Energy-Efficient Ultra-Low Power PPG Wearable System for Long-Term IoT Health Monitoring

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment

of the requirements for the degree

Master of Science

by

Peter Le

December 2023

Abstract

This work presents an ultra-low power (ULP) photoplethysmography (PPG) end-to-end wearable system that offers continuous physiological monitoring capabilities for Internet-of-Things (IoT) applications. The increasing demand for wearable health monitoring devices necessitates the development of such energy-efficient PPG systems. Leveraging existing ULP System-on-Chip (SoC), Analog-Front End (AFE) chip, and a Bluetooth Low-Energy Transmitter (BLE-TX) chip, our work integrates these chips into an integrated wearable IoT system for continuous real-time PPG health-monitoring. Employing a systems-level driven approach, rapid prototyping of the system leveraging the flexible design of the ULP chips enables a 'plug-and-play' approach towards IoT system design. All components are fabricated using the TSMC 65nm Low-Power (LP) CMOS process. This system achieves minimal power consumption at just 62.8 µW during idle periods and 148.5 µW during active data transmission when aggressively duty-cycled at 2.5%. This compact system measures in at just 75mm x 70mm x 35mm and achieves an operational period of 147 days when powered by a high-capacity 3.7V, 500-mAh CR3555 rechargeable battery, making it suitable for ULP IoT applications. This thesis will first by introduce a modular design language for low-powered end-to-end systems, followed by a practical demonstration of this concept through PPG system integration, from the process of PCB design, system testing, to optimization for low-power.

Acknowledgments

I would like to express my sincere gratitude to those who have contributed to the completion of my master's degree. Special thanks go to my fellow students, Daehuyn Lee and Omar Faruqe, for helping with the testing for the project, which is a very huge time commitment from their end. Xinjian Liu's and Peng Wang's guidance and expertise were also invaluable throughout the entire process.

I am deeply thankful for the support and encouragement provided by my advisor, Benton Calhoun, whose insight and feedback greatly helped with the project, as well as encouragement and mentorship towards the completion of my Master's program.

I would also like to extend my appreciation to Omar Abdelatty at the University of Michigan for his collaboration and assistance with the integration of the Bluetooth Low-Energy transmitter board. His contribution played a crucial role in completing the integrated systems testing.

Furthermore, I would like to thank my supervisor, Geoffrey Edelman, and branch head, David Calvo, of the Acoustic Division at the Naval Research Laboratory, for their understanding during my Master's program. Their flexibility with the projects and work deadlines allowed me to manage my time commitments between work and school.

Lastly, I would like to express my gratitude to my family and friends for their unwavering support and encouragement during the process of my Master's program.

Table of Contents

Contents

Chapter 1

Introduction

1.1 Problem Statement & Motivation

Rapid advancements in IoT applications, spanning from personal healthcare [4][6][7], to industrial monitoring to attritable sensors for defense [38], necessitate self-powered sensor nodes capable of sensing, computing, and transmitting vital information to remote devices. The specific environmental conditions in which these sensor nodes operate and the unique signals they are designed to capture requires different system-level requirements, including power budgets, sensing modes, transmission range, storage capacity, and more. To meet these diverse demands, system designers often opt for a generalized system architecture to IoT applications such as processor speed, sensing modalities, power source, transmission of data to the user, and user interaction given the data. The conceptualization of the IoT system is depicted in Figure 1.1.1 through a generalized block diagram.

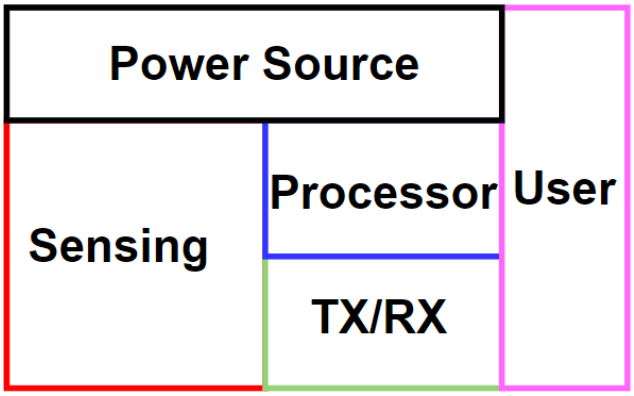


Figure 1.1.1: Typical modules of a self-powered IoT system

From our prior works, we have demonstrated very adaptive circuits, specifically fine-tuned for ultra-low power applications in self-powered and IoT systems. These components include a four-channel multi-modal V/I/R/C AFE [4], a microcontroller (MCU) [13], a multiple-input-multiple output energy energy harvesting management unit (MIMO-EHPMU) [37], and a Bluetooth low-energy transmitter (BLE-TX) from University of Michigan [34]. However, we haven't integrated all of it into a complete system designed for continuous wearable-health.

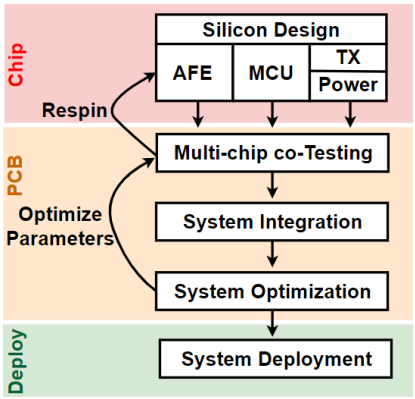


Figure 1.1.2: Conventional system design process

Conventional methods for the system-level design of wearable IoT systems typically follow a synchronous process, illustrated in Figure 3.1.1. In this sequence, the chip components undergo initial design (silicon design), followed by a comprehensive series of tasks, such as multi-chip co-testing, integration onto a Printed Circuit Board (PCB), optimization of board-level parameters, and, finally, integrated system deployment.

However, due to the involvement of multiple chips co-developed from various universities and organizations, and the difference of scheduling between each different group, often involving multiple revisions, the protracted nature of the system design process fails to align with the rapid pace of evolving IoT requirements. Any shift in requirements inevitably results in project delays caused by the continuous revisions due to the nature of the synchronous process. In order to achieve the stated goal of designing an ultra-low power or self-powered health monitoring system, we will need to adopt a new approach to system-level design.

1.2 Thesis Contribution

1. Our PPG system architecture incorporates aggressively duty-cycled sub-components at the system level, employing existing application-specific integrated circuits (ASICs) optimized for ultra-low power that integrate transmitter (TX), AFE, SoC, LED driver, and Bluetooth-Low Energy Transmitter (BLE-TX) transmission to the phone. This design minimizes power consumption across all components, achieving the lowest end-to-end power consumption.
2. Utilizing a systems-level approach, the modular iterative prototyping of the wearable IoT system leverages a flexible modular approach towards the integration of existing ultra-low-power chips, allowing for a 'plug-and-play' method in the development of IoT systems.

The specific contributions of this thesis are summarized as follows:

- Development of an Ultra-low power wearable platform with flexible modal and IoT capabilities.
    - Design of PCBs integrating low-power chips using the modular design concept: System-on Chip baseboard and the Analog-Front End daughterboard.

- System integration and testing.
    - Collaboration with University of Michigan to integrate BLE-TX daughterboard to the integrated system.
    - Low-power solution testing and verification of the system to meet low-power requirements.
    - System Modeling and applications of prior works to engineer a low-power solution.
    - Integration of the RISC-V software to the platform.

1.3 Thesis Organization

The rest of the thesis is structured as follows:

Background & Prior Arts: Chapter 2 provides background information on typical PPG circuits, as well as provide a general overview of components typical PPG systems are using. Additionally, it includes a comparison of various state-of-the-art end-to-end PPG systems.

Design & Implementation: Chapter 3, we introduce the concept of a modular system design methodology for rapid prototyping for IoT systems. We then implement this concept in practice by providing a comprehensive overview of the end-to-end integrated PPG system. In Chapter 4, we detail the system's implementation, highlighting its key components and functionalities, as well as printed circuit board (PCB) layouts.

Measurement: In Chapter 5, demonstrate that our end-to-end system excels in most wearable low-power metrics similar to other state-of-the-art devices among various design metrics, illustrating our efforts to achieve optimal power efficiency from a system-level perspective. The fully integrated ultra-low power PPG system with very low power consumption at 62.82 µW

when at idle power, 148.5 µW when at active power and transmitter is active and duty cycled at 2.5%, and an average power consumption of 141.10 µW.

Conclusion: Chapter 6 provides a comprehensive summary of the low-power end-to-end Photoplethysmogram (PPG) system design and integration endeavors. It also discusses prospective avenues for enhancing the system's resilience and robustness, and additional efforts to make the system more self-powered.

Chapter 2

Background

This chapter provides the background information on various types of existing circuits Photoplethysmography (PPG), and provides a summary of current-state of the art end-to-end integrated wearable PPG systems.

2.1 Photoplethysmography Circuits

Photoplethysmography (PPG) is a non-invasive, low-cost optical technique for continuous monitoring of the heart rate. It was first introduced in the 1930s, describing the reflection mode to monitor blood volume changes [1]. PPG has become increasingly popular in the field of wearable technology, with commercial products like fitness trackers or smartwatches. These new applications and products offer continuous physiological data monitoring and analysis, significantly improving our life quality and productivity.

The theory of operation for PPG is based on the concept that there is a variation in how blood reflects light based on different oxygenation levels. Most common PPG sensors use either an infrared light-emitting diode (IR-LED) or a green LED as the main light source. IR-LED and green LED are most commonly used because of their deeper penetration into muscle tissues, which achieves higher resolution and accuracy. Between the wavelengths of 400 and 2000 nm, the maximum optical penetration depth is less than 7mm. As a result, the wrist or neck are considered ideal locations for LED placement [10].

Typically, a minimal signal-to-noise ratio (SNR) of 28.5 dB is required to achieve a heart rate error of less than 1 beat per minute (bpm) and a minimum SNR of 39 for SpO2 errors of <2%. [3] PPG signals require the addition of Current Digital-to- Analog Converters (IDACs) to the AFE circuit to cancel the ambient light, which adds extra power consumption to the overall system [11]. With the goal of minimizing power consumption in the hardware, it is essential to identify at a system level and incorporate the additional necessary components to achieve power reduction.

This requires optimizing the integration between microcontroller units, communication units, and PPG components. Wearable PPG system usually consists of an analog front-end for signal conditioning, and LED driver for the PPG sensor, processor, data storage, and data transmission.

The readout circuit must meet two critical requirements to effectively process the PPG signal. Firstly, it needs to achieve a high signal-to-noise ratio (SNR) and dynamic range. This is essential because the PPG signal contains both an AC component, which represents variations in blood volume due to heartbeats, and a substantial DC component. The DC component is primarily caused by light absorption in non-arterial areas of the body, including bones, tissues, and tendons [10]. Therefore, before extracting the meaningful AC component, it's crucial to eliminate this interfering DC component.

[4-5] found that LEDs with a wavelength of 535 nm (green LED) were preferable, primarily due to their larger AC-to-DC current ratio compared to red or IR-LEDs. While the AC components of different colored LEDs were found to be similar, a model utilizing the commercial off-the-shelf SFH7600 [6] revealed that red and IR LEDs exhibit DC components nearly 10 times greater than green LEDs. This necessitates a higher dynamic range for the analog front end

(AFE). LED power dominates the system's power [4].

The Transimpedance Amplifier (TIA) and the Light-to-Digital Converter (LDC) are the two leading readout topologies extensively employed for PPG sensors under 500 uW [11]. A TIA readout circuit incorporates a transimpedance amplifier and an ADC to convert the photocurrent generated by the PPG sensor into a voltage, which is subsequently quantized into digital output. Conversely, the LDC topology utilizes an integrator to quantize the photocurrent into digital

output.

Notably, the author observed that PPG sensors that are based on LDC technology exhibited superior performance in terms of signal-to-noise ratio (SNR), dynamic range (DR), and power efficiency, which are important for chest-based sensing. However, [2] found that the mean amplitude from finger measurements was significantly larger than that of chest or arm placements; therefore, the TIA readout topology is sufficient for finger-based PPG sensing. Moreover, the fingertip achieves the best signal-to-noise ratio (SNR) [7]. To achieve a balance between power and SNR, the PPG's VLED (LED voltage) of 2.5–2.8 is utilized for self-powered operations. Additionally, this approach [7] offers orders-of-magnitude reductions in power consumption compared to other alternatives, thanks to the aggressive duty cycling. A decrease in the average LED power requires an increase in the AFE power, driven by the increased bias current demand resulting from the need for a faster AFE response. Therefore, from the model, [8] optimized the hardware parameters for a power budget of 20μW, requiring 11.5 bits of ADC ENOB, 0.6VDD, 0.5nA TIA reference current, 26 Hz sampling rate, and an LED voltage of 2.5–2.8V for all skin tones. More comprehensive explanations for the parameters were noted in [4], with the low TIA bias current greatly reducing power consumption while providing sufficient SNR.

Event-driven sensing achieved the best power performance given that the LED also has to be on when the peak or change in slope occurs in the PPG signal. However, event-driven sampling necessitates additional software and circuitry for peak detection as well as reconstruction of the signal, increasing the power budget for the SoC. Therefore, in this work, TIA readout topology and aggressive duty-cycling are utilized to achieve ultra-low power.

The main takeaways for the design of our integrated PPG system therefore are listed as follows:

- PPG LED voltage between the ranges of 2.5-2.8V
- Finger-based PPG interface for optimal SNR and power.
- Green LED is preferable due to its larger AC-to-DC current ratio
- A TIA readout topology is implemented due to its simplicity

- Special emphasis will have to be placed on reducing LED power, as it will be a significant contributor to power consumption (i.e. aggressive duty-cycling)
- Aggressive duty-cycling will be utilized due to its minimal software requirements and to reduce processing power on the RISC-V core

Chapter 3

System Description & Design

3.1 Modular System Design

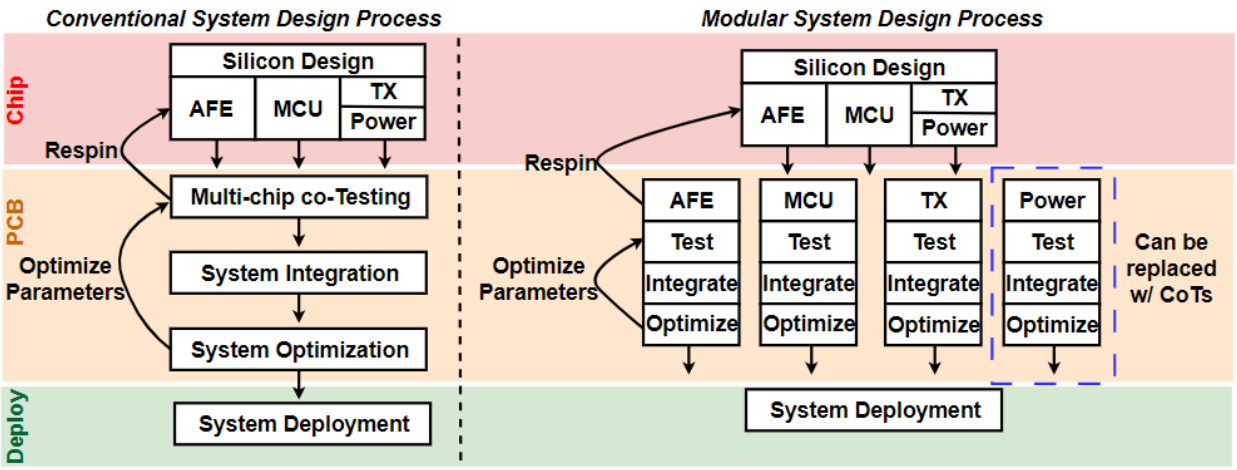
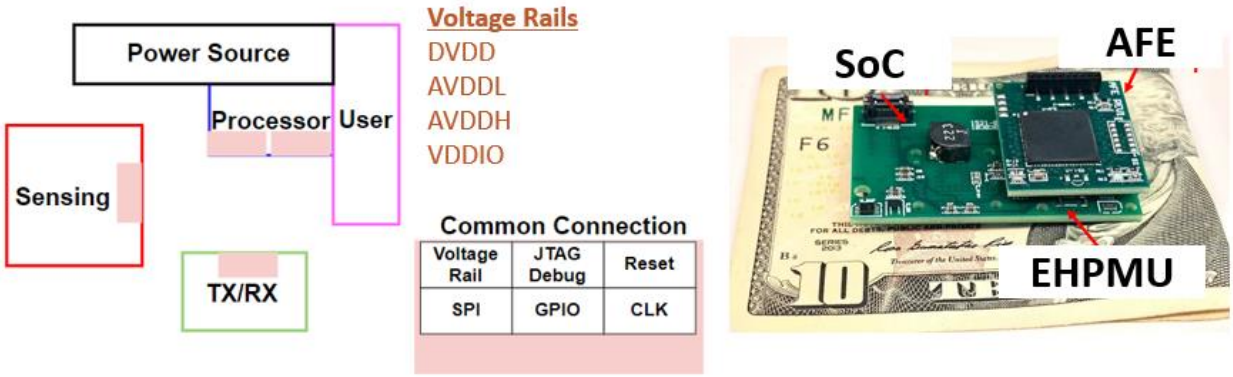


Figure 3.1.1: Flow-chart outlining conventional system design versus our modular system design.

Traditional approaches to the system-level design of wearable IoT systems often adhere to a conventional monolithic process, as shown in Figure 3.1.1. In this sequence, the chip components undergo initial design (silicon design), followed by an extensive series of tasks, including multi-chip co-testing, integration onto a Printed Circuit Board (PCB), optimization of board-level parameters, and ultimately, system deployment. Alternatively, in the silicon design approach, all the essential functionalities for IoT (MCU, AFE, TX/RX, and power) are consolidated onto a single die. However, this integration limits system modularity for future



(a) (b)

Figure 3.1.2: a) Common design features in the modular design concept, (b) AFE daughter board integrated onto the SoC baseboard as an adapter.

revisions. The synchronous nature of this process implies that any revisions necessitate multiple iterations of the integrated system, posing a challenge to meet the rapidly evolving requirements of IoT.

Our methodology involves separating the IoT functionalities across multiple chips, specifically the AFE, MCU, TX, and power components. Each of these elements is co-designed independently by separate teams, each adhering to distinct interface requirements. Employing a systems-level approach, we conceptualize each chip as a block within the design process. This comprehensive process encompasses chip design for testing, integration, and optimization in a unified manner, with each chip progressing concurrently or asynchronously in a streamlined pipeline.

In this design process, the SoC, which contains the RISC-V processor, is the baseboard. The SoC core includes the processor, internal and external memory, clocks, as well as the system bootup configurations, usually stored in non-volatile memory. The daughterboard PCB shares similar dimensions to the TX/RX, AFE, and power source, allowing for interchangeability between different components. Additionally, the daughterboards also share SPI, GPIO, debug, and power busses, standardizing the connections between the chips, as shown in Figure 3.1.2. This strategy draws parallel with the concept of add-on hats for the Raspberry Pi, ultimately aiming for a 'plug-and-play' system design. Notably, this approach facilitates rapid iterative revisions of

individual components, as well as utilizing breakout boards equipped with CoTs components for rapid prototyping and functional testing of between boards. Illustrated in Figure 1.1, a portion of the block can be seamlessly replaced with various modules customized to meet specific application requirements. For instance, if there's a shift in the sensing method or a change in the power source as required by a specific use-case, the remaining blocks remain unaffected, eliminating the need for a complete redesign of the system. Similarly, if the processor on the baseboard requires some a modification, the daughterboards should remain unchanged.

3.2 Hardware Design

The proposed end-to-end PPG system is shown in Figure 3.2(a) and is comprised of a main baseboard that houses both the SoC and energy-harvesting power module unit (EHPMU), alongside two daughter boards (AFE and TX from University of Michigan), and an externally connected LED driver board affixed to the fingertip. An Android phone using the Android Beacon Library [32] is used to stream constant data packages transmitted from the PPG system. The integrated PCB setup (PCB size 75mm x 70mm x 35mm) is shown in Figure 3.2(b).

Therefore, the system requirements for an end-to-end integrated wearable PPG system consists of the following:

- Customized flexible processor for on-body PPG streaming for health applications, with an Analog Front End (AFE), data storage, and a method for data transmission for an end-to-end solution
- Low power, aiming for around (50-100 uW) of idle power and (<1mW) of active power to meet future self-powered energy goals.

- Common design language between the daughterboards for inter-operability similar to a Raspberry Pi hat.
- Small form-factor that can be easily realized for wearable health applications in a clinical setting, as per the requirements of ASSIST.

To realize the system requirements, we have developed an integrated system which consists of the following PCB boards:

- **A baseboard SoC** provides power regulation, and the core logic chip is responsible for managing the Serial-peripheral Interface (SPI) interface. It also incorporates JTAG debugging interfaces as well. The baseboard features an on-chip clock and an optional external 32 kHz clock on the board. Furthermore, it includes non-volatile memory for data storage. For connectivity, the baseboard interfaces with the daughter boards via board-to-board connections via mating pin headers for integration with the BLE-TX board and AFE daughter boards.
- **An AFE daughterboard** provides the necessary analog circuitry required for signal conditioning for the PPG's measurements, bridging the gap between the PPG sensor interface and the digital SoC.
- **A TX daughter board** enables BLE beacon transmission of the measured PPG data to a smartphone.
- **An LED driver board** featuring an SFH7060 reflective-mode PPG LED sensor designed for precise fingertip measurements. The SFH7060 sensor comprises of two green LEDs, a red LED, an infrared LED, and a photodiode.

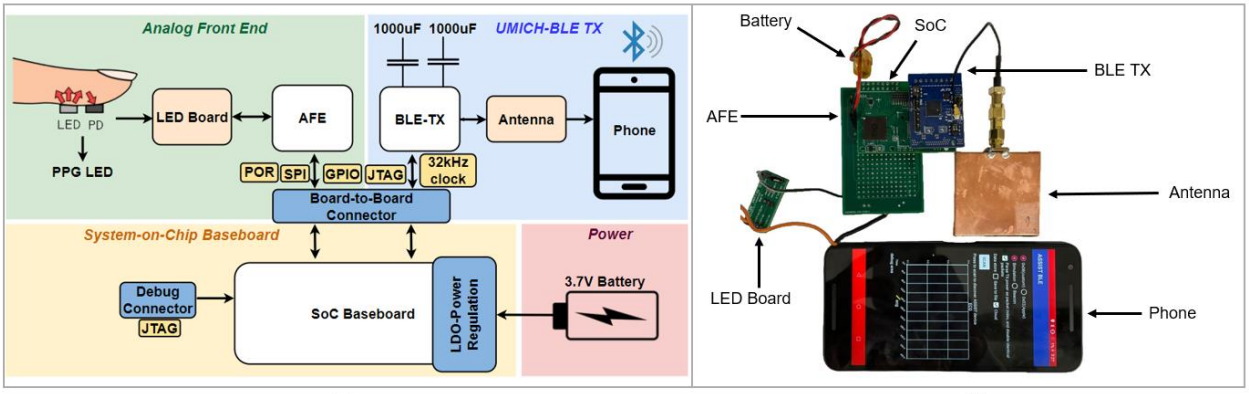


Figure 3.2: a) Block diagram of the integrated PPG system, (b) Integrated hardware

Using the modular design approach mentioned in the previous sections, the board designs for all of the daughter boards (AFE and the BLE-TX) are specified to be mated via 2.54mm pin heads to the SoC baseboard which contains the RISCV core. The flexibility in having the processor as the baseboard ensures compatibility as well as facilitating iterative and efficient revisions of the daughter boards. A more detailed description of the integration efforts is provided in Chapter 4.

Chapter 4

System Implementation

4.1 Hardware: Integrated System

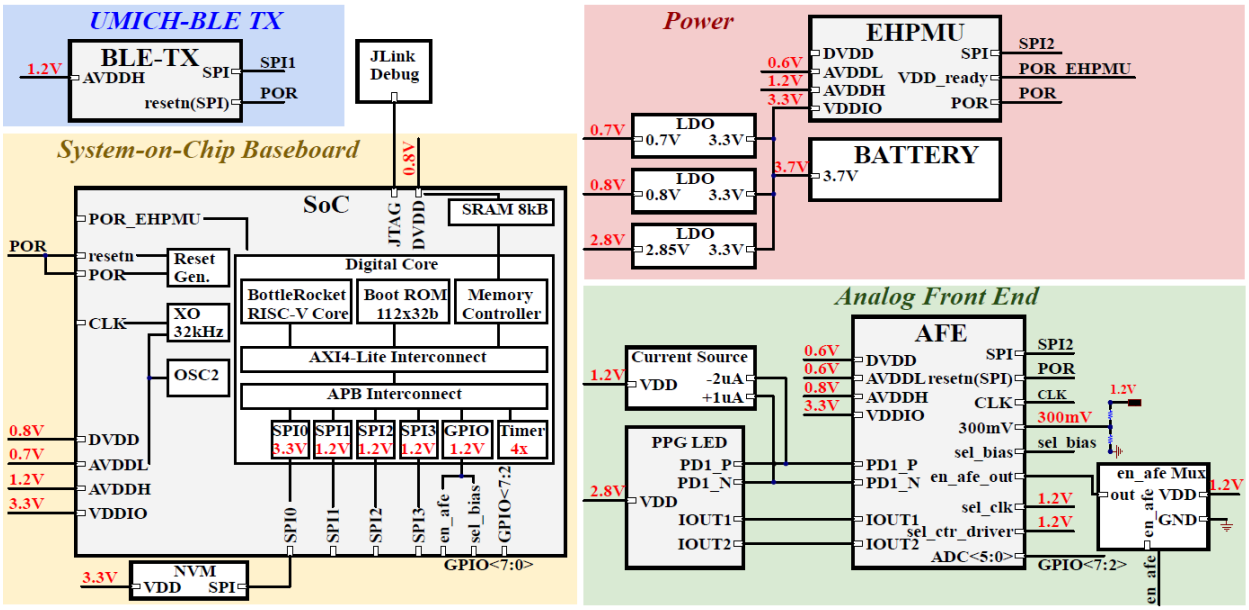


Figure 4.1: Detailed integrated system block diagram

A detailed system block diagram for the integrated system is shown in Figure 4.1. The baseboard SoC provides a common board-to-board interface with the daughterboard, sharing similar voltage rails, SPI, reset, and GPIOs. These shared voltages—specified as digital voltage (DVDD), analog low-voltage (AVDDL), analog high-voltage (AVDDH), and IO voltage (VDDIO)—were initially set at 0.6V, 0.6V, 1.2V, and 3.3V, respectively. However, real-world variations in the chip necessitated different voltage requirements for each chip, prompting the

Table 4.1.1: Soc Power Component Breakdown

Name	Voltage (V)	Component	Block
DVDD	0.8	SoC	RISC-V, SRAM
	0.6	AFE	ADC
AVDDL	0.7	SoC	CLK, FLL, OSC
	0.6	AFE	LED, ADC
AVDDH	1.2	SoC	POR, ESD, VDDL
	0.8	AFE	PPG
	1.2	TX	TX
VDDIO	3.3	NVM	
	3.3	AFE	
	3.3	TX	(optional)
	2.8	LED	

Table 4.1.2: Digital Interfaces

Digital Interface	Voltage (V)	Connection
SPI0	3.3	NVM
SPI1	1.2	BLE TX
SPI2	1.2	EHPMU
SPI3	1.2	AFE
GPIO	1.2	General-purpose I/O
JTAG	1.2	Debug

addition of extra LDOs in the design to finely adjust the voltages as needed. The resulting breakdown of the voltage rails for each of the chip components can be found in Table 4.1.1.

Each of the daughterboard will share common connections for the voltages, reset, GPIO, and debug. The custom chip's SPI (SPI1-SPI3) voltage rails are set to 1.2V to reduce power, while the COTS NVM (SPI0) uses a 3.3V rail. A breakdown of the system's digital interfaces is shown in Table 4.1.2.

4.2 Hardware: System-on-a-Chip

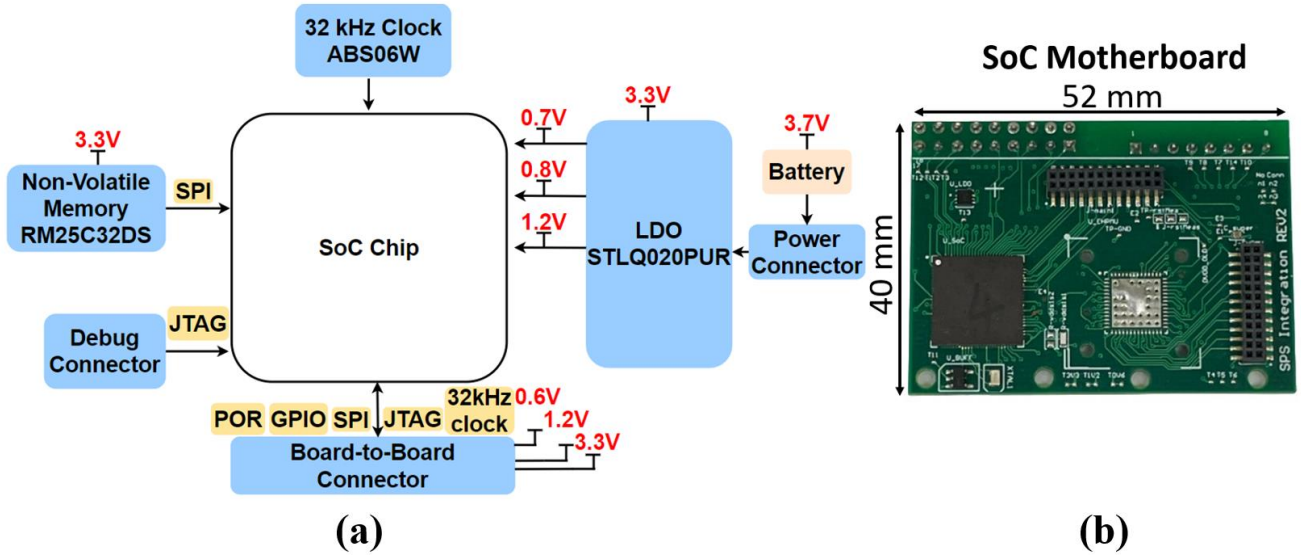


Figure 4.2: (a) Block diagram for the SoC PCB (b) and the PCB board.

A modular System on a chip motherboard (PCB size 40mm x 52mm) is shown in Figure 4.2. The board consists of an SoC located on the left side, non-volatile (NVM) on the bottom side, and EHPMU is on the right. The AFE and TX are connected via 2.54mm male pin headers.

The 6-layer SoC PCB incorporates the BottleRocket RISC-V processor derived from Rocket processor [16] using dynamic leakage-suppression logic [13], and an energy harvester processing unit (EHPMU) processor. However, in our system, the EHPMU is software disabled and is instead replaced by the 3.7V coin-cell battery. An STLQ020PUR analog low dropout (LDO) regulator takes in the 3.3V input and produces the 0.7V, 0.86V, and 0.8V outputs for the SoC.

The SoC's digital backend operated robustly at 0.7V, 1.2V for the analog backend, while 3.3V is used to power the non-volatile memory (NVM). The SoC system features 8 6-bits GPIO ports, with GPIO0 used for bias voltage selection and GPIO1 for the AFE power gating control. GPIO 2 through 7 are used to collect data from the output of the AFE's ADC. The data is then stored into the SRAM. The SoC processor then reconstructs the data into a format the TX can transmit.

The SoC communicates with the rest of the chips over four SPI interfaces: NVM on SPI0, BLE TX on SPI1, EHPMU on SPI2, and AFE on SPI3. A boot select jumper provides bootup selection. When BOOT_SEL = 1 the core executes instructions stored on the ROM. When BOOT_SEL = 0, the system enters a spin loop with instruction 0x0000A001 until JTAG is acknowledged. External JTAG pin headers provide an interface with a Segger J-Link debugger, which provides debugging and observability for chip testing. A common power-on-reset line is provided by the SoC to ensure that the system will start in the same condition every time the system is powered up. The 0.7V rail provides digital voltage, while the 0.8V powers the SRAM.

The bootup program in the read-only memory (ROM) relies on persistent non-volatile memory, specifically the Adesto RM25C32DS NVM, which has 32KBIT of memory. The SPI interface device operates at 1.8MHz read speed and features a compact form factor, measuring at just 4.4mm x 3mm with and consumes around 600 μW of power when reading from memory.

In order to perform the data acquisition and processing at ultra-low power levels, an application-specific integrated circuit (ASIC) system-on-chip (SoC) was designed. The SoC is implemented in a 65nm CMOS Low-Power (LP) technology. The SoC includes a 32-bit microprocessor, an 8KB SRAM, a boot-ROM, a crystal oscillator, and a suite of digital interfaces for flexible chip-to-chip communication. The crystal oscillator generates a 32KHz clock frequency for the SoCs and other connected chips. With 4 serial-parallel-interfaces (SPIs) and 8-b GPIO, the SoC can control and communicate with multiple different chips. In the integrated system, the SPI controls the AFE, TX and NVM. Once the SoC starts up, the integrated boot-ROM allows the SoC to automatically load and execute the program from one of the SPI interfaces. Except for the SRAM which needs a 0.8V supply voltage, the other analog components are powered by a 1.2V voltage rail. A 0.7V rail is used for the digital block. With an off-chip LDO to generate the 0.8V rail, the whole system only requires a single 1.2V supply. The SoC read operation draws 8.05

µW while the write operation draws 8.1 µW, with IO interfaces power consumption included, demonstrating low-power read/write operations.

4.2 Hardware: Analog-Front-End

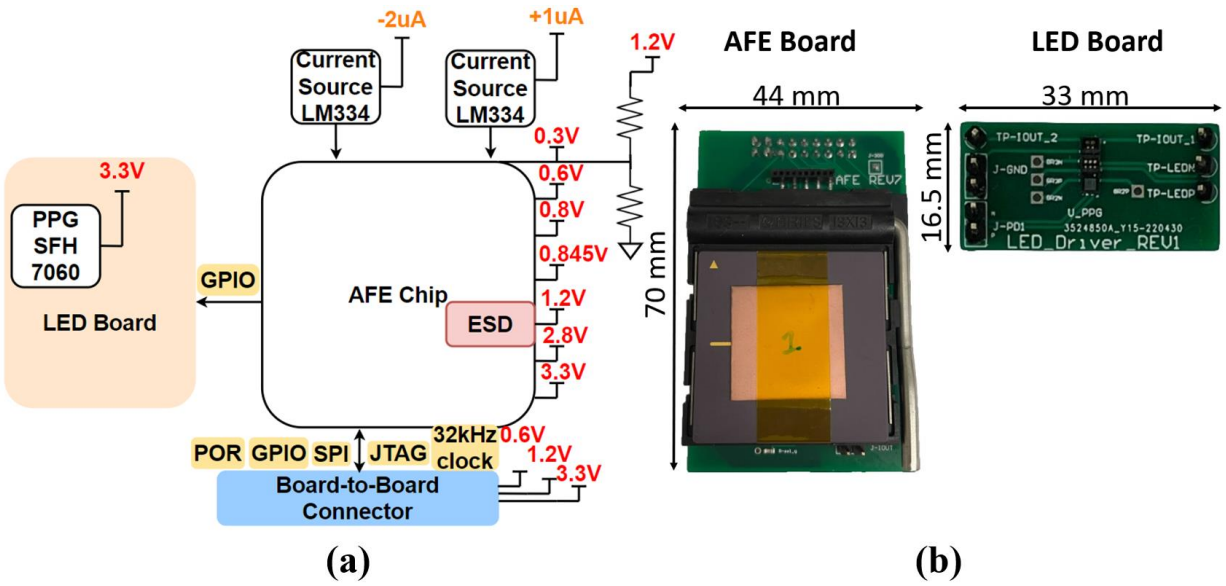


Figure 4.2.1: (a) Block diagram of the AFE subsystem and photo of the AFE and (b) LED PCB.

The AFE board includes an AFE chip in a PGA100 package, achieving a small form factor. The overall system is shown in Figure 4.2.1. The internal components of the AFE chips is shown in Figure 4.2.3. Additional 0.1uF and 1uF decoupling capacitors are shunted to each voltage rail in the AFE. An Abracon ABS06W-32.768KHZ-K-2-T 32kHz source clock from the main SoC board or an onboard the on-chip 32kHz clock can be selected via the SEL_CLK jumper to power the AFE's digital logic. An 18-pin pin header provides board-to-board connection between the SoC and AFE daughter board. The AFE offers 6 ADC outputs. The PD1 and PD2 connections are driven to the LED driver board, enabling duty-cycling control. The TIA pinouts allow direct observation from the transimpedance amplifier buffer. While the multi-modal AFE supports gas, ECG, and oxygen sensors, these functionalities have been disabled in this integrated system to

save power, but future works may include these options. In order to achieve further low power, the digital ESD pad and the digital rail was separated, with the ESD operating via a 1.2V rail, while the digital rail operates at 0.845V. A low supply voltage of 0.6V is used to supply the rest

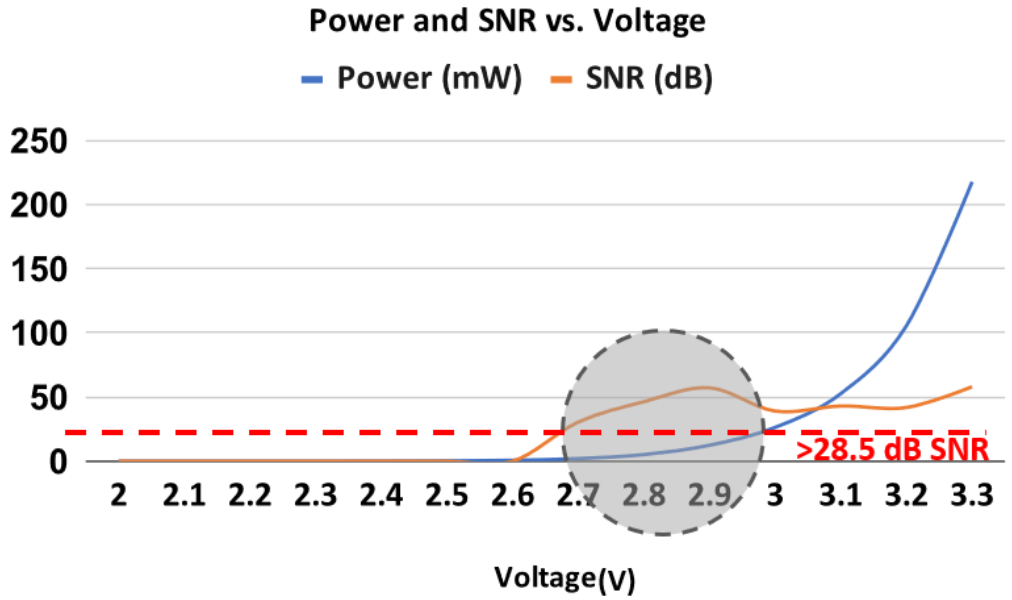


Figure 4.2.2: Power vs. SNR of the PPG's green LED.

of the digital backend to reduce power consumption without significant front-end noise performance degradation [4]. A fixed-value off-chip 300mV voltage reference is supplied via a resistor divider network from the 1.2V rail. The SEL_CTRL_DRIVER line is responsible for the digital select function for LED power gating.

2.8V was chosen as the driving voltage for the LED driver based on the measurements in Figure 4.2.2, which indicated that the lowest driving LED voltage range that achieves the highest SNR is between the ranges of 2.7V-2.9V. A minimal SNR of 28.5 dB is required to achieve a heart rate error of less than 1 beat per minute (bpm) [11], and is shown as the dotted red line, and therefore acts as our threshold line. At 2.8V, the power consumption for the LED driver was measured at 2.1 mW with an SR of 40 dB when ran with no duty-cycling.

The transimpedance gain of the AFE is fixed at 4MΩs. Additionally, two Texas Instruments LM334 constant current sources are used to provide constant bias current to the TIA amplifiers and are powered by a 1.2V coin-cell battery, providing a bias current of +1uA and -2uA

respectively. The AFE is equipped with three jumpers: one for J-SEL-CLK, another for SEL_CTRL_DRIVER, and a third for establishing the IOUT connection to the PPG, facilitating duty-cycling of the LED driver.

The TIA's reference voltage of 300mV is provided by a resistor divider network. Additional observation pinouts are provided via 1.27mm pitch headers.

To achieve a high SNR from the PPG and allow for flexible sensor placement, the PPG is integrated onto a separate LED board (33 mm x 16.5 mm) as shown in Figure 4.2.1(b) and connected to the AFE board using jumper cables. We employed the Osram Biofy SFH7060 PPG sensor [5], utilizing only the green LEDs. [4][5] found that LEDs with a wavelength of 535 nm (green LED) were preferable, primarily due to their larger AC-to-DC current ratio compared to red or IR-LEDs. While the AC components of different colored LEDs were found to be similar, a model utilizing the commercial off-the-shelf SFH7600 [6] revealed that red and IR LEDs exhibit DC components nearly 10 times greater than green LEDs. [4] found that when measured at the fingertip, the LED current ranges 745μA-2.37mA, while when measured at the wrist, the LED current is 40.7mA - 139.8mA. Hence, our system is designed to measure PPG from the fingertip. The LED board operates on 2.8V generated by the LDO from the SoC motherboard and consumes on average anywhere from 7.8μW up to 5.3mW with 2.5% duty-cycle to always-on respectively.

The readout circuit needs to achieve a high SNR and dynamic range to effectively handle the PPG signal, which comprises a substantial DC component that must be eliminated prior to extracting the AC component representing the actual PPG signal. The AC component is generated by variations in blood volume within the arteries due to heartbeats, while the DC component is formed by light absorption in other areas of the body, such as bones, tissues, and tendons [10].

Notably, the author observed that PPG sensors that are based on LDC technology exhibited superior performance in terms of SNR, dynamic range (DR), and power efficiency, which are important for chest-based sensing. However, [2] found that the mean amplitude from finger measurements was significantly larger than that of chest or arm placements; therefore, the TIA readout topology is sufficient for finger-based PPG sensing. Moreover, the fingertip achieves the

best SNR [7]. To achieve a balance between power and SNR, the PPG's LED voltage of 2.5–2.8 is utilized for self-powered operations. Additionally, this approach [7] offers orders-of-magnitude reductions in power consumption compared to other alternatives, thanks to the aggressive duty cycling. A decrease in the average LED power requires an increase in the AFE power, driven by the increased bias current demand resulting from the need for a faster AFE response. Therefore, from the model, [8] optimized the hardware parameters for a power budget of 20μW, requiring 11.5 bits of ADC ENOB, 0.6VDD, 0.5nA TIA reference current, 26 Hz sampling rate, and an LED voltage of 2.5–2.8V for all skin tones. More comprehensive explanations for the parameters were noted in [4], with the low TIA bias current greatly reducing power consumption while providing sufficient SNR.

Event-driven sensing achieved the best power performance given that the LED also has to be on when the peak or change in slope occurs in the PPG signal. However, event-driven sampling necessitates additional software and circuitry for peak detection as well as reconstruction of the signal, increasing the power budget for the System-on-chip (SoC). Therefore, in this work, The PPG system utilizes a TIA readout topology with aggressive duty-cycling in order to achieve ultra-low power.

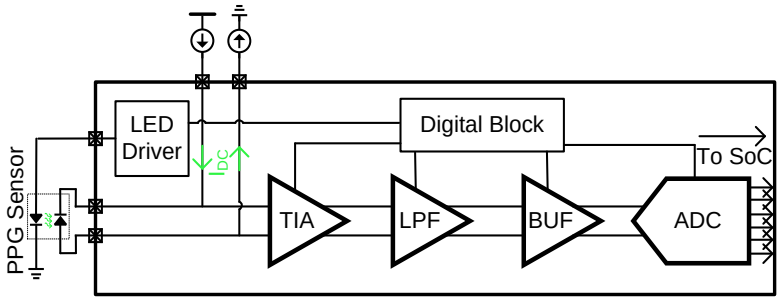


Figure 4.2.3: Overview of internal blocks of the AFE

4.3 Hardware: Bluetooth Low-Energy Transmitter

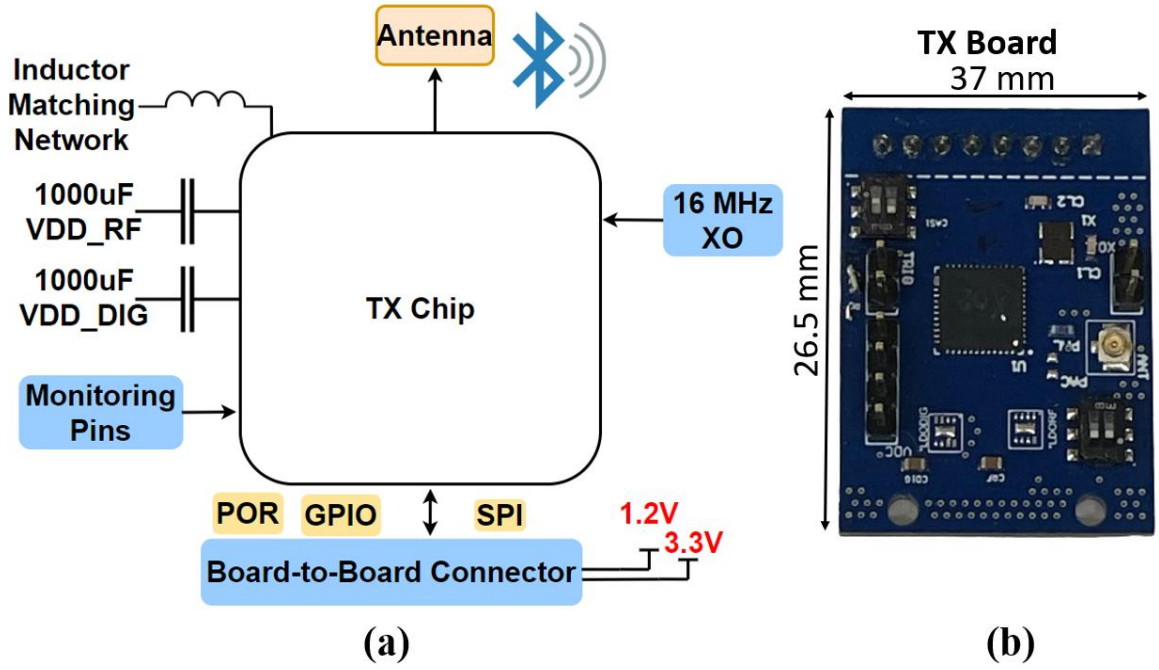


Figure 4.1.31: (a) Block diagram of the TX subsystem and (b) photo of the TX PCB.

The Michigan TX board features a four-layer design, operating at a 1.2V supply voltage power both the RF and digital blocks via a 1.2V step-down LDO. In order to mitigate voltage spikes caused by RF components' crosstalk after the initial program loading, 1000uF decoupling capacitors were added on both the VDD_DIG and VDD_RF rails. The VCO operates at a center frequency of 2.48 GHz, and there is a potential for transmission mode drift within a range of a few hundred kHz. It is worth noting that we observed the ambient temperature in the test environment affecting the VCO's drift. To ensure precision prior to conducting tests, we perform calibration on the center frequency, making necessary adjustments to the internal registers to achieve a stable transmission at 2.48 GHz, with a maximum output energy of -48.37 dBm shown in Figure 4.1.3.2. The TX output is connected to the SMB port, which, in turn, interfaces with the antenna for transmitting signals to the phone. The master SoC on the motherboard controls the TX through the SPI interface. The TX can be reset via an asynchronous active high reset.

In order to transmit the PPG data to a phone, we decided to wirelessly transmit the recorded PPG data to a mobile device for ongoing data transfer to the cloud and further analysis. Bluetooth low-energy (BLE) was selected as the communication standard due to its relaxed specifications, low power consumption, and the proliferation of BLE in mobile devices, which simplifies connectivity to the node.

Bluetooth Low Energy is particularly advantageous for applications that involve intermittent or continuous transmission of small volumes of data, making it ideal for devices such as sensors that prioritize low power usage, which is crucial for IoT applications. BLE excels at accommodating numerous communication nodes with minimal latency demands, exceptionally low power consumption, and brief connection periods, all while delivering a comparable communication range to traditional Bluetooth more efficiently [17].

Given these considerations, the radio was designed to operate only as a BLE beacon (BLE compatible), eliminating the additional circuitry required in a standard BLE that supports different operational modes (BLE compliant). Since the sensor node only transmits data intermittently, the radio was designed to enter a sleep mode periodically to minimize the energy consumption. Under this operational mode, a short start-up time and a low-leakage current in sleep mode are essential to maximize energy efficiency. During the start-up period, the radio cannot transmit data, therefore the energy dissipated in the process represents an energy loss. The start-up time of most radios is predominantly determined by the crystal oscillator and radio frequency settling time; therefore, these specifications are critical in the design of heavily duty-cycled applications.

State-of-the-art crystal frequency references report start-up times of 460 µs [29] and as low as 200 µs [30]. These numbers represent a start-up energy consumption of 15.8 nJ and 36.7 nJ, respectively. For perspective, a BLE packet can take 200 µs for transmission. In our design, the 16 MHz crystal oscillator consists of a three-stage inverter chain to act as a negative resistance and compensate for losses in the quartz crystal. The circuit design was optimized for efficient boosting of the negative resistance to achieve faster start-up time and lower start-up energy. With these considerations, the achieved performance corresponds to a measured start-up time of 150 µs while consuming only 10.5 nJ.

The BLE transmitter sends the acquired data from the sensor in the form of non-connectable advertisements packets. A low noise LC local oscillator is used to allow for open-loop operation without the need for a phase/frequency-locked-loop. This method minimizes power consumption while complying with the required frequency stability. The data whitening and cyclic redundancy check (CRC), required by the BLE standard, are all integrated on-chip. The transmission rate of the BLE radio is defined by the PPG sampling rate. At the defined 7.7 Sa/s sampling rate, the BLE radio transmits 2.5 packets/s.

The BLE TX consumes an average power of 2.17mW to transmit a 368-bit advertisement packet in over a 600µs transmission period, leading to a total energy consumption of 3.5nJ per bit. Duty-cycling is implemented in between each successive transmission through power gating, achieving an average power consumption of 3.72µW (1.86× sleep power) when transmitting a BLE advertising message every 753ms [28].

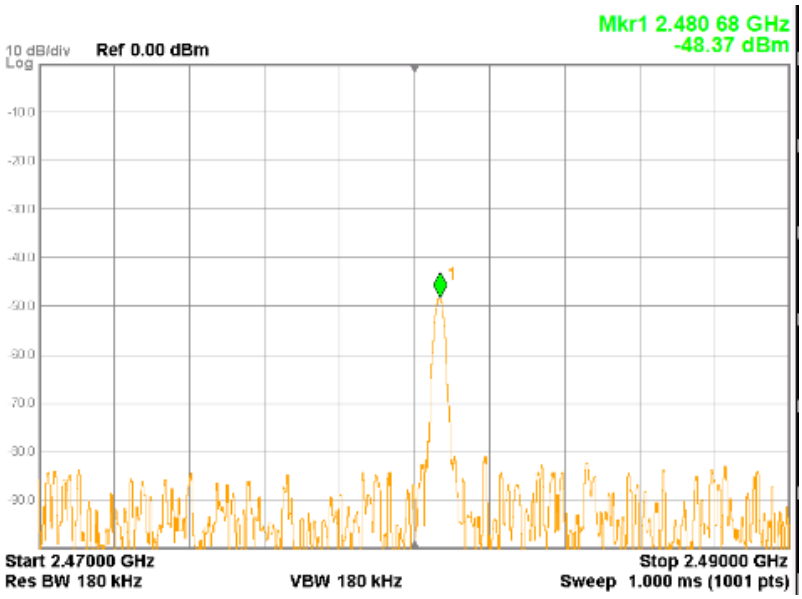


Figure 4.1.32: Measured spectrum of BLE transmitter with center frequency of 2.48 GHz with a maximum output power of -48.37 dBm.

4.5 Hardware: Energy Harvesting Processing Management Unit

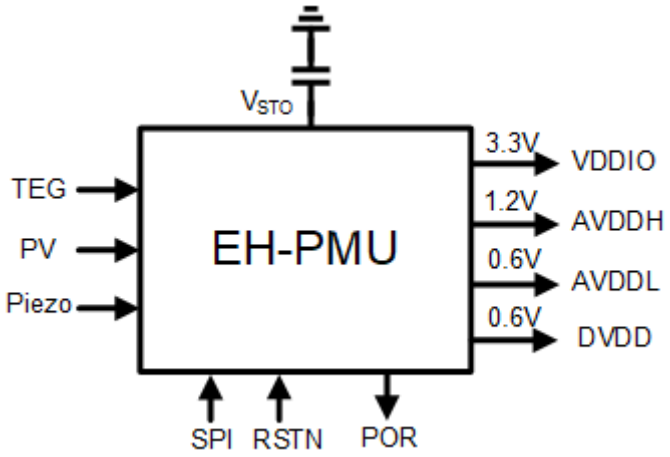


Figure 4.1.5: Energy harvesting management unit block diagram

The multiple-input-multiple-output (MIMO) energy harvester measurement unit was implemented on the baseboard, with the goal originally to have a fully self-powered IoT system. The system is shown as a block diagram in Figure 4.1.5, with the 4 output voltages supplying the rest of the system. However, we had trouble with getting the EHPMU to work reliably, therefore the power measurements for the integrated system did not include the EHPMU.

4.6 Operating Principle: Duty-Cycling

AFE uses a 6-bit SAR ADC. The LED driver works with a 3.3V supply. AFE uses 0.86V for the LTIA operations and 0.6V for the digital backend. To facilitate the acquisition and processing of PPG signals under conditions of limited energy harvesting power, a low- power approach is adopted involving the design and implementation of an ultra-low-power PPG analog front-end (AFE) system.

The PPG sensing signal chain comprises of an SFH7060 PPG sensor, which is followed by a chopper-stabilized programmable-gain transimpedance amplifier (PG-TIA), followed by a chopping-spike filter (CS Filter), and then into a 6-bit successive approximation register analog-

to-digital converter (SAR ADC). To mitigate potential saturation of the PPG signal within the PG-TIA output, an off-chip DC offset cancellation (DCOC) technique is employed using an external current source of +1uA and -2uA using a commercial off the shelf (COTs) LM334. Figure 4.2.3 presents an overview of the PPG AFE's architecture. A transimpedance amplifier (TIA) is employed to convert the AC current signal derived from the PPG sensor into an AC voltage signal. This voltage signal is then subsequently filtered through a low-pass filter to eliminate high-frequency noise. To ensure optimal performance, an input buffer is placed before the ADC to meet the necessary drive strength required to operate the capacitor DAC within the ADC module.

The overall system power is primarily dominated by the LED driver. As continuous operation mode draws on average 20 mW, it becomes necessary to lower the AFE power to meet the overall system power budget. Therefore, the PPG system employs aggressive duty-cycling to lower the overall power.

The AFE acquires the PPG signal at regular sampling intervals. Upon activation of the internal signal chain block, the Transimpedance Amplifier (TIA) module is engaged for a predetermined duration, as determined by the microprocessor's settings. The digital block of the AFE determines the duty-cycling rate of the PPG sensor.

The duty cycling rate in the RISCV program is expressed in milliseconds (ms). The sampling period is set by a 12-bit register, or a value of 4095. For instance, when the sampling period is set to 40.94ms with an input clock of 32KHz, a 20ms duty cycle occupies 48.852% of the sampling period, as indicated in Table I. Fig. 8(d) illustrates that the sampling period for the PPG is 13ms, which corresponds to a duty cycle rate of approximately 2.5%. Fig. 8(a) and Fig. 8(b) shows the measured PPG results when fully cycled in both the time domain and the frequency spectrum respectively.

To summarize, the system utilizes an aggressively duty-cycled scheme to duty-cycle the LED power into the AFE.

TABLE I: 40.95MS SAMPLING PERIOD DUTY-CYCLING

Duty-Cycle Settings (ms)	Duty-Cycle (%)
1	0.024426
5	0.12213
10	0.24426
20	0.48852
30	0.73278
40	0.97704

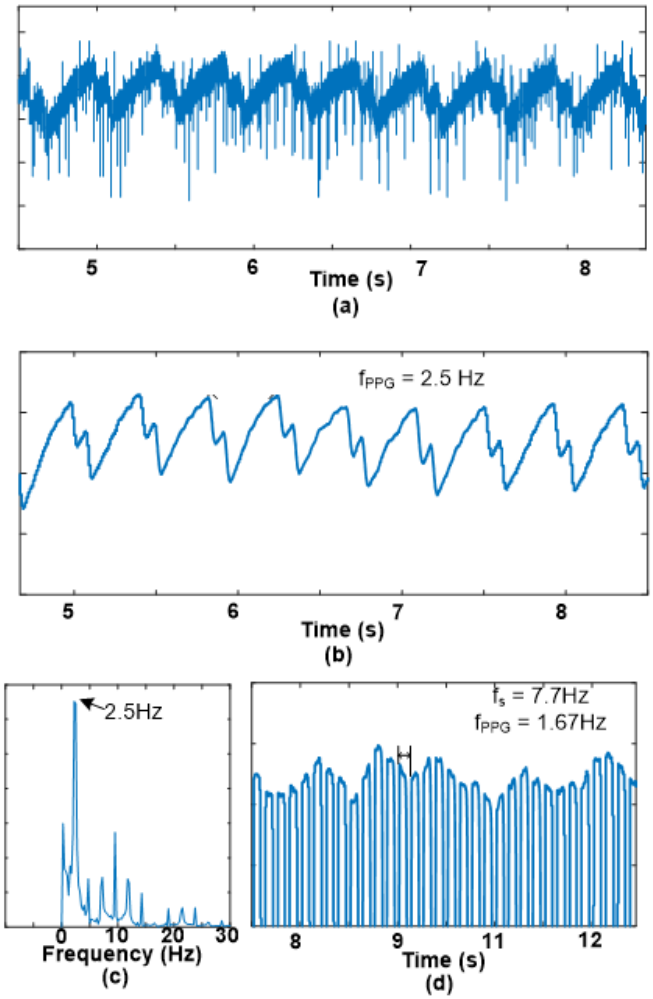


Figure 4.1.6: (a) Measured PPG signal in always-on mode, (b) (filtered), (c) and FFT of the measured PPG signal. (d) Measured PPG signal in duty-cycled mode (with sampling frequency fs = 7.7 Hz).

4.7 Software

The instruction set is based off the BottleRocket RISC-V core (RV32IMC) [16]. RISCV is an open-source instruction set developed at the University of Berkley in 2010. The RISCV core serves as the crucial link between the master SoC and the two AFE and TX slaves, facilitating communication through the SPI protocol. The RISCV program records the incoming AFE signals into the data register, establishing the TX sequence. This process continues until the data register is full, at which point the TX is activated, transmitting the data to the phone via BLE.

The program is described as follows.

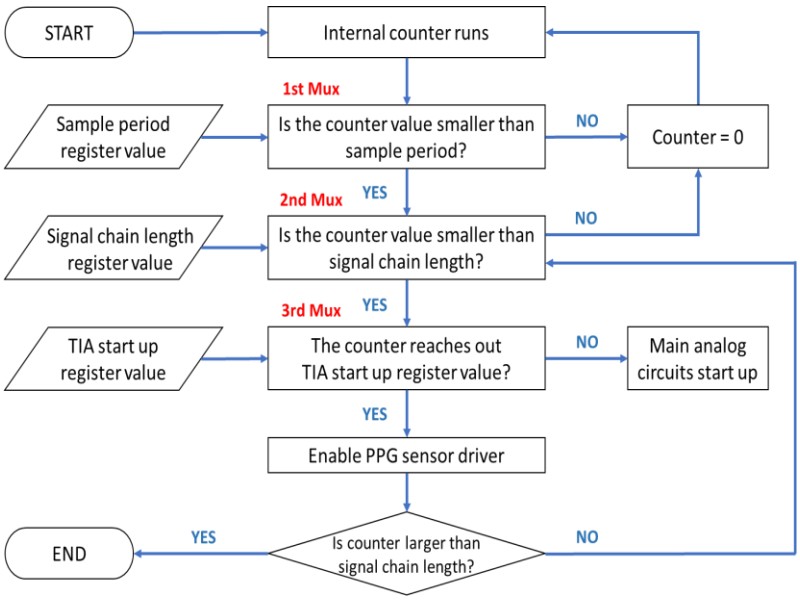


Figure 4.7.1: Flow chart of duty cycling sequences.

- The internal counter starts when the reset signal is called.
- The first mux selects the output based on the condition that the counter value is smaller than the set sample period. If the counter value is larger than the received sample period, the counter value will be zero.
- The second mux shows the output based on the condition that the counter value is smaller

than the set signal chain length register value. This signal chain length directly influences the duty cycle because this chain length determines the operation period of the internal analog circuit of AFE. If the counter value is larger than the received register values, the counter value will be zero.

- The third mux determines the PPG sensor operating frequency through the counter value of the counter module. This mux enables the PPG sensor if the counter value is larger than the set TIA startup register value. If the value is smaller than the TIA register, the LED sensor will not start operation, and the internal analog circuit will continuously work to be ready to receive PPG signals from the sensor.
- The final stage shows if the signal chain allows it to end. The signal chain operation is disabled when the counter value is larger than the set signal chain length value. In other words, the signal chain process will work until the counter reaches the register of signal chain length.

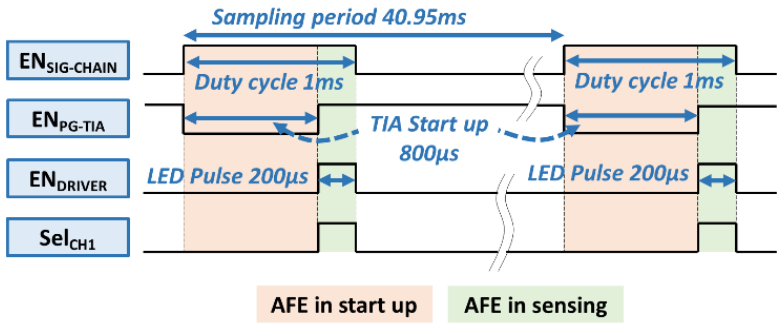


Figure 4.7.2: Timing diagram of the system operation

The system's duty cycle is described in Fig. 10. The following control registers are described in more detail.

- **Sampling period:** The AFE's period can be programmed from register control by communication to the microprocessor. This sampling period indicates the heartbeat frequency.
- **TIA startup register:** showing the startup time of the analog circuit before starting the LED sensor. This register is a programmable part for duty cycle operation, and it also has an

important role in deciding LED duty cycling. Because LED power consumption is dominant among the entire power

- **En driver**: This is presenting the LED driver enable period. There is no dedicated register for the period of the LED driver. However, this enable time is able to be programmed through arithmetic operation with the TIA startup register setting.
- **Sel CH1:** The AFE supports two channel interfaces separately; to serve this function, the digital circuit has a mux to select between channel 1 and channel 2 interfaces. The proposed integration system drives only a green LED sensor; hence, the digital controller selects channel 1 only.

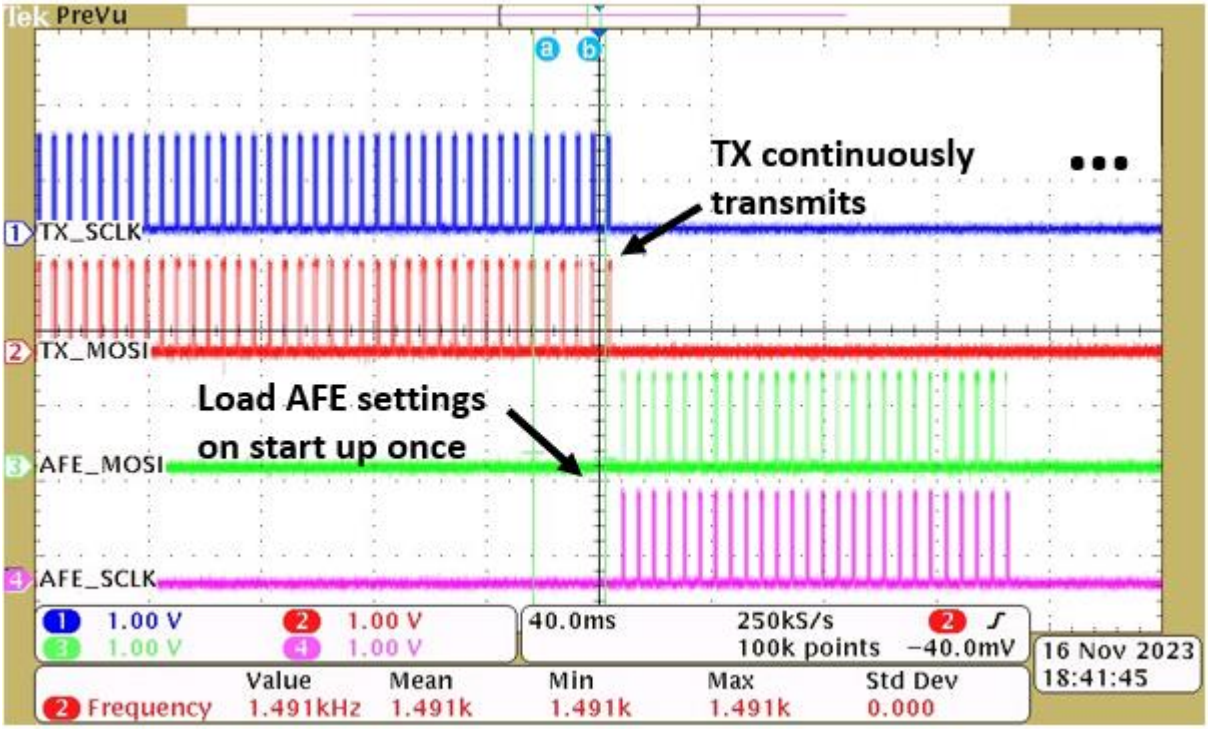


Figure 4.7.3: SPI outputs from the AFE and TX

Figure 4.7.3 illustrates the system's operation, where the System-on-Chip (SoC) transmits data to the Analog Front End (AFE) through AFE_MOSI on system startup. A single packet is dispatched to configure the AFE settings, configuring the sampling rate, duty-cycling rate, etc.

Subsequently, the AFE consistently delivers the output from the ADC via GPIO to the transmitter (TX). The TX then transmits the data packets. Figure 4.7.4 shows the outputs from the positive and negative outputs from the TIA (TIP_P and TIA_N respectively). Refer to Figure 4.2.3 for reference as to where the TIA output is in the digital stack. The quantized ADC output is shown below in pink.

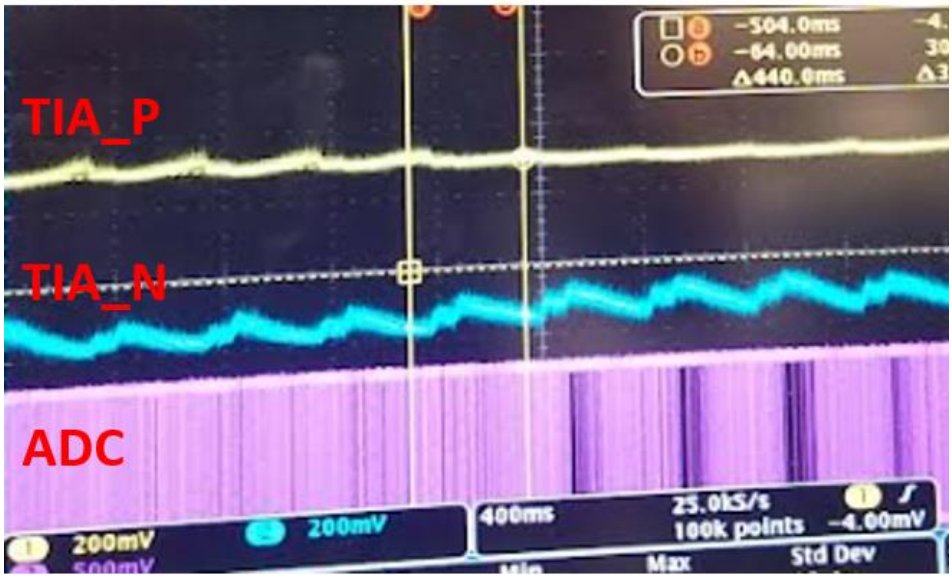


Figure 4.7.4: AFE outputs

Chapter 5
Measurement Results and Discussion

The proposed system is implemented across three chips using TSMC 65nM LP CMOS process. The SoC is used to process the information and save to SRAM, with a core area of 1.56 x 1.95

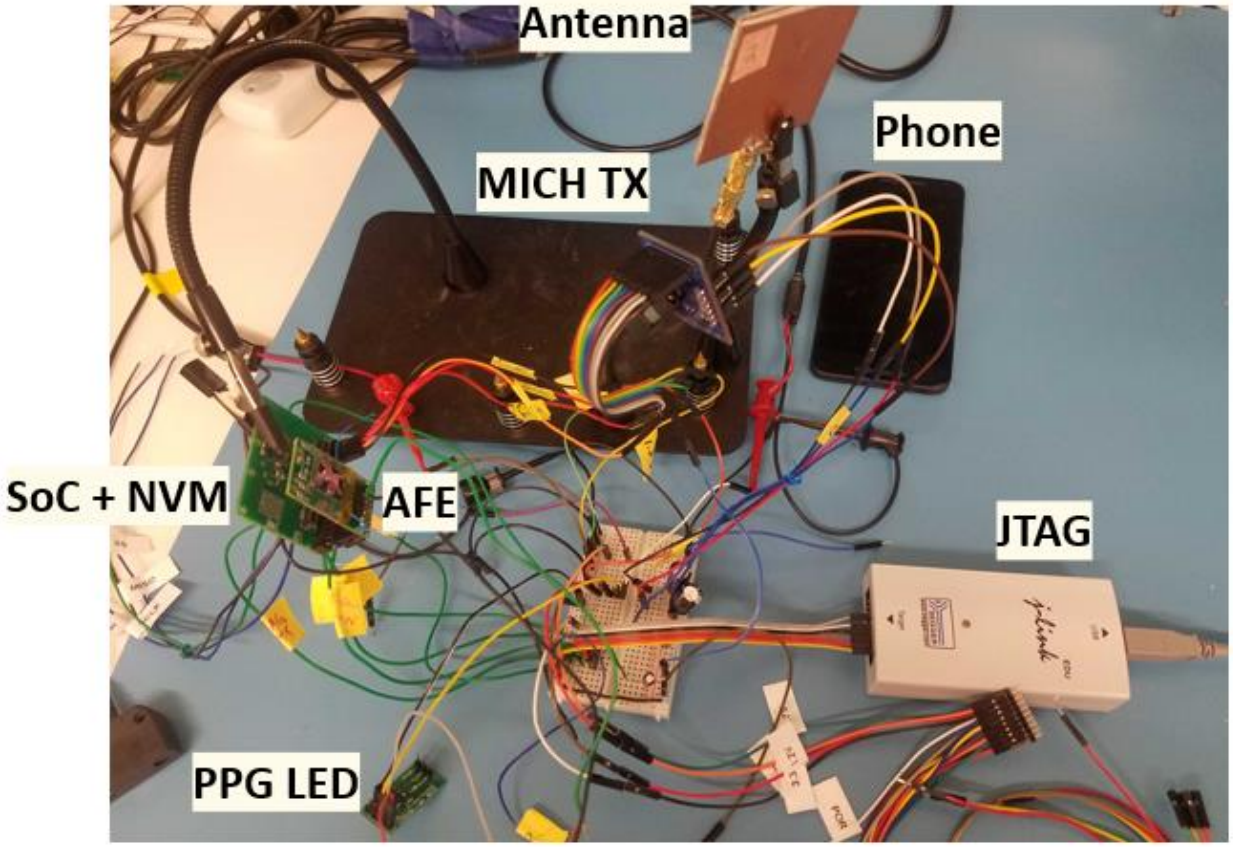


Figure 5.1: Benchtop Test setup

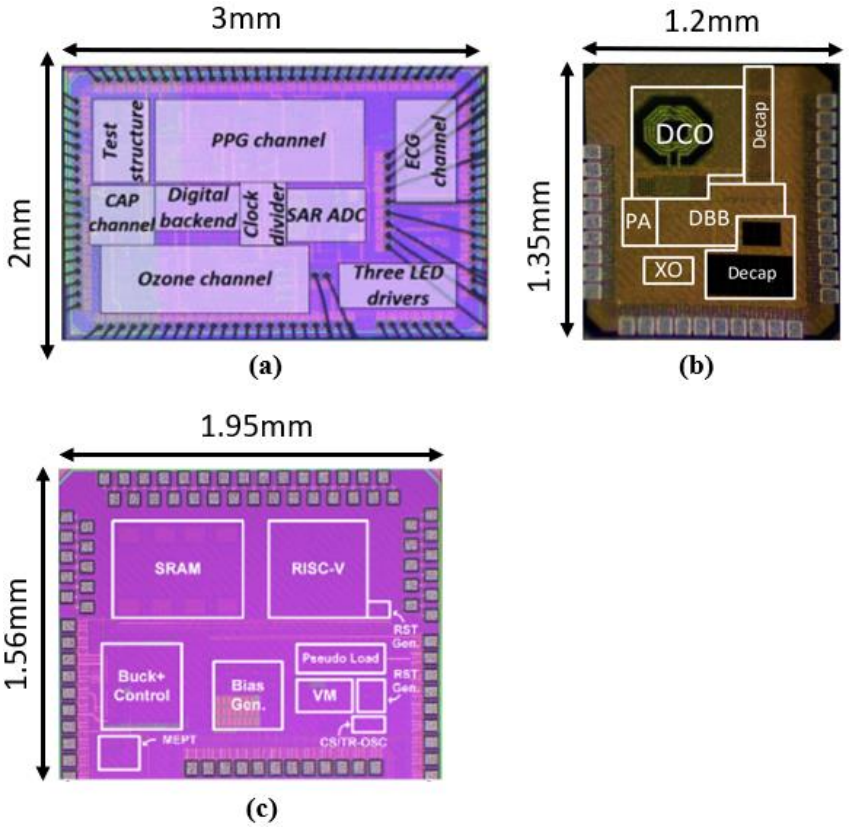


Figure 5.2: Microphotographs of the low-power chips fabricated in 65nm CMOS process: (a) AFE, (b) BLE Transmitter, RISC-V core SoC.

mm^2 bonded in QFN 100 package. The TX chip includes a Bluetooth-low energy transmitter implemented in 1.2 x 1.35 mm^2. A die photo of the custom BLE transmitter highlighting the power amplifier (PA), digital baseband (DBB), crystal oscillator (XO), and digitally controlled oscillator (DCO) is shown in Figure 5.2(b). The microphotographs of these three chips are shown in Figure 5.2.

5.1 Power Measurements

The overall power consumption of the system is shown with the full power breakdown by voltage rails is given in Figure 5.1.1. The benchtop setup is as shown the minimal achievable active power is 148.5 µW when duty-cycled at 1ms. The minimal achievable idle power is achieved when the system is duty-cycled down to 1ms at 62.82 µW. The overall specifications for the fully-integrated PPG system is shown in Table 5.1, breaking down the power

consumption for each component, voltages rails, and other general specifications.

TABLE 5.1: SUMMARY SPECIFICATIONS OF THE PROPOSED SYSTEM

General Specifications		
Chip Area	1.56mm x 195mm (SoC) 3mm x 2mm (AFE) 1.2mm x 1.35mm (TX)	
Technology	TSMV 65nm LP CMOS	
Supply Voltage	3.7V (Battery) 0.6V, 1.2V, 3.3V (chip ESD)	
AFE	Power Consumption	74.3uW (Active 2.5% duty-cycle) 45.83uW (Idle 1ms duty-cycle)
	Voltage	0.3V, 0.6V, 0.8V, 0.845V, 1.2V
	Sampling Frequency	7.7 Hz
	Duty-Cycling	2.5%
SoC	Power Consumption	8.13uW Active 8.13uW Idle
	Voltage	0.6V, 0.7V, 0.8V, 1.2V
	Memory	32 Kbit non-volatile
TX	Power Consumption	~2.17mW (full)
	Voltage	1.2
	VCO Tuning Range	2.4 – 2.4835GHz
	Output Power	-48.37 dBm
	Duty Cycle	2.5
	Energy per bit	3.5 nJ
	Transmission Rate	2.5 packets/s

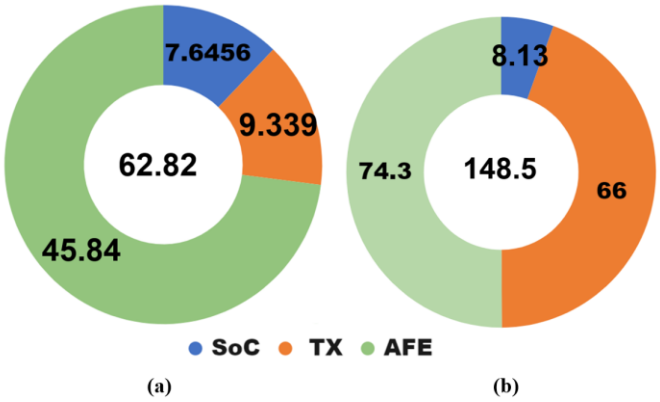


Figure 5.1.1: Pie-chart of the power breakdown for the three boards- SoC, AFE (includes the LED driver board), and TX. The center value shows the cumulate system power. (a) shows the idle system power when duty-cycled at 1ms (2.5%) in units of µW. (b) shows the active system when duty-cycled at 1ms (2.5%) in units of µW.

Figure 5.1.2. shows the received data packages to the android phone from the TX. Overlaid is the output from a low pass filter of the signal using MATLAB's default filter options. System measurements were done with subject's thumb pressed on PPG sensor, as shown in Figure 5.1.3. The Figure 5.1.4. shows the power consumption of the system over time for 10 ms [Figure 5.1.4(a)] and 1 ms [Figure 5.1.4(b)]. Measurements at different duty-cycles are shown. A summary of the overall power consumption with different duty cycles are shown in Figure 5.1.5.

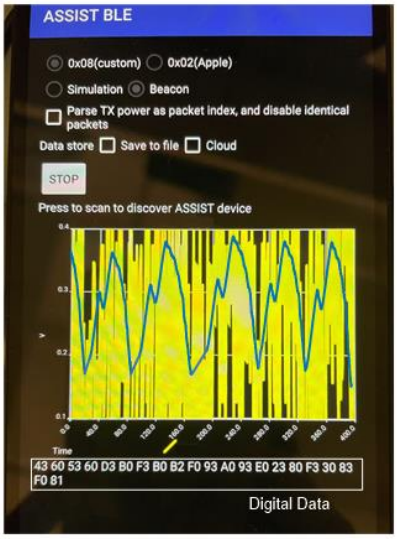


Figure 5.1.2: Screenshot of PPG data received via Bluetooth Low energy with a phone and signal after filtered through a low-pass filter.

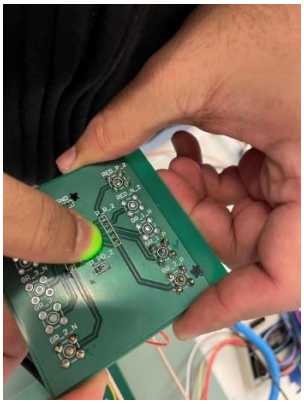


Figure 5.1.3: Revision 1 LED board measuring PPG signal via a green LED.

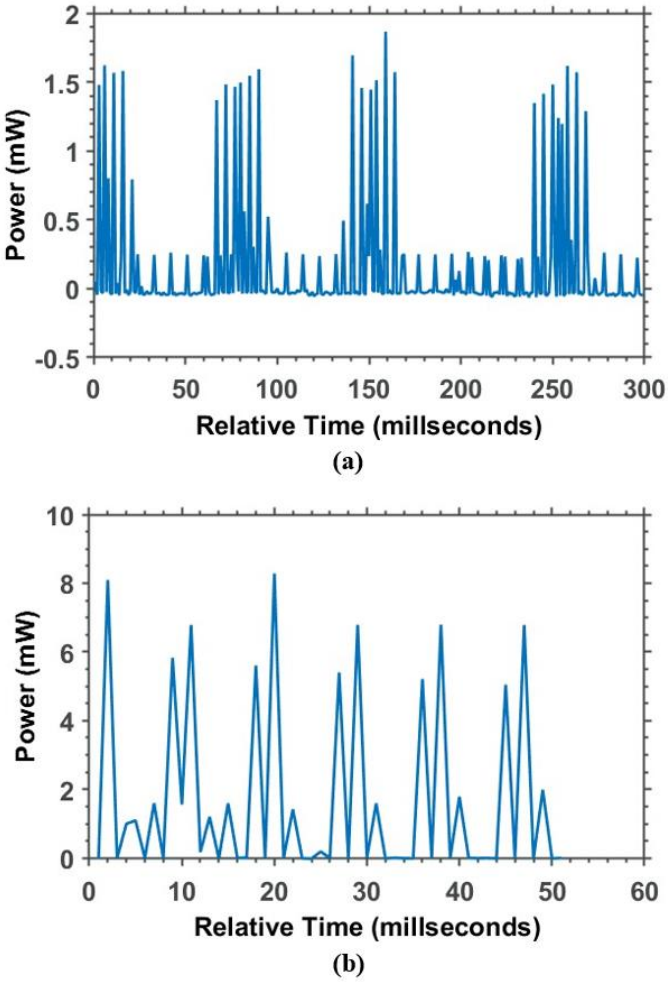


Figure 5.1.4: Power vs. time for a.) 10 ms duty-cycle and b.) 1ms duty-cycle.

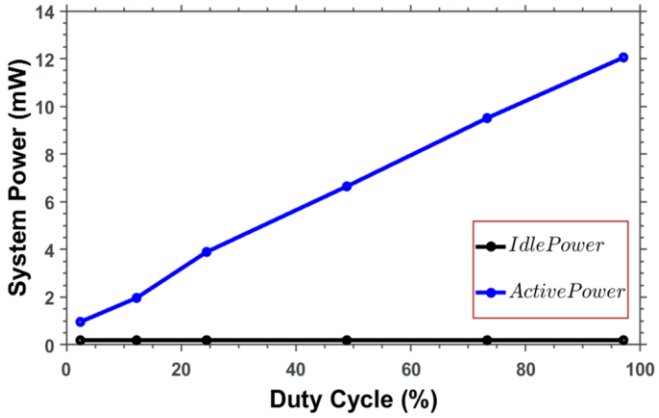


Figure 5.1.5: Measured average system power plotted as a function of duty cycle.

5.2 Comparison to State of the Art

Integrated cutting-edge wearable end-to-end PPG systems are defined by three benchmarks: low power usage, data transmission to the end user, as well as small-form factor. For a fair comparison, the related works for the PPG system must incorporate an Analog Front End (AFE), a processor, data storage, and a method for data transmission. These metrics serve as the foundation for defining 13 parameters, which are detailed in Table 5.2, highlighting comparison with literatures [18]-[26], [33]. In direct comparison with other systems, our work has resulted in excellent energy efficiency, with a power consumption of only 62.82 µW during idle periods and 148.5 µW during active phases when utilizing a 2.5% duty cycle.

Table 5.2: State of the Art Comparison

	This Work	**[18] (2022)**	**[19] (2020)**	**[20] (2021)**	**[21] (2022)**	**[22] (2019)**	**[23] (2021)**	**[24] (2019)**	**[25] (2017)**	**[26] (2017)**	**[33] (2023)**
Tech (nm)	65	180	CoTs	CoTs	CoTs	55	CoTs	CoTs	130	CoTs	CoTs
Supply Voltage (V)	1.2	3	5	3.7	3.7	1.2	3.3-5	3.3	3.7	3.3	4.2
Standby Power (µW)	62.82	-	23	-	4000	-	-	289	-	-	-
Active Power (µW)	148.5	510	18030	-	5000	2000	-	3426	5543	28980	18200
Readout Architecture	TIA	TIA	-	-	-	TIA	-	-	TIA	-	-
ADC (bits)	6b SAR	12b	12b	-	24b	13b OS SAR	10b	-	14b SAR	-	24b ΔΣ
FS_{PPG} (Sa/s)	7.7	125	-	-	125	128	125	256	100	-	hundreds
SNR (dB)	40	-	-	-	-	78.8	-	-	>80	-	-
Multimodal Sensing	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Communication Protocol	BLE beacon	BLE	BLE/ Wi-Fi	BLE	USB	BLE	BT	Audio	BLE	SD	BLE
Duty-Cycle (%)	2.5	100	100	-	100	5-10	100	100	1	100	100
Power Source	Battery (147 days)	Wired	Battery (21.98 hrs)	Battery	Battery (30 hrs)	Battery (3 weeks)	Wired	Phone jack (Self-powered)	Battery (5 days)	Solar (1-hour charge/ 3.4 hrs runtime)	Battery (15 hrs)
Dimension (mm x mm x mm)	75 x 70 x 35	-	42 x 40 x n/a	33 x 22.8 x n/a	27.5 x 27.5 x 16	-	-	-	34 x 40 x n/a	>60 x 72 x n/a	16 x 21 x 14

5.3 Discussion on Modular Design Approach

The modular system design method provides great flexibility as it treats each component of the IoT (refer to Figure 1.1.1) as separate blocks as part of one system. With our integration efforts, we went through 2 board revisions and 2 respins for the SoC, 2 chip bond-outs for the AFE, 6 board revisions for the AFE, 2 board revisions for the Michigan BLE-TX This enables testing only parts of the integrated system testing while anticipating the arrival of new revisions. One example we encountered during our design was that the AFE required a new revision to bond out parallel ADC outputs and combining the DVDD and AVDDL rails into one rail driven at 0.6V. Simultaneously, the SoC necessitated additional connections to GPIO pads, with AVDDL requiring 0.7V and DVDD needing 0.8V. In a conventional monolithic testing scenario, one would have had to wait for the new bonded AFE chip to arrive and redo the SoC PCB to continue. However, the modular system allows for independent testing of the AFE with TX or the SoC with TX.

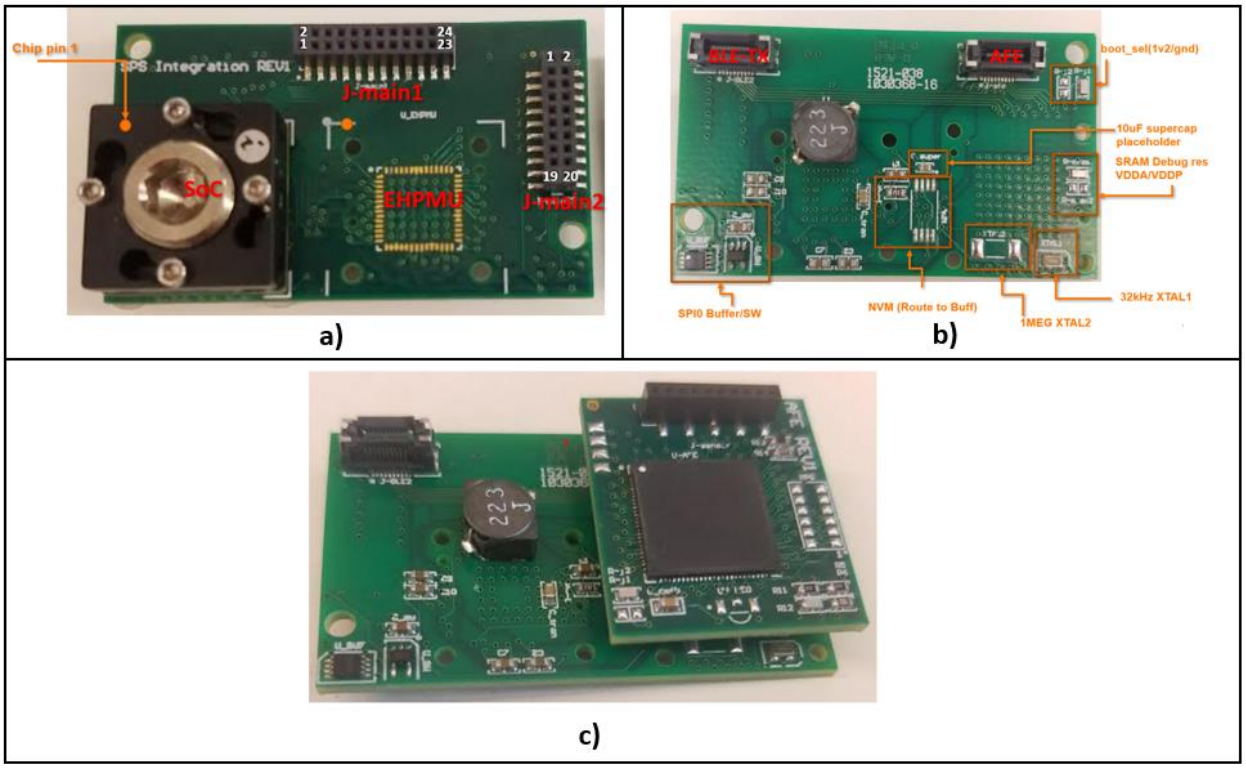


Figure 5.3: Revision 1 board. a) Top-view of the SoC baseboard, b) Bottom-view of the SoC baseboard, c) Integrated system view.

In another instance, Revision 1 of the SoC baseboard, referred to in Figure 5.3, faced a drive strength inadequacy, necessitating the addition of an external buffer for effective communication with the NVM over SPI. The board-to-board connector (Molex 2039550203) had issues with signal integrity and mechanical robustness when connected for integrated testing. Revision 2 removes the connector and replaces it with a regular 2.54mm pitch header. Furthermore, the SoC also required the DVDD rail to be separated as 0.8V rather than combined between the AVDDL and DVDD at 0.6V in rev.1 With the modular approach, as all of the boards are separate ecosystems, we can continue testing the AFE and BLE-TX daughterboard with the old rev. 1 SoC until the new rev 2 baseboard arrives. When rev2 arrives, it would just be quick test of AFE and BLE-TX as a matter of ‘plug-and-play’.

Chapter 6

Conclusions and Future Directions

In this work, we introduce a compact (75mm x 70mm x 35mm) end-to-end PPG system with three ultra-low power ASICs designed in the TSMC 65nm low power CMOS process: The System-on-Chip (SoC), Analog Front End (AFE), and Transmitter (TX). The PPG system demonstrates ultra-low power characteristics, with an idle power consumption of 62.82 µW and an active power consumption of 148.5 µW, while maintaining an average power measurement of 141.10 µW. These measurements show that with 3.7 V, 500-mAh CR3555 battery, the estimated operational lifespan of the system can be up to 147 hours when duty-cycled at 2.5%.

The system was developed using a modular system design process, where each chip was tested on separate PCBs modeled after a Raspberry PI hat, enabling very rapid 'plug-and-play' prototyping of different chips and PCB revisions to enable an integrated IoT system. This work will be submitted to the Journal of Biomedical and Health Informatics (JBHI) later in 2023 as part of the collaborative effort funded by the NSF NERC ASSIST Center.

To summarize, our system met the following requirements:

- Customized flexible RISC-V SoC for on-body PPG streaming for health applications, with an Analog Front End (AFE), data storage, and a BLE-TX for an end-to-end solution to the end-user.
- Low power of less than 100 µW of idle power and 148.5 µW of active power, which is less than the ~1mW minimum requirement to meet future self-powered energy goals when using our MIMO EHPMU.
- Common design language between the daughterboards for inter-operability similar to a Raspberry Pi hat.
- Small form-factor of (75mm x 70mm x 35mm) that can be easily realized for wearable health applications in a clinical setting, as per the requirements of ASSIST.

6.1 Self-Powered Energy Harvesting

The end-to-end solution proposed in this study not only demonstrates ultra-low power performance but also underscores the need for further enhancements to make this system truly wireless and wearable self-powered. A notable path for improvement lies in wearable energy harvesting, with one concept being wrist harvesting presented in [36]. One possible extension of our system inspired by this work, aimed at addressing the issue of battery depletion is by adding an additional multi-input single-inductor multi-output (MIMO) EHPMU [37] available on the SoC motherboard, optionally with wrist harvesting as an additional energy input node. As shown in Figure 4.1, the system has an onboard MIMO EHPMU chip for multi-modal energy harvesting that wasn't fully integrated in this work. Future integration efforts would allow us to eliminate the need for conventional batteries in the system and achieve fully self-power. The EHPMU achieves up to 1.2mW greatly exceeds the average power consumption of our system, which stands at 141.10 μW when operated when duty-cycled at a rate of 2.5%.

6.2 Improvements to LED Power

Our power measurements correlated with our prior works in [4] and found the LED power is a critical factor dominating wearable PPG systems, thereby warranting future exploration in areas such as level-crossing continuous-time ADC [35] for sparse signal sensing. As the LED power dominates the system power, having a system that can efficiently digitize weak signals in continuous time systems at lower-power threshold would be beneficial. In [35], a 16-level crossing ADC is roughly equivalent to a 4-bit conventional ADC while achieving better than 100 dB of signal-to-noise ratio for signal reconstruction with lower overall system power for sparse signals.

Additionally, adaptive LED algorithms catering to different skin tones should be incorporated based on the models in [6] demonstrating that skin phototype is the dominating factor in SNR, and therefore power. Furthermore, the potential of light-to-digital converter (LDC)-based PPG

[11] for reduced power consumption in the readout circuit, in contrast to traditional TIA circuits approaches could result in significant reduction in AFE readout power. Finally, [31] introduces a novel NO-LED PPG method that could significantly mitigate power consumption as it completely gets rid of the LED by optimizing the AFE for sensing from the photodiode in different ambient environments without requiring the LED to be turned on.

6.2 Deployable System for Clinical Tests

Moving forward, our research will progress to validate the effectiveness of these end-to-end low-power wearable PPG systems in a deployable clinical environment, with the ultimate goal of integrating them into real-world clinical trials. To accomplish this, rigid-flex PCBs and direct over molding techniques similar to [20] and [21] would be used to provide a more streamlined product that accounts for user-interfaces. Additionally, smaller more robust board-to-board connectors like the Hirose DF17s should be utilized for smaller board footprints. This approach is poised to address the prevailing challenges associated with battery life, physical design, and user comfort, which currently hinder the practicality of many healthcare wearables for extended health monitoring.

Appendix A

App C Code

App_1ms.c

```
#include "../../address_map.h"
//#include "../string.h"
void SPI3_WRITE(int command, int address, int data, int command_length, int address_length, int
data_length);
void SPI3_READ(int command, int address, int command_length, int address_length, int read_length);
void SPI1_WRITE(int command, int address, int data, int command_length, int address_length, int
data_length);
void SPI1_READ(int command, int address, int command_length, int address_length, int read_length);
void wait(int count);

int main(int argc, char **argv) {
		GPIO_PADDIR = 0x000000ff;//Bits 7-4 output, 3-0 input
		wait(5);
		//SPI3_CLKDIV = 0x00000100;
		//SPI1_CLKDIV = 0x00000100;
		//wait(5);
//TX
		SPI1_WRITE(0x01, 0x01, 0x3f01, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x02, 0xa8cf, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x03, 0x4571, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x04, 0x271b, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x05, 0x24db, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x06, 0xb137, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x07, 0xdde8, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x08, 0x0c15, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x09, 0x021f, 1, 15, 16);
		wait(5);
		SPI1_WRITE(0x01, 0x0a, 0x1bb5, 1, 15, 16);
```

```
        wait(5);
        SPI1_WRITE(0x01, 0x0b, 0x1c23, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x0c, 0x75d7, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x0d, 0xd75d, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x0e, 0x5d75, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x0f, 0x75d7, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x10, 0x9559, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x11, 0x10c4, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x12, 0xfd3c, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x13, 0xecb2, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x14, 0xf34b, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x15, 0xed3c, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x16, 0x6aae, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x17, 0xa69a, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x18, 0x2304, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x19, 0xd555, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1a, 0x5d75, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1b, 0x1304, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1c, 0x5451, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1d, 0x10c4, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1e, 0x9a69, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x1f, 0xabaa, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x20, 0xf34b, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x21, 0x9a69, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x22, 0x69a6, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x23, 0xa69a, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x24, 0x8f62, 1, 15, 16);
        wait(5);
        SPI1_WRITE(0x01, 0x25, 0xddc1, 1, 15, 16);
        wait(5);
```

```
SPI1_WRITE(0x01, 0x26, 0x40ff, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x27, 0xb789, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x28, 0xa506, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x29, 0xc277, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x2a, 0xfb80, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x2b, 0x76E0, 1, 15, 16);//center freq, 2a before
wait(5);
SPI1_WRITE(0x01, 0x2c, 0x8a48, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x2d, 0xdfc3, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x2e, 0x0703, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x2f, 0x021f, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x30, 0xa011, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x31, 0x6130, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x32, 0x0d08, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x33, 0x9759, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x34, 0xa87b, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x35, 0x7afb, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x36, 0xebc3, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x37, 0x55c3, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x38, 0x3b22, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x39, 0x9887, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3a, 0x10b3, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3b, 0xbac8, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3c, 0x9171, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3d, 0x6b7d, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3e, 0x0a02, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x3f, 0x0b3c, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x40, 0x3001, 1, 15, 16);
wait(5);
SPI1_WRITE(0x01, 0x41, 0x1000, 1, 15, 16);
```

```
            wait(5);
            SPI1_WRITE(0x01, 0x01, 0x3f03, 1, 15, 16);
            wait(5);
//PROGRAM AFE
            SPI3_WRITE(0x01,0x0,0x0050,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x2,0x2065,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x4,0x0064,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x5,0xa032,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x6,0x300a,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x7,0x86a0,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x8,0x1c8d,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x9,0x4e20,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xa,0xc014,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xb,0x00f7,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xc,0x2f0a,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xd,0xf0bc,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xe,0x03c2,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0xf,0x3c20,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x10,0x0060,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x3a,0x0202,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x1,0x3063,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x3,0x4ffe,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x32,0xf0b0,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x38,0x8790,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x2f,0x4800,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x39,0xf080,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x8,0x5c8d,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x19,0x0f7d,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x19,0x8f7d,1,7,16);
            wait(5);
            SPI3_WRITE(0x01,0x7c,0x0000,1,7,16);
```

```
		wait(5);

		/*while((GPIO_PADIN & 0x00000080) != 0x00000080){
			wait(5);
			}*/
		unsigned int DATAOUT = 0x00000000;
		int TEMPDATA[801];
		wait(2);
		int j = 0;
		int ADDR = 0x00000002;
		wait(2);
		int INDEXTX = 0x00000000;
		wait(2);
		int TEMPTX = 0x00000000;
		wait(2);
		int FLAG =0;
		wait(2);
		while(1){
			for (int i = 0; i <= 801; i = i+1){
				TEMPDATA[i] = GPIO_PADIN & 0x000000FF;
				wait(1);
			}

			for (int j =0; j <= 801; j = j+2){
				//if (FLAG == 0){
				//	if((TEMPDATA[j]<<2) > 0xAA || TEMPDATA[j+1]<<2 > 0xAA){   //
threshold
				//			FLAG = 1;
				//	}
				//}
				//DATAOUT = ((TEMPDATA[j]<<10) | (TEMPDATA[j+1]<<2)) +
0x00003939;
				DATAOUT = TEMPDATA[j];
				wait(1);
				SPI1_WRITE(0x01, ADDR, DATAOUT, 1, 15, 16);
				wait(4);
				if (ADDR == 0x0000000B){
					ADDR = 0x00000002;//reset the address
					//send the data first time
					wait(2);
					TEMPTX = (INDEXTX << 8) | 0x00000003;//
					wait(5);
					SPI1_WRITE(0x01, 0x01, TEMPTX, 1, 15, 16);
					wait(5);
					for(int i = 0; i < 20 ; i = i+1){
					SPI1_WRITE(0x01, 0x42, 0x0, 1, 15, 16);
					wait(5);
					}

					//send the data sencond time
					TEMPTX = (INDEXTX << 8) | 0x00000001;
					wait(5);
					SPI1_WRITE(0x01, 0x01, TEMPTX, 1, 15, 16);
					wait(5);
					for(int j = 0; j < 20 ; j = j+1){
```

```
                                        SPI1_WRITE(0x01, 0x42, 0x0, 1, 15, 16);
                                        wait(5);
                                }
                                if(FLAG == 1){
                                        wait(2);
                                        TEMPTX = (INDEXTX << 8) | 0x00000003;//
                                        wait(5);
                                        SPI1_WRITE(0x01, 0x01, TEMPTX, 1, 15, 16);
                                        wait(5);
                                        for(int i = 0; i < 20 ; i = i+1){
                                                SPI1_WRITE(0x01, 0x42, 0x0, 1, 15, 16);
                                                wait(5);
                                        }

                                        //send the data sencond time
                                        TEMPTX = (INDEXTX << 8) | 0x00000001;
                                        wait(5);
                                        SPI1_WRITE(0x01, 0x01, TEMPTX, 1, 15, 16);
                                        wait(5);
                                        for(int j = 0; j < 20 ; j = j+1){
                                                SPI1_WRITE(0x01, 0x42, 0x0, 1, 15, 16);
                                                wait(5);
                                        }
                                        FLAG == 0;
                                        wait(1);
                                }
                                INDEXTX = INDEXTX + 1;
                                wait(2);
                                if (INDEXTX == 0x00000100)
                                {
                                        INDEXTX = 0x00000000;
                                        wait(2);
                                }
                        }
                        else {
                                ADDR = ADDR + 1;
                                wait(2);
                        }


                }
        }
        /*while((SPI3_RXFIFO & 0x00008000) != 0x00008000){
                SPI3_READ(0x00, 0x43, 1, 7, 16);
                wait(5);
                } */


        return 0;
}
void SPI3_WRITE(int command, int address, int data, int command_length, int address_length, int
data_length){
  SPI3_SPICMD = command<<(32-command_length);
  SPI3_SPIADR = address<<(32-address_length);
  SPI3_TXFIFO = data<<(32-data_length);
```

```
   SPI3_SPILEN = (command_length & 0x0000003F) | ((address_length<<8)  & 0x00003F00) |
((data_length<<16) & 0xFFFF0000);
   SPI3_STATUS = 0x00000102;
   return;
}

void SPI3_READ(int command, int address, int command_length, int address_length, int read_length){
   SPI3_SPICMD = command<<(32-command_length);
   SPI3_SPIADR = address<<(32-address_length);
   SPI3_SPILEN = (command_length & 0x0000003F) | ((address_length<<8)  & 0x00003F00) |
((read_length<<16) & 0xFFFF0000);
   SPI3_STATUS = 0x00000101;
   return;
}

void SPI1_WRITE(int command, int address, int data, int command_length, int address_length, int
data_length){
   SPI1_SPICMD = command<<(32-command_length);
   SPI1_SPIADR = address<<(32-address_length);
   SPI1_TXFIFO = data<<(32-data_length);
   SPI1_SPILEN = (command_length & 0x0000003F) | ((address_length<<8)  & 0x00003F00) |
((data_length<<16) & 0xFFFF0000);
   SPI1_STATUS = 0x00000102;
   return;
}


void SPI1_READ(int command, int address, int command_length, int address_length, int read_length){
   SPI1_SPICMD = command<<(32-command_length);
   SPI1_SPIADR = address<<(32-address_length);
   SPI1_SPILEN = (command_length & 0x0000003F) | ((address_length<<8)  & 0x00003F00) |
((read_length<<16) & 0xFFFF0000);
   SPI1_STATUS = 0x00000101;
   return;
}


void wait(int count){
   int counter = 0;
   while(counter <= count){
      counter = counter + 1;
   }
}
```

Appendix C
Matlab

PowerOverTime.m

```
% Peter Le
% Power over time overview measurements
% Link to plot https://www.mathworks.com/matlabcentral/fileexchange/100766-
professional-plots

clear; clc; close all;
PS = PLOT_STANDARDS();
% Specify the CSV file paths
csvFilePath4 = '1ms.csv';
csvFilePath5 = '10ms.csv';
fontSizeSetting = 18; %axis font size

LDOVoltage = 0.281;

% Specify the starting rows for reading the data
startingRow = 44893;  % TX-1MS.csv Use this value, buffer started reading good data
from this row **DO NOT CHANGE
startingRow2 = 10;     % volt28.csv  **DO NOT CHANGE
startingRow3 = 10;     % volt0845.csv   **DO NOT CHANGE
startingRow4 = 10;
startingRow5 = 10;

% Read the data from the CSV files
data4 = readmatrix(csvFilePath4);
data5 = readmatrix(csvFilePath5);

% Extract current (in uA) and relative time (in seconds) from the data starting from
the specified rows
% DO NOT MODIFY THIS.


current_uA_4 = (data4(startingRow4:startingRow4+298, 1)- LDOVoltage)*1e6 ;
relative_time_sec_4 = data4(startingRow4:startingRow4+298, 22);

current_uA_5 = (data5(startingRow5:startingRow5+50, 1)- LDOVoltage)*1e6 ;
relative_time_sec_5 = data5(startingRow5:startingRow5+50, 22);



% Calculates the average power over given time duration %%%%%%%%%%
% Calculate the power for each data point in 1ms.csv
%power_mW_4 = current_uA_4 .* LDOVoltage / 1000;
power_mW_4 = current_uA_4;

% Calculate the power for each data point in 10ms.csv
```

```
%power_mW_5 = current_uA_5 .* LDOVoltage / 1000;E
power_mW_5 = current_uA_5;

% Specify the time durations (in seconds) for averaging
duration_4 = relative_time_sec_4(end) - relative_time_sec_4(1);
duration_5 = relative_time_sec_5(end) - relative_time_sec_5(1);

% Calculate the average power for 1ms.csv
average_power_4 = trapz(relative_time_sec_4, power_mW_4) / duration_4;

% Calculate the average power for 10ms.csv
average_power_5 = trapz(relative_time_sec_5, power_mW_5) / duration_5;

% Print the average power to the console in uW
fprintf('Average Power for 1ms.csv: %.2f uW\n', average_power_4);
fprintf('Average Power for 10ms.csv: %.2f uW\n', average_power_5);

%prints average power when TX is transmitted only
%Above 1mW threshold, it will average
if average_power_4 > 1
    fprintf('Average Power for 1ms.csv for TX transmission: %.2f uW\n',
average_power_4);
end

if average_power_5 > 1
    fprintf('Average Power for 10ms.csv for TX transmission: %.2f uW\n',
average_power_5);
end




% Plot the data for 1ms.csv  %%%%%%%%%%%%%%%%%%%%%%
figure(1);
fig1_comps.fig = gcf;
hold on


plot(relative_time_sec_4, current_uA_4/1000, 'LineWidth', 2);
hold off
xl1= xlabel('Relative Time (millseconds)');
yl1 = ylabel('Power (mW)');
%title('Power over Time (1ms rail)');



STANDARDIZE_FIGURE(fig1_comps);

% Plot the data for 10ms.csv  %%%%%%%%%%%%%%%%%%%%%%
figure(2);
fig2_comps.fig = gcf;
hold on

plot(relative_time_sec_5, current_uA_5/1000, 'LineWidth', 2);
```

```
xl2 = xlabel('Relative Time (millseconds)', 'FontSize', 10);
yl2 = ylabel('Power (mW)', 'FontSize', 10);
%title('Power over Time (10ms rail)');

STANDARDIZE_FIGURE(fig2_comps);

xl1.FontSize = fontSizeSetting;
yl1.FontSize = fontSizeSetting;
xl2.FontSize = fontSizeSetting;
yl2.FontSize = fontSizeSetting;


%=========================================================
% SAVE FIGURE AS AN IMAGE
%SAVE_MY_FIGURE(fig1_comps, '1ms_plot.png', '1ms');
%SAVE_MY_FIGURE(fig2_comps, '10ms_plot.png', '10ms');

figure_resolution = 600;
fig_filename = '1ms_plot.png';
exportgraphics(fig1_comps.fig, fig_filename, 'Resolution', figure_resolution);

figure_resolution = 600;
fig_filename = '10ms_plot.png';
exportgraphics(fig2_comps.fig, fig_filename, 'Resolution', figure_resolution);
```

Glossary

Acronyms and Abbreviations

PPG	Photoplethysmography
ADC	Analog-to-digital converter
AFE	Analog front-end
ASIC	Application-specific integrated circuit
AVDDH	Analog Voltage-High
AVDDL	Analog Voltage-Low
BLE	Bluetooth Low-Energy
BT	Bluetooth
CLK	Clock
CMOS	Complementary metal–oxide–semiconductor
COTS	Commercial off-the-shelf
DAC	Digital to analog converter
DCOC	DC offset cancellation
DVDD	Digital Voltage
DVFS	Dynamic Voltage Frequency Scaling
ECG	Electrocardiogram
EHPMU	Energy harvesting processing management unit
ENOB	Effective number of bits
ESD	Electrostatic discharge
FFL	Frequency-locked loop
FFT	Fast Fourier Transform
GPIO	General Purpose Input/output
IO	Input/output
IOUT	Output current
JTAG	Joint Test Action Group
LC	Inductor-Capacitor oscillator
LDC	light-to-digital converter
LDO	Low-dropout
LED	Light emitting diode
LP	Low-power
LTIA	Logarithmic transimpedance amplifier

MCU	Microcontroller
MIMO	Master in Slave Out
MOSI	Master Out Slave In
MPPT	Maximum Power Point Tracking
NVM	Nonvolatile Memory
OS	Operating system
OSC	Oscillator
PCB	Printed circuit board
POR	Power-on-reset
QFN	Quad-flat no-leads
RF	Radio frequency
RISCV	Reduced Instruction Set Computer V
ROM	Read-only memory
RX	Receive
SAR	Successive-approximation-register
SMB	Subminiature version B
SNR	Signal-to-noise ratio
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
TIA	Transimpedance amplifier
TSMC	Taiwan Semiconductor Manufacturing Company
TX	Transmit
ULP	Ultra-low power
VCO	Voltage-controlled oscillator
VDD	Voltage
VDDIO	Voltage Input/output
VDDL	Voltage-Low
XO	Clock oscillator

Bibliography

[1] J. Allen, "Photoplethysmography and its application in clinical physiological measurement", Physiol. Meas., vol. 28, no. 3, pp. R1-R39, Mar. 2007.

[2] V. Hartmann, H. Liu, F. Chen, Q. Qiu, S. Hughes, and D. Zheng, “Quantitative comparison of photoplethysmographic waveform characteristics: Effect of measurement site,” Front Physiol, vol. 10, p. 198, Mar 2019.

[3] Castaneda D, Esparza A, Ghamari M, Soltanpur C, Nazeran H. A review on wearable photoplethysmography sensors and their potential future applications in health care. Int J Biosens Bioelectron. 2018;4(4):195-202. doi: 10.15406/ijbsbe.2018.04.00125. Epub 2018 Aug 6. PMID: 30906922; PMCID: PMC6426305.

[4] Wang, Peng. Ultra-Low Power Multi-Modal Sensor Interface Circuits and Systems for Personalized Physiological Monitoring. University of Virginia, Electrical Engineering - School of Engineering and Applied Science, PHD (Doctor of Philosophy), 2021, 2021, doi.org/10.18130/qm7h-hn73.

[5] “Osram-SFH 7050. (2014). [Online]. Available: https://dammedia. osram.info/media/resource/hires/osramdam2496553/sfh%207050%20-%20photoplethysmography%20sensor.pdf.”

[6] K. A. Flynn, N. B. Ownby, P. Wang and B. H. Calhoun, "Modeling Energy-Aware Photoplethysmography Hardware for Personalized Health Care Applications Across Skin Phototypes," 2021 IEEE Biomedical Circuits and Systems Conference (BioCAS), Berlin, Germany, 2021, pp. 01-06, doi: 10.1109/BioCAS49922.2021.9644946.

[7] H. L. Bishop, P. Wang and B. H. Calhoun, "Application-Driven Model of a PPG Sensing Modality for the Informed Design of Self-Powered, Wearable Healthcare Systems," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180900.

[8] P. Wang and B. H. Calhoun, "A Photoplethysmography Analog Front-End Model for Rapid Design of Personalized Healthcare Hardware," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 1145-1149, doi: 10.1109/ISCAS48785.2022.9937734.

[9] S. Mohsen, A. Zekry, K. Youssef, and M. Abouelatta, “A Self-Powered Wearable Wireless Sensor System Powered by a Hybrid Energy Harvester for Healthcare Applications,” Wireless Personal Communications, vol. 116, no. 4, pp. 3143-3164, September 2020

[10] N. M. H. Long and W. -Y. Chung, "Wearable Wrist Photoplethysmography for Optimal Monitoring of Vital Signs: A Unified Perspective on Pulse Waveforms," in IEEE Photonics Journal, vol. 14, no. 2, pp. 1-17, April 2022, Art no. 3717717, doi: 10.1109/JPHOT.2022.3153506.

[11] Z. Ebrahimi and B. Gosselin, "Ultra-Low Power Photoplethysmography (PPG) Sensors: A Methodological Review," in IEEE Sensors Journal, doi: 10.1109/JSEN.2023.3284818.

[12] “ADPD4100/ADPD4101 Multimodal Sensor Front End.”

[13] D. S. Truesdell, J. Breiholz, S. Kamineni, N. Liu, A. Magyar and B. H. Calhoun, "A 6–140-nW 11 Hz–8.2-kHz DVFS RISC-V Microprocessor Using Scalable Dynamic Leakage-Suppression Logic," in IEEE Solid-State Circuits Letters, vol. 2, no. 8, pp. 57-60, Aug. 2019, doi: 10.1109/LSSC.2019.2938897.

[14] Flynn, Katheryn. Modeling Applications for Informed Self-Powered System Design. University of Virginia, Electrical Engineering - School of Engineering and Applied Science, MS (Master of Science), 2023, 2023, doi.org/10.18130/sbn9-np69.

[15] Yu Song, Daniel Mukasa, Haixia Zhang, and Wei Gao Accounts of Materials Research 2021 2 (3), 184-197 DOI: 10.1021/accountsmr.1c00002

[16] RISC-V foundation: Instruction set architecture (ISA), 2019, [online] Available: https://riscv.org/.

[17] Chaari Fourati, L., Said, S. (2020). Remote Health Monitoring Systems Based on Bluetooth Low Energy (BLE) Communication Systems. In: Jmaiel, M., Mokhtari, M., Abdulrazak, B., Aloulou, H., Kallel, S. (eds) The Impact of Digital Technologies on Public Health in Developed and Developing Countries. ICOST 2020. Lecture Notes in Computer Science(), vol 12157. Springer, Cham. https://doi.org/10.1007/978-3-030-51517-1_4

[18] Y. Wang, F. Miao, Q. An, Z. Liu, C. Chen and Y. Li, "Wearable Multimodal Vital Sign Monitoring Sensor With Fully Integrated Analog Front End," in IEEE Sensors Journal, vol. 22, no. 13, pp. 13462-13471, 1 July1, 2022, doi: 10.1109/JSEN.2022.3177205.

[19] Wu, T.; Wu, F.; Qiu, C.; Redoute, J.-M.; Yuce, M.R. A Rigid-Flex Wearable Health Monitoring Sensor Patch for IoT-Connected Healthcare Applications. IEEE Internet Things J. 2020, 7, 6932–6945

[20] Yusuf A. Bhagat, Krishnaveni Das, and Thong Bui "Where the silicone meets the wrist: direct over molding of biometric sensors on a wearable band", Proc. SPIE 11757, Smart Biomedical and Physiological Sensor Technology XVIII, 117570D (12 April 2021); https://doi.org/10.1117/12.2585897

[21] Ganti, V.G.; Carek, A.M.; Nevius, B.N.; Heller, J.A.; Etemadi, M.; Inan, O.T. Wearable Cuff-Less Blood Pressure Estimation at Home via Pulse Transit Time. IEEE J. Biomed. Health Inform. 2021, 25, 1926–1937

[22] Song, S. et al. A 769 μW Battery-Powered Single-Chip SoC With BLE for Multi-Modal Vital Sign Monitoring Health Patches. IEEE Transactions on Biomedical Circuits and Systems 13, 1506–1517 (2019).

[23] J. Přibil, A. Přibilová and I. Frollo, "Wearable PPG Sensor with Bluetooth Data Transmission for Continual Measurement in Low Magnetic Field Environment," 2021 International Conference on Applied Electronics (AE), Pilsen, Czech Republic, 2021, pp. 1-4, doi: 10.23919/AE51540.2021.9542901.

[24] C. Yang, Y. Dong, Y. Chen and N. Tavassolian, "A Smartphone-Only Pulse Transit Time Monitor Based on Cardio-Mechanical and Photoplethysmography Modalities," in IEEE Transactions on Biomedical Circuits and Systems, vol. 13, no. 6, pp. 1462-1470, Dec. 2019, doi: 10.1109/TBCAS.2019.2936414.

[25] A. Sharma et al., "A Sub-60- μA Multimodal Smart Biosensing SoC With >80-dB SNR, 35- μA Photoplethysmography Signal Chain," in IEEE Journal of Solid-State Circuits, vol. 52, no. 4, pp. 1021-1033, April 2017, doi: 10.1109/JSSC.2016.2642205.

[26] T. Wu, M. S. Arefin, J. -M. Redouté and M. R. Yuce, "Flexible wearable sensor nodes with solar energy harvesting," 2017 39th Annual International Conference of the IEEE

Engineering in Medicine and Biology Society (EMBC), Jeju, Korea (South), 2017, pp. 3273-3276, doi: 10.1109/EMBC.2017.8037555.

[27] Zhou, Z.-B.; Cui, T.-R.; Li, D.; Jian, J.-M.; Li, Z.; Ji, S.-R.; Li, X.; Xu, J.-D.; Liu, H.-F.; Yang, Y.; et al. Wearable Continuous Blood Pressure Monitoring Devices Based on Pulse Wave Transit Time and Pulse Arrival Time: A Review. Materials 2023, 16, 2133. https://doi.org/10.3390/ma16062133

[28] O. Abdelatty et al., "A Low Power Bluetooth Low-Energy Transmitter with a 10.5nJ Startup-Energy Crystal Oscillator," ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC), Cracow, Poland, 2019, pp. 377-380, doi: 10.1109/ESSCIRC.2019.8902902.

[29] Lei, K.-M., Mak, P.-I., Law, M.-K. & Martins, R. P. A Regulation-Free Sub-0.5V 16/24MHz Crystal Oscillator for Energy-Harvesting BLE Radios with 14.2nJ Startup Energy and 31.8pW Steady-State Power. 2018 IEEE International Solid - State Circuits Conference - (ISSCC) (2018).

[30] Griffith, D., Murdock, J. & Roine, P. T. A 24 MHz Crystal Oscillator with Robust Fast Start-Up Using Dithered Injection. 2016 IEEE International Solid-State Circuits Conference (ISSCC) (2016).

[31] S. M. Sadaghiani and S. Bhadra, "Acquiring Photoplethysmography (PPG) Signal Without LED," 2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Kuala Lumpur, Malaysia, 2023, pp. 1-6, doi: 10.1109/I2MTC53148.2023.10175960.

[32] AltBeacon. (2016). Android Beacon Library. GitHub. https://github.com/AltBeacon/android-beacon-library

[33] S. Frey, M. Guermandi, S. Benatti, V. Kartsch, A. Cossettini and L. Benini, "BioGAP: a 10-Core FP-capable Ultra-Low Power IoT Processor, with Medical-Grade AFE and BLE Connectivity for Wearable Biosignal Processing," 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS), Berlin, Germany, 2023, pp. 1-7, doi: 10.1109/COINS57856.2023.10189286.

[34] Abdelatty, Omar. Energy-Efficient Wireless Connectivity and Wireless Charging For Internet-of-Things (IoT) Applications. University of Michigan, Electrical Engineering - School of Engineering and Applied Science, PHD (Doctor of Philosophy), 2021, 2021, https://dx.doi.org/10.7302/2887.

[35] Y. Tsividis, "Event-Driven Data Acquisition and Digital Signal Processing—A Tutorial", IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 57, No. 8 (2010).

[36] M. Cai and W. -H. Liao, "A Coaxial Wrist-Worn Energy Harvester for Self-Powered Internet of Things Sensors," in IEEE Internet of Things Journal, vol. 10, no. 1, pp. 133-143, 1 Jan.1, 2023, doi: 10.1109/JIOT.2022.3202166.

[37] S. Li, X. Liu and B. H. Calhoun, "A 32nA Fully Autonomous Multi-Input Single-Inductor Multi-Output Energy-Harvesting and Power-Management Platform with 1.2×105 Dynamic Range, Integrated MPPT, and Multi-Modal Cold Start-Up," 2022 IEEE International Solid- State Circuits Conference (ISSCC), San Francisco, CA, USA, 2022, pp. 1-3, doi: 10.1109/ISSCC42614.2022.9731732.

[38] L. Mishra, Vikash and S. Varma, "Internet of Things for Military Applications," *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2020, pp. 118-123, doi: 10.23919/INDIACom49435.2020.9083730.