intervals between samples from each circuit. Typically each sensor reports every 3.4 seconds. However, the maximum interval is significantly higher for many circuits because of intermittent loads.	118
9.7	Time ranges over which the Triumvi sensors report readings. Each horizontal bar indicates the time span over which the sensor reported measurements. The data missing from even numbered circuits between 20 and 40 are due to the absence of circuit breakers (and circuits) at those positions.	119

9.8	Reporting time for each circuit. The time is calculated as the delta between the first packet seen from the sensor and the last packet seen from the sensor in the dataset. We consider the length of the experiment to be the longest reporting time (30 days). Most circuits reported at least occasionally for 25 days or more. Six circuits reported over a much shorter window, which could be the result of sensor failure or circuits with rarely used loads.	120
9.9	Short-term deployments in three homes. The energy consumed by each circuit in three panel boxes and the aggregate was measured with our system and a ground truth meter over the course of approximately 2.5 days. Across all of the circuits, our system shows an average error of 3.7%, typically underestimating the correct value. This level of accuracy is sufficient for many energy audits, and is capable of providing insight on where additional metering may be warranted.	121
9.10	The Ligeiro implementation of the indoor lighting Monjolo sensor.	121
9.11	For each sensor we estimate how long the light to which it is attached is powered per day. We assume a light is on if we receive two packets from the Ligeiro in the same minute, and we average over the total number of days between the first packet we saw and the last.	122
9.12	Using the estimate of daily on time, we then compare the cost of running the light at that duty-cycle for a year using three different lighting types. For comparison we assume all lights are a 60 W incandescent equivalent. This type of analysis can be used to determine which lights are economically viable to upgrade given a certain payback period.	122
9.13	Overall packet reception ratio (PRR) for each Ligeiro node. Ligeiros transmit an incrementing sequence number in each packet, and we use this to determine the percentage of packets we received from each sensor. The variability in reception rates is likely due to interference in the wireless spectrum, the range between the sensor and the gateway, and intermittent gateway failures.	123
9.14	We observe the median transmission interval of each sensor to understand how often the sensor can transmit when the light is on. Note the y-axis is on a log scale. For most sensors the Ligeiros can transmit more frequently than once every 10 seconds. The sensors that exhibit activations at a slower rate likely are attached to lights that are infrequently or never used.	123
9.15	Comparing median transmission time to estimate light usage. For sensors with median packet reception intervals that are below about 10 s, it is difficult to correlate with light usage. However, if a sensor's packets are received very rarely and the median interval is much higher (greater than 100 s), as expected the light can be assumed to be used very little or never.	124
9.16	Average and standard deviation of consecutively dropped packets. The bold lines are the average number of packets that are dropped consecutively, and the error bars is the standard deviation. Nearly all of the Ligeiros on average had fewer than 10 packets dropped before a successful packet reception. This shows that packet failures are intermittent and prolonged outages are not common.	125

9.17	Histogram of the number of packet drop lengths seen by each sensor. Generally, no more than 50 packets are dropped consecutively.	125
9.18	Multi-radio gateway hardware and case. The gateway is powered by an Intel Edison, and contains IEEE 802.15.4, BLE, Wi-Fi, GSM, and GPS wireless radios. It also supports Ethernet, micro SD cards, and USB, and includes two status LEDs.	125
9.19	Software architecture for the gateway. At the top, a number of receiver blocks are capable of accepting packets from a wide array of devices. The MQTT broker on the gateway handles routing packets from the receivers, to various processing blocks, and out to external services or databases. Additionally, other services running on the gateway provide interfaces for discovering gateways, writing applications, and interacting with devices.	127
9.20	Gateway-hosted webpages. The first shows circuits and Triumvis in real-time. This helps with deployment by showing which Triumvis are reporting for which circuits. The second allows for locally stored data to be downloaded for offline processing.	128
9.21	Cloud service architecture. The gateway forwards collected packets to the cloud service, which uses a metadata repository to add metadata to incoming messages, and stores data in an InfluxDB database. A user can then query the database through the Grafana graphing and visualization interface.	129
9.22	Cloud-based graphing software. An example graph of Triumvi power data using the Grafana graphing application.	130

LIST OF TABLES

2.1	AC metering related work.	17
8.1	Energy-harvesting node architecture properties for a variety of building monitoring applications. This table highlights a number of potential building monitoring applications and the system properties that are likely sufficient for implementing an energy-harvesting sensor for the particular application. Each subsystem—power, trigger, sensing, and communication—needs one property to enable a functioning sensor. Some applications are best suited by event detection, some just need to detect harvestable energy, others are best served with periodic sampling, but all support intermittent power supplies making them candidates for energy-harvesting sensors.	106
8.2	Common interface usage by shared components. The components in Figure 8.2 represent each column and the rows are each signal. The GPIO pins are mapped to multiplexed pins on the processor modules to ensure they are flexible enough to support a variety of sensor modules. For example, the airflow sensing board uses one GPIO to enable the airflow sensor, and one GPIO as an ADC input to measure the sensor output. The timekeeping board uses the GPIO pins as a digital bus to allow the processor module to communicate with the real-time clock.	111

ABSTRACT

Industry forecasts project the number of connected devices will outpace the global population by orders of magnitude in the next decade or two. These projections are application driven: smart cities, implantable health monitors, responsive buildings, autonomous robots, driverless cars, and instrumented infrastructure are all expected to be drivers for the growth of networked devices. Achieving this immense scale—potentially trillions of smart and connected sensors and computers, popularly called the “Internet of Things”—raises a host of challenges including operating system design, networking protocols, and orchestration methodologies. However, another critical issue may be the most fundamental: If embedded computers outnumber people by a factor of a thousand, how are we going to keep all of these devices powered?

In this dissertation, we show that energy-harvesting operation, by which devices scavenge energy from their surroundings to power themselves after they are deployed, is a viable answer to this question. In particular, we examine a range of energy-harvesting sensor node designs for a specific application: smart buildings. In this application setting, the devices must be small and sleek to be unobtrusively and widely deployed, yet shrinking the devices also reduces their energy budgets as energy storage often dominates their volume. Additionally, energy-harvesting introduces new challenges for these devices due to the intermittent access to power that stems from relying on unpredictable ambient energy sources. To address these challenges, we present several techniques for realizing effective sensors despite the size and energy constraints. First is Monjolo, an energy metering system that exploits rather than attempts to mask the variability in energy-harvesting by using the

energy harvester itself as the sensor. Building on Monjolo, we show how simple time synchronization and an application specific sensor can enable accurate, building-scale submetering while remaining energy-harvesting. We also show how energy-harvesting can be the foundation for highly deployable power metering, as well as indoor monitoring and event detection. With these sensors as a guide, we present an architecture for energy-harvesting systems that provides layered abstractions and enables modular component reuse. We also couple these sensors with a generic and reusable gateway platform and an application-layer cloud service to form an easy-to-deploy building sensing toolkit, and demonstrate its effectiveness by performing and analyzing several modest-scale deployments.

CHAPTER 1

Introduction

“If you can’t measure it, you can’t manage it” is a quote often attributed to Peter Drucker, and while he meant it in the context of running a business, the sentiment applies much more broadly. Only after understanding the current state is it possible to devise reasonable improvement plans, and only after measuring the changes is it possible to evaluate their success. Historically, the primary limitation to measurement was human bandwidth—any measurement required a person to be involved. Today, however, that is not the case. Computing has enabled monitoring far beyond what we could accomplish ourselves. And as this computer driven monitoring continues to have a positive impact for business efficiency, personal health, public safety, environmental sustainability, and a host of other areas, the demand to increase what we can measure, and therefore manage, is going to continue to grow long into the future.

One of the primary enablers of this growth lies with sensor technology. Sensors are present in every new smartphone, new car, and new building made today. These embedded monitors significantly enhance the operation of these objects, but as their numbers grow the complexity of effectively utilizing them also increases. This shift in computing led by these sensors and other pervasive computing elements presents far-reaching opportunities, but also raises many new challenges.

1.1 The Trillion Sensor Power Problem

Some industry projections forecast that 200 billion sensors will be networked by 2020—an order of magnitude increase from 2015 [58]. Within the next decade that number is projected to grow to a *trillion*, and within two decades we could see *ten trillion* networked sensors [125], or well over 1,000 devices per person. This scale out in computing presents exciting opportunities for smart cities, implantable health monitors, autonomous robots, driverless cars, responsive buildings, instrumented infrastructure, and many other applica-

tions where adding monitoring, control, automation, or sensing could dramatically change or improve how those systems operate. For example, fine-grained monitoring of air pollution could lead to policy changes for vehicle routing and improved health for city occupants. Distributed power grid sensing in developing countries could better allocate maintenance resources and lead to higher electricity reliability. Factories and industrial operations could supervise their equipment to ensure their operation stays productive. While we are starting to see this transition towards data-driven improvements today, growth in the number of deployed sensors will accelerate, making these and many other applications possible.

With the device scale required to realize this diverse set of applications comes many challenges related to successfully employing so many devices. Current methods of networking, managing, orchestrating, discovering, and powering devices will likely not scale to support the trillion sensor vision. Communication protocols for retrieving data from and sending commands to devices will have to evolve as the number of networked devices increases. Metadata management, device commissioning, and device replacement will be amplified concerns that today are manageable in small numbers. Many applications will not rely on a single device or type of device, but will instead require ensembles of devices that are programmed together to achieve a goal, and these interconnections remain difficult to express and program today. With a large number of deployed devices present in any given space, simply discovering what resources are available for programmers, users, and even other devices becomes a significant challenge.

But one critical issue is perhaps the most fundamental: how are these devices going to be powered? Sensing, processing, communication, and actuation requires energy, but providing energy to a trillion or more distinct physical points presents a significant challenge. Current solutions for providing energy prove inadequate at large scale. Using wall-power or frequent charging, as is typical with many classes of computing, including servers, desktops, and smartphones, becomes intractable when devices heavily outnumber outlets. The common approach to powering embedded sensors is by using batteries that can provide the node with energy for a year or more. At scale, however, batteries present an intractable maintenance overhead and tremendous waste. To illustrate, consider a typical wireless sensor, like a Fitbit [43], that draws an average current of $150 \mu\text{A}$. With a small coin-cell battery, the sensor will last 60 days, but at trillion sensor scales, every person on Earth will have to replace a sensor's battery every *1.2 hours*—an unsustainable proposition.

Clearly, realizing this level of device scale will require new techniques for powering these devices. One promising option is to employ energy-harvesting, where the device collects energy at runtime rather than being deployed with its entire energy allotment. These devices scavenge energy from the environment from a variety of sources, including solar, heat,

vibration, and others, to power themselves. The advantage of this approach is it can enable perpetual operation: as long as the device can continue harvesting, it can continue sensing, and without the need for battery replacement. This benefit comes at a cost, however, as energy-harvesting also entails very limited energy budgets and intermittent and unpredictable energy availability. We claim that reaching the scale of the trillion sensor vision will require embracing energy-harvesting while also addressing the challenges of moving from the stability and predictability of batteries to the intermittency and unpredictability inherent in energy-harvesting operation.

1.2 Better Buildings: A Pressing Social Need

Of the numerous potential applications for pervasive and perpetual sensing, we identify one that is ripe for additional sensing: improving existing buildings. In the U.S. alone, buildings consume more than 40% of the total energy at a cost of more than \$430 billion [136]. Reducing this demand could have a significant impact on reducing greenhouse gas (GHG) emissions, and the U.S. Department of Energy’s (DOE) Building Technologies Office estimates that reaching their goal of a 50% reduction in building energy use would cut U.S. energy-related GHG emissions by about 20% relative to 2010 [136]. Furthermore, the U.S. National Science and Technology Council has noted that “for the foreseeable future, the greatest national energy saving potential lies with improvements to existing buildings” [89], and the DOE’s 2015 Quadrennial Technology Review notes that it is “essential to reduce energy consumption in buildings in order to meet national energy and environmental challenges” [135]. With the goal of improving existing buildings to reduce energy consumption, the important first question is “which improvements?” Many reports identify that in order to reduce energy in buildings we must first understand how it is being used, and that starts with metering key points in buildings [90, 91, 134]. In a well instrumented building, every electrical circuit would be metered, including all large HVAC equipment, all rooms would report occupancy count, every plumbing fixture would track water usage, the state of all lights would be collected, and the ambient temperature, humidity, and air quality throughout the building would be known.

Building monitoring and sensing is typically accomplished through a Building Management System (BMS) which controls the building’s HVAC and other core systems. However, not all buildings have these systems [94], and they overwhelmingly include limited sense points, particularly with respect to energy consumption. Upgrading a BMS is often expensive or infeasible, given the complexity of installing new sensors and wiring them to the BMS control unit. An alternative sensing apparatus is required that can be easily added to existing

buildings with minimal invasiveness while providing data streams about a building that can be used to improve operational efficiency.

Many sensors that assist with this goal are wall- or battery-powered. Wall-powered sensors are constrained in their deployment as finding outlets can be challenging and running AC mains to them is expensive. Battery-powered nodes can be deployed more easily, but are either compact ($<15\text{ cm}^3$) with short lifetimes ($\approx 12\text{ months}$) due to limited battery capacity, or are large and obtrusive ($>150\text{ cm}^3$) with much longer lifetimes ($>2\text{ years}$) due to being able to accommodate a larger battery.

Buildings present a rich application setting for a new tier of sensing and allow us to ground our research in a practical context with real constraints. There is a clear societal and environmental need for reducing their energy footprint, and in fact the Energy Independence and Security Act of 2007 in the U.S. sets a goal for all commercial buildings to be net-zero energy by 2050. Buildings also have a large impact on building occupants, as people spend approximately 90% of their time indoors [138]. Improving the ability of buildings to operate more effectively and ensure occupant comfort is another important aspect of future building technologies. However, many buildings do not have the existing systems or monitoring capabilities to realize these improvements. Further, retrofitting buildings with today's technology solutions is often difficult, expensive, and aesthetically problematic. A shift in the technological substrate that enables us to cost-effectively address these issues can unlock significant potential in the buildings application space.

1.3 Energy-Harvesting Sensors for Buildings

Sensing in buildings is often accomplished with wired sensors added when the building was constructed or renovated. Wired sensors are highly reliable and typically cost effective when incorporated into an existing construction project. However, adding new hardwired sensors independent of building upgrades is often infeasible. Battery powered sensors remove this installation barrier, but add additional, and often unwanted, maintenance requirements for battery charging or replacement to the overall building management. This dissertation argues that the core technology shift for better sensors that see deployments at scale, even in existing, unrenovated buildings, is changing from wired, expensive to install or battery-based, fixed-lifetime sensors to energy-harvesting sensors with perpetual lifetimes. This change, however, raises many new technical challenges for how we design, program, and utilize these sensors.

Energy-harvesting, from indoor photovoltaics, vibrations, thermal gradients, magnetic fields, RF waves, and other sources, allows embedded computing systems to scavenge their own energy and greatly extend their lifetimes. The tradeoff is these sources are often highly

variable, both spatially and temporally, and raise two main challenges for system design: **energy intermittency** (energy is not always available) and **energy unpredictability** (when the energy will be available is unknown). Under battery power, devices schedule their computation, sensing, and communication tasks according to networking protocols and application goals with the assumption that the battery can provide energy when needed to complete a task. With energy-harvesting, energy may not be available when existing protocols and applications require a certain task. Moreover, as devices scale down in size, so too does their energy storage which, today, dominates node volume. Outdoors, plentiful sunlight and a lack of size and aesthetic concerns facilitate energy-harvesting for networked devices, but indoors nodes must be smaller (to be aesthetically compatible with the building) and operate on orders of magnitude lower power budgets (as constrained by energy availability and device size).

Often, outdoor energy-harvesting nodes have used batteries in conjunction with solar panels to mitigate some of the effects of energy intermittency and unpredictability by providing an energy reserve that can be used at night or on cloudy days. Using this technique indoors, however, is challenging due to the smaller form-factor of the nodes and the cycle-life property of batteries in which they have a limited number of cycles before their capacity diminishes. With a large battery, a single cycle can support a sensor node for many days or months. With the small batteries possible in small, indoor nodes, many cycles will be required, wearing them out much faster. Therefore, using non-battery-based techniques for managing intermittency and unpredictability are advantageous.

This dissertation explores several sensing techniques that enable building-scale sensing while making energy-harvesting devices viable to build the toolkit shown in Figure 1.1. The first of these techniques is to optimize existing hardware designs and software stacks for *low energy*—and not just low-power—operation. This means that the device is not just on average drawing very little power, but also that it must complete all atomic operations—wakeup, sense, and send—while consuming only the minute amounts of energy that can be collected and stored by the node in its tiny energy buffer. This requires substantial changes to the power supply and startup behavior of system software.

The next technique takes a non-traditional approach to addressing the intermittency and unpredictability concerns with energy-harvesting. Instead of attempting to mask the effects of energy-harvesting, this technique instead embraces the intermittency and uses the rate at which the node is able to harvest energy as a way to monitor the energy consumption of building loads. This approach works as long as energy harvested by the sensor is proportional to the energy consumed by the load, which is often the case for loads that are located inside of a building.

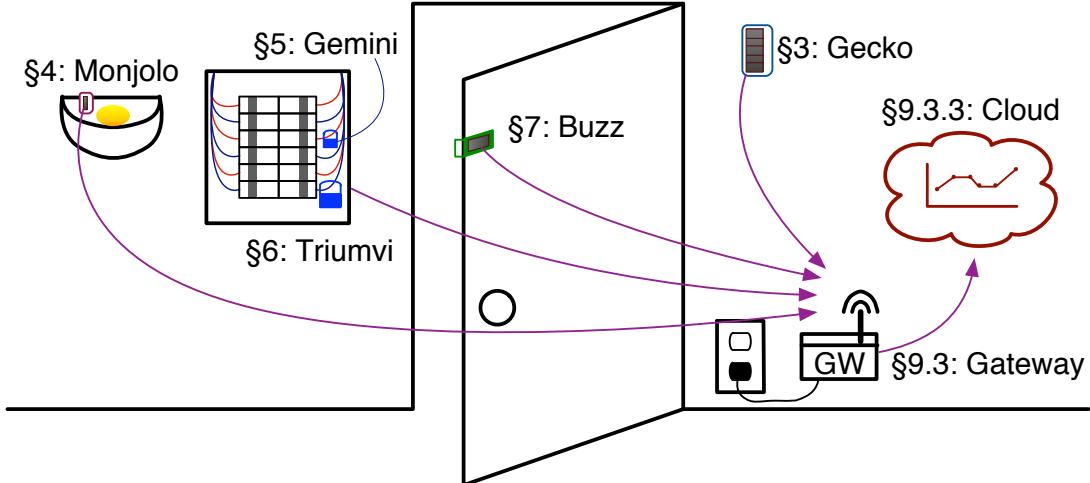


Figure 1.1: Energy-harvesting sensors for building metering and monitoring. Sensors monitor key points in buildings, including lighting, electricity consumption, door state, and ambient conditions. They leverage a shared gateway platform and cloud backend for data collection and analysis. This building-centric toolkit leverages a variety of techniques to enable perpetual sensors that can help improve buildings.

Another technique this dissertation explores is co-optimizing harvesting and sensing for a very specific sensor type. By sacrificing generality, the sensing capability and accuracy of an energy-harvesting node can be greatly improved by exploiting domain-specific features of a particular sensor type. For instance, water use in a building is often bursty, and can be measured by detecting on/off events rather than cumulative flow. AC current is typically periodic (at 50 or 60 Hz) and measuring a single AC cycle can be used to estimate the power draw of the load. Both of these examples are sensor specific, but with co-optimization an energy-harvesting sensor can successfully leverage the physical properties to be effective.

The final technique is to decouple sensing from harvesting. Often, the phenomenon to be measured is not related to a good source of harvestable energy, and existing energy-harvesting systems overprovision their power supply to be successful. However, by relaxing the coupling between the energy source and the process, parameter, or event being monitored, more sense points within a building are possible with small form-factor devices.

Following these techniques—optimizing for low energy operation, embracing energy intermittency, exploiting characteristics of the underlying signal, and decoupling harvesting from sensing—we demonstrate that a number of building-centric energy-harvesting sensors are feasible, that they yield useful data, and they open up new problems. This dissertation also presents implementations of these design techniques, deploys them at modest scale in real settings, and demonstrates how these sensors work in a building context.

1.4 Thesis Statement

Enabling the trillion sensor vision will require energy autonomous operation for the majority of the sensors. An architecture that embraces and addresses the energy intermittency and unpredictability resulting from energy-harvesting operation, by minimizing idle power and active energy, embracing anycast and broadcast communications, and strategically deferring energy-intensive operations enables small, unobtrusive, and inexpensive devices that can address practical sensing problems and operate perpetually.

1.5 Contributions of this Dissertation

This dissertation presents novel techniques for realizing energy-harvesting sensors that target real-world sensing needs inside of buildings and makes the following contributions:

First is an analysis of the energy availability indoors for harvesting. This frames the constraints and opportunities for the energy-harvesting devices presented here, and for other indoor harvesting and sensing systems. This work appeared at IPSN'12 and was done in collaboration with Lohit Yerva, Apoorva Bansal, Thomas Schmid, and Prabal Dutta [153].

Using the constraints determined by experimental analysis of available energy, we develop energy-harvesting sensors that are suited to the indoor environment. These devices lead to a new sensing paradigm unique to energy-harvesting that embraces, rather than attempts to mask, the intermittency due to harvesting. This sensing paradigm was presented at SenSys'13 and developed in collaboration with Samuel DeBruin and Prabal Dutta [21], and a version for hot water pipes was presented at EnsSys'14 and done in collaboration with Branden Ghena and Prabal Dutta [14]. This paradigm creates a data stream from a sensor's ability to harvest rather than a discrete sensor. When the harvesting rate is proportional to a physical phenomenon, timing information encoded in this output stream can be used as a proxy to understand the underlying phenomena. Distilling the sensor to a power supply and radio enables reductions in size and cost for the device, and transfers the burden of signal interpretation to the nearby gateways or remote cloud servers. This indirect measurement technique, however, can suffer from accuracy issues when the transfer function from energy source to packet transmission rate is unknown. To address this issue, we present a technique using a known global energy aggregate to disaggregate and attribute energy with the help of the energy-harvesting sensors. This method was presented at e-Energy'14 and was joint work with Meghan Clark and Prabal Dutta [16].

While sensing energy in a building, and particularly electricity consumption, a drawback to using energy-harvesting devices is their lossy nature due to limited energy availability. We

address this concern by carefully focusing on a particular sensor type and its relationship to the underlying properties of the sensed signal. We explore this with an AC power meter that uses a time synchronization scheme that enables tight synchronization between two signals (AC voltage and current) to create a true power meter with high accuracy while leveraging all of the benefits of energy-harvesting operation. This meter was presented at RTSS’14 and was joint work with Prabal Dutta [13].

We then focus on deployability, and show how an energy-harvesting AC meter can be optimized for deployment—and how energy-harvesting is critical to the meter’s ease-of-use. Energy-harvesting enables very quick install times (approximately 30 seconds per circuit), which reduces the overall cost and significantly lowers the barrier for getting submetering data. This metering system, joint work with Ye-Sheng Kuo and Prabal Dutta, has been submitted for peer review.

To complement metering energy sources, we also develop a toolkit of energy-harvesting sensors for event detection and monitoring in buildings. We then analyze these devices for commonalities, and find that a common architectural pattern emerges. This architecture was presented at BuildSys’14 in collaboration with Prabal Dutta [12]. This architecture simplifies creating new sensors by encouraging layering, reusable components, and clean abstractions—leading to an ecosystem of modular, interchangeable components.

We also identify two networking options for energy-harvesting devices: single-hop communication with an always-on receiver and as edge nodes in a wireless mesh network. In both cases the key insight is to offload some of the energy costs related to communication to devices with more predictable (although not necessarily plentiful) access to power.

We test these devices in actual deployments to understand how they operate in real-world settings with respect to deployability and performance. We also describe the gateway and networking architecture that make these deployments scalable.

Finally, we identify a variety of open problems and research opportunities that this work brings into focus, including energy-harvesting sensors with multiple tiers of energy storage, software layers for masking intermittent sensors, and frameworks for repurposing deployed devices.

CHAPTER 2

Background and Related Work

Energy-harvesting sensing for indoor environments builds upon a large corpus of work in the ubiquitous sensing, wireless mesh networking, energy-harvesting, and energy metering domains. Broadly, these subfields all aim to push computing beyond datacenters, desks, and pockets and into the ambient environment. This is driven by the availability of smaller, lower power circuits, microprocessors, and wireless radios that make it possible to decouple computers from system administrators and users and deploy them at an unprecedented scale. The motivations for this scale are similarly broad, spanning from human-focused applications, to real-time monitored spaces, to longer-lasting devices, to data-driven decision making. These technology drivers and application motivations form the foundation for the new sensors presented in this dissertation.

2.1 Pervasive and Ubiquitous Sensing

Mark Weiser coined the term “Ubiquitous Computing” in 1988 to describe a computing platform in which computers, sensors, and displays are embedded in the environment until they become effectively invisible [143]. This vision imagines seamless interaction between people, the environment, and embedded computation to enable new applications that are otherwise impossible, including person and device localization, virtual surface interaction, automated virtual and physical authorization, and pervasive real-time monitoring. Much of Weiser’s and his colleagues’ focus at PARC centered around the human interface aspect of the vision, including developing new interfaces for mobile computers, collaborative interfaces for shared computers, and person-specific interfaces that can travel with the user. But Weiser also noted that “the technology required for ubiquitous computing comes in three parts: cheap, low-power computers that include equally convenient displays, a network that ties them all together, and software systems implementing ubiquitous applications.” [143] This short list of requirements actually encodes many complex and challenging problems

that are even more challenging at the envisioned scale of ubiquitous computing. The work in this dissertation aims to address a specific portion of the listed requirements, namely the low-power computers that can be easily embedded in the environment at scale, by exploring small, self-powered sensor platforms. These devices can provide many of the input data streams that the ubiquitous computing applications require.

2.2 Wireless Sensor Networks

One aspect of the ubiquitous computing vision that has been explored in depth are wireless sensor networks (WSNs), or spatially distributed sensor nodes that work together to monitor or control a space [19, 42, 80]. Technology trends towards smaller, lower power, and less costly processors enabled small, battery powered nodes that can be deployed widely and expected to run autonomously for months to years at a time [15, 19]. With similar improvements in low power wireless radios, these nodes exploit the density of their deployments to transmit data through their neighbors and back to a gateway node, creating a mesh network covering the deployment [45, 53, 105, 145, 147]. Deployments of WSNs have been used to observe Redwood trees in California [132], sense vibrations on the Golden Gate Bridge [69], track Zebras in Kenya [64], localize gunshots [115], monitor datacenters [74], and many other sensing applications.

This new opportunity for small computers to do long-term unattended monitoring opened many research avenues. First, the community needed new hardware platforms that optimized for low power operation and could last for months or years on a small battery. Next, new operating systems and hardware abstractions [50], like TinyOS [73] and Contiki [28] balanced tradeoffs between resource use (in terms of RAM and code space), expressiveness, ease of programming, and energy. WSNs also needed sensors that could observe environmental phenomena, but at power budgets that attempt to maximize system lifetime when using a battery [33, 141, 151]. Once nodes could sense, they also needed to send, which led to new radio technologies [48], low power MAC layers [1], and new networking and routing algorithms [4, 26, 34, 45, 53, 146].

2.2.1 Hardware Platforms

Essential to the success of wireless sensor networks are the hardware platforms that balance power limitations with performance requirements and enable higher-level operating system, networking, routing, and application research [52]. Some early sensor motes include the BTnode [9] and the TelosB [106]. These two nodes represent very different design points

for sensor motes that illustrate some of the inherit tradeoffs for WSN platforms. BTnode provides a minimal hardware feature set, focusing only on a microcontroller (MCU) and radio. It does, however, include several I/O expansion ports for interfacing with external sensor modules, user interfaces, or additional wireless radios. By minimizing the MCU GPIO pins that are used internally, BTnode is able to provide considerable flexibility for adding external modules. BTnode also uses a Bluetooth radio for its wireless connectivity. This primary advantage that Bluetooth provides is that it is interoperable with many mobile phones. These phones can then provide an interface for users of the node, as well as a backhaul opportunity for data collection. In contrast, the TelosB takes a very different approach. Its hardware is designed to be fully turnkey, and includes not just an MCU and radio, but also a sensor suite and programming header. This allows the node to be fully standalone without requiring additional modules or an external programming header. The downside is that these additional components add considerable cost, and consume much of the I/O interface to the MCU, limiting the potential for expansion and customization. The TelosB uses an IEEE 802.15.4 radio [54] for its wireless interface. This radio provides a very open platform for wireless networking and research, but typically requires custom gateways and is difficult to interoperate with other existing systems.

While there are many points on this design space, BTnode and TelosB illustrate many of the primary tradeoffs. While BTnode is very flexible and supports many customization options, the lack of built-in sensors or other hardware makes it difficult to rapidly prototype new ideas and perform feasibility tests. TelosB on the other hand, can be tested and deployed as is, but its form-factor and minimal expansion options can make it difficult to package for deployment. As for the radios, Bluetooth provides more existing connectivity options, but at the cost of significantly higher power draw than 802.15.4 radios. 802.15.4 enables nodes with longer lifetimes, but requires more deployment time and cost as there are not existing networks the nodes can leverage for connectivity.

This range of tradeoffs, plus hardware advancements, has lead to many other hardware platforms, including the iMote2 [87], and the Epic mote platform [32]. The iMote2 focused on computational performance by increasing the RAM, flash, and CPU speed of the node to levels an order of magnitude greater than comparable systems. The cost of this is its power draw, where both its active and sleep current draw are an order of magnitude greater as well. Epic optimized for ease of use across a range of use cases, including research, prototyping, and deployment. By providing only a core set of hardware features (MCU, radio, and non-volatile storage), independent power domains, access to a wide range of I/O signals, and an extensive development board, researchers can easily experiment and measure the board, hackers can use the development board to quickly try new ideas, and scientists

can deploy the same module they used for testing as a solder-on addition to their custom sensor board. However, because Epic only natively includes a few core components, users of the module often have to recreate common circuits manually, for example a programming interface, power supply, debugging LEDs, and antennas, adding additional time and expense.

Newer designs, such as Firestorm [3] and OpenMote [133], continue to provide experimental platforms for research, but with updated hardware capabilities that support new software systems. Again, the presence of multiple new platforms demonstrates the continued tradeoffs in form-factor, radio interoperability, software support, power draw, and deployability. The common theme, however, is that these hardware platforms have been essential for enabling the research community to develop new wireless sensor network protocols, operating systems, and applications, while being able to iterate quickly and test in actual deployments.

2.2.2 Wireless Mesh Networks

Wireless sensor networks are commonly battery powered and aim to transmit their data back wirelessly. This makes them applicable in scenarios where wired power and networking would be infeasible. To reduce the amount of energy spent transmitting wireless packets, the nodes use short-range links (1-20 m) to reduce the transmit power required. To enable a WSN to monitor a larger space, the networks typically organize into a mesh network in which nodes sense and send, as well as route packets for other nodes.

To make mesh networking power efficient and to enable bi-directional communication, many schemes have been presented for duty-cycled wireless communication. These schemes attempt to minimize the time the wireless radio on the node must be active (either transmitting or receiving) to reduce the power draw of the node, while still ensuring two (or more) nodes can effectively communicate. At a high level, these medium access control (MAC) schemes can be grouped into two categories based on whether the node trying to transmit initiates the transfer, or the receiving node initiates the transfer by announcing it is ready to receive. The first group is known as *sender-initiated*, and the second is known as *receiver-initiated*.

Two common examples of sender-initiated MAC layers are LPL [105] and X-MAC [10]. LPL requires the transmitter of a packet to transmit a long preamble before transmitting the actual data. Potential receivers then periodically wake up, sample the channel, and then, based on the energy on the wireless channel, decide if there is a pending transmission they should stay awake for to receive. X-MAC extends this primitive in two ways. First, it encodes the destination address in the preamble, so all non-recipient nodes that overhear the transmission can return to sleep mode before receiving the actual packet. Second, the

preamble is transmitted in a strobe fashion, allowing the receiver to interrupt the preamble transmission to reduce the time each node must be active. These sender-initiated schemes are advantageous because they are simple to implement, asynchronous, do not require a coordinated sense of time, and require very little state to be shared between nodes (essentially they just have to know the preamble length and channel sample frequency). However, using preambles and periodic sampling both increases the amount of time the transmitter must be active, and requires that the receiver periodically wake up and listen, even if there is no incoming packet. Additionally, if during a receiver wakeup it detects energy on the channel from interference and not a valid packet preamble, it will stay awake listening, even if another node is not intending to transmit.

In the second group, RI-MAC [124] and A-MAC [30] are examples of receiver-initiated MAC layers. In RI-MAC, nodes periodically announce they are able to receive by sending a data beacon. When a node wishes to transmit, it activates its radio in receive mode, and waits for a beacon from its intended destination. When it hears the correct beacon, it transmits the data packet to the receiver. This removes the need for any long preamble to be transmitted, reducing contention on the wireless channel and enabling more data traffic while remaining energy efficient. A-MAC builds on this concept of a receiver-initiated MAC layer by having the sender respond to a beacon or probe from the receiver not with the data frame, but instead with a hardware ACK. The sender then transmits the data frame a short time later. This allows the receiver, which has to consistently probe for receptions, to reliably go back to sleep when there is no incoming data because unrelated channel energy or other contention is unlikely to mimic an ACK frame. This minimizes wasted energy due to unnecessary wake-ups.

These MAC layers, and many others [24, 36, 79, 86, 139, 152], enable a node to aggressively duty-cycle its radio while still permitting link-layer communication. To enable an actual network of low-power nodes capable of running real applications, additional protocols are needed that support neighbor discovery, message scheduling, and packet routing within the network. These too must be geared for low-power operation to support the battery-limited power budgets of the nodes in the network. Examples of these higher-level protocols include Time-Slotted Channel Hopping (TSCH) [55], Dozer [11], 6LoWPAN [85], IPv6 for low-power nodes [53], the Collection Tree Protocol [45], RPL routing [146], and Thread [131]. These techniques, however, rely on the predictability of batteries as an energy store. Batteries provide an abstraction by which a node can wakeup and transmit at any time, as long as the average number of wakeups and transmissions is low. This allows wakeups to be precisely and asynchronously scheduled to enable low-power mesh networks.

2.3 Energy-Harvesting Sensors

Newer, low power microcontrollers, coupled with low power circuits and sensors, operating systems that optimize for cold boot, sleeping, and wakeup, and aggressively duty-cycled networking stacks, enable sensors that can be deployed maintenance free for more than a year. However, this lifetime is very dependent on the size of the battery, and to ensure a sufficient lifetime, the batteries are often over-provisioned. This increases the size of the node while not eliminating a finite lifetime.

To avoid the fixed lifetimes, size penalties, and costs of batteries, some sensors have moved to energy-harvesting power supplies in which energy to run the node is collected at runtime rather than being entirely provisioned during node installation time. Energy-harvesting operation can enable nodes to extend their lifetimes beyond what would be possible with a battery.

2.3.1 Outdoor Energy-Harvesting

Many energy-harvesting sensor systems have targeted outdoor deployments where solar energy is plentiful during the day. For example, Trio [31] is an outdoor sensor network testbed based on the Prometheus energy-harvesting node [61]. To achieve continuous sustainability at approximately a 30% duty-cycle, the node uses a 30 cm² solar panel, and the node with sensors and a supercapacitor is substantially larger. Heliomote [67, 111] uses a 49 cm² solar panel with an 1800 mAh rechargeable battery and also targets outdoor deployments. The node's duty-cycle is adaptive based on current harvesting conditions but is estimated to be active about 20% of the time. The HydroWatch [127] weather node measures hydrological cycles in forest watersheds and uses a 34 cm² solar panel with a 6000 mWh battery. The sensor node is well optimized with a very low duty-cycle of about 1%. Because of the nature of the deployment in a dense forest, the solar panel provided minimal harvesting capability (typically less than 50 mWh per day), and therefore the optimized sensor node was required for the deployment to be sustainable.

The Trio and Heliomote examples are able to support relatively high duty-cycles because they are largely unconstrained in size and benefit greatly from the energy density of solar radiation. Outdoors, the practical and aesthetic limitations related to size are minimal, so panels, nodes, and energy storage can all be over-provisioned with minimally adverse affects. Even with these advantages and with at least some access to solar energy, the Hydrowatch project found that a majority of its nodes in the network were unable to operate sustainably, despite the optimized sensor node. As energy-harvesting sensor deployments move indoors, many of these advantages no longer apply, and as Chapter 3 will show, the energy availability

from solar radiation is significantly greater than from artificial indoor lighting.

2.3.2 Indoor Energy-Harvesting

Indoor sensing applications can also benefit from nodes with perpetual operation but size and energy constraints are much more severe than in the outdoor arena. As such, various projects exist that include indoor harvesters or optimizations for those circuits that recognize the reduced harvesting potential indoors [6, 66, 110, 126, 140]. A harvester, however, is not a complete sensing system and other projects, including many of the systems in this dissertation, examine sensing systems that are based on energy-harvesting.

Hande et al. present an energy-harvesting wireless mesh router node [49] designed to operate in a hospital environment near fluorescent lights. To enable the node to be an effective router, it was designed to be always-on. To accomplish this, the node uses a large number of solar panels (484 cm^2 in total) and must be deployed in pairs as each node only achieves a 50% duty cycle. In contrast, by not trying to emulate battery powered mesh networks, the work in this dissertation avoids excessive solar panels.

DoubleDip [83] is a water flow monitoring system attached to pipes that uses an energy-harvesting thermoelectric generator (TEG) as both a power supply and water event detector. A sudden water flow event causes a temperature gradient to be present on the pipe and a resulting spike in the output voltage of the TEG. DoubleDip detects this spike and records a water event in the pipe. Harvesting from water pipes is not always a reliable source of energy, however, and in four of DoubleDip's six deployment scenarios, the node's operation consumed more energy than it was able to harvest. In contrast, this dissertation presents techniques for enabling systems to scale their operation to the available energy.

Trinity [149], another indoor energy-harvesting sensor, is a system for detecting and measuring airflow. Trinity uses one piezoelectric device to harvest and another, simpler piezoelectric device as a sensor to measure airflow. Trinity's runtime power draw and harvesting capability are closely aligned to provide operation similar to a battery powered node, but at a lower duty cycle. While this aids inter-node communication, it forces Trinity to be relatively large to provide sufficient power. By reducing the operating requirements for the sensor, the work in this dissertation is able to sense with much smaller devices.

The EnHANTS [46] project explores the challenges for a new tier of energy-neutral, self-reliant nodes that are designed to track objects indoors. Their tags are custom fabricated hardware in a flexible sticker design that can easily attach to everyday devices. To achieve this form factor, the nodes are fully custom designed, which makes them exciting future possibilities, but not complete nodes that are being tested or deployed today.

The cubic millimeter [95] project aims to incorporate an entire sensor node within one cubic millimeter of volume. This requires designing and fabricating custom chips for every layer of the sensor node stack. While this hardware platform is at least an order of magnitude smaller and lower energy than the systems presented in this dissertation, its custom nature means it is not commonly available yet. In the future, the techniques presented in this dissertation may be applicable to this new hardware class, enabling even less obtrusive sensors for buildings.

2.4 Building Monitoring

Given the issues concerning buildings described in Section 1.2, there are many related works that focus on building sensing technology to provide tools and methods for improving building energy use and understanding the state of buildings. In this section we describe a selection of sensing systems that are designed for buildings.

2.4.1 Ambient Sensing

There are a host of commercially available and research-level ambient sensors that monitor general conditions, like temperature [71], humidity [119], light [121], and air quality [18, 62] inside of a building. These devices use a range of wireless protocols (e.g. Zigbee, Bluetooth Low Energy, or Wi-Fi), support a range of interaction patterns (e.g. through a proprietary gateway, using the local Wi-Fi network, or directly with a smartphone), and offer a range of user interfaces (e.g. a cloud-powered website or a smartphone app). For a power source, they mainly use batteries or are AC mains powered. In small numbers, this approach works well as battery replacement is relatively infrequent (on the order of every six months to a year), and the devices do not outnumber available plugs. At larger scale, however, or in a commercial building these types of devices represent a much larger maintenance overhead.

EnOcean [40] offers a line of energy-harvesting ambient sensors that use a sub 1 GHz wireless radio and typically a solar panel for harvesting. These can monitor indoor temperature, humidity, light, window position, and occupancy. These sensors are easier to maintain than their battery powered counterparts, but do require a custom gateway nearby and, while they are commercially available, they currently exist as evaluation modules instead of consumer products.

	Meter Type	Current Sensing	Voltage Sensing	Wireless	Harvests Energy	Runtime Calibration	True Power	Standalone
Watts Up?.net [142]	Plug load	Shunt resistor	Direct	No	No	No	Yes	Yes
WeMo Insight [7], ACme [60]	Plug load	Shunt resistor	Direct	Yes	No	No	Yes	Yes
TED [129]	Entire panel	Split-core CT	Direct	No	No	No	Yes	No
PowerCost [107]	Entire home	Visual	N/A	Yes	No	No	Yes	No
SEM3 [117]	Circuit level	Split-core CT	Direct	No	No	No	Yes	Yes
Magnetometer [100]	Circuit level	Magnetometer	RMS	Yes	No	Yes	Yes	No
Virtualization [116]	Plug load/circuit	Current transformer	Virtualized	Yes	No	No	Yes	No
Stick-on [150]	Circuit level	Piezoelectromagnetic	None	Yes	Yes	Yes	No	No
EMF [112]	Plug load	Split-core CT	Direct	Yes	No	Yes	Yes	No
PowerBlade [22]	Plug load	Inductor	Direct	Yes	No	No	Yes	Yes
Pressac [108], PAN-10 [96]	Circuit level	Split-core CT	None	Yes	Yes	No	No	Yes
GREEN [118]	Circuit level	Giant magneto-resistive	None	Yes	No	Yes	No	Yes
PASEM [77]	Circuit level	Hall-effect	Capacitive	Yes	No	Yes	Yes	No

Table 2.1: AC metering related work.

2.4.2 Occupancy Detection

Tailoring HVAC systems for efficiency and occupant comfort requires knowing where people are in a building to optimally direct resources. Also, many responsive applications at a building scale require localizing occupants in the building. Systems for detecting occupancy are often based on thermopile arrays [8], which can detect people using heat signatures, or PIR motion sensors [78], which rely on detecting movement to infer occupancy. These sensors are often bulky and power inefficient or require direct installation into ceilings or walls, failing many of the size, maintenance, and installability requirements of building monitoring sensors.

2.4.3 Electricity Consumption

When trying to monitor and improve the energy efficiency of buildings, a common first step is to submeter loads to determine where energy is being consumed. Submetering at the building, circuit, or plug load are all possible. Because power meters can often power themselves from precisely what they are sensing, powering these devices is often not key concern. Instead, deployability becomes essential, as power meters that are expensive, not wireless, hard to install, inaccurate, inflexible, hard to calibrate, or difficult to configure impose a high barrier to actually getting them deployed in buildings. Table 2.1 compares power meters across a range of properties relevant to this metering domain. The remainder of this section describes the related work in specifically AC power metering and how these various categories affect the efficacy of various metering design points.

2.4.3.1 Plug-Load Meters

Plug-load meters [7, 60, 68, 75, 109, 120, 142, 144] are capable of providing high-fidelity, accurate power measurements of individual loads and appliances. By their very nature they

have access to AC mains power and therefore powering them is not an issue. However, size is still important as large and bulky devices are difficult and often undesirable to deploy. Recent efforts have been made to scale down the AC-DC power supply to enable power-strip friendly power meters [22, 23] that can enable pervasive energy metering at the plug load level in a building.

However, for loads that are hard to move (e.g. kitchen appliances) or built in (e.g. lighting), attaching a plug load meter is difficult or impossible. Other techniques are then required to get submetered data for these loads. Another common issue with plug load meters is that they are typically active, meaning that they draw power even when the metered load is off.

2.4.3.2 Whole-Building and Circuit Level Meters

An alternative to plug-load metering is whole-home or whole-building metering in which a single meter monitors the power draw of the entire building. Two examples of this type of device are the The Energy Detective (TED) [129], and Blue Line Innovations' PowerCost Monitor [107]. The TED uses a current transformer (CT) where a CT is clipped around each phase of the main electricity line running to the building. The CTs are wired to a controller, which taps into a spare breaker per phase to get access to the voltage signal and to power the controller. The measured power data is then transmitted over power line communication (PLC) or Ethernet to a gateway. The main drawbacks to this approach are that the installation requires power to be shutoff to install the voltage contacts and there must be spare breakers present.

The PowerCost meter takes a different approach, and leverages the existing home energy meter from the utility. The sensor attaches to the meter and wirelessly replays the home's energy readings to a gateway inside of the house. This makes it much easier to install, but the device is battery powered and needs periodic battery replacement.

When entire home or building is too course grained, meters can also be installed at the circuit level. This both enables metering loads that do not plug in, such as HVAC equipment or built-in lighting, and reduces the number of meters that must be deployed versus using plug load meters. Panel and circuit level meters [41, 70, 82] provide reasonable sensing insight while only having to be installed in panel boxes and not at every load. These, like the TED, require running wires to current transformers on every circuit, and direct access to the voltage phases through empty breakers. While these meters are well equipped to provide accurate and high-resolution data, they can be difficult to install, particularly in retrofit applications. Often, the installation cost of hiring an electrician and the downtime cost of disconnecting power during installation dominates the total cost of installing the meters. To

try to alleviate this, the next sections cover meters that attempt to balance ease-of-installation with submetering resolution by exploring non-contact approaches to AC electricity metering.

2.4.3.3 Non-Contact Meters

Realizing that contacting the AC line for power metering incurs high installation costs and limited deployment scenarios, many recent meters have explored various non-contact options. Broadly, they fall in to two groups. The first includes a series of “stick-on” meters that attach to the outside of circuit breaker (near the switch) and sense the electromagnetic field generated by the current flowing to the circuit. The second group uses a current transformer (CT), but does not require wiring the CT to a control box. This type of device clips around the wire running to the breaker and uses the CT to measure the current. Some of these devices use or are capable of energy-harvesting to further remove the need for additional infrastructure or batteries during installation.

Patel et al. designed an externally powered house-level stick-on power meter using magnetometers designed to be attached to the circuit breaker in the house [100]. The sensor measures only current, which is later multiplied by RMS voltage by a PC. A similar design, called GREEN [118], by Send et al. uses a giant magneto-resistive (GMR) magnetic sensor to measure the current flowing through an individual circuit breaker. Unlike the Patel device, it can be attached to individual breakers to meter individual circuits. Another stick-on meter, this time using a piezoelectromagnetic (PEM) device [150], furthers this idea of stick-on AC metering by using a sensor that can also act as an energy-harvester [97]. This enables the PEM device to be self-powered when attached to the breaker. These three stick-on designs all share a common drawback: because they do not have access to the voltage channel they cannot accurately calculate true power. For reactive and switching loads, which have non-unity power factors, this error can be significant.

A fourth stick-on meter, the PASEM device [77], uses a hall effect sensor to measure current, but also adds a capacitive coupling technique to sample the voltage channel. This essentially creates a rough capacitor between the sensor and the metal internal to the breaker, and allows the device to extract a voltage signal. When the breaker type and geometry is well known, and the sensor is placed correctly, this voltage sensing technique can be as accurate as 1% error. By obtaining the voltage signal, the PASEM device can calculate true power. It cannot, however, power itself, so the sensors on each breaker must be wired back to a power source.

All four of these meters are susceptible to interference and crosstalk from circuits adjacent to the one they are metering. This means all of these designs include substantial processing to mitigate this interference and go through a runtime calibration procedure to

detect the extent of this interference. This calibration procedure often requires toggling circuits or plugging in known loads to certain circuits. This calibration step significantly complicates the installation procedure. While the devices may be simple to physically install, calibration can significantly increase the total cost of using the stick-on sensors.

Two commercial solutions, the Pressac CT [108] and the Panoramic Power PAN-10 [96] are energy-harvesting current meters based on current transformers (CT). The current transformers clip on to a circuit, and use the CT to measure current and power the meters. This removes the need for runtime calibration while retaining an easy to deploy form-factor. These meters, like the “stick-on” PEM meter, lack access to the voltage channel, however, and cannot calculate true power or power factor.

Finally, an approach proposed by Schmid et al. decouples current and voltage sensing [116]. A CT measures current at one point, and a direct contact voltage sensor measures the voltage signal at another. The voltage channel is then parametrized and transmitted to the current sensor, which is time-synchronized with the voltage monitor and can synthesize its own copy of the voltage signal. The final step of computing power is done offline. This enables physically decoupling voltage and current measurements which reduces the impact of needing direct access to voltage. We expand on this concept in Chapter 5 and show how this decoupling can be a significant benefit for deployability.

CHAPTER 3

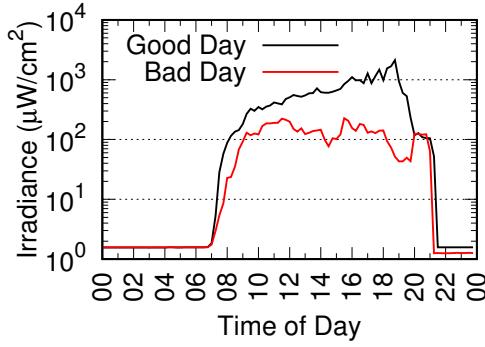
Gecko: Adding Energy-Harvesting to Wireless Sensor Networks

To bring energy-harvesting sensors indoors, the Gecko node seeks to use a harvestable indoor energy source to perpetually power a sensor for indoor ambient conditions monitoring. This would enable building operators, energy-efficiency advocates, and researchers to easily instrument and monitor a building without the need to hard-wire sensors or replace batteries. Bringing energy-harvesting sensors indoors requires both adapting the harvesting frontend to available energy sources and matching total system energy costs to the energy budgets provided by those sources. Section 3.1 covers some options for indoor energy-harvesting sources.

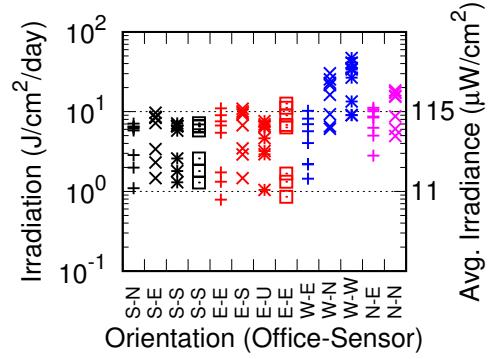
After considering several possible options, we evaluate using indoor lighting as the harvestable energy source for this sensor node. Lighting is an attractive source as it is plentiful in indoor spaces, and buildings are lit when occupied. Section 3.3 then explores options for storing the harvested energy so it can be used for the sensing operation. Section 3.4 combines the two primitives (harvesting and storage) into a complete design for a wireless energy-harvesting system. With a system capable of harvesting and wireless transmitting periodic measurements, Section 3.5 identifies techniques for adding some wireless sensor network primitives to the low energy operation of Gecko. Finally, Section 3.6 evaluates the performance of the node, and in particular looks at the sample rate of the Gecko platform.

3.1 Energy-Harvesting Options

Several options for indoor energy-harvesting exist, including vibration, thermal gradients, electromagnetic fields, airflow, ambient RF, and ambient light. Most, including vibration, thermal gradients, EM fields, and airflow occur in only very specific locations and are most useful when sensing specific pieces of equipment or particular locations. For example,



(a) Average Office Conditions



(b) Daily Irradiation

Figure 3.1: Energy availability from indoor lighting. (a) shows the average irradiance of four typical offices over the course of a partly sunny day (good) and a rainy day (bad). (b) shows daily irradiation across 13 sensors deployed in four offices over a one week period including sunny and rainy days. Although there is nearly a 50x difference in harvested energy across space and time, a roughly $10 \mu\text{W}/\text{cm}^2$ lower bound average radiance exists.

monitoring radiant heating with energy harvested from thermal gradients effectively couples the harvesting source and monitoring point.

For more general indoor monitoring, ambient RF and ambient light harvesting allow for more flexibility when deploying sensors. The available energy is significantly different however, with ambient RF only able to provide about $1 \mu\text{W}/\text{cm}^2$ [154] and ambient light able to provide two or more orders of magnitude more energy per the same area. Given the harvesting potential and ubiquity of light inside buildings, the Gecko design uses an indoor photovoltaic cell as its energy-harvesting frontend. The next section evaluates how much energy is available from this type of harvesting under different indoor conditions.

3.2 Indoor Lighting: Enough to Power Sensors

To measure the energy available from ambient light, measurements were taken the walls inside several offices in a standard office building. Figure 3.1a shows the average irradiance of four different offices (13 distinct locations) for a partly sunny (“good”) and rainy (“bad”) day. The data shows that even on a rainy day, average irradiance largely exceeds $100 \mu\text{W}/\text{cm}^2$ during daylight hours, suggesting many indoor locations are viable for leaf nodes. Figure 3.1b shows that daily irradiation can vary by more than an order of magnitude for a single location.¹ Most locations receive $1-10 \text{ J}/\text{cm}^2/\text{day}$ (or an average of $11-115 \mu\text{W}/\text{cm}^2$), which is enough to ensure frequent leaf activity. Figure 3.2 shows the instantaneous, rather than

¹Irradiation is accumulated irradiance per unit time, and it is measured in $\mu\text{W}/\text{cm}^2/\text{day}$.

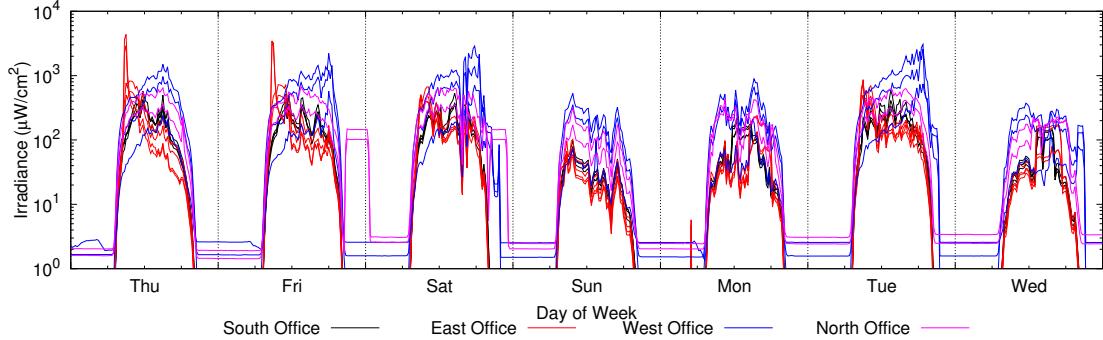


Figure 3.2: Irradiance (on a log-linear scale) over a typical week in four offices, oriented in four different directions. Each office is equipped with two to four typically wall-mounted sensors, each facing in different directions within the office, to capture different light environments. The horizontal lines correspond to the irradiance necessary for a leaf node to send a message every minute given the respective light source type. On most days, most leaf nodes have enough light to send a message once a minute, but on some days, some do not. This highlights the importance of node placement and solar cell sizing. It also illustrates the need to support a wide operating dynamic range.

average, irradiance of the underlying data over a one week period. The traces also show that during Friday and Saturday evenings, the North Office was illuminated, and during Wednesday evening, the West Office was illuminated. This suggest evening or nighttime human activity in those rooms. In all three cases of human activity, the light levels exceeded $100 \mu\text{W}/\text{cm}^2$.

The U.S. Department of Labor's regulation on minimum office illumination [137] is 30 foot-candles, which corresponds to about 320 lux. The regulations are lower for indoor corridors and hallways at 5 foot-candles, or about 53 lux. From our own experiments using a commercial lux meter and a TAOS TSL230BR [128] light to frequency converter we find that 50 lux of a fluorescent light corresponds to $18.6 \mu\text{W}/\text{cm}^2$, 100 lux to about $29.1 \mu\text{W}/\text{cm}^2$, and 320 lux to $74.9 \mu\text{W}/\text{cm}^2$.

3.3 Energy Storage Options

Once energy is harvested by the photovoltaic front-end, it must be stored locally to aggregate the harvested energy to a sufficient level to power sensor nodes, and to buffer the variability inherit in the harvesting source. Energy storage options for small, low power electronics typically consist of batteries, capacitors, and supercapacitors, each with their own trade-offs. Batteries typically provide the best energy density per volume. This gives small devices the best opportunity for maximum energy storage while minimizing volume. Capacitors

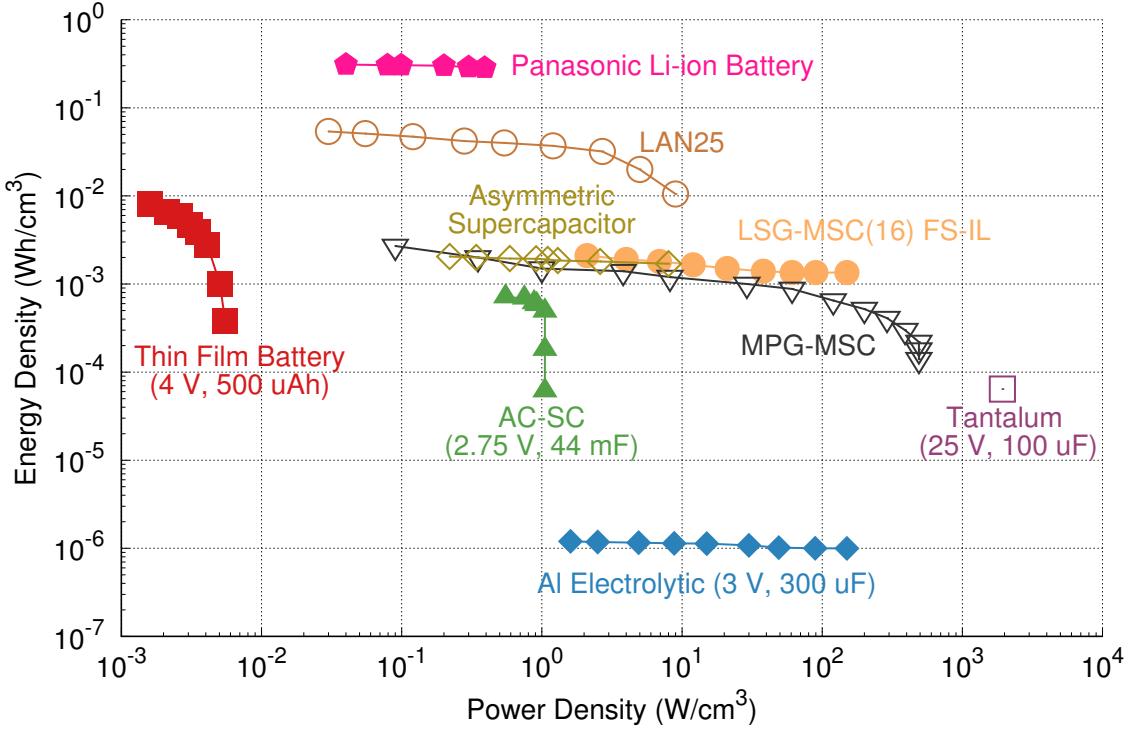


Figure 3.3: Ragone plot of energy storage technology from established technology to research prototypes. A thin film battery [101], a Panasonic lithium-ion battery [88], an aluminum electrolytic capacitor [38], a tantalum capacitor [25], an activated carbon supercapacitor [38], a reduced graphene film micro-supercapacitor (MPG-MSC) [148], a laserscribed graphene micro-supercapacitor with fumed silica nanopowder (LSG-MSC FS-IL) [37], a fiber-based micro-asymmetric supercapacitor (ASC) [63], and a lithium-ion hybrid capacitor (LAN25) [123] are compared. Batteries excel at energy storage and offer one to two orders of magnitude better energy density per volume than other options, but have limited discharge rates. Capacitors can supply power quickly, but are limited in their capacity. Commercially available supercapacitors sit in the middle, and research prototype supercapacitors are increasing both power density and energy density.

excel at power density and can supply the most power per volume of the storage options. This benefits low power systems that need to rapidly transition between power states and quickly access their energy store. Batteries have poor power density, and capacitors have poor energy density, however. Supercapacitors sit in between, and provide higher energy densities than capacitors and higher power densities than batteries. Figure 3.3 shows the trade-offs between these two features for a range of energy storage options.

Beyond their ability to store and discharge energy, other concerns also factor into choosing an energy storage technology. Batteries, for example, have a limited number of recharge cycles before their capacity significantly diminishes. For an energy-harvesting platform

where its normal option is harvesting and using energy, this limitation presents a significant challenge. Thin-film batteries, advantageous for their dimensions (e.g. STMicroelectronic’s EnFilm batteries are 0.22 mm thick [122]) are limited both in cycle life and energy capacity (the EnFilm battery is only 700 μ Ah). Supercapacitors and capacitors often exhibit high leakage currents, or the rate that energy stored by the device is lost due to the internal conductivity of the capacitor. For very low power designs and harvesting regimes (i.e. tens of μ A of harvesting current), this leakage can be significant and inhibits the device’s ability to accrue sufficient energy. Choosing a storage device for energy-harvesting system requires managing these various trade-offs.

For building an indoor ambient sensor, the Gecko design uses capacitors as its primary energy storage device. As a sensor that is continuously harvesting and using energy, the cycle limitation of batteries discourages their use. Additionally, since indoor lighting conditions can be highly variable, the energy income of the device is highly variable. In low lighting conditions the leakage of supercapacitors makes them undesirable options. Capacitors are also prone to leakage, but careful selection of the capacitor type can dramatically reduce this effect. Towards this, Gecko uses tantalum capacitors which exhibit low leakage, at the expense of increased cost. To compensate for the low energy density of capacitors, the design is able to operate on small bursts of energy and has little capacity for long-term storage of energy.

As the technology for laser-scribed graphene [37] and other new supercapacitors matures, this may enable new optimizations for energy storage in these devices that are not currently possible, and potentially increase the performance of the techniques Gecko uses.

3.4 Leaf Node Design: Revisiting Core System Components

To realize an energy-harvesting sensor node, core aspects of the system need to be revisited in light of the limited energy budget of the platform. Specifically, this includes the power supply, timekeeping, and the operation of the sensor.

3.4.1 Power Supply

The energy-harvesting power supply (Figure 3.4) integrates a solar panel, capacitors, over-voltage protection, various trip and enable triggers, a latching circuit, regulator, voltage sense circuits, and a rich interface to the power supply. We evaluate two different solar panels – an IXYS XOB17-4X3 (crystalline silicon) and a Sanyo AM-1437 (amorphous silicon) –

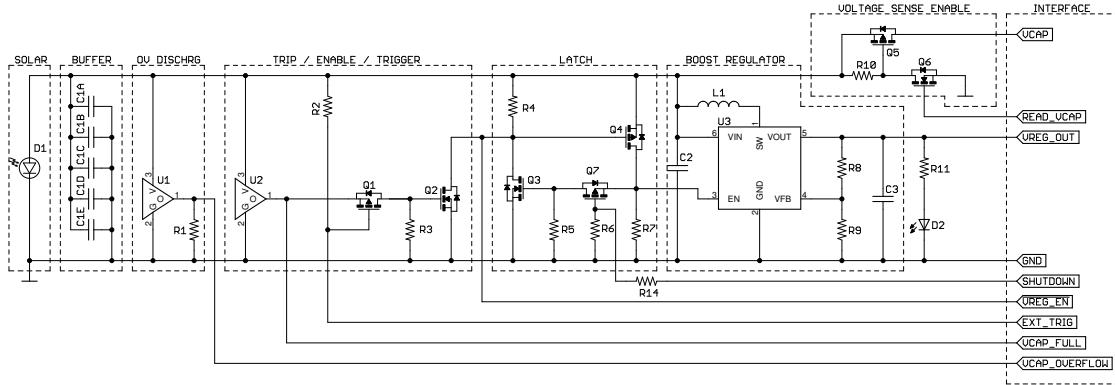


Figure 3.4: The power supply stores indoor solar currents of just a few microamps and provides periodic burst of regulated voltage. The power supply exports several monitoring and control signals to allow applications to tailor their operation to the key power supply events.

for their response to different light sources. We chose (two 100 μF) tantalum capacitors as the primary energy buffer due to their high energy density and low DC Equivalent Series Resistance (ESR) compared to ceramic capacitors. The activity and overvoltage triggers use low-current voltage supervisors from Panasonic and Microchip, which draw only 0.25–0.5 μA , respectively. We use the MCP1640 boost converter with true load disconnect to provide a stable voltage and switch power to the Epic Core. Finally, a FET-gated sense path allows the microcontroller to measure the capacitor voltage.

3.4.2 Timekeeping

A real-time clock (RTC) provides a periodic time trigger to instigate leaf activity. Our leaf node uses the NXP PCF2127A RTC [92], a temperature-compensated crystal oscillator which runs at 32.768 kHz. This device offers excellent time-keeping stability ($\pm 3 \text{ ppm}$), low current draw (0.65 μA) on backup battery, and flexible triggering options. Collectively, these features support efficient synchronization between leaf and branch nodes. The RTC also offers a battery-backed SRAM, which allows a node to maintain state across activity cycles without keeping the microcontroller powered on. The prototype’s batteries can be replaced with rechargeable super-capacitors to allow nearly infinite charge and discharge cycles at the cost of long term time keeping.

3.4.3 Sensor Operation

The hardware goes through three distinct operating stages:

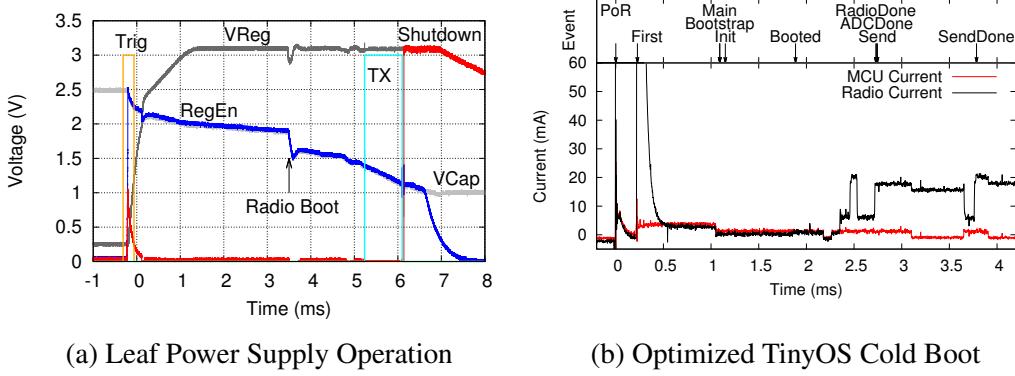


Figure 3.5: (a) Operation of the energy harvesting power system. The capacitor voltage is seen dropping from about 2.5 V to about 1 V. The boost regulator starts up and establishes a 3.0 V supply until the core requests shutdown after it finishes its radio transmission. *Radio Boot* refers to the execution of the first instruction, enable radio regulator. (b) Stages of an optimized TinyOS cold-boot overlaid with the processor and radio current. Our design achieves cold-boot to packet transmission in less than 4 ms using all of the optimizations discussed. The net effect is a reduction in TinyOS cold boot by a factor of $69\times$ in time and $14.6\times$ in energy.

Charge. From a fully-discharged state, the solar cells begin to charge the buffer capacitors. Voltage supervisor U2, a Panasonic MS1382SEL rated to trip at 2.5 V, monitors the capacitor voltage. Since the solar cell maximum power point occupies a fairly narrow range of a few hundred millivolts, as Figure 3.6a and Figure 3.6b show, we are able to use fixed voltage levels for the charge (and discharge) thresholds rather than needing or employing a costly maximum power point tracking (MPPT) scheme.

Startup. When the capacitors reach U2’s trip voltage (Figure 3.4), U2’s output goes high, initiating the active phase. Assuming EXT_TRIG is low when this happens,² P-FET Q1’s source goes high, turning Q1 on, and passing the U2’s high output to Q2’s gate, turning Q2 on. This drives Q4’s gate low, turning it on, and causing regulator U3’s VREG_EN to be driven high, initiating its startup. Instantly, the default *on* Q7 enables the bypass Q3. The bypass latch (Q4 keeps Q3 on, which keeps Q4 on) keeps the voltage regulator enabled even though U2 disables Q2, when the capacitor voltage eventually drops to 2.3 V (maximum U2’s hysteresis) as U3 goes into regulation. We use the MCP1640 boost regulator, which provides high conversion efficiency near our startup threshold voltage, and a slightly wider input voltage range as well for longer discharges, as Figure 3.6c shows. As U3 starts up and goes into regulation, it too discharges the buffer capacitor

²This could be done efficiently by, for example: (i) connecting EXT_TRIG to the output of the RTC’s periodic interrupt or (ii) connecting VCAP_FULL to an N-FET’s gate, EXT_TRIG to an N-FET’s drain, and GND to and N-FET’s source. We use option (i) and preprogram the RTC during hardware bringup.

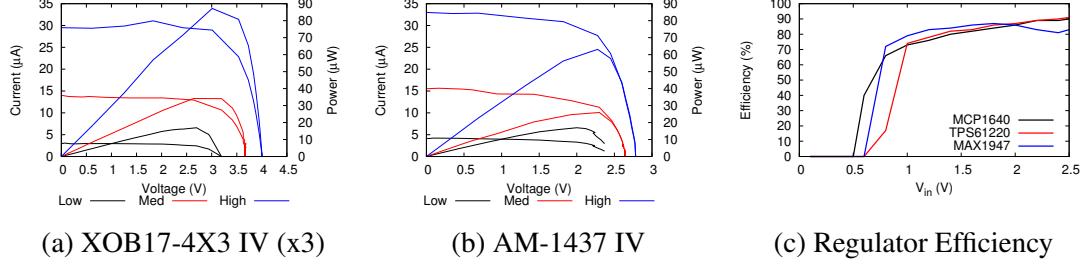


Figure 3.6: Solar cell current vs voltage, and power vs voltage, under low ($12.7 \mu\text{W}/\text{cm}^2$), medium ($53.8 \mu\text{W}/\text{cm}^2$), and high ($101.7 \mu\text{W}/\text{cm}^2$) indoor lighting levels. Although solar cell output current varies by an order of magnitude, the maximum power point varies within a small range of approximately 2.6 V to 3.0 V for (a) and 2.1 V to 2.3 V for (b), or about (10-15%). This lets us use fixed-threshold voltage trip switches for charge/discharge operation under typical indoor lighting conditions. (c) MCP1640 was chosen due to its efficiency over a large input voltage range, and because it has output disconnect and low leakage current.

Active. After startup, the processor boots, configures the radio, takes sensor readings, and transmits/receives packets. Once done, the processor drives `SHUTDOWN` high, which disables the regulator and disconnects power to the Epic Core, saving unused energy. Figure 3.5a details the operation of the startup and active stages.

3.5 Optimizing for Low Energy Operation

Low-energy and low-power leaf nodes need efficient and optimized system software support, and tight hardware-software interaction. Low-power operation begins in the first few instructions the processor executes and continues through system startup, communication, and sleep/shutdown. We detail some of the software operations that a leaf node employs to maximize energy efficiency.

3.5.1 Parallelized and Optimized Software Startup

Typical low power wireless sensor nodes expect to power on from a cold boot a single time, and spend most of their time in a low power sleep mode in order to prolong their battery life. This means that the energy cost of that initial cold boot is insignificant versus the energy cost of sleep mode over the expected lifetime of the node. However, in the Gecko operating model, where the node is continuously cold booting and spends most of its time completely powered off, the energy cost of startup becomes significant. We take careful steps to minimize the energy cost of startup to optimize the energy profile of the node.

The default TinyOS [73] distribution requires 237 ms and consumes 2.174 mJ from power-on to `Radio.sendDone()`. Most of the time is spent waiting for the 32 kHz clock to start oscillating. Adding an external 32 kHz source reduces the boot time to 48 ms and 1.81 mJ. Much of the remaining startup costs are spent initializing memory. Setting the microcontroller clock speed to the maximum possible frequency (\sim 8 MHz in the case of the MSP430F1611) before initializing memory reduces the cold-boot to 18.6 ms and 0.61 mJ. This figure can be further reduced to achieve a cold-boot time of 3.4 ms and 149 μ J by removing several layers of the TinyOS radio stack, parallelizing radio startup and memory initialization, and using the bare radio interface to transmit a packet as shown in Figure 3.5b.

The Gecko software also employs concurrency to minimize latency during system initialization (contrary to the far more typical serialized system initializations common in existing systems). The radio initialization is started, for example, before the C runtime zeroes out data memory, and the two operations proceed in parallel. Similarly, the packet to be transmitted is pipelined into the radio’s transmit FIFO.

The result of these software optimizations is shown in Figure 3.5b. This figure shows a detailed timeline of a cold-boot to packet transmission. The labels indicate stages of the boot process. They are, from left to right, PoR: Power-on-Reset; First: first instruction; Main: call of `main()`; Bootstrap: after call to `platform_bootstrap()`; Init: after call of `PlatformInit.init()`; Booted: first instruction of `Boot.booted()`; RadioDone: Radio initialization done; ADCDone: ADC initialization done; Send: start of packet; SendDone: Return from send, ready to shut off. The corresponding latencies are: PoR to First: 0.17 ms, First to Main: 0.87 ms, Main to Bootstrap: 0.5 μ s, Bootstrap to Init: 58 μ s, Init to Booted: 436 μ s, Booted to RadioDone: 842 μ s, RadioDone to AdcDone: 8.1 μ s, AdcDone to Send: 9.1 μ s, Send to SendDone: 1.04 ms.

3.5.2 Synchronizing Energy-Harvesting Nodes

Another aspect of the system that must be optimized for Gecko’s low energy operation is synchronizing the Gecko nodes. Typical node-to-node synchronization is done with running timers set to expire at the same time on each node. When the timer fires on each node, they expect the other node is also awake and able to send or receive a wireless packet. With Gecko, however, because the node spends most of its time powered down, the core system does not have the ability to keep a running timer active to schedule its next wakeup. Instead, we look to other mechanisms for allowing these energy-harvesting nodes to synchronize. One such mechanism is to use an environmental event that both nodes can detect simultaneously. For example, if all of the lights in a room suddenly turn off, and the nodes in that room detect a

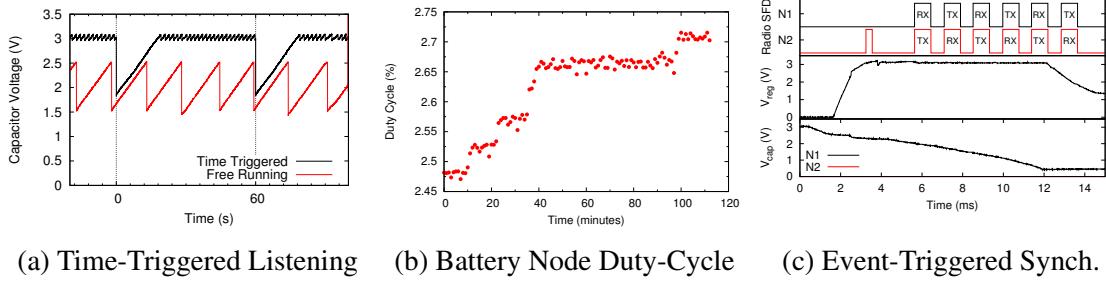


Figure 3.7: Achieving initial synchronization. Methods for achieving initial synchronization include both asynchronous neighbor discovery and synchronous event-triggered discovery. (a) A free-running leaf transmits a packet when it has sufficient energy. A triggered leaf transmits at a well-known interval. (b) Each point represents the duty cycle of each minute. Each step in the graph corresponds to the discovery of a leaf node. Even while synchronized with five leaf nodes, the branch node’s duty cycle only slightly increases from 2.45% to 2.75%. (c) A trace of synchronous event-triggered discovery is shown. Leaf (N1) and branch (N2) nodes both respond to a sudden change in light level (triggered by a zero-power light detector circuit), choose a short backoff, and transmit packets back and forth until the leaf depletes the energy in its storage capacitor.

sudden transition from light to dark, they can use that as a trigger to wakeup and exchange packets quickly before their energy reserves are depleted.

To explore the viability of basic event-triggered synchronization, we design a simple, zero-power, light-activated trigger switch. It consists of a solar cell connected in parallel with a burden resistor. The solar cell’s negative terminal is connected to an N-channel FET’s source (and leaf GND), and the positive terminal is connected to the FET’s gate. The FET’s drain is connected to the power supply’s external trigger or an interrupt line. This forces the leaf node to cold boot, and the branch node to wake up from sleep, in response to a sudden change in lighting. The nodes then exchange packets.

Figure 3.7c shows the operation of two nodes, a leaf (N1) and a branch (N2), in response to an external trigger. The leaf node transmits first, and then the branch node responds. The two nodes transmit packets back and forth until the leaf depletes its energy. This illustrates that for sufficiently “sharp” triggers, like light, acoustics, or sudden movement, it is possible for both leaf and branch nodes to synchronize.

3.5.3 Gecko to Low Power Mesh Communication

Energy-harvesting nodes like Gecko can also leverage other sensor network nodes that may be installed in a building to help with networking and data collection. Existing mesh networks of battery powered sensors can act as a backhaul for data generated by the energy-harvesting

nodes. The challenge is synchronizing the Gecko nodes with the heavily duty-cycled battery powered nodes. The battery powered nodes typically run a low power MAC protocol (e.g. LPL [105], TSCH [55], ContikiMAC [27], or others) to schedule transmissions within the network. However, due to the low energy operation of Gecko, it cannot participate in any of these schemes. We need a method that allows the two different classes of nodes to establish a shared wakeup slot while staying within the energy limitations of Gecko and duty-cycle requirements of the battery powered nodes.

To accomplish this, the Gecko nodes uses its onboard RTC to wake up on a fixed period, take a sensor reading, transmit a packet, and listen for inbound traffic. If a Gecko node does not have enough energy to perform these operations, it simply skips one (or more) activity cycles and simply resumes its activities when it has accrued enough energy. The key is that the nominal activity period is fixed and has low jitter and drift. Branch nodes learn of leaf node schedules using a low duty-cycle neighbor discovery protocol. Branch nodes subsequently track the nominal wakeup times of leaf nodes and listen briefly (i.e. 20 ms, searching over a variable-length guard time) for leaf transmissions, and to deliver any leaf-bound traffic.

We use a variant of the Disco [29] neighbor discovery protocol in which nodes transmit beacons on eventually overlapping schedules. In our case, only leaf nodes transmit and only branch nodes listen. The listen and transmission schedules are chosen such that the worst case discovery latency is about 50 min and the discovery burden is small for both.

Leaf nodes transmit beacons during a 5 ms window (“slot”) every 60 s. This translates to a nominal duty cycle of 83 ppm (or 0.0083%). Figure 3.7a shows a free-running leaf transmits packets whenever it has sufficient energy but that a time-triggered leaf transmits packets at multiples of 60 s, allowing a branch node to both employ a compatible neighbor discovery schedule and predict future transmission times.

Branch nodes listen for beacons during a 5 ms window (“slot”) every 245 ms (or 49 slots). This translates to a nominal duty cycle of 2.04% but inefficiencies and latencies in radio startup result in a higher duty cycle (2.5%), which translates to a roughly 1.5 mW power load due to discovery. The baseline discovery power is higher than ideal, but using newer radios that offer similar radio sensitivity at roughly 15% of the receive current, the baseline discovery power could be reduced to about 500 μ W. Given the nominal duty cycles of 0.0083% and 2.04%, the worst case discovery latency is $(5 \text{ ms}) / (0.0083\% \times 2.04\%)$, or about 50 min (2940 s) in the absence of communications failure, as Figure 3.7b shows.

3.6 A Viable Indoor Energy-Harvesting Sensor Node

By addressing the harvesting, energy storage, power supply, low power operation, communication, and synchronization concerns of an indoor energy-harvesting sensor, Gecko introduces a new type of leaf node that relies on scavenged energy for its main sense and transmit functionality. We microbenchmark Gecko to understand the energy costs of the main phases of its operation after the design has been optimized for energy-harvesting. This provides a method for analyzing other sensing scenarios and harvesting frontends for how effective they may be for a sensing system. We also measure the resulting sample rate of the node in an indoor environment. While energy autonomy is advantageous from a deployment standpoint, if the node is unable to sample at a sufficient rate the data stream may not be able to support the types of applications that call for embedded sensing.

3.6.1 Wakeup, Sense, and Transmit Energy Consumption

Because Gecko is a node designed to rapidly cold boot, sense, and transmit a packet, we evaluate it for how much energy is required for these steps. Figure 3.5b shows a trace of Gecko’s current draw during a boot, sample, and send cycle. The entire operation consumes 149 μJ . The microprocessor boot consumes 5 μJ to get to a state where the first instruction can run. The initialization phase, including setting up the c runtime, starting the radio initialization, and configuring the embedded operating system consumes 68 μJ . This gets Gecko to a state where it can begin its application-specific code. Sampling the sensors while continuing to wait for the radio to initialize consumes 21 μJ . Finally, transmitting the radio packet consumes 55 μJ .

As a note, this analysis only considers the energy consumed directly by the MCU, radio, and sensors. The total energy needed to power a sensor node will be greater due to inefficiencies in the harvester and power supply, as well as other losses in the system.

3.6.2 Sampling Rate in Indoor Environments

With a sensor that has a measurement rate proportional to its energy-harvesting source, the activation energy required must be low enough to sustain a sample rate that is sufficient for the intended applications. To evaluate this, we observe the rate Gecko is able to transmit readings in various lighting levels, across various light source types, and with multiple solar cell variants, with the results shown in Figure 3.8. Gecko is able to harvest and begin transmitting packets at irradiance levels of about 20 $\mu\text{W}/\text{cm}^2$, and can transmit once per minute at approximately 40 $\mu\text{W}/\text{cm}^2$. For ambient environmental monitoring, one sample

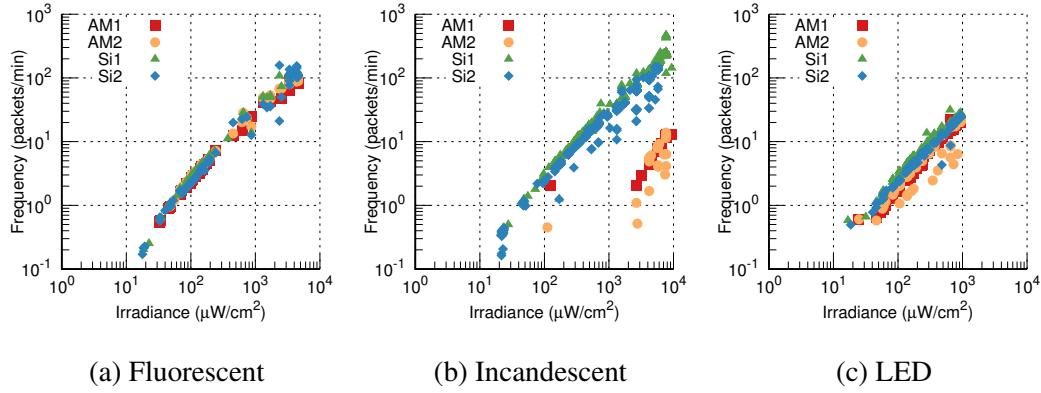


Figure 3.8: Gecko sample and transmission rate for a range of indoor lighting levels and multiple solar cell types. Gecko is able to work over three decades of indoor irradiance levels. At an irradiance level of about $40 \mu\text{W}/\text{cm}^2$, Gecko is able to transmit once per minute which is a standard sample rate for an indoor ambient monitoring sensor.

per minute is a standard measurement rate. In very bright conditions where the irradiance is 1000 $\mu\text{W}/\text{cm}^2$, which in buildings would be in rooms with sunlight, the transmission rate reaches about 30 packets per minute, or a 0.5 Hz sample rate.

3.7 Summary

By examining the energy available indoors to energy-harvesting nodes from lighting and creating a sensor node capable of periodic measurements powered by energy-harvesting, we show the feasibility of indoor monitoring with nodes that scavenge energy for themselves. We also show that often low power wireless software stacks ignore cold boot time and energy costs. Gecko optimizes for low energy operation and can operate on a limited energy-harvesting power budget, demonstrating one pillar of the thesis statement. We also analyze a networking scheme where a duty-cycle powered node can receive packets from energy-harvesting nodes. The Gecko sensor prototype also highlights an interesting property of energy-harvesting nodes: their sample rate is proportional to their harvesting source. We more fully explore this relationship in Chapter 4.

CHAPTER 4

Monjolo: Energy-Harvesting Energy Meters

Chapter 3 demonstrates the feasibility of powering networked sensor nodes with indoor lighting. It also illustrates the energy intermittency characteristic of energy-harvesting nodes: the node can only sample when its harvesting source is present, in Gecko’s case ambient light, and after it has accumulated enough energy. This intermittency makes employing existing low power sensors and software stacks challenging, but provides a unique opportunity for energy-harvesting devices in indoor sensing contexts.

A key observation for indoor sensing is that *the transfer and use of energy in a building often emits a side-channel of energy that can be harvested*. That is, the exact phenomena we wish to measure—energy use in a building—can often indirectly power a simple sensor. By designing a sensor that simply harvests from this side-channel until it has accrued enough energy to wakeup and transmit a packet, I found that the resulting rate of packets can be used to understand the underlying energy consumption source. Thus, the energy-harvesting power supply *is* the sensor. The duty cycle of the sensor is proportional to the energy being transferred or consumed. We call this mode of measurement the Monjolo principle, and demonstrate that by exploring the energy-harvesting design space to one extreme—distilling the sensor down to the harvester itself—we can create useful sensors that exploit, rather than attempt to mask, the intermittency of energy-harvesting. This paves the way for scaling down sensors to only a harvester and radio, which reduces size, complexity, and cost.

4.1 Monjolo Design Principles: Measurement Based on Harvesting Rate

To illustrate the Monjolo design, we begin with a simplified version of Monjolo that makes its basic operating principles clear. We then progressively relax simplifying assumptions and introduce real-world concerns to yield a design that works in practice.

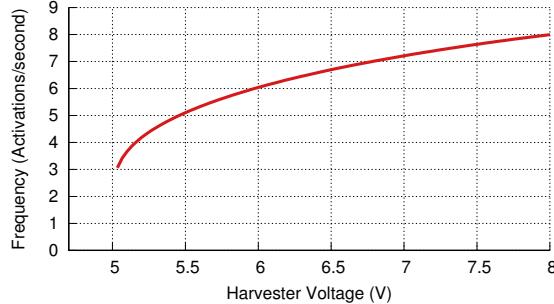


Figure 4.1: Theoretical response rate for Monjolo.

4.1.1 Basic Operating Principle

The core operating principle of the Monjolo meter is quite simple: energy is harvested into a reservoir at a rate proportional to the energy consumption of the underlying harvesting source. When the reservoir—in this case a capacitor—has accrued a certain amount of energy, the energy harvester activates a microcontroller and radio to quickly transmit a packet using this harvested energy. The microcontroller then ensures that a fixed quanta of energy is discharged from the reservoir over a fixed amount of time before allowing the reservoir to begin recharging. When the reservoir reaches the threshold again, the wakeup and transmission cycle repeats.

We claim that the interval between packets can be used to determine the underlying energy consumption, even if the packet transmission rate does not scale linearly with the consumption. As long as the packet transmission rate increases monotonically, and the packets are quickly and successfully received by the data aggregator, we can estimate load power from the inter-packet interval.

4.1.2 Monjolo Theoretical Model

To understand how the Monjolo principle may work in practice, we derive a model for how the activation rate of the Monjolo sensor relates to the underlying harvesting source. We start with the equation for charging a capacitor:

$$Q = CV_b \left(1 - e^{-\frac{t}{RC}} \right) \quad (4.1)$$

And then solve for t to get how long it should take to charge the capacitor that acts as the energy storage for Monjolo.

$$t = -RC \cdot \ln \left(1 - \frac{Q}{CV_b} \right) \quad (4.2)$$

Substituting $Q = CV$, we can estimate the time to charge a capacitor given the input voltage and the target voltage (V_c):

$$t = -RC \cdot \ln \left(1 - \frac{V_c}{V_b} \right) \quad (4.3)$$

To convert to activation frequency we invert t , and also add a delay for the runtime of the Monjolo node while it is booting and transmitting a packet.

$$f = \frac{1}{-RC \cdot \ln \left(1 - \frac{V_c}{V_b} \right) + t_{RUNTIME}} \quad (4.4)$$

By choosing reasonable values for series resistance, the storage capacitance, the target voltage, and the node runtime, we can estimate the Monjolo response, as shown in Figure 4.1.

4.1.3 Dealing with Data Loss

Using the design in Section 4.1.1, every wireless packet that is transmitted must be received to accurately reconstruct the Monjolo curve and underlying energy consumption. If packets are lost, the estimation error will increase and the energy will be underestimated. To mitigate this source of error, Monjolo transmits a monotonically increasing counter value with each packet transmission and the aggregator caches the last packet timestamp and counter value received from each Monjolo sensor. The aggregator can then divide the difference in the packet sequence numbers by the wall time elapsed to account for lost packets. This scheme does not correct for the loss in temporal resolution from dropped packets, as any delay between packets prevents the estimate from being updated. Also, we can only estimate the average energy consumed during the period in which the packets are dropped by assuming a constant draw.

4.1.4 Moderating Packet Delivery Rates

One problem with the Monjolo design is that its packet transmission rate is a function of the measured energy: as the source it is measuring increases, so does the packet transmission rate. Further, the Monjolo system in Section 4.1.3 has no long-term sense of time so it cannot moderate its own packet delivery rate (its power is disconnected after each activation). If multiple Monjolo sensors are colocated, this could in principle saturate the wireless channel, resulting in data loss from packet collisions, or inequitable data delivery ratios.

We address this problem by decoupling the node activations from the packet transmissions. Node activations continue to increment a counter, but the packet transmission rate is capped. One challenge is how to cap the transmission rate, since this requires keeping

track of time in the absence of steady source of power. Our solution notes each packet transmission internally by charging a small capacitor. Over time this capacitor is slowly discharged through a large resistor. If Monjolo’s digital core wakes up before the capacitor has discharged to a low threshold, the core increments the counter but does not transmit a packet. This RC timekeeping circuit provides a simple mechanism to keep time and squelch packet transmissions.

Adding the RC timekeeping circuit decouples packet transmission frequency from the measured energy. This means that the wakeup counter from any two packets, along with the time difference between them, can be used to accurately determine the average consumption over that interval. In addition to the activation counter, which can increment many times between packet transmissions, Monjolo also keeps a sequence number that increments with every packet, which enables the aggregator to detect dropped packets. One important note about suppressing packet transmissions is that Monjolo itself must continue to consume the same quanta of energy during each activation cycle regardless of whether it transmits a packet or not.

4.1.5 Handling Non-linear Recharge Rates

Ideally the recharge rate of the reservoir capacitor – and by extension the activation rate of Monjolo – would be linear with the energy source. However, even if it is not, if the relationship could be described with a simple transfer function, it would allow for flexibility in reporting value for which no exact calibration values exist. In practice, non-linearities in the system mean that this relationship cannot be described by a simple analytical function. We address this by fitting a model to the empirically-derived relationship between activation rate and underlying energy use, and then use this model to estimate the corresponding energy use from the instantaneous activation rate (or interval). We further elaborate on the transfer function and on how we find and use this model in Section 4.2.2.

4.2 Monjolo System Architecture

Monjolo is comprised of two main components: the energy-harvesting sensor and a data aggregator that computes the rate of activations. The sensor and the aggregator must be within single-hop packet transmission range of each other, but a single aggregator can support many Monjolo sensors. The meter senses individual loads while the aggregator converts the resulting activation rate to calibrated units that reflect the energy consumption of the sensed load.

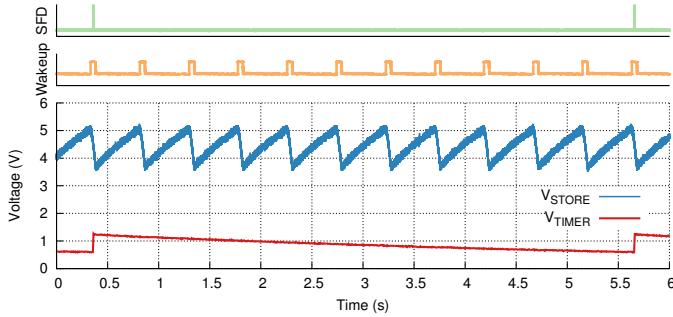


Figure 4.2: Example Monjolo energy harvesting operation. A trace of the energy-harvester charging up the main reservoir capacitor, C_{store} . When the capacitor reaches 5.1 V, the microcontroller boots up. The microcontroller samples the V_{timer} line to determine whether to send a packet or not. When the node wakes up at 5.6 s there is only approximately 0.6 V on the timing capacitor, causing the node to send a packet before recharging C_{timer} .

4.2.1 Hardware Platform

Each Monjolo sensor requires an energy-harvester suited for the energy consumption point to be sensed and a radio. In its basic form, a Monjolo sensor transmits a packet when the power supply has accrued enough energy to do so. This simplicity in Monjolo’s hardware architecture enables its flexibility and can lead to very non-invasive and miniature hardware implementations.

In the general purpose case, each Monjolo sensor also requires a method for ensuring a fixed quanta of energy is used on each activation and a counter. In the Monjolo instantiations in this work, a microcontroller manages the counting and radio transmissions. This increases the energy cost of each activation, but simplifies development and prototyping, while also increasing flexibility.

The general hardware architecture of a Monjolo sensor is comprised of an energy-harvester, a capacitor-based power supply, a microcontroller, a rudimentary timing circuit, and a radio. The examples in Section 4.3 include these basic components.

4.2.2 Sensor Operation

Figure 4.2 shows an example trace of Monjolo operation. As the energy-harvester harvests, its charges a capacitor and increases the voltage on that capacitor, V_{store} , until it reaches a threshold at 5.1 V. A regulator then uses the capacitor voltage to supply a 3.3 V V_{out} rail that powers the remaining board circuitry. This causes the microcontroller to boot up.

During boot, the microcontroller enables the radio to allow for consistent energy consumption across Monjolo activations. After reading and writing non-volatile storage to

increment the wakeup counter, the microcontroller samples the R-C timer by measuring the voltage on the timing capacitor (V_{timer}) to determine if sufficient time has passed since the prior transmission. If the voltage on the cap is low enough, as it is in Figure 4.2 near 5.6 s, Monjolo transmits a packet which causes the radio to issue an SFD pulse as indicated in the figure.

The microcontroller then restarts the RC timer by turning on a MOSFET to recharge the capacitor. If the node sent a packet, it updates the non-volatile memory with the sequence number for the next transmission. Finally, with its tasks complete, the microcontroller leaves the radio in receive mode to discharge the storage capacitor to a low voltage threshold, at which point the power supply disconnects the V_{out} rail, allowing the harvester to recharge the capacitor.

4.2.2.1 Concurrent Charging and Discharging

With unpredictable harvesting from the underlying energy source, it is possible that while Monjolo is active the energy-harvester is still able to harvest energy. However, this can potentially violate the constraint established in Section 4.1.1 that a fixed energy quanta must be consumed over a fixed period of time during each activation cycle. If the harvesting was always constant this effect would be cancelled out, but with variable harvesting different amounts of energy can be consumed between the high and low voltage thresholds. If harvesting during the active time slows the observed activation rate it can alias on to an activation rate for a lower underlying energy consumption point, leading to an estimation error.

To compensate for this situation, Monjolo nodes disable their harvester while in active mode. This helps ensure a fixed energy quanta is used, and as a result more accurate and reliable measurements, at the cost of lost harvesting potential during the time the Monjolo node is active.

4.2.2.2 Wireless MAC Layer

Monjolo operates with a simple MAC protocol running in a star topology. Each node simply transmits when it wakes up to an always-on data aggregator. We choose not to use a more advanced MAC layer because Monjolo is unable to support any existing MAC protocol that relies on scheduling wakeups between nodes. This limitation exists for two primary reasons. First, Monjolo has no backup energy storage to supply an on-demand wakeup. Therefore, it cannot activate at an arbitrary time as required by scheduled MAC protocols. Second, Monjolo has no means of keeping a sufficiently accurate timer active to generate

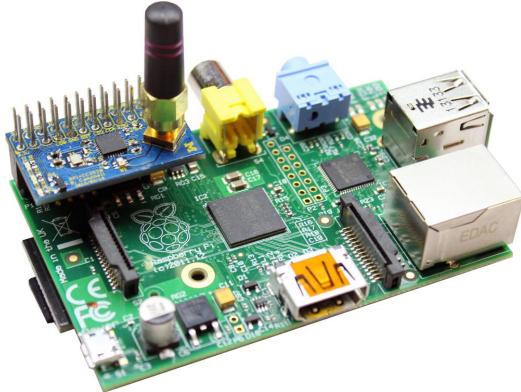


Figure 4.3: RPi Gateway. The gateway is an IEEE 802.15.4 packet receiver shield based on the TI CC2520 radio paired with a Raspberry Pi mini-computer.

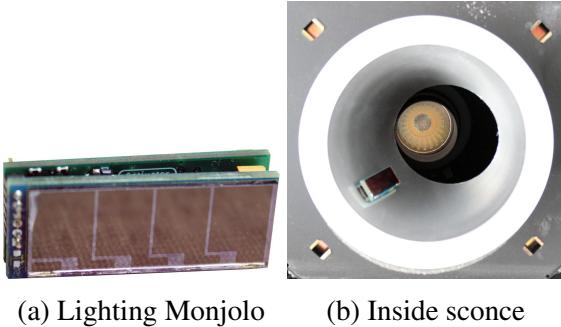
these on-demand wakeups. Since the microcontroller is disconnected from power between activations, all timing information on the microcontroller itself is lost.

This limited MAC layer is acceptable for the Monjolo application. The new approach to sensing advocated by the Monjolo principle – essentially a modern interpretation of the well-established pulse counting method used by utility meters – trades accuracy and reliability for simplicity of the node. Further, because Monjolo takes steps to avoid saturating the wireless channel, and the impact of packet loss is only a transient reduction in temporal precision and not a reduction in overall accuracy, a MAC protocol that ensures reliable or collision-free data transmission is unnecessary.

4.2.3 Data Aggregator

To collect packets from the Monjolo sensors, we use a Raspberry Pi (RPi) [113] embedded computer with a custom TI CC2520 radio shield. The CC2520 is an IEEE 802.15.4 compatible radio and allows the RPi to receive packets from Monjolo nodes. Using the RPi as a base station provides a platform capable of timestamping, processing, and forwarding the data streams from Monjolo sensors.

To estimate the energy consumption where the Monjolo is measuring, the RPi listens for all Monjolo packets. Upon reception, it computes the elapsed time since the last packet and the difference in the counter value, and feeds that into the model that relates energy to activation rate. The estimate is timestamped and transmitted to a server for storage, aggregation, and further processing.



(a) Lighting Monjolo (b) Inside sconce

Figure 4.4: Indoor lighting Monjolo. The power supply of Gecko is re-designed to be the form factor of a commercially available solar panel and the remaining Monjolo hardware components are sized to fit on a matching daughterboard. This sensor can easily and unobtrusively placed near a light.

4.3 Monjolo Implementations: Sensing Energy Consumption in Buildings

To demonstrate the Monjolo principle in practice, we develop several versions of the basic sensor to monitor different energy consumption points within buildings. The first version, a Monjolo for indoor lighting, is the natural extension of the Gecko project that combines the harvester from Gecko with the other Monjolo hardware components. The second instantiation, one for thermal gradients, targets primarily hot water consumption points in a building. This Monjolo uses a thermoelectric generator as a harvester. The third Monjolo monitors AC electricity use by individual loads and circuits. It uses a current transformer with a rectifier as its harvesting front-end.

4.3.1 Indoor Photovoltaics Monjolo

Following the lead of the Gecko project, the Monjolo principle also works when the sensor harvests from indoor lighting. The activation rate of the sensor is proportional to the intensity of the lighting, and this can be correlated with the power draw of the lighting. The power draw of lighting in buildings is often difficult to meter directly because it is often wired to separate panel boxes from other circuits, driven by at different AC voltages, and difficult to trace back to a central point. By simply placing an unobtrusive Monjolo sensor inside or near a light, as in Figure 4.4b, Monjolo provides an easy-to-deploy method for metering built-in lighting.

Figure 4.5 shows how this type of Monjolo responds to both incandescent and fluorescent lights over a range of light level settings and power draws. As expected, as the lighting draws

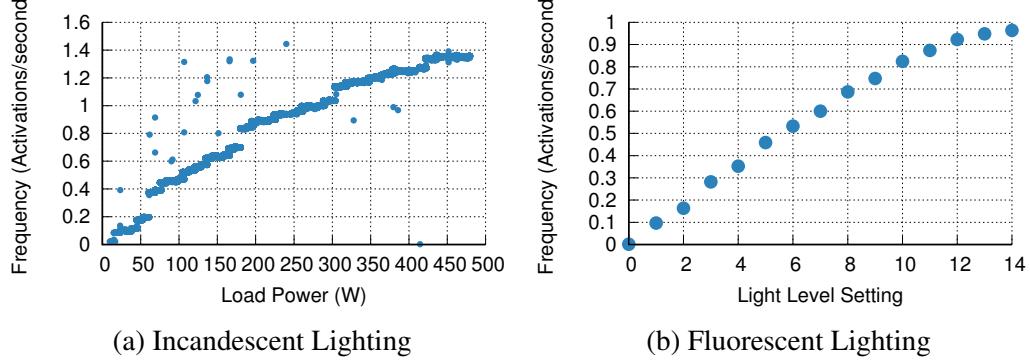


Figure 4.5: Activation rate of indoor lighting Monjolo from incandescent and fluorescent lights.

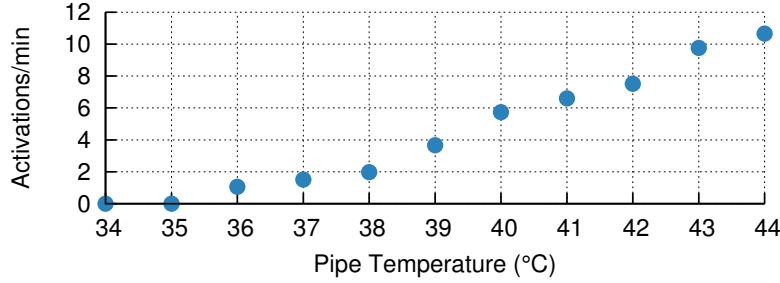


Figure 4.6: Activation rate of Thermes at a range of pipe temperatures in 23 degrees C ambient conditions. As the temperature difference between each side of the Peltier module increases, so does the average rate of activations over the first five minutes of harvesting.

more power the activation rate of the Monjolo sensor increases.

4.3.2 Thermal Gradients Monjolo

Another instantiation of the Monjolo principle, and another critical sense point in buildings, is Thermes (Figure 4.7, a sensor that meters heat or hot water pipes. Thermes uses a thermo-electric generator (TEG) and the temperature differential between hot water flowing in a pipe and the ambient air to harvest. The rate of harvesting is proportional to the temperature differential and, by extension, the temperature of the water in the pipe, as shown in Figure 4.6. To detect water events we note the rate at which the energy-harvesting power supply is able to harvest.

To achieve this, a power supply attached to the TEGs stores harvested energy in a bank of capacitors. Until the voltage of the capacitors reaches a certain level, the microcontroller and radio are power gated, allowing for a faster recharge rate than maintaining sleep mode. Once sufficient energy has been accumulated, the power supply activates the node which

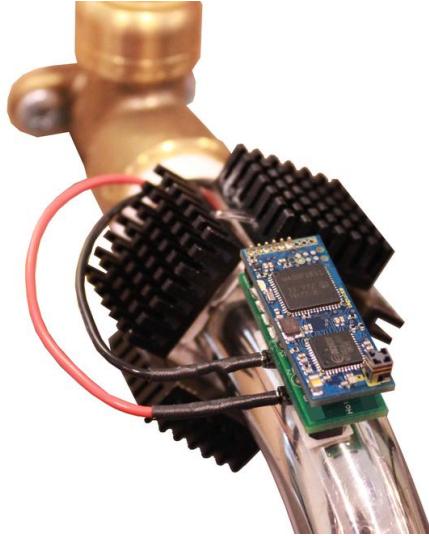


Figure 4.7: Thermes device attached to a pipe. Thermes is designed to be a bracelet that can easily stick around a hot water pipe.

immediately transmits a packet. The node remains on until the stored energy reaches a lower threshold, ensuring that a fixed amount of energy is used on each activation.

Due to the operating regime of this sensing style and the small energy budget that accompanies it, the device itself knows little about what it is monitoring. Instead, an always-on central receiver service accepts the transmitted packets and processes them to understand properties of what is being sensed. The receiver calculates the rate of activations, or the rate of harvesting, based on the timestamp of the packets. Further, the service is responsible for estimating when the water event stopped and started. This is necessary both because there is a delay between when the pipe begins to heat and when the power supply is charged, and because the pipe will remain hot after the water stops causing additional activations.

The only state maintained on the microcontroller is an activation counter. This counter is transmitted in every packet and protects the system against dropped packets by allowing the receiver to calculate an average activation rate even if a packet is missed. Alternatively, this counter can be leveraged to rate-limit packet transmissions. For example, a transmission could occur every five activations, rather than on every activation.

This sensing style and design point allows for a small, wireless “bracelet” like device, composed of a series of TEG devices, the power supply, and a computation core, to be clipped around a pipe in order to monitor it. By not requiring a display, battery, or additional sensors, the size can be kept small for unobtrusive monitoring.

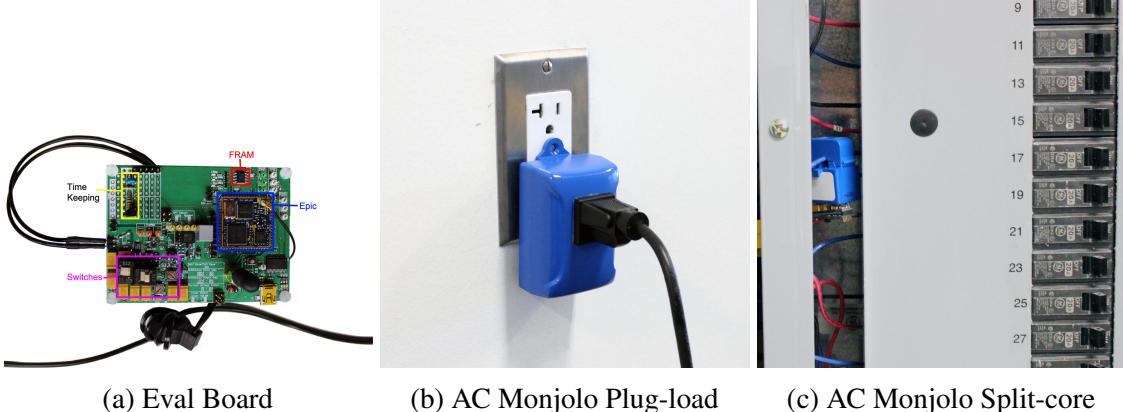


Figure 4.8: Instantiations of AC Monjolo. An evaluation version allows for adjusting various capacitance and operational settings to explore how design choices affect the system. The plugload version uses an internal solid-core current transformer and can meter any plug-load. The split-core version can clip around a wire in a circuit panel to meter the entire circuit.

4.3.3 AC Power Meter Monjolo

The third implementation of the Monjolo principle for monitoring buildings is an AC power meter. The meter, shown in Figure 4.8b, uses a current transformer to harvest from the electromagnetic field generated by AC current flowing to a load. Current transformers (CT) produce an output voltage that is proportional to the AC current flowing in the wire they are around. Typically this signal is measured directly, but for Monjolo it is rectified and used as the energy-harvesting frontend. This harvester is both suitable for the Monjolo operation and allows the meter to be non-contact: the AC line does not have to be interrupted nor tapped into. This, along with a split-core CT, enables a circuit-level version (Figure 4.8c) that can easily clip around a wire in a circuit panel for quick deployment. The meter also works with plug-loads, but the load does have to be temporarily unplugged to install the meter.

With harvesting from an AC wave, however, there are additional complexities related to ensuring a successful Monjolo meter given the underlying harvesting characteristics. The remainder of this section details the meter’s implementation as well as the challenges and workarounds for this type of harvesting frontend.

4.3.3.1 Energy-harvesting Power Supply

The AC Monjolo’s energy-harvesting power supply serves as the system’s only sensor. The power subsystem harvests energy from the primary AC load and accrues enough energy over time to wake up and transmit a packet. The aggregator uses the frequency of these wakeups to infer primary power. The following constraints are necessary for proper operation: (i) the

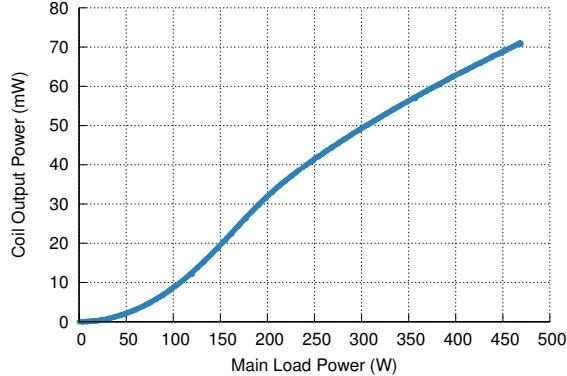


Figure 4.9: Transfer function of the current transformer used in the energy-harvesting power supply when connected to a 430Ω resistor. The power the coil can deliver varies non-linearly as the load power changes. This affects the activation interval of the sensor, causing the activation rate to be non-linear with load power.

system must be able to harvest energy over the full spectrum of possible load powers, (ii) the activation rate must be a monotonically increasing function of the load's power draw, and (iii) the power supply must never supply so much power as to enter an always-on state. In an always-on state, the harvester can supply more power to the system than the system can dissipate. This saturates the harvester, eliminates the wakeups, and leaves the system without its one sensor.

To achieve these goals with low power and space overhead, we employ the CR Magnetics CR2550, a 1:3000 turns-ratio current transformer [17]. The CR2550 magnetically couples to the load's supply wire, and supplies the Linear Technology LTC3588 energy harvester [76]. We find that six turns of the primary coil, for a turns ratio of 1:500, ensures that the minimum input voltage requirement of the LTC3588 is met at a primary load power of 17 W and the system still alternates between charge/discharge states at 480 W. Hence, AC Monjolo can meter loads from 17 W to at least 480 W. This range could be extended further by increasing the primary turns and increasing the energy quanta consumed in each cycle. Figure 4.9 shows the delivered power from the coil across a 430Ω resistor, which draws an average current that is close to that of the LTC3588 load. The curve is non-linear at load power levels below 200 W and this non-linearity manifests itself in AC Monjolo's activation rate.

The LTC3588 energy harvesting integrated circuit has two piezo inputs and uses an internal full wave rectifier to charge a capacitor over time. By disconnecting one of these inputs and shunting it to ground, the user can force the LTC3588 to only perform half-wave rectification. We use this mode and employ half-wave rectification to minimize the AC Monjolo's internal power draw. Using full-wave rectification would only increase the activation rate, and thereby increase the average power draw, while just slightly improving

the granularity of the power measurements.

After the AC waveform is rectified, the LTC3588 employs a boost converter to increase the low amplitude signal to charge the storage capacitor (C_{store}). When this capacitor reaches the threshold voltage of 5.1 V (for a 3.3 V output), the LTC3588 enables a buck converter to supply this output on the V_{out} rail. When the voltage on C_{store} drops to 3.8 V, the LTC3588 disables V_{out} and continues to charge C_{store} back to 5.1 V.

Two capacitors play a critical role in system performance: C_{store} and C_{out} , the latter a capacitor on the regulator's V_{out} line. The values of these capacitors affect the system operation and measurement granularity. Increasing the value of C_{store} increases the time required to recharge and reduces granularity, but it also increases the runtime duration of each activation cycle. Increasing the value of C_{out} reduces voltage ripple on V_{out} and increases runtime, but also increases ramp-up time for V_{out} .

We evaluate several options for the capacitance C_{store} with the goal of finding the minimum capacitance with which the system can complete all of its required tasks. Because the power delivered by the current transformer for a given load power is fixed, capacitance can affect only the activation rate but not overall system power. The activation duty cycle, and by extension the average power draw, is fixed because the period and runtime scale proportionally as the value of C_{store} changes. Therefore, the minimum possible capacitance needed to run the digital core is optimal. A higher capacitance reduces granularity without reducing power draw. For our system's operating range of 17 W to 480 W, we empirically determine that a value of 500 μ F for C_{store} provides high granularity and sufficient time to complete all operations, including updating the counter, sampling the timing capacitor, and transmitting a packet. We use a tantalum capacitor for C_{store} because tantalums have low leakage and high charge storage density.

We evaluate three possible choices for the output capacitance: 47 μ F, 200 μ F, and 247 μ F. We observe that although runtime does increase with increased input capacitance, the effective runtime actually decreases due to the increase in ramp-up time. Since all three values sufficiently reduce voltage ripple, we find the optimal value to be 47 μ F for our particular design point.

4.3.3.2 Analog Packet Suppression Timer

To suppress rapid packet transmissions, AC Monjolo employs an RC timer. The digital core charges a capacitor (C_{timer}) after each packet transmission. This capacitor discharges over time, allowing a sensor to keep an approximate sense of elapsed time since the prior packet transmission. At every activation, AC Monjolo samples V_{timer} and suppresses packet transmissions if the capacitor has not discharged to a low threshold level.

Our implementation of AC Monjolo charges a $2.2 \mu\text{F}$ electrolytic capacitor which discharges both through its own leakage and through a $4.7 \text{ M}\Omega$ resistor. The capacitor is sized to limit the inrush current while offering for a multi-second discharge delay. The value of the resistor creates a time constant of about 10 seconds. We find that for the RF conditions and deployment density in our lab, a reasonable inter-packet interval is about 5 s, but designing for a τ of 10 s allows each AC Monjolo sensor to enact a higher delay in software, if needed, such as when there is a high density of sensors.

4.3.3.3 Digital Processing Core

AC Monjolo's digital core consists of three main components: a microcontroller (TI MSP430F1611), an IEEE 802.15.4 compatible radio (TI CC2420), and FRAM (Ramtron LM25L04B). The microcontroller and radio are elements of the Epic Core [32]. The FRAM handles the non-volatile storage of the activation counter and packet sequence number. FRAM is preferred to flash or EEPROM for its high-speed reads and low-energy writes. To reduce power, the flash chip on the Epic Core is removed.

The software on the node is optimized for fast and simple operation. Each activation consists of an FRAM read and write, an ADC sample, and a potential radio transmission. Because the LTC3588 disconnects the V_{out} rail when the V_{store} voltage falls below a low threshold, the microcontroller cold boots during each activation cycle. This both simplifies the software and corrects transient errors, as the subsequent boot resets the microcontroller state.

4.3.3.4 Concurrent Activations and Charging

The choice of the LTC3588 has interesting ramifications for overall AC Monjolo operation. First, the LTC3588 does not disconnect the input power source while the output is enabled. When the current transformer input voltage exceeds the minimum LTC3588 input threshold, the LTC3588 will supply current to C_{store} as usual. Unfortunately, this violates the constraint established in Section 4.1.1 that a fixed energy quanta must be consumed over a fixed period of time during each activation cycle. Because the LTC3588 is able to charge C_{store} faster at higher load wattages, the duration and energy depletion of each wakeup will increase with larger primary loads. The change in the wakeup duration introduces error into the power estimation algorithm, which is based on activation rate. If a ΔP increase to a primary load P_{load} causes the same increase in both the AC Monjolo activation rate and duration, then the overall packet transmission rate of both P_{load} and $P_{\text{load}} + \Delta P$ will be identical and indistinguishable.

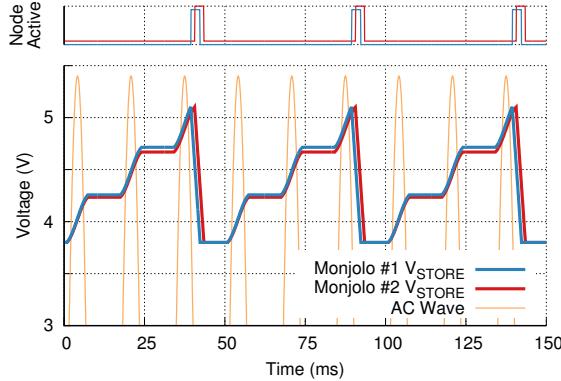


Figure 4.10: The charging cycle of two AC Monjolo sensors with a half-wave rectifier. AC Monjolo #1 has a slightly higher load attached and is able to charge up slightly faster than AC Monjolo #2. However, after the discharge phase both AC Monjolo sensors start charging at the same point in time, causing their wakeup frequency to equalize with only a slight phase shift. This lack of continuous charging leads to a stair-step relationship between activation rate and load power, which can result in a loss of precision, particularly at high load powers.

To compensate for this situation, we add two pull-down MOSFETs, one on each output leg of the energy-harvesting current transformer, with the MOSFET gates connected to the regulator's V_{out} line. This causes the LTC3588 to disconnect its own inputs when its buck converter is enabled and the digital core is active, thus ensuring a fixed discharge time and energy during each activation.

4.3.3.5 Quantization Issues from Harvesting AC

A by-product of using an AC input signal as the energy harvesting source is that there is a period during each 60 Hz cycle when the current transformer output does not meet the minimum LTC3588 input voltage, so AC Monjolo cannot charge. This seemingly benign observation has deep implications for AC Monjolo's ability to map different load powers to distinct activation rates. To better explain this phenomenon we use a slightly stylized illustration of the charging cycles of two AC Monjolo sensors as shown in Figure 4.10. In this simulation, AC Monjolo #1 is connected to a slightly higher power load than AC Monjolo #2. Therefore, its input AC waveform is also slightly higher in magnitude. We also assume for the simplicity of the illustration that each Monjolo node transmits a packet during every activation cycle.

Both Monjolo nodes require three cycles of the 60 Hz wave to charge, with the higher power circuit connected to AC Monjolo #1 reaching 5.1 V slightly sooner. It follows that the data aggregator will receive AC Monjolo #1's packet equally sooner. Both capacitors discharge at the same rate (a rate which is a fixed property of the system) and finish

discharging during a negative “off” half-cycle. At this point both AC Monjolos *must* wait until the next “on” cycle to start recharging. This delay means that although AC Monjolo #1’s packet will consistently arrive moments before AC Monjolo #2’s, *both will wakeup and transmit at the same rate*. The result is that all load powers that cause AC Monjolo to finish discharging in the same “off” period quantize to the same activation rate.

This quantization phenomenon has a more pronounced effect at higher load powers, where the LTC3588 requires fewer cycles of the 60 Hz wave to recharge V_{store} . In this operating regime, the amount that the primary load must increase in order to increase the input power such that one fewer 60 Hz cycle is required to meet the activation threshold is much larger than at much lower loads. Stated differently, the amount each charge cycle must additionally contribute to V_{store} to drop from 100 charge cycles to 99 charge cycles is much lower than the amount of extra energy required to drop from 10 cycles to 9. This manifests in the stair-step effect visible at higher load powers in Figure 4.11a. For instance, as the load increases from 445 W to 465 W, the resulting AC Monjolo activation rate remains constant.

Figure 4.11a does not exhibit crisp steps, however, because of the consistent time duration and energy use during the AC Monjolo discharge phase. Without the enhancements described in Section 4.2.2.1, once AC Monjolo begins discharging, it continues to discharge until it reaches a point in time at which the rectifier is in an “off” state. That is, AC Monjolo uses any “on” state to prolong its wakeup duration and does not discharge C_{store} until the subsequent “off” state. This forces every wakeup interval to delay until it can take the next step up. Figure 4.11b shows that without the MOSFETs, the quantizing effect is significantly worse. By satisfying the consistent discharge requirement, the quantizing effect can be mitigated for many load power ranges, illustrating precisely why the consistent discharge requirement is necessary for this design.

Using half-wave rectification exacerbates the quantizing effect because it increases the duration of the “off” cycle. In practice, however, we notice that using half-wave rectification only displays pronounced steps at a load power greater than 270 W, and the effect on percent error is relatively small – at most $\pm 5\%$ of the true value.

4.4 Monjolo Evaluation

We use the Monjolo implementations to evaluate how well the Monjolo principle works as the base of an energy meter. The primary evaluation metric is the accuracy of the meter with respect to how well the system reports energy or power consumption. We must ensure the data created by the meter is actionable, but we do not intend the meter to produce revenue-grade measurements. The other key evaluation metric is the update rate of the meter, or how

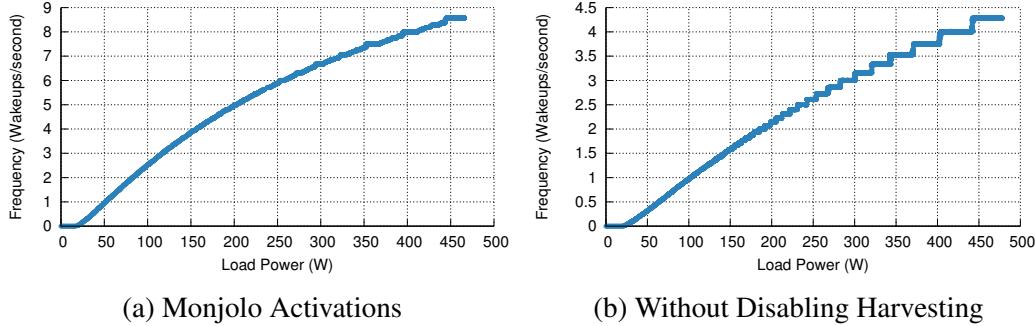


Figure 4.11: Frequency of Monjolo activations over a range of loads. (a) shows that as the primary load’s power increases from zero to 475 W, the activation rate increases from zero to 8.5 Hz. Although not linear, this monotonically increasing relationship between activation rate and load power suggests that Monjolo may be a viable approach to metering a range of AC loads. In (b), the activation rate is quantized. If the LTC3588 energy harvester is permitted to continue to supply energy to C_{store} during the discharge phase of operation, then the quantizing effect inherent in the system is substantially more pronounced. Particularly at high load powers, the steps span a nearly 50 W range, which dilutes precision and introduces $\pm 5\%$ error.

often a measurement is produced. With energy-harvesting and power proportionality, the meter’s reporting rate is dependent on the underlying energy source. If for common loads this rate is too slow, the meter is not viable in actual scenarios.

We also evaluate Monjolo by returning to the model from Section 4.1.2 and comparing how well it predicted the actual operation of the Monjolo meters.

4.4.1 Sensing Accuracy

To evaluate how accurate Monjolo is as an energy meter, we focus on a particular implementation, namely the AC Monjolo. In this section, we observe the end-to-end accuracy of the meter by comparing the final reported power measurement from the AC Monjolo system with the reading from a calibrated ground truth AC meter. The Monjolo is calibrated against a unity power factor load for these tests.

To start, we test on a fully resistive load, by using a test rig that randomly selects power values for the AC load and switches after random intervals. This highlights AC Monjolo’s response to different load powers and a range of power transitions. Figure 4.12 shows the results of a 30 minute run of this application. AC Monjolo tracks ground truth very closely when the load is constant with an error of only 1%.

We also examine how Monjolo works with reactive loads by running AC Monjolo against ground truth for four other loads, as shown in Figure 4.13. These loads exhibit a variety of

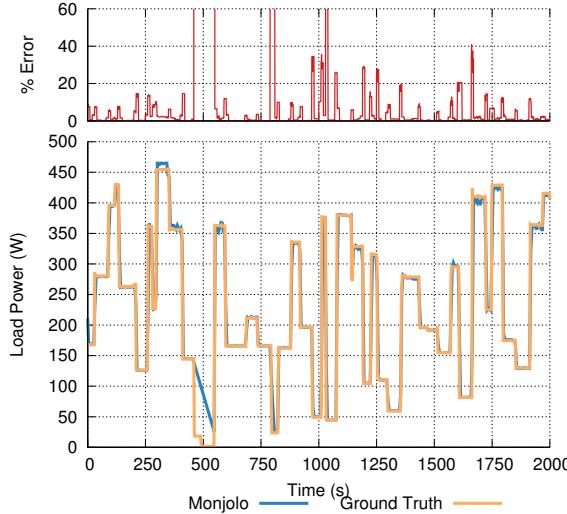


Figure 4.12: Accuracy of AC Monjolo’s power measurements over a range of different loads. When the load is constant, AC Monjolo’s error is very low at only 1%. When the load changes, AC Monjolo’s relatively slow response time causes sharp instantaneous error while the data aggregator waits to receive a packet informing it that the load has changed. When the load is very small, as it is at $t = 500$ s, AC Monjolo is unable to transmit and the resulting error is quite large, although a less naive power estimation algorithm would be able to correct for this. Ignoring the few sections where AC Monjolo is drastically incorrect, AC Monjolo shows an average error of 3.7% across a randomly varying resistive load.

different real-world power profiles. For loads with rapid changes in their power draw, AC Monjolo averages the power draw and is unable to report the swings. Also, for loads with non-unity power factors, AC Monjolo generally over-estimates power draw. This is likely because it is measuring not just true power but also reactive power. AC Monjolo is able to track the actual power draw of the various loads but continues to overestimate.

When well calibrated, Monjolo can be highly accurate, as demonstrated with the resistive load case for AC Monjolo. Compensating for the errors present with reactive loads may be possible with additional calibration, or by informing the data aggregator of the power factor of the measured load in advance. The following chapters show how hardware and software improvements to the AC Monjolo design can also enable higher accuracy.

4.4.2 Activation Rate

Another important metric for all energy-harvesting sensors is activation rate, or how often they are able to measure. If harvesting is too slow the resulting data stream may not be able to support higher-level applications and algorithms. Again we look to AC Monjolo, and compare its two implementations: the plug load version with a solid-core current transformer

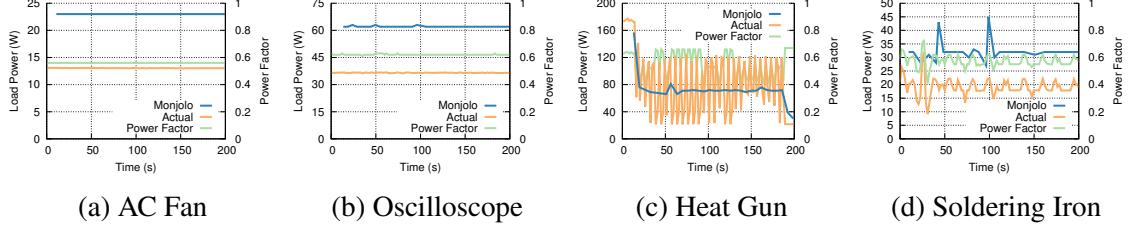


Figure 4.13: AC Monjolo performance on loads with non-unity power factors. Each figure shows the ground truth power and power factor measurement of the load, and AC Monjolo’s estimate of the power. If the power factor remains constant, AC Monjolo estimates the power with a constant offset, as shown in 4.13a and 4.13b. The heat gun in 4.13c is an interesting case when the high frequency oscillations cause AC Monjolo to estimate an average value. The fluctuating power factor in 4.13d causes AC Monjolo to sharply overestimate, but it quickly recovers. While non-unity power factors cause AC Monjolo to lose accuracy, we believe the magnitude offset can be compensated for with other mechanisms.

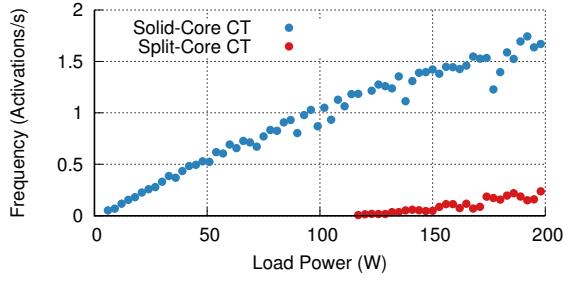


Figure 4.14: Activations/second for two coils over a range of resistive loads. The first coil is a solid-core current transformer (CT) with 10 windings of the AC phase line around it. The second is a split-core CT. Below 5 W for the solid-core and below 117 W for the split-core the device is unable to harvest enough energy to activate and therefore is unable to perform any metering. Beyond that, as the load power increases, the ability for the node to harvest increases.

with several secondary turns, and a circuit-level version with a split-core current transformer and a single secondary turn.

To quantify how often an energy-harvesting power meter can perform a measurement, we sweep over a range of power draws using a fully resistive test load and monitor the activation rate of the sensor. The results are shown in Figure 4.14. For the solid-core version, the node is able to activate once the load reaches 5 W, and it activates more rapidly as the load power increases. At 100 W the sensor activates roughly once per second, which because the sensor needs two activations to take and transmit a measurement, equates to a maximum sample rate of 0.5 Hz.

For the split-core version, fewer windings mean a higher power load is required for

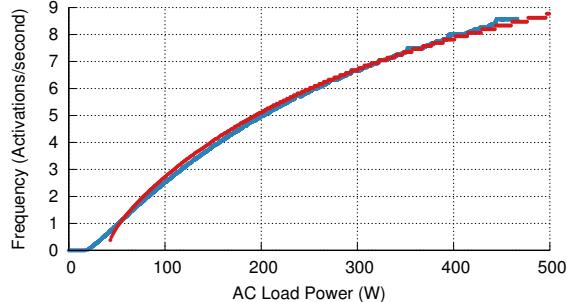


Figure 4.15: Theoretical response rate for Monjolo compared to observed response for AC Monjolo.

the node to begin activating. At approximately 117 W, the power supply is able to harvest sufficient energy to begin sampling. As in the solid-core design, the sample rate increases with load power. Because the split-core design clips around a wire it is most suitable for metering entire circuits. At the circuit level, the power of multiple loads will sum and the higher minimum measurement power will have less effect than with individual loads.

4.4.3 Comparison to Theoretical Model

We revisit the theoretical model for Monjolo in light of the observed response seen by the implementations. The comparison when values measured from the AC Monjolo are inserted into the model is shown in Figure 4.15.

4.5 Deltaflow: Correcting for Uncalibrated Pulse Streams

When well calibrated, Monjolo sensors can meter with sufficiently high accuracy. However, in many cases it is not feasible to calibrate a Monjolo for a specific load. To compensate, we explore a system that attempts to compensate for errors from individual meters by leveraging an aggregate energy draw signal. We then treat the Monjolo sensors attached to loads that contribute to the aggregate as generic pulse sensors. Each pulse sensor operates according to a simple principle: the higher the power draw of the load, the greater the frequency of the pulses. The Deltaflow system uses these pulse streams as hints about each load to disaggregate the aggregate measurement into each individual load’s contribution.

The Deltaflow system architecture is shown in Figure 4.16. A power meter that is monitoring aggregate energy flow upstream from the target loads (e.g., a utility meter) reports aggregate measurements to the Deltaflow server. Additionally, the individual loads to be metered are instrumented with a suitable pulse sensor that transmits a representative

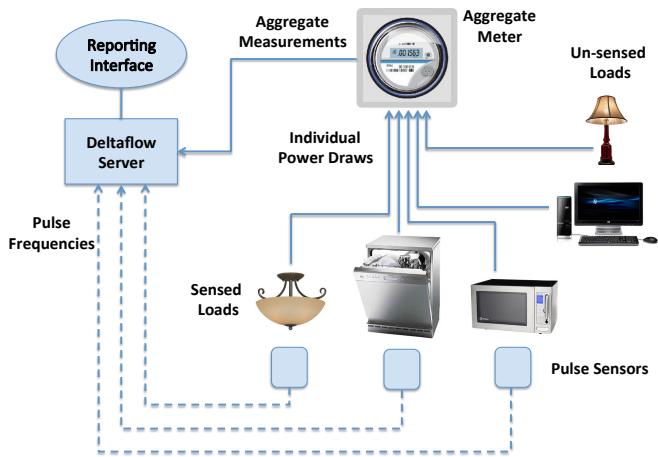


Figure 4.16: System architecture. The Deltaflow server takes in aggregate power draw measurements and the activation frequencies of energy-harvesting sensors attached to individual loads. By augmenting aggregate power measurements with sensor activation (or pulse) frequencies that are correlated with power draw, Deltaflow determines the individual energy consumption of the sensed loads. The system is able to function even if some loads remain unsensed.

pulse stream to the Deltaflow server. By augmenting the aggregate measurements with pulse frequencies that change as the individual load's power draw changes, the Deltaflow server characterizes the sensor's response to changes in the power state of the load it is attached to. The server uses these models of the sensors to perform calibrated disaggregation and provide power draw estimates for the individual loads being monitored.

4.5.1 Dissaggregation Methodology

The Deltaflow server receives two input streams: aggregate ground truth measurements (e.g. Watt or Whr) and pulse frequencies from each sensor attached to the instrumented loads. The server uses each of the pulse streams to decompose the aggregate measurements into the contributions from each load.

4.5.1.1 Sensor Calibration

The server starts by trying to estimate a *calibration function* for each sensor. This calibration function must be determined at runtime, and not *a priori*, for two main reasons: manufacturing differences, and sensor placement, which is unknown until after installation yet affects the response of the lighting Monjolo. We determine the sensor specific calibration function by first noting that the sensor pulse frequency, s_i , of load i is a function of the power draw of

load i

$$s_i = f_i(p_i) \quad (4.5)$$

Examples of this relationship are shown for AC Monjolo sensors in Figure 4.11a and for lighting Monjolos in Figure 4.5. Given that we know s_i and want to determine p_i , we can express the relationship as:

$$p_i = f_i^{-1}(s_i) = g_i(s_i) \quad (4.6)$$

Then, given a pulse rate, we can determine a power estimate for each sensed load using the calibration function g_i .

To determine each g_i we use information about how the pulse streams change when the aggregate measurements change. However, to avoid searching through the infinite space of possible functions, we choose a general form of g_i with which to work. We additionally assume that the general form will be the same for every instance of a particular type of sensor. Requiring foreknowledge of the function for each class of sensor is reasonable because it can be determined once, at design time. This shifts the burden of configuration from sensor users to manufacturers.

We use datasets of the Monjolo activation rates to estimate the sensor calibration function. We find that the best-fit functions are second-order polynomials, and therefore make the assumption that the calibration function can be approximated by a monotonically increasing polynomial of degree two or less. Thus Equation 4.6 becomes:

$$p_i = \alpha_i s_i^2 + \beta_i s_i + \gamma_i \quad (4.7)$$

Now, to determine the calibration function, Deltaflow need only choose the coefficient parameters α_i , β_i , and γ_i for each sensor that best fit with historical data.

4.5.1.2 Monjolo Sensors on Every Load

To determine the polynomial coefficients, we start with the simplest case: assume all n loads comprising the aggregate are metered. If we let M_t be the ground truth aggregate meter reading at time t and let $s_{i,t}$ be the pulse frequency for sensor i attached to load i at time t , then we have the relationship:

$$M_t = \sum_{i=1}^n p_i(s_{i,t}) \quad (4.8)$$

Using the least squares method, let \mathbf{M} be a vector of aggregate measurements M_t taken at different times t . Let \mathbf{A} be a matrix of calibration function terms derived from pulse frequencies, where each row \mathbf{A}_i is a vector of all the terms in each sensor's calibration

function at time t , stripped of their coefficients.

$$\mathbf{A}_i = \begin{bmatrix} s_{1,t}^2 & s_{1,t} & 1 & s_{2,t}^2 & s_{2,t} & 1 & \dots & s_{n,t}^2 & s_{n,t} & 1 \end{bmatrix} \quad (4.9)$$

Let \vec{x} be the vector of coefficients where $x_{i,j}$ is the coefficient of the j th term in the calibration function for sensor i .

$$\vec{x} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \dots & \alpha_n & \beta_n & \gamma_n \end{bmatrix} \quad (4.10)$$

Given data points for aggregate power draw and sensor pulse frequencies from some m different points in time, Equation 4.8 can be written in matrix form as:

$$\mathbf{M} = \mathbf{A} \vec{x} \quad (4.11)$$

$$\mathbf{M} = \begin{bmatrix} s_{1,t_1}^2 & s_{1,t_1} & 1 & \dots & s_{n,t_1}^2 & s_{n,t_1} & 1 \\ s_{1,t_2}^2 & s_{1,t_2} & 1 & \dots & s_{n,t_2}^2 & s_{n,t_2} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ s_{1,t_m}^2 & s_{1,t_m} & 1 & \dots & s_{n,t_m}^2 & s_{n,t_m} & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \\ \vdots \\ \alpha_n \\ \beta_n \\ \gamma_n \end{bmatrix} \quad (4.12)$$

With these definitions, we can formulate our optimization problem as:

$$\min_{\vec{x}} \|\mathbf{M} - \mathbf{A} \vec{x}\|_2^2 \quad (4.13)$$

The solution to this problem is the value of \vec{x} —i.e. the set of coefficients for our calibration functions—that gives load estimates that best match our aggregate readings.

Note that while our calibration functions may be non-linear, by providing the values of each non-linear term we have reduced calibration to a linear least squares problem. Use of the l_2 norm makes the regression more susceptible to noisy outliers than the l_1 norm, but it is much easier to compute. Outlier identification techniques can be used if the system converges to an unsatisfactory result.

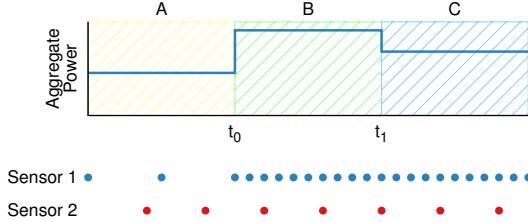


Figure 4.17: Effect of hidden loads. The change in aggregate power at time t_0 is attributed to the load metered by Sensor 1 whose frequency changes at that time. The aggregate also changes at t_1 , but since neither sensor changes, we attribute the change at t_1 to a hidden load and ignore state C. This allows us to discard the effects of non-aliasing uninstrumented hidden loads.

4.5.1.3 Compensating for Unmetered Loads

The above approach describes finding the calibration coefficients in the ideal case when all loads are instrumented. Unfortunately, that approach does not work if even a single load is left uninstrumented, as Equation 4.8 no longer holds. The aggregate M_t now becomes the sum of both sensed loads and *hidden loads*.

$$M_t = \sum_{i=1}^n p_i(s_{i,t}) + \sum_{i=1}^k h_{i,t} \quad (4.14)$$

Let us assume that there are no aliasing hidden loads—that is, no hidden load changes at the same time as a sensed load. In that case, we can choose to select just those data points that represent states where sensed loads changed. Figure 4.17 illustrates such a situation. At time t_0 the pulse frequency of Sensor 1 changes along with the aggregate. However, at time t_1 , the aggregate changes but both Sensor 1 and Sensor 2's frequencies stay the same. We attribute the change in aggregate power to a hidden load and ignore state C. We examine aliasing loads further in the discussion section.

For a data set in which between any two times $t = 0$ and $t = 1$ only sensed loads change, the following holds true:

$$M_1 - M_0 = \left(\sum_{i=1}^n p_i(s_{i,1}) + \sum_{i=1}^k h_{i,0} \right) - \left(\sum_{i=1}^n p_i(s_{i,0}) + \sum_{i=1}^k h_{i,0} \right)$$

$$\Delta M_t = \sum_{i=1}^n p_i(s_{i,1}) - p_i(s_{i,0})$$

$$\Delta M_t = \sum_{i=1}^n (\alpha_i s_{i,1}^2 + \beta_i s_{i,1} + \gamma_i) - (\alpha_i s_{i,0}^2 + \beta_i s_{i,0} + \gamma_i)$$

$$\Delta M_t = \sum_{i=1}^n \alpha_i (s_{i,1}^2 - s_{i,0}^2) + \beta_i (s_{i,1} - s_{i,0}) \quad (4.15)$$

When calculating the change in estimated load power draw, the constant term γ_i drops out of the model. However, due to the energy-harvesting nature of our pulse sensors, we know that when the load power draw p_i is zero, the pulse rate s_i is also zero. This means that γ_i must be zero as well for all i . Our calibration function is now of the form:

$$p_i = \alpha_i s_i^2 + \beta_i s_i \quad (4.16)$$

If we define a new matrix $\Delta\mathbf{A}$ such that

$$\Delta\mathbf{A}_i = \begin{bmatrix} (s_{1,t_1}^2 - s_{1,t_0}^2) & (s_{1,t_1} - s_{1,t_0}) & \dots \end{bmatrix} \quad (4.17)$$

then

$$\Delta\mathbf{A} \vec{x} = \Delta\mathbf{M} \quad (4.18)$$

Solving the optimization problem

$$\min_{\vec{x}} \|\Delta\mathbf{M} - \Delta\mathbf{A} \vec{x}\|_2^2 \quad (4.19)$$

will give us the coefficients for all of the calibration functions. Many existing libraries provide optimizers that can easily solve optimization problems of this formulation. In our implementation we used the non-negative linear least squares function provided by Python's SciPy library.

4.5.1.4 State Identification

The data points used to construct the optimization problem critically determine the performance of the regression. For example, when a load changes power states, the aggregate may reflect the change sooner than the pulse sensor which only activates once enough energy has accrued to power the sensor, thus causing a synchronization error. We find that identifying steady states in the data (using heuristic thresholds) and selecting a median or mean data point to represent the state compensates for this error. These data points are then used to determine the deltas between distinct adjacent stable states when constructing the calibration Equation 4.18.

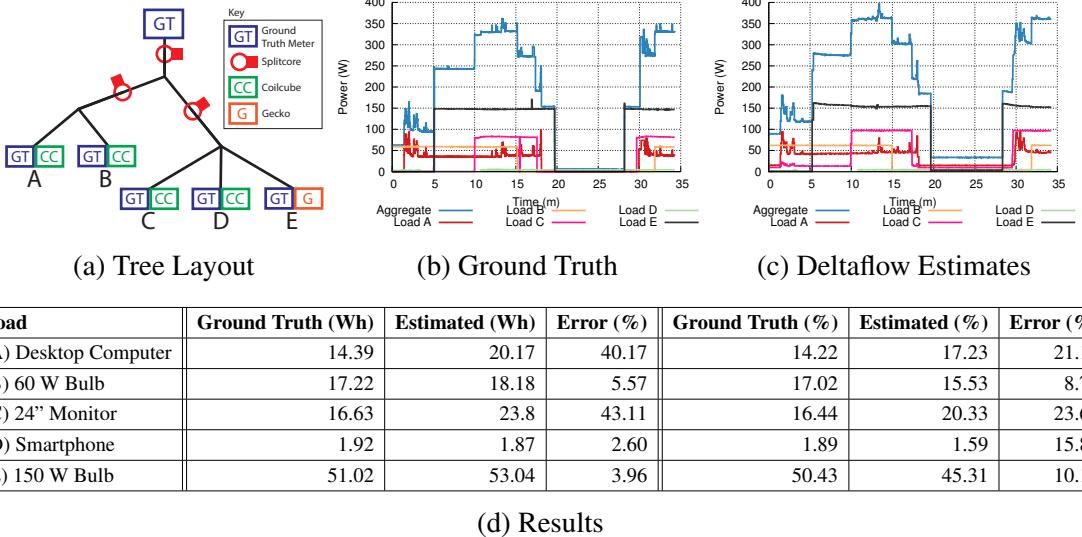


Figure 4.18: Deltaflow performance on a realistic tree of loads. Figure 4.18a shows the hierarchy of loads and the corresponding sensors. Figure 4.18b shows the ground truth power traces for the aggregate and each load. Figure 4.18c shows the Deltaflow estimates for all five loads and an estimated aggregate for comparison purposes. Figure 4.18d shows the breakdown of overall energy usage by each load and the estimate provided by Deltaflow, as well as the percent of the total energy usage each load represents. While the error for loads (C) and (E), which change power states at similar times, is high ($>40\%$), Deltaflow handles loads that transition independently successfully. Also, Deltaflow is able to successfully integrate streams from multiple types of pulse counting sensors.

4.5.2 Deltaflow Performance

To evaluate Deltaflow, we mimic a realistic load tree as one might find in a residential or commercial building by constructing a load hierarchy and instrument it with pulse sensors as shown in Figure 4.18a. The five loads at the leaves of the tree are (A) desktop computer, (B) 60 W incandescent light bulb, (C) 24" monitor, (D) smartphone charger, and (E) 150 W incandescent light bulb. The 150 W light bulb is monitored with a lighting Monjolo sensor and the other four loads are attached to AC Monjolo sensors. Three split-core current transformer sensors monitor the aggregate and each main branch of the tree. Each load and the aggregate is also metered with a ground truth meter, so we have full knowledge of the power draws.

Figure 4.18b shows ground truth power for each load as well as the aggregate, and Figure 4.18c shows the resulting estimates of power draw from the Deltaflow system and the sum of the estimates to form an estimate of the aggregate. The primary error in Deltaflow's results is overestimation. Notably, between $t = 20$ min and $t = 28$ min, all loads except for

(D) are off, causing the ground truth aggregate to be very low. However, Deltaflow estimates the off loads at a higher power state, causing the estimated aggregate to be significantly higher. This error primarily stems from the design of the pulse meters. When no current is flowing to each load, the energy-harvesting pulse meters are unable to report any pulses. Without data from the pulse streams, Deltaflow does not update its estimate for the load power until it receives a new pulse, which it then uses to backfill the missing data. This causes it to overestimate during periods in which most loads are turned off and there are no transitions. To compensate for this error, heuristics could be added that predict the likelihood the load is off given no incoming pulses and update the estimates accordingly. Alternately, the pulse sensors could be modified to report their state if they detect a sudden loss in harvestable energy income.

To quantify how Deltaflow performs in a use-case likely to be relevant to consumers, we explore how well Deltaflow estimates kilowatt-hours (kWh), the unit of energy by which consumers typically buy their electricity. Further, we evaluate how well Deltaflow is able to disaggregate the total kWh consumed into the percent contribution of each load. This information would be particularly useful for a consumer to determine where to direct effort when seeking energy efficiencies. Figure 4.18d shows this breakdown. The percent error of the estimated kWh values for loads (B), (D), (E) is under 6%. These loads tended to change independently of other loads, allowing Deltaflow to better determine their calibration functions and disaggregate the total to these loads. Loads (A) and (C), however, are likely used together and transitioned power states at similar times. This adversely affects Deltaflow's ability to successfully disambiguate two loads leading to the relatively high ($\approx 40\%$) errors in the kWh estimation. These errors would likely be reduced with a longer running experiment that exhibits a greater number of transitions that Deltaflow could use for calibration.

Also included in Figure 4.18d is a breakdown of each load based on the fraction of the total kWh each load consumes. While the errors in absolute kWh are still present in the breakdown, general trends about load consumption are still clear in the Deltaflow estimates: load (E) dominates the contributions of the other loads, loads (A), (B), and (C) have approximately the same contribution, and load (D) is an insignificant fraction of the total. These results show that Deltaflow supports critical comparative analyses of this nature that are essential for determining how best to invest constrained resources toward increasing efficiencies.

Figure 4.18 also demonstrates the effectiveness of integrating pulse meters that are based on different types of sensors and energy-harvesting front ends. Monitoring load (E) with only a lighting Monjolo node does not hinder Deltaflow's ability to disaggregate the load. Although we use only three different kind of pulse sensors in these experiments, we envision

many other kinds might be used in the future to monitor not only electricity, but water and gas as well.

4.6 Summary

With Monjolo, we show that for a certain class of energy sensing, an effective sensor can be derived from the energy-harvester itself. This enables very simple sensor hardware that drives down the cost of energy sensing and enables simple and straightforward deployments. Each meter uses a broadcast packet to reach a nearby gateway which can then calculate the packet rate. This broadcast communication technique demonstrates another aspect of the thesis statement that enables nodes to address energy intermittency and unpredictability. Monjolo also provides a template for adapting the method to other energy consumption points by choosing a different energy-harvester. We also show how incorporating a single aggregate meter into the system can lead to increased performance of the Monjolo sensors. For AC energy metering, Monjolo excels at the circuit level where it can be easily clipped around a wire in a circuit panel. However, because the AC Monjolo does not have access to the voltage channel, its accuracy suffers with non-unity load power factors. More extensive calibration on a per-load basis can compensate for this reduced accuracy, but that increases the deployment complexity and cost.

In Chapter 5, we build on the baseline AC Monjolo meter by embracing the aspects that ease deployability, namely the clip-on, non-contact metering frontend, while adding a virtualized voltage technique that provides the meter with the AC voltage waveform parameters. This allows for much better accuracy without runtime or load-specific calibration.

CHAPTER 5

Gemini: Accurate Non-Contact Power Metering by Virtualizing Voltage

Adding energy-harvesting to indoor sensors can enable maintenance-free operation and opportunities for greater scale than with battery or wall-powered sensor nodes. In the context of buildings, one important feature to measure is the electricity consumption of loads. Access to real-time power measurements can direct energy reduction efforts, identify equipment faults, and enable new smart building applications. A major challenge to deploying existing meters at building scales is the cost of both the meters themselves and their installation. Many existing meters require panel box replacement, require shutting off power, or require a calibration process. These factors significantly increase the barrier of entry to obtaining power meter data in existing buildings that are not submetered.

New power meter designs may be able to address some of these issues. As Chapter 4 notes, energy-harvesting in the context of AC electricity metering can enable current transformer (CT) based non-contact meters that easily clip around the wire to be measured. This form-factor can easily fit inside of existing panel boxes and is non-disruptive because it does not require a circuit panel shutdown. With the AC Monjolo design, however, measurement accuracy for loads with non unity power factors suffers because the Monjolo sensor only observes current and has no access to the voltage channel. True power measurement requires time synchronized sample-by-sample multiplication of the current and voltage waveforms, and without voltage data any phase shift between the two signals will result in potentially significant errors in the power calculation.

Gemini addresses this issue by enabling an energy-harvesting current-only sensor to obtain a representation of the voltage channel. The sensor can then reconstruct the voltage signal locally and perform a true power measurement. This increases the accuracy of the power measurements while preserving the clip-on, non-contact deployability benefits of the Monjolo design. Also, because Gemini uses a CT for both harvesting and current measurement, it can directly measure the current waveform and therefore does not require

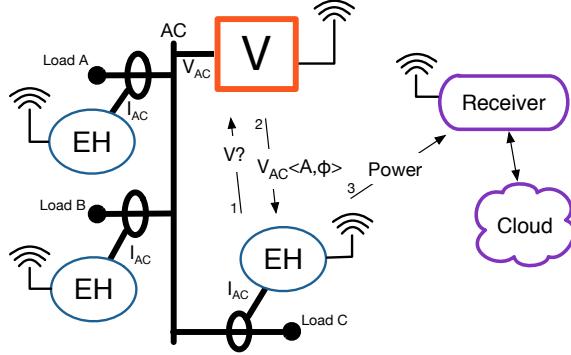


Figure 5.1: Gemini overview. A single voltage monitor (V) continuously monitors the voltage channel of the AC circuit. Energy-harvesting power meters (EH) use current transformers to meter each load. To calculate true power, the EH meter requests the voltage parameters from the voltage monitor, samples the local current waveform, synthesizes the voltage waveform, computes power, and transmits the result to a receiver which forwards the data to the cloud. Maintaining timing across domains is key to the success of this approach.

runtime calibration after the meter is installed.

The key to achieving this functionality is virtualizing the voltage channel, wirelessly transmitting it, and ensuring it is correctly phase-aligned with the current channel. This enables a non-contact meter which cannot measure voltage directly to still calculate true power. The remainder of this chapter describes the core components of the system, how current and voltage are synchronized on the energy-harvesting node, and the resulting accuracy of the system.

5.1 Gemini: Decoupling Voltage and Current Measurement

In contrast to traditionally integrated power meters, Gemini’s design employs distributed voltage and energy-harvesting current monitors, connected wirelessly, to cooperatively calculate power. Distributing the current and voltage acquisition in this way greatly simplifies the installation of power meters as the two signals are often not easily accessible in the same location.

Figure 5.1 shows the primary components of the system, the voltage monitor (V) and the energy-harvesting power meter (EH), which work in tandem to measure power. The voltage monitor continuously monitors the voltage channel and provides a virtualized representation of the voltage waveform to the energy-harvesting nodes on-demand. Each energy-harvesting node measures its local current waveform and multiplies that by the voltage waveform

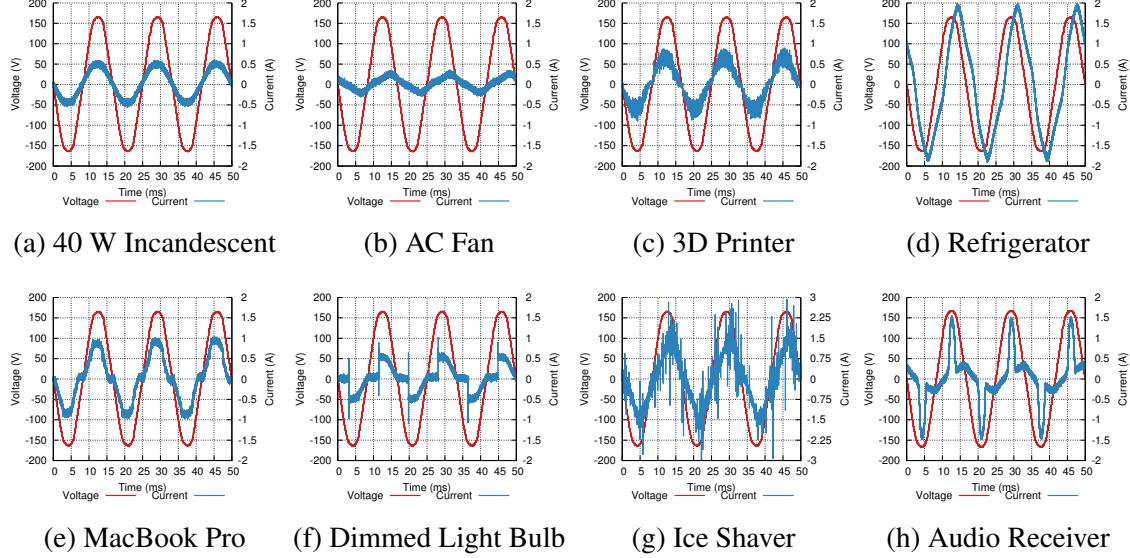


Figure 5.2: Voltage and current waveforms of eight loads. The current waveforms vary wildly, but the voltage waveforms are consistently sinusoidal and can be represented by one or just a few Fourier coefficients. The sinusoidal nature of the voltage waveform makes it a better candidate to be virtualized, thus supporting our architectural choice to virtualize voltage.

obtained from the voltage monitor to calculate true power. The energy-harvesting node then transmits the power data to the cloud via a receiver.

5.1.1 Virtualizing Voltage

To realize this split metering architecture, and to successfully calculate power with Gemini, either the current measuring device must learn about the voltage waveform from the voltage monitor or vice-versa. In our design, we transmit information about the voltage waveform to the current sensor to calculate power. The reason for this is that the voltage channel is more easily parametrized than the current channel. That is, voltage can be modeled with phase information and one or a few Fourier coefficients. Figure 5.2 provides some intuition for this distinction by showing the voltage and current waveforms for several loads. The voltage waveform is consistently sinusoidal while the current exhibits significant harmonic distortion. Because sending raw waveform samples is infeasible due to limited packet payloads and energy resources, virtualizing the voltage channel and transmitting its smaller parameter set is the preferred approach.

Another tenet of the Gemini design is that only a small number of voltage monitoring sensors are required to support a much larger number of current sensing power meters. This

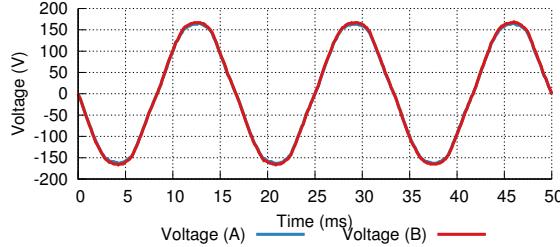


Figure 5.3: Comparison of two voltage waveforms at physically disparate points in the same electrical system. Measuring the voltage at two different loads with different power draws on different circuits tapped off of the same leg of a transformer results in voltage waveforms that are nearly identical. This suggests that measuring voltage at a single watchpoint is sufficient for calculating the power draw of multiple loads.

holds if the local voltage measurement can be replaced with a voltage measurement obtained elsewhere in the load tree. To verify this, we sample the voltage waveform at two points on the same leg of a transformer from two different circuits behind two different electrical panels with a 300 W load on one point and a 1.5 kW load on the other. The resulting voltage waveforms are shown in Figure 5.3. The waveforms show no phase distortion and a slight amplitude variation, suggesting that remotely sensing voltage is viable. One other important factor related to voltage measurement is that the phase of the voltage channel local to the power meter (which is not measured) must match the phase of the virtualized voltage channel used for power calculation.

5.1.2 Energy-Harvesting

To meet the ease-of-installation constraint, the current sensor must be able to sample the current waveform while remaining non-contact. To power the sensor, we are unable to use an AC mains based AC-DC converter because that requires tapping into the circuit, and we do not use batteries because of their size and the maintenance burden of replacing them. Instead, we use an energy-harvesting power supply that uses the output of a current transformer to charge a capacitor. This approach is largely borrowed from the Monjolo meter [21].

This power supply is necessarily intermittent; harvesting from a current transformer does not supply sufficient power to run a sensor node continuously. It is also unpredictable, as variable-power loads will alter the operation of the harvester. This causes the power supply to limit the sample rate of the meter based on how quickly and often it can recharge.

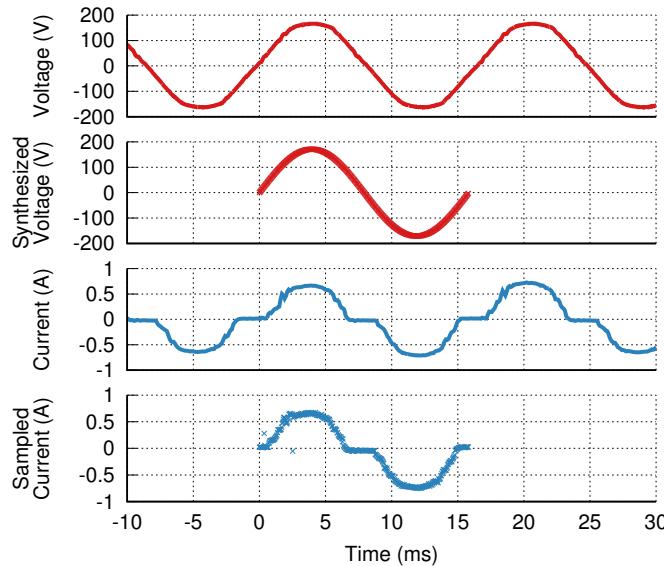


Figure 5.4: Voltage and current waveforms. Ground truth current and voltage waveforms are shown above their synthesized/sampled counterparts. The voltage waveform is synthesized from the parameters sent by the voltage monitor and the current waveform is sampled from the current transformer monitoring the load. The discrete waveforms are multiplied together, point-by-point, to compute the load’s power draw.

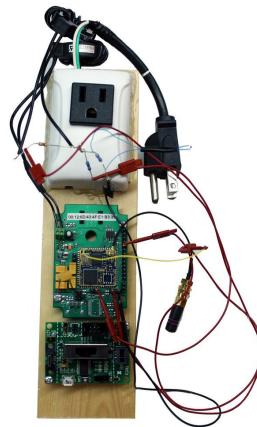
5.1.3 Power Meter System Operation

Power is calculated by the energy-harvesting current sensor. The sensor begins by requesting information about the voltage channel from an always-on voltage monitor. After sending the request, it samples the current waveform until the voltage monitor responds. It then synthesizes a discrete representation of the voltage channel properly phase-aligned to the sampled current waveform, as shown in Figure 5.4. It multiplies the voltage and current samples together and averages them to find the instantaneous power draw of the load. Harvested energy permitting, the current sensor sends the power measurement to a receiver which timestamps and forwards the data to the cloud, otherwise the sample is stored for future transmission.

The receiver is an always-on listener that may be a dedicated gateway for the low-power wireless network or it may be a mains-powered node, such as a voltage monitor, which can forward the packet on behalf of the meter.



(a) Voltage Monitor



(b) Energy-Harvesting Node

Figure 5.5: Sensing hardware prototypes. The voltage monitor is a plug-load form-factor meter that can plug in to an outlet near a circuit panel box to retrieve the local voltage signal. The energy-harvesting node is prototyped with a solid-core current transformer but the hardware design extends to a split-core CT without modification.

5.2 Implementation Details

This section describes the hardware components that comprise the voltage and current sensors, as well as specifics of the software algorithms running on the meters.

5.2.1 Voltage Monitor

The voltage monitor continuously records and can on-demand report parameters of its local AC voltage signal. Our implementation uses a TI CC2538 SoC [130] for computation and wireless communication, and an Analog Devices ADE7753 [2] for AC voltage waveform analysis.

The voltage monitor stores the peak voltage of the most recent AC cycle and locally timestamps the most recent rising zero-crossing of the voltage waveform. This allows the voltage monitor to make a simple parametrization with the phase and magnitude of the voltage waveform for dissemination to interested clients. The ADE7753 also provides access to digital samples of the entire voltage waveform allowing for more sophisticated parametrizing schemes to be used in the future.

The voltage monitor provides the parametrized voltage information as a service. An interested client sends a wireless packet to the voltage monitor requesting the parametrized values. The monitor immediately replies with the magnitude and phase of the voltage signal. Because we do not assume a synchronized timebase between the client and monitor, the

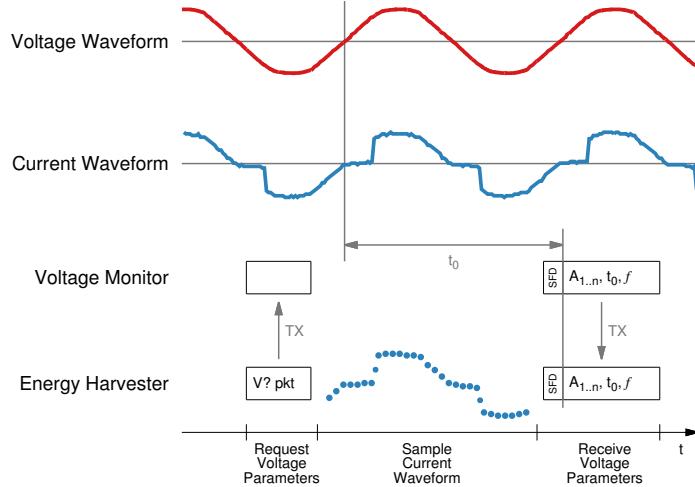


Figure 5.6: Transmitting AC voltage phase information. After the energy-harvesting node requests the voltage parameters, the voltage monitor responds with parameters of the voltage waveform such as Fourier coefficients, phase, and frequency. Phase is encoded as the time offset (t_0) between the response packet's SFD signal and the voltage signal's most recently rising zero-crossing. The energy-harvesting node uses its local timestamp of the SFD signal to phase align the current waveform samples to the voltage channel without requiring synchronized clocks.

phase information is transmitted as a time offset between the start of frame delimiter (SFD) signal of the outgoing reply packet and the most recent rising zero-crossing of the voltage waveform, as shown in Figure 5.6. The transmitter and receiver establish a common time reference by assuming the SFD signal asserts simultaneously on both devices.

The receiver uses its local SFD timestamp and the rising zero-crossing time offset provided by the voltage monitor to align the voltage and current waveforms. Any delays in packet transmission and processing are ignored because all offsets are calculated from the synchronized SFD signal.

5.2.2 Energy-Harvesting Current Sensor and Power Meter

The energy-harvesting device (Figure 5.7) is responsible for sampling the AC current waveform and calculating the instantaneous power draw of the load or circuit.

5.2.2.1 Power Supply and Coil Circuit

The energy-harvesting power supply is based on the Monjolo power meter's design [21]. The harvesting source is the AC output of a current transformer that is wrapped with 10 windings of the AC phase line. This AC signal is fed to a Linear Technology LTC3588 [76].

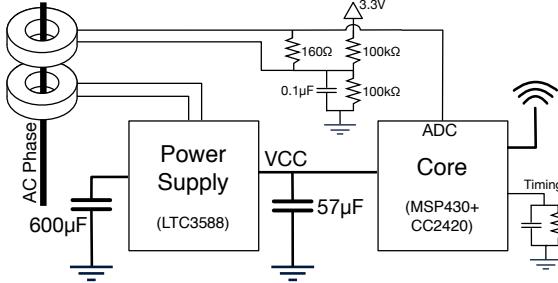


Figure 5.7: Energy-harvesting power supply and measurement circuit. One current transformer is used for harvesting and the other for measurement. Future implementations could multiplex one current transformer for both operations.

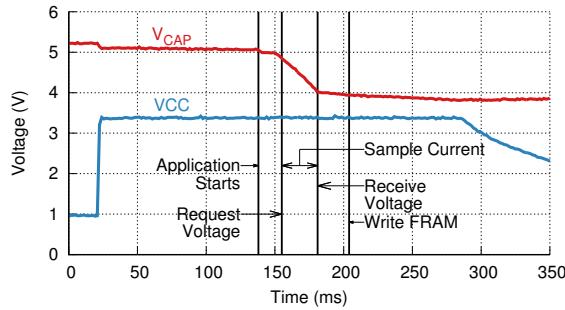


Figure 5.8: Energy-harvesting operation. V_{CAP} is the voltage on the $600 \mu\text{F}$ capacitor bank and V_{CC} is the 3.3 V supply voltage.

The LTC3588 is configured to charge a $600 \mu\text{F}$ capacitor bank to 5.25 V and then enable a 3.3 V output regulator that powers the main system. When the main storage capacitors discharge to below 3.8 V , the regulator is disabled, the storage capacitors begin recharging, and the system runs on the charge stored in the $57 \mu\text{F}$ output capacitor until the voltage drops too low and the system is forced off. One iteration of this cycle is shown in Figure 5.8.

In our implementation, a second current transformer is used for sampling the current waveform. The output of this current transformer is connected across a burden resistor and fed first into a biasing circuit and then into an ADC (shown in the top of Figure 5.7). The burden resistor affects the scaling factor between the actual current waveform and the current transformer output and increasing the burden resistor increases the peak-to-peak output voltage. To ensure sufficient resolution of the voltage signal from small primary loads we use a 160Ω resistor. Selecting a smaller burden resistor is necessary for larger loads as the ADC saturates. The biasing circuit is required to shift the output of the current transformer from being centered around 0 V to being centered around 1.65 V . This ensures that all output waveforms up to 3.3 V peak-to-peak can be successfully sampled by the ADC.

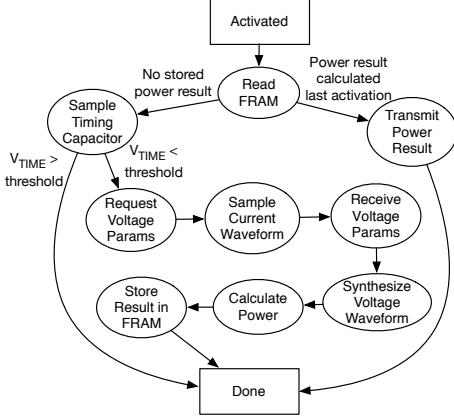


Figure 5.9: State machine for the energy-harvesting power meter. Nodes use the timing capacitor to rate-limit sampling and determine when to compute power.

5.2.2.2 Computation Core

The energy-harvesting node is based on the Epic sensor node module [32] which provides a microprocessor and radio. Our design adds a 4 Kb Cypress FM25L04B FRAM module [20] for persistent storage between activations, and a parallel resistor-capacitor circuit for roughly timing the period of activations to rate-limit activations for high-power loads, as in the Monjolo design [21].

5.2.2.3 Operation

The energy-harvesting node's operation follows the state machine shown in Figure 5.9. Upon activation, it reads the FRAM to determine if a power measurement was calculated on the previous activation and needs to be transmitted. If so, it transmits the power value to a receiver, transitions to the done state, and waits for the power supply to exhaust. If there is no stored power result, the node samples the voltage of the timing capacitor to determine the approximate elapsed time since the last activation. If the capacitor voltage is above a threshold, the node simply sits idle until the power supply capacitors discharge and the node is forced off again. If the voltage is below the threshold, the node continues operating. This rate-limiting prevents a node from saturating the wireless channel by transmitting packets too frequently. We size the timing resistor and capacitor to limit power calculations to no more than once every five seconds.

When an energy-harvesting node determines that it should calculate power, it requests the voltage parameters from a voltage monitor. After transmitting the request, it begins sampling the current waveform every 40 μ s while awaiting a response from the voltage monitor. The response from the voltage monitor arrives approximately 16 milliseconds after

the request, which allows the energy-harvesting node to sample a full period of the current waveform. Upon packet reception, the energy-harvesting node ceases sampling and begins calculating the average power over the prior AC period.

The response packet from the voltage monitor contains the offset from the SFD signal of the response to the last rising zero-crossing of the voltage waveform in terms of the local clock on the voltage monitor. To convert that offset to the local clock of the energy-harvesting node, the node computes the number of current waveform samples since the voltage zero-crossing. It then uses a precomputed array of sine wave samples to multiply the current waveform samples with the synthetic voltage waveform to obtain the power waveform. The power waveform samples are then averaged to calculate the average power draw. Due to the limited energy budget from the energy-harvesting power supply, the power calculation is stored in FRAM and not transmitted immediately. On the next activation the result is transmitted to the cloud. This imposes a small time delay for the power measurement but allows the system to operate within its energy budget.

5.3 Gemini Evaluation

Evaluating the Gemini power metering system focuses on the accuracy of the meter, but also covers how timing errors and current-voltage mismatch errors affect the system.

5.3.1 Accuracy: 8.7% Average Error, 2.2 W Absolute Error

To evaluate Gemini’s performance as a power meter, we sweep over a range of resistive loads (i.e. loads with unity power factor) and measure how well Gemini tracks ground truth (as measured with a PLM-1LP [103]), as shown in Figure 5.10. For resistive loads below 5 W the energy-harvesting meter is unable to accumulate enough energy to activate. Between 5 and 95 W, the estimated power tracks ground truth well with an average absolute error of 0.61 W and average percent error of 1.2%. Above 95 W the peak-to-peak output of the current transformer exceeds the 3.3 V ADC reference and clips, causing greater error. Reducing the value of the burden resistor would allow for these larger loads to be metered.

While Gemini performs well for purely resistive loads (such as incandescent light bulbs), many loads do not have a unity power factor and a purely sinusoidal current waveform. To evaluate Gemini with such loads, we meter four AC loads with both Gemini and a ground truth power meter. The experimental traces are displayed in Figure 5.11.

3D Printer. The printer starts in an idle state before heating the extruder head between minutes four and six, then actively prints before returning to idle at minute 24. Gemini tracks

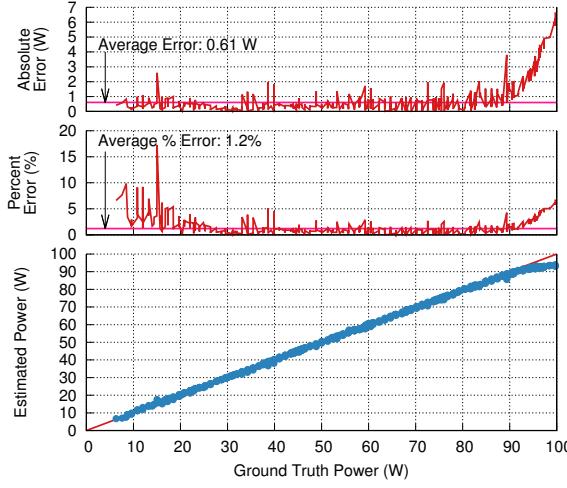


Figure 5.10: Accuracy over a range of resistive loads. We meter a range of loads with a unity power factor using Gemini. The x-axis shows the ground truth power draw of the load. In the bottom graph, Gemini’s estimate is overlaid on the line representing an ideal device. The middle graph shows the percent error of the estimate, and the top graph shows the absolute error of the estimate. The average percent error is 1.2% and the average absolute error is 0.61 W. The error grows when the load is greater than 95 W due to ADC clipping of the current waveform. These results demonstrate that Gemini is a viable power meter for resistive loads.

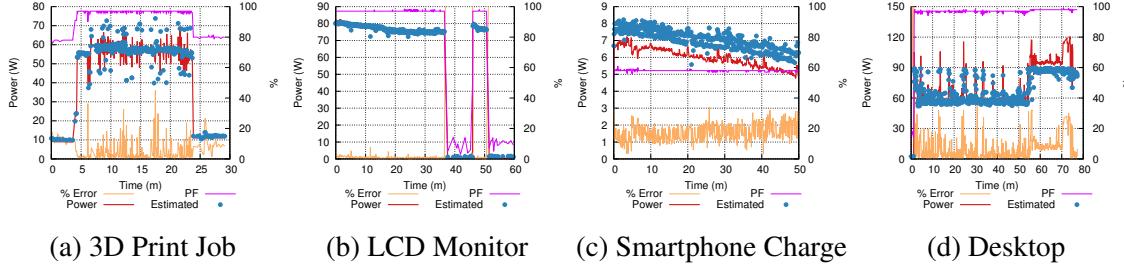


Figure 5.11: Power metering accuracy for four real-world loads. We meter four loads with a ground truth meter and Gemini. Gemini is able to mirror ground truth well with the highest error appearing when loads either rapidly change power draw or when loads draw very little power (e.g. drawing ~ 1 W). For instance, because (c) has a low power draw, the average percent error is relatively high at 17.2%, but the average absolute error is only 1.0 W. The highest average absolute error is 4.9 W for (d), mostly due to the current waveform clipping after the 55 minute mark. These test cases demonstrate Gemini’s viability as a power meter for real-world loads that operate within Gemini’s (configurable) operating range.

these transitions well, even being able to sample the increasing power transition between idle and heating. Rapid swings in the power draw between 50 and 70 W cause spikes in Gemini’s percent error as the energy-harvesting power supply limits the sample rate of Gemini causing spikes to be reported late or missed all together. Overall, however, Gemini

reports an average percent error of 4.9% and an average absolute error of 2.2 W.

24” LCD Monitor. The monitor is cycled between the on, off, on, and standby modes. During the on mode, Gemini is quite accurate with an average error of 0.85% and average absolute error of 0.67 W. In the standby and off modes Gemini overestimates, but the overall total average error is still only 9.7% and 0.68 W. A key result of this test is that Gemini continues to operate even when the monitor is drawing only 0.5 W in the low power states.

Charging Smartphone. In this test the power factor is roughly constant at 0.58. Again Gemini tracks the slowly decreasing power draw well. The average percent error is relatively high at 17.2%, but because phone chargers draw little power, the overall average absolute error is only 1.0 W.

Desktop Computer. The final load is a desktop computer that starts in sleep mode, enters normal operation for 55 minutes and then plays video for 20 minutes. At the beginning, the energy-harvesting power meter is able to activate even as the load draws only 1.1 W. During normal operation, Gemini performs well, albeit with the same spikes in error as the load power changes rapidly like in the 3D printer case. In the final phase of the test, when the desktop plays a video, Gemini underestimates slightly. This is, as in the fully resistive case, due to clipping of the current waveform when sampling it with the ADC. Even with the clipping error, however, the overall average error is 6.3% and 4.9 W.

For a rough comparison of Gemini’s error to commercially available meters, we compare the reported specifications of several of them with the empirical data collected using Gemini. The Watts Up? .Net [142] claims a 1.5% error, the Kill-A-Watt [68] reports a 0.2% error, and the PLM-1LP [103], our ground-truth meter, cites a 0.5% error.

5.3.2 Misaligned Current and Voltage Errors

The Gemini design hinges on the accurate transfer of timing information about the phase of the voltage waveform from the voltage monitor to the energy-harvesting meter. Timing errors due to clock offsets, clock drift, or queuing delays cause phase errors between the synthesized voltage and sampled current waveforms on the energy-harvesting meter, resulting in power calculation errors. To characterize the effect of such errors, we introduce artificial timing delays by shifting the voltage waveform when calculating the average power of the eight loads in Figure 5.2. Figure 5.12 shows power estimation error as a function of the timing error. In the extreme, with $\pm 180^\circ$ phase error, or 8.33 ms timing error, we observe at least 200% error in the power calculation. This demonstrates that minimizing timing error is critical to Gemini’s operation.

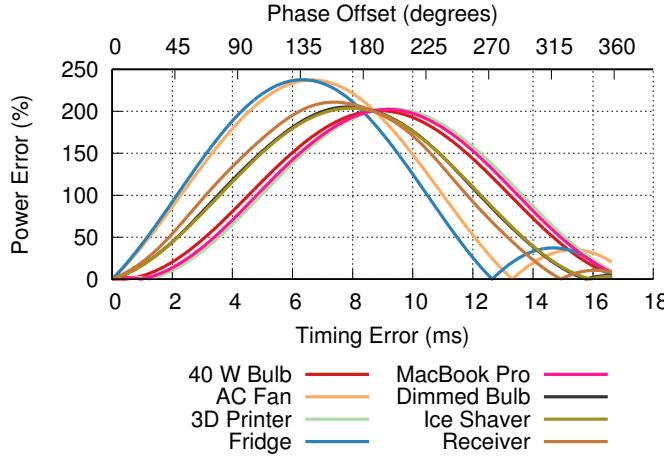


Figure 5.12: Effect of timing offsets when calculating power. Even 1 ms of phase error can result in 50% measurement error.

5.3.3 Correctly Matching Current and Voltage

Buildings typically have two- or three-phase power systems, meaning circuits on different legs of the transformer will have voltage phase offsets between them. In order to correctly calculate power, Gemini must match the correct voltage phase to the load being metered. Loads with a highly displacement based power factor could make selecting the correct voltage channel ambiguous. In practice, however, two factors make this type of error unlikely. First, many loads have switching power supplies which exhibit harmonic distortion to their current waveforms but little displacement. Second, higher power loads are required to include power factor correction which manipulates the current waveform to appear more like a resistive load.

5.3.4 Design Limitation

The Gemini design demonstrates that with proper attention to detail, simple versions of complex protocols (e.g. time synchronization) are possible in small energy-harvesting systems. This capability is particularly advantageous for circuit level power metering because the non-contact nature of the sensor eases installation complexity. However, the design still has a number of drawbacks. First, the meter relies on a wireless link between the current and voltage sensors. If this link is disrupted due to interference, or the voltage monitor is placed too far away from the current sensor, Gemini fails to successfully take measurements. Additionally, Gemini leverages two current transformers: one for the harvesting and one for current sensing. This adds cost and size, making deploying the sensors more difficult.

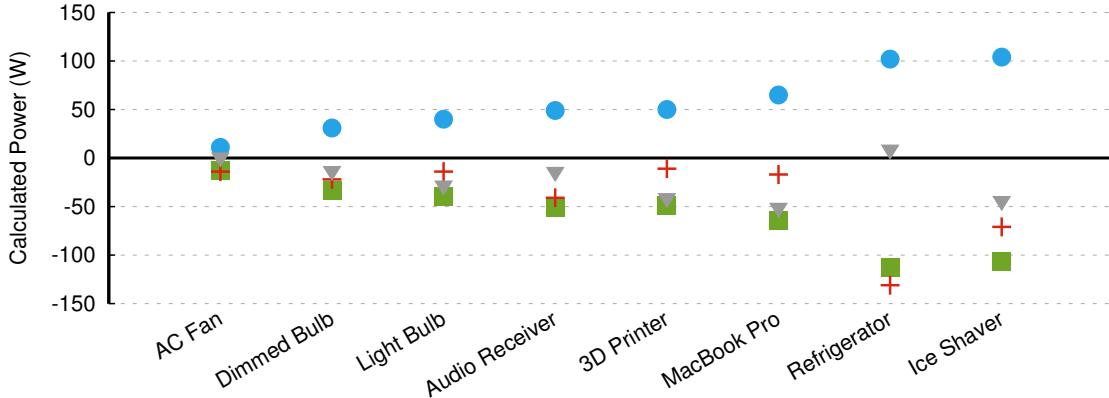


Figure 5.13: Power calculations with different phase errors. The instantaneous power for the eight loads in Figure 5.2 is calculated with the current and voltage waveforms correctly aligned and with the current waveform shifted as would happen if the incorrect voltage leg of a transformer was used. In all eight cases the correct result is the greatest positive result, suggesting that using `max()` as a heuristic is often sufficient for correctly selecting the voltage channel

5.4 Summary

Existing power meters are limited in their ability to solve the critical problem of electrical submetering in buildings. Installation challenges, maintenance headaches, and metering accuracy hamper the widespread adoption of submetering. Gemini addresses these challenges by offering clip-on installation, energy-harvesting operation, and accurate power measurement. Gemini combines virtual sensors and energy-harvesting meters to yield a novel design point in the metering space. Gemini also addresses the drawbacks with prior approaches by decoupling and distributing the AC voltage and current measurement acquisitions, and recombining them over a low-bandwidth, time-synchronized, wireless channel, to offer non-invasive real, reactive, and apparent power metering. By strategically avoiding more complex time synchronization schemes and instead leveraging the synchronization possible with a single wireless packet, Gemini is able to provide accurate results on an energy-harvesting energy budget. This supports the thesis statement by showing one example of deferring energy-intensive operations to enable an energy-harvesting sensor. Gemini represents a significant advancement from previous designs that are difficult to install, require periodic battery maintenance, or cannot calculate true power.

However, Gemini does still require a two part installation (the current meters and the voltage monitor), and wireless contention in rooms with many circuits could cause system failures in real-world scenarios. Further, correctly matching voltage and current phases could present a significant deployment challenge. The next chapter on the Triumvi meter

attempts to address these concerns. Triumvi is a truly standalone meter that leverages the deployability benefits of the Gemini and Monjolo energy-harvesting meters.

CHAPTER 6

Triumvi: A Truly Deployable Energy-Harvesting Power Meter

The Gemini design demonstrates that by fine-tuning the relationship between sensor operation and physical properties of the underlying phenomenon, energy-harvesting sensors can be accurate and reliable, and even have better deployability properties than other designs. With AC power metering specifically, the total cost of obtaining the data is often dominated by installation costs and not by the meters themselves. Many systems require invasive installations, often including interrupting the power, wiring to and possibly adding new circuit breakers, and even adding a new panel box. This chapter introduces Triumvi, a new approach that is a completely standalone meter that harvests energy to power itself, meters current and detects voltage phase locally, calculates true power, and encrypts the data before wirelessly transmitting it. This enables Triumvi to simply clip around wires in a panel box, allowing a circuit to be metered in less than 30 seconds.

The Triumvi design is inspired by previous investigations into the power metering design space, but is the first meter to address the challenges of integrating the subsystems while improving on every facet from previous designs. Compared to previous power meters, Triumvi is smaller, requires fewer capacitors, requires only a single current transformer, does not require runtime calibration, adds a switching circuit for repurposing the current transformer, provides data confidentiality, and ensures reliable accuracy, all while eliminating the limitations of previous designs that prevent promising systems from being deployable.

The fully non-contact, self-powered Triumvi device can operate as a long-term instrument inside of a circuit panel. Triumvi’s simple clip-on installation, and its expansion ports, however, allow Triumvi to extend to other deployment paradigms as well. For instance, a group of Triumvi sensors could perform “energy audits” to collect short-term data without the expense and overhead of installing a built-in panel meter, or to meter key circuits of interest. Triumvi also includes an optional expansion pack with a battery to enable short- or medium-term deployments while mitigating the variability inherent to energy-harvesting.

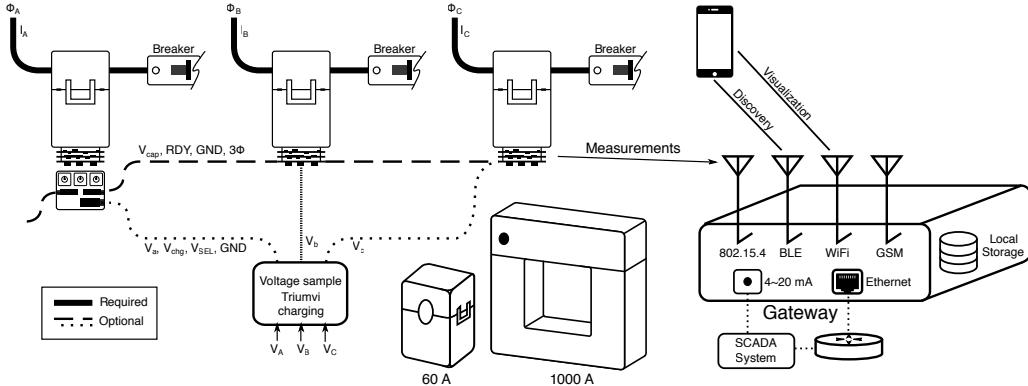


Figure 6.1: Triumvi system overview. Meters clip to wires exiting circuit breakers and harvest energy to power themselves, detect voltage phase, sample AC current, and wirelessly transmit true power measurements. Optionally, they share charge, measure synchronously, or use an external voltage signal or power source. The nearby gateway collects the measurements and publishes them to local users, existing BMS/SCADA systems, local storage, or cloud services. This metering system, with its easy-to-install sensors that support flexible metering options, leverages the benefits of energy-harvesting operation to make effective circuit-level energy meters.

The expansion ports also allow Triumvi to adapt how it senses, for example by converting the meter to a three-phase meter. By extending the Triumvi design to a metering platform, Triumvi enables a high degree of deployment versatility.

6.1 Triumvi: Standalone Meter with Integrated IV Sensing

The accuracy and deployability goals of Triumvi inform the three main requirements of the meter:

1. That it be an accurate true power meter.
2. That it be non-contact.
3. That it be standalone.

Each requirement has a particular effect on the design. The first, accuracy, requires sample-by-sample multiplication of voltage and current waveforms to calculate true power. Therefore, Triumvi must have access to both the current and voltage waveforms. In order to be non-contact and easily clipped around a circuit inside of a panel box, Triumvi must not use a traditional AC-DC power supply which requires direct contact to the AC mains. And to be a

standalone device it must be capable of calculating power on its own to reduce complexity and cost.

The result is a design consisting of three main components illustrated in Figure 6.2: an energy-harvesting power supply based on a current transformer, a circuit for repurposing the current transformer for current measurement, and a capacitive coupling circuit for measuring voltage. Energy-harvesting allows the meter to be non-contact, while the current and voltage sampling circuits allow it to be accurate and standalone.

6.1.1 Energy-Harvesting Power Supply

The energy-harvesting power supply is responsible for powering the sensing and communication cores of the device. Harvesting is based on rectifying the output of a current transformer (CT) that is clipped around the AC phase line running to a breaker in a circuit panel. When the circuit is drawing current the electromagnetic field it generates induces a current in the transformer, which is rectified and used to charge a bank of capacitors. This design is extended from previous work [13, 21] that uses current transformers for energy-harvesting.

By harvesting energy to power itself, Triumvi avoids issues with other power sources. First, it removes the need for the meters to be wired to a power supply. This supply may be a breaker, which likely adds the expense of an electrician to install it, or an external power supply plugged into a nearby outlet, which may not be available. Alternatively, the meters could be battery powered, which permits the standalone form factor, but imposes a lifetime constraint and maintenance requirement.

In certain cases, however, these alternative power sources may be advantageous. The main drawback to energy-harvesting from the AC lines is that harvestable energy is proportional to load current, and as a result the maximum sample rate of the meter is dictated by the load of the circuit. Short-term deployments may be able to avoid this limitation by making use of a battery (and the provided reliable sampling interval) as long as the deployment is within the battery's lifetime. Deployments with stricter sampling requirements needs may be willing to forgo some of the deployment simplicity of the base Triumvi system by wiring a dedicated power source to each meter to ensure a consistent sample rate. Therefore, we ensure the Triumvi platform includes support for these options. External power from a battery or dedicated supply can be provided through the expansion header on the Triumvi board.

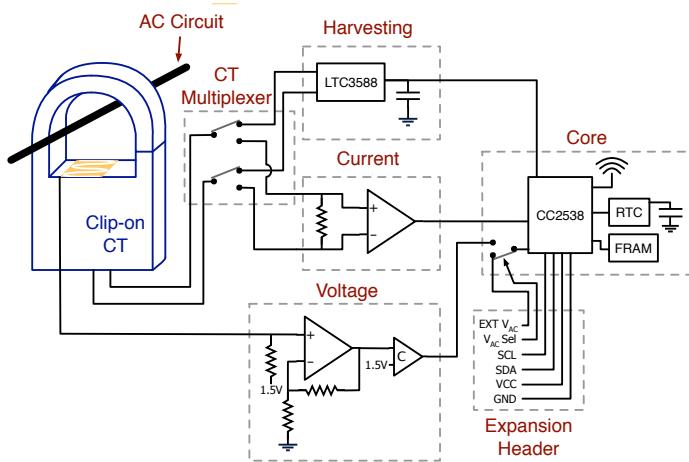


Figure 6.2: Triumvi system block diagram. The three main system components (harvesting, current, and voltage) use inputs from the split-core current transformer and voltage sense conductor to provide the processing core power, and scaled current and voltage signals, to compute AC power. The core’s real-time clock (RTC) and low-energy memory (FRAM) can time-stamp and store data locally when the network is unavailable. An expansion header allows external voltages to be used (for the voltage channel signal or to power the meter, if higher accuracy or resolution is required).

6.1.2 Multiplexing the Current Transformer

Triumvi uses the same current transformer from the power supply circuit to also measure current. Triumvi is able to selectively enable a burden resistor across the CT inputs to facilitate accurate point-by-point sampling of the current channel. The key to enabling the burden resistor while maximizing harvesting capability is careful design of the circuit that switches between the two modes. Simply inserting FETs between the CT and burden resistor is insufficient as the CT output voltage can be too high to reliably enable or disable the FETs, causing leakage that prevents harvesting or affects current measurements. Instead, our design is able to completely disable the burden resistor when harvesting.

When measuring current, the output of the CT with the burden resistor is fed into an instrumentation amplifier. The amplifier biases the CT output around the midpoint of the ADC that is sampling the signal. The microcontroller can now measure current simply by sampling the ADC.

Triumvi’s ability to multiplex the current transformer is critical for the meter to achieve its goals, as no other options for acquiring the current signal are sufficient. Using a second CT—one for harvesting and one for sampling—would make the device too large to easily install, and add cost. Using a shunt resistor requires direct contact to the circuit, and using a sensor, like a hall effect sensor, to detect the electromagnetic field would reduce accuracy

and impose a calibration step, both of which are unacceptable for a deployable power meter. Therefore, Triumvi must implement the multiplexing circuit, even with the slight interference the additional circuitry adds to the CT signal.

6.1.3 Voltage Measurement

Real-time access to the AC voltage channel is critical for power meters to be able to accurately meter loads with non-unity power factors. Triumvi uses a capacitive voltage sensing technique where a flat conductor is placed near the wire to be metered runs. This forms a capacitor between the sensing conductor and the wire.

Lorek et al. [77] used a similar design for monitoring voltage on the outside of the breaker itself, and demonstrated that this technique can accurately measure voltage with about 1% error. However, this measurement technique does require calibration, as the geometry of the breaker, the layout of the sensing device, and the installation placement all affect the measurement. In Triumvi’s case, the gauge of the wire and the orientation of the wire running through the CT affect the measurements. To remove the need to perform runtime calibration of the voltage input, we extract only phase information from the voltage signal. The phase of the voltage signal is less affected by orientation than the amplitude.

To extract phase, the raw signal from the capacitive voltage monitor is amplified and biased around $VCC/2$. This is fed into a comparator which outputs a square wave where each transition marks a zero crossing of the AC voltage signal. This entire voltage monitoring circuit is power gated until voltage measurements are needed.

From the phase information, Triumvi assumes a sinusoidal voltage signal at the nominal voltage amplitude. Prior work has shown that the voltage signal remains sinusoidal even with the presence of large loads [13, 116]. Amplitude, however, does vary over the course of a day, causing some error in the resulting power calculation. While present, this error is substantially less than the error resulting from not having voltage phase information. We predefine the nominal voltage of the circuit to be measured when programming the Triumvi device.

As with the current measurement, this method of acquiring voltage is the only viable option for Triumvi. The traditional method, direct contact, is not feasible given deployment constraints, and virtualization of the signal (as in Gemini [13]) imposes too many failure points by distributing voltage acquisition. The capacitive coupling technique, while itself limited, is the best available option.

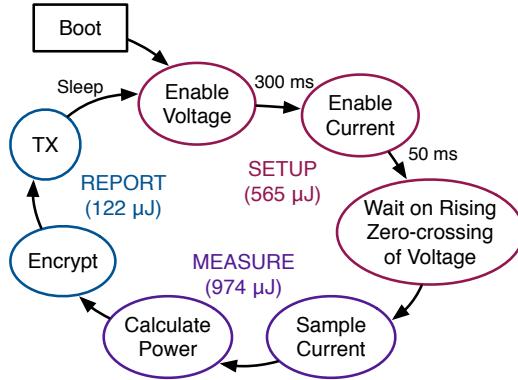


Figure 6.3: Triumvi state machine. Once Triumvi has booted due to sufficient energy, it iterates through setup, measure, and reporting phases before waiting for its energy store to replenish.

6.1.4 System Operation

Triumvi follows the state machine shown in Figure 6.3. The device starts in harvesting mode and charges the storage capacitor until it has accumulated enough energy to perform a measurement.

When this threshold has been met Triumvi first enables the voltage measurement circuit and allows it to stabilize. Next, the system enables the current measurement circuit—which disables harvesting—and again waits for stabilization. With the voltage and current measurement circuits enabled, Triumvi begins sampling the current waveform after the next rising zero-crossing of the voltage signal, which synchronizes the two signals. After sampling for one AC cycle, Triumvi disables the voltage and current measurement circuits and Triumvi begins computing power. When the result is ready, it encrypts the measurement and wirelessly transmits the payload.

After completing the entire measurement, the microcontroller enters sleep mode and sets a timer to wake itself for the next measurement. If there is sufficient harvesting during this period the sample will occur normally, otherwise the microcontroller will be powered off and will reboot when energy is available.

Triumvi includes two extensions to the base system operation. First, Triumvi includes onboard non-volatile storage and can store measurements rather than transmitting after every sample. This allows Triumvi to collect samples and transmit less frequently. Locally storing is a less energy-intensive operation than transmitting, allowing Triumvi to increase its sample rate, at the cost of less real-time operation. Further, Triumvi can maximize its sample rate by only locally storing measurements, which can be downloaded later.

Additionally, Triumvi includes a real-time clock (RTC). This allows Triumvi to not only

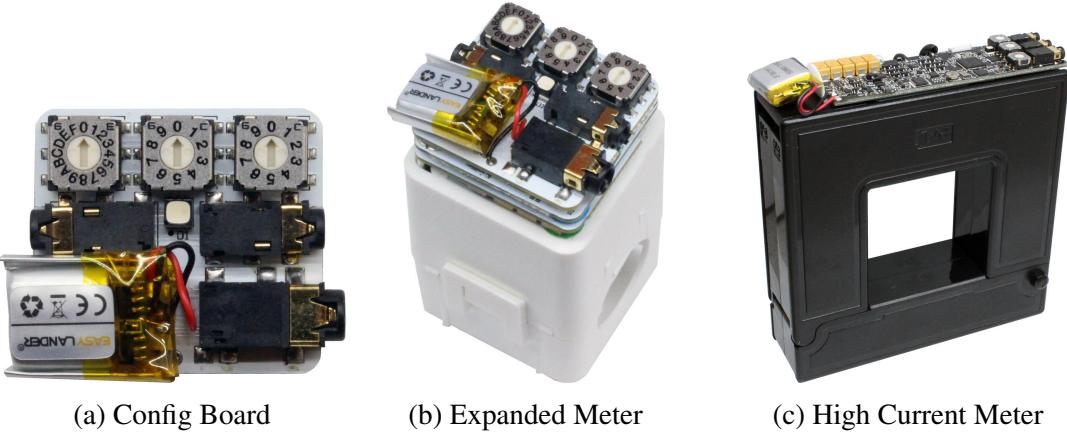


Figure 6.4: Triumvi configuration expansion board and high current version. This configuration attachment contains three knobs for setting which panel the meter is in, and which circuit it is metering. It also includes an optional battery, a recharging circuit, and 2.5 mm and 3.5 mm jacks for charge sharing, synchronous measurements, external power, and external voltage waveform signals. The large load meter can measure 20 A to 1000 A loads.

store power measurements, but also energy accumulation. The RTC is also powered by a supercapacitor, enabling it to keep time even if Triumvi is unable to harvest, which preserves the accuracy of the total accumulated energy. Calculating energy allows Triumvi to provide two additional services. First, recording and transmitting energy values helps mitigate issues resulting from dropped packet transmissions. While a dropped packet causes a loss in the resolution of the power metering data, transmitting accumulated energy in addition allows the backend system to determine the average power when packets are missed. The second service Triumvi can provide is interval metering where Triumvi can record total energy consumed by the circuit at different intervals during the day. For example, Triumvi can report energy consumption for every half-hour throughout the day.

6.1.5 Three Phase Loads

Many loads in a building are not single-phase loads. For instance, electric stoves, HVAC equipment, and industrial motors are all split-phase or three phase loads. Accurately metering multiple phase loads requires taking simultaneous measurements across all three phases of the load. The system can meter three phase loads by clipping one meter on each phase, but because of the unpredictable energy-harvesting the samples from each phase are unlikely to be synchronized. To enable synchronized sampling, two or more Triumvi meters can be wired together into a Wired-NOR configuration. While the meters are charging, they each use their storage capacitors to pull the shared line high. When each meter is charged

and ready to take a measurement, it pulls the line low through a diode. When all connected meters have done this, all of the nodes will see the line go low and proceed with their power measurement in a synchronized fashion.

6.1.6 Large Loads

Metering and monitoring large industrial equipment (in the range of 20 to 1000 A), such as arc furnaces, water pumps, and large motors, can be used for peak-demand shaving and equipment fault detection, but the same meter cannot be used for residential-scale loads and these large consumers. To demonstrate that the metering approach can scale to larger loads, we develop a Triumvi meter that can support up to 1000 Amps (Figure 6.4c). The meter design scales while only requiring changes to the biasing resistors for the amplifier circuits. The larger range also enables a function that is difficult on the smaller meter: battery recharging. Due to the higher load currents, the Triumvi meter can recharge its onboard battery to permit sampling while the metered load is off.

6.1.7 Detailed Load Analysis

Some applications require more insight than just power or power factor, and actually need the raw AC current waveform for more advanced power analysis or power quality monitoring. Our system supports these applications by transmitting the raw current waveform samples when requested. Each Triumvi meter is able to collect and transmit samples of the current waveform for an entire AC cycle (a total of 120 samples) on each activation.

6.1.8 Upgrades for Proven Deployments

Because of the flexibility in the system design, the meters can be upgraded *in situ* if a trial deployment proves successful and the additional labor cost is justified. This progressive approach is a substantial departure from other systems that require the entire cost to be borne up front.

The first upgrade involves providing power for the sensors to enable a more reliable sampling interval. One option for this is to enable charge sharing between Triumvi meters. In this configuration, the sharing meters have their harvesting capacitor banks wired together so that each harvester is charging the cumulative energy store. This allows a meter with a higher harvesting potential to subsidize other meters. Another option is to provide an external power supply for the meters and daisy-chain the power supply wires between the

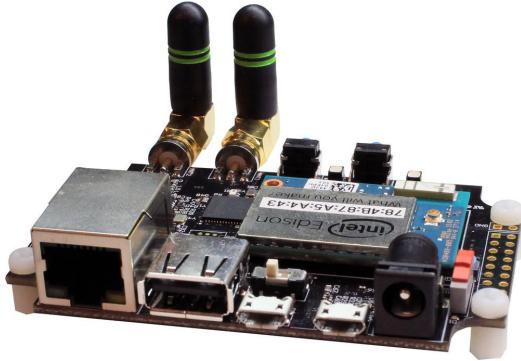


Figure 6.5: Gateway hardware. The Triumvi gateway contains an Intel Edison [57] and a dedicated CC2538 [130] co-processor for receiving the 802.15.4 packets from Triumvi.

meters. This upgrade enables reliable sampling but does not require replacing the already deployed and configured sensors in the panel box.

The other option for an in situ upgrade is providing the voltage channel information externally to each meter. This too does not require removing or replacing the existing meters and enables higher accuracy when calculating power. To implement this option, wires are connected to an unused breaker, run to a combined power supply and voltage isolation circuit, and then daisy-chained to each installed meter.

6.1.9 Network and Gateway Operation

When reporting data for real-time analysis, each Triumvi meter expects an always-on gateway (Figure 6.5) within range to receive measurement packets. As the energy-harvesting can be unpredictable, reliably scheduling transmissions at a certain time slot is not always possible. The gateway both logs the power data locally, and is able to forward packets to a central data collector. One backend the gateway supports is EmonCMS [39], which is an open source tool that supports collecting and visualizing energy data, as well as power to energy conversion and other stream processing features. This tool provides real-time monitoring of power data streams.

In addition, the gateway provides local “walk-up” access to power meter data for the circuit panel. The gateway aggregates data from all reporting Triumvi devices and provides the power data as a webpage. This allows for quick spot-checking power draw of nearby circuits.

6.1.10 Deployment Considerations

A portion of the Triumvi design is related to features that support Triumvi configuration and deployment. First, Triumvi includes a reset button that clears any stored aggregate energy measurements on the meter. This allows Triumvi to simply be reset or to be re-initialized if the meter is moved to a new circuit. The reset button must be held down until the meter has power to operate.

Second, to correctly interpret the data from the Triumvi devices, they must be correctly labeled with which panel box and circuit they are metering. To facilitate this, Triumvi includes an optional configuration expansion board, shown in Figure 6.4a, that exploits the structured nature of panel boxes and includes three knobs for setting the panel ID and circuit number the Triumvi is attached to. With these set, Triumvi tags each data packet it transmits with the measured circuit number and panel box ID. This removes a potentially cumbersome deployment step of manually recording and later entering the meter-to-circuit mapping. It also simplifies all future data processing as services, such as the walk-up gateway and backend data analytics, do not need to maintain their own device mappings. Pushing this meta information directly onto the deployed devices not only makes the deployment easier, but also promotes flexibility when building applications on top of the data stream.

6.1.11 System Variations

The base Triumvi system can be extended to accommodate different application requirements and scenarios. First, if a deployment has relaxed installation requirements and requires higher accuracy, the voltage signal can be externally monitored and the zero-crossings can be reported to each Triumvi through the expansion header. Similarly, if a higher or reliable same rate is required, each node can be powered through the expansion header, and energy-harvesting will be disabled.

If a reliable sample rate is necessary and the deployment is not intended to be permanent, an expansion board can be attached to Triumvi that includes a battery. In Figure 6.4a, the configuration board shown includes an optional rechargeable battery and charging circuit. The battery provides reliable power to facilitate a steady sample rate, while not encumbering the installation process.

Triumvi can also operate without a gateway or network connectivity for a short-term deployment using its local logging feature and its RTC for timestamping. When the Triumvi units are retrieved their data can be downloaded and analyzed.

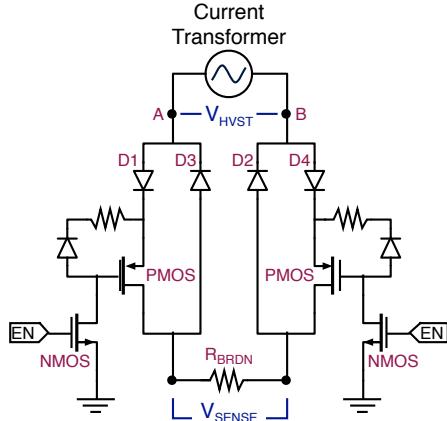


Figure 6.6: Current transformer multiplexing circuit. This circuit is responsible for switching the CT between harvesting and measurement operation. When EN is high, the MOSFETs connect the inputs of the CT across the burden resistor through diodes D1-D4 and enable current sampling. When EN is low, the PMOS FETs stay off, preventing any current flow through R_{BRDN} and enabling harvesting. Multiplexing the current transformer is essential for saving size and cost by requiring only a single CT.

6.2 Triumvi Implementation

Triumvi's implementation centers on the energy-harvesting circuit, the circuit that switches the current transformer from harvesting to measuring, the microcontroller and software that calculates power, and the power reporting transmission.

6.2.1 Energy-Harvesting Circuit

The energy-harvesting circuit is a well established design based on the LTC3588-1 [76] and prior work [13, 21]. The LTC3588 rectifies the incoming signal from the TT 100-SD [72] CT and uses $300 \mu\text{F}$ of capacitance for energy storage and an additional $100 \mu\text{F}$ capacitor for output buffering. The output of the harvester feeds a 3 V LDO for powering the rest of the system.

6.2.2 Current Transformer Multiplexing

When Triumvi is taking a measurement and must switch the CT into measurement mode, the CT multiplexing circuit is responsible for applying a burden resistor across the inputs of the CT. Importantly, it must also completely disconnect the burden resistor when harvesting. Figure 6.6 shows the main components of this circuit.

The mode of the CT is set using the EN line. When EN is high, the CT is in measurement

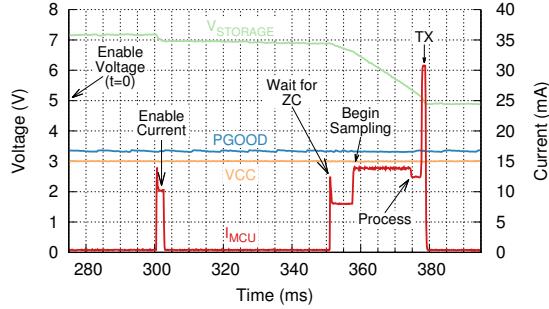


Figure 6.7: Current and voltage trace of a Triumvi measurement. The voltage on the storage capacitor, PGOOD, VCC, and the current draw of the microcontroller are shown while Triumvi measures power. At the start (time equals zero) Triumvi enables the voltage measurement circuit and waits 300 ms (this initial event and waiting period are omitted from the figure but included in timing and power calculations). It then enables the current measurement circuit and waits 50 ms for the output to stabilize. Next it waits for a rising zero-crossing of the AC voltage signal before sampling the current for 16.6 ms. After sampling it computes power, forms an encrypted packet, and transmits the packet.

mode, and when EN is low the harvester is operational. In harvesting mode, EN being low turns off the NMOS FETs, causing the gate of the PMOS FETs to float to the same voltage as the input from the CT, causing the PMOS FET to be off. This disconnects R_{BRDN} from the CT as the only remaining path through D2 and D3 is blocked by the diodes. When EN is high, the NMOS FETs are on, pulling the gates of the PMOS FETs low and enabling the PMOS transistors. This completes a path for current to flow through R_{BRDN} . During half of the AC cycle when point A is at a higher voltage than point B, current flows through D1, across R_{BRDN} , and then through D2. During the other half of the cycle the opposite happens, as current flows through D4 and back through D3. In measurement mode the harvesting circuit is still connected, but the introduction of R_{BRDN} causes the input voltage to be too low, effectively disabling the harvesting circuit.

6.2.3 Calculating and Reporting Power

When sufficient power is available, the TI CC2538 [130] microcontroller coordinates the circuits and measurement inputs to take an AC power reading. Figure 6.7 shows the timing of Triumvi’s operation. Triumvi starts by enabling the voltage detection circuit and waiting for 350 ms for the zero-crossing detector to stabilize. After 300 ms have elapsed, the current measurement circuit is enabled and Triumvi waits for the remaining 50 ms for that circuit to stabilize as well. When both channels are ready, Triumvi waits for a rising zero-crossing of the voltage signal and then begins sampling the current channel at 7.2 kHz for one period (which for a 60 Hz signal is 120 samples). After sampling, the voltage and current

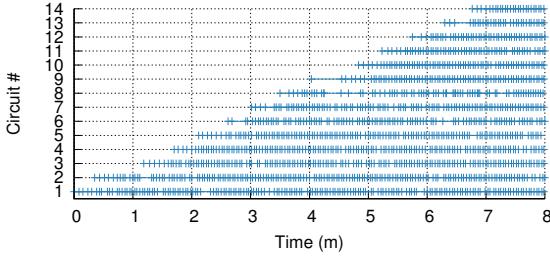


Figure 6.8: Installation time for 14 circuits. As Triumvi devices are configured with the panel ID and circuit number, and clipped around the correct circuit, they begin transmitting power measurements. Each mark represents a received packet, and the start time of each horizontal line marks when the Triumvi for that circuit was installed. Based on the start time of the last circuit, installing 13 meters took about seven minutes for a familiar, but inexperienced user. Including the time for removing and reinstalling the front panel (required for all CT-based installations) and first circuit install, we can reasonably assume a half hour to instrument a 14 circuit panel.

measurement circuits are disabled (causing the power supply to resume harvesting), and the microcontroller begins computing power. A pre-computed sine wave for voltage is phase aligned to the sampled current based on a factory calibration of the phase offset between the voltage sense circuit’s output and the actual AC voltage zero-crossings. After multiplying the two signals point-by-point, the power signal is averaged to compute true power.

With the completed power measurement, Triumvi prepares to transmit the reading while sufficient energy remains. First, it adds a nonce to the outgoing data and encrypts the payload using AES. Then it transmits the reading to a wireless gateway within range. The computation, encryption, and transmit phase takes 4.34 ms. The current Triumvi implementation uses a 2.4 GHz 802.15.4 [54] transceiver for communication, but alternative radios could be used instead to achieve the same goal.

The entire measurement and report takes 380 ms and 1.7 mJ. When completed, Triumvi sets a timer and enters sleep mode. The timer is set to wake the device for the next measurement, and its interval is based on a heuristic of the previous interval and the last power measurement. If this device is first starting after installation or a period of no harvesting, this timer defaults to 4 seconds.

6.3 Triumvi System and Deployment Strategy

To provide one measure of the deployability of the non-contact and self-contained Triumvi meter, we measure how long it takes to install 14 clip-on meters (Figure 6.8). Each Triumvi meter was equipped with the configuration expansion board and battery. The marks on

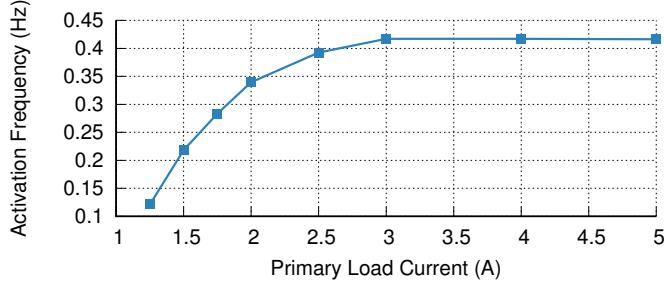


Figure 6.9: Activation rate as a function of primary load. Triumvi’s measurement rate is observed for a range of loads with power factor of 1.0. Below 1.25 A, Triumvi is unable to harvest enough power to successfully measure, but Triumvi can sample at nearly 0.5 Hz with loads above 3 A, which is rate-limited to mitigate wireless channel congestion.

each horizontal line indicate received packets and start when after the meter was set to the correct circuit and the meter was installed. The interval between the start of each line is approximately 30 s, and the installation took about seven minutes. Factoring in overhead for accessing the panel, it is likely an average deployment could be completed in less than a half hour.

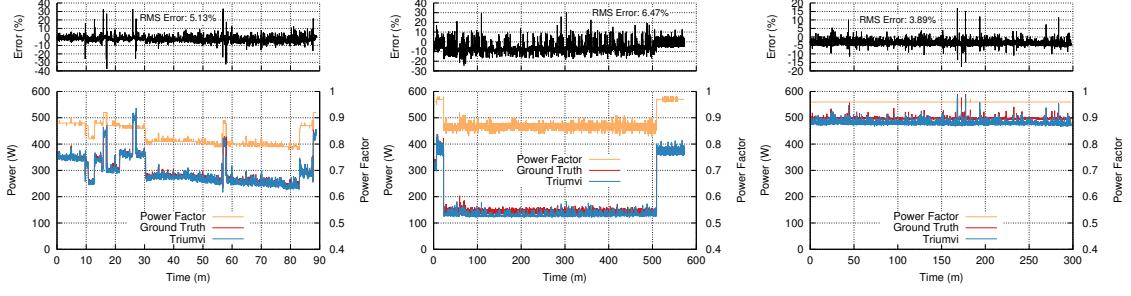
This short installation time makes the Triumvi system well suited to both long-term deployments and short-term “energy audits”. Whereas other systems require invasive installations (turning off power) or include long lead times (hiring an electrician), Triumvi’s installation costs are low enough to make the system feasible for a one day or one week deployment. Such a deployment could substantially guide decisions as whether to pursue a longer or more complete deployment.

6.4 Triumvi Performance

We measure the core performance of Triumvi by examining its accuracy with real loads and in comparison to a current-only version.

6.4.1 Measurement and Harvesting Range

One primary drawback of using energy-harvesting for power metering is that sample rate is tied to the load being measured. To understand the sample rate Triumvi can provide, we observe its packet transmission rate across a range of loads. Figure 6.9 shows the resulting activation rate. Below 1.25 Amps, (or 150 W at 120 V), the meter cannot harvest enough to activate. At 1.25 A, the meter can report a measurement approximately every 10 seconds. This rate increases until the load is drawing 3 A, where the Triumvi measurement rate is



(a) Computers + Lab Equipment (b) Desktop + Audio + Lighting (c) Server Rack

Figure 6.10: Triumvi accuracy with three real-world circuits. Triumvi is compared against ground truth for three circuits representative of three different environments where circuit panel meters may be deployed. The circuit in 6.10a contains two desktop computers, four monitors, and various lab equipment, as may be present in an office building. Even with a power factor consistently below 0.9, Triumvi is able to meter the circuit with 5.13% RMS error. 6.10b shows a circuit with LED lighting, surround sound audio, and a desktop computer, as may be present in a residential setting. Triumvi slightly under-reports power and overall RMS error is 6.47%. The third circuit, 6.10c, contains a rack of servers as may exist in a more industrial setting. The servers are a very constant load which Triumvi underestimates but otherwise tracks well, including two spikes around 170 minutes into the test. For that circuit, Triumvi displays only 3.89% RMS error. These traces demonstrate Triumvi’s ability to meter real-world circuits.

capped at approximately one packet per two seconds. This cap is intended to reduce the channel contention when many Triumvi meters are co-located.

6.4.2 Accuracy

To evaluate the accuracy of Triumvi as an overall power meter, we meter three real-world circuits with Triumvi and a PLM-1LP [103] ground truth meter. The traces, shown in Figure 6.10, reflect circuits likely found in three different building types: office, residential, and industrial. Over all three traces, Triumvi provides power measurements with 5.16% RMS error when the meter is able to harvest and take samples. The error comes primarily from an offset present during the steady-states in Figure 6.10b and Figure 6.10c and, as is common with energy-harvesting power meters, from spikes when the load suddenly changes. Triumvi must continue harvesting until it is able to measure and detect the change in load power.

For all three circuits, the power factor is always less than 1.0. Even with the aggregation of many loads, some containing power factor correction, real-world circuits do not have unity power factors and calculating true power is essential for accuracy. While Triumvi’s

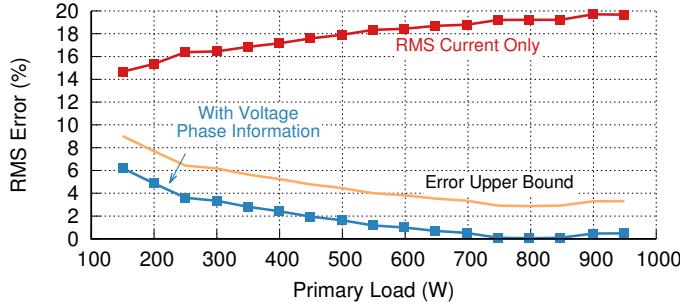


Figure 6.11: Triumvi error with and without voltage phase information. For a range of loads with power factor of 0.85, the RMS error of Triumvi’s power measurement using only RMS current and using sampled current plus voltage phase information is shown. Without any voltage information the error averages 18.5% RMS, and with phase information the average error is 2.5% RMS.

accuracy is slightly lower than commercial-grade panel meters, Triumvi is not intended to provide revenue-grade metering. Some error is acceptable when addressing the submetering problem. However, in non-ideal cases (non unity power factor loads) without the ability to calculate true power, the error would grow to outside of an acceptable range.

To further investigate the accuracy of Triumvi, we explore the accuracy benefits of having voltage phase information on power calculations. We run Triumvi again with loads ranging from 100 W to 1000 W and power factors of 0.85. In the first run, Triumvi reports true power, and in the second it reports just RMS current and is later multiplied by RMS voltage. The RMS error of these power measurements is shown in Figure 6.11. The RMS error over the range is on average 7 times better with voltage phase information instead of using just RMS current.

While it can measure phase, the Triumvi design assumes a constant voltage amplitude. The resulting upper bound error from the voltage fluctuations common in buildings is also shown in Figure 6.11. This error, while still present, is significantly lower than the error caused by no voltage information. Even with the error present, the Triumvi design significantly reduces measurement accuracy when compared to current-only meters.

6.5 Summary

With Triumvi, we show how carefully integrating components from previous work on energy-harvesting power meters with a new current transformer multiplexing circuit, voltage sensing technique, charge sharing, and synchronization, plus improvements in metadata association, can create a deployable and standalone energy meter. Triumvi is accurate, even under

displacement and harmonic power factors, and yet can be deployed in under 30 seconds per circuit. While the core sensing technique is often the focus of new meters, Triumvi shows that focusing on deployment ease can significantly improve the device and is also feasible on just an energy-harvesting power budget.

The improvement from AC Monjolo, to Gemini, and then to Triumvi also highlights how carefully tuning the system can lead to a very usable energy meter even with a non-traditional power supply. Triumvi is an example of a small and unobtrusive sensor designed for a practical sensing problem that benefits from the techniques described in earlier chapters, and demonstrates an important aspect of the thesis statement. In the following chapters, we take a broader perspective to explore how else energy-harvesting sensors can be useful indoors. We also investigate how continuing to share modular energy-harvesting components, as with these power meters, can aid in the rapid development of easy-to-deploy energy-harvesting sensor systems.

CHAPTER 7

Beyond Energy: Building Up the Building Sensing Toolkit

Triumvi, and the Gemini and AC Monjolo meters it builds upon, demonstrates how an energy-harvesting sensor can provide accurate power measurements, while scaling a deployment to a building-level footprint. While energy consumption is an important parameter to monitor, many smart building applications require more input data streams than just power. Recent work into smarter and more efficient buildings proposes many avenues for improvement, including building control algorithms [5, 98], dataset analyses [44, 59, 65], feedback mechanisms [104, 114], and personalized HVAC systems [51, 81]. A common thread through these strategies is that they require sensing devices to feed information about the state of the building and its occupants into models or algorithms. More easily acquiring these data streams, particularly in retrofit situations, lowers the barrier to implementing these new strategies.

To make these data streams easier to obtain, we propose leveraging energy-harvesting technology for the sensors to reduce the cost and complexity of installation. Energy-harvesting reduces the cost of deploying and managing sensors by eliminating the need to provision for mains power or periodically replace batteries. It also reduces the intrusiveness and improves the aesthetics of the sensors by enabling them to be smaller in size. These sensors can be easily installed in a building by attaching them at key sense points. As we show, however, this increased deployment ease comes at a cost: the unpredictability of the energy supply results in reduced data accuracy and fidelity. Augmenting the sensors with small and inexpensive energy storage devices could address these drawbacks, but such devices do not exist as commercial products with unlimited cycle life today.

The remainder of this chapter studies three such building monitoring sensors, all built with energy-harvesting power supplies. Beyond their sensing modalities, these devices differ from the energy-harvesting power meters by decoupling the energy-harvesting source from the phenomenon to be measured. This approach is advantageous when the sense point is not



Figure 7.1: Vibration sensor. A zero-power piezoelectric wakeup trigger activates the sensor when it is moved. When placed on the edge of a door, the sensor can be used to detect door open and close events.

a viable harvesting source or when a more fruitful harvesting source is physically nearby. This chapter also evaluates the sensors to understand how energy-harvesting affects their resulting data streams.

Buzz, Breeze, and BigBen are energy-harvesting sensors designed to monitor different aspects of building operations that could increase building efficiency or occupant productivity. These sensors provide concrete examples of how small energy-harvesting sensors can perform in the context of a specific application domain. They are also intended to be a launching point for future sensors that leverage some shared components, such as the power supply, processor board, or wakeup triggers, but observe other aspects and events of a building.

7.1 Detecting Door Open and Close Events

One aspect of building usage that can be useful to monitor is door open and close events. This can be used as one input to infer occupancy, as a trigger to turn lights on, or as a tool for general building usage analysis.

7.1.1 Buzz: Vibration and Sharp Motion Detector

Our first sensor, named Buzz and shown in Figure 7.1, operates as a vibration sensor. It uses a piezo-film vibration sensor [102] that, when vibrated, generates enough voltage to activate the trigger on the power supply. Once activated, the sensor uses its onboard radio to transmit a packet announcing that the vibration event occurred.

One application of this sensor is door monitoring. Buzz is attached to the door and when the door opens and closes it vibrates the sensor enough for the door open event to be logged.

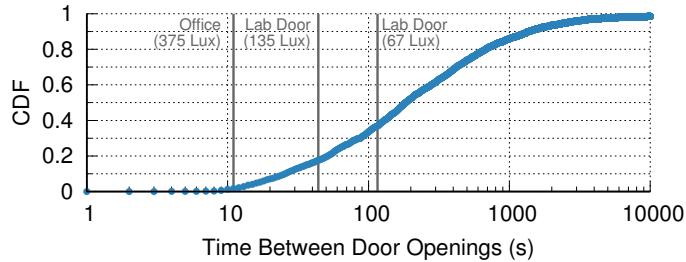


Figure 7.2: CDF of the interval between door open events. Plotted on a log scale x-axis is the CDF of time intervals between subsequent door opening events of a door over a month period. Also shown are the recharge times for the solar based energy-harvesting power supply in different lighting conditions. Sensors in rooms with natural light would be able to detect most door open events, and even in moderately lit rooms at least 65% of door open events would be detected.

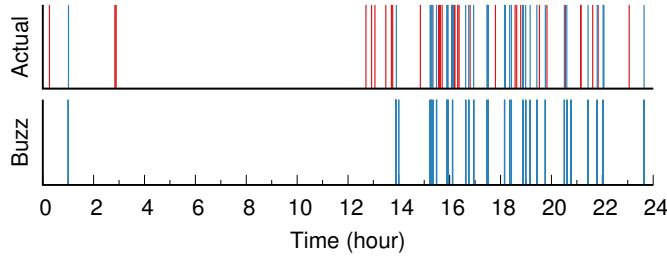


Figure 7.3: Door open events across a 24 hour period. Buzz was deployed on the door of a lab with moderate lighting and no exterior windows. Each vertical line represents a door open event as detected by a ground truth sensor and the energy-harvesting Buzz sensor. The red dashed lines in the ground truth row are door open events that Buzz did not detect. Buzz successfully detected 66 out of the total 100 door open events. This aligns well with our expected detection rate from Figure 7.2.

Similarly, the sensor can be attached to chairs to detect when the chair becomes occupied.

7.1.2 Buzz: 65% of Events Detected

We configure Buzz, the vibration sensor, to detect door open events. This application lends itself naturally to the event trigger mechanism. The primary limiting factor of this sensor is the recharge rate of the power supply: if a door opens before the supply has recharged from the last open event, the sensor will be unable to detect the second opening. To estimate how the recharge rate enforced sample rate would affect the sensor's ability to detect when a door is opened, we analyze the time between subsequent door open events for a door over a one month period. Figure 7.2 shows the CDF of the interval between subsequent door openings plotted with the x axis in a log scale. Also marked in the figure are the approximate sensor



Figure 7.4: Breeze airflow sensor. The solar energy harvester is placed near a light source, such as a window, and a MEMS airflow sensor is placed near an air vent. When the harvester has sufficient energy it samples the airflow sensor and reports its readings. Decoupling the harvesting from the measurement point enables both faster sampling and a smaller sensor.

recharge rates for the doors in the lab and office. In the office with natural light the sensor would miss only 2% of door open events. In the lab with normal lighting the sensor would have missed 35% of door open events over the month period. The missed door events would all have been within two minutes of each other, however, and it is feasible that finer temporal resolution may not be required for high-level building management algorithms.

To test Buzz in practice, we attach the sensor to a door over a 24 hour period. We also use a mains powered door opening sensor for ground truth. Detected door openings versus actual door openings are displayed in Figure 7.3. Buzz is largely effective at detecting door open events, catching 66% of the openings. Openings that occurred after a long period of no activity were missed, likely due to the power supply being unable to charge when room was unoccupied and the lights were off. Also, events that occurred in rapid succession did not allow Buzz enough recharge time, causing those events to be missed as well.

7.2 Monitoring HVAC Vents

Occupant comfort is an important goal of a building’s operation, yet existing HVAC systems are often very course grained with zones that cover multiple rooms. To gain more insight into how the building HVAC system is operating, and to detect potential faults, a building operator may need to monitor individual air vents.

7.2.1 Breeze: Airflow and Air Vent Meter

Breeze (Figure 7.4) uses an analog MEMS airflow sensor [93] and opportunistic triggering to detect when an air vent is blowing air. The airflow sensor is placed near the vent to be monitored and the solar cell and power supply are placed in a nearby location conducive to photovoltaic harvesting. After each recharge, the power supply triggers automatically.

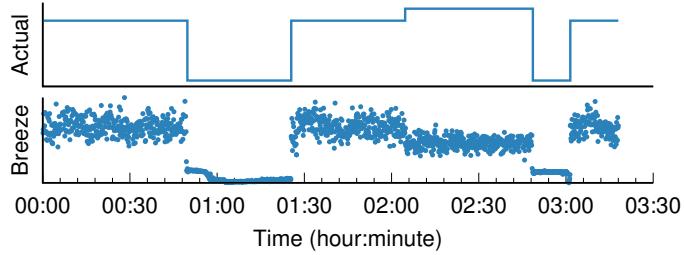


Figure 7.5: Airflow metered over time. A Breeze sensor is placed near an air vent as the vent cycles on and off and changes its flow volume. The airflow sensor is uncalibrated and therefore we don't include y-axis units. While noisy, the data from Breeze are sufficient to determine when air is blowing from the vent and to determine when the flow changes.

Upon booting, the device powers the airflow sensor and waits for the output line to stabilize before beginning to sample the sensor at 1 kHz. After each sample the measurement is stored locally. This continues until Breeze depletes its stored energy reserves. On the next recharge and activation, Breeze averages the measurements it sampled on the last activation and wirelessly transmits the result. For airflow sensing this split operation is acceptable if subsequent activations occur quickly enough that the samples do not become out of date before they are transmitted. In our deployment we expect the power supply to be near an abundant source of energy, allowing for activations separated by no more than 5 s. If this was not the case, we could enlarge the energy store on the power supply to allow for sampling and transmission on the same activation, although at the expense of longer intervals between sampling.

Breeze is also an example of an energy-harvesting sensor where it is advantageous to decouple harvesting and sensing. Harvesting from air vents is possible, but typically requires a physically larger harvesting unit than from lighting. By separating these tasks, the overall size of the sensor can be reduced.

7.2.2 Breeze: 0.125 Samples/Second

To demonstrate external sensing on an energy-harvesting node we developed Breeze, a sensor that periodically measures airflow. To evaluate Breeze, we placed its airflow sensor near an air conditioning vent and its power supply very close to an overhead light. Again, to collect ground truth we placed a mains powered sensor on the same vent. Figure 7.5 shows the results from the two sensors. Because Breeze is uncalibrated we omit the absolute magnitude from its samples. The on-to-off transition at the 42 minute mark and the corresponding drop in Breeze's readings shows that Breeze is, at a minimum, able to detect the on/off state of the vent. When the volume of airflow increases, at 2:06, Breeze's readings drop. This occurs



Figure 7.6: Light based occupancy sensor. The energy-harvesting sensor detects state changes in the measured light by detecting when the sensor transitions between different harvesting rates. For example, in a room with only a single source of light, the sensor can detect when the light turns on by observing when the sensor transitions from not harvesting to harvesting. With more ambient light in the room this transition may not be as stark, but there will still be a detectable change in harvesting rate. Timestamping to determine the rate of harvesting is done with a real-time clock.

because Breeze measures air speed, and the volume rate limiting is occurring at the vent. Since the HVAC unit continues to supply the same amount of air, the increased opening at the vent causes the air to move slower. Breeze was attached very close to an overhead light for this experiment and averaged a sample period of 7 s. When light is available, Breeze is a suitable sensor for airflow detection. When the lights are off or no daylight is present, Breeze will be unable to sample.

7.3 Monitoring Interior Lighting

For energy efficiency analyses, and as another input for occupancy detection, monitoring the light usage in a building can be particularly useful. Also this may be useful as a planned study without gateways and with manual data collection to inform if energy efficiency upgrades are warranted.

7.3.1 BigBen: Light/Occupancy Sensor

Our third sensor is a lights on/off sensor named BigBen (Figure 7.6). BigBen is designed to be installed very close to a light source allowing BigBen to do light state detection directly with the power supply: if the power supply is able to harvest then the light must be on. BigBen does not measure light intensity, but rather the binary state of the light. Specifically,

it detects when the light is on based on when activations occur and when the light is off based on the lack of the trigger firing. When the light is on, many triggers will occur that signify that the light has simply remained on. Rather than store this redundant information, BigBen only makes note of when the light goes from off to on or vice-versa.

Due to the intermittent nature of BigBen, a state change of the light manifests as a change in the time between activations. If BigBen is activating at a certain rate and that rate suddenly drops, the light turned off, and if the rate suddenly increases, the light turned on. Detecting these rate change events presents an interesting challenge: how does BigBen know if the activations rate changes? Each activation appears identical and by default there is no time keeping. To address this, we supplement BigBen with a low-power Micro Crystal RV-3049-C3 [84] real-time clock (RTC) powered by a small (17 mAh) battery. This provides a reliable time source that BigBen can use to timestamp each activation. Now calculating activation deltas, and therefore light change events, is straightforward.

Using the RTC provides an additional advantage: change events can be timestamped against a global clock. This is useful because BigBen logs light change events locally rather than transmitting them. By having access to the RTC, BigBen can provide information about when the lights turned on and off, not just how many times. BigBen logs rather than transmits because it was designed to monitor rooms with poor connectivity over a short period of time (~ 3 months). This makes retrieving data from all of the sensors feasible. If the trial run demonstrates that light monitoring is worthwhile, the sensor can be redeployed and configured to transmit rather than log.

When lighting state can be considered a proxy for occupancy, BigBen can serve as a reasonable occupancy detection sensor. This is based on the observation that many rooms in buildings and offices contain motion sensitive lights. That is, when motion, and therefore people, are detected in a room the lights stay illuminated, and when no motion is detected the lights automatically turn off. BigBen can detect these light changes and use them to roughly infer when the room was occupied. This is still an estimation, but in certain cases it may be a suitable method for deploying an energy-harvesting occupancy sensor.

7.3.2 BigBen: 40 Second Time Jitter

We deploy BigBen, our timestamping sensor that senses only based on its ability to recharge and eschews connectivity for local logging, on a light that is illuminated automatically only when there are occupants in the room. At the end of the measurement period the data is dumped from the node's storage and compared against the control system's ground truth commands as shown in Figure 7.7. In a 24 hour evaluation period, BigBen tracks ground

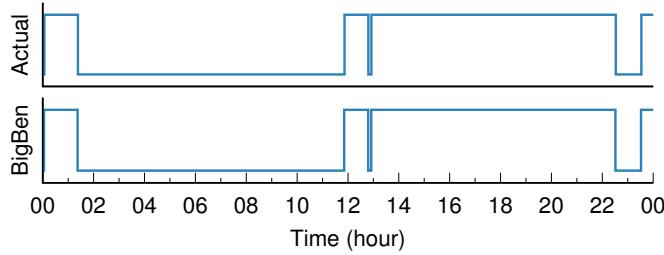


Figure 7.7: Light state over time. BigBen monitors the on/off state of a light fixture for 24 hours. It detects change events well and shows only a 40 s offset error from ground truth, likely due in part to jitter errors when initializing the RTC.

truth state changes very well, with only a constant 40 s timestamp error.

Because BigBen does not transmit immediately after detecting a lighting change, it must timestamp the events itself, and this requires an RTC powered by a small rechargeable battery. To estimate how long BigBen will be able to successfully timestamp light change events we compare the battery and the RTC. We use a 17 mAh battery and the RTC [84] draws 800 nA, which means the RTC will run for approximately 2.4 years. In practice, however, we expect BigBen to serve as a temporary sensor which will be replaced with a networked device before the battery completely discharges.

7.4 Energy-Quality Tradeoffs

As the evaluations of each of these sensors shows, the tradeoff for perpetual and deployable sensing based on small energy-harvesting power supplies is some reduced data fidelity. This changes the typical paradigm for how sensors are used. Often sensor data are considered reliable. In the energy-harvesting case, however, events can be missed, or the sensor can stop reporting simply because the lights are off, and not because the event is not occurring.

To compensate for this, two potential methods might exist for hiding the underlying data issues from the sensors. First is a hardware addition to the sensor. By adding a backing energy store (perhaps from Section 3.3), the node may be able to mask some of the variability by diverting excess harvested energy to the backing energy store, and using that to detect events when the capacitors are not sufficiently charged. Figure 7.8a shows a CDF of estimated door events captured with Buzz augmented with additional rechargeable energy storage. With 1 mJ of storage (i.e. the assumed amount of energy to detect one door event), we estimate that Buzz could detect about 70% of the door events. As the amount of energy storage increases, Buzz can harvest additional energy between door events (when the lights are on) and store this energy to use when door events occur in rapid succession. When Buzz can store about

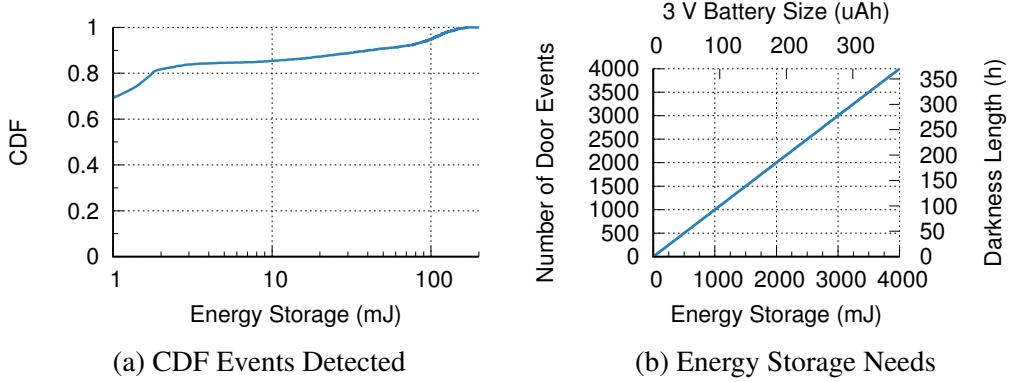


Figure 7.8: (a) shows a CDF of detected door events for a range of rechargeable energy storage capacities. We assume 25 μW of harvesting between the hours of noon and 2 AM, no harvesting between 2 AM and noon, 3 μW of leakage, and 1 mJ per door event detection. We then analyze the door events from Figure 7.2 to determine how many events Buzz could detect with varying amounts of onboard energy storage. (b) shows how many door events could be detected or how long the sensor could last without harvesting and still detect the next door event for a range of energy storage capacities.

168 mJ of energy (a 37 mF capacitor at 3 V or 16 μAh battery at 3 V), it should be able to capture all door events in the Figure 7.2 test case. Additionally, energy storage should allow a Buzz node to detect events even after periods of darkness (e.g. after a weekend or vacation period). Figure 7.8b shows how many hours of darkness the node could tolerate (and how many door events it could detect) with a certain amount of fully charged energy storage capability. For instance, Buzz would need about 3,630 mJ of stored energy to detect the first door opening after two weeks of darkness.

The second relies on a software layer above the sensors. This approach treats the data streams from the energy-harvesting sensors less like typical data streams and more like hints. Then, by combining hints from multiple sources, and potentially a model learned over time, the system outputs a much higher confidence signal about the conditions or events occurring in a building. These improvements may enable many of the reliability guarantees of traditional sensors while also preserving the energy-harvesting deployability benefits.

7.5 Summary

With these additional sensors, the energy-harvesting sensor toolkit for monitoring buildings continues to grow. This toolkit can help characterize buildings by providing a measurement apparatus to understand how a building is currently operating, understand buildings by correlating events and energy consumption, and improve buildings by providing data streams

to smart building algorithms and applications. The common characteristic that enables this is their deployability, driven by using energy scavenged from the environment to power the devices. Some devices in the toolkit defer energy-intensive operations (e.g. sending a radio packet), a technique in the thesis statement, and all of the devices are small, unobtrusive, and inexpensive and designed for practical sensing problems as specified in the thesis statement.

This growing toolkit also provides an opportunity to examine commonalities across the various devices. While the increased usability and reduced maintenance costs are advantageous, relying on energy-harvesting complicates the design and implementation of these sensors. By finding and isolating similarities, some parts of the design may be reusable. This would simplify creating new sensors by being able to leverage known and working components. We explore this approach more fully in the following chapter, arriving at an integrated modular architecture for energy-harvesting sensor systems.

CHAPTER 8

An Integrated Modular Architecture for Energy-Harvesting Sensor Systems

In Chapter 3 we discussed an ambient environmental sensor, in Chapters 4 to 6 we covered energy meters, and in Chapter 7 we explored event detection sensors. These sensors all have in common that they use energy-harvesting power supplies and target indoor monitoring, but the details of how they accomplish their goals vary, and in some cases significantly. This exposes an issue with energy-harvesting devices. Designing these systems often requires considering the entire device stack, from the power supply to the microcontroller and sensors to the software, at every step in the process. The traditional embedded systems abstractions break down, and power supply details leak into the design of every other part of the system. This makes creating energy-harvesting devices much more challenging than other embedded devices. However, these sensors also identify new architectural patterns that are common to all of these devices.

8.1 Architectural Overview

By examining the commonalities in the energy-harvesting sensors presented here and in related work, we derive an architecture that captures the core layers present in all of the devices. The overall node architecture is depicted in Figure 8.1. Each sensing device is comprised of four main subsystems: the energy-harvesting power supply, a node triggering mechanism, the primary sensing apparatus, and a data communication module. Each sensor accrues energy at run time with a suitable energy-harvesting frontend. The architecture specifies minimal constraints on the power supply for successful system operation, however, the harvesting supply determines the sensing capabilities and performance of the overall device. Power supplies that range from ones that can supply the device continuously to anemic ones that offer very limited, intermittent runtimes are supported. Application goals define the power supply requirements.

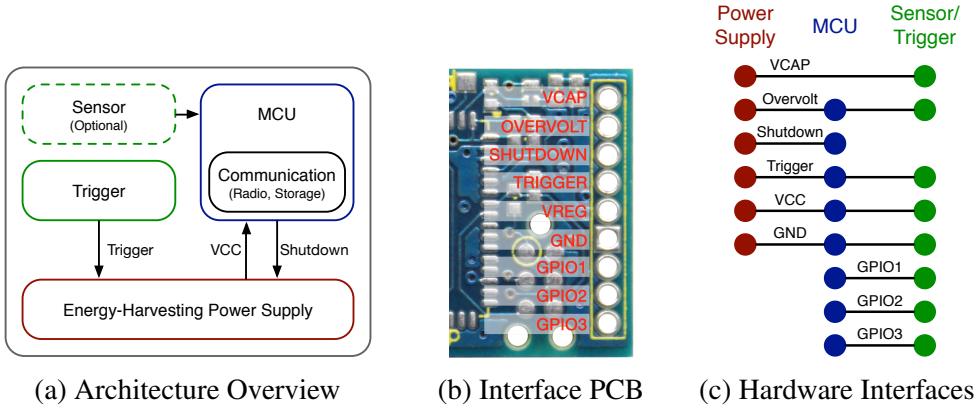


Figure 8.1: Energy-harvesting node architecture and interfaces between physical components. The physical interface is effectively a shared bus as the various layers (power supply, controller, and sensor) stack and share the same signals. Not every layer must use every signal, however. For example, the power supply has no use for the generic GPIO pins, but the sensor modules often use them for ADC readings or a digital bus interface.

The trigger mechanism is responsible for activating the computational, sensing, or communication resources on the sensor, depending on application needs. The trigger may fire as soon as sufficient energy is available or in response to an external event, such as a door opening.

An on-board sensor may be used to monitor some building condition. The sample rate is bound by the power supply harvesting rate and the range of usable sensors is governed by the energy store. Low power sensors that can obtain a reading quickly are best suited for this architecture.

Finally, the data communication module conveys the information collected by the device to a central building control system or other data management service. This can be handled in real-time, batched, or logged locally and transferred in bulk at a later time.

8.2 Design Space of Energy-Harvesting Indoor Sensors

With the general layers shown in Figure 8.1, we use common indoor sensing applications to identify the design options available for each layer when instantiating a physical sensor. Different environmental factors, monitoring or detecting goals, and data rate requirements will guide which options are best for a particular application. Table 8.1 highlights the main options for each subsystem and a series of potential building-based sensor applications. A mark in a column for a particular sensor application indicates that that option is viable for that application. The next section provides a more detailed breakdown of the range of

Sensor Application	Power		Trigger			Sensing			Communication		
	Sustained	Intermittent	Periodic	Opportunistic	Event	External	Event	Energy	Immediate	Batch	Log
Door	✓	✓			✓		✓		✓		
Occupancy	✓	✓	✓	✓		✓		✓	✓	✓	✓
Temperature	✓	✓	✓	✓		✓		✓	✓	✓	✓
Light	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Motion	✓	✓			✓	✓	✓		✓		
Air Quality	✓	✓	✓	✓		✓			✓	✓	✓
Water Usage	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Solar Radiation	✓	✓	✓	✓		✓			✓	✓	✓
Chair Occupancy	✓	✓			✓		✓		✓		

Table 8.1: Energy-harvesting node architecture properties for a variety of building monitoring applications. This table highlights a number of potential building monitoring applications and the system properties that are likely sufficient for implementing an energy-harvesting sensor for the particular application. Each subsystem—power, trigger, sensing, and communication—needs one property to enable a functioning sensor. Some applications are best suited by event detection, some just need to detect harvestable energy, others are best served with periodic sampling, but all support intermittent power supplies making them candidates for energy-harvesting sensors.

options.

8.3 Architectural Layers and Abstractions

For each layer in the architecture, we identify potential options for a node’s design and tradeoffs for selecting each option. To realize a viable sensor within this architecture, each layer must be represented by at least one option.

8.3.1 Energy-Harvesting Power Supplies

The power supply is responsible for harvesting energy for the node’s operation. Depending on the harvester’s capabilities and node power requirements, the power supply may be continuous, i.e. its power output is equal to or greater than the average power draw of the node, or it may be intermittent, i.e. it is only capable of powering the node for short periods of time between charging cycles. The architecture supports both paradigms, however, particular applications may require one or the other.

A common harvesting source for indoor applications is small ($\sim 5 \text{ cm}^2$) photovoltaic solar cells because when a building is occupied it is typically lit. These photovoltaics can often be sized to fit node dimension constraints or average power requirements. Larger solar cells in sufficiently lit environments are capable of supporting continuous operation, while smaller solar cells can only provide intermittent power Chapter 3. Other indoor harvesting

sources have similar tradeoffs, including airflow based [149] and water pipe based [83].

Optionally, the power supply system may include backup energy storage to buffer variations in harvesting capability, much as the power grid does with renewable sources. A rechargeable battery or supercapacitor could be charged with excess harvested energy when the harvesting source is plentiful and discharged when it is needed, such as at night in the case of the solar cell. The stored energy may also be useful to continuously power a real-time clock or other timebase or to provide a burst of energy to power a wireless radio for bulk data transfer.

One important property of the power supply that enables the proposed architecture is a trigger, latch, and shutdown mechanism. Typically, energy-harvesting power supplies supply power on their output rails immediately when power is available. In the case of continuous supplies, power is always available. For some applications this may be useful, but others may only want to consume the limited energy reserves after a period of time has elapsed or an event has occurred. To support this, the power supply must have the capability to store energy without activating the remainder of the sensor until a trigger pin is asserted. Upon assertion, the power supply latches and keeps the power rail activated until energy is exhausted or the device requests that it be disconnected by asserting the shutdown pin. The power supply should also be capable of disabling this feature and supplying the power rail immediately after energy is available. This key property enables successful node operation and event detection with energy budgets on the order of millijoules.

The power supply is a very important subsystem as its characteristics and capabilities directly affect the operation of the entire sensor. The sample rate, supported sensors, and measurement capability of the device are directly based on the output and duty cycle of the node's power supply. In this way, the power supply's characteristics define many of the constraints on the rest of the sensor, such as the runtime duration, maximum instantaneous power draw, per-event energy consumption, and other operational limitations. Therefore, the power supply and application must be well paired to ensure a successful device.

8.3.2 Triggers: Decoupling Harvesting from Computation

The trigger subsystem determines when the sensor should take a sample, perform some computation, or make note of an event. Using the trigger abstraction and an event based model to control the operation of the device allows the architecture to natively support intermittent power supplies as the intermittency can behave as a trigger. Different applications will use different types of triggers to best accomplish their sensing goals. Trigger types fall into three main categories: periodic, opportunistic, and event based.

The **periodic trigger** is a trigger designed to fire at a regular interval. If the harvesting source is predictable and the sensor has some timebase, it may be feasible to have the trigger activate periodically in order to take regular samples. This periodicity matches how many battery-powered sensors operate when monitoring an environment and provides a fixed sample rate. If the harvesting potential changes there may not be sufficient stored energy to sample and the sample will be lost, just as a sample can be lost due to faulty packet transmission in a battery powered network.

The **opportunistic trigger** is based on an intermittent power supply. This trigger is activated by the power supply detecting that the stored energy has surpassed a threshold and supplying power to the rest of the sensor. This allows the device to sample at the maximum rate allowed by the intermittent power supply and not be limited to a precomputed rate that optimizes for average power. If harvestable energy is abundant, the sensor will potentially be able to sample faster than the rate budgeted for a battery-powered node.

The **event based trigger** activates the sensor in response to an external event. This allows the device to store up energy from an intermittent power supply until the event occurs. Detecting the event will likely require dedicated hardware that is capable of asserting the trigger pin on the power supply in response to the desired event occurring. Also, the rate at which the node can detect events is bounded by the rate the power supply can charge as the trigger is only active when there is sufficient energy to otherwise activate.

8.3.3 Sensing on a Limited Energy Budget

Certain building monitoring applications may need one or more dedicated sensors to fulfill the application's goals. Others may just rely on the trigger mechanism for sensing: the mere occurrence of the trigger provides the desired event detection. Yet other applications may simply rely on the ability to harvest energy or not.

Applications that employ dedicated sensors must ensure the sensors fit into the energy budget provided by the power supply. Most sensors will likely not be active continuously, but instead for very short intervals when power is available and sensing is warranted. Coupling sensing with event detection may be useful, and the trigger mechanism can provide a useful rate limiting for sensing: when the trigger fires, sample the sensor. This motivates using sensors that require little time between power-on and valid measurements. Once the sensor has been sampled it can be power-gated again allowing for a quicker recharge.

Using the trigger as a sensor is useful for applications that must detect events, such as doors or windows opening and closing, lights turning on, airvents activating, and chairs being sat on. These applications can rely solely on a trigger mechanism activating the output

rail of the power supply to sense a particular phenomena. It is possible this sensing style may be combined with a dedicated sensor to both detect an event and measure the conditions at the time of the event. For instance, a temperature reading may be taken immediately after a window open event to gain insight into the motivation for opening the window.

In some cases the ability of the power supply to harvest at all may be sufficient for the application's sensing needs. Similar to Monjolo (Chapter 4, the mere presence of energy from which to harvest can often be a sufficient sensor. For example, a solar cell powered device placed close to a light can tell when the light is on simply based on when it is able to harvest. Further, the relative rate of harvesting may provide additional information such as the brightness of the light. In this way, an energy-harvesting sensor can be constructed with no true sensor at all.

8.3.4 Data Communication or Storage

The final operating phase is to communicate, store, or otherwise handle the result from the sensing subsystem. To best enable real-time data processing and visualization, one option is to transmit the data immediately after sampling. This would likely require a wireless receiver to be within range and always listening. We propose this is feasible with a combination of gateways designed to bridge low power networks and the Internet, and other, mains-powered sensors, such as power meters, distributed throughout the building that can forward the data packets.

If delaying data transfer from the sensing device to the building control or data management service is tolerable for a certain application, batching multiple samples and transmitting the group may be a superior solution. This could potentially allow the harvesting device to save power by transmitting a single packet with multiple readings and incurring less payload overhead, and therefore transmit time and power, rather than sending multiple data packets each with their own headers and other overhead. Buffering packets may also allow the device the flexibility to probe for online receivers at each activation and, if possible, join a wireless network temporarily or pair with a device capable of relaying its data.

If a network with which to transmit the data is unavailable, the device has the option of logging the samples locally into some nonvolatile storage, such as flash or FRAM. This has the advantage of being a stand-alone sensor, but the drawbacks of requiring a data retrieval system and being limited by the storage space available for data. Likely this would require physically retrieving the sensor, downloading the stored data, and reinstalling the device. Other schemes may be feasible, such as a targeted wireless retrieval mechanism where a data mule [99] individually activates each device and requests its data. The device could store the

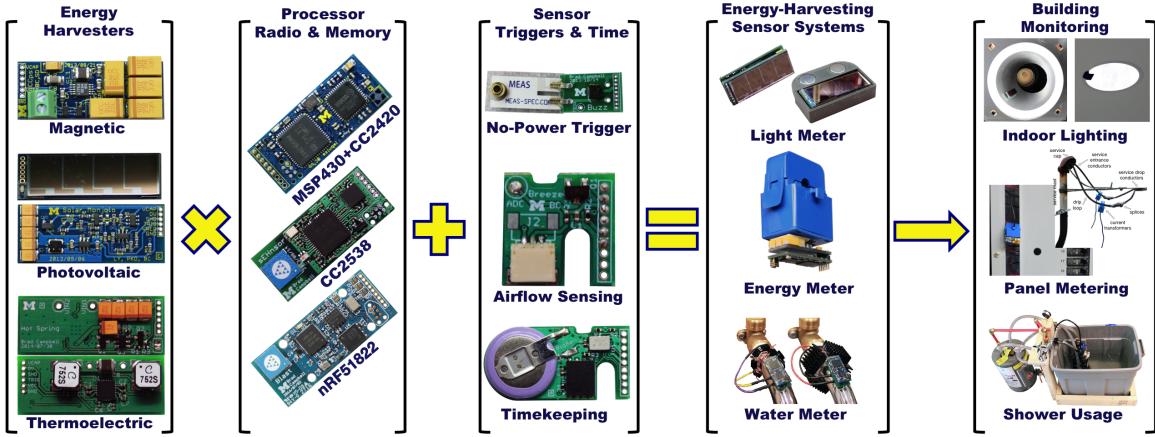


Figure 8.2: Composing the components in this architecture can help realize new sensors for different applications. A sensing system is composed of a harvesting frontend matched with a control and wireless board plus any additional needed sensors. The created systems can then be used to monitor different aspects of a building.

energy it would otherwise use for periodic transmissions to enable the bulk wireless transfer.

8.4 Architectural Evaluation

We argue that with this architecture we can compose a range of sensor systems that specialize in various sensing applications. Figure 8.2 shows how these different layers can be combined to create different types of sensors for different applications. The choices in each column allow for adapting the sensor to application needs. The harvesting frontend should be selected to maximize harvesting potential while meeting deployment and size needs. The processor and radio boards should be selected for computational power required, as well as the radio interface best suited for packet reception and user interface. Any additional sensors or triggers can be added based on the sensing requirements. The result is an energy-harvesting system composed of existing components that can be deployed to monitor specific points of interest inside of buildings.

8.4.1 System Architecture In Related Work

Some previous work has also identified the need for eliminating batteries from indoor sensing devices by moving to energy-harvesting power supplies. While these projects presented an energy-harvesting approach to a particular application, they can be well described as a part of our node architecture. We highlight two such projects here—DoubleDip, and Trinity—and describe how their designs help support our architecture as a viable framework for building

Signal	Magnetic	Photovoltaic	Thermoelectric	MSP430	CC2538	nRF51822	No-power trigger	Airflow	Timekeeping
VCAP	✓	✓	✓				✓		
Oversupply		✓		✓	✓	✓			
Shutdown	✓		✓	✓	✓	✓			
Trigger	✓		✓	✓	✓	✓	✓		
VCC	✓	✓	✓	✓	✓	✓		✓	✓
GND	✓	✓	✓	✓	✓	✓	✓	✓	✓
GPIO1			✓	✓	✓		✓	✓	
GPIO2			✓	✓	✓		✓	✓	
GPIO3			✓	✓	✓		✓		

Table 8.2: Common interface usage by shared components. The components in Figure 8.2 represent each column and the rows are each signal. The GPIO pins are mapped to multiplexed pins on the processor modules to ensure they are flexible enough to support a variety of sensor modules. For example, the airflow sensing board uses one GPIO to enable the airflow sensor, and one GPIO as an ADC input to measure the sensor output. The timekeeping board uses the GPIO pins as a digital bus to allow the processor module to communicate with the real-time clock.

energy-harvesting sensors.

DoubleDip [83] is a water flow monitoring system attached to pipes that uses an energy-harvesting thermoelectric generator (TEG) as both a power supply and water event detector. A sudden water flow event causes a temperature gradient to be present on the pipe and a resulting spike in the output voltage of the TEG. DoubleDip detects this spike and records a water event in the pipe. DoubleDip blurs the lines between a sustained power supply and an intermittent one by including a backup battery. A DoubleDip node that has limited opportunity to harvest can still respond to water events by powering itself from the battery, however, the system design caters to intermittent operation. It uses an event based trigger (a water flow event), an accelerometer as an external sensor once it has detected water movement, and transmits data immediately or batches and waits for sufficient power to be available. Again, intermittency is masked by waiting for an event-based wakeup before performing any sensing or computation.

The other design is another indoor energy-harvesting sensor named Trinity [149], a system for detecting and measuring airflow. Trinity uses one piezoelectric device to harvest and another, simpler piezoelectric device as a sensor to measure airflow. Trinity's runtime power draw and harvesting capability are closely aligned to provide operation similar to a battery powered node, but at a lower duty cycle. While this aids inter-node communication, it forces Trinity to be relatively large to provide sufficient power. Because of Trinity's harvesting capabilities we consider its power supply to be sustained. It uses a periodic sensor to initiate sensing, a dedicated vibration sensor for measuring airflow, and is able to transmit

after sampling. Trinity is the most similar to a battery powered node and demonstrates that our architecture supports applications where the node design is less limited by intermittency.

8.5 Summary

With this architecture, comprised of a harvester, trigger, controller, and sensor, energy-harvesting sensor systems can be decomposed into their component layers. By adding well-defined interfaces between the layers, the subcomponents can be independently created and tested. This allows for systems to be built out of shared components, much like in many traditional embedded systems. This building-block approach makes designing a new sensor for a particular sensing goal easier by allowing component reuse, and should increase the overall utility of energy-harvesting sensors to improve buildings. This architecture also encompasses many of the aspects of the previous chapters to support the main points in the thesis statement. In the next chapter we explore how the devices created within this architecture are able to address practical sensing problems while operating perpetually.

CHAPTER 9

Deployment Experiences

The previous chapters describe a range of energy-harvesting sensors with varying sensing goals. To understand how well these devices work in practice, we deploy them in real-world contexts and evaluate how they perform. As noted in the thesis statement, a main motivation for this class of sensors is that they can address practical sensing problems. This chapter explores the performance of a selection of the presented sensors. We analyze how the data they create can provide valuable insight. We also explore the limitations of energy-harvesting devices and describe the additional layers of the overall sensing stack that are required for these deployments to be successful: the gateway and cloud tiers. Gateways collect the transmissions from the sensors, do optional local data processing, and either store data locally or push it to the cloud. The cloud is responsible for data archiving, as well as data processing and presentation.

9.1 Non-Contact Energy-Harvesting Energy Metering

The first sensor we examine is the Triumvi circuit-level power meter. As the culmination of a series of improvements to the baseline Monjolo-style AC electricity meter, Triumvi is designed for ease-of-deployment and metering accuracy. We run four deployments of the Triumvi meters: one in a house and three in different apartments. The house deployment is a month-long trial with complete circuit coverage reflecting a realistic residential setting for the Triumvi meters. We analyze various aspects of the resulting dataset to understand not only how well the power meters performed, but also how well the devices as energy-harvesting sensors performed. The apartment deployments represent “energy audits,” or short-term deployments where sensors are used in a diagnostic fashion to identify opportunities for improvement or for more extensive sensing. Each deployment lasts 24 to 72 hours and includes a ground truth meter (an eGauge EG3000 meter [41]) against which the energy measurements are compared.



Figure 9.1: Multiple generations of Triumvi meters installed inside of circuit panels by simply clipping them around the conductor to be monitored.

9.1.1 Installation Overhead and Deployment Speed

One of the primary arguments for using energy-harvesting with power metering is its ability to ease the installation burden by allowing for non-contact sensors. To understand how this feature translates to an actual deployment, we analyze the installation statistics for the house deployment.

Installing Triumvi requires several steps: 1) Removing the front panel of the circuit panel box; 2) Dialing in the correct configuration settings on each Triumvi; 3) Clipping the correct meter around the correct circuit; and 4) Replacing the circuit panel cover. If the deployment requires charged Triumvis that use a battery to provide reliable samples, the Triumvi sensors must be charged prior to installation. To aid with charging and configuring, Figure 9.2 shows ten Triumvis in a storage tray that can both charge Triumvis and provide a convenient method for configuring the ten devices before installation.

The configuration trays are used for the house deployment. This aids the deployment because the panel ID and circuit ID settings can be “dialed-in” to each meter all at once. The meters are then installed in the panel. The entire deployment, including removing the panel box front panel, configuring and installing 32 Triumvi sensors, and powering on the gateway, takes 34 minutes. The other deployments are similarly quick when installing the Triumvi meters, but adding the ground-truth meter significantly increases the deployment time.

9.1.2 Dataset Efficacy

To evaluate the deployment of Triumvi in the house, we analyze various metrics related to energy-harvesting operation and energy sensing.



Figure 9.2: Triumvis charging before configuration. The Triumvi storage tray can charge up to ten Triumvis, and it is also a convenient jig for setting the panel and circuit knobs before deploying inside of a panel.

As an overview of the deployment, we plot the power traces from 28 circuits and measure the aggregate for a one day period within the study, as shown in Figure 9.3. This shows several features common to household power usage. First, most of the circuits remain relatively constant with a low power draw. Second, one circuit contains a clear cycling on-off behavior at a regular period. That is typical for a refrigerator which must cycle during the entire day. Third, are bursty, short-use loads, like toasters, microwaves, or dryers. These have high power draws when active, but are typically used infrequently so their energy use as a proportion of the aggregate is often minimal. Lastly, as we expect, the house has a higher aggregate power draw when occupied and the occupants are awake.

Power draw is often useful for detecting activities and patterns, but from an energy reduction viewpoint, integrating power to calculate energy provides a simpler mechanism for comparing different loads or circuits. Figure 9.4 shows the breakdown of the energy used by each circuit in the one day monitoring period. Circuit #29 is the largest consumer, which is most likely the refrigerator. The energy breakdown also allows for targeted investigations into energy consumption. Rather than identifying each circuit and the attached loads initially, only those circuits with substantial energy consumption need to be investigated.

For the deployment in the home, ground truth power data is not available. To provide one measure of the validity of the data, we compare the aggregate consumption measured by the Triumvi meters on the incoming power feeds with the sum of all of the metered circuits. We find that the two aggregate meters (two are needed because it is a two-phase system), sum to a measured 13.8 kWh for the day. The cumulative energy from the individual circuits totals 14.1 kWh, or a 2.2% error. This puts the likely error well within the error range for each Triumvi meter and suggests a certain level of precision across the sensors.

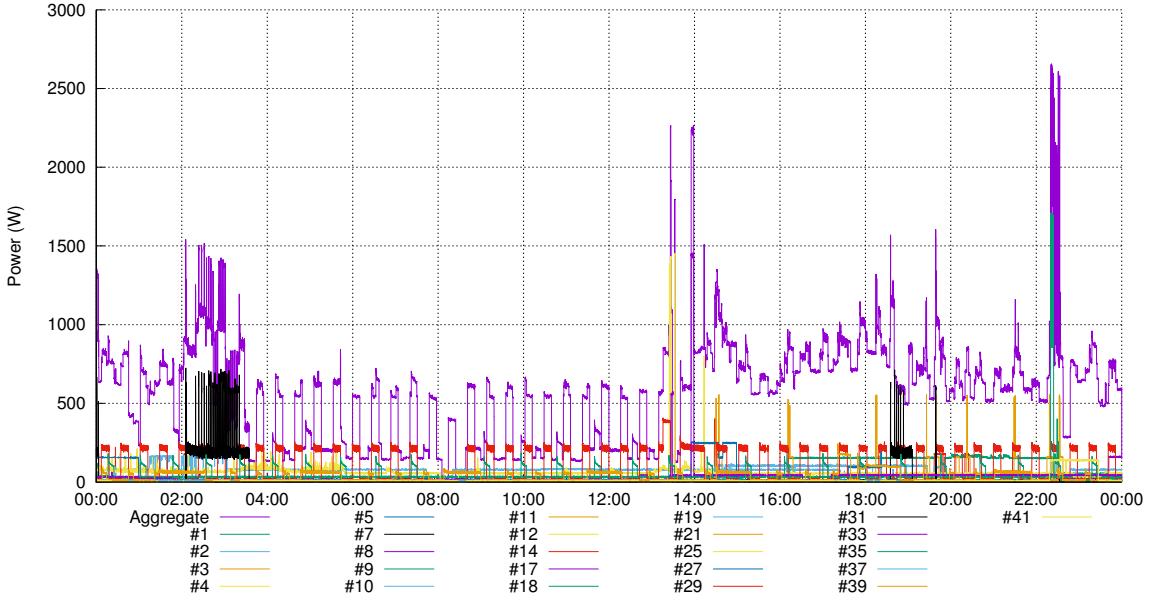


Figure 9.3: Circuit-level power breakdown for a house over the course of one day. The time axis is in GMT/UTC.

9.1.2.1 Dataset Metrics

In total, we collected 1,693,966 measurements over a 30 day span. Figure 9.5 shows the breakdown of the measurement count across the circuits. Most sensors reported about 50,000 measurements, with a few never reporting. Some sensors were on high-power circuits or the aggregate and were able to harvest and transmit more reliably.

To get a sense of the completeness of the data set, we inspect the packet frequency of the two aggregate meters. It may be reasonable to assume that the aggregate power draw (or half of the aggregate power draw) of a home is enough to power a Triumvi even as loads are switching on and off. For the aggregate meters, we consider the meter to be active if it sends a packet at least once every five minutes. For aggregate meter “A2”, we find that over the course of the deployment it was active 56.8% of the total time. For “A1”, the active time was 40.8%. These numbers are lower than anticipated. This could be due in part to the gateway intermittently failing, the Triumvi meters intermittently failing, the RF environment changing and causing different packet reception rates, the home periodically not drawing enough energy for Triumvi to harvest, or, likely, some combination. Additional investigation is required to better isolate this issue.

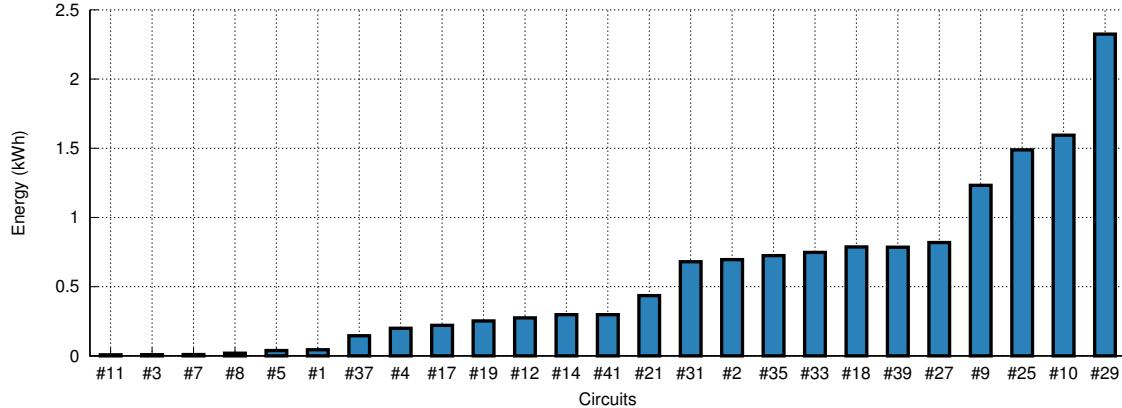


Figure 9.4: Energy breakdown by circuit for a house over one day. With Triumvi, the power measurements can be used to disaggregate energy based on each circuit.

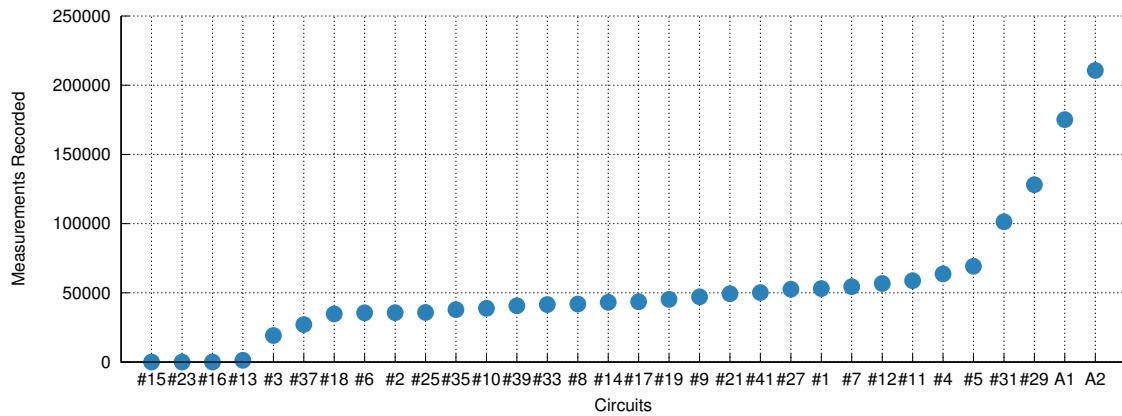


Figure 9.5: Number of measurements recorded for each circuit. Three circuits reported no measurements, and circuit #13 reported very few (1,281).

9.1.2.2 Triumvi Activation Rates

As with the other energy-harvesting systems in this work, we measure the activation rate of the Triumvi sensors deployed in the home. Figure 9.6 shows the median interval between packets for each Triumvi on each circuit. Typically, each meter transmits every 3.4 seconds (equating to a 0.3 Hz activation rate). Even the meters which rarely activate have this median frequency. This suggests that Triumvi’s internal rate limiting is large enough that the minimum threshold for harvesting is enough to sustain the 0.3 Hz rate. We also observe very short intervals between packets, which is likely due to reception errors, and very long intervals likely due to times when the load is off or drawing so little power that Triumvi cannot harvest.

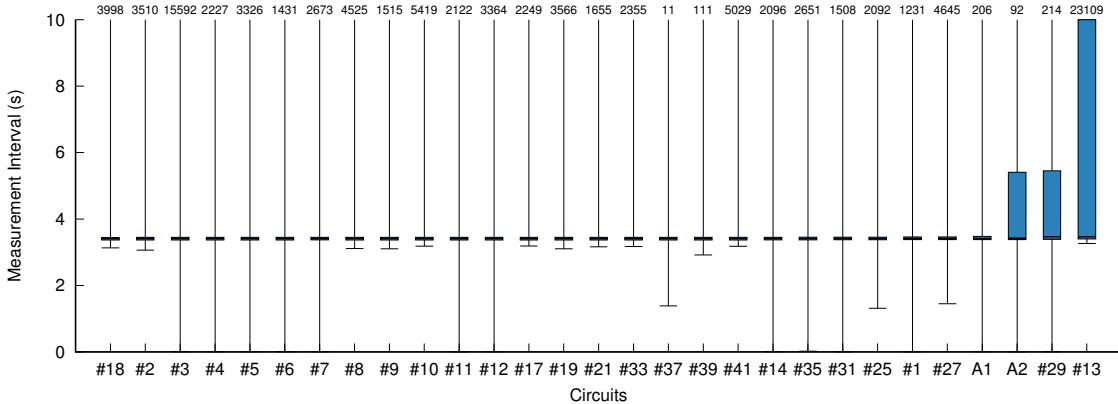


Figure 9.6: Observed measurement rate of the Triumvis on each circuit. Shown are the minimum, first quartile, median, third quartile, and maximum of the intervals between samples from each circuit. Typically each sensor reports every 3.4 seconds. However, the maximum interval is significantly higher for many circuits because of intermittent loads.

9.1.2.3 Sensor Participation

We observe the active time for each Triumvi as defined by the window of time between the first measurement from the device and the last. Figure 9.7 shows a bar for each Triumvi when we saw measurements from that sensor. Figure 9.8 shows the effective length of each bar in days. Most sensors were active across the entire deployment, while some were only active for a limited period (approximately a week) and others were only active at the start of the deployment. This could be due to sensor failure or the onboard battery exhausting its charge on an unused or rarely used circuit. More investigation is required to find the actual cause.

9.1.2.4 Energy Calculation

To look further into how well Triumvi can measure total energy, we analyze the energy audits from the apartments. In a more constrained test we are able to include a ground truth meter to which we compare the results. Figure 9.9 shows the energy breakdown for the three apartments. Overall, our system displays 3.7% error when measuring energy. Most of the error is from underestimating the actual energy use, which may be attributable to our prototype using a CT rated for 100 A which has low accuracy for very low primary loads. A larger toolkit of core Triumvi devices scaled for different amperage circuits may address this issue, and constitutes future work.

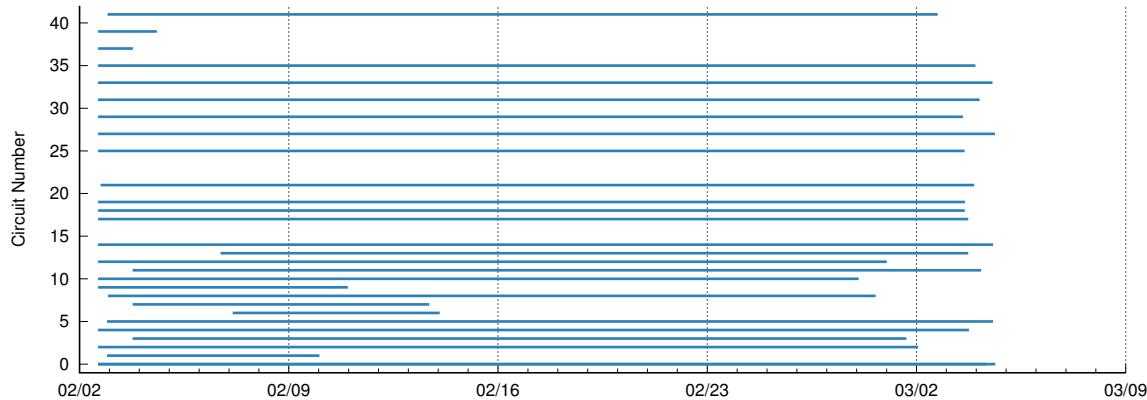


Figure 9.7: Time ranges over which the Triumphi sensors report readings. Each horizontal bar indicates the time span over which the sensor reported measurements. The data missing from even numbered circuits between 20 and 40 are due to the absence of circuit breakers (and circuits) at those positions.

9.2 Monjolo Indoor Light Sensing

The second sensor we evaluate through real-world deployments is the indoor lighting Monjolo sensor. In contrast to the AC Monjolo, which suffers from calibration issues, and the thermoelectric Monjolo, which suffers from saturation issues, the lighting Monjolo is an effective sensor for monitoring lighting. Because the meter is small, it can be deployed unobtrusively very close to a light. This greatly increases its harvesting potential while also limiting its activations to only when the light is on. Also, many lights have only two states: on and off. Therefore, light state detection can be achieved by just observing the presence of packets from the sensor. This can be expanded to detecting energy by recording the manufacturer’s stated power draw of the light bulb and multiplying that figure by the amount of time the light is powered. Further, since there may be many lights in a building but individual switches often control many lights, entire banks of lights can be metered by a single Monjolo sensor.

To see how this design works in practice, we develop an implementation of the Monjolo design called Ligeiro and shown in Figure 9.10. Ligeiro includes a version of the power supply described in Chapter 3 and a Bluetooth Low Energy (BLE) radio for sending activation packets. We use BLE for its combination of low energy operation that the Monjolo sensor can support and for its interoperability with smartphones. The sensor is housed in a case that is magnetically attached to a light fixture for ease of installation.

We deploy 42 Ligeiro sensors for approximately a month to observe how this sensing technique works in practice.

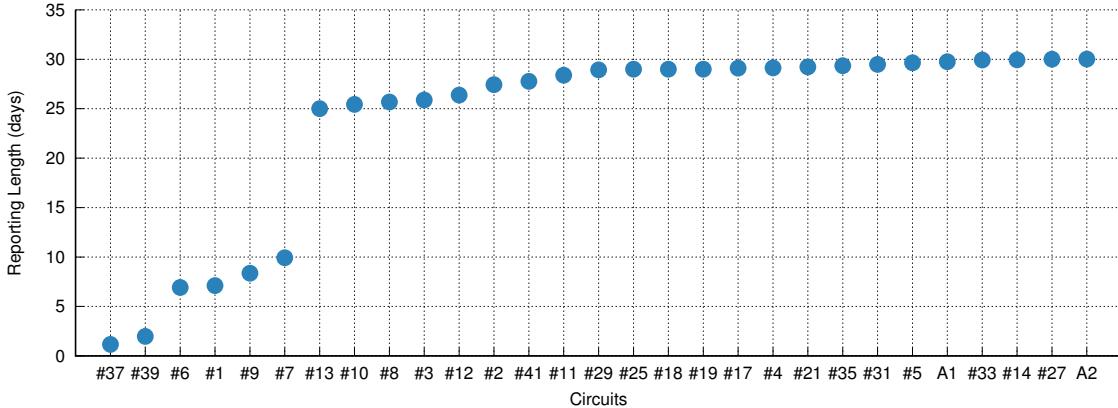


Figure 9.8: Reporting time for each circuit. The time is calculated as the delta between the first packet seen from the sensor and the last packet seen from the sensor in the dataset. We consider the length of the experiment to be the longest reporting time (30 days). Most circuits reported at least occasionally for 25 days or more. Six circuits reported over a much shorter window, which could be the result of sensor failure or circuits with rarely used loads.

9.2.1 Installation Overhead

Installing a Ligeiro sensor is as simple as sticking it to a ferromagnetic surface near a light with the solar panel facing the light. If the deployment requires using the bulb wattage to estimate energy, the wattage of each bulb monitored by the Ligeiro must be recorded as well. For this, we leverage the fact that bulbs are labeled by their manufacturer with a rated wattage.

9.2.2 Sensor Performance

With Ligeiro sensors we can determine when a light is on and how much time a light is on per unit time. These data streams can feed various applications such as occupancy detection and energy disaggregation. For the 42 deployed Ligeiros we calculate the amount of time that the attached light is on. The results are shown in Figure 9.11. Six of the lights are never turned on during the deployment. We include them because we do observe packets from the sensors, but the transmissions are very sporadic and are likely from ambient or natural light and not from the attached light. A majority (21) of the lights spend on average less than an hour on per day across the deployment period. The remainder (15) are on for more than an hour a day, with sensors “2e” and “1a” on for 5 hours and 9.4 hours per day, respectively.

This data can be used to estimate the cost of using these lights in terms of energy costs, and can be used to analyze the payback period for upgrading lights. For instance, estimating \$0.1222 per kWh (the national average in the U.S [35]), and 60 W for an incandescent light

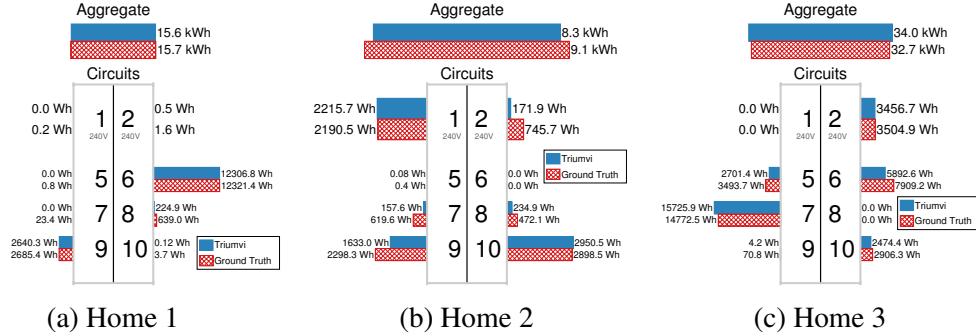


Figure 9.9: Short-term deployments in three homes. The energy consumed by each circuit in three panel boxes and the aggregate was measured with our system and a ground truth meter over the course of approximately 2.5 days. Across all of the circuits, our system shows an average error of 3.7%, typically underestimating the correct value. This level of accuracy is sufficient for many energy audits, and is capable of providing insight on where additional metering may be warranted.



Figure 9.10: The Ligeiro implementation of the indoor lighting Monjolo sensor.

bulb, 14 W for a fluorescent light bulb, and 7 W for an LED light bulb, we can compare the yearly costs of running each of the lights in the deployment with the different lighting technologies. Figure 9.12 shows the cost of using each light in the study (assuming it is or is equivalent to a 60 W incandescent light) for a year with the three main types of lighting. To analyze the payback period for upgrading to LED lighting, assume the replacement LED light bulb is \$5.00. For the light that is on for over seven hours per day, moving from incandescent to LED would have a payback period of just over three months. Going from fluorescent to LED would require about 2.1 years to recover the cost of the bulb, which is well within the lifetime of an LED bulb. As a point of comparison, consider the light attached to sensor “24”. That light is on for 1.5 hours per day on average, and would require about 1.4 years to pay back the switch from an incandescent bulb to LED, and about ten years to move from fluorescent to LED. If the bulb is already fluorescent, it is likely not financially worth the switch in that instance.

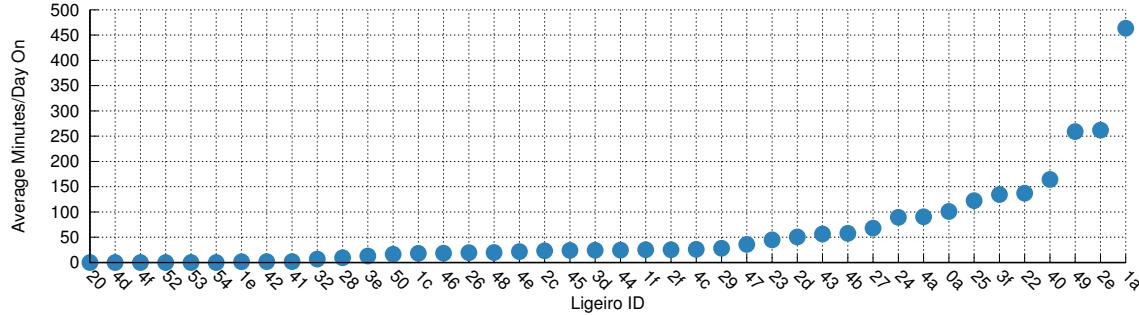


Figure 9.11: For each sensor we estimate how long the light to which it is attached is powered per day. We assume a light is on if we receive two packets from the Ligeiro in the same minute, and we average over the total number of days between the first packet we saw and the last.

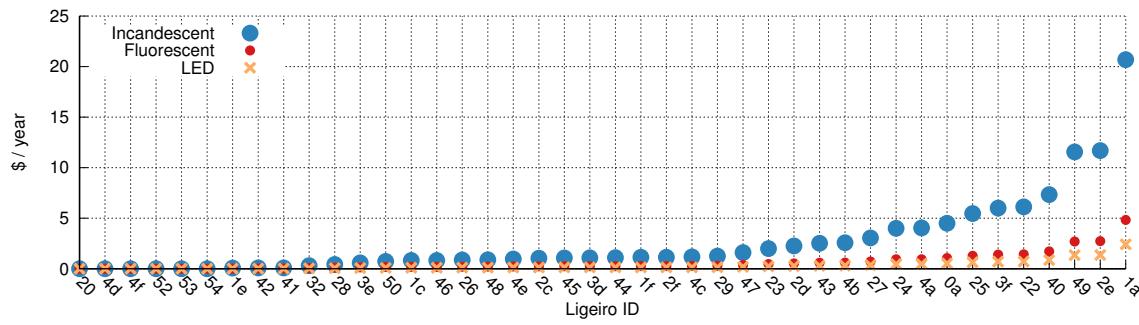


Figure 9.12: Using the estimate of daily on time, we then compare the cost of running the light at that duty-cycle for a year using three different lighting types. For comparison we assume all lights are a 60 W incandescent equivalent. This type of analysis can be used to determine which lights are economically viable to upgrade given a certain payback period.

This type of analysis can accurately guide capital expenditures relating to lighting upgrades. The sensors, by being easy to retrofit into existing buildings, can enable data driven decision making without the cost of expensive audits or the guesswork of estimating usage and on-time.

9.2.2.1 Packet Reception Rate

To understand the performance of these energy-harvesting sensors, we use a sequence number in the transmitted packet to detect dropped packets and calculate the fraction of transmitted packets that are received by the gateways, also known as the packet reception rate (PRR). Figure 9.13 shows the PRR from each sensor. Given the limited energy budget of the sensor and the broadcast-only communications, we expect some packet loss. We see that the reception rate ranges from 4% to 83%.

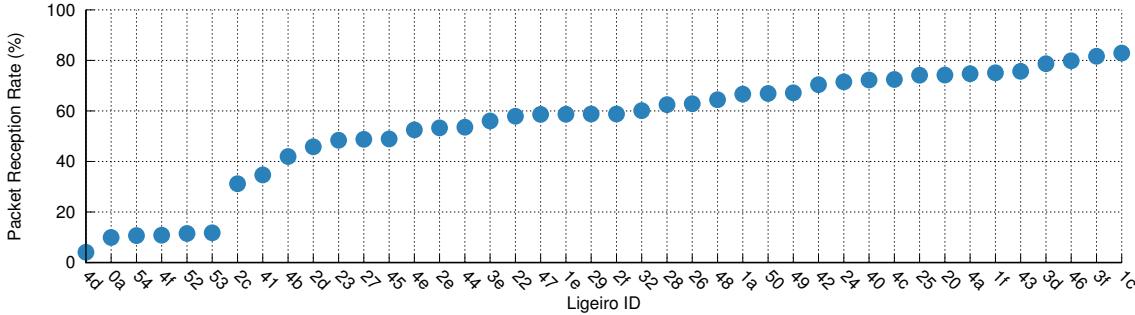


Figure 9.13: Overall packet reception ratio (PRR) for each Ligeiro node. Ligeiros transmit an incrementing sequence number in each packet, and we use this to determine the percentage of packets we received from each sensor. The variability in reception rates is likely due to interference in the wireless spectrum, the range between the sensor and the gateway, and intermittent gateway failures.

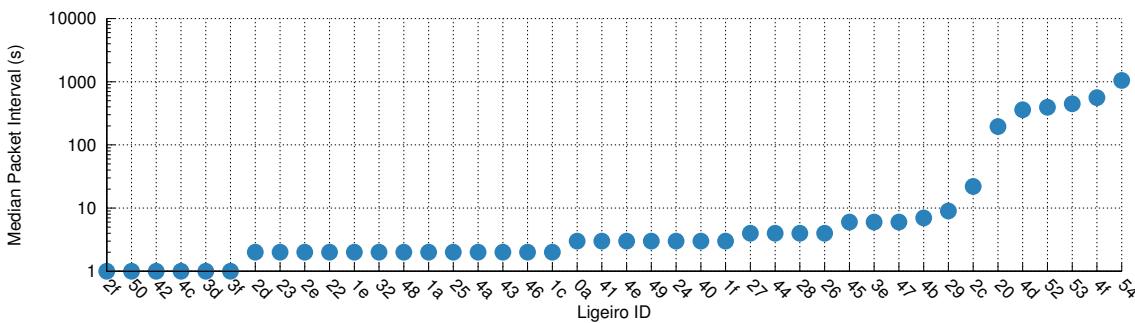


Figure 9.14: We observe the median transmission interval of each sensor to understand how often the sensor can transmit when the light is on. Note the y-axis is on a log scale. For most sensors the Ligeiros can transmit more frequently than once every 10 seconds. The sensors that exhibit activations at a slower rate likely are attached to lights that are infrequently or never used.

We also again observe the activation rate of the energy-harvesting sensors. Figure 9.14 shows the median packet transmission interval for each sensor during the entire deployment. Note that the y-axis is on a log scale. Most of the sensors have a median transmission interval of 10 seconds or less. This is expected as most packets are transmitted when the light is on and therefore the harvesting source is plentiful. However, the sensors that were placed on lights that were never detected to turn on exhibit much higher median intervals as they were likely harvesting with sporadically available ambient light. This discrepancy is essential for classifying the light as on or off using just the activation rate of the sensor.

We also compare the median received interval to light usage, and show the correlation in Figure 9.15. For sensors with relatively frequent packet reception rates it is difficult to discern any information about the attached light. However, for sensors with long gaps

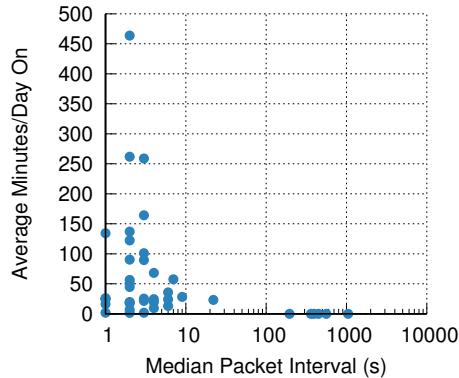


Figure 9.15: Comparing median transmission time to estimate light usage. For sensors with median packet reception intervals that are below about 10 s, it is difficult to correlate with light usage. However, if a sensor's packets are received very rarely and the median interval is much higher (greater than 100 s), as expected the light can be assumed to be used very little or never.

between received packets, the light can be assumed to be unused.

To get a sense of the coverage of the Ligeiro data in time, we investigate the number of consecutive packets that are transmitted from each sensor but not received by a gateway. This allows us to understand if the Ligeiro data is still reliable even with the relatively low packet reception rates. For example, if a sensor had a PRR of 50%, it is possible that nearly all of the packets were received during the first half of the study and very few were collected in the second half. This would indicate that the load on time averages are incorrect due to long periods of missed packets. However, if the packets are dropped very intermittently, the data processing backend can compensate. For example, when packets are rapidly being transmitted because the light is on, dropping packets in that interval can be compensated for by observing the packet rate over the interval in which the packet was dropped.

Figures 9.16 and 9.17 show how many consecutively missed packets are observed from each sensor. In Figure 9.16 we see the average burst length of the packet losses as well as the standard deviation in the error bars. On average, fewer than ten packets are dropped before successfully receiving from a sensor. Figure 9.17 shows a histogram of the number of consecutively dropped packets for each device. There is a long tail of dropped packets, likely due to a malfunctioning gateway at one of the deployment sites, but the majority of outages last for less than 50 packets.

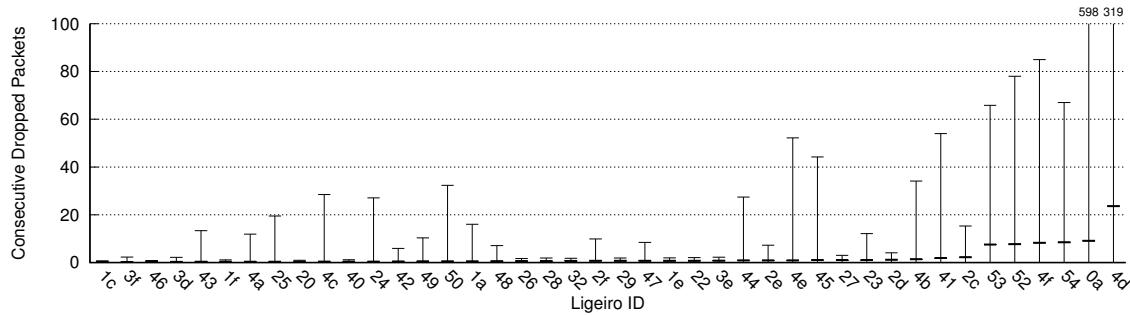


Figure 9.16: Average and standard deviation of consecutively dropped packets. The bold lines are the average number of packets that are dropped consecutively, and the error bars is the standard deviation. Nearly all of the Ligeiros on average had fewer than 10 packets dropped before a successful packet reception. This shows that packet failures are intermittent and prolonged outages are not common.

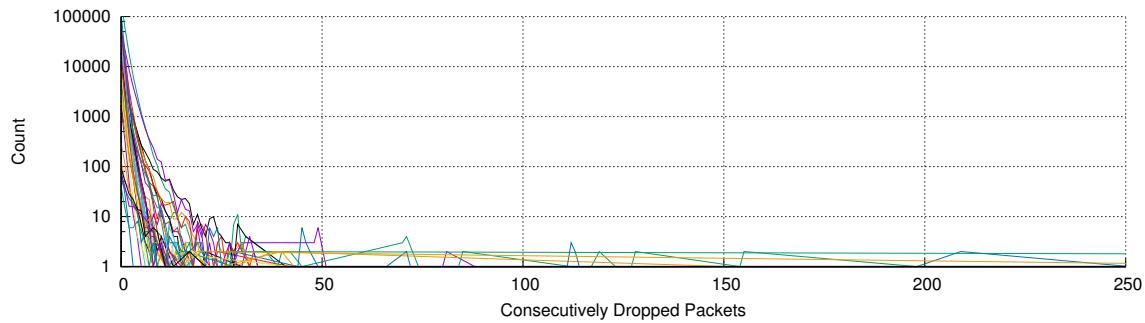


Figure 9.17: Histogram of the number of packet drop lengths seen by each sensor. Generally, no more than 50 packets are dropped consecutively.

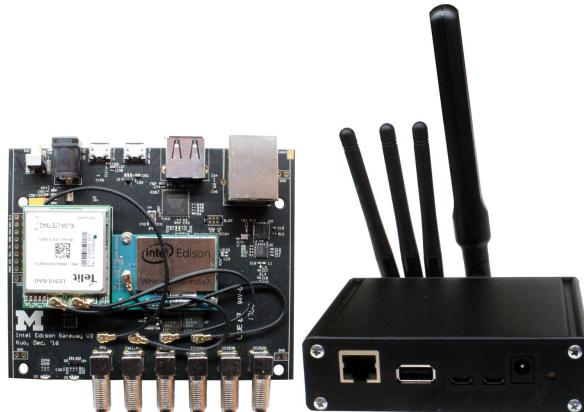


Figure 9.18: Multi-radio gateway hardware and case. The gateway is powered by an Intel Edison, and contains IEEE 802.15.4, BLE, Wi-Fi, GSM, and GPS wireless radios. It also supports Ethernet, micro SD cards, and USB, and includes two status LEDs.

9.3 Gateway Support Infrastructure

Deployments of these energy-harvesting sensors require a nearby gateway to collect, store, analyze, archive, or retransmit packets for the sensors. Figure 9.18 shows the latest hardware version of our gateway platform. The gateway is responsible for accepting packets on both Bluetooth Low Energy (BLE) and IEEE 802.15.4 low-power wireless interfaces, and can relay those packets to the cloud using Ethernet, Wi-Fi, or GSM radios.

9.3.1 Single-hop Networking

The energy-harvesting sensors we have deployed rely on a single-hop link to a gateway. This requirement stems from the limited energy stores on the energy-harvesting nodes and the optimizations to the operation of these sensors.

While traditional wireless sensor networks envisioned gateways being difficult or expensive to deploy, particularly in outdoor or remote environments, in frequently occupied buildings this is no longer the case. The main limiting factor to deploying a gateway in buildings is mains power availability. We have found that sharing an outlet with existing equipment (e.g. projectors, Wi-Fi routers) provides a convenient method for powering gateways. The other need a gateway has is an Internet connection for configuration, updates, and data backhaul. Today, in buildings this is often not a challenge. Wi-Fi or Ethernet are often ubiquitous. In the event they are not, our gateway also supports a cellular Internet connection.

Using single-hop networking also dramatically simplifies the network operation and overhead of deploying a new sensor. The gateways must simply be always-on listeners to collect any transmitted packets. They do not need to coordinate a network, assign addresses, or facilitate routing. The sensors then do not have to maintain schedules or routing tables, they can simply transmit when their energy permits. This reduces the complexity of the sensor software. It also aids new device deployment as the node does not need to be configured with a series of network- or deployment-specific parameters that are common to more advanced network and routing protocols. This approach does require a comprehensive deployment of gateways to support the deployed devices. However, we anticipate that future Wi-Fi access points will include support for low power radios (and indeed commercial products with this feature are already starting to appear) and that an additional deployment of gateways will not be required.

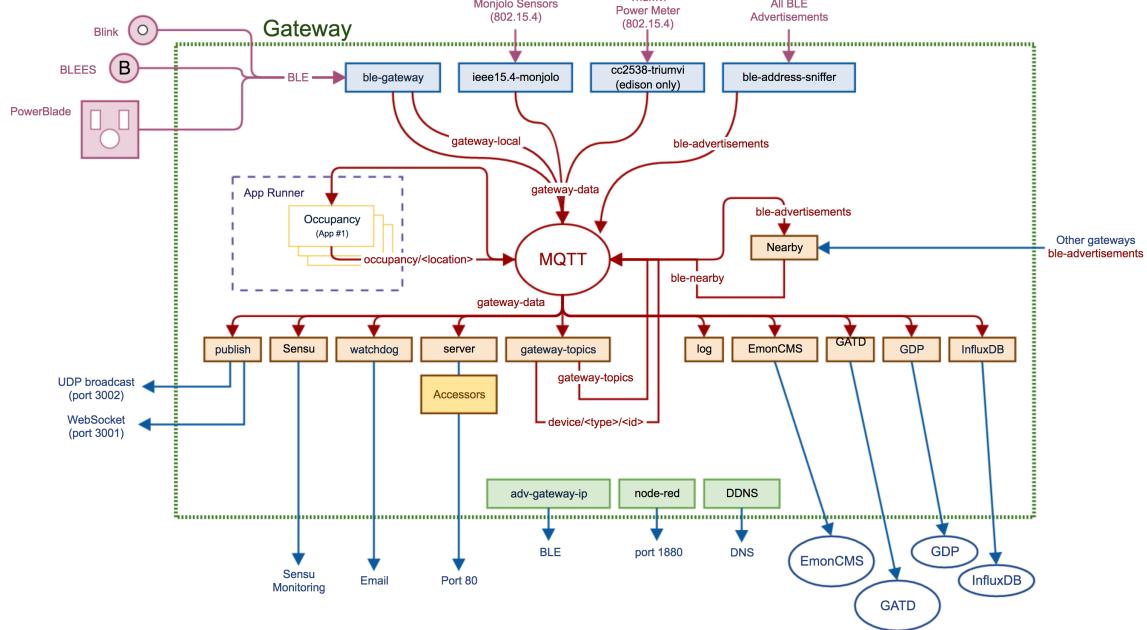
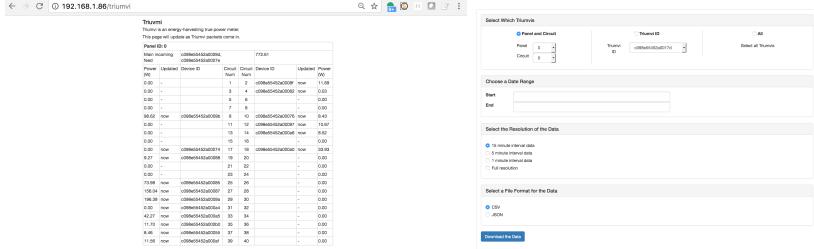


Figure 9.19: Software architecture for the gateway. At the top, a number of receiver blocks are capable of accepting packets from a wide array of devices. The MQTT broker on the gateway handles routing packets from the receivers, to various processing blocks, and out to external services or databases. Additionally, other services running on the gateway provide interfaces for discovering gateways, writing applications, and interacting with devices.

9.3.2 Gateway Software Architecture

The gateway software architecture is designed to be a flexible platform to support many different types of sensors and deployment needs. The architecture, diagrammed in Figure 9.19, is centered around an MQTT broker and supports many data input receivers, onboard processing blocks, and remote publishing protocols.

The gateway's main purpose is to receive packets from deployed devices. Typically, there are two main challenges for doing this. The first is ensuring the gateway and device speak the same protocol, and the second is ensuring the gateway can understand the data format the device is using. A typical approach is to have a well specified protocol and data format that the device can follow to ensure that it is interoperable with the gateway. This is problematic, however, because the breadth of existing devices makes it challenging, if not impossible, for a single standard to emerge. To compensate, our gateway takes a different approach. Our gateway understands a range of wireless protocols and makes no assumption about the data format that individual devices use. Instead, devices point to a small snippet of code, in our case JavaScript, that can parse and interpret the data from the node and convert it to the internal data format of the gateway. This then allows devices to effectively configure



(a) Deployment UI

(b) Data Download

Figure 9.20: Gateway-hosted webpages. The first shows circuits and Triumvis in real-time. This helps with deployment by showing which Triumvis are reporting for which circuits. The second allows for locally stored data to be downloaded for offline processing.

the gateway at runtime with the tool needed to understand their data.

Ideally, device creators would author these conversion files. However, that is not a requirement and the gateway can support third party converters to help build a critical mass of supported devices.

Once the data are collected and parsed, the gateway can push them to other internal processing blocks, to applications running on the gateway, or to cloud-based services for monitoring, archival, or visualization. For Triumvi, the gateway supports two key features. First, the gateway hosts a webpage (shown in Figure 9.20a) that shows which Triumvis are reporting and which circuits they are attached to. This helps when deploying to make sure things are responding as intended. Second, the gateway locally logs Triumvi data so that it can be stored and retrieved without the gateway needing to be connected to the Internet. Figure 9.20b shows the web interface for downloading local Triumvi data.

Another service the gateway provides is a method for Triumvi nodes to query the gateway for the current time. This allows Triumvi nodes to reset their onboard real-time clocks (RTC) to the correct time. Triumvis have a supercapacitor backed RTC that can maintain time for about a month. If the supercapacitor's stored energy is depleted, the RTC will lose its sense of time and the Triumvi will ask the gateway for an updated timestamp. Because Triumvi has limited runtime due to its energy-harvesting power supply, the gateway must be able to respond to a time request very quickly while the Triumvi still has energy to receive the packet. To ensure this, the gateway uses a co-processor with a built-in wireless radio. The co-processor handles returning the correct time to the Triumvi, and because the co-processor includes the radio it can ensure a quick turnaround between receiving the request and transmitting the correct time.

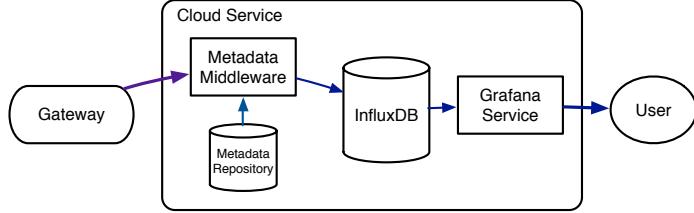


Figure 9.21: Cloud service architecture. The gateway forwards collected packets to the cloud service, which uses a metadata repository to add metadata to incoming messages, and stores data in an InfluxDB database. A user can then query the database through the Grafana graphing and visualization interface.

9.3.3 Data Collection, Processing, and Visualization

Once data is collected by a gateway, it is often pushed to the cloud for storage, processing, and visualization. The architecture for the cloud service is shown in Figure 9.21. To continue to ensure that devices are easy to deploy, our infrastructure does not require any configuration when a new device is added. The parsed data packets from the device are transparently stored in the cloud.

We use InfluxDB [56] as our primary cloud data storage. We chose InfluxDB because it is designed for timeseries data and it supports multiple data types including booleans and strings. The support for multiple data types is essential for us to ensure generality for arbitrary devices. While InfluxDB is an effective time series data store, it has no mechanism for appending metadata to incoming records. Often, it is beneficial to store data about the deployed device, such as where it is or what it is sensing. Rather than distribute metadata to each gateway and have the gateway append it to each data packet, we add middleware to the cloud receiver for InfluxDB. This middleware inspects each incoming packet and checks if there is any metadata for the device that generated the packet, and if so, appends the correct metadata before inserting the record into the database.

To visualize data we use Grafana [47], a platform for visualizing time series data. Grafana pulls directly from InfluxDB and makes it easy to create graphs, tables, and other charts from the device data streams. An example graph is in Figure 9.22. These graphs can be built up into dashboards that allow at-a-glance monitoring of multiple aspects of a deployment.

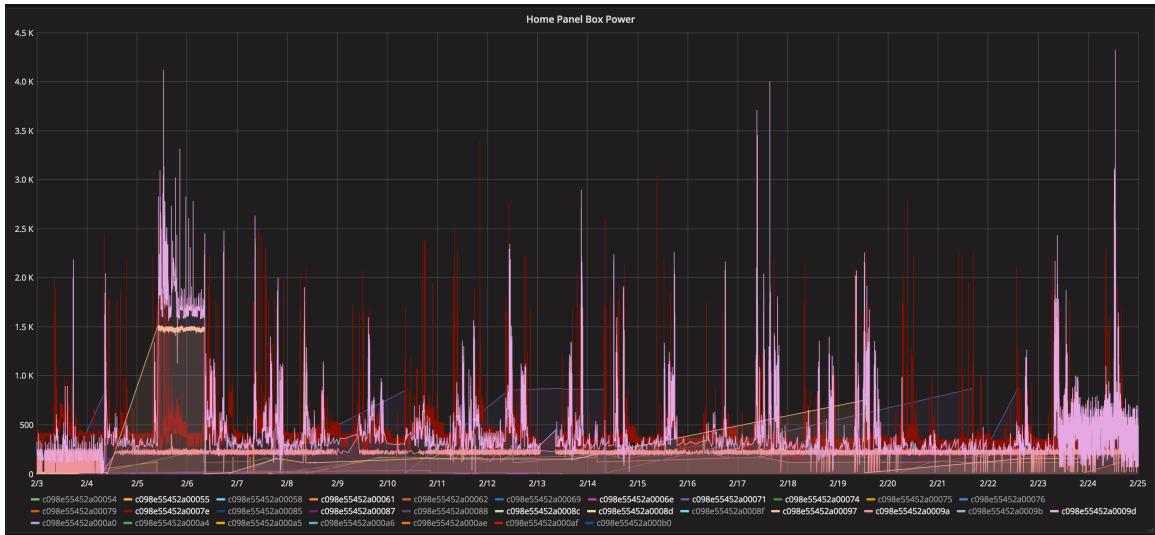


Figure 9.22: Cloud-based graphing software. An example graph of Triumvi power data using the Grafana graphing application.

CHAPTER 10

Conclusion

Ubiquitous computing, the Internet of Things, wearables, industrial monitoring and their driving applications are going to continue to create a proliferation of sensors in the built environment. Effectively using these devices, and extracting the value they promise, is going to require overcoming many challenges related to how the devices are deployed, programmed, networked, and integrated. Another key challenge that is going to continue to grow in importance as these devices continue to significantly outnumber people, is how they are going to be powered.

Using AC mains power for devices is effective, but difficult to scale, due to the limited number of plugs, the limited location of outlets, and the physical size of AC-DC power supplies. Batteries provide much more flexibility, but require periodic maintenance when their energy store is depleted. As the number of devices grows, this maintenance overhead will be untenable. This makes a third option, energy-harvesting, an attractive one. With energy-harvesting, devices scavenge for their own energy after they are deployed. This can enable perpetual sensing where the device operates for as long as it is installed.

Energy-harvesting, however, also comes with tradeoffs. Scavengable energy is often intermittent, unpredictable, and available only in small quantities. This makes existing sensors and protocols difficult to use, as they typically assume a reliable source of energy. To overcome these challenges, while enabling perpetual sensing, we introduce a series of designs for energy-harvesting systems that enable useful sensing while operating on energy harvested at a rate of just microwatts. The first design, Gecko, uses optimized hardware and software to enable an ambient monitoring sensor that can sample once a minute using energy harvested from indoor lighting. Gecko also demonstrates a technique for using energy-harvesting sensors in conjunction with battery powered mesh networks.

As an energy-harvesting sensor, Gecko's sample rate is tied to the available energy, and this relationship is explored more fully by the Monjolo class of sensors. Monjolo devices monitor energy consumption by observing how rapidly they are able to harvest energy, and

we demonstrate versions that use lighting, AC electricity, and heat sources. To improve the accuracy of the AC Monjolo, we develop Gemini, an energy-harvesting energy meter that uses a virtualized voltage signal to obtain accurate AC energy measurements. Triumvi extends this design into a standalone device with many configuration options to support an array of deployment scenarios.

We also investigate sensors that monitor other aspects of buildings, including door openings and closings and air vent usage. These sensors decouple harvesting from sensing by harvesting from energy sources unrelated to their sensing goal. With this growing toolkit of sensors, we define an architecture, centered around a trigger interface that decouples harvesting from the sensing application, to enable reusable components between energy-harvesting systems. This enables new sensors to be rapidly developed by defining a common physical interface and enabling sharing of key hardware components.

These energy-harvesting sensors not only demonstrate the potential of computing with an energy-harvesting power budget, but are also designed to address a real problem: how do we make existing buildings more energy efficient? All of the devices are designed to operate indoors by scavenging energy available in typical buildings. This sensor suite can then be used to instrument a building to better understand how it is using energy, and, more importantly, where it is *wasting* energy. To show how these sensors perform when deployed, we also perform multiple deployments at modest scale and evaluate the energy-harvesting sensors. We find that they are indeed easy to deploy and capable of continuous data collection, but also that the resulting data streams require additional processing, including detecting when the load is off and converting activation packets to estimated power, to hide the limitations of energy-harvesting sensors.

The systems presented often argue for an extreme design point where the entire sensor runs on a small bank of capacitors. This raises potential questions, however, as to what extent the systems could benefit from additional energy storage, and what physical device or devices should be used for the energy storage? Future work could include exploring more complex designs with a hierarchy of energy storage to enable more sensing applications.

Other avenues for future work include developing new systems that help energy-harvesting sensors—which are often sporadic in their sample rates or limited in their processing capabilities—interact with existing higher-level systems such as machine learning algorithms or stream processing pipelines. This overlay system could run in the cloud or the gateway and would provide a sensor abstraction that upper layers expect but use energy-harvesting sensors as the data sources. Related, often energy-harvesting sensors require the processing that may normally happen on the node itself to be performed elsewhere for energy reasons. However, adding small snippets of processing code is actually often

complicated as user-friendly runtimes for this often do not exist. Having a platform for easily and scalably processing these data streams would also help mask some of the drawbacks of the underlying energy-harvesting sensors.

As with any sensor deployment, even as the devices become simpler to deploy, minimizing the number of required sensors reduces the cost and overhead of the deployment. Additionally, with the promise of perpetual computing comes the potential for the devices to be deployed for long periods of time. The requirements for the deployed devices might change over time as applications and goals change, and the energy-harvesting devices need some provision for altering their operation to match. Doing this on an energy-harvesting platform, however, is not straightforward and will require new techniques suitable for low energy systems.

While single-hop, unreliable networking is appealing from a simplicity standpoint and helps enable energy-harvesting based data collection, packet drops lead to lost data and their cause is difficult to diagnose. The receiver, the wireless channel, and the sensor could all be responsible and better tools are needed to gain more introspection. Also, the energy and performance tradeoffs among reliable (or not) packet transmission, local packet storage, local processing and compression, data timeliness, and data loss need to be further explored. Perhaps the redundancy in the unreliable, fire-and-forget operation used by the nodes in this dissertation is sufficient for the applications they target. Alternatively, it may be more effective to periodically ensure reliable data transfer, even if this requires the node to be able to timestamp locally stored packets.

At most the sensors in this dissertation encrypt their payloads with an AES cipher, but future devices will need a more robust approach for security and privacy. However, implementing this on the limited energy budget of an energy-harvesting sensor will require new techniques, potentially integrating checkpointing schemes for energy-loss tolerant processing and support from the nearby gateway. Bootstrapping and commissioning will also present challenges as accruing enough energy for a typical key exchange or setup procedure may prove to be a lengthy process.

Likely future ubiquitous energy-harvesting sensor deployments will require new processing, management, and security frameworks, multi-level energy storage, and reusable hardware platforms to be successful, but the systems presented in this dissertation are a start towards not relying on batteries yet enabling key applications at the scales we expect.

BIBLIOGRAPHY

- [1] F. Alfayez, M. Hammoudeh, and A. Abuarqoub. A survey on mac protocols for duty-cycled wireless sensor networks. *Procedia Computer Science*, 73:482 – 489, 2015. URL: <http://www.sciencedirect.com/science/article/pii/S187705091503495X>, doi:<http://dx.doi.org/10.1016/j.procs.2015.12.034>.
- [2] Analog Devices. ADE7753 Datasheet.
http://www.analog.com/static/imported-files/data_sheets/ADE7753.pdf.
- [3] M. P. Andersen, G. Fierro, and D. E. Culler. Enabling synergy in iot: Platform to service and beyond. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 1–12, April 2016. doi:[10.1109/IoTDI.2015.45](https://doi.org/10.1109/IoTDI.2015.45).
- [4] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*, 8(4):34:1–34:33, Sept. 2012. URL: <http://doi.acm.org/10.1145/2240116.2240123>, doi:[10.1145/2240116.2240123](https://doi.org/10.1145/2240116.2240123).
- [5] B. Balaji, H. Teraoka, R. Gupta, and Y. Agarwal. Zonepac: Zonal power estimation and control via HVAC metering and occupant feedback. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys’13, pages 18:1–18:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528304>, doi:[10.1145/2528282.2528304](https://doi.org/10.1145/2528282.2528304).
- [6] S. P. Beeby, R. Torah, M. Tudor, P. Glynne-Jones, T. O’donnell, C. Saha, and S. Roy. A micro electromagnetic generator for vibration energy harvesting. *Journal of Micromechanics and microengineering*, 17(7):1257, 2007.
- [7] Belkin WeMo Insight. <http://www.belkin.com/us/p/P-F7C029/>.
- [8] A. Beltran, V. L. Erickson, and A. E. Cerpa. Thermosense: Occupancy thermal based sensing for hvac control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys’13, pages 11:1–11:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528301>, doi:[10.1145/2528282.2528301](https://doi.org/10.1145/2528282.2528301).

- [9] J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund, and L. Thiele. Prototyping wireless sensor network applications with btnodes. In *European Workshop on Wireless Sensor Networks*, pages 323–338. Springer, 2004.
- [10] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys ’06, pages 307–320, New York, NY, USA, 2006. ACM. URL: <http://doi.acm.org/10.1145/1182807.1182838>, doi:10.1145/1182807.1182838.
- [11] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: Ultra-low power data gathering in sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, IPSN ’07, pages 450–459, New York, NY, USA, 2007. ACM. URL: <http://doi.acm.org/10.1145/1236360.1236417>, doi:10.1145/1236360.1236417.
- [12] B. Campbell and P. Dutta. An energy-harvesting sensor architecture and toolkit for building monitoring and event detection. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys ’14, pages 100–109, New York, NY, USA, 2014. ACM. URL: <http://doi.acm.org/10.1145/2674061.2674083>, doi:10.1145/2674061.2674083.
- [13] B. Campbell and P. Dutta. Gemini: A non-invasive, energy-harvesting true power meter. In *Real-Time Systems Symposium (RTSS), 2014 IEEE*, pages 324–333, Dec 2014. doi:10.1109/RTSS.2014.36.
- [14] B. Campbell, B. Ghena, and P. Dutta. Energy-harvesting thermoelectric sensing for unobtrusive water and appliance metering. In *Proceedings of the 2Nd International Workshop on Energy Neutral Sensing Systems*, ENSsys ’14, pages 7–12, New York, NY, USA, 2014. ACM. URL: <http://doi.acm.org/10.1145/2675683.2675692>, doi:10.1145/2675683.2675692.
- [15] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang. Design considerations for distributed microsensor systems. In *Proceedings of the IEEE 1999 Custom Integrated Circuits Conference (Cat. No.99CH36327)*, pages 279–286, 1999. doi:10.1109/CICC.1999.777291.
- [16] M. Clark, B. Campbell, and P. Dutta. Deltaflow: Submetering by synthesizing uncalibrated pulse sensor streams. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy ’14, pages 301–311, New York, NY, USA, 2014. ACM. URL: <http://doi.acm.org/10.1145/2602044.2602070>, doi:10.1145/2602044.2602070.
- [17] CR Magnetics. CR2550 Series Low Cost Remote Current Indicator. <http://www.crmagnetics.com/Products/Assets/ProductPDFs/2550.pdf>.

- [18] Cube Sensors. <https://cubesensors.com/>.
- [19] D. Culler, D. Estrin, and M. Srivastava. Guest editors' introduction: Overview of sensor networks. *Computer*, 37(8):41–49, Aug. 2004. URL: <http://dx.doi.org/10.1109/MC.2004.93>, doi:10.1109/MC.2004.93.
- [20] Cypress FM25L04B-DG. <http://www.cypress.com/?mpn=FM25L04B-DG>.
- [21] S. DeBruin, B. Campbell, and P. Dutta. Monjolo: An energy-harvesting energy meter architecture. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, 2013.
- [22] S. DeBruin, B. Ghena, Y.-S. Kuo, and P. Dutta. Powerblade: A low-profile, true-power, plug-through energy meter. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 17–29, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2809695.2809716>, doi:10.1145/2809695.2809716.
- [23] S. DeBruin, J. Grunnagle, and P. Dutta. Scaling the wireless AC power meter. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 153–154, New York, NY, USA, 2012. ACM. URL: <http://doi.acm.org/10.1145/2185677.2185724>, doi:10.1145/2185677.2185724.
- [24] M. Demirbas, O. Soysal, and M. Hussain. A singlehop collaborative feedback primitive for wireless sensor networks. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, April 2008. doi:10.1109/INFOCOM.2008.270.
- [25] Digikey. KEMET T491X107K025AT. <http://www.digikey.com/product-detail/en/kemet/T491X107K025AT/399-8410-1-ND/3472133>.
- [26] A. Dunkels. Full tcp/ip for 8-bit architectures. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 85–98, New York, NY, USA, 2003. ACM. URL: <http://doi.acm.org/10.1145/1066116.1066118>, doi:10.1145/1066116.1066118.
- [27] A. Dunkels. The contikimac radio duty cycling protocol, 2011.
- [28] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, LCN '04, pages 455–462, Washington, DC, USA, 2004. IEEE Computer Society. URL: <http://dx.doi.org/10.1109/LCN.2004.38>, doi:10.1109/LCN.2004.38.
- [29] P. Dutta and D. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Sensys '08: Proceedings of the 6th International Conference on Embedded Networked Sensor Systems*, pages 71–84, Nov. 2008.

- [30] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. A-mac: A versatile and efficient receiver-initiated link layer for low-power wireless. *ACM Trans. Sen. Netw.*, 8(4):30:1–30:29, Sept. 2012. URL: <http://doi.acm.org/10.1145/2240116.2240119>, doi:10.1145/2240116.2240119.
- [31] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In *2006 5th International Conference on Information Processing in Sensor Networks*, pages 407–415, April 2006. doi:10.1145/1127777.1127839.
- [32] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensornet systems. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys ’08, pages 267–280, New York, NY, USA, 2008. ACM. URL: <http://doi.acm.org/10.1145/1460412.1460439>, doi:10.1145/1460412.1460439.
- [33] W. P. Eaton and J. H. Smith. Micromachined pressure sensors: review and recent developments. *Smart Materials and Structures*, 6(5):530, 1997. URL: <http://stacks.iop.org/0964-1726/6/i=5/a=004>.
- [34] C. T. Ee, R. Fonseca, S. Kim, D. Moon, A. Tavakoli, D. Culler, S. Shenker, and I. Stoica. A modular network layer for sensorsets. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI ’06, pages 249–262, Berkeley, CA, USA, 2006. USENIX Association. URL: <http://dl.acm.org/citation.cfm?id=1298455.1298479>.
- [35] Average Price of Electricity to Ultimate Customers by End-Use Sector. https://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a.
- [36] A. El-Hoiydi and J.-D. Decotignie. Low power downlink mac protocols for infrastructure wireless sensor networks. *Mob. Netw. Appl.*, 10(5):675–690, Oct. 2005. URL: <http://dl.acm.org/citation.cfm?id=1160143.1160152>.
- [37] M. F. El-Kady and R. B. Kaner. Scalable fabrication of high-power graphene micro-supercapacitors for flexible and on-chip energy storage. *Nature communications*, 4:1475, 2013.
- [38] M. F. El-Kady, V. Strong, S. Dubin, and R. B. Kaner. Laser scribing of high-performance and flexible graphene-based electrochemical capacitors. *Science*, 335(6074):1326–1330, 2012. URL: <http://science.sciencemag.org/content/335/6074/1326>, arXiv:<http://science.sciencemag.org/content/335/6074/1326.full.pdf>, doi:10.1126/science.1216744.
- [39] EmonCMS.org. <https://emoncms.org>.

- [40] EnOcean Wireless Sensors. <https://www.enocean.com/en/products/>.
- [41] eGauge EG3000 Meter. <http://egauge.net/products/EG3000/>.
- [42] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 4, pages 2033–2036 vol.4, 2001. [doi:10.1109/ICASSP.2001.940390](https://doi.org/10.1109/ICASSP.2001.940390).
- [43] Fitbit One Wireless Activity + Sleep Tracker. <https://www.fitbit.com/one>.
- [44] A. Frye, M. Goraczko, J. Liu, A. Prodhan, and K. Whitehouse. Circulo: Saving energy with just-in-time hot water recirculation. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys’13, pages 16:1–16:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528287>, [doi:10.1145/2528282.2528287](https://doi.org/10.1145/2528282.2528287).
- [45] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’09, pages 1–14, New York, NY, USA, 2009. ACM. URL: <http://doi.acm.org/10.1145/1644038.1644040>, [doi:10.1145/1644038.1644040](https://doi.org/10.1145/1644038.1644040).
- [46] M. Gorlatova, P. Kinget, I. Kymmissis, D. Rubenstein, X. Wang, and G. Zussman. Energy harvesting active networked tags (EnHANTs) for ubiquitous object networking. *IEEE Wireless Communications*, 17(6):18–25, 2010.
- [47] Grafana: The open platform for beautiful analytics and monitoring. <https://grafana.com/>.
- [48] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile. Ieee 802.15.4: a developing standard for low-power low-cost wireless personal area networks. *IEEE Network*, 15(5):12–19, Sept 2001. [doi:10.1109/65.953229](https://doi.org/10.1109/65.953229).
- [49] A. Hande, T. Polk, W. Walker, and D. Bhatia. Indoor solar energy harvesting for sensor network router nodes. *Microprocessors and Microsystems*, 31(6):420 – 432, 2007. Special Issue on Sensor Systems. URL: <http://www.sciencedirect.com/science/article/pii/S0141933107000415>, [doi:https://doi.org/10.1016/j.micpro.2007.02.006](https://doi.org/10.1016/j.micpro.2007.02.006).
- [50] V. Handziski, J. Polastre, J. H. Hauer, C. Sharp, A. Wolisz, and D. Culler. Flexible hardware abstraction for wireless sensor networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, pages 145–157, Jan 2005. [doi:10.1109/EWSN.2005.1462006](https://doi.org/10.1109/EWSN.2005.1462006).
- [51] L. A. Hang-yat and D. Wang. Carrying my environment with me: A participatory-sensing approach to enhance thermal comfort. In *Proceedings of the*

5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, BuildSys'13, pages 21:1–21:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528286>, doi:10.1145/2528282.2528286.

- [52] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy. The platforms enabling wireless sensor networks. *Commun. ACM*, 47(6):41–46, June 2004. URL: <http://doi.acm.org/10.1145/990680.990705>, doi:10.1145/990680.990705.
- [53] J. W. Hui and D. E. Culler. Ip is dead, long live ip for wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 15–28, New York, NY, USA, 2008. ACM. URL: <http://doi.acm.org/10.1145/1460412.1460415>, doi:10.1145/1460412.1460415.
- [54] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2011.
- [55] Ieee standard for local and metropolitan area networks—part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer, April 2012. doi:10.1109/IEEEESTD.2012.6185525.
- [56] InfluxDB: Scalable datastore for metrics, events, and real-time analytics. <https://github.com/influxdata/influxdb>.
- [57] Intel Edison. <http://www.intel.com/content/www/us/en/do-it-yourself/edison.html>.
- [58] Internet of Things Infographic. <http://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png>.
- [59] S. R. Iyer, V. Sarangan, A. Vasan, and A. Sivasubramaniam. Watts in the basket?: Energy analysis of a retail chain. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 4:1–4:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528303>, doi:10.1145/2528282.2528303.
- [60] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity AC metering network. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 253–264, Apr. 2009.
- [61] X. Jiang, J. Polastre, and D. Culler. Perpetual environmentally powered sensor networks. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. URL: <http://dl.acm.org/citation.cfm?id=1147685.1147765>.
- [62] Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. MAQS: A mobile sensing system for indoor air quality. In *Proceedings of the 13th International Conference on Ubiquitous Computing*,

- UbiComp '11, pages 493–494, New York, NY, USA, 2011. ACM. URL: <http://doi.acm.org/10.1145/2030112.2030187>, doi:10.1145/2030112.2030187.
- [63] H. Jin, L. Zhou, C. L. Mak, H. Huang, W. M. Tang, and H. L. W. Chan. Improved performance of asymmetric fiber-based micro-supercapacitors using carbon nanoparticles for flexible energy storage. *Journal of Materials Chemistry A*, 3(30):15633–15641, 2015.
- [64] 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, pages 96–107, New York, NY, USA, 2002. ACM. URL: <http://doi.acm.org/10.1145/605397.605408>, doi:10.1145/605397.605408.
- [65] J. Kadengal, S. Thirunavukkarasu, A. Vasan, V. Sarangan, and A. Sivasubramaniam. The energy-water nexus in campuses. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 15:1–15:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528288>, doi:10.1145/2528282.2528288.
- [66] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst.*, 6(4), Sept. 2007. URL: <http://doi.acm.org/10.1145/1274858.1274870>, doi:10.1145/1274858.1274870.
- [67] A. Kansal, D. Potter, and M. B. Srivastava. Performance aware tasking for environmentally powered sensor networks. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '04/Performance '04, pages 223–234, New York, NY, USA, 2004. ACM. URL: <http://doi.acm.org/10.1145/1005686.1005714>, doi:10.1145/1005686.1005714.
- [68] Kill-A-Watt Wireless. <http://www.p3international.com/products/consumer/p4220.html>.
- [69] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 254–263, April 2007. doi:10.1109/ IPSN.2007.4379685.
- [70] N. Klingensmith, D. Willis, and S. Banerjee. A distributed energy monitoring and analytics platform and its use cases. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 5:1–5:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528283>, doi:10.1145/2528282.2528283.

- [71] Koto Air Indoor Home Monitoring. <https://koto.io/>.
- [72] LEM TT Split-core Current Transformer. http://www.lem.com/hq/en/component?option=com_catalog/task,displayserie/serie,TT/output_type,./.
- [73] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. *TinyOS: An Operating System for Sensor Networks*, pages 115–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
URL: http://dx.doi.org/10.1007/3-540-27139-2_7,
doi:10.1007/3-540-27139-2_7.
- [74] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao. Racnet: A high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’09, pages 15–28, New York, NY, USA, 2009. ACM. URL: <http://doi.acm.org/10.1145/1644038.1644041>,
<doi:10.1145/1644038.1644041>.
- [75] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *IPSN ’07: Proceedings of the 6th international conference on Information processing in sensor networks*, Cambridge, Massachusetts, Apr. 2007.
- [76] Linear Technology. LTC3588 datasheet.
<http://cds.linear.com/docs/en/datasheet/35881fa.pdf>.
- [77] M. Lorek, F. Chraim, K. Pister, and S. Lanzisera. Cots-based stick-on electricity meters for building submetering. *Sensors Journal, IEEE*, 14(10):3482–3489, Oct 2014. <doi:10.1109/JSEN.2014.2346765>.
- [78] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’10, pages 211–224, New York, NY, USA, 2010. ACM. URL: <http://doi.acm.org/10.1145/1869983.1870005>,
<doi:10.1145/1869983.1870005>.
- [79] J. Lu and K. Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *IEEE INFOCOM 2009*, pages 2491–2499, April 2009. <doi:10.1109/INFCOM.2009.5062177>.
- [80] 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, pages 88–97, New York, NY, USA, 2002. ACM. URL: <http://doi.acm.org/10.1145/570738.570751>,
<doi:10.1145/570738.570751>.
- [81] P. Mansourifard, F. Jazizadeh, B. Krishnamachari, and B. Becerik-Gerber. Online learning for personalized room-level thermal control: A multi-armed bandit

- framework. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 20:1–20:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528296>, doi:10.1145/2528282.2528296.
- [82] A. Marchiori and Q. Han. Using circuit-level power measurements in household energy management systems. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 7–12, New York, NY, USA, 2009. ACM. URL: <http://doi.acm.org/10.1145/1810279.1810282>, doi:10.1145/1810279.1810282.
- [83] P. Martin, Z. Charbiwala, and M. Srivastava. DoubleDip: Leveraging thermoelectric harvesting for low power monitoring of sporadic water use. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 225–238, New York, NY, USA, 2012. ACM. URL: <http://doi.acm.org/10.1145/2426656.2426679>, doi:10.1145/2426656.2426679.
- [84] Micro Crystal RV-3049-C3 RTC.
<http://www.microcrystal.com/index.php/products/real-time-clocks>.
- [85] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks. RFC 4944, RFC Editor, September 2007.
<http://www.rfc-editor.org/rfc/rfc4944.txt>. URL:
<http://www.rfc-editor.org/rfc/rfc4944.txt>.
- [86] R. Musaloiu-E., C. J. M. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 421–432, April 2008.
doi:10.1109/IPSN.2008.10.
- [87] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. Imote2: Serious computation at the edge. In *2008 International Wireless Communications and Mobile Computing Conference*, pages 1118–1123, Aug 2008.
doi:10.1109/IWCMC.2008.194.
- [88] G. Nagasubramanian, R. Jungst, and D. Doughty. Impedance, power, energy, and pulse performance characteristics of small commercial li-ion cells. *Journal of Power Sources*, 83(12):193 – 203, 1999. URL:
<http://www.sciencedirect.com/science/article/pii/S0378775399002967>,
doi:[http://dx.doi.org/10.1016/S0378-7753\(99\)00296-7](http://dx.doi.org/10.1016/S0378-7753(99)00296-7).
- [89] National Science and Technology Council–Committee on Technology. Federal research and development agenda for net-zero energy, high-performance green buildings, Oct. 2008.

- [90] National Science and Technology Council–Committee on Technology. Submetering of building energy and water usage: Analysis and recommendations of the subcommitte on buildings technology research and development, Oct. 2011.
- [91] National Science Board. Building a sustainable energy future: U.S. actions for an effective energy economy transformation, Aug. 2009.
- [92] NXP. PCF2127A: Integrated rtc, tcxo and quartz crystal, May 2010.
- [93] Omron D6F-V03A1. [http://components.omron.com/components/web/pdflib.nsf/0/343C62397126E69185257201007DD6C6/\\$file/D6F-V_1010.pdf](http://components.omron.com/components/web/pdflib.nsf/0/343C62397126E69185257201007DD6C6/$file/D6F-V_1010.pdf).
- [94] Pacific Northwest National Laboratory. Small- and Medium-Sized Commercial Building Monitoring and Controls Needs: A Scoping Study, Oct. 2012.
- [95] P. Pannuto, Y. Lee, Y.-S. Kuo, Z. Foo, B. Kempke, G. Kim, R. G. Dreslinski, D. Blaauw, and P. Dutta. Mbus: An ultra-low power interconnect bus for next generation nanopower systems. In *Proceedings of the 42Nd Annual International Symposium on Computer Architecture*, ISCA '15, pages 629–641, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2749469.2750376>, doi:10.1145/2749469.2750376.
- [96] Panoramic Power PAN-10 Sensor. <http://www.panpwr.com/technology>.
- [97] I. Paprotny, Q. Xu, W. W. Chan, R. M. White, and P. K. Wright. Electromechanical energy scavenging from current-carrying conductors. *IEEE Sensors Journal*, 13(1):190–201, Jan 2013. doi:10.1109/JSEN.2012.2211868.
- [98] A. Parisio, D. Varagnolo, D. Risberg, G. Pattarello, M. Molinari, and K. H. Johansson. Randomized model predictive control for HVAC systems. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 19:1–19:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528299>, doi:10.1145/2528282.2528299.
- [99] U. Park and J. Heidemann. Data muling with mobile phones for sensornets. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 162–175, New York, NY, USA, 2011. ACM. URL: <http://doi.acm.org/10.1145/2070942.2070960>, doi:10.1145/2070942.2070960.
- [100] S. N. Patel, S. Gupta, and M. S. Reynolds. The design and evaluation of an end-user-deployable, whole house, contactless power consumption sensor. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2010.
- [101] D. Pech, M. Brunet, H. Durou, P. Huang, V. Mochalin, Y. Gogotsi, P.-L. Taberna, and P. Simon. Ultrahigh-power micrometre-sized supercapacitors based on onion-like carbon. *Nature nanotechnology*, 5(9):651–654, 2010.

- [102] Piezo Sensor - LDT Series.
http://meas-spec.com/product/t_product.aspx?id=2484&terms=LDT0-028K*.
- [103] PLM-1LP. http://www.epd.com/power_meters.html#PLM-1.
- [104] J. Ploennigs, B. Chen, A. Schumann, and N. Brady. Exploiting generalized additive models for diagnosing abnormal energy use in buildings. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 17:1–17:8, New York, NY, USA, 2013. ACM. URL:
<http://doi.acm.org/10.1145/2528282.2528291>,
`doi:10.1145/2528282.2528291`.
- [105] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 95–107, New York, NY, USA, 2004. ACM. URL: <http://doi.acm.org/10.1145/1031495.1031508>,
`doi:10.1145/1031495.1031508`.
- [106] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. URL: <http://dl.acm.org/citation.cfm?id=1147685.1147744>.
- [107] PowerCost Monitor. <http://www.bluelineinnovations.com/powercost-monitor-2>.
- [108] Pressac CT Clamp. <http://www.pressac.com/current-transducer-enocean-ct-clamp>.
- [109] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.
- [110] Y. Qiu, C. V. Liempd, B. O. het Veld, P. G. Blanken, and C. V. Hoof. 5w-to-10mw input power range inductive boost converter for indoor photovoltaic energy harvesting with integrated maximum power point tracking algorithm. In *2011 IEEE International Solid-State Circuits Conference*, pages 118–120, Feb 2011.
`doi:10.1109/ISSCC.2011.5746245`.
- [111] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. URL:
<http://dl.acm.org/citation.cfm?id=1147685.1147764>.
- [112] N. Rajagopal, S. Giri, M. Berges, and A. Rowe. A magnetic field-based appliance metering system. In *Cyber-Physical Systems (ICCPs), 2013 ACM/IEEE International Conference on*, pages 229–238, April 2013.
- [113] Raspberry Pi. <http://www.raspberrypi.org/>.

- [114] A. Rogers, S. Ghosh, R. Wilcock, and N. R. Jennings. A scalable low-cost solution to provide personalised home heating advice to households. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 1:1–1:8, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2528282.2528284>, doi:10.1145/2528282.2528284.
- [115] J. Sallai, A. Lédeczi, and P. Völgyesi. Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 96–107, New York, NY, USA, 2011. ACM. URL: <http://doi.acm.org/10.1145/2070942.2070953>, doi:10.1145/2070942.2070953.
- [116] T. Schmid, D. Culler, and P. Dutta. Meter any wire, anywhere by virtualizing the voltage channel. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 25–30. ACM, 2010.
- [117] Siemens Embedded Micro Metering Module. <http://w3.usa.siemens.com/powerdistribution/us/en/product-portfolio/power-monitoring/energy-management-products/Pages/embedded-submetering-module.aspx>.
- [118] R. Send, Q. Xu, I. Paprotny, R. White, and P. Wright. Granular radio energy-sensing node (green): A 0.56 cm³ wireless stick-on node for non-intrusive energy monitoring. In *SENSORS, 2013 IEEE*, pages 1–4, Nov 2013. doi:10.1109/ICSENS.2013.6688133.
- [119] SensorPush: Humidity & Temperature Smart Sensor. <http://www.sensorpush.com/products/ht1>.
- [120] Smarthome iMeter Solo. <http://www.smarthome.com/2423A1/iMeter-Solo-INSTEON-Power-Meter-Plug-In/p.aspx>.
- [121] Stack Responsive Light Bulbs. <https://www.stacklighting.com/>.
- [122] STMicroelectronics. EnFilm: Thin-film batteries. <http://www.st.com/en/power-management/enfilm-thin-film-batteries.html?querycriteria=productId=SC1107>.
- [123] X. Sun, X. Zhang, H. Zhang, N. Xu, K. Wang, and Y. Ma. High performance lithium-ion hybrid capacitors with pre-lithiated hard carbon anodes and bifunctional cathode electrodes. *Journal of Power Sources*, 270:318 – 325, 2014. URL: <http://www.sciencedirect.com/science/article/pii/S0378775314012002>, doi:<http://dx.doi.org/10.1016/j.jpowsour.2014.07.146>.
- [124] Y. Sun, O. Gurewitz, and D. B. Johnson. Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*,

- SenSys '08, pages 1–14, New York, NY, USA, 2008. ACM. URL:
<http://doi.acm.org/10.1145/1460412.1460414>,
<doi:10.1145/1460412.1460414>.
- [125] T sensors summit for trillion sensor roadmap.
<http://www.tsensorssummit.org/Resources/TSensors%20Roadmap%20v1.pdf>, 2013.
- [126] Y. K. Tan and S. K. Panda. Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes. *IEEE Transactions on Industrial Electronics*, 58(9):4424–4435, Sept 2011.
<doi:10.1109/TIE.2010.2102321>.
- [127] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 407–418, April 2008.
<doi:10.1109/IPSN.2008.67>.
- [128] TAOS. Tsl230brd programmable light-to-frequency converters.
- [129] TED: The Energy Detective. <http://www.theenergydetective.com/>.
- [130] Texas Instruments CC2538. <http://www.ti.com/product/cc2538>.
- [131] Thread Group. Thread. <http://threadgroup.org>.
- [132] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 51–63, New York, NY, USA, 2005. ACM. URL:
<http://doi.acm.org/10.1145/1098918.1098925>,
<doi:10.1145/1098918.1098925>.
- [133] P. Tuset-Peiró, X. Vilajosana, and T. Watteyne. Openmote+: A range-agile multi-radio mote. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 333–334, USA, 2016. Junction Publishing. URL: <http://dl.acm.org/citation.cfm?id=2893711.2893794>.
- [134] U.S. Department of Energy. Wireless Energy Metering Challenge.
https://www1.eere.energy.gov/buildings/publications/pdfs/alliances/wireless_meter_challenge_presentation_050113.pdf.
- [135] U.S. Department of Energy. Quadrennial Technology Review: An Assessment of Energy Technologies and Research Opportunities, Sept. 2015.
- [136] U.S. Department of Energy–Building Technologies Office. Multi-Year Program Plan 2016–2020, Feb. 2016.
- [137] US Dept. of Labor. Illumination regulations #1926.56 (standards - 29 cfr).

- [138] U.S. EPA. The inside story: A guide to indoor air quality, Sept. 1993.
- [139] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM. URL: <http://doi.acm.org/10.1145/958491.958512>, doi:10.1145/958491.958512.
- [140] R. Vullers, R. van Schaijk, I. Doms, C. V. Hoof, and R. Mertens. Micropower energy harvesting. *Solid-State Electronics*, 53(7):684 – 693, 2009. Papers Selected from the 38th European Solid-State Device Research Conference ESSDERC08. URL: <http://www.sciencedirect.com/science/article/pii/S0038110109000720>, doi:<https://doi.org/10.1016/j.sse.2008.12.011>.
- [141] B. A. Warneke and K. S. J. Pister. Mems for distributed wireless sensor networks. In *9th International Conference on Electronics, Circuits and Systems*, volume 1, pages 291–294 vol.1, 2002. doi:10.1109/ICECS.2002.1045391.
- [142] Watts Up? .Net.
<https://www.wattsupmeters.com/secure/products.php?pn=0&wai=0&spec=1>.
- [143] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.
- [144] T. Weng, B. Balaji, S. Dutta, R. Gupta, and Y. Agarwal. Managing plug-loads for demand response within buildings. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 13–18. ACM, 2011.
- [145] G. Werner-Allen, P. Swieskowski, and M. Welsh. Motelab: a wireless sensor network testbed. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks*, 2005., pages 483–488, April 2005.
doi:10.1109/IPSN.2005.1440979.
- [146] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. Rpl: Ipv6 routing protocol for low-power and lossy networks. RFC 6550, RFC Editor, March 2012.
<http://www.rfc-editor.org/rfc/rfc6550.txt>. URL:
<http://www.rfc-editor.org/rfc/rfc6550.txt>.
- [147] A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 221–235, New York, NY, USA, 2001. ACM. URL: <http://doi.acm.org/10.1145/381677.381699>, doi:10.1145/381677.381699.
- [148] Z.-S. Wu, K. Parvez, X. Feng, and K. Müllen. Graphene-based in-plane micro-supercapacitors with high power and energy densities. *Nature communications*, 4, 2013.

- [149] T. Xiang, Z. Chi, F. Li, J. Luo, L. Tang, L. Zhao, and Y. Yang. Powering indoor sensing with airflows: A trinity of energy harvesting, synchronous duty-cycling, and sensing. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 16:1–16:14, New York, NY, USA, 2013. ACM. URL: <http://doi.acm.org/10.1145/2517351.2517365>, doi:10.1145/2517351.2517365.
- [150] Q. R. Xu, I. Paprotny, M. Seidel, R. M. White, and P. K. Wright. Stick-on piezoelectromagnetic ac current monitoring of circuit breaker panels. *IEEE Sensors Journal*, 13(3):1055–1064, March 2013. doi:10.1109/JSEN.2012.2234738.
- [151] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86(8):1640–1659, Aug 1998. doi:10.1109/5.704269.
- [152] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 321–334, New York, NY, USA, 2006. ACM. URL: <http://doi.acm.org/10.1145/1182807.1182839>, doi:10.1145/1182807.1182839.
- [153] L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta. Grafting energy-harvesting leaves onto the sensornet tree. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 197–208, New York, NY, USA, 2012. ACM. URL: <http://doi.acm.org/10.1145/2185677.2185733>, doi:10.1145/2185677.2185733.
- [154] F. Yildiz. Potential ambient energy-harvesting sources and techniques. *Journal of Technology Studies*, 35(1), 2009.