January 2016

Hardware Architectures for Low-power In-Situ Monitoring of Wireless Embedded Systems

Woo Suk Lee
Purdue University

Graduate School Form 30
(Revised 08/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Woo Suk Lee

Entitled: Hardware Architectures for Low-Power In-Situ Monitoring of Wireless Embedded Systems

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

VIJAY RAGHUNATHAN

ANAND RAGHUNATHAN

KAUSHIK ROY

SAURABH BAGCHI

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

VIJAY RAGHUNATHAN

Approved by Major Professor(s): _____

_____

Approved by: V. Balakrishnan 07/26/2016

Head of the Department Graduate Program Date

HARDWARE ARCHITECTURES FOR LOW-POWER IN-SITU MONITORING OF WIRELESS EMBEDDED SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Woo Suk Lee

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2016

Purdue University

West Lafayette, Indiana

Dedicated to all the members of my family, especially my parents and wife, for their unconditional love and support.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and deep appreciation to my advisor Professor Vijay Raghunathan for his patience, guidance, enthusiasm, and support throughout the Ph.D. journey. He has been a motivating factor to me in all aspects of life. Without his guidance and persistent help, this dissertation would not have been possible. I would also like to thank Professor Kaushik Roy, Professor Anand Raghunathan, and Professor Saurabh Bagchi for serving on the advisory committee and providing me with valuable comments and suggestions to improve my research. I am also deeply indebted to my former and current members of Embedded Systems Lab for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for everything we have had in the last six years.

No word can express how grateful I am to my family: my parents, Dae-Kyun Lee and Chun-Sun Jung, and my sister, Areum Lee, for their unconditional love and support. I would like to say a heartfelt thank you to my father for showing me the large world in my early years and inspiring me to explore the world. He has always been the most respectable person in my life. The warmth as a father and the enthusiasm as a scholar always inspire my attitude towards life and career. A very special thanks to my mother for her love and prayers that have sustained me thus far. I also would like to thank my parents-in-law, Weon-Doo Kim and In-Wol Lee.

Last, but certainly not least, I would like to express my sincere gratitude to my wife, Hyangmi Kim, for her endless love, support, and encouragement. I cannot adequately express how I am grateful to her for being my lifetime companion. Were it not for her love and encouragement, I would have failed at the time when the sickness and despair came to me. I am also thankful to my beloved daughter, Claire Jiah Lee, for her lovely smiles brightening my every day.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Lee, Woo Suk PhD, Purdue University, August 2016. Hardware Architectures for Low-power In-Situ Monitoring of Wireless Embedded Systems. Major Professor: Vijay Raghunathan.

As wireless embedded systems transition from lab-scale research prototypes to large-scale commercial deployments, providing reliable and dependable system operation becomes absolutely crucial to ensure successful adoption. However, the untethered nature of wireless embedded systems severely limits the ability to access, debug, and control device operation after deployment —*post-deployment or in-situ visibility*. It is intuitive that the more information we have about a system's operation after deployment, the better/faster we can respond upon the detection of anomalous behavior. Therefore, post-deployment visibility is a foundation upon which other runtime reliability techniques can be built. However, visibility into system operation diminishes significantly once the devices are remotely deployed, and we refer to this problem as a *lack of post-deployment visibility*.

A fundamental factor that limits post-deployment visibility is the resource-constrained nature of these devices, in particular, the severe energy constraints typically present in them. It makes traditional reliability techniques (*e.g.*, modular redundancy) undesirable and even infeasible. In this dissertation, we tackle the key challenge of *lack of post-deployment visibility* in wireless embedded systems. Specifically, we attempt to answer the following question: *"Is it possible to design hardware architectures for wireless embedded systems that enable fine-grained post-deployment visibility, but impose only a minimal (or possibly even zero) power overhead?"* We answer this question in the affirmative and propose three different hardware architectures named Spi-Snooper, Senergy, and TeleProbe that enable us to achieve this goal.

1. INTRODUCTION

The widespread adoption of wireless embedded systems has enabled an era of mobile devices (*e.g.*, smart phones, Internet of Things (IoT)), and their applications even extend to mission-critical and life-assisting systems (*e.g.*, factory automation, medical devices, *etc*). As wireless embedded systems transition from lab-scale research prototypes to large-scale commercial deployments, providing reliable and dependable system operation becomes absolutely crucial to ensure high-quality services. However, the untethered nature of wireless embedded systems severely limits the ability to access, debug, and control the device operation after deployment—*post-deployment or in-situ visibility*. In this context, the *post-deployment visibility* stands for the ability to obtain fine-grained information about a system's operation after deployment from a remote location. It is unquestionable that the more information we have about the system's operation, the more agile action we can take upon the detection of an anomaly. Therefore, visibility is a foundation that serves as a base for other reliability techniques. However, the visibility diminishes significantly once these systems are remotely deployed, and we refer to this problem as *lack of post-deployment visibility*.

The most fundamental factor that limits the visibility is the resource-constrained nature of these devices. The portable design required for these devices confines the size and necessitates a small, on-board power source (*e.g.*, a battery). It renders traditional reliability techniques (*e.g.*, modular redundancy) undesirable and even infeasible. The most prevalent monitoring technique so far includes monitoring software onto the main processor in the system and exploits the primary wireless channel to retrieve monitoring data. Although the method enables the monitoring of system behavior with no additional hardware components, it is a poor design choice in that the monitoring software perturbs system behavior and the reporting through the primary

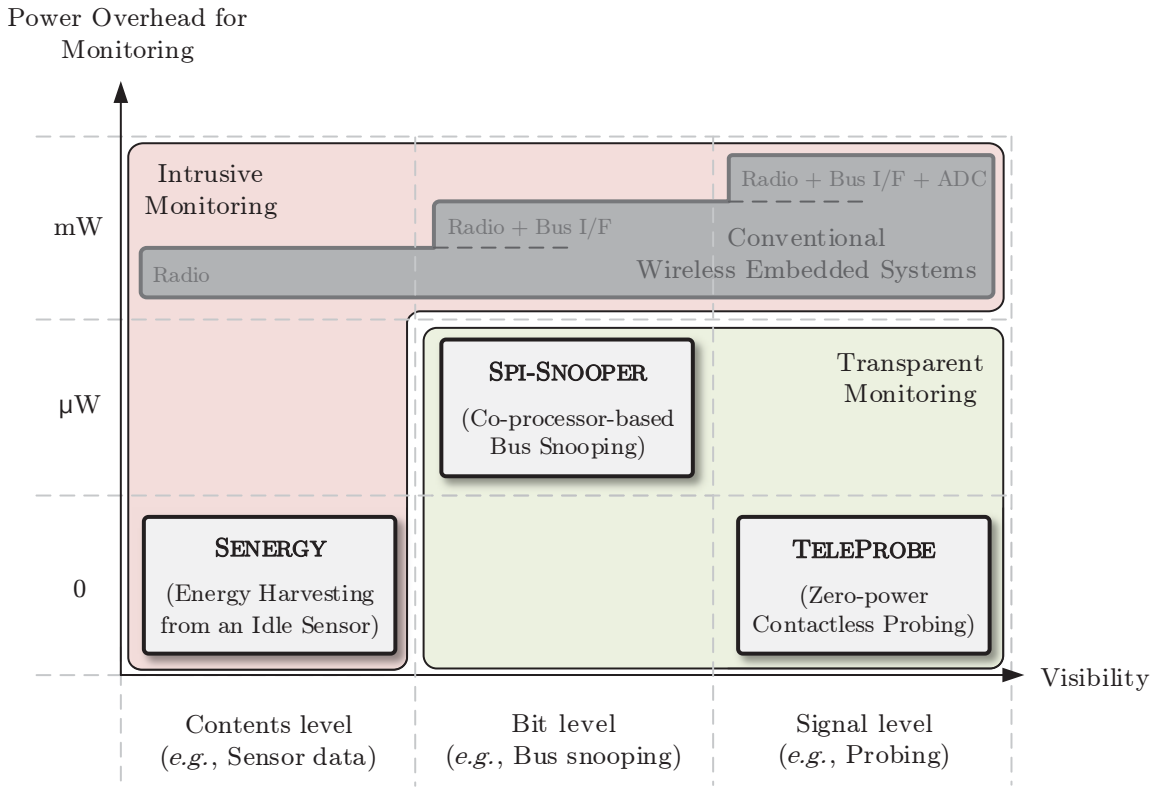


Figure 1.1. Research contributions made in this thesis

wireless channel places additional demands on the radio, which is typically the most power-hungry component in a mobile device.

This thesis tackles the fundamental problem—*lack of post-deployment in-situ visibility*—in wireless embedded systems. Specifically, we attempt to answer the following question: *"Is it possible to design hardware architectures for wireless embedded systems that enable fine-grained post-deployment in-situ visibility, but impose only a minimal (or possibly even zero) power overhead?"* We answer this question in the affirmative and propose hardware architectures that ultimately achieve an electrical signal level visibility with near-zero power consumption. The research contributions made in this thesis, namely SPI-SNOOPER, SENERGY, and TELEPROBE, are illustrated in Figure 1.1 according to their level of visibility and the associated power overhead. The first work, SPI-SNOOPER, presents a wireless sensor node platform that integrates a

reliability co-processor into its hardware architecture. SPI-SNOOPER offloads all the monitoring tasks to the co-processor so that user applications fully utilize the main processor and system resources. Rather than reporting the system status using the wireless channel, the co-processor processes the monitoring tasks within the platform, based on a bus-snooping technique that provides full access to the network communication in a transparent manner. Although the co-processor-augmented SPI-SNOOPER architecture significantly enhances the visibility and reduces the power overhead associated with the monitoring, there are certain types of wireless embedded systems that cannot afford to handle the overhead incurred by the co-processor. The second work, SENERGY, addresses this issue using micro-scale energy harvesting from an idle sensor. With SENERGY, we propose a sub-threshold exponential charge pump architecture that harvests energy from a photodiode sensor during idle time. Utilizing the harvested energy, the SENERGY wireless sensor node platform measures and transmits light intensity during active time, achieving perpetual operation of the wireless sensor node. The ability to collect sufficient energy to operate the entire platform easily compensates the power overhead incurred by the co-processor proposed with SPI-SNOOPER. Finally, the third work, TELEPROBE, proposes a contactless in-situ remote measurement system for implantable medical devices (IMDs), which achieves oscilloscope-like electrical signal probing with near-zero power consumption. By enabling a near-zero power contactless probing mechanism for IMDs, we demonstrate how the architectural support from hardware can help address the issue of visibility even for such severely resource-constrained wireless embedded systems. In the following sections, we briefly describe the three key components of this thesis: SPI-SNOOPER, SENERGY, and TELEPROBE.

1.1 Spi-Snooper: Hardware-Software Approach for Transparent Network Monitoring in Wireless Sensor Networks

The lack of post-deployment visibility into system operation is one of the major challenges in ensuring reliable operation of remotely deployed embedded systems such as wireless sensor nodes. Over the years, many software-based solutions (in the form of debugging tools and protocols) have been proposed for in-situ system monitoring. However, all of them share the trait that the monitoring functionality is implemented as software executing on the same embedded processor that the main application executes on. This is a poor design choice from a reliability perspective. We make the case for a joint hardware-software solution to this problem and advocates the use of a dedicated reliability co-processor that is tasked with monitoring the operation of the embedded system. As an embodiment of this design principle, we present Spi-Snooper, a co-processor-augmented hardware platform specifically designed for network monitoring. Spi-Snooper is completely cross-compatible with the Telos wireless sensor nodes from an operational standpoint and is based on a novel hardware architecture that enables transparent snooping of the communication bus between the main processor and the radio of the wireless embedded system. The accompanying software architecture provides a powerful tool for monitoring, logging, and even controlling all the communication that takes place between the main processor and the radio. We present a rigorous evaluation of our prototype and demonstrate its utility using a variety of usage scenarios.

1.2 Senergy: Micro-scale Energy Harvesting from an Idle Sensor

Wireless sensor devices are heavily duty-cycled to minimize energy consumption. To further reduce the energy required for sensing, these devices often prefer passive sensors that produce output power proportional to physical quantity. Therefore, while a sensor output is not being sampled (*i.e.*, idle time), the output power from the sensor is unused and, hence, wasted. Inspired by the observation, we propose

a sub-threshold exponential charge pump architecture that works with an ultra-low capacity power source (*e.g.*, passive sensor) that has an output voltage and current as low as 250 mV and 6 µA. Utilizing the charge pump as an energy harvester to a system, we eventually design and implement SENERGY, a batteryless wireless sensing platform that uses a photodiode sensor as a sole power source. Specifically, the SENERGY prototype reports light intensity through a 2.4 GHz radio whenever it has harvested sufficient enough energy from the photodiode sensor, thus *achieving perpetual operation*. We use the prototype to evaluate the proposed system based on two application scenarios, namely adaptive transmission of sensor data and providing an uninterrupted power supply to an on-board real-time clock.

1.3 TELEPROBE: Zero-power Contactless Probing for Implantable Medical Devices

The development of implantable medical devices (IMDs) has revolutionized the monitoring, diagnosis, and treatment of a wide range of medical conditions. Given their direct impact on human safety, the need for reliable operation is a fundamental, non-negotiable requirement in IMDs. While wireless connectivity is becoming common in IMDs to non-intrusively monitor a patient's health and device status, the wireless link-based conventional approaches incur significant energy overheads for data acquisition, processing, and active radio transmission. While low-power transceivers have been introduced to reduce the energy consumed by the radio itself, the energy consumed by the microcontroller for processing data and controlling the radio has often been overlooked. As a result, in IMDs that have a stringent energy constraint, runtime monitoring of an IMD for extended durations over a wireless channel is, simply put, an impractical solution. To address this challenge, we present TELEPROBE, an in-situ remote measurement system for IMDs, which enables continuous and direct wireless readout of analog and digital electrical signals using an inductively-coupled LC tank circuit, without imposing any power overhead on the

IMD. We have designed and implemented fully functional prototypes of TeleProbe and demonstrated its utility in the context of three practical usage scenarios: behavior validation with power analysis, monitoring an off-chip serial communication bus, and active data transmission.

The rest of this thesis is organized as follows. Chapter 2 presents the hardware-software approach for network monitoring in wireless sensor networks, Spi-Snooper. Chapter 3 details the sensor-powered energy harvesting solution that enables perpetual operation of a wireless sensing platform, Senergy. Chapter 4 describes the zero-power contactless probing mechanism for implantable medical devices, TeleProbe. Finally, Chapter 5 concludes the thesis.

2. SPI-SNOOPER: HARDWARE-SOFTWARE APPROACH FOR TRANSPARENT NETWORK MONITORING IN WIRELESS SENSOR NETWORKS

2.1 Introduction

As networked embedded systems (such as wireless sensor nodes) transition from lab-scale research prototypes to large-scale commercial deployments, providing reliable and dependable system operation becomes absolutely crucial to ensuring widespread adoption and commercial success. Due to the fact that sensor nodes operate in dynamic and unpredictable physical environments that cannot be recreated in a lab setting, it is now generally accepted that pre-deployment testing alone (using simulators, emulators, *etc.*) is not sufficient to guarantee reliability and that *in-situ monitoring of nodes after deployment is a must*. In accordance with this belief, a number of techniques have been proposed for post-deployment node-monitoring and control ([1–4] are a few examples). However, all of them share the common trait that the node-monitoring functionality is implemented as software executing on the same embedded processor that the application executes on.

From a reliability perspective, this is a poor design choice due to several reasons. First, the monitoring software shares (and competes for) resources, such as CPU cycles and memory space, with the main application software, further depriving the application of these already-scarce resources. Second, the presence of this additional software can perturb the timing behavior of the application, possibly suppressing some subtle bugs, or causing a large slowdown in application execution. Third, such an architecture inherently has a single point of failure; *e.g.*, if the processor hangs or freezes (possibly due to a bug in the main application code), the monitoring software is rendered essentially useless.

We believe that a joint hardware-software approach is both required and ideal to address the problem of post-deployment node monitoring and control. In particular, we advocate a rethinking of the hardware architecture of wireless embedded systems, such as sensor nodes, to include an additional (low-cost and low-power) component that we call the *reliability co-processor*, which is responsible for monitoring the operation of the sensor node. As we will show, logically and physically separating the monitoring functionality from both the application software and the main processor in this manner allows the monitoring to be conducted in a decoupled, non-intrusive, and transparent manner, enhancing reliability. We present an embodiment of the above design principle in the form of SPI-SNOOPER[1], a co-processor-augmented hardware architecture specifically designed for network monitoring. We select network monitoring because, as we will demonstrate, carefully monitoring the bi-directional communication activity in a wireless embedded node can reveal a significant amount of information about its operation. However, our design (and more generally, any reliability co-processor-augmented design) is certainly not limited to network monitoring alone and can be used for a variety of other scenarios as well. Specifically, we make the following contributions:

- We present SPI-SNOOPER, the first wireless sensor node platform that integrates a reliability co-processor into its hardware architecture. SPI-SNOOPER provides many novel hardware features: (a) the reliability co-processor can passively monitor (*i.e.*, snoop) all the transactions on the Serial-Peripheral Interface (SPI) bus that connects the main processor and the radio on our sensor node. This enables the co-processor to have complete visibility into all the information transmitted and received by the node. The snooping is fully transparent in the sense that the main processor and radio are not aware that the bus is being monitored, (b) in addition to passively monitoring processor-radio communication, the co-processor also has the ability to disconnect the main processor

[1] This work had been done in collaboration with Mohammad Sajjad Hossain (sajjad@purdue.edu) from the Embedded Systems Lab at Purdue University while he was enrolled in the university.

from the SPI bus and take control of the bus. In addition to cutting off a processor that exhibits faulty behavior, this also allows the co-processor to independently transmit and receive packets to/from other nodes, (c) the reliability co-processor can also access other hardware components on the sensor node, (*e.g.*, read various sensors). This allows the co-processor to independently validate the behavior of the main application software, and (d) the co-processor can reset/reboot the main processor if desired (*e.g.*, if the co-processor detects that the main processor has hung or is operating using corrupted state information).

- Spi-Snooper features a lightweight, yet powerful software architecture that allows it to exploit the unique hardware features in an accurate, reliable, and transparent manner.

- We design, implement, and evaluate several usage scenarios for the Spi-Snooper platform and demonstrate how it can be used to significantly enhance the level of post-deployment visibility and control for remotely deployed wireless sensor nodes.

The remainder is organized as follows. Section 2.2 and 2.3 present the novel hardware/software architectures and implementation of Spi-Snooper respectively. Section 2.4 describes a rigorous evaluation that we performed on Spi-Snooper and the various usage scenarios that we implemented to demonstrate the utility and capability of the Spi-Snooper architecture. Section 2.5 describes related work and Section 2.6 concludes the work with some discussion and avenues for future work.

2.2 Spi-Snooper Hardware Design

Spi-Snooper features a reliability co-processor-augmented hardware architecture that enables monitoring and control of network communication in a manner that is transparent to the main-processor. This section describes the hardware architecture and implementation of Spi-Snooper in detail.

For compatibility reasons, we decided to base our hardware architecture around an existing wireless sensor node platform. Although there are numerous such platforms available [5], we picked the TELOS mote as our base platform of choice due to its widespread adoption and use in the wireless sensor network research community. In the TELOS design, the main processor and radio IC are placed separately and communicate over an SPI communication bus, which is exposed as a trace on PCB. Hence, it is possible to physically tap into the bus without modifying the original hardware architecture of the TELOS mote. However, physical access to the SPI interface of an off-the-shelf TELOS mote seems impossible in practice due to the delicate and thin traces on the compact form factor PCB. Therefore, we built our own PCB implementation of SPI-SNOOPER, which was based on the existing TELOS schematics [6], suitably enhanced with the co-processor and the additional logic required for monitoring and controlling the SPI bus that connects the main processor and the radio.

2.2.1 The TELOS mote

The TELOS [7] mote is a popular, open-source platform that features a Texas Instruments (TI) MSP430F1611 microcontroller as the main processor and an IEEE 802.15.4-compliant TI CC2420 radio transceiver. It also has on-board temperature, humidity, and light sensors and 1 MB of external flash for data logging.

The hardware interface between the main processor and the radio on the TELOS mote is depicted in Figure 2.1. As shown in the figure, the main processor and the radio utilize a total of ten data and control lines for interfacing. They exchange data over an SPI bus using four signal lines (`SIMO`, `SOMI`, `SCLK`, and `CSn`). The SPI is a full-duplex synchronous serial data link that supports communication between devices in a master/slave configuration. In the case of the TELOS mote, the processor acts as the master and initiates all bus transactions, while the CC2420 radio acts as the slave. Although the CC2420 radio cannot initiate bus transfers, it utilizes

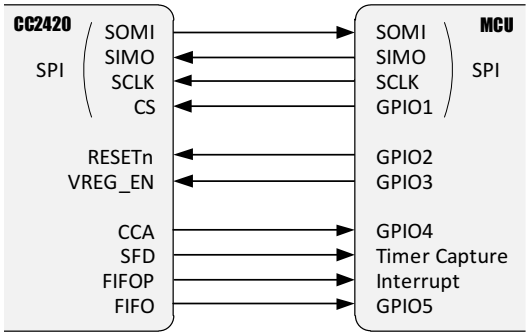
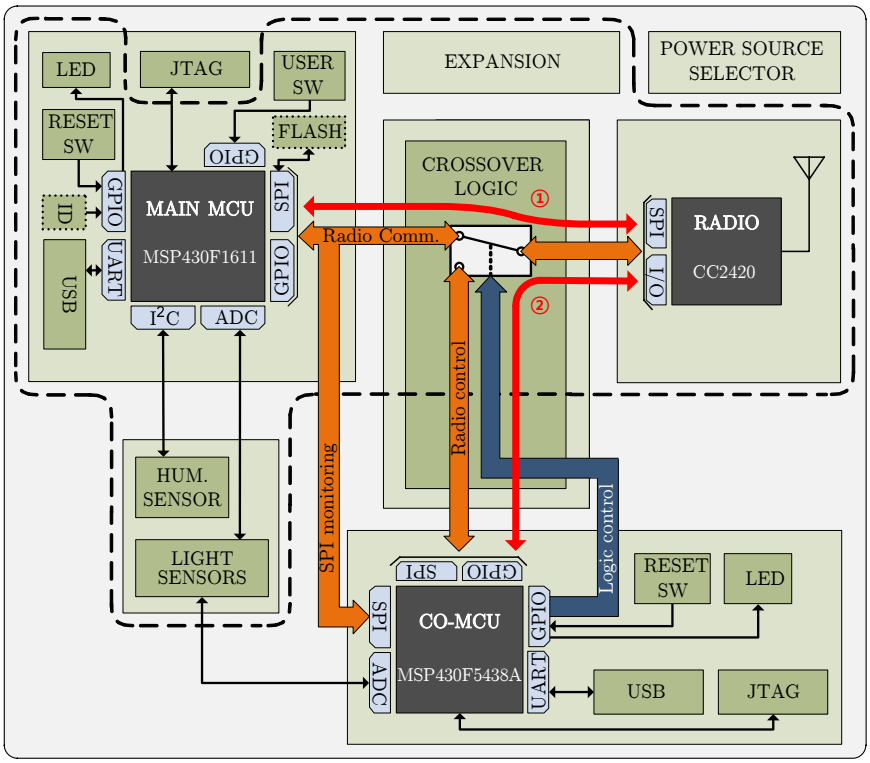


Figure 2.1. The interface between the radio and the main processor on the TELOS mote

four other control lines (`SFD, CCA, FIFOP,` and `FIFO`) for reporting events occurred in the radio. For example, `FIFO, FIFOP`, and `SFD` pins are used to notify the main processor of a packet reception. Additionally, the main processor can reset the radio and control the voltage regulator inside the radio via the `RESETn` and `VREG_EN` lines.

2.2.2 SPI-SNOOPER hardware architecture

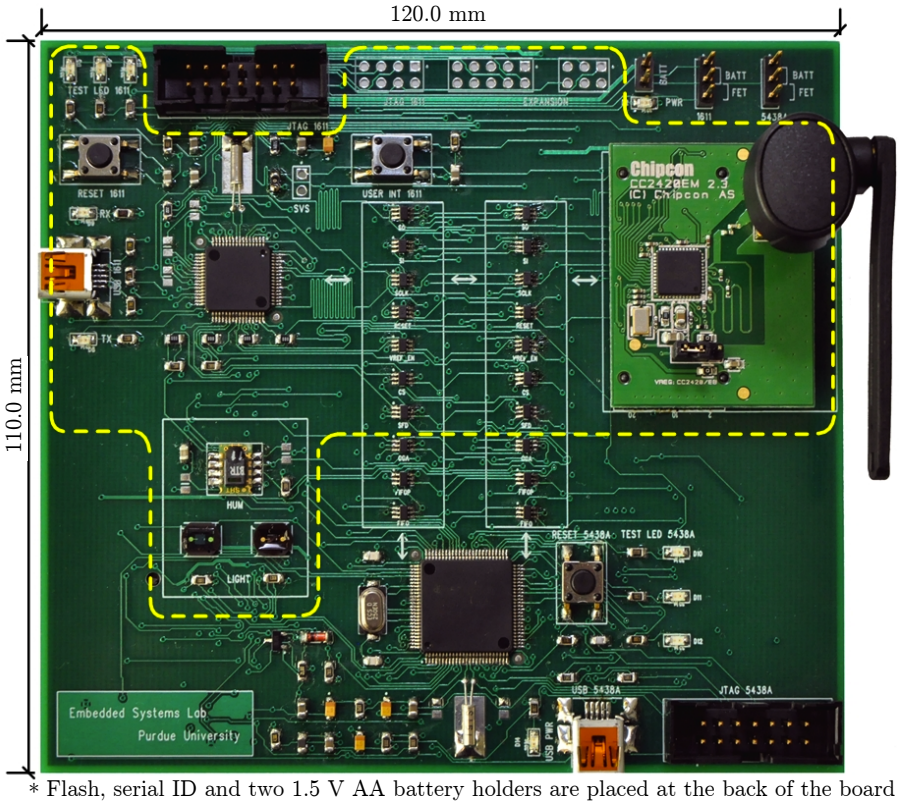
Figure 2.2 and Figure 2.3 show the block diagram and the photograph of the SPI-SNOOPER platform, respectively. As shown in the figures, the SPI-SNOOPER hardware platform consists of the four major components: main processor, reliability co-processor, radio, and crossover logic. Inside the dashed line is the circuitry identical to the TELOS mote (except for the crossover logic). Therefore, the main processor (TI MSP430F1611) and the radio (TI CC2420) are the same as that can be found on the TELOS mote. The main processor is in charge of controlling the system peripherals such as data flash, serial ID, LEDs, switches, and sensors. For the radio, we utilize an off-the-shelf evaluation module (TI CC2420EMK) in order to reduce the complexity involved in the radio circuit design. The crossover logic (further described in Section 2.2.5) placed in the middle of the main processor and the radio splits the SPI bus, which is connecting them, into two segments. For normal operation, the crossover



* Flash, serial ID and two 1.5 V AA battery holders are placed at the back of the board

Figure 2.2. Block diagram of the SPI-SNOOPER platform

logic is configured to a pass-through mode, and the main processor and the radio gets connected, achieving the same functionality as the TELES mote. During the normal operation, the co-processor can monitor (*i.e.*, snoop) every transaction taking place on the SPI bus in real-time using the dedicated snooping lines branched from the original bus. If an anomaly is detected by the co-processor through snooping, the co-processor controls the crossover logic to divert the bus connection so that the co-processor takes over the control of the radio. Besides the ability to control the radio, the co-processor is also able to reset the main processor and independently access the light sensors. The SPI-SNOOPER board is a 3 V system and selectively powered by a USB port, debugger, or a set of two AA batteries.



* Flash, serial ID and two 1.5 V AA battery holders are placed at the back of the board

Figure 2.3. Photograph of the Spi-Snooper platform

2.2.3 Reliability co-processor

The main considerations in selecting which microcontroller (MCU) to use as the reliability co-processor are that it should be low-cost, low-power, provide sufficient computational resources, be easy to integrate into a larger design with minimal complexity, be easy to program, and have all the peripherals required to support the snooping of the SPI bus. After a survey of the several off-the-shelf MCUs, we selected the TI MSP430F5438A as the reliability co-processor for the Spi-Snooper platform as it meets all the requirements listed above.

The MSP430F5438A is a 16-bit MCU that can run at a maximum frequency of 25 MHz. It has three 16-bit timers, a 12-bit analog-to-digital (A/D) converter, up to four universal serial communication interfaces (USCI), and up to 87 general

purpose IO pins [8]. An added advantage of selecting this microcontroller is that it uses the same MSP430 instruction set architecture as the main processor on the TELOS, which eases programming effort (*e.g.*, the same software toolchain can be used for both processors). It is also more power efficient than the main processor (*e.g.*, the MSP430F5438A consumes 350 µA at 1 MHz with 3 V supply voltage in the active mode (AM), whereas the MSP430F1611 consumes 500 µA under the same condition [8, 9]). The power consumption of SPI-SNOOPER will further be analyzed in section 2.4.1.

It should be noted that a microcontroller used as a reliability co-processor is not necessarily the MSP430F5438A. Although the MSP430F5438A was chosen due to its adequate performance, any MCUs that are able to run at over 20 MHz would be a proper choice because the maximum SPI clock frequency of the CC2420 radio is 10 MHz.

2.2.4 Monitoring the processor-radio SPI bus

In order to monitor (*i.e.*, snoop) the SPI bus between the main processor and the radio, the signal lines of the SPI bus are forked off to two independent SPI interfaces of the reliability co-processor. The two SPI interfaces are configured as slaves and share the clock signal, `SCLK`, with the bus being monitored for synchronization purposes. One of the two SPI interfaces (`CoProc_SPI_1`) is dedicated to monitoring the `SIMO` (slave input master output) line of the SPI bus. As the name implies, the `SIMO` line contains a serial bitstream of the data that the main processor (*i.e.*, the bus master) sends to the radio (*i.e.*, the bus slave). At each clock transition of the SPI bus between the main processor and the radio, `CoProc_SPI_1` also receives the bitstream that was being sent from the main processor to the radio. The other SPI interface on the co-processor (`CoProc_SPI_2`) is dedicated to monitoring the `SOMI` (slave output master input) line of the SPI bus. As the name implies, the `SOMI` line contains a serial bitstream that the radio sends to the main processor. This line is connected

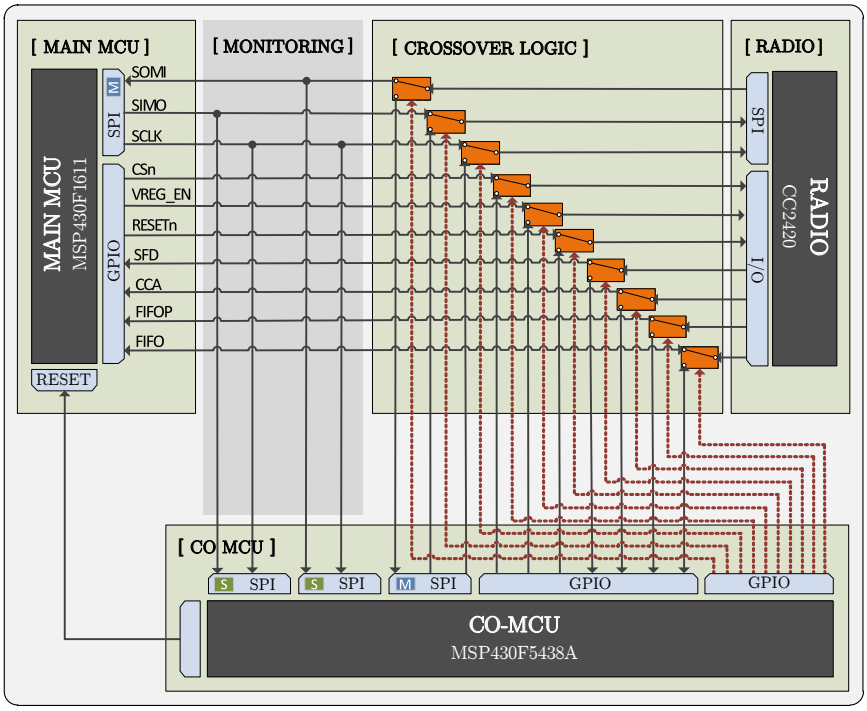


Figure 2.4. Configuration of the monitoring and crossover logic

to the `SIMO` pin of `CoProc_SPI_2`. At each clock transition of the SPI bus being monitored, `CoProc_SPI_2` also receives the bitstream that was being received by the main processor from the radio. The area marked with “Monitoring" in Figure 2.4 shows how the processor-radio SPI bus is branched out to the co-processor. The two SPI interfaces of the co-processor operate in a fully interrupt-driven manner for energy-efficient monitoring.

2.2.5 Crossover logic

In addition to the bus monitoring described above, the reliability co-processor on the SPI-SNOOPER platform also has the capability of disconnecting the main processor from the radio and assuming the full control of the radio. For this functionality, as the main processor does, the co-processor also has an access to the SPI bus as well

as control lines through crossover logic. The crossover logic consists of 10 single-pole double-throw (SPDT) switches (TI SN74LVC1G3157). The SPDT switches provide sufficient switching speed (typically 0.5 ns) and operates within a wide supply voltage range (1.65 V to 5.5 V), while consuming 0.05 µA (typical) for maintaining its status regardless of specific supply voltage level [10]. These SPDT switches are controlled by the reliability co-processor, as illustrated in Figure 2.4. Depending on the control input to the SPDT switches, the switches bridge the SPI interface and the control lines of the radio either to the main processor (*Route 1* in Figure 2.2) or to the co-processor (*Route 2* in Figure 2.2).

The co-processor manages its own rule to make a decision on when to actively engage the device operation. The decision making algorithm resides in the co-processor as software, and it is entirely up to administrators of the node. Rather, the hardware architecture provides a fail-safe mechanism so that any possible faulty behavior of the co-processor cannot affect the main application. In particular, by default, the crossover logic is configured to connect the SPI interfaces of the main processor and the radio if the control signals to the SPDT switches are absent.

2.2.6 Cost analysis

Table 1 shows the cost of the additional components on the Spi-Snooper platform in addition to the ones included in the Telos platform, assuming the production of 1,000 boards.

2.3 Spi-Snooper Software Architecture

In order to maintain the transparent nature of Spi-Snooper, we make absolutely no modification to the software that runs on the main processor. The only information we need is the type of the operating system it runs. The reason is that different operating systems configure the SPI bus and the CC2420 radio differently (more on this in Section 2.3.3). For the co-processor, rather than porting an OS, we developed

Table 2.1. Additional cost required for a SPI-SNOOPER hardware platform (not including components found on the TELOS mote) in quantities of 1000.

Component	**Count**	**Cost (USD)**	
		Unit	**Total**
MCU	1	4.55	4.55
USB	1	2.95	2.95
SPDT Switch	10	0.13	1.30
PCB	1	8.50	8.50
Misc	-	-	5.00
Total	-	-	22.30

the software from scratch in C language to keep it lightweight and efficient. Figure 2.5 shows the key components of the SPI-SNOOPER software architecture.

SPI-SNOOPER can operate in two different modes:

- **Passive mode:** In this mode, the co-processor listens to all SPI communication as a slave (Section 2.2.4). The main purpose of this mode to log the communication taking place on the SPI bus. In this mode, the co-processor can also access the sensors for integrity checking of certain types of data that the main processor sends to other nodes. SPI-SNOOPER can switch to the *active mode* based on certain types of events (*e.g.*, if the rate of transmission exceeds a threshold) or commands received from other nodes (*e.g.*, the base station).

- **Active mode:** In this mode, the co-processor works as a master of the SPI bus. It manipulates the crossover logic (Section 2.2.5) to assume the control of the bus and the radio. Then, it can use the radio to communicate with other devices in the network or to route packets so that the network connectivity is maintained.

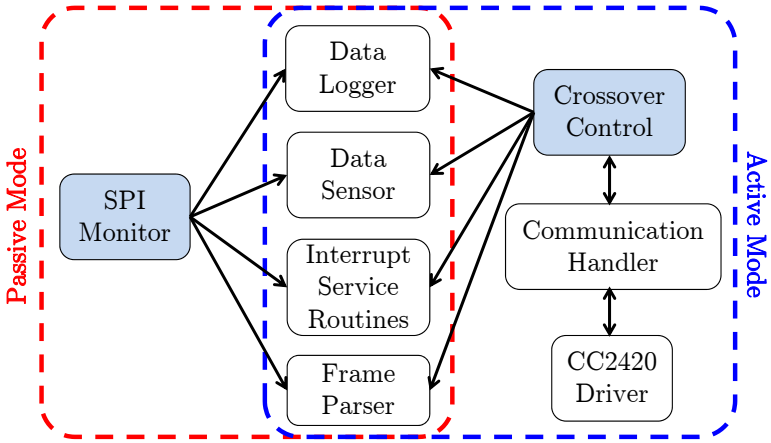


Figure 2.5. SPI-SNOOPER software architecture

Now, we describe some of the key components of the SPI-SNOOPER software as shown in Figure 2.5.

- **SPI monitor:** In passive mode, the co-processor is configured as an SPI slave and snoops the SPI bus that connects the main processor and the radio. The major task of this component is to identify and record the incoming/outgoing packets. It can make use of other components in the software stack as well. For example, it often uses the data logger to log the captured packets to the flash or to the serial port.

- **Crossover control:** SPI-SNOOPER is able to assume the full control of the SPI bus to directly interface the radio. This control may be necessary in certain scenarios if the co-processor suspects malicious behavior caused by the main processor. This module may use other components, such as communication handler, to communicate with other nodes in the network.

- **Communication handler:** In active mode, the communication handler is responsible for moving data to and from the radio IC. The handler interfaces with the low level radio driver to send and receive data through the radio. It also maintains a routing table for multi-hop communication.

- **Data logger:** In the case that reporting the collected data through the radio is expensive or even impossible (*e.g.*, in the passive mode), the data logger either stores the data locally or forwards it through the serial port. To be specific, the data logger uses the on-board flash storage in MSP430F5438A, mostly in meta-data format for a later retrieval (Section 2.4.2). The data is stored in batches to reduce the logging overhead (*e.g.*, write time to the flash). It can also forward the logs through the serial port during run-time if a device for analysis (*e.g.*, a PC) is attached to it.

- **Data sensor:** This component enables SPI-SNOOPER to access the on-board sensors. The sensor data measured by the co-processor can be compared with the one contained in a packet that the main processor is transmitting.

- **Applications:** The application is a software wrapper that organizes the aforementioned software components. The application implements various types of logging and/or anomaly detection algorithm, depending on usage scenarios.

In addition to the major components listed, the SPI-SNOOPER software architecture for the co-processor also includes the frame parser, the interrupt service routines, *etc.* As compared to the co-processor, the main processor is running the Contiki [11] operating system as it does in the TELOS mote.

2.3.1 Contiki operating system

Contiki [11] is a lightweight operating system (OS) developed for resource-limited networked embedded systems such as wireless sensor networks (WSNs). Contiki contains two communication stacks: uIP [12] and Rime [13]. uIP is a TCP/IP stack for IP-based communication and Rime is a lightweight communication stack designed for low-power radios.

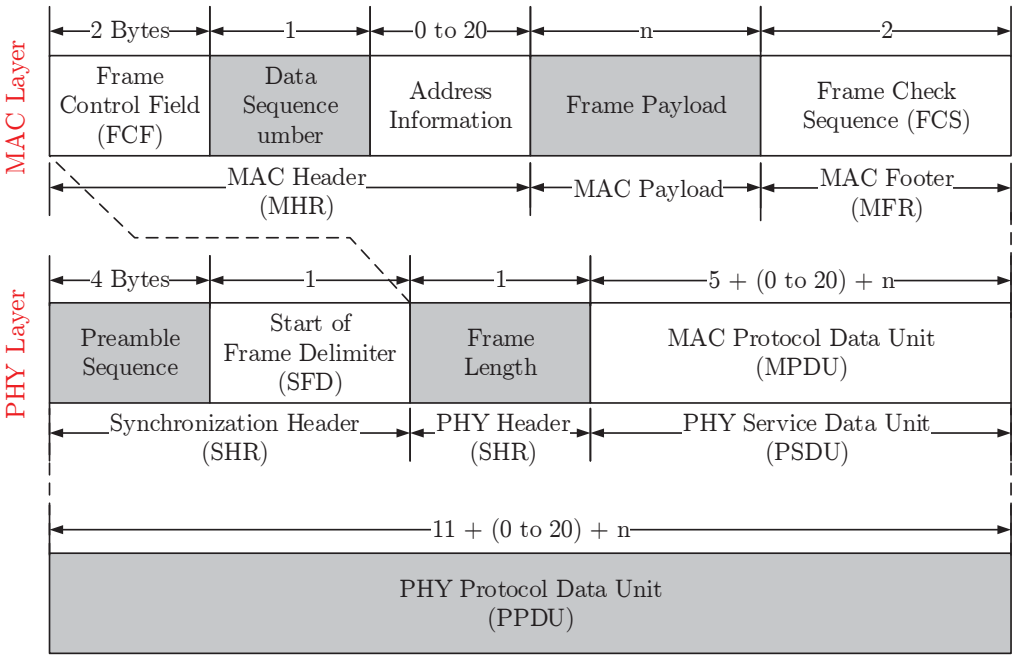


Figure 2.6. Schematic view of the IEEE 802.15.4 frame format

2.3.2 Monitoring network communication

The co-processor in Spi-Snooper can listen to all communication between the main processor and the radio. In this section, we will describe how the co-processor identifies incoming or outgoing packets.

Detecting incoming packets

Among the pins used for interfacing the radio (see section 2.2.1), the `FIFOP` pin is used to signal the MCU when a complete frame has been received. As soon as this interrupt signal is received, the co-processor starts recording the bytes that are passed from the radio to the main processor (*i.e.*, from `SOMI` of the radio to `SOMI` of the main processor). This recording continues until the receiving buffer at the radio is empty, which can be identified by monitoring the `FIFO` pin. Even though the `FIFOP` pin generates an interrupt when the frame is received at the radio, the main processor

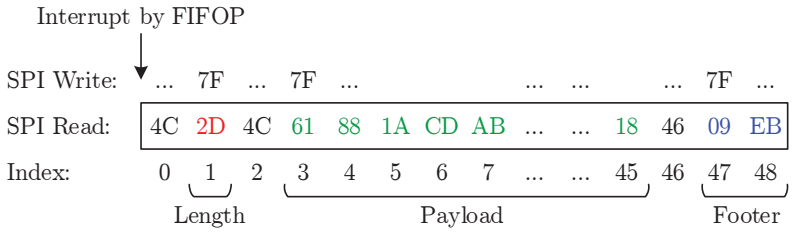


Figure 2.7. Example byte stream on the SPI bus to receive a radio packet

may delay the receiving process by few cycles. As a result, just by observing the status of those two pins, we cannot deterministically identify the presence of a packet on the SPI bus. When the main processor is ready to read the received packets queued in the receiving buffer of the radio, it writes `CC2420_RXFIFO | 0x40` (*i.e.*, `0x7F`) to the SPI bus. Henceforth, we refer to this byte as `START_READING`. Therefore, once the co-processor notices that a packet has arrived at the radio, it starts tracking the `START_READING` byte. In that way, the location of the `START_READING` byte helps identify the exact starting point of an incoming packet. Figure 2.7 shows an example sequence of bytes, showing how the main processor running Contiki is interfacing the radio to receive packets. To be specific, a packet is read in three parts: the length (red), the payload (green) and the footer (blue). Reading of each part is initiated by sending the `START_READING` byte to the radio.

Detecting outgoing packets

In the previous section, we have shown how the two pins in the radio, namely `FIFOP` and `FIFO`, help identify incoming packets. However, for outgoing packets, the radio does not provide such a dedicated mechanism to notify that an SPI transaction for outgoing transmission is ongoing. Before explaining the detection mechanism for outgoing packet, it is important to understand the high level overview of how Contiki sends a packet using the radio. The sending process takes place in two steps: *Prepare* and *Transmit*. During *Prepare*, the MAC header and the MAC payload (Figure 2.6)

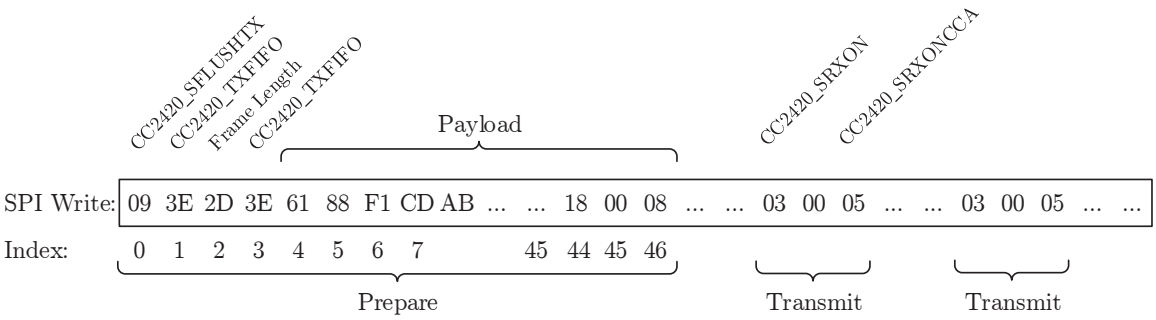


Figure 2.8. Example byte stream on the SPI bus to transmit a radio packet

are loaded to the transmitting buffer of the radio. The MAC footer is automatically added by the radio hardware. In Contiki, at the beginning of *Prepare*, the main processor sends a strobe signal (`CC2420_SFLUSHTX` or `0x09`) to the radio. After that, the main contents of the packet is sent: (a) the frame length (one byte) and (b) the MAC header and the MAC payload (Figure 2.6). If CRC check is enabled, two bytes of checksum value for the payload is also sent lastly. Before each step, the address (`CC2420_TXFIFO` or `0x3E`) of the transmitting buffer is written to the SPI bus. The analysis reveals the following pattern: `0x09 0x3E frame_length 0x3E`. If the pattern is found in the byte stream from the main processor to the radio, we can conclude that a sending process for a new packet is initiated. However, *Prepare* is a process that just loads contents onto the transmitting buffer of the radio, and the actual transmission takes place during the next step: *Transmit*.

During *Transmit*, the main processor instructs the radio to start the actual transmission, which may or may not be successful based on the status of the radio. The sequence of bytes that are written to the SPI bus during this phase depends on the configuration of the radio. If the radio is configured to send a packet with the clear channel assessment (CCA), the main processor sends two strobe signals: `CC2420_SRXON` (`0x03`) and `CC2420_STXONCCA` (`0x05`). If CCA is not enabled, it only sends `CC2420_STXON` (`0x04`). An arbitrary number of `CC2420_SNOP`s (`0x00`) may also be sent in between the aforementioned signals. To resend an already loaded packet

(*e.g.*, if retransmission is enabled), the processor just needs to issue the above strobe signals again. Thus, multiple *Transmit* steps may follow a single *Prepare* step. After detecting a packet during *Prepare*, SPI-SNOOPER also records the number of times that the packet was attempted to send by recording the number of occurrences of *Transmit*. Figure 2.8 shows an example bytestream that has sent through the SPI bus while a packet is being transmitted twice. Note that the payload length is 43 even though the frame length is set to 45 (2D in hex). This is because the radio automatically adds two bytes of frame control sequence before sending a packet.

2.3.3 SPI-SNOOPER in other operating systems

Fundamentally, the SPI-SNOOPER hardware architecture is capable of working with any sensor network OS, not only with Contiki (*e.g.*, TinyOS [14]). But the same software may not be compatible across different types of OS. The reason is that different OS handles the radio communication in different way. One example is how incoming packets are handled in TinyOS, where it follows a different radio configuration than Contiki. In Section 2.3.2, we explained how we detect incoming packets using the interrupt generated by the `FIFOP` pin of the radio. By default, `FIFOP` is an active high signal. But in TinyOS, it is configured to be an active low signal instead. Thus, the same detection mechanism will not work for TinyOS. We can overcome such issues by modifying the detection mechanisms in the SPI-SNOOPER software, without modifying the underlying hardware. This is true for other operating systems as well.

2.4 Evaluation

We evaluate SPI-SNOOPER using a set of microbenchmarking and application scenarios. We first discuss the core functionalities of SPI-SNOOPER in the microbenchmark and demonstrate how SPI-SNOOPER is used to enhance the reliability of networked embedded systems. The use cases demonstrated in this section are namely

Table 2.2. Memory footprint of Spi-Snooper

Program	**Memory Usage (bytes)**	
	RAM	**ROM**
Contiki (Hello World)	5,396	23,254
monitoring + no logging + no crossover	617	2,448
monitoring + flash logging + no crossover	905	3,124
monitoring + serial logging + no crossover	903	3,180
monitoring + serial logging + crossover	1,210	5,084

(a) network communication logging, (b) integrity verification, (c) Abnormal behavior handling, and (d) providing an emergency backdoor. Other than Spi-Snooper, all the motes are Telos, and the Spi-Snooper is functionally identical to the Telos mote, except for the presence of the co-processor. The main processors on those devices are running Contiki 2.5, and the Rime communication stack [13] is used for network communication.

2.4.1 Microbenchmarking

Memory footprint

Table 2.2 shows the memory footprint of the Spi-Snooper software for different configurations. Specifically, the table states the RAM and the ROM usage for various logging options and for the selective use of the crossover logic. As shown in the table, the Spi-Snooper software incurs around 1 KB RAM and 5 KB ROM usage with the crossover logic and the serial logging enabled. For the sake of comparison, we also include the memory usage of a basic Contiki program compiled using the default compilation options.

Accuracy of snooping network communications

The experiment results for accuracy of snooping incoming/outgoing packets are discussed in this section. The incoming and outgoing packets sent by the main processor were captured by the co-processor and logged onto the internal flash. The logs are retrieved later during the post-processing phase and compared with the original packets that the main processor transmitted.

First, for the test of outgoing packet snooping, the main processor on the Spi-Snooper node is configured to periodically broadcast 802.15.4-compliant packets with a variable payload length and a variable transmission rate. To be specific, we varied the number of packets transmitted per a second from 1 to 128, and also varied the length of payload. Most radio settings (*e.g.*, transmission power, transmission channel, *etc.*) were left default for the radio configuration in Contiki, and only the radio duty cycling layer is set to use `nullrdc` to avoid packet retransmission. In all the test cases, Spi-Snooper achieved 100% detection without any miss. Second, for the test of incoming packet snooping, all the settings were left the same, but the main processor on the Spi-Snooper node is configured to a receive-only mode. Unlike the previous case, this time, a Telos mote broadcasts packets. For the same variation of the packet transmission rate and the payload length, again, the co-processor on the Spi-Snooper node successfully captured all the incoming packets without any loss.

Accuracy of sensing

Telos contains two analog light sensors (*i.e.*, total solar radiation (TSR) sensor and photosynthetically active radiation (PAR) sensor) that have different spectral response range and peak sensitivity. The Spi-Snooper hardware architecture (Section 2.2.2) supports simultaneous access to the sensors in a transparent manner from the main processor as well as the co-processor. The co-processor's ability to access the sensors is useful in variety of use cases such as integrity verification. In that regard, it is important to guarantee that the sensor values measured by both the main

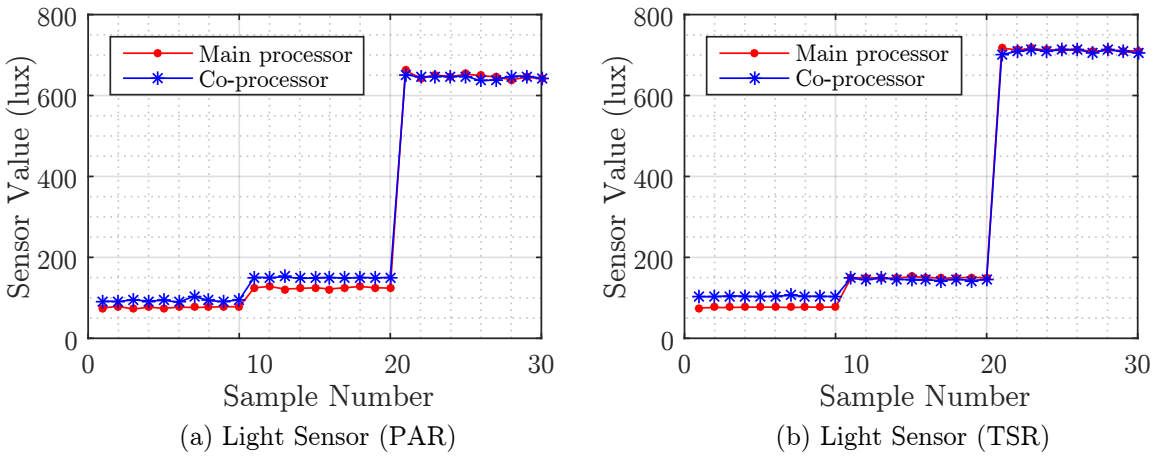
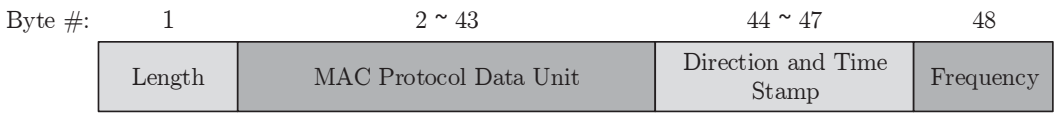


Figure 2.9. Comparison of the light sensor data measured by the main processor and the co-processor in the SPI-SNOOPER platform

processor and the co-processor are in an acceptable error range. Figure 2.9 shows the sensor data simultaneously measured in every 5 seconds by the main processor as well as the co-processor under three different lighting conditions. As can be seen in the figure, their measurement results are in a strong agreement.

Overhead of logging

The co-processor on SPI-SNOOPER is able to log the captured packets using two types of log format depending on the need for entire contents of a packet. In particular, the co-processor logs either entire MAC protocol data unit (MPDU) of the packet (*format A log* as shown in Figure 2.10(a)) or essential information (*i.e.*, source address and destination address) extracted from the MPDU (*format B log* as shown in Figure 2.10(b)). In both cases, the length, direction (incoming or outgoing), timestamp, and frequency of the packet are included in the log. Among those, the frequency denotes how many times the same packet is captured consecutively, and this information is useful if re-transmission is enabled.



(a) Format A Log: logging the entire contents of a packet

(b) Format B log: lgging the metadata of a packet

Figure 2.10. Two types of the log formats in SPI-SNOOPER

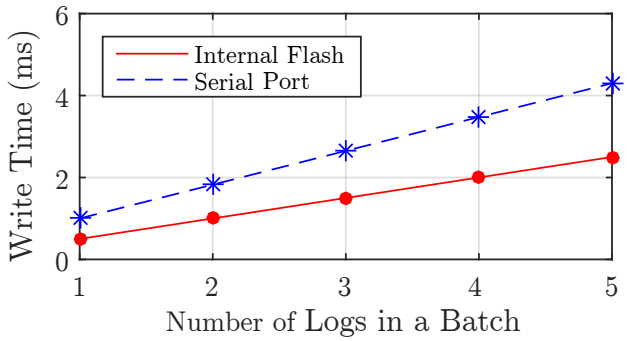


Figure 2.11. Time required to write logs onto flash and serial port

The logs can either be stored onto the internal flash or be forwarded using the serial port. Although writing to the flash is faster than the serial port (baud of 115,200 bps) as shown in Figure 2.11, in the case of using the internal flash, all the interrupts are disabled while data is being written to the flash. Hence, in order to minimize the interruption caused by the writing process, we make use of internal RAM as a temporary storage and migrate the logs packets to the flash in batches. For instance, the logs described in the example in Section 2.4.1 were stored in the internal flash in batches of 5 logs. As the number of logs in a batch increases, the flash writing time also linearly increases as shown in Figure 2.11, and this is also true for the serial port. With the 256 KB internal flash of the co-processor, it is possible to store up to 5,461 *format A logs* or up to 32,768 *format B logs*.

Power consumption

In this section, we characterize the power overhead incurred by the components added to the original TELOS mote in the transformation to SPI-SNOOPER. In practice, it is hard to measure the power consumption of only the added components since the extension to SPI-SNOOPER is built onto an operational TELOS mote. Therefore, we measured the power consumption of SPI-SNOOPER as well as the original TELOS mote separately and then, compared. Each main processor of the platforms are running the same Contiki application, consecutively broadcasting 10 packets for the duration of 0.5 s in approximately every 10 s. The supply voltage of both the TELOS mote and SPI-SNOOPER were 3 V, and all the features of SPI-SNOOPER were enabled.

Figure 2.12 shows the measurement results for both the TELOS mote and SPI-SNOOPER. For better visibility, the two graphs have certain amount of offset in time although they are running the same application. As seen in the figure, the average current consumption was around 18.15 mA and 22.75 mA for the TELOS mote and SPI-SNOOPER, respectively. This results in the average offset of 4.6 mA, which is translated into SPI-SNOOPER's power contribution to the transformation from TELOS mote to SPI-SNOOPER. It is the maximum power overhead for a worst case scenario in that no power-saving techniques were applied to the co-process that accounts for the largest portion of the added power overhead. The reason for showing the worst case example is to give an idea how much power budget to allocate to accommodate the co-processor-based technique. In fact, the specific microcontroller that we use as the co-processor [8] supports 5 different software-configurable low-power modes. The support for low-power modes is common feature in modern microcontrollers, and a microwatt-level power consumption is easily achievable without any hardware support.

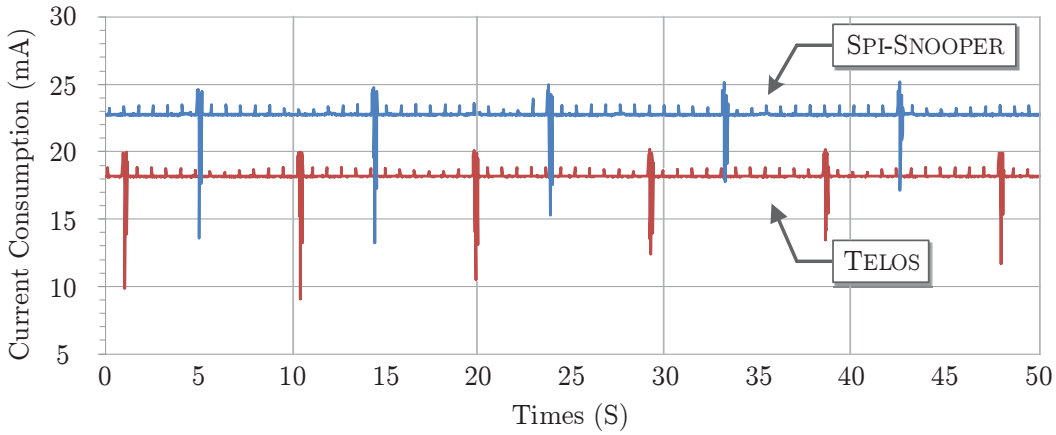
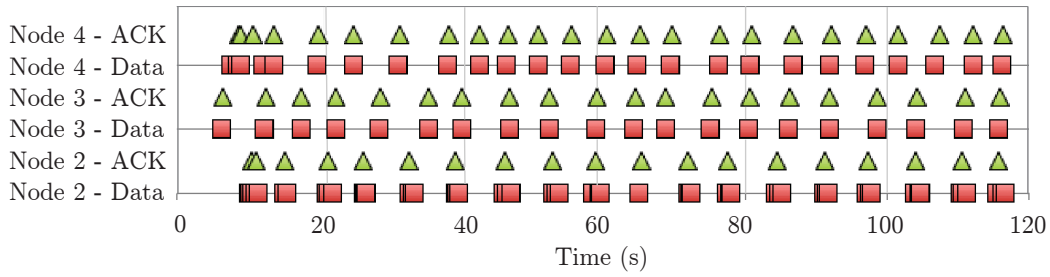


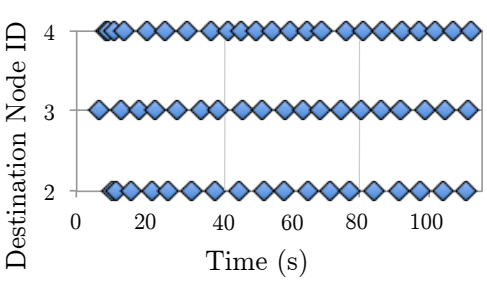
Figure 2.12. Current consumption of TELOS and SPI-SNOOPER

2.4.2 Logging network communication

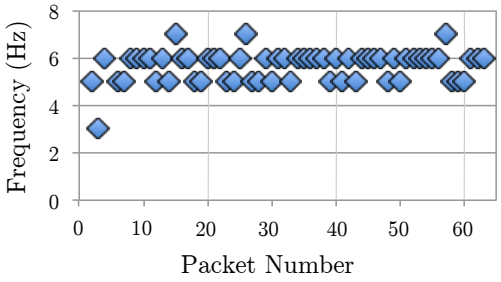
In this section, we conduct two experiments showing how the SPI bus transactions logged onto the on-board flash or serial port can later be used to analyze the network communication that took place. For the first experiment, a single-hop start topology network was configured using an SPI-SNOOPER mote as a central node (node 1) and three TELOS motes as peers (node 2-4). The three TELOS nodes are sending packets to the SPI-SNOOPER mote with a random interval of 4-6 seconds. We used `contikimac` in the RDC layer with packet acknowledgment and retransmission enabled. The size of each data and acknowledgment packet (PPDU) was 45 bytes and 5 bytes respectively. We logged entire contents of all incoming and outgoing packets to the flash in the form of the *format A log*. Later on, we retrieved the stored logs from the flash and reconstructed the traces of the incoming and outgoing packets of the SPI-SNOOPER mote using the timestamps included in the logs. The reconstructed time-lines for all the incoming and outgoing packets are shown in Figure 2.13(a) and Figure 2.13(b), respectively. Note that only outgoing data packets are shown in Figure 2.13(b) since the radio was configured to send acknowledgment for incoming data packets automatically. Since retransmission was enabled, multiple attempts were made to send a data packet until acknowledgment was received or timeout



(a) Incoming data packets and acknowledgements.



(b) Outgoing data packets.



(c) Frequency of outgoing packets.

Figure 2.13. Trace of incoming and outgoing packets generated from the logs collected by the co-processor in the Spi-Snooper platform.

occurred. Figure 2.13(c) shows the frequency of such attempts to send each packet from the central node. For the second experiment, a Spi-Snooper mote and a Telos mote are configured to broadcast packets with extremely high frequency of 64 packets/second. The size of each packet was 21 bytes. The reconstructed time-line for the second experiment is shown in Figure 2.14. The figure shows that the number of incoming packets was less than the number of outgoing transmission attempted. This is because many of incoming packets did not arrive due to high rate of collision. The two experiments demonstrate how the architectural supports of Spi-Snooper enable a fine-grained visibility into low-level details about network communication, which otherwise would have been difficult or impossible to achieve.

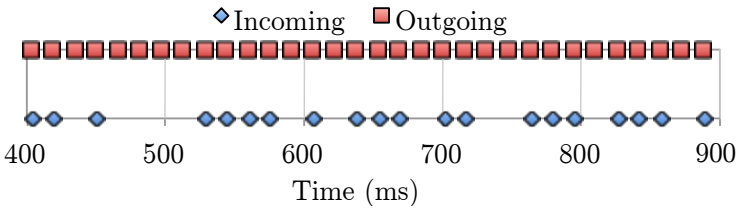
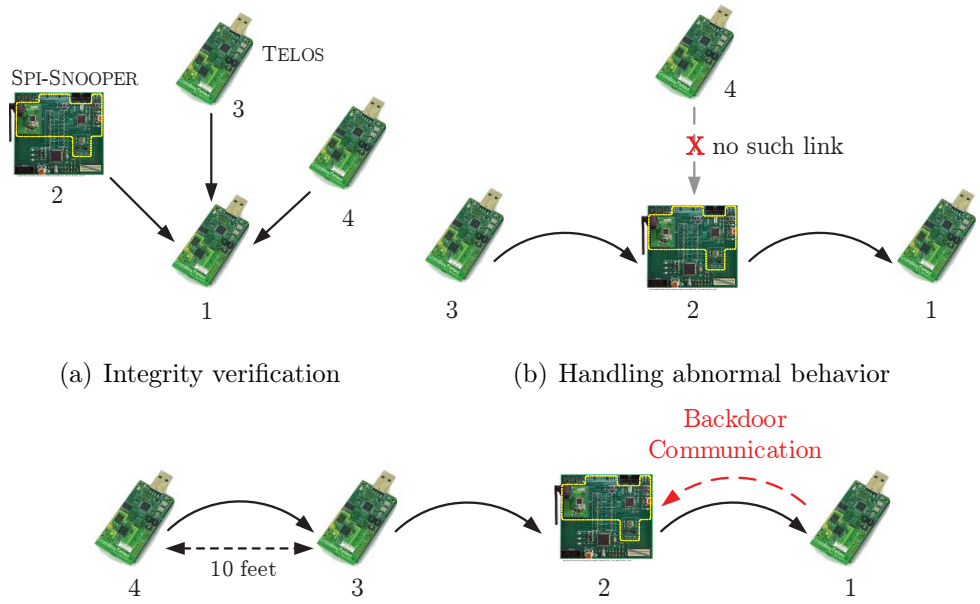


Figure 2.14. Trace generated from logs for high frequency packet transmission between two nodes



(a) Integrity verification

(b) Handling abnormal behavior

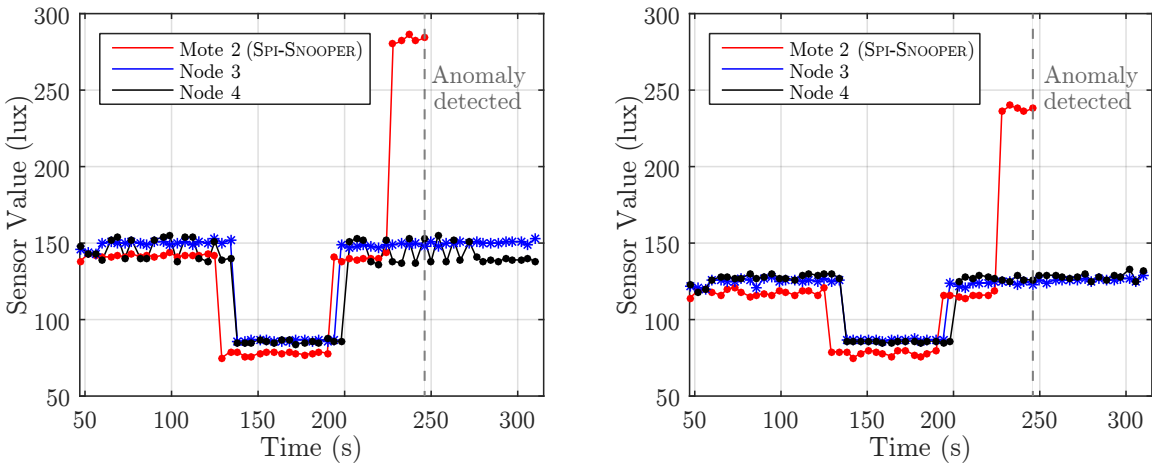
(c) Providing an emergency backdoor

Figure 2.15. Topologies used for the experiments

2.4.3 Integrity verification

With prior knowledge about the device behavior and the format of the application packet, SPI-SNOOPER can be used to verify the integrity of data (*e.g.*, range check) contained in the application packet. For example, the light sensor (TSR and PAR) values contained in an outgoing packet can be analyzed by the co-processor, which also has an independent access to the same light sensors. In fact, this example is illustrated

in this section. Figure 2.15(a) shows the network topology used for this experiment. All the devices are TELOS motes, except for the node 2, which is SPI-SNOOPER. All the motes were running `sky-collect`, one of the example user applications that comes with the original Contiki distribution. The motes running this program periodically collect sensor data (*e.g.*, light, temperature, *etc.*) and forward them to the base station (node 1) in every 4-6 seconds. Once the co-processor detects that an application packet containing sensor data is being transmitted, it independently accesses the same set of sensors that the main processor used to collect the sensor data and compares its own reading with the data contained in the packet. Specifically, We maintain a history of the most recently sent 5 sensor data and compare them with the values collected by the co-processor. If all the values sent within the time window differ by more than 20% from their corresponding values collected by the co-processor, we conclude that the device is malfunctioning and disconnect the main processor from the radio using the crossover logic. In this example, we just assume that the sensors are working properly, but the error handling policy can be application-specific. To emulate the case that the sensor data measured by both the main processor and the co-processor differ more than 20%, we intentionally programmed node 2 in such a way that, after some time, it starts multiplying the collected values by 2 before sending them. As can be seen in Figure 2.16, all the motes start sending the sensed values for the two light sensors in normal indoor light condition at time zero. We keep our lights off for the period of 125-190 seconds. The sensed values from all the motes in Figure 2.16 also corroborate that. Around the 220th second, node 2 starts sending values that were significantly higher than the other motes even though all motes were nearby. The co-processor is able to determine that the main processor is malfunctioning and disconnects it from the radio. In the following section, we show how SPI-SNOOPER can keep the communication alive with other motes in the network even after the main processor is disconnected.



(a) Light sensor (TSR) (a) Light sensor (PAR)

Figure 2.16. Average light intensity values received by the base station in the experiment described in Section 2.4.3

2.4.4 Handling abnormal behavior

The co-processor in an Spi-Snooper mote can analyze incoming and outgoing packets and determine certain types of malicious/buggy activities caused by the main processor. We demonstrate such an example in this section. Figure 2.15(b) shows the experimental setup. Node 3 is sending packets to node 1 in every 4-6 seconds using the mesh routing protocol of the Rime communication stack. Since node 1 and node 3 are outside their wireless coverage, packets are being routed through node 2, which is an Spi-Snooper mote. After some time, node 1 starts receiving packets originated from node 4. However, there are no incoming packets destined to node 2 from node 4. Hence, this could be due to a bug or a malicious behavior (*e.g.*, a wormhole attack) of the main processor of node 2. Until the number of such incident exceeds certain threshold (5 occurrences in this experiment), the co-processor forwards packets that seems to have originated from node 4. However, above the threshold, the co-processor takes over the control of the radio using the crossover logic (Section 2.2.5). Even in this case, the co-processor controls the radio and keeps relaying the packets originated from other nodes so that the connectivity of the network is maintained. In order to

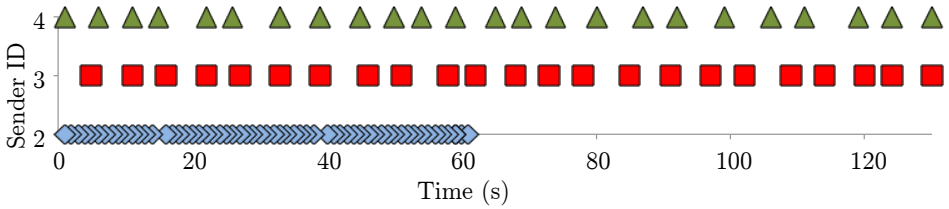


Figure 2.17. Packets received by the base station in the experiment described in Section 2.4.5

support seamless take-over, while the main processor is healthy, the co-processor has built and maintained its own routing table from the history of packet forwarding observed via the SPI bus snooping. Hence, it has a routing table similar to that of the main processor, and the table contains information such as address of immediate destination, hop count, *etc.* This experiment demonstrates how Spi-Snooper helps to isolate a malicious or buggy node from the network and to keep providing services that are essential to maintain the connectivity of the network.

2.4.5 Providing an emergency backdoor

Spi-Snooper allows to establish a backdoor channel that is transparent to the main network. A special command or query can be sent over the backdoor channel without intervening the main network. The network topology used for the experiment is shown in Figure 2.15(c). All the motes are Telos motes, except for the node 2, which is an Spi-Snooper mote. The motes (node 1-3) are sending packets to the base station (node 1) using the mesh routing protocol of the Rime communication stack. While node 1 and 2 are sending packets in every 4-6 seconds, node 2 is sending packets at a much higher rate (1 packet/second) as shown in Figure 2.17. After about a minute, the base station decides not to receive any packet from node 2, which may corrupt the network due to the high data rate resulting in excessive packet collision. Then, the base station send a special command 'disconnect and forward' to disconnect the main processor in node 2 but to keep forwarding packets from

other motes. The special command can only be recognized by the co-processor in node 2, and the co-processor responds to the request as instructed. As shown in the figure, no more packets are delivered from node 2 to the base station, but the base station still receives packets from other motes. The format of the special command packet used for the backdoor communication is the same as the regular application packet. In order for the special packet to be addressed only to the designated co-processor, the special command packet has a destination address that is outside the range of the main network. For example, we allocate the addresses space of 1-100 to the main network, and the addresses of the co-processor start with 101. That way, we can differentiate certain co-processors from another if a network includes multiple Spi-Snooper motes. Specifically, in this experiment, the co-processor on the Spi-Snooper mote (node 2) has '102' as its own address. Since the special command packet has the same format as the application packet, it is also delivered to the main processor. If an added protection is required, the special command packets can be encrypted so that the application running on the main processor will not be able to understand the actual meaning of the packet. In practice, different types of addressing mechanism can be used. In fact, the main reason that we used the aforementioned addressing mechanism is that this is the best way to avoid unnecessary flooding in the network that uses the Rime communication stack. For instance, we can further enhance backdoor communication by using uIP [12] that intrinsically supports port-based packet classification.

2.5 Related Work

The concept of using co-processors (or watchdog processors) has been prevalent in the traditional computers and high-performance computing systems for decades [15–18]. The typical use of the co-processor is to supplement the computational inefficiency of the main processor (*e.g.*, graphics co-processor) or to provide an additional layer of security to the main processor (*e.g.*, cryptographic co-

processor). [19] is a survey of concurrent system-level error detection techniques using a watchdog processor for traditional PC-based systems. However, none of these solutions were designed to work with networked embedded systems such as WSNs. There are few other works that are applicable to WSNs. Some of them [20–23] are purely software-based or require pre- or post-processing of execution traces. For example, LiveNet [1] suggested the concept of a deployment support network (DSN), where the authors just connected two individual TELOS nodes (target node and monitoring node) via wires in order to communicate and implicitly exchange certain data of interest by using software routines that reside on both the target node and the monitoring node. Therefore, LiveNet is neither transparent nor efficient in terms of energy and cost. In comparison, some other class of work use hardware-software approach. FlashBox [24] is a hybrid hardware-software solution where an additional microcontroller is used to log the occurrences of interrupts in the main processor. For the logging to work, applications have to be compiled using a modified version of `avr-gcc`. A more recent work, Aveksha [25], is another hardware-software approach that utilizes on-chip debug module to monitor the processor in a TELOS mote. It monitors and records only the internal state of the processor. However, unlike Aveksha, SPI-SNOOPER is not only able to monitor the entire system (*e.g.*, network communication, sensors, *etc.*), but also able to actively involve the system operation to ensure reliable and robust operation.

2.6 Summary

We have presented SPI-SNOOPER, a co-processor-augmented hardware-software approach that enables monitoring and controlling of the SPI bus communication between the main processor and the radio in a mote. The co-processor monitors every network communication taking place in a mote in a transparent manner using snooping technique. Upon the detection of anomaly, the co-processor actively engages the device operation and takes over the control of the radio using the crossover logic.

With the proposed architectural support of SPI-SNOOPER, we are not only able to achieve a fine-grained visibility into system operation but also able to take an immediate action to prevent an error occurred in a device from corrupting the network. We have designed and developed a fully functional prototype that extends the base design of the TELOS [7] mote with the co-processor. We also have demonstrated how SPI-SNOOPER enhances reliability of WSNs with a set of practical examples and experiments using the prototype and TELOS motes running the Contiki [11] operating system. In principle, the proposed co-processor-augmented architecture is device-agnostic and not confined to either TELOS or Contiki [11]. With prior knowledge about the type of operating systems and the format of radio packets, the proposed scheme is generally applicable to any wireless sensor node platforms. In conclusion, SPI-SNOOPER provides a fine-grained post-deployment in-situ visibility into system operation, which is crucial to ensure quality services of WSNs.

3. SENERGY: MICRO-SCALE ENERGY HARVESTING FROM AN IDLE SENSOR

3.1 Introduction

Wireless sensor devices are heavily duty-cycled to minimize energy consumption. To further reduce the energy required for sensing, these devices often prefer passive sensors that produce output power proportional to physical quantity. Therefore, while a sensor output is not being sampled (*i.e.*, idle time), the output power from the sensor is unused and, hence, wasted. This observation inspires the use of a micro-scale energy harvesting technique that collects the energy being wasted during idle time to power the device. The micro-scale energy harvesting is one of the powering techniques that operate low-power electronic components or systems using ambient energy. In a micro-scale energy harvesting system, an energy harvester converts environmental energy into electrical energy and regulates the converted energy into a usable form (*e.g.*, voltage regulation). Often, the energy harvester stores the energy being harvested into a storage element (*e.g.*, battery, supercapacitor) until required amount of energy is harvested. Hence, micro-scale energy harvesting is well-suited for a duty-cycled application that allows sufficient time for energy harvesting.

However, the sensor-based micro-scale energy harvesting introduces two fundamental challenges. First, the output power from a sensor is time-varying and often minuscule. Unfortunately, current energy harvesting solutions are designed for specific performance requirements, such as high voltage gain [26,27], high drive capability [28], and high efficiency [29], and not optimized for interfacing with such a low-capacity power source. Second, unlike the existing solutions that employ a dedicated energy transducer, the sensor-based energy harvesting utilizes a sensor as both a sensing element and a power source. Therefore, it is almost impossible to explicitly separate the

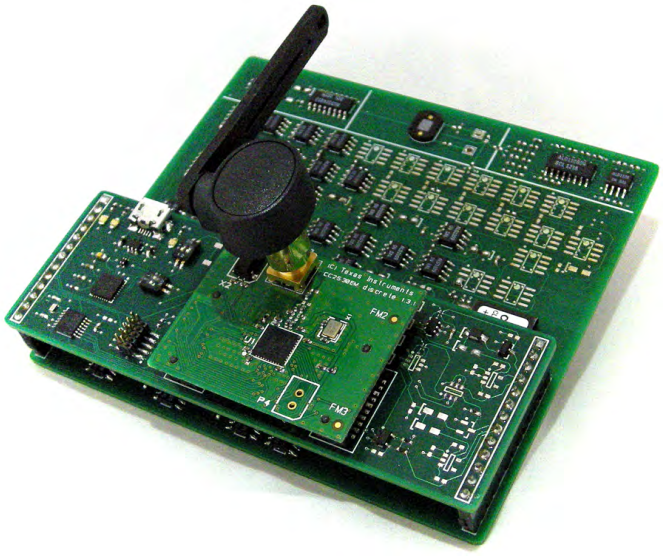


Figure 3.1. Photograph of SENERGY, a batteryless energy-neutral wireless sensing platform that utilizes a photodiode sensor as both a sensing element and a power source

energy harvester from an application, and a systematic method has to be presented to address an implication of the dual use of a sensor. In order to address the challenges, we propose SENERGY, a batteryless wireless sensing platform that is powered solely by the energy harvested from an idle sensor. We prove the concept by implementing a board level prototype of SENERGY using only off-the-shelf components and a photodiode sensor (Figure 3.1). Specifically, we make the following contributions:

- We propose an exponential topology charge pump architecture that works with an ultra-low-capacity power source that has an output voltage and current as low as 250 mV and 6 µA, respectively. We use the charge pump to design a system powered by a sensor.

- We present SENERGY, a board level prototype of the proposed energy harvester architecture and a 2.4 GHz wireless connectivity-equipped target system using off-the-shelf components. A photodiode is multiplexed to function as both a sensing element and a power source.

- We demonstrate the utility of SENERGY by implementing and evaluating two usage scenarios: 1) adaptive transmission of sensor data, and 2) perpetual operation of a mission-critical low-power sub-system (*e.g.*, real-time clock).
- To the best of our knowledge, this is the first design that proposes a complete self-powered batteryless solution in which the entire system is built only with off-the-shelf components and is powered by a sensor.

The remainder is organized as follows. Section 3.2 discusses the charge pump architecture proposed and used in our energy harvesting solution. Section 3.3 and 3.4 present hardware and software architectures of the energy harvester and accompanying target system, respectively. Section 3.5 describes a rigorous evaluation that we performed on SENERGY with various usage scenarios demonstrating the utility and capability. Section 3.6 describes related work and Section 3.7 summarize this work.

3.2 Charge Pump Architecture

The output voltage of a passive sensor is often very low and ranging in a few hundreds of millivolt. Additionally, the output voltage is time-varying as it depends on the physical phenomenon being sensed (*e.g.*, light intensity). Hence, the low voltage has to be boosted to a general operating voltage (*e.g.*, 3.3 V) before being utilized. A charge pump (CP) [30] is a voltage converter that is used to create a higher or lower (in case of negative polarity) voltage by employing a network of capacitors interconnected with switching devices such as MOSFETs and diodes. Depending on the charge pump architecture, two or more non-overlapping clock signals control the switching devices so as to efficiently share charge among the capacitors. As compared to inductive voltage converters, such as Boost or Flyback converters, charge pumps are typically preferred since it is simper in design, smaller in footprint, and lower in cost [31]. Therefore, we decided to base our energy harvester around a charge pump architecture. In this section, we explain the proposed charge pump architecture and evaluate its functionality.

3.2.1 Limitations of existing charge pump architectures

Over the past decades, several CP architectures (discussed in Section 3.6) have been proposed to optimize the CPs for different aspects such as efficiency, size, cost, *etc.* However, the ability to interface with a low-capacity power source (*e.g.*, passive sensor) has seldom been explored. A key property required for interfacing a low-capacity power source is a low-power control logic. A control logic is a mandatory circuitry generating control clocks to alternate the architectural formation of a CP to boost input voltage. Despite the functional importance, if the power consumption of the control logic is too high (*i.e.*, quiescent current is too high), very little power is left available for harvesting. To address this issue, we propose a custom exponential CP architecture that is controlled by a sub-threshold clock generator, whose power consumption is as low as 38 nW with a board level implementation built using off-the-shelf discrete components.

3.2.2 Proposed charge pump architecture

The core building block, henceforth referred to as a Complex Voltage Doubling Block (CVDB), of the proposed exponential CP architecture and its basic operation are illustrated in Figure 3.2 and Figure 3.3, respectively. The CVDB is a self-contained generic voltage doubling block in the sense that it doubles input voltage regardless of an overall CP architecture where it belongs to. A CVDB consists of two input ports and one output port. Every successive half clock period, the block transforms from a parallel configuration of capacitors to a series configuration and vice-versa by using two non-overlapping control clocks (generation of the control clocks will be discussed in Section 3.2.4). As shown in Figure 3.3(a), during the first half clock period ($\phi_1 = H$), the capacitors are in a parallel formation, and C_1 and C_2 get charged to V_{in}. During the following second half clock period ($\phi_1 = L$), the capacitors are connected in series summing up the voltages V_{C_1} and V_{C_2} as shown in Figure 3.3(b). Thus, the output voltage gets boosted to $2V_{in}$ in the steady state.

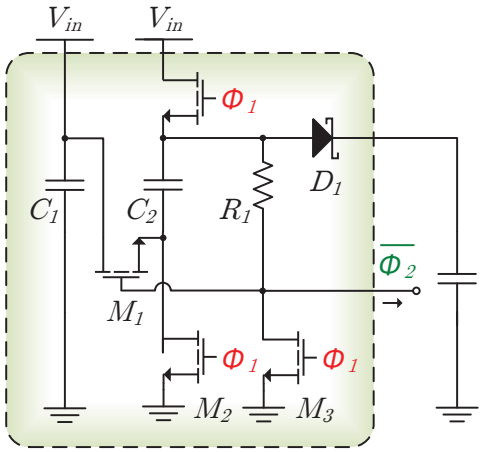
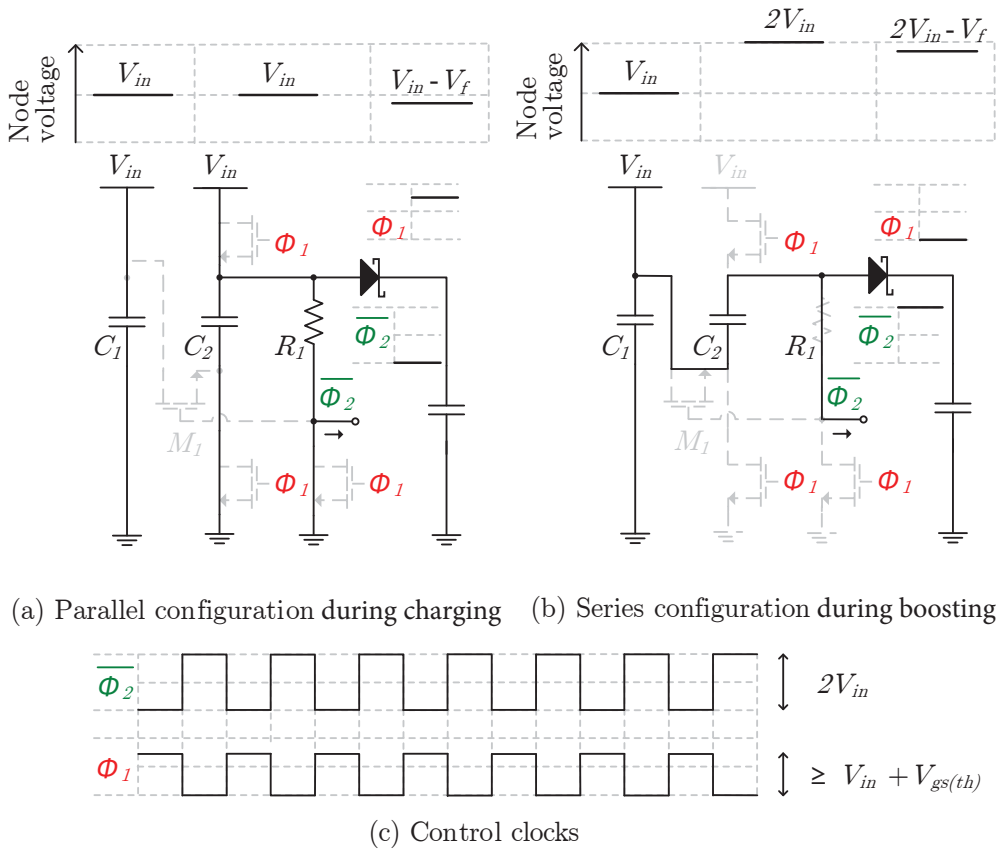


Figure 3.2. The core building block (CVDB) of the proposed charge pump architecture

Since the voltage doubling is accomplished by the MOSFET switch M_1 that connects C_1 and C_2 in series during the boosting phase, the drain and source voltages of M_1 have to be equal to completely transfer charges between the top plate of C_1 and the bottom plate of C_2. This requires the gate voltage (V_g) to be greater than or equal to $V_{C_1}+V_{gs(th)}$. In other words, M_1 requires V_g greater than that of the power source. To address this issue, the interconnecting MOSFET switch M_1 acquires the required V_g from a static resistive-load inverter (R_1-M_3) operated using the boosted voltage output ($\overline{\phi_2}$). Thus, the magnitude of the back-gating voltage is always sufficient to completely turn on M_1 if the voltage level of the input ports are identical and $V_{gs(th)}$ is less than or equal to V_{in}. After the diode that blocks reverse current flow, a load capacitor connected to the output port can be charged up to $2V_{in} - V_f$, where V_{in} is input voltage, and V_f is forward voltage drop of the diode.

3.2.3 Example of two-stage configuration

In this section, an example of two-stage exponential CP configuration is demonstrated to give an idea how the core building block (CVDB) introduced in Section 3.2.2



(a) Parallel configuration during charging (b) Series configuration during boosting

(c) Control clocks

Figure 3.3. Basic operation of the CVDB

constructs an exponential CP architecture. The two-stage exponential CP configuration and its basic operation are shown in Figure 3.4. As seen in the figure, it includes a Fundamental Voltage Doubling Block (FVDB) in addition to the CVDB. The FVDB is identical to the CVDB, except for the absence of the back-gating inverter. The FVDB shares the inverter output, $\overline{\phi_2}$, of the CVDB in the same stage to avoid additional power dissipation. Thus, each stage requires only one CVDB, and the remaining voltage doublers can be configured using FVDBs.

The output voltage, $2V_{in} - V_f$, of both the CVDB and the FVDB in the first stage is fed to another CVDB in the second stage as its inputs. The second stage CVDB produces output voltage of $4V_{in} - 3V_f$ and, thus, accomplishes exponential voltage gain. The output voltage of a particular stage can be expressed as (3.1), where N

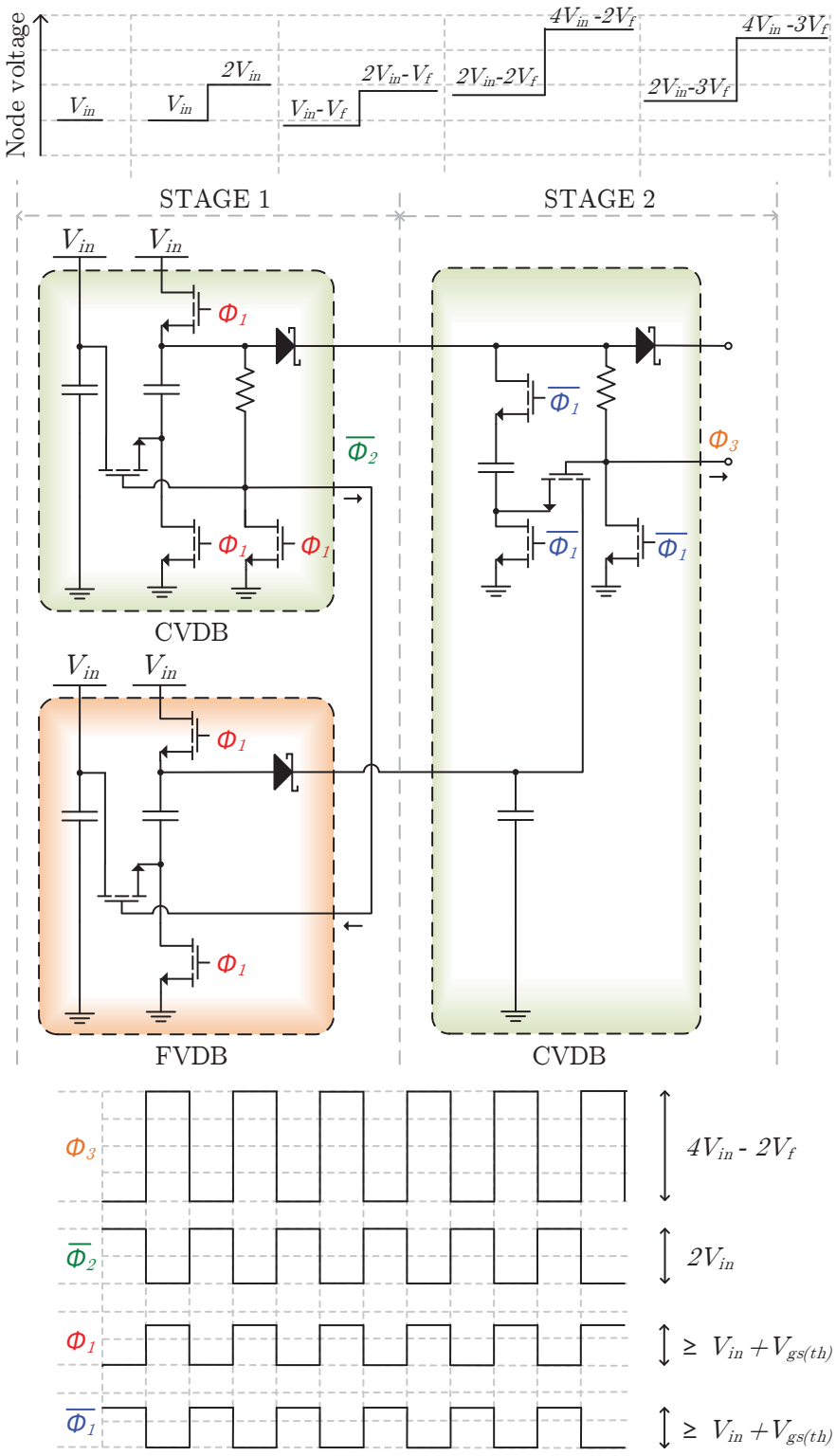


Figure 3.4. Example of a two-stage exponential charge pump configuration and its operation

denotes the stage number, V_{in} is the input voltage, and V_f is forward voltage drop of a diode.

$$V_{out} = 2^N(V_{in} - V_f) + V_f \tag{3.1}$$

Each voltage doubling block alternates its hardware architectural formation between parallel and series configurations for the capacitors. While the voltage doubling blocks in the same stage behave the same sequence of parallel and series configurations, those in the adjacent stage behave the opposite sequence. Considering the two-stage configuration as an example, while the capacitors of the voltage doubling blocks in the first stage are in series configuration to double the capacitor voltage, the capacitors of the CVDB in the second stage are in parallel configuration to be charged up to the output voltage of the first stage voltage doubling blocks. During the following half clock period, the CVDB in the second stage doubles the voltage of its own capacitors. As a consequence, the capacitors of the CVDB in the second stage function as intermediate energy storage elements. Considering the minuscule amount of power available from the source (*e.g.*, passive sensor), this is one of the most important architectural features contributing to the overall success of the proposed exponential CP architecture.

3.2.4 Control unit design

The CP architecture requires a control unit that generates two non-overlapping clocks that are used to alternate the circuit formation between series and parallel configurations. The control unit is an essential component of all CP architectures. However, the power consumption of the control unit translates into an overhead and often renders a CP infeasible or impractical to use due to the little power left available after operating the control unit. For example, the control unit in Dickson CP [32] has to be driven with a high amount of current to prevent the clock used for voltage doubling from collapsing. To address this problem, we designed an ultra-low power

control unit consisting of a sub-threshold nanowatt level RC oscillator that is followed by a clock magnitude amplifier. Figure 3.5 shows the architecture of the control unit.

The nanowatt level power consumption is achieved by exploiting the sub-threshold characteristics of N-channel enhancement MOSFETs. A MOSFET is operating in the sub-threshold region if the gate to source voltage, V_{gs}, is lower than its threshold voltage $V_{gs(th)}$. In this region, most applications consider the MOSFET to be turned off since the current flowing through the MOSFET is negligibly small. However, a small amount of current still flows through, and an exponential relationship with V_{gs} is observed. Therefore, even in the sub-threshold region, we have control over the drain to source current I_{ds}. As an example, for a particular commercial MOSFET, ALD110904 ($V_{gs(th)}$=0.4 V) from Advanced Linear Devices, that we use for SENERGY, the RC oscillator can operate down to 0.14 V [33]. The frequency of the clock generator is generally determined by $f_{osc} = 1/(2\pi R_5 C_{osc})$. The charging of C_{osc} is limited by R_3 + R_4 and the discharging of C_{osc} is limited by the current drive of M_3. Once the circuit starts oscillating, two non-overlapping clocks are produced using the output buffer amplifiers M_4 and M_5. Subsequently, the clock magnitude amplifier increases the magnitude of the clocks to a predefined level. From an architectural standpoint, the clock magnitude amplifier is a linear topology CP, which is built using the CVDB as a core building block.

In the following sections, we discuss and evaluate the hardware architecture and implementation of the SENERGY platform. The SENERGY platform consists of an energy harvester board (EHB) and a target board (TB), which share a passive sensor as both a sensing element and a power source. The EHB harvests the output power of a passive sensor using the proposed exponential CP. Then, the harvested energy is stored into an energy storage element during idle time to intermittently power the entire TB or perpetually powers a mission-critical low-power sub-system of the TB.

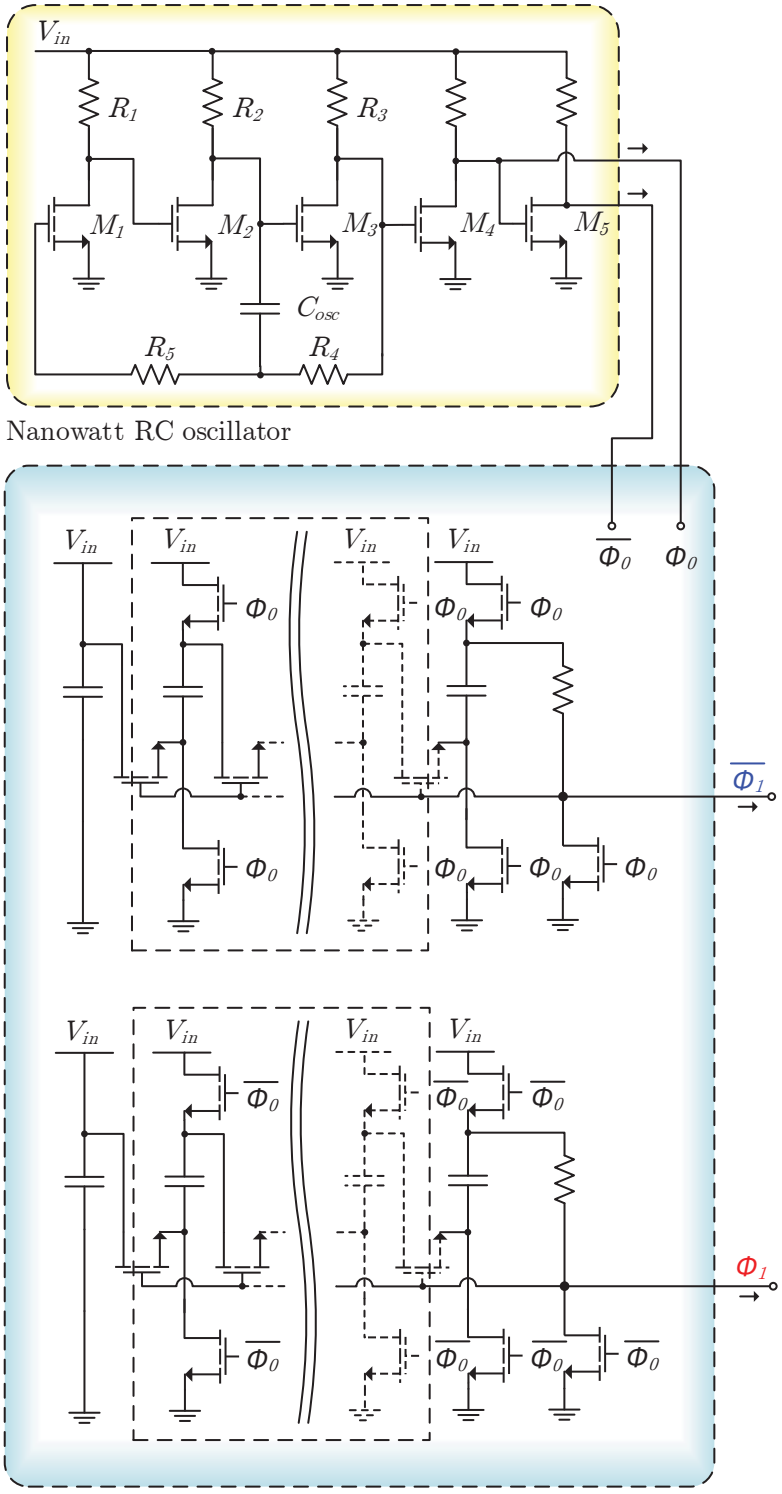


Figure 3.5. Control unit architecture

3.3 SENERGY Energy Harvester Board

The prime design objective of the SENERGY EHB is to provide a hardware architectural support for the dual use of a passive sensor as both a sensing element and a power source. Figures 3.6(a) and 3.6(b) depict the hardware block diagram and implementation of the EHB, respectively.

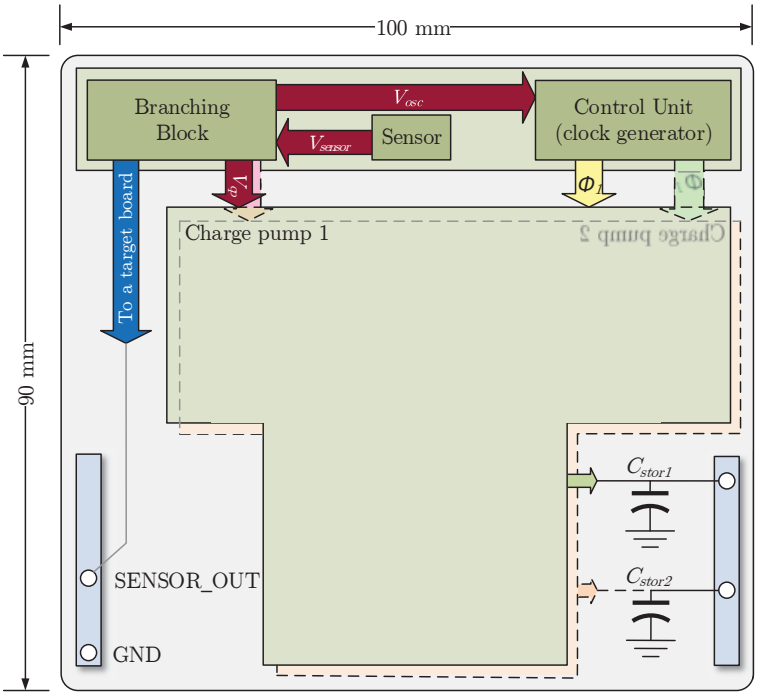
3.3.1 Hardware architecture

As shown in Figure 3.6(a), the hardware architecture of the EHB consists of four blocks (*i.e.*, a passive sensor, branching block, control unit, and a pair of the proposed exponential charge pump). The branching block diverts the output power of a passive sensor either to the energy harvester, which includes the control unit and a pair of the proposed exponential CP, or to a target board for sensor data measurement. The pair of the proposed exponential CP operating on opposite phases of the control clock fully utilizes the output power of the sensor for energy harvesting. To be specific, one CP is in the charging phase while the other CP is in the boosting phase, and vice versa according to transition of the control clock. Finally, the output of each charge pump is stored onto two different capacitors C_{stor1} and C_{stor2}. The energy available on both the capacitors is used to power a target system. The EHB hardware architecture is further demonstrated in detail with an implementation in the following section.

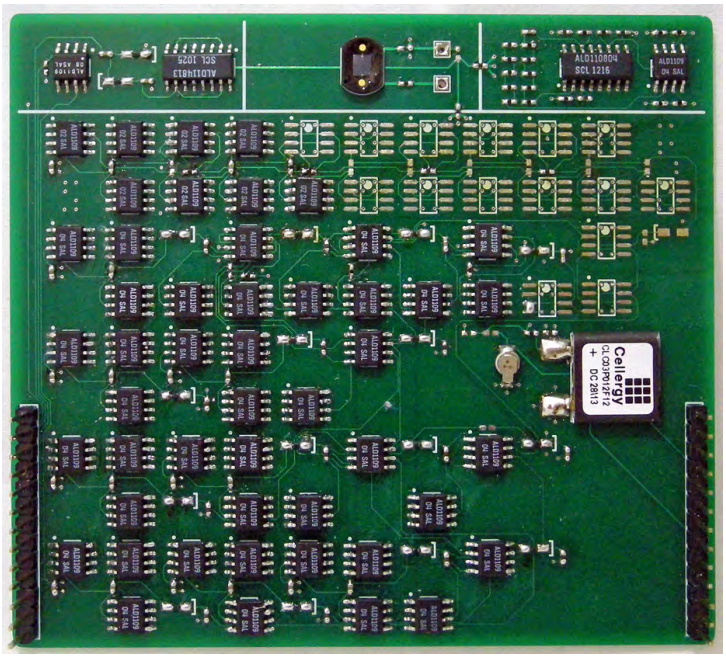
3.3.2 Hardware implementation

We choose a photodiode S1133-01 from Hamamatsu Photonics as the sensor for the EHB. The short circuit current of a photodiode varies linearly with the intensity of incident light. The measured short circuit current, I_{sc}, of S1133-01 is 2.1 µA under 100 lx. The open circuit voltage, V_{oc}, is typically 430 mV for the light intensity of an ordinary office environment and 600 mV for outdoor on a cloudy day, respectively.

[1] Dashed line denotes that the components are placed on the back side [2] The other charge pump (CP2) is located at the back of the board



(a) Hardware block diagram of the energy harvester board[1]



(b) Front side image of the energy harvester board[2]

Figure 3.6. SENERGY energy harvester board

Since the photodiode is functionally a current source, the output voltage diminishes as the current draw increases. Therefore, the input impedance of the energy harvester is carefully sized and controlled to prevent the voltage of the photodiode from collapsing.

Although a perfect isolation is preferred for the branching block to completely divert the sensor output either to the energy harvester of the EHB or to the TB, for certain types of sensors (*e.g.*, photodiode), such a mutually exclusive operation might not be necessary. In particular, the branching logic of the EHB employs a 1 Ω shunt resistor between the photodiode and the CPs as shown in Figure 3.11. While the impact of the shunt resistor is negligible for the energy harvesting as well as the sensor measurement, this architecture simplifies overall design. The TB can sample the sensor data by measuring the voltage difference across the shunt resistor (discussed in Section 3.4).

During energy harvesting, the output power of the sensor is delivered to the pair of the proposed exponential charge pump that has four-stage configuration for each. The organization of the voltage doubling blocks and the associated output voltage levels are shown in Figure 3.7. For the resulting output voltage, the diode loss accumulates exponentially with the number of stages and, hence, degrades the voltage boosting performance considerably. Equation (3.1) highlights the importance of minimizing diode loss in our CP implementation. Therefore, we use an ultra-low forward voltage drop Schottky Barrier diode[3] that has forward voltage drop of 100 mV for tens of µA of current. Additionally, the EHB employs low-threshold (0.2 V and 0.4 V) MOSFETs[4] for both the charge pump and the control unit to achieve efficient switching in the presence of low input voltages.

The energy being harvested is stored into the storage capacitors C_{stor1} and C_{stor2}, which are connected to the last stage of each CP. The size of the capacitors can be adaptively determined for application requirements. Furthermore, the two capacitors can be tied up to facilitate bigger capacity.

---

[3] NSR0240P2T5G from On-Semiconductor [4] N-channel enhancement mode MOSFET ALD110904 ($V_{gs(th)}$=0.4 V) and ALD110902 ($V_{gs(th)}$=0.2 V) from Analog Linear Devices

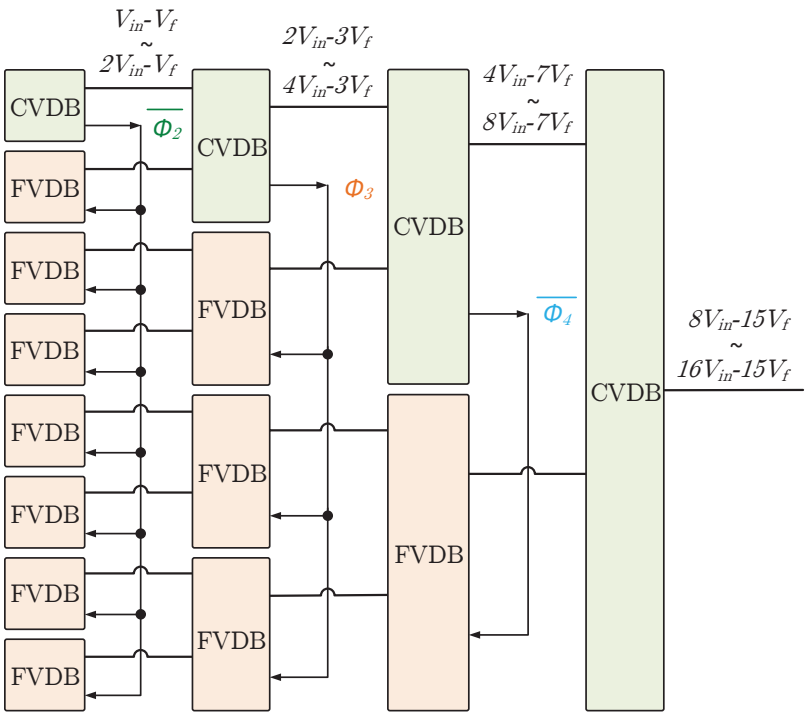


Figure 3.7. Four-stage configuration of the proposed exponential charge pump architecture implemented on the energy harvester board

3.3.3 Evaluation

The voltage boosting performance of the proposed exponential CP implementation is shown in Figure 3.8. To characterize the performance, we measured the input current, input voltage, and output voltage of the CP while the incident light intensity on the photodiode was varied. In the experiment, the output of each CP was tied up and stored into a capacitor of 1 µF. Despite the large number of discrete components count and the resistive inverters in the CVDBs, the CP begins to boost input voltages and current from 250 mV and 6 µA, respectively. The measured current consumption of the control unit alone was only 150 nA, and the difference to the minimum operating current is interpreted as a leakage resulted from the discrete component-based implementation. The evaluation result shows the architectural success of the proposed exponential CP, and the leakage current can be significantly minimized with an IC integration.

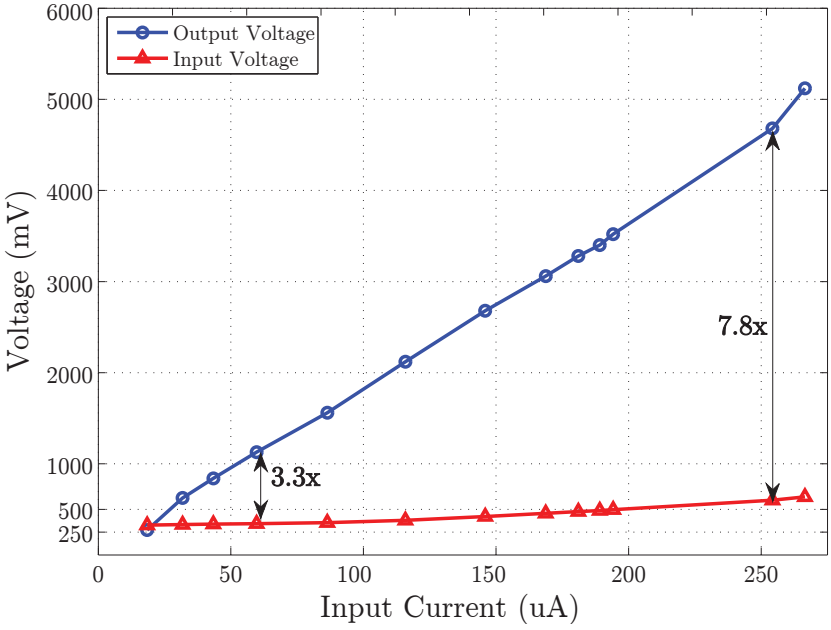


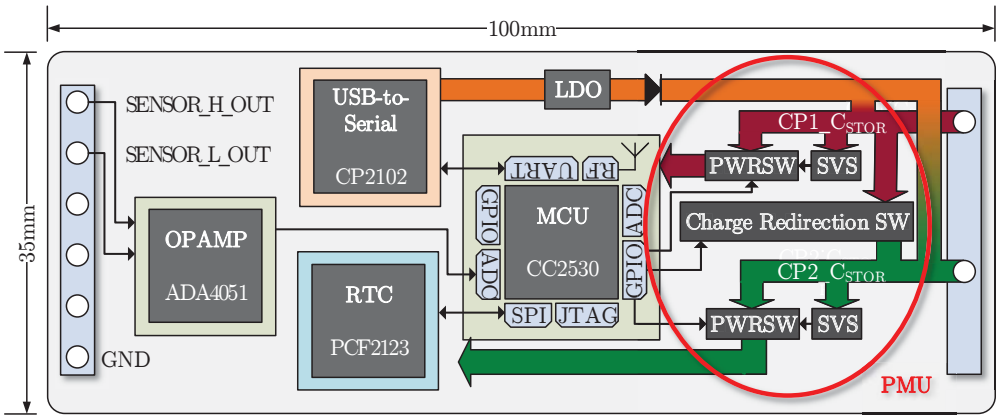
Figure 3.8. Charge pump voltages vs. input current

3.4 SENERGY Target Board

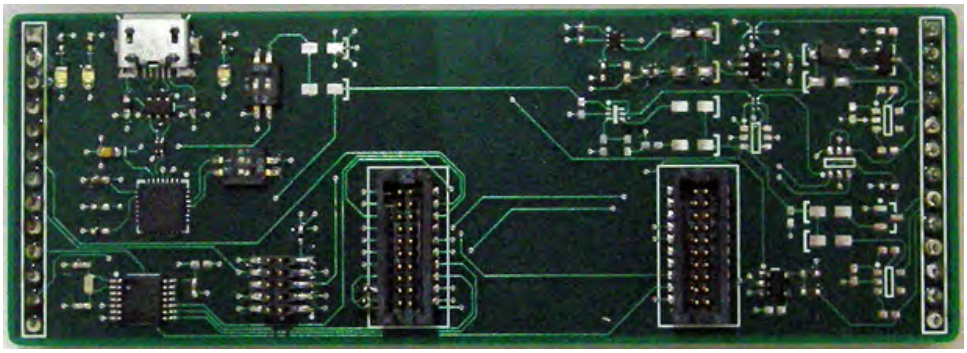
3.4.1 Hardware architecture and implementation

To demonstrate the systematic way of using a passive sensor as both a sensing element and a power source, we designed and implemented an application TB that is to be powered by the harvested output power of the sensor available on the EHB. The TB is designed to be plugged on top of the EHB so that they can share the photodiode and the energy storage elements. The hardware block diagram and the photograph of the implementation are shown in Figures 3.9(a) and Figure 3.9(b), respectively. As shown in the figures, the TB consists of five blocks (*i.e.*, RF communication module, power management unit, sensor readout circuit, real-time clock, and USB-to-serial interface).

[5] CC2530 RF module is not shown



(a) Block diagram of the SENERGY target board



(b) Photograph of the SENERGY target board[5]

Figure 3.9. Block diagram and photograph of the SENERGY target board

RF communication module

The TB is a typical wireless sensor node that transmits ambient light intensity over the 2.4 GHz ISM band using an RF SoC CC2530 from Texas Instruments. The TB uses a commercial CC2530 RF communication module to avoid design complexity associated with the RF design.

Power management unit

The power management unit (PMU) controls the current flow between the EHB and the TB. Specifically, the PMU consists of a charge redirection switch (CRS) and

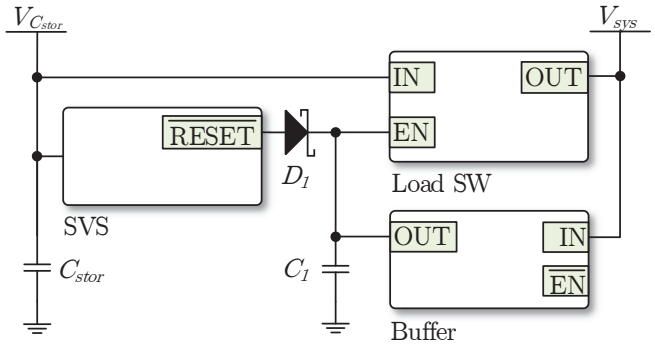


Figure 3.10. Power gating circuit in the power management unit of the SENERGY target board

a pair of power gating circuit (PGC) for each CP. It is important to isolate the TB from the storage capacitors of the EHB until a sufficient amount of energy has been harvested. The PGC in the PMU conducts the isolation with only 160 nA. The PGC includes a supply voltage supervisor (SVS)[6], a load switch[7], and a buffer[8]. The SVS constantly monitors the voltage level of the C_{stor} and releases the active-low reset pin once the capacitor voltage reaches a predefined threshold voltage. Although it is an efficient and effective way of activating the load switch, however, the load switch is immediately deactivated since the voltage of C_{stor} falls below the threshold voltage of V_{SVS_off} as soon as C_{stor} starts discharging to operate the TB. In order to prevent such an immediate shutdown, a buffer with a very short propagation delay (5.3 ns) is placed right after the load switch. The capacitor C_1 connected to the output pin of the buffer gets charged first and holds up the reset signal line of the SVS. As a result, the active duration of the load switch (*i.e.*, the duration of powering to the TB) can be controlled by accordingly sizing C_1 for application requirements. In addition, an application running on the TB also can deactivate the load switch by pulling down its enable signal upon the completion of a given task. Furthermore, in this case, the CRS can redistribute or balance the remaining energy of C_{stor1} and C_{stor2}. The CRS is a switch that bridges the current path between C_{stor1} and C_{stor2}. The charge

[6] TPS3839L30 from TI [7] AP2281 from Diodes [8] SN74AUP1G57 from TI

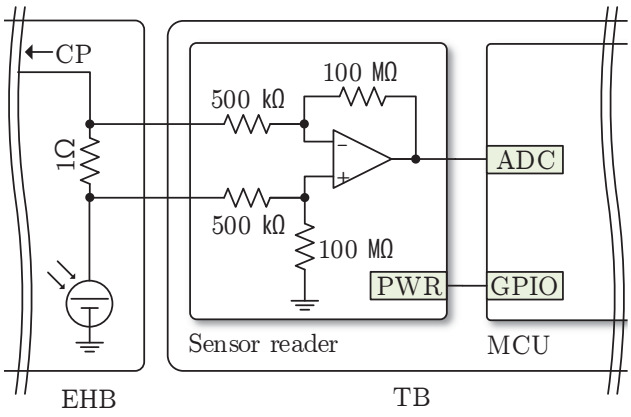


Figure 3.11. Sensor readout circuit

redistribution mechanism enables flexible reconfiguration of the storage capacitors by introducing a channel for energy exchange. The use of the CRS will be discussed in Section 3.5.1 with an example scenario.

Sensor readout circuit

The schematic view of the sensor readout circuit is depicted in Figure 3.11. The sensor readout circuit is fundamentally a differential amplifier that amplifies the voltage difference across the shunt resistor that is placed in the middle of the photodiode and the CP. The operational amplifier[9] that consumes only 13 µA is driven using the output power of a GPIO pin of the CC2530 MCU. With the 12-bit ADC and 1.15 V internal band-gap voltage reference of the MCU on the TB, it is possible to measure up to around 273,800 lx with a resolution of 133 lx.

Real-time clock andn USB-to-serial interface

To maintain a notion of time, the TB is also equipped with a real-time clock (RTC) IC[10] that consumes only around 100 nW of power. In addition, for RTC time

[9] ADA4051 from Analog Device [10] PCF2123 from NXP

synchronization, the TB also has a USB-to-serial interface[11]. Before deployment, the RTC time can be synchronized with the wall-clock time through this interface. While the TB is plugged into the USB interface, it is also possible to charge the storage capacitors using USB port power.

3.5 Experimental Evaluation

We evaluated the SENERGY platform using two experiments. Fundamentally, there are two operating strategies for applications depending on their power requirements. If the power requirement of an application exceeds the output capability of the CP, we can harvest the output power over time and intermittently power the application. Otherwise, if the power requirement is less than or equal to the output capability of the CP, we can perpetually power the application. The first experiment, namely adaptive sensing, is comprehensive in the sense that it includes both aspects of the intermittent and the perpetual powering. In the second experiment, we demonstrate an in-depth discussion about a perpetually-powered mission-critical low-power sub-system such as real-time clock.

3.5.1 Adaptive sensing

In this application scenario, we demonstrate how SENERGY enables sampling and transmitting of RTC-timestamped sensor data by utilizing the sensor as both a sensing element and a sole power source of the entire platform. The objective is to intermittently power the entire system to sense and transmit the light sensor data while perpetually power the real-time clock so as not to lose the information of the synchronized time.

Algorithm 1 describes the operational procedure to accomplish the objective. We define C_{tb} as the capacitor that supplies power from the EHB to the TB, and C_{rtc} as the dedicated power source of the RTC sub-system. Before deployment, the on-board

[11] CP2102 from Silicon Labs

Algorithm 1 Sense and Transmit Evaluation Application

Inputs

1: V_{th1}: Minimum voltage required to execute given application

2: V_{th2}: Critical RTC voltage

3: L_{th1}: Minimum light level required for EHB to harvest energy

Steps

```
1:  while TB is powered do
2:      if USB is connected then
3:          Charge C_tb and C_rtc;
4:          Synchronize RTC with wall-clock time;
5:      else
6:          if RTC time is synchronized then
7:              s ← getLightSensorData();
8:              t ← getRTCtime();
9:              Timestamp the sensor data;
10:             if s >= L_th1 then
11:                 if Sleep timer is not initialized or expired then
12:                     V_Ctb = getVoltageOfC_tb;
13:                 end if
14:                 if V_Ctb >= V_th1 then
15:                     V_Crtc = getVoltageOfC_rtc;
16:                     if V_Crtc <= V_th2 then
17:                         Enqueue the sensor data;
18:                         Redirect charge from C_tb to C_rtc;
19:                     else
20:                         Construct a packet;
21:                         Transmit the packet;
```

Algorithm 1 Sense and Transmit Evaluation Application (continued)

```
22:                 end if
23:             else
24:                 Enter extended sleep mode;
25:             end if
26:         else
27:             Enqueue the sensor data;
28:             Enter extended sleep mode;
29:         end if
30:     end if
31:     Enter normal sleep mode;
32: end if
33: end while
```

RTC is synchronized using the USB-to-serial interface. In the meantime, the storage capacitors of the EHB also get fully charged. Thus, SENERGY avoids the initial charging overhead of the capacitors. Once deployed, first, software flags associated with the time synchronization are checked to verify whether the RTC time has been synchronized with the wall-clock time. As soon as this is confirmed, the application samples the sensor data. However, before transmitting the measured sensor data, the TB compares the measured light intensity with a predefined threshold light intensity to check if the light intensity (*i.e.*, the output power of the sensor) is sufficient enough to continuously power the RTC. In order not to lose the time-sync, the algorithm makes a conservative decision and puts the system into a sleep mode after queuing the sensor data in a circular buffer. To be specific, SENERGY supports two different sleep modes, namely, extended sleep mode (ESM) and normal sleep mode (NSM). The ESM allows extended time for the EHB to harvest sufficient energy in case of constrained light intensity whereas the NSM just reduces the power consumption between successive transmissions. Hence, if the measured light intensity is below

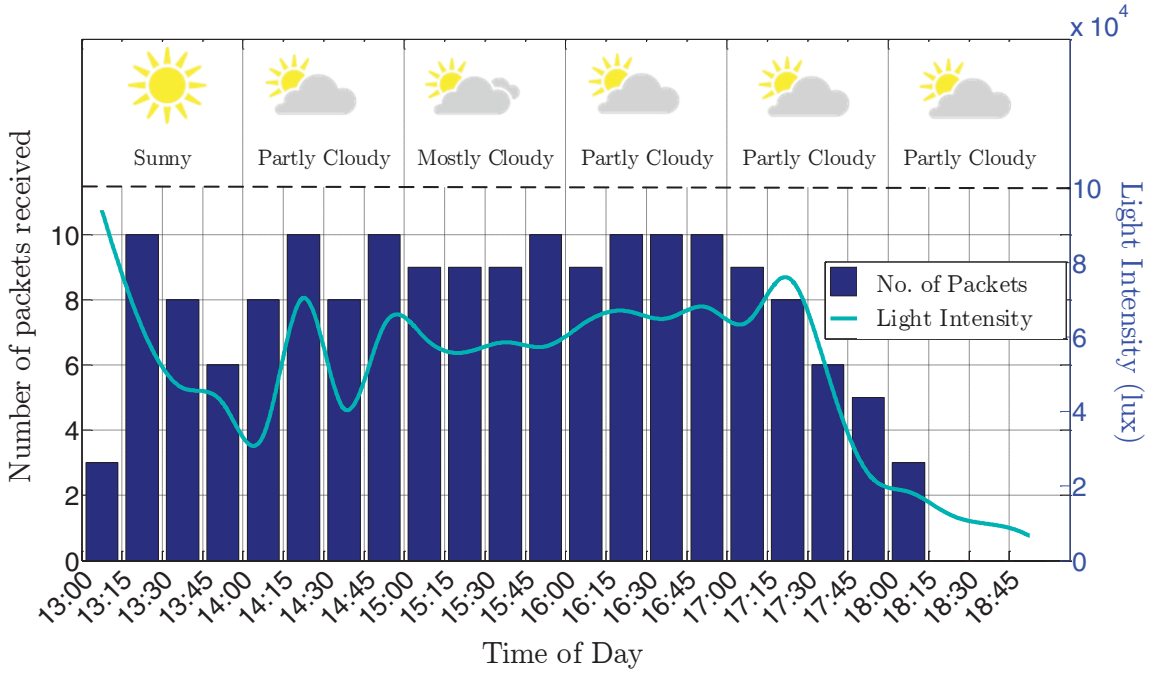


Figure 3.12. Number of successful sensor data transmissions vs. time of day

L_{th1}, the system enters the ESM. As a next step, the energy remaining in C_{rtc} is measured. In the event that the remaining energy is less than the energy required for maintaining RTC time, the energy available in C_{tb} is redirected to C_{rtc} using the CRS. Otherwise, SENERGY transmits a packet with the timestamped sensor data before entering NSM.

The experimental setup and test conditions are described below. The output of the CPs are tied up and supplied to the 330 µF capacitor (C_{tb}). The RTC IC on the TB is configured to be powered by a 30 mF super-capacitor (C_{rtc})[12]. The super-capacitor is fully charged before deployment using USB port power when the TB is connected to a PC via USB for time synchronization. The TB can be powered whenever the voltage of the super-capacitor attains 2.66 V, which is the predefined threshold voltage level of the SVS in the PMU. A base station is placed at a distance of 10 m to receive packets from SENERGY platform. The experiment was deliberately conducted in an outdoor environment during a day with varying weather conditions.

[12] PAS311HR-VA6R from Taiyo Yuden

On the day of experiment, the weather condition was radically changing, and it started by being sunny with clear sky and transitioned to a cloudy overcast weather as the day progressed. Moreover, in the course of the experiment, there was light rain intermittently.

Figure 3.12 shows the light intensity, and the number of packets received at the base station for every 15 minutes. In addition to that, the weather condition for every hour is depicted above the graph (taken from The Weather Channel[13]). As shown in the figure, the number of packets received depends on the light intensity, except at the beginning of the experiment. This is because the 330 µF capacitor is initially discharged and therefore, energy is spent in charging it up. During the experiment, a total of 188 packets were received and the average interval between each packet was 100 seconds. Therefore, the SENERGY platform can be used for any light intensity sensing application that has sensing interval longer than the average charging time. Considering the fact that such applications are heavily duty cycled and that outdoor light intensity changes relatively slowly, the minimum interval that SENERGY supports can easily meet the operation requirements of most applications.

3.5.2 Perpetually powered sub-system

A RTC is a critical component in embedded systems, which enables correlation of its internal activities with the external world. Other than time keeping, RTC performs several other important functions such as synchronization, alarms, and periodic interrupts. Therefore, it is imperative that the RTC sub-system has an uninterrupted power supply. The RTC sub-system can perpetually operate if the output power of the CP meets its power requirements. Due to the ultra-low power consumption of a typical RTC IC, the EHB is capable of supplying the required power even in challenging environments. For instance, the RTC IC used in this work can operate down

[13] The experiment was conducted on Apr 24, 2014 in West Lafayette, IN, USA

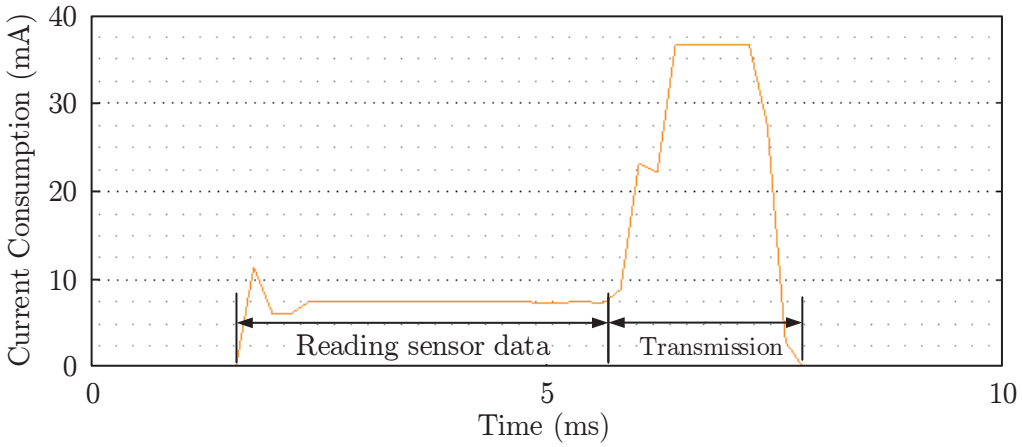


Figure 3.13. Power consumption of the SENERGY target board power consumption for sensing and transmitting a single data packet

to 1.1 V with current consumption of 100 nA. We experimentally confirmed the perpetual operation of RTC.

In the preceding experiment, we confirmed that the SENERGY platform can transmit a total of 188 packets per 315 min under mostly cloudy weather condition. As shown in Figure 3.13, the TB consumes 0.02 µA h of energy at 2.66 V for 6.25 ms. Given the fact that each sense and transmit operation consumes 0.02 µA h, we compute the total energy harvested by the EHB to be 3.76 µA h. This is equivalent to the amount of energy required to operate the RTC IC for 37.6 hours. The analysis concludes that the amount of the harvested energy is more than enough to operate the RTC IC overnight until the next sun rise.

3.6 Related Work

3.6.1 Self-powered systems

Self-powered sensor systems, which rely on some form of energy harvesting or energy scavenging, are widely prevalent [34–38]. Often, such systems are constructed with a dedicated energy harvesting component (like a solar cell, piezoelectric element,

thermoelectric generator, etc.), and an independent sensor. The primary focus of research in such systems has been to optimize the energy harvesting circuitry [39,40] or to optimize the sensor itself [41]. For example, Tsui et al. [42] fabricate and demonstrate a computational module with a dedicated energy harvesting component to power the system with input voltages as low as 190mV. The sensing circuit, in this case, is independent of the energy harvester. In addition, a thorough analysis and discussion of the various trade-offs present in such a design are not discussed. Comparatively, SENERGY proposes to scavenge the energy output from the sensor during idle time and use it as an energy source for the system during active state. In [43,44], self-powered sensor systems are described wherein the sensor output is multiplexed as a power source. Pan et al. [43] simulate a system powered intermittently by low level vibrations of a piezoelectric element that is also used as a sensor. However, the simulations do not account for the RC losses involved in a real deployment and tends to an ideal case. On the other hand, [44] demonstrates a self-powered inertia sensor but does not analyze a system where the sensor is used as the power source. Our work demonstrates SENERGY, a wireless embedded system that is integrated with a photodiode, which functions as a light sensor in active state, and as a power source in idle state. We quantify our results and discuss the trade-offs involved in designing such a perpetual system.

3.6.2 Charge pump architectures

In energy harvesters, utilizing charge pumps for boosting the input voltage is a well-known technique. There has been sufficient interest in the research community for optimizing the efficiency of charge pumps [45–47]. Charge pump architectures can be broadly classified into Dickson, Fibonacci, and exponential. In a Dickson charge pump [48], the voltage gets boosted linearly with each successive stage. In contrast, the Fibonacci and exponential architectures are non-linear architectures. Voltage gets incremented as a Fibonacci sequence across stages in the Fibonacci architecture [49],

while voltage is boosted exponentially with each subsequent stage in an exponential charge pump architecture [47, 50]. SENERGY borrows concepts from all the three architectures but is most closest to the exponential architecture. A Dickson charge pump multiplies the voltage by transferring charge across symmetric stages separated by diodes [48, 51, 52]. Diodes are implemented to isolate each stage, and to prevent any back-flow of charge. Implementing diodes minimizes the control logic required for the architecture. However, the associated voltage drop with diodes decreases the multiplication efficiency of the charge pump. For example, in a charge pump architecture that employs MOSFET-based diodes, the diode drop can be attributed to the V_{th} of the FET. On the other hand, charge transfer switches (CTS) use MOSFETs that are statically controlled by the output voltages of forward stages for toggling the MOSFET switch ON and OFF [53]. By implementing a feedback control, the gate-source voltage of the MOSFET is increased, and therefore the lower voltage output of a stage can be equal to the highest voltage at the input. Wu et al. [54] improves the efficiency of the CTS-based architecture by introducing dynamic feedback control from succeeding stages. Our charge pump architecture in SENERGY utilizes diodes as well as dynamic CTS to optimize the voltage efficiency and minimize the control circuitry power dissipation.

Common charge pump architectures employ two non-overlapping clocks for operation. Therefore, two charge-pumps could be operated in parallel with inverted control logic and this ensures a continuous charge transfer to the output stage capacitor [55]. SENERGY also follows the same principle to maximize the energy harvesting time. Conventional exponential charge pump circuits are symmetric in nature [47]. The architecture is designed such that a fundamental block is repeated in each stage. SENERGY, even though it is an exponential charge pump in nature, has a custom charge-pump architecture utilizing sub-threshold MOS-characteristics to achieve the desired output. The charge pump architecture consists of multiple distinct fundamental blocks, which make up the different stages of the architecture. Our approach

adopts the strengths of all the previous architectures to optimize for conversion efficiency in the presence of non-idealities.

3.7 Summary

We have presented a novel embedded systems architecture that utilizes a sensor as both a sensing element and a power source. We also demonstrated the entire design flow that spans from energy harvesting to utilization of the harvested energy. As a proof of concept, we designed and implemented the SENERGY platform, a batteryless energy-neutral wireless sensing platform that is powered by a photodiode sensor.

4. TELEPROBE: ZERO-POWER CONTACTLESS PROBING FOR IMPLANTABLE MEDICAL DEVICES

4.1 Introduction

The development of implantable medical devices (IMDs), such as cardiac implants, neuro-stimulators, *etc.*, has revolutionized the monitoring, diagnosis, and treatment of a wide range of medical conditions. Given their direct impact on human safety, the need for *reliable* operation is a fundamental, non-negotiable requirement in IMDs. The challenge of ensuring reliable operation is exacerbated by the fact that IMDs are physically inaccessible after implantation, which severely limits post-deployment visibility into system operation. To address this issue, IMDs are increasingly being equipped with wireless connectivity to enable health care professionals to non-intrusively monitor a patient's health and device status. Although wireless connectivity in IMDs enables convenient and timely access to medical data and device status, it incurs a significant power overhead that translates into diminished battery life (*i.e.,* more frequent battery replacement, which usually involves surgery). Therefore, run time monitoring of an IMD for extended durations over a wireless channel is, simply put, an impractical solution.

Test instruments such as oscilloscopes (which allow the direct probing and monitoring of electrical signals/nodes of interest in a hardware platform) have long proven to be an invaluable tool in the arsenal of troubleshooting/debugging aids available to embedded system designers. Inspired by their widespread utility, this work attempts to answer the following question: *"Is it possible to design a remote, contactless probing mechanism for IMDs that enables oscilloscope-like monitoring of signals, but imposes only a minimal (or possibly even zero) power overhead on the IMD itself?"* We answer this question in the affirmative and demonstrate the first system (to the best of our

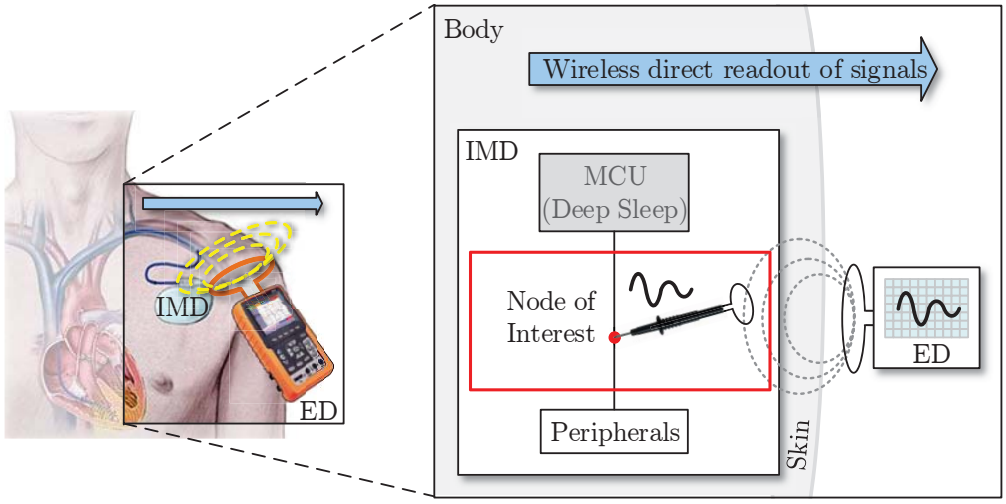


Figure 4.1. Conceptual overview of the TelePROBE system

knowledge) that enables continuous and direct wireless readout of analog and digital electrical signals from an IMD with *zero-power consumption on the IMD side*. The conceptual diagram of the probing mechanism we propose is depicted in Figure 4.1. Inside the IMD is a *passive* probe-like part attached to the node of interest (NOI) that needs to be monitored. An external device (ED) wirelessly reads the signal from the probe in real time, without the need for the microcontroller (MCU) or radio transceiver to be involved in the readout process. Furthermore, in addition to the passive probing, the proposed mechanism can also be used for ultra-low power wireless data transmission from the IMD to the ED in an *active* manner. Specifically, we make the following contributions:

- We propose TelePROBE, an in-situ remote measurement system for IMDs, which achieves oscilloscope-like functionality by allowing direct readout of analog/digital signals wirelessly. TelePROBE is based on the basic technique of inductively-coupled LC readout. We detail the operating principle behind TelePROBE using an analytical model and perform simulation studies to demonstrate the basic technique in action.

- Based on the LC readout technique, we construct a *continuous* monitoring solution for any arbitrary analog/digital signal in the IMD. The most important feature of TeleProbe is that it requires no power from the IMD. The combination of direct readout and inductive coupling enables TeleProbe to completely eliminate all the power overheads associated with conventional radio-based wireless monitoring.

- The active use of the proposed system also enables ultra-low power wireless data transmission from the IMD to the ED. The energy required for the communication is 95 fJ/bit for the on-board probe as a physical layer and 114 nJ/bit for the entire system where the processing core and the communication peripheral in a MCU are active.

- We have designed and implemented prototypes of both an IMD and an ED as embodiments of the proposed concept. We use the prototypes to evaluate the performance of TeleProbe and demonstrate its utility and capability in the context of three usage scenarios, namely i) monitoring of the power state of IMDs for inference of device behavior, ii) continuous monitoring of traffic on an internal I^2C digital communication bus, and iii) ultra-low power active wireless data transmission from an IMD to an ED.

4.2 Related Work

In this section, we present a brief overview of prior work that offloads the communication overhead in wireless links from IMDs to EDs.

4.2.1 LC readout

LC (inductor-capacitor) readout is a wireless measurement technique for a capacitive or an inductive sensor at a remote location. The fundamental principles are identical for both capacitive and inductive sensors, hence we describe the principles

with capacitive ones. The sensors typically do not require a local power source to function. Rather, the sensors autonomously convert a physical quantity of interest into a capacitance value. The sensors form a LC tank circuit with a complementary inductor of fixed value so that the sensor output changes the resonant frequency of the LC tank circuit. An ED with another inductor (primary coil) is inductively coupled to the inductor (secondary coil) of the LC tank circuit. The ED sweeps over the certain frequency range where the resonant frequency is possibly located in. From the resonant frequency found, the ED derives the current capacitance of the sensor and, in turn, the physical quantity. The signal processing involved in finding the resonant frequency is resource-demanding and computationally-intensive. However, the LC readout system offloads such demands to EDs and maintains the sensor circuitry as simple as possible. Nopper *at el.* [56, 57] introduced various types of LC sensor applications and readout techniques. The LC sensors are preferable for IMDs where the absence of an on-board power source facilitates miniaturization. Chen *at el.* [58] reported a 4-mm planner coil-based LC sensor implant for glaucoma patients' intraocular pressure sensing, and Fonseca *at el.* [59] designed a flexible LC sensor for abdominal aortic aneurysms pressure measurement. However, the LC readout technique is limited to special types of sensors and not applicable to system-level monitoring.

4.2.2 Backscattering

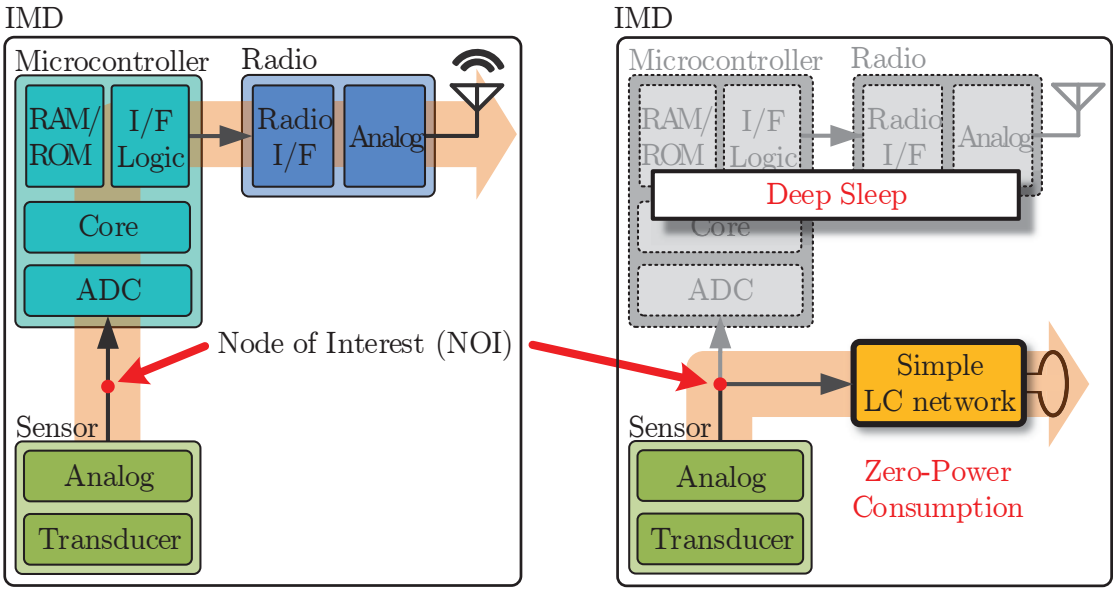
Backscattering is a digital communication technique that reflects power with a modulation to encode data. Backscatter communication is initiated by a reader radiating power to a remote device. The remote device reflects a portion of the received power back to the reader using various types of modulation techniques [60]. The reader decodes the data from the reflected power. The remote device can be substantially miniaturized because reflecting power can be done with a very simple circuitry compared to active transmitters. Often the remote device is operated using the re-

ceived power at the expense of added hardware components such as rectifiers and voltage regulators. As compared to the traditional communication techniques, the backscattering technique significantly reduces the power overhead for the remote device. For instance, Besnoff *at el.* [61] reported a near field backscatter supporting 16.4 pJ/bit at 30 Mbps for in-vivo biotelemetry, and Thomas *at el.* [62] introduced far field backscatter with 16-QAM achieving 15.5 pJ/bit at 96 Mbps.

Although the backscattering is a general communication mechanism that dramatically reduces the power consumption of the communication link itself, the power consumed by a controller (*e.g.*, MCU) that governs the link is often neglected in spite of its significant contribution to the system power consumption. Backscatter communication inevitably requires the controller to stay active for data acquisition and processing, which is never free in terms of power. In fact, for instance, the pico-joule level energy consumption reported in [61] is of the radio-related components only, such as the RF switch, which is to replace conventional radio circuit. In contrast, TeleProbe keeps the controller in a low power mode during signal monitoring. Furthermore, TeleProbe also supports an active data transmission that achieves several orders of magnitude smaller power consumption compared to the aforementioned backscatters.

4.3 TeleProbe Circuit and System

In this section, we first discuss the design principles and main features of TeleProbe to enable remote, conactless probing with zero power consumption. Next, we derive an analytical model and perform simulation studies to describe the TeleProbe circuit. Finally, the operating principle of the TeleProbe system is explained.



(a) Conventional wireless monitoring (b) Proposed wireless direct readout

Figure 4.2. Advantage of the proposed system

4.3.1 TelePROBE overview

The architectural difference between TelePROBE and conventional wireless monitoring system is depicted in Figure 4.2 using an example of analog sensor reading. Reading sensor output is the most elementary in-situ monitoring of IMDs. However, even the simple task involves multiple steps for data processing and transmission, as illustrated in Figure 4.1(a). In order to read an analog signal from the sensor, *i.e.*, the voltage of the NOI, the MCU needs to convert the analog voltage into a digital value, prepare packets containing the data, and transmit the packets through the radio. The MCU and the radio must actively engage in this process and dissipate a significant amount of power, which make prolonged monitoring prohibitive.

We accomplish zero-power measurement by eliminating the needs for data conversion, processing, and transmission. Figure 4.1(b) illustrates the basic concept of TelePROBE that we propose. TelePROBE achieves direct wireless readout of the NOI voltage (V_{noi}) using a simple LC tank circuit. The LC tank is an intermediary

that automatically converts V_{noi} to the resonant frequency, which is remotely measurable by the ED through inductive coupling. Specifically, the LC tank circuit employs a *varactor* for the conversion. A varactor is a passive device that exhibits a varying capacitance depending on the voltage applied across it, which is called the *tuning voltage*. We use V_{noi} as the tuning voltage in order for the ED to back-translate V_{noi} from the resonant frequency measured. Neither the MCU nor the radio is engaged in this readout process.

This varactor-based LC readout technique features the following advantages desirable for severely energy- and size-constrained IMDs:

- **Zero-power consumption.** The LC tank circuit in the IMD does not actively consume power to function, and the power overhead is fully imposed on the ED, which is relatively less energy-constrained. Although the presence of the LC tank circuit introduces a leakage path, however, the minuscule leakage current, typically ranging in the picoampere (pA) level, is considered negligible compared to the system-level power consumption of a typical IMD and generally assumed to be zero. Furthermore, the direct readout truly bypasses other system components of the IMD (*e.g.*, MCU, radio) and eliminates the associated power dissipation.

- **Small footprint.** The LC tank circuit requires only a few discrete components (two components at a minimum, *i.e.*, a varactor and an inductor of a fixed value).

- **Low impact on signal integrity.** The diode characteristics of a varactor prevent excessive leakage. Also, loading effects caused by the varactor capacitance can be effectively suppressed using an isolation resistor between the NOI and the LC tank circuit.

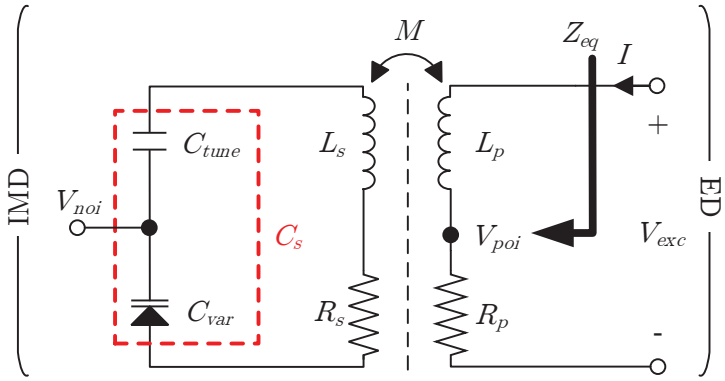


Figure 4.3. Fundamental circuit model of TelePROBE

4.3.2 TelePROBE circuit

The TelePROBE circuit is composed of the LC tank circuit in the IMD and the reader circuit in the ED, which are inductively coupled to each other. We first explain the fundamental principle of the inductively coupled circuits using an analytical model and SPICE simulation, then, describe how the varactor-based LC tank circuit is configured for the readout of IMD signals.

Fundamental circuit model

Figure 4.3 shows the schematic view of a pair of inductively coupled circuits. The left-hand side is the secondary circuit to be implemented in the IMD, and the right-hand side is the primary circuit to be implemented in the ED. The equivalent capacitance of the secondary circuit, C_s, is the combination of the variable varactor capacitance C_{var} and the constant base capacitance C_{base}. While the resonant frequency varies with C_{var}, the base of the variation is determined by C_{base}. Once an inductive channel is established, the equivalent impedance Z_{eq} viewed from the ED can be derived using circuit analysis as,

$$Z_{eq} = R_P + j2\pi f L_P \left(1 + \frac{k^2 \left(\frac{f}{f_0}\right)^2}{1 + j\frac{1}{Q}\frac{f}{f_0} - \left(\frac{f}{f_0}\right)^2} \right), \tag{4.1}$$

where L_p is the inductance of the primary coil, R_p is the resistance of the resistor in series with the L_p, f is the excitation frequency, f_0 is the resonant frequency of the secondary circuit, $Q = {R_S}^{-1}(L_S {C_S}^{-1})^{1/2}$ is the quality factor of the secondary circuit, and k is the coupling coefficient [56, 57]. Calculating the impedance real part from (4.1) leads to

$$\mathrm{Re}\{Z_{eq}\} = R_P + 2\pi f L_P k^2 Q \left(\frac{\frac{f}{f_0}}{1 + Q^2 \left(\frac{f}{f_0} - \frac{f_0}{f} \right)^2} \right). \tag{4.2}$$

If the Q is large compared to unity ($Q \gg 1$), $\mathrm{Re}\{Z_{eq}\}$ is maximized at the resonant frequency of the secondary circuit, *i.e.*,

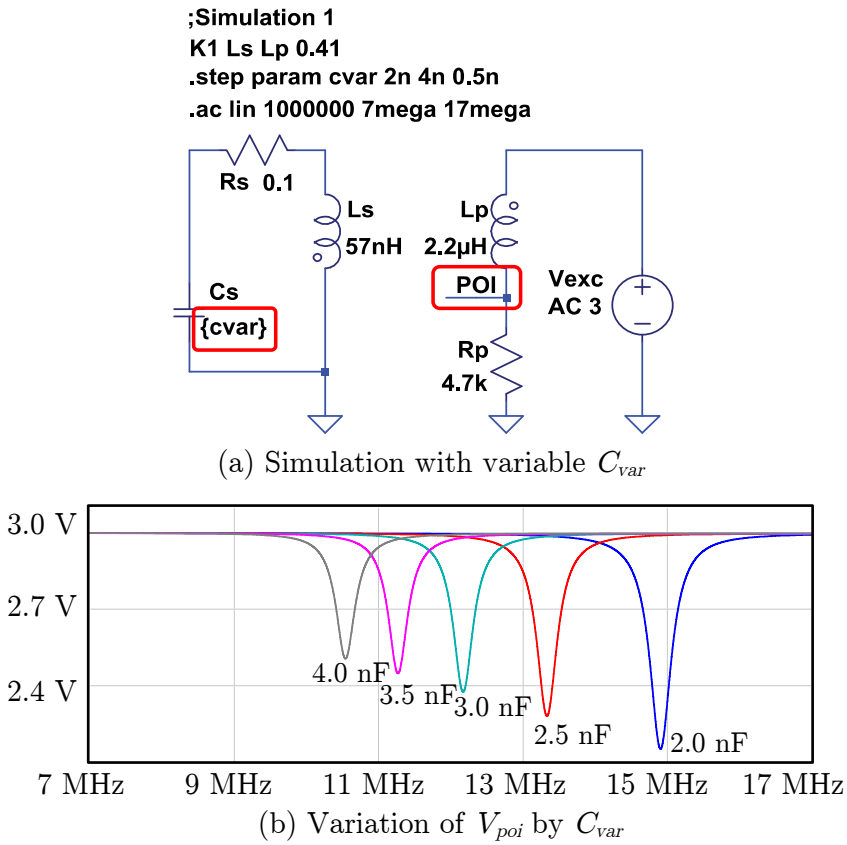
$$R_{max} \approx \left. \mathrm{Re}\{Z_{eq}\} \right|_{f=f_0} = R_p + 2\pi f_0 L_P k^2 Q. \tag{4.3}$$

Therefore, the resonant frequency, which is monotonically varying according to V_{noi}, can be obtained from the magnitude frequency response. For example, if excitation voltages (V_{exc}) of constant peak-to-peak amplitude and variable frequency are applied to the primary circuit, the R_p and L_p in series connection function as a frequency-dependent voltage divider, and the voltage of the measurement point (V_{poi}) gives its minimum at the resonant frequency of the secondary circuit.

SPICE simulation

Parametric sweep simulations of the fundamental circuit have been performed using LTspice IV for variable capacitance C_{var} and coupling coefficient k, respectively. The simulation setup and results are shown in Figures 4.4 for the varying C_{var}, and in Figures 4.5 for the varying k, respectively. As shown in Figure 4.4(b), the magnitude frequency response of V_{poi} exhibits a unique dip at $f = f_0$. Therefore, by virtue of the one-to-one mapping between V_{noi} and the varactor capacitance, the ED can find V_{noi}, which caused the capacitance to change.

For accurate and precise measurements, C_{var} should be a sole parameter affecting the V_{poi} measurement. However, the coupling coefficient k may have a minor impact

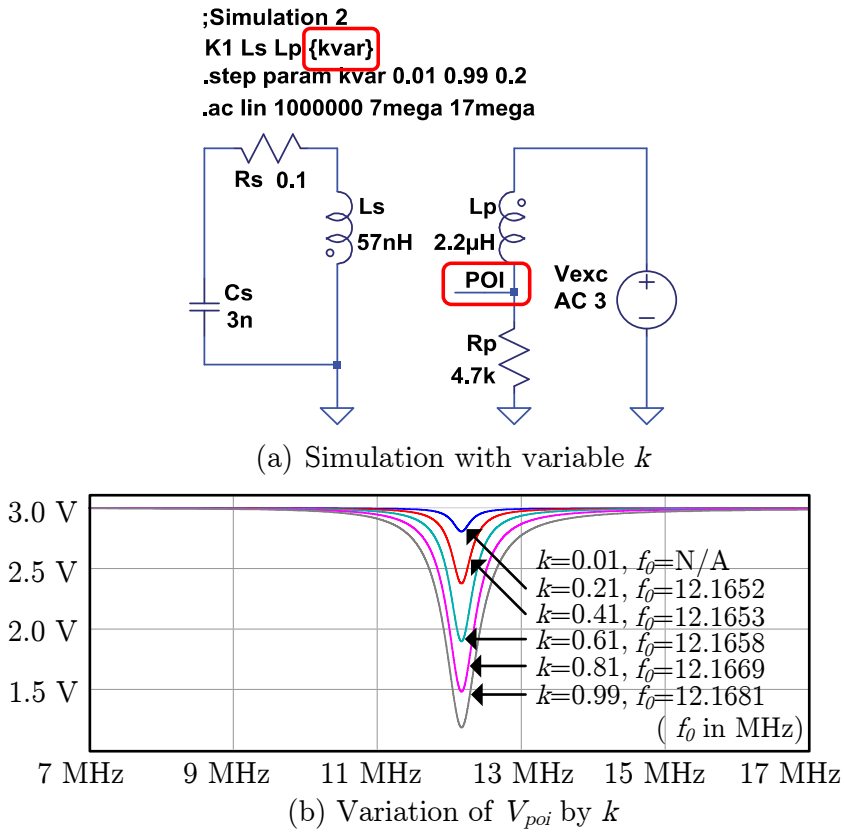


(a) Simulation with variable C_{var}

(b) Variation of V_{poi} by C_{var}

Figure 4.4. LTspice parametric sweep simulation results for variable capacitance

as well. The coupling coefficient is mostly affected by the geometry of the coils, and the relative location of the IMD with respect to the ED cannot be precisely determined after implantation. The simulation results demonstrate the negligible impact of k on f_0. As shown in Figure 4.5(b), the variation of k only affects the magnitude of minimum V_{poi}, and the locations of the minimum V_{poi} (*i.e.*, resonant frequency) are nearly invariant of k. For $0.01 \leq k \leq 0.99$, the RMSE of f_0 found from the simulation as compared with its theoretical value is only 4.751 kHz, which corresponds to ± 2.34 pF for $C_s = 3$ nF ($\pm 0.078\%$). The error can be kept at a negligible level by having high Q compared to unity or can be accurately calculated by (4.2) with the Q value measured using the full-width-at-half-maximum (FWHM) method [56].



(a) Simulation with variable k

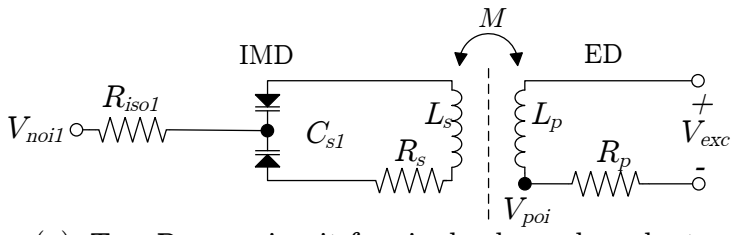
(b) Variation of V_{poi} by k

Figure 4.5. LTspice parametric sweep simulation results for variable coupling coefficient

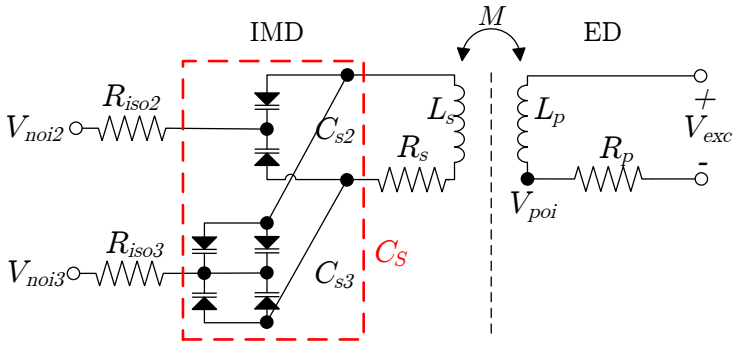
Another thing to note is the small inductance of the coil for the IMD. As seen in (4.2) and (4.3), R_{max} is a function of L_p only, hence L_s can be kept small. This property is well suited for IMDs that require miniaturized implementation. In this simulation, L_s is set to 57 nH, which is the inductance of the 4-mm disk coil used for the intraocular pressure sensor in [58].

TeleProbe LC tank circuit design

In practice, adding C_{base} and C_{var} introduces an additional load capacitance to the NOI. Although this additional capacitance is not significant, we further suppress potential signal distortion by i) employing an isolation resistor R_{iso} between the NOI



(a) TELEPROBE circuit for single-channel readout



(b) TELEPROBE circuit for dual-channel readout

Figure 4.6. TELEPROBE LC tank circuits

and the varactor, and ii) replacing the base capacitor with another varactor for back-to-back configuration. The modified LC tank circuit for readout of a single-channel analog/digital signal is presented in Figure 4.6(a).

TELEPROBE also supports multi-channel monitoring of digital signals. Instead of using a single pair of back-to-back varactors, we compose a varactor network so that its equivalent capacitance is unique dependent on the combination of multiple bits. As long as each bit combination is represented as a unique C_{var}, it is possible to monitor multiple NOIs. Figure 4.6(b) shows a simple example of a varactor network for dual-channel digital signals readout. In this example, all the individual varactors are of the same capacitance range, hence the range of the equivalent capacitance of NOI3, C_{s3}, is double that of NOI2, C_{s2}. Table 4.1 shows the total equivalent capacitance $C_s = C_{s2}||C_{s3}$ of the LC tank circuit. Logic low (L) and logic high (H) correspond to the maximum and minimum varactor capacitance, respectively. As seen in the table, C_s uniquely represents four electrical states of NOI2 and NOI3,

Table 4.1. Equivalent capacitance of the varactor network of Figure 4.6(b) to represent the four electrical states of two digital signals

V_{noi2}	C_{s2}	V_{noi3}	C_{s3}	$C_s = C_{s2} \| \| C_{s3}$
L	$\max(C_{s2})$	L	$2\max(C_{s2})$	$3\max(C_{s2})$
L	$\max(C_{s2})$	H	$2\min(C_{s2})$	$2\min(C_{s2}) + \max(C_{s2})$
H	$\min(C_{s2})$	L	$2\max(C_{s2})$	$\min(C_{s2}) + 2\max(C_{s2})$
H	$\min(C_{s2})$	H	$2\min(C_{s2})$	$3\min(C_{s2})$

hence the ED can simultaneously determine the values of these two bits from a single measurement of resonant frequency. This is not limited to the dual-channel readout, but generally scalable for multi-channel readout as long as we can compose a varactor network where multiple values of C_{var} are clearly distinguishable from each other.

Both of the single-channel analog signal readout and the dual-channel digital signals readout circuits are implemented as part of working prototypes and evaluated with practical usage scenarios in Section 4.6.

4.3.3 TeleProbe system

For continuous signal monitoring, TeleProbe repeats searching for the resonant frequency of the LC tank circuit and mapping it to the voltage level of the NOI. By constructing a time-series of V_{noi}, we obtain an oscilloscope-like view of the signal. This process is illustrated in Figure 4.7. In this example, we continuously measure the variation of V_{noi} shown in Figure 4.7(a). The resonant frequency of the LC tank circuit in the IMD continuously changes depending on V_{noi}. The ED finds the minimum V_{poi} that corresponds to the resonant frequency through frequency sweeping, as shown in Figure 4.7(b). Hence, the time taken for sweeping the frequency range, T_{sweep}, is the time interval between two consecutive samples of V_{noi}. This interval is to be minimized in order to capture the transient variation of V_{noi}, but at the same time,

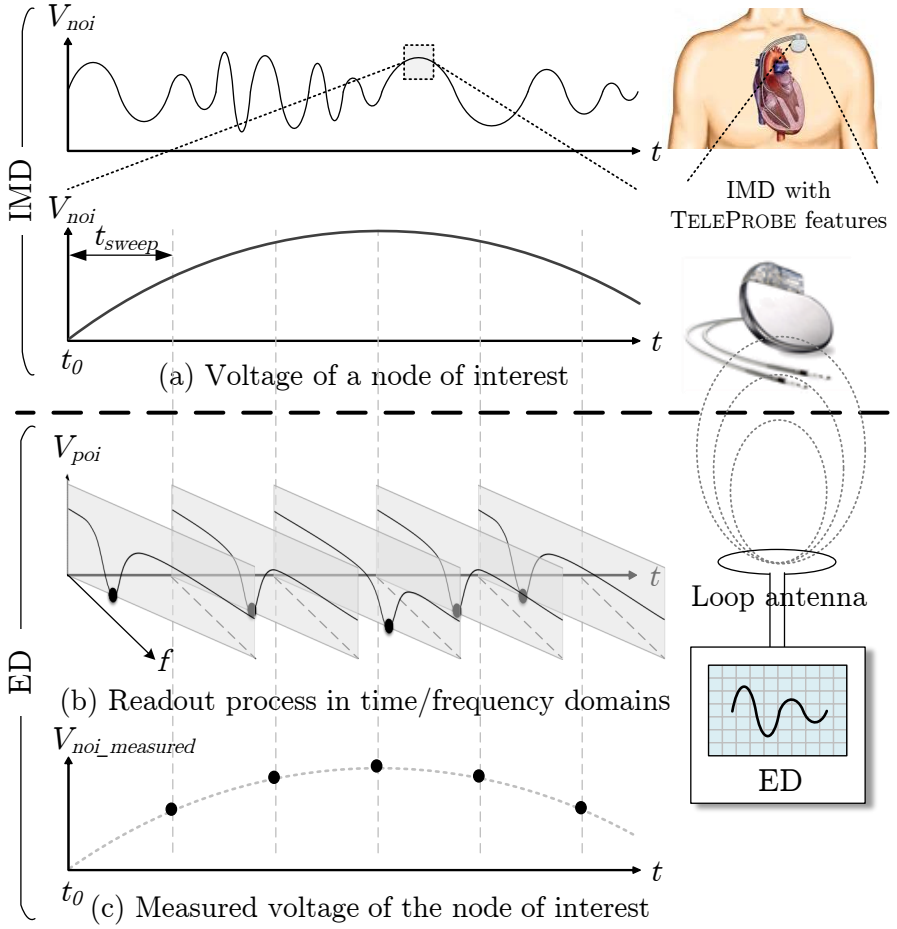


Figure 4.7. Illustration of the TelePROBE system operation

should be long enough to allow precise scanning for the resonant frequency. Finally, from the measured time-series resonant frequency, we construct the time-series V_{noi}, as shown in Figure 4.7(c).

The readout process is independent of how V_{noi} is resulted. To be specific, regardless of a passive or an active method resulting in V_{noi}, the process constructs a live view of V_{noi} on the ED as it is. Therefore, in addition to the passive monitoring of V_{noi}, the IMD is able to transmit data by explicitly controlling V_{noi}. Then, the same readout process constructs the transmitted data on the ED, achieving an ultra-low power unidirectional communication from the IMD to the ED.

4.4 Performance Metrics

TeleProbe is comparable to an oscilloscope, so it can be evaluated by similar performance metrics. Table 4.2 lists the performance evaluation metrics and major influential factors for both an oscilloscope and TeleProbe. In this section, we discuss these metrics and design efforts to improve these in TeleProbe.

4.4.1 Accuracy and precision

In an oscilloscope, the accuracy and precision of measurement largely depend on the performance of the ADC and the input amplifiers. On the other hand, in TeleProbe, these factors are mainly governed by the quality factor (Q-factor) of the LC tank circuit and the strength of inductive coupling (coupling coefficient). A high Q-factor improves frequency selectivity and, hence, results in a high signal-to-noise ratio (SNR). We can achieve a high enough Q-factor using off-the-shelf commercial components at a reasonable cost. Also, at a high Q-factor, error induced by coupling coefficient is negligible as demonstrated in Section 4.3.2.

4.4.2 Resolution

The resolution of an oscilloscope is tightly coupled with how precisely the ADC digitizes input voltage to digital codes. In TeleProbe, the resolution is mainly determined by the number of discrete steps in a frequency sweep. In order to achieve an effective resolution of n-bit, there should be at least 2^n steps in a single sweep. The resolution can be improved by using a fine-grained frequency sweep generator.

4.4.3 Sampling rate

Similarly to an oscilloscope that the sampling rate is generally determined by the speed of analog-to-digital signal processing, that of TeleProbe is determined by the time taken for a single sweep. Therefore, a high sampling rate can be achieved

Table 4.2. Performance metrics for oscilloscopes and TeleProbe

Metric	**Oscilloscope**	TeleProbe
Accuracy, Precision	- ADC and gain accuracy and precision	- Coupling coefficient - Quality factor
Resolution	- ADC resolution	- Number of steps in a sweep
Sampling rate	- ADC Speed	- Sweep speed
Impact on NOI	- Loading effect due to a probe contact	- Loading effect due to the LC tank circuit

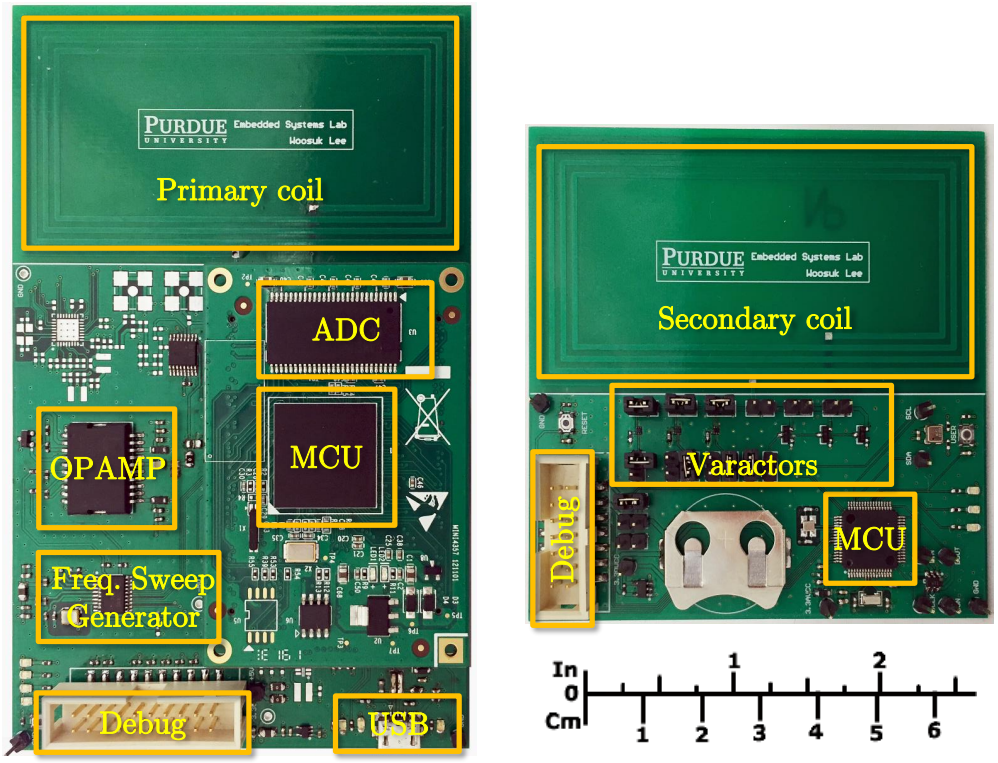
either by reducing the time for each frequency excitation or reducing the number of sweeping steps.

4.4.4 Impact on the signal integrity of NOI

The probing device (*e.g.*, an oscilloscope probe or the TeleProbe LC tank circuit) introduces inevitable adverse impacts on the NOI and may alter the original signal. These impacts should be minimized to ensure high signal integrity and high measurement quality. In TeleProbe, the leakage current is negligibly small owing to the diode characteristics of a varactor, and the isolation resistor effectively suppresses the loading effects, preserving the signal integrity of the NOI.

4.5 Prototype Implementation

We designed and implemented prototypes of an IMD and an ED as embodiments of the proposed TeleProbe system. The ED prototype is a reader device to wirelessly measure the system behavior of the IMD, which is available through the LC tank circuit. The IMD prototype is an up-scale pressure sensing device that has all the



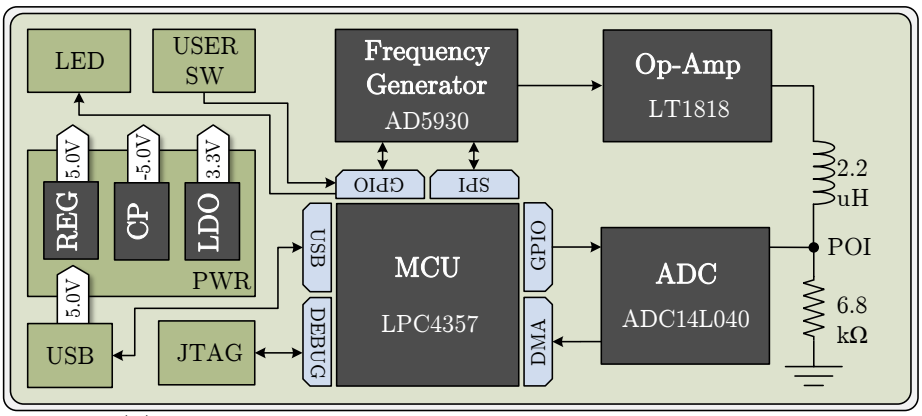
(a) TelePROBE ED prototype (b) TelePROBE IMD prototype

Figure 4.8. Photographs of the TelePROBE prototypes

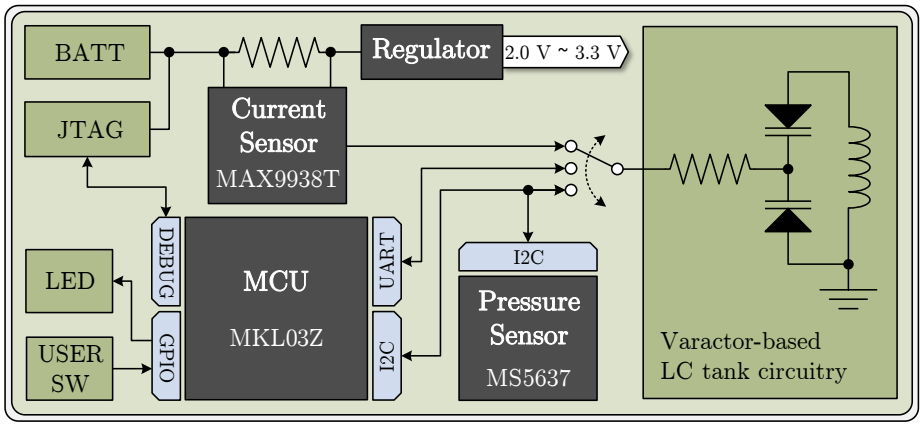
measurement points brought out for the experimental purpose. Figure 4.8 shows the photographs of the TelePROBE evaluation prototypes.

4.5.1 TelePROBE ED prototype

The ED prototype is a low-cost, small form factor network/impedance analyzer designed to find out the resonant frequency of the inductively coupled LC tank circuit. Figure 4.9(a) depicts the block diagram of the TelePROBE ED prototype. The excitation frequency is generated by a frequency sweep generator IC (Analog Devices AD5930) that supports output frequency up to 25 MHz at up to 4,096 levels of granularity, which corresponds to a 12-bit resolution. The excitation signal is amplified using an opamp (Linear Technology LT1818) and used as V_{exc} to drive the primary



(a) Block diagram of the TelePROBE ED prototype



(b) Block diagram of the TelePROBE IMD prototype

Figure 4.9. Block diagrams of the TelePROBE prototypes

circuit as described in Section 4.3.2. For every frequency excitation, the V_{poi} is measured using a 14-bit parallel ADC (Texas Instruments ADC14L040) once the primary circuit attains a steady-state for each excitation. In particular, the primary coil is sized to 2.2 µH and care has been taken to put its self-resonant frequency far beyond the operating frequency range of the ED prototype so as not to interfere measurements. With the resistor of 6.8 kΩ in series connection, the power to the primary circuit is around 15 mW at a maximum.

The ED prototype is designed to be a portable device in terms of power consumption and consumes only 344 mA. The ED prototype is powered by USB port, which is also used to communicate with PC for post-signal processing. To be specific, the

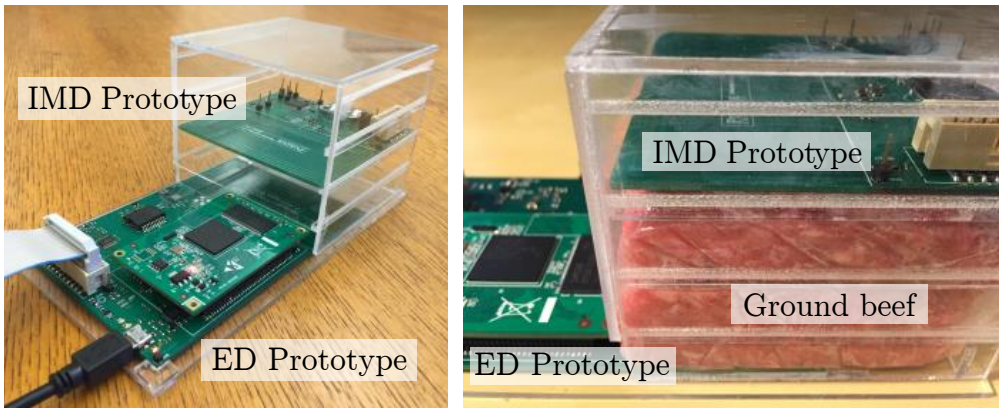


Figure 4.10. TelePROBE prototypes and the experimental setup

magnitude frequency response for V_{poi}, which is measured using the ADC, can either be processed internally or dumped to PC via USB to find out the minimum V_{poi} that corresponds to the resonant frequency of the LC tank circuit.

4.5.2 TelePROBE IMD prototype

The IMD prototype is a battery-operated pressure sensing device that emulates an implantable medical sensor device. Figure 4.9(b) depicts the block diagram of the TelePROBE IMD prototype. The system voltage is configurable for testing purpose. A low-power 32-bit ARM Cortex M0+ MCU (NXP KL03) reads the pressure sensor (Measurement Specialties MS5637) over I^2C bus [63]. The MCU also measures the system-wide current consumption using the high-side current measurement IC (Maxim MAX9938T) through ADC. Inside the varactor-based LC tank circuitry, the same circuit introduced in Figure 4.6 is implemented with NXP BB171 varactors, which comes in a small SOD323 (1.7×1.25×0.8 mm) package. The LC tank circuit is configurable for either single-channel or dual-channel readout using hardware jumpers. For the single-channel readout, the LC tank circuit is exclusively connected to either the output of the current sensor IC or the TX of UART peripheral in the MCU. For the dual-channel readout, each input of the LC tank circuit is respectively connected

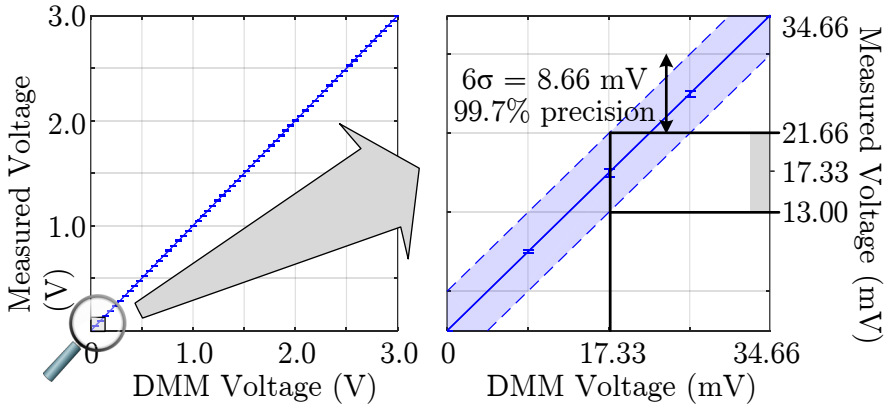


Figure 4.11. Measurement performance of the TelePROBE ED prototype supporting 8.66 mV resolution with 99.7% precision

to the `SDA` and `SCL` signals of the I^2C bus. For design simplicity, the secondary coil in the LC tank circuit has the identical design to the primary coil used in the ED prototype.

4.6 Evaluation

In this section, we first evaluate TelePROBE based on the performance metrics discussed in Section 4.4. We further demonstrate the utility and capability of TelePROBE using the prototypes in the context of three usage scenarios. During the evaluation, distance between the IMD and the ED prototypes were varied while the two coils are perfectly aligned inside a structured acrylic frame. Upon the variation in distance, the gap between the two coils are filled with 85% lean ground beef to emulate human body model [64]. The TelePROBE prototypes and experimental setup are shown in Figure 4.10.

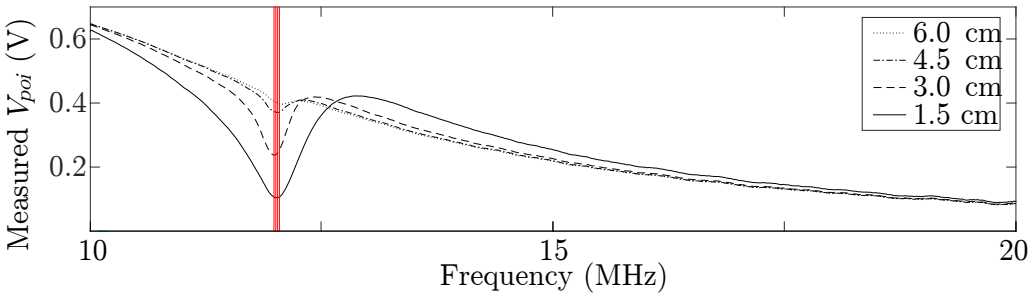


Figure 4.12. Precise reading over distances up to 6 cm

4.6.1 Measurement performance

Accuracy, precision, and resolution

Measuring V_{noi} involves multiple conversions of physical quantities, such as C_{var}, f_0, and V_{poi}, but we are not interested in the accuracy of these conversions because they are only intermediate variables for mapping V_{noi}. Rather, the precision and resolution of end-to-end mappings of V_{noi} is the only concern. Figure 4.11 shows the evaluation results for the precision and resolution of the mapping. For the evaluation, the single channel readout circuit (Figure 4.6(a)) implemented on the IMD prototype is connected to the output of an external 12-bit DAC that generates analog voltage from 0 V to 3 V. The DAC output that is uniformly divided into 64 steps in the 3 V range is measured by both a DMM and the ED prototype, then compared. The maximum standard deviation of 100 measurements for each DAC output voltage is 1.9719, which corresponds to 1.44 mV. Consequently, the effective resolution is 5.77 mV (519 discrete levels) with 95.4% precision ($\pm 2\sigma$) or 8.66 mV (346 discrete levels) with 99.7% precision ($\pm 3\sigma$).

To evaluate the effects of the coupling coefficient, distance between the coils was varied while the coils were perfectly aligned. The maximum distance that yields correct measurements is around 6 cm, and the results with various distances are shown in Figure 4.12. As shown in the figure, the impact of the distance between the IMD and the ED on the measurement is almost negligible.

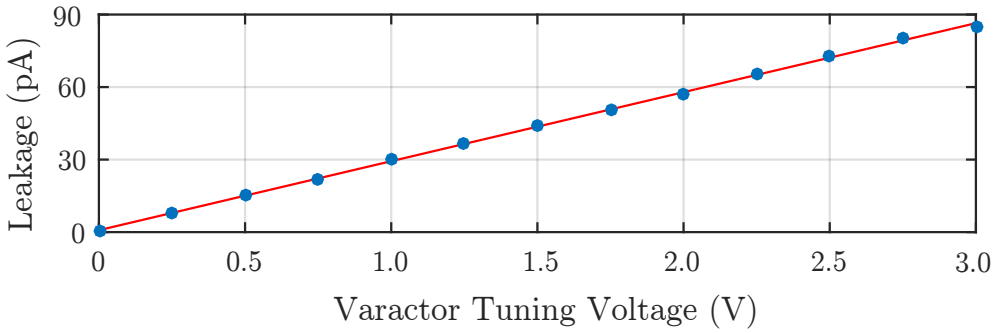


Figure 4.13. Measured varactor leakage current (BB171)

Sampling rate

As discussed in Section 4.4, the resolution and sampling rate are traded off against each other. We designed the ED prototype to support 100 kHz sampling considering the I^2C bus specification. This corresponds to a 10-µs sampling interval. Frequency sweeping, measuring V_{poi}, and mapping to V_{noi} are done within this 10 µs.

Impact on the signal integrity of NOI

The leakage current and the loading effect caused by the LC tank circuit should be as small as possible to ensure the high integrity of original signals. The leakage current of the NXP BB171 varactor used in the IMD prototype was characterized using the Keithley 6430 source meter. As shown in Figure 4.13, the leakage current is linearly proportional to the varactor tuning voltage within 3 V range. The leakage current ranging in the picoampere (pA) level is negligibly small enough not to significantly affect the original signals. Furthermore, we effectively suppress the loading effect using isolation resistor between the NOI and the LC tank circuit. As an example, for the multi-channel readout circuit connected to the I^2C bus on the IMD prototype, the isolation resistors are sized to 8.2 kΩ in order to have the RC time constant of 0.984 µs with the maximum equivalent varactor capacitance of 120 pF.

Table 4.3. Varactor configurations for I^{2}C bus monitoring

SDA level	C_{s2} (pF)	SCL level	C_{s3} (pF)	C_s (pF)	f_0 (MHz)
L	40	L	80	120	9.79
L	40	H	40	80	11.99
H	20	L	80	100	10.73
H	20	H	40	60	13.85

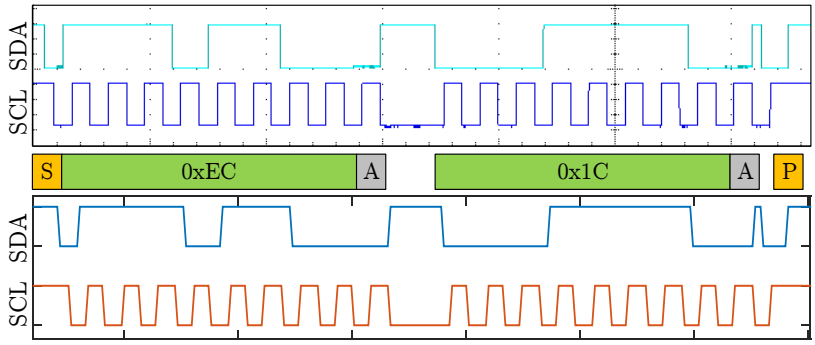
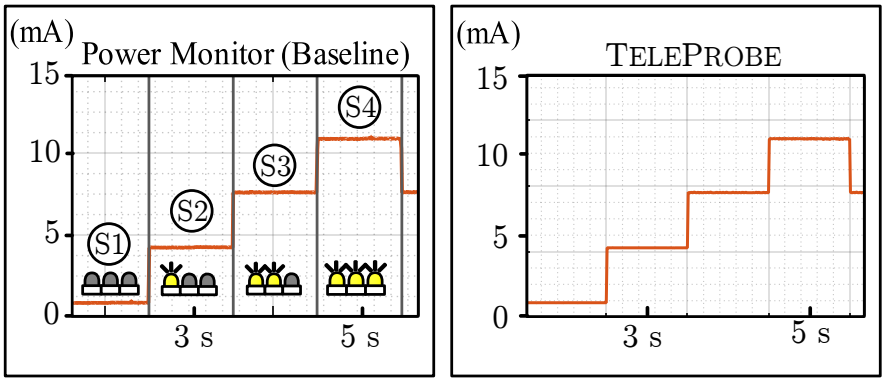


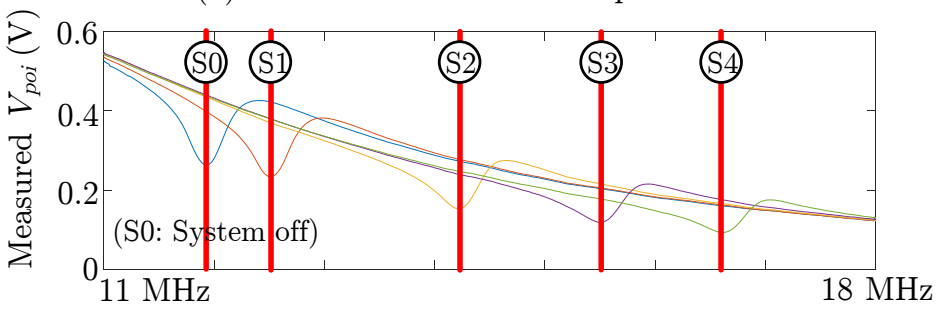
Figure 4.14. Comparison of I^{2}C bus operation monitored with an oscilloscope and TeleProbe

4.6.2 Monitoring of I^{2}C bus

We demonstrate the multi-channel digital signal readout capability of TeleProbe using the I^{2}C bus on the IMD prototype. Table 4.3 states the capacitance values of the varactor configuration for the dual-channel readout circuit introduced in Figure 4.6(b). The four electrical states of SDA and SCL exhibit a unique equivalent capacitance, which is identified by the magnitude frequency response of V_{poi}. For the test, a reset command is sent from the MCU to the pressure sensor over the I^{2}C bus and measured using both an oscilloscope and the ED prototype. The TeleProbe measurements are dumped to PC in real-time and post-processed to a waveform using Matlab. As depicted in Figure 4.14, TeleProbe obtains identical bus monitoring results to the oscilloscope measurement.



(a) Measured current consumption



(b) TELEPROBE measurement results in frequency domain

Figure 4.15. Comparison of real-time power measurement using a commercial power monitor device and TELEPROBE

4.6.3 Behavior validation with power analysis

Monitoring of power consumption enables a useful inference about system operation [65, 66]. Operation of an IMD can be validated by correlating the online power measurement with its pre-characterized power consumption. Using TELEPROBE, we can monitor the power consumption of an IMD in real-time with no power overhead. For evaluation purpose, the IMD prototype is designed to have five power states by toggling one of the three on-board LEDs in every one second, and the states are denoted as S0 through S4. As seen in Figure 4.15, TELEPROBE achieves identical power measurements to a commercial power monitor device (Monsoon Power Monitor).

4.6.4 Active data transmission

The utility of TeleProbe is certainly not limited to the passive monitoring. The active use of the LC tank circuit enables ultra-low power data transmission from the IMD to the ED. Instead of passively monitoring digital signal lines, the active communication explicitly controls the lines. The same mechanism used for the passive monitoring constructs the transmitted data on the ED. We make a practical use case for the active communication using a single digital line that is driven by the TX of UART peripheral in the MCU. The asynchronous nature of the UART protocol is well suited for the single line communication where a synchronization clock is absent. The ED identifies the electrical state of the TX line (logic high/low) by sweeping the corresponding two resonant frequencies and dumps the results to PC. The UART receiver counterpart is implemented as software on PC using 16x oversampling technique. The overall internal processing time of the ED prototype results in the UART baud of 1,200 bps at maximum. The performance of the active data transmission has been evaluated in terms of energy per bit and bit error rate over distance.

Energy per bit

For the concept of an active use of the LC tank circuit as a physical layer of the communication, the energy consumption is calculated based on the measured varactor leakage current shown in Figure 4.13. The leakage current is 57 pA at 2 V, which is the system voltage of the IMD prototype. The leakage path exists through the two varactors in back-to-back configuration and results in the energy per bit of 95 fJ/bit, if an equal chance of the logic state high/low is assumed. This is several orders of magnitude smaller than what had been achieved for the physical layer of the previous backscatter communications, which is generally considered as the most energy efficient communication technique. However, an empirically valid analysis should take into account the associated energy consumption of the controller as well. In that sense, the

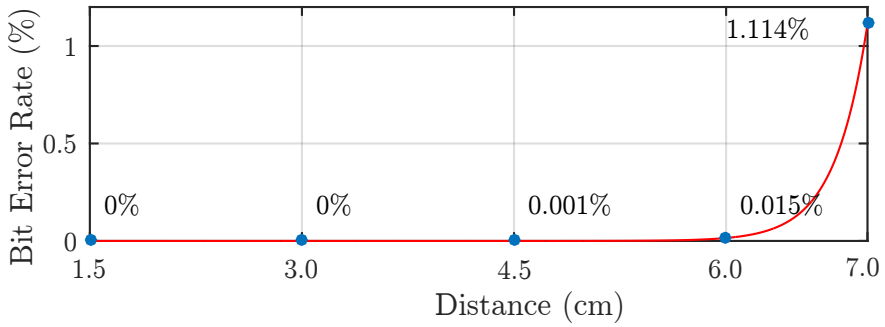


Figure 4.16. Bit error rate over distance for the active wireless data transmission from the IMD to the ED

overall current consumption of the MCU in the IMD prototype is 68.5 µA during active mode with an ongoing UART transmission, resulting in effective energy consumption of 114 nJ/bit. This is comparable to that of a commercial RF SoC. For instance, the energy per bit characteristic of a well-known 2.4 GHz RF transceiver SoC TI CC2530 is 184 nJ/bit when the CPU is idle and the radio is set to the most aggressive power saving settings (*i.e.*, TX power of -28 dBm, operating voltage of 2 V, and the default data rate of 250 Kbps). Moreover, the energy consumption can further be reduced at higher baud since the physical layer's contribution to the system power consumption is considerably insignificant. Note that the 1,200 bps is an upper bound specific to the ED prototype, but the TeleProbe architecture itself does not limit the speed.

Bit error rate

The bit error rate (BER) was empirically characterized with respect to the varying distance between the two coils on the IMD and the ED prototypes. To minimize the impact of the varying distance on the variation of the resonant frequency, the ED adaptively calibrates before initiating the communication. Specifically, at the initial stage, the ED scans over possible frequency range to identify the two resonant frequencies corresponding to the logic states high/low of the TX line. Given the fact that a resonant frequency corresponding to the logic state low cannot be greater than

that of state high, the ED is able to automatically associate the resonant frequencies found with the respective logic states. This feature enables a robust communication regardless of the distance variation. For the test, number of bit flips are measured out of 25,600 bytes transmitted (0 to 255 for 100 times), and the results is shown in Figure 4.16. As shown in the figure, the BER is less than 0.015% up to 6 cm, and it corresponds to 30 bit flips out of 204,800 bit transmitted.

4.7 Summary

We have presented TeleProbe, an in-situ measurement system for IMDs, which enables wireless direct readout of electrical signals with zero-power consumption. As a core to the system, the proposed varactor-based LC tank circuit functions as an on-board probe on the IMD and facilitates oscilloscope-like fine-grained visibility into IMD operations. The electrically passive characteristics and small size of the proposed LC tank circuit are well suited for IMDs that are severely energy- and size-constrained. We discussed the performance metrics of TeleProbe and introduced the design efforts to improve these in TeleProbe. We have designed and implemented prototypes of both an IMD and an ED as embodiments of the proposed concept, and demonstrated the utility and capability of TeleProbe using the prototypes within three usage scenarios. The scenarios exemplified the passive as well as the active use of the proposed system for monitoring of analog/digital electrical signals and the wireless data transmission, respectively. Specifically, in addition to the zero-power passive monitoring, the active use of the proposed system achieved an ultra-low power wireless data transmission from the IMD to the ED with the energy consumption of 95 fJ/bit for the LC tank circuit as a physical layer and 114 nJ/bit for entire system where the processing core and the communication peripheral in the MCU are active.

5. CONCLUSION

As wireless embedded systems transition from lab-scale research prototypes to large-scale commercial deployments, providing reliable and dependable system operation becomes absolutely crucial to ensure high-quality services. However, the untethered nature of wireless embedded systems severely limits the ability to access, debug, and control device operation after deployment—*post-deployment or in-situ visibility*. A fundamental factor that limits post-deployment visibility is the resource-constrained nature of these devices, in particular the severe energy-constraints typically present in them. We believe that a hardware-based approach is both required and ideal to address the issue of limited visibility. In particular, we advocate a rethinking of hardware architecture to enable energy-efficient, yet fine-grained monitoring.

As embodiments of the above design principle, this thesis presents three hardware architectures, namely Spi-Snooper, Senergy, and TeleProbe, that ultimately achieves an electrical signal level visibility with near-zero power consumption. The first work, Spi-Snooper, presents a wireless sensor node platform that integrates a reliability co-processor into its hardware architecture. Rather than reporting the system status using the wireless channel, the co-processor processes the monitoring tasks within the platform, based on the bus-snooping technique that achieves the full access to the network communication in a transparent manner. Although the co-processor-augmented Spi-Snooper architecture significantly enhances the visibility and reduces the power overhead associated with the monitoring, there are certain types of wireless embedded systems that cannot afford to handle the overhead incurred by the co-processor. The second work, Senergy, addresses this issue using micro-scale energy harvesting from an idle sensor. With Senergy, we propose a sub-threshold exponential charge pump architecture that harvests energy from a photodiode sensor during idle time. Utilizing the harvested energy, the Senergy

wireless sensor node platform measures and transmits light intensity during active time, achieving perpetual operation of the wireless sensor node. The ability to collect sufficient enough energy to operate the entire platform easily compensates the power overhead incurred by the co-processor proposed with Spi-Snooper. Lastly, the third work, TeleProbe, proposes a contactless in-situ remote measurement system for implantable medical devices (IMDs), which achieves oscilloscope-like electrical signal probing with near-zero power consumption. By enabling a near-zero power contactless probing mechanism for IMDs, we demonstrate how the architectural support from hardware can help address the issue of visibility even for such severely resource-constrained wireless embedded systems.

REFERENCES

REFERENCES

[1] M. Dyer, J. Beutel, T. Kalt, P. Oehen, L. Thiele, K. Martin, and P. Blum, "Deployment support network," in *Wireless Sensor Networks*. Springer, 2007, pp. 195–211.

[2] N. Kothari, K. Nagaraja, V. Raghunathan, F. Sultan, and S. Chakradhar, "HERMES: A Software Architecture for Visibility and Control in Wireless Sensor Network Deployments," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2008, pp. 395–406.

[3] K. Whitehouse and et al., "Marionette: Using RPC for Interactive Development and Debugging of Wireless Embedded Networks," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 416–423.

[4] J. Yang, M. Soffa, L. Selavo, and K. Whitehouse, "Clairvoyant: a comprehensive source-level debugger for wireless sensor networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2007, pp. 189–203.

[5] Wikipedia, "List of wireless sensor nodes," March 2016. [Online]. Available: http://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes [Accessed: June 8, 2016].

[6] Moteiv, "Telos: Ultra low power IEEE 802.15.4 compliant wireless sensor module (Rev. B)," Telos datasheet, December 2004, [Preliminary].

[7] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 364–369.

[8] Texas Instruments, "MSP430F543xA, MSP430F541xA mixed signal microcontroller (Rev. B)," MSP430F543xA, MSP430F541xA datasheet, January 2010, [Revised July 2015].

[9] Texas Instruments, "MSP430F15x, MSP430F16x, MSP430F161x mixed signal microcontroller (Rev. G)," MSP430F15x, MSP430F16x, MSP430F161x datasheet, October 2002, [Revised March 2011].

[10] Texas Instruments, "Single-pole double-throw analog switch (Rev. G)," SN74LVC1G3157 datasheet, January 2003, [Revised June 2015].

[11] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *IEEE International Conference on Local Computer Networks (LCN)*, November 2004, pp. 455–462.

[12] A. Dunkels, "Full tcp/ip for 8 bit architectures," in *International Conference on Mobile Systems, Applications and Services (MobiSys)*, May 2003, pp. 85–98.

[13] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *ACM Conference on Networked Embedded Sensor Systems (SenSys)*, November 2007, pp. 335–349.

[14] Wikipedia, "TinyOS," March 2016. [Online].
Available: https://en.wikipedia.org/wiki/TinyOS [Accessed: June 8, 2016].

[15] A. Pataricza, I. Majzik, W. Hohl, and J. Hönig, "Watchdog processors in parallel systems," *Microprocessing and Microprogramming*, vol. 39, no. 2-5, pp. 69–74, 1993.

[16] I. Majzik, W. Hohl, A. Pataricza, and V. Sieh, "Multiprocessor checking using watchdog processors," *Computer Systems Science and Engineering*, vol. 11, no. 5, pp. 301–310, 1996.

[17] D. J. Lu, "Watchdog processors and structural integrity checking," *IEEE Transactions on Computers*, vol. 31, no. 7, pp. 681–685, 1982.

[18] A. Benso, S. Di Carlo, G. Di Natale, and P. Prinetto, "A watchdog processor to detect data and control flow errors," in *IEEE International On-Line Testing Symposium (IOLTS)*, 2003, pp. 144–148.

[19] A. Mahmood and E. J. McCluskey, "Concurrent error detection using watchdog processors-a survey," *IEEE Transactions on Computers*, vol. 37, no. 2, pp. 160–174, 1988.

[20] V. Sundaram, P. Eugster, and X. Zhang, "Efficient diagnostic tracing for wireless sensor networks," in *ACM Conference on Networked Embedded Sensor Systems (SenSys)*, 2010, pp. 169–182.

[21] H. Thane, D. Sundmark, J. Huselius, and A. Pettersson, "Replay debugging of real-time systems using time machines," in *International Parallel and Distributed Processing Symposium (IPDPS)*, 2003, pp. 8–pp.

[22] L. Luo, T. He, G. Zhou, L. Gu, T. F. Abdelzaher, and J. A. Stankovic, "Achieving repeatability of asynchronous events in wireless sensor networks with envirolog," in *IEEE International Conference on Computer Communications (INFOCOM)*, April 2006, pp. 1–14.

[23] B. Chen, G. Peterson, G. Mainland, and M. Welsh, "Livenet: Using passive monitoring to reconstruct sensor network dynamics," in *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2008, pp. 79–98.

[24] S. Choudhuri and T. Givargis, "Flashbox: A system for logging non-deterministic events in deployed embedded systems," in *ACM Symposium on Applied Computing (SAC)*, March 2009, pp. 1676–1682.

[25] M. Tancreti, M. S. Hossain, S. Bagchi, and V. Raghunathan, "Aveksha: A hardware-software approach for non-intrusive tracing and profiling of wireless embedded systems," in *ACM Conference on Networked Embedded Sensor Systems (SenSys)*, 2011.

[26] J.-T. Wu and K.-L. Chang, "MOS charge pumps for low-voltage operation," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 4, pp. 592–597, Apr 1998.

[27] L.-K. Chang and C.-H. Hu, "An exponential-folds design of a charge pump styled DC/DC converter," in *IEEE Power Electronics Specialists Conference (PESC)*, June 2004, pp. 516–520.

[28] C. Lu, S. P. Park, V. Raghunathan, and K. Roy, "Efficient power conversion for ultra low voltage micro scale energy transducers," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2010, pp. 1602–1607.

[29] P. Favrat, P. Deval, and M. Declercq, "A high-efficiency CMOS voltage doubler," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 410–416, March 1998.

[30] G. Palumbo and D. Pappalardo, "Charge pump circuits: An overview on design strategies and topologies," *IEEE Circuits and Systems Magazine*, vol. 10, no. 1, pp. 31–45, March 2010.

[31] EE Times, "Comparing regulated charge-pump and inductor-based DC/DC converters," September 2006. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1273125 [Accessed: July 7, 2016].

[32] J. Dickson, "On-chip high-voltage generation in mnos integrated circuits using an improved voltage multiplier technique," *IEEE Journal of Solid-State Circuits*, vol. 11, no. 3, pp. 374–378, June 1976.

[33] Advanced Linear Devices, "0.14V RC oscillator circuit with separate logic output buffer," Application Note (Schematic No. osc-42005.0), 2002.

[34] C. Lu, V. Raghunathan, and K. Roy, "Efficient design of micro-scale energy harvesting systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 254–266, September 2011.

[35] E. Sardini and M. Serpelloni, "Self-powered wireless sensor for air temperature and velocity measurements with energy harvesting capability," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1838–1844, 2011.

[36] I. Tolentino and M. Talampas, "Design, development, and evaluation of a self-powered GPS tracking system for vehicle security," in *IEEE Sensors*, October 2012, pp. 1–4.

[37] W.-G. Seah, Z. A. Eu, and H. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges," in *International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (VITAE)*, May 2009, pp. 1–5.

[38] Y. Hu, L. Huang, W. Rieutort-Louis, J. Sanz-Robinson, J. Sturm, S. Wagner, and N. Verma, "A self-powered system for large-scale strain sensing by combining CMOS ICs with large-area electronics," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 838–850, April 2014.

[39] Y. Hu, L. Huang, W. S. A. Rieutort-Louis, J. Sanz-Robinson, J. C. Sturm, S. Wagner, and N. Verma, "A self-powered system for large-scale strain sensing by combining CMOS ICs with large-area electronics," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 838–850, April 2014.

[40] J. Kim, P. Mok, and C. Kim, "23.1 A 0.15V-input energy-harvesting charge pump with switching body biasing and adaptive dead-time for efficiency improvement," in *IEEE International Solid-State Circuits Conference (ISSCC)*, February 2014, pp. 394–395.

[41] W. Zhou and L. Zuo, "A self-powered piezoelectric vibration control system with switch pre-charged inductor (SPCI) method," in *American Control Conference (ACC)*, June 2013, pp. 4753–4758.

[42] C.-Y. Tsui, H. Shao, W.-H. Ki, and F. Su, "Ultra-low voltage power management and computation methodology for energy harvesting applications," in *Symposium on VLSI Circuits (VLSIC)*, June 2005, pp. 316–319.

[43] J. Pan, B. Xue, and Y. Inoue, "A self-powered sensor module using vibration-based energy generation for ubiquitous systems," in *International Conference On ASIC (ASICON)*, vol. 1, October 2005, pp. 403–406.

[44] M. Suzuki, T. Takahashi, and S. Aoyagi, "Self powered inertia sensor based on vibration energy harvester using electret and ferroelectric plate," in *International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS & EUROSENSORS XXVII)*, June 2013, pp. 1843–1846.

[45] G. Palumbo and D. Pappalardo, "Charge pump circuits: An overview on design strategies and topologies," *IEEE Circuits and Systems Magazine*, vol. 10, no. 1, pp. 31–45, March 2010.

[46] L. Pylarinos, "Charge pumps: An overview," 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.4085&rep=rep1&type=pdf [Accessed: July 23, 2016].

[47] L. Gobbi, A. Cabrini, and G. Torelli, "A discussion on exponential-gain charge pump," in *European Conference on Circuit Theory and Design (ECCTD)*, August 2007, pp. 615–618.

[48] J. Dickson, "On-chip high-voltage generation in MNOS integrated circuits using an improved voltage multiplier technique," *IEEE Journal of Solid-State Circuits*, vol. 11, no. 3, pp. 374–378, June 1976.

[49] F. Su and W.-H. Ki, "Design strategy for step-up charge pumps with variable integer conversion ratios," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 5, pp. 417–421, May 2007.

[50] L.-K. Chang and C.-H. Hu, "High efficiency MOS charge pumps based on exponential-gain structure with pumping gain increase circuits," *IEEE Transactions on Power Electronics*, vol. 21, no. 3, pp. 826–831, May 2006.

[51] J. Witters, G. Groeseneken, and H. Maes, "Analysis and modeling of on-chip high-voltage generator circuits for use in EEPROM circuits," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1372–1380, October 1989.

[52] T. Tanzawa and T. Tanaka, "A dynamic analysis of the Dickson charge pump circuit," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1231–1240, August 1997.

[53] J.-T. Wu, Y.-H. Chang, and K.-L. Chang, "1.2 V CMOS switched-capacitor circuits," in *IEEE International Solid-State Circuits Conference (ISSCC)*, February 1996, pp. 388–389.

[54] J.-T. Wu and K.-L. Chang, "MOS charge pumps for low-voltage operation," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 4, pp. 592–597, April 1998.

[55] T. Kawahara, T. Kobayashi, Y. Jyouno, S.-i. Saeki, N. Miyamoto, T. Adachi, M. Kato, A. Sato, J. Yugami, H. Kume, and K. Kimura, "Bit-line clamped sensing multiplex and accurate high voltage generator for quarter-micron flash memories," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1590–1600, November 1996.

[56] R. Nopper, R. Niekrawietz, and L. Reindl, "Wireless readout of passive lc sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 9, pp. 2450–2457, 2010.

[57] R. Nopper, R. Has, and L. Reindl, "A wireless sensor readout systemŮCircuit concept, simulation, and accuracy," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 8, pp. 2976–2983, 2011.

[58] P.-J. Chen, S. Saati, R. Varma, M. S. Humayun, and Y.-C. Tai, "Wireless intraocular pressure sensing using microfabricated minimally invasive flexible-coiled LC sensor implant," *Journal of Microelectromechanical Systems*, vol. 19, no. 4, pp. 721–734, 2010.

[59] M. A. Fonseca, M. G. Allen, J. Kroh, and J. White, "Flexible wireless passive pressure sensors for biomedical applications," in *Solid-State Sensor, Actuator, and Microsystems Workshop*, 2006, pp. 37–42.

[60] P. V. Nikitin and K. S. Rao, "Theory and measurement of backscattering from rfid tags," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 6, pp. 212–218, 2006.

[61] J. S. Besnoff and M. S. Reynolds, "Near field modulated backscatter for in vivo biotelemetry," in *IEEE International Conference on RFID (RFID)*, 2012, pp. 135–140.

[62] S. J. Thomas and M. S. Reynolds, "A 96 mbit/sec, 15.5 pj/bit 16-qam modulator for uhf backscatter communication," in *RFID (RFID), 2012 IEEE International Conference on*. IEEE, 2012, pp. 185–190.

[63] W. S. Lee, A. Kim, B. Ziaie, V. Raghunathan, and C. R. Powell, "Up-link: An ultra-low power implantable wireless system for long-term ambulatory urodynamics," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2014, pp. 384–387.

[64] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan, "Vibration-based secure side channel for medical devices," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.

[65] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber *et al.*, "Wattsupdoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices," in *USENIX Workshop on Health Information Technologies*, 2013.

[66] M. Tancreti, M. S. Hossain, S. Bagchi, and V. Raghunathan, "Aveksha: A hardware-software approach for non-intrusive tracing and profiling of wireless embedded systems," in *ACM Conference on Networked Embedded Sensor Systems (SenSys)*, 2011.

VITA

VITA

Woo Suk Lee received the B.S. degree in Electronics and Electrical Engineering and the M.S. degree in Electronics, Electrical and Instrumentation Engineering from Hanyang University, Korea, in 2007 and 2009, respectively. Since 2010, he has been pursuing the Ph.D. degree in Electrical and Computer Engineering at Purdue University, West Lafayette, IN. He was with Microsoft in summers of 2012, 2014, and 2015. His research interests include hardware and software architectures for embedded computing systems, the Internet of Things (IoT), implantable and wearable medical devices with an emphasis on reliable system design, low power design, and micro-scale energy harvesting.

He received the design contest awards at the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED) in 2014, and at the IEEE International Conference on VLSI Design (VLSID) in 2015.