Development of an Electrostatic Energy Harvester for Implantable Devices

Auteur : Irabor, George
Promoteur(s) : Redouté, Jean-Michel
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "electronic systems and devices"
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20233

University of Liège - School of Engineering and Computer Science

Development of an Electrostatic Energy Harvester for Implantable Devices

An Autonomous Approach to Powering Medical Implants

Supervisor: Dr. Jean-Michel Rédouté

Master's thesis completed in order to obtain the degree of Master of Science in Electrical Engineering

by George Irabor

Academic year 2023-2024

Abstract

This thesis presents the design, development, and testing of an electrostatic energy harvester (ESEH) tailored for implantable medical devices, particularly pacemakers. The system leverages the biomechanical movements associated with human breathing to generate electrical energy, using variable capacitors to convert mechanical energy into a usable electrical form. This research aimed to address the significant drawbacks of battery-powered devices, such as limited lifespan and the need for frequent surgical replacements.

The study involved a detailed examination of the ESEH system's design and operational principles, highlighting its ability to adjust capacitance dynamically to optimize energy capture from fluctuating biomechanical movements. Experiments conducted tested various configurations of the energy harvester under controlled conditions to assess efficiency and reliability.

The experimental results demonstrated a clear correlation between capacitance adjustments and voltage changes across the capacitor plates. Specifically, when the capacitance was incrementally increased from 30pF to 300pF, the voltage observed across the plates rose from an initial 1.7 volts to a maximum of 2.8 volts under optimal conditions.

While the system could successfully harvest energy, its efficiency was influenced by several factors, including the timely switching of system states and the operational limits imposed by system components like the switching circuit and the unity gain buffer.

However, managing the inverse relationship between capacitance and voltage affected the optimal timing for energy capture and storage. Modifications to the circuit design and control strategy were necessary to improve the system's performance. These included adding an external switching circuit to isolate the variable capacitor when harvesting and integrating a more effective measurement setup to minimize energy losses during voltage assessments.

The research confirms the viability of using ESEH for powering implantable medical devices, with the potential to significantly reduce the reliance on batteries and decrease the frequency of surgical interventions. Future work will focus on refining the technology to improve scalability, energy efficiency, and integration with existing medical devices, moving closer to practical applications in the healthcare sector.

Acknowledgements

I am profoundly grateful for the support and guidance I have received during this project, and I would like to extend my heartfelt thanks to all those who made this journey both possible and rewarding.

First, my sincerest appreciation to Gabriel Digregorio, whose dedication to answering my questions and creating a supportive work environment was invaluable. Gabriel's commitment not only enhanced my project experience but also provided a solid foundation for my work.

I am equally thankful to Morgan Diepart for his substantial assistance throughout the project. Morgan's willingness to address all my inquiries, no matter how silly, was a tremendous help and greatly appreciated.

Special thanks to Angel, whose timely interventions were nothing short of miraculous. His ability to anticipate and address my needs was truly like having a mind reader by my side.

I owe a debt of gratitude to Professor Rédouté for his inspirational teaching and unwavering belief in my abilities. His courses were instrumental in building my confidence and ensuring I was well-prepared to tackle this project.

I would also like to express my profound thanks to my brother, Dr. Julian Irabor, for his enduring support throughout my master's program. Living and learning in Liège was made all the more enriching and feasible thanks to his encouragement and support.

Lastly, I dedicate this project to Hervé Pierre, the designer of the PCB, whose enthusiasm and creativity left a lasting impression during our brief acquaintance. I hope this project serves as a tribute to his memory. May he rest in peace.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Cardiovascular diseases are the leading cause of mortality globally, often necessitating the deployment of pacemakers to regulate heart rhythms [3]. Traditional energy sources for these devices, primarily batteries, present significant limitations in terms of lifespan and size [4].

Frequent battery replacements not only introduce physical risks due to repeated surgeries but also impose psychological stress on patients, exacerbating anxieties related to surgical outcomes and device reliability [5]. The inherent limitations of current battery technology in medical implants can lead to increased healthcare costs, while also contributing to environmental degradation through the substantial use of resources and generation of waste [6].

Energy harvesting has emerged as a pivotal innovation in sustainable and autonomous energy development, particularly for powering biomedical implants. This technology captures and stores energy from ambient sources, including thermal, solar, and mechanical energies [7, 8].. Mechanical energy harvesting (MEH), especially through electrostatic methods, is particularly noted for its adaptability and efficiency in converting mechanical vibrations into electrical energy [7, 8].

Electrostatic energy harvesting (ESEH), a type of MEH, involves the conversion of mechanical energy into electrical energy through the manipulation of electric fields and charges within a variable capacitor system [2]. This mechanical energy can be harnessed from consistent physiological activities, such as breathing, as it could provide a continuous and reliable energy source for these devices [9, 10].

ESEH offers a compelling alternative that promises to extend the device's opera-

tional life and reduce the frequency of surgical interventions for battery replacements [11].

1.2 Research Objectives

This study aims to explore the feasibility and effectiveness of electrostatic energy harvesting (ESEH) using variable capacitors in biomedical applications. Specifically, the research objectives are to:

- A comprehensive review of existing energy harvesting technologies and their applications in biomedical devices.
- A detailed examination of electrostatic principles as applied to energy harvesting.
- Simulation and optimization of an ESEH system specifically designed for use in pacemakers.
- Analysis of the practical challenges and potential solutions in implementing ESEH in a real-world biomedical context.

Chapter 2

Literature Review

2.1 Energy Harvesting Techniques

Energy harvesting from the body is a growing area of research within the field of power management for implantable medical devices. These methods primarily focus on converting mechanical energy into electrical energy using ambient body energy sources. This section explores various techniques such as piezoelectric, triboelectric, thermal and using variable capacitors and their associated challenges.

- **Piezoelectric harvesting:** Piezoelectric materials generate electricity through mechanical deformation. This process involves inducing a change in the electrical field within the piezoelectric material due to mechanical stress, which is then captured by storage capacitors [12]. Traditionally, the primary issue with piezoelectric harvesting has been the brittleness of piezoceramic materials, which can limit their application in environments requiring flexibility and durability. Moreover, the energy output from piezoelectric materials is often inconsistent, influenced significantly by the frequency and amplitude of the mechanical stress [13]. To address these challenges, recent developments include the use of flexible piezoelectric materials like PVDF. These materials are utilized in novel tensile-mode energy harvesters, which offer a more uniform strain distribution compared to bending-mode harvesters. A noteworthy advancement in this field is a tensile-mode piezoelectric energy harvester that employs a three-hinged force amplification mechanism. This design incorporates a rigid beam and an elastic PVDF film connected via a revolute joint, with the film pre-stretched to bear dynamic tensile loads under harmonic excitations. Theoretical and experimental validations reveal that this setup exhibits a significant hardening effect due to its nonlinear geomet-

ric configuration, enhancing both voltage output and harvesting bandwidth by 500% and 1250%, respectively, compared to traditional bending-mode cantilevered harvesters [14].

- **Triboelectric harvesting** This technique utilizes the contact electrification between two different materials. When integrated with a harvesting mechanism, the charge accumulated from this contact is transferred to a storage system [15]. Despite its potential, maintaining consistent contact and separation in a controlled manner can be challenging, especially in the dynamic environment of a human body. Furthermore, the long-term durability and stability of the materials used also pose significant challenges [15].

- **Thermal Energy Harvesting:** This technique investigates the use of thermal energy harvesters and thermoelectric generators (TEGs) for converting body heat into electrical energy, focusing on optimizing power conversion efficiency. Two TEG configurations are compared: (1) a μTEG with low thermal resistance paired with a high-efficiency DC-DC converter, and (2) an mTEG with high thermal resistance coupled with a low-input voltage DC-DC converter. The mTEG configuration yields up to 65% higher output power in lab settings and 1–15% in real-world conditions, dependent on physical activity and environmental factors. Both systems can reach power densities of up to 14 μW/cm^2 and achieve 16% to 24% of theoretical maximum efficiency. The variability in body temperature and external environment can impact the efficiency of these harvesters, necessitating careful system design [16, 17].

- **Variable capacitors** have found significant applications in biomedical devices, particularly in scenarios where ambient energy availability varies significantly. These capacitors dynamically adjust capacitance to enhance power management and extend device lifespans [18]. An example of this technology is seen in MEMS-based inertial vibration scavenging micro-generators. These devices employ a variable capacitor with one moving plate attached to a proof mass. Energy is extracted as the plates separate under a constant charge, with this non-resonant device operating over a wide range of excitation frequencies and amplitudes. Remarkably, energy extracted per cycle is reported at 120 nJ, which is significantly above previous reports, showcasing its efficacy [19].

  Another aspect of variable capacitors is their utilization in electrostatic micro-power generators, where inertial forces work against the electric field of the capacitor, converting mechanical energy into electrical potential energy. Such capacitors are crucial for portable, wearable, or implantable electronic

devices, harnessing body motion as a mechanical energy source. Initial testing of a micromachined variable capacitor showed a capacitance variation from 100 pF to 1 pF as the mass moves, which could lead to a hundred-fold increase in voltage under constant charge mode. Tests demonstrated that periodic high voltage outputs up to 2.3 kV can be achieved, corresponding to an energy conversion rate of 2.4 µJ per cycle or 24 µW at a 10 Hz vibration frequency [20]. However, implementing variable capacitor technology in a compact and biocompatible format is technically challenging. The systems must also be highly reliable, as any failure in the energy storage component can result in device malfunction.

2.2 Electrostatic Energy Harvesting in Animals and Humans

2.2.1 Observations in Animals

Energy harvesting from animals has been explored, focusing on mechanical, thermal, and biochemical energies. Dagdeviren et al. (2017) discuss various in vivo applications, highlighting the integration of novel materials and device layouts for energy harvesting from animals, which could be applied to human medical devices as well [10]. These novel materials and device layouts consist of advanced piezoelectric materials, such as ZnO nanowires, used for their flexibility and compatibility with biological tissues. Triboelectric nanogenerators (TENGs) made from biocompatible polymers like PDMS and PTFE harness mechanical movements through the triboelectric effect. The use of biocompatible and biodegradable materials like silk fibroin ensures the safety and sustainability of implanted devices. The study emphasizes flexible and stretchable designs to conform to dynamic body movements, ensuring efficient operation without discomfort.

Multi-layered structures combining different energy harvesters, such as piezoelectric and triboelectric, enhance energy generation and storage capabilities. These devices can be implanted beneath the skin or within organs to harvest energy from muscle movements, blood flow, and other activities [11]. Dagdeviren et al. (2017) demonstrated that these novel materials and layouts significantly enhance the efficiency and feasibility of in vivo energy harvesting.

2.2.2 Observations in Humans

Human-based energy harvesting has shown considerable promise with the advent of devices capable of harnessing energy from daily human activities. Zhou et al.

(2021) categorize different energy harvesters based on biomechanical and biochemical energies derived from various body movements [23]. Invernizzi et al. (2016) discuss the use of triboelectric nanogenerators (TENGs) which utilize the triboelectric effect to generate energy through the contact and separation of different materials. TENGs are constructed from biocompatible polymers such as PDMS (Polydimethylsiloxane) and PTFE (Polytetrafluoroethylene), known for their flexibility and compatibility with biological tissues. These materials, when incorporated into multi-layered and flexible structures, significantly enhance the efficiency and feasibility of energy harvesting from human motion. They discuss the remarkable potential of TENGs, capable of achieving conversion efficiencies of around 70 % under optimal conditions, making them a viable solution for self- sustaining medical devices.

2.3 Advancements in Variable-Capacitance Energy Harvesters

The development of variable-capacitance electrostatic energy harvesters (VCEH) has been significantly advanced by MEMS technology, which facilitates the miniaturization of components. These advancements, as articulated by [2], emphasize the design considerations for miniaturization and performance enhancement to suit biomedical applications.

- **MEMS Technology:** Enables the integration of tiny mechanical structures with electronic circuits, enhancing compactness and functionality. Techniques like photolithography and deep reactive ion etching are used to construct precise, miniaturized components [21].
- **Material Selection:** High dielectric materials such as silicon dioxide (SiO_2), hafnium dioxide (HfO_2), and titanium dioxide (TiO_2) are preferred for their ability to store and convert electrical energy efficiently, while providing mechanical robustness [22].
- **Energy Conversion Efficiency:** MEMS-based VCEHs convert mechanical energy to electrical energy more efficiently than traditional devices, with conversion rates significantly higher under optimal conditions [23].

2.3.1 Optimization of Variable Capacitors

- **Capacitance Maximization:** Designs that maximize capacitance change with minimal mechanical movement, employing movable plates that adjust

separation distance or overlapping area in response to mechanical stimuli [24].

- **Energy Harvesting Techniques:** Comparison of resonant and non-resonant mechanisms to determine the most efficient method for specific applications. Non-resonant devices operate over a wide range of excitation frequencies and amplitudes, suitable for variable conditions [19].
- **Experimental Results:** Studies show significant power outputs, with a micromachined variable capacitor achieving an energy conversion rate of 2.4 µJ per cycle at a vibration frequency of 10Hz, corresponding to an output power of 24 µW [20].

2.3.2 Applications and Benefits

- **Biomedical Devices:** Miniaturized VCEHs are particularly useful in implantable medical devices, where space is limited and reliability is critical. These devices harvest energy from body movements, ensuring a continuous power supply for essential functions [**?**].
- **Wearable Electronics:** Variable capacitors efficiently harvest energy from body movements, providing a reliable power source for devices such as fitness trackers and health monitors [23].

2.4 Analysis of Capacitive Energy Storage Systems

Capacitive energy storage systems are integral in the realm of energy harvesting for implantable medical devices, primarily due to their capability to efficiently manage and store energy. These systems utilize capacitors, which store energy electrostatically by maintaining a static electric charge. Here, we discuss various innovative methods and findings from diverse research efforts that highlight the advancements and challenges associated with these technologies.

One notable advancement is the development of electrostatic double-layer capacitors (EDLCs), commonly known as supercapacitors. These devices are renowned for their high power density and rapid charging capabilities. Unlike batteries, EDLCs can endure many more charge and discharge cycles without degradation, which is essential for applications where energy availability frequently fluctuates [25].

Additionally, microfabrication techniques have led to the creation of microsuper-

capacitors. These tiny capacitors can be integrated directly onto chips along with other microelectronic devices, making them particularly suitable for MEMS-based energy harvesters [26]. However, reducing the size of supercapacitors often results in a decrease in capacitance and overall energy storage capacity, posing significant challenges in maintaining reliable performance.

In another innovative approach, hybrid energy storage systems have been developed by combining capacitors with batteries. This hybrid system leverages the high energy density of batteries and the high power density of capacitors, thereby enhancing the overall efficiency and lifespan of the storage system [27].

Moreover, the incorporation of nanomaterials, such as graphene or carbon nanotubes, has significantly improved the performance of capacitors. These nanomaterials increase both the capacitance and energy density of the capacitors, making them more efficient [28]. Nevertheless, the scalable production of nanomaterials at an economically feasible cost and their integration into existing manufacturing processes remain challenging.

These studies collectively provide a comprehensive perspective on the advancements and challenges in capacitive energy storage systems. They underscore the diverse approaches being explored to enhance the suitability of these systems for implantable medical devices.

2.5 Microcontroller-based Measurement Techniques

Vostrukhin et al. provide an in-depth analysis of microcontroller-based converters for capacitance measurement, which are essential for controlling and optimizing the performance of electrostatic energy harvesters [4]. Their method involves charging the capacitor with a pulse and then using an onboard timer to precisely measure the discharge time. This measurement and control technique allows for a precise control over the charging and discharging of the electrostatic energy harvester using variable capacitor technology.

2.6 Challenges and Future Directions

Despite significant advancements, challenges remain in optimizing energy conversion efficiency, device longevity, and integration with existing medical devices. Future research is directed towards overcoming these hurdles, focusing on improving material properties and circuit configurations [8]. For instance, Harerimana et al. (2020) present circuit topologies for low power energy harvesting with efficiencies ranging from 15% to 50%, demonstrating the potential and limitations of

current designs [6]. Daneshvar et al. (2021) describe a variable-capacitance energy harvester optimized with miniaturized inductors, showing improved performance over traditional designs [29]. Mendiratta et al. (2018) highlight the importance of capacitor optimization in improving conversion efficiency for piezoelectric energy harvesting, achieving up to 14.86% efficiency [30]. Bieske et al. (2017) introduce a new topology for accurate characterization of electrostatic energy harvesters, enhancing measurement accuracy and efficiency [31]. Stein et al. (2016) provide a theoretical comparison of energy harvesting methods, emphasizing the charge pump method for its efficiency in MEMS applications [32].

Chapter 3

Methodology

This is part of a larger study to fully develop a unique variable capacitor that can be inserted into the human body to take advantage of the energy of breathing to generate electrostatic energy. The primary objective of this project is to design an effective system that can seamlessly switch between different states, leveraging microcontrollers and electronic systems to optimize energy capture.

The focus of my contribution lies in taking forward the work already initiated on the PCB design and its fabrication. My responsibilities include finalizing the design, rectifying any existing flaws, and providing robust solutions. Additionally, I am tasked with developing an algorithm to control the system states, programming the microcontroller, and conducting tests to validate the effectiveness of the switching mechanism. These tests utilize a controlled variable capacitor designed to simulate the final application setting, ensuring that the system functions effectively under realistic conditions.

The research is being conducted at the Microsystems Laboratory of the Montefiore Institute, University of Liège, Belgium. It builds upon the theoretical framework proposed by Daneshvar et al., focusing on the dynamic capabilities of variable capacitors for energy harvesting in medical applications.

3.1 Operational Theory

The operational principle of the system hinges on the notion that mechanical movements, such as those induced by breathing, can alter the capacitance characteristics of a capacitor. These alterations can occur through changes in the distance between the plates, variations in the dielectric material, or adjustments in the plate area. When these capacitance changes occur under a constant initial

charge, electrical energy is generated through the conservation of charge principle. As the capacitance varies while the charge remains constant, the energy stored in the capacitor is modified, manifesting as a voltage increase or decrease across the capacitor plates [2].

This stored energy can then be harnessed by connecting the variable capacitor to a load, allowing it to act as an energy source and discharge through the connected load. The process comprises four distinct steps:

1. **Investment Phase**: The capacitor, at its maximum capacitance $C_{\max}$, is charged to an initial voltage V_r. This phase establishes the energy storage within the capacitor.

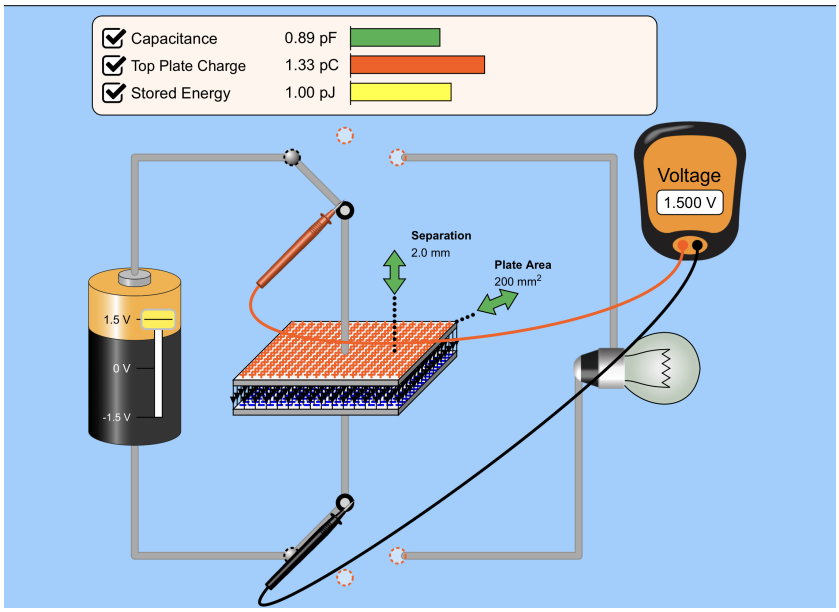

Figure 3.1: Investment Phase: Initial charging of the capacitor (Image courtesy of the University of Colorado's PHET interactive simulations) [1]

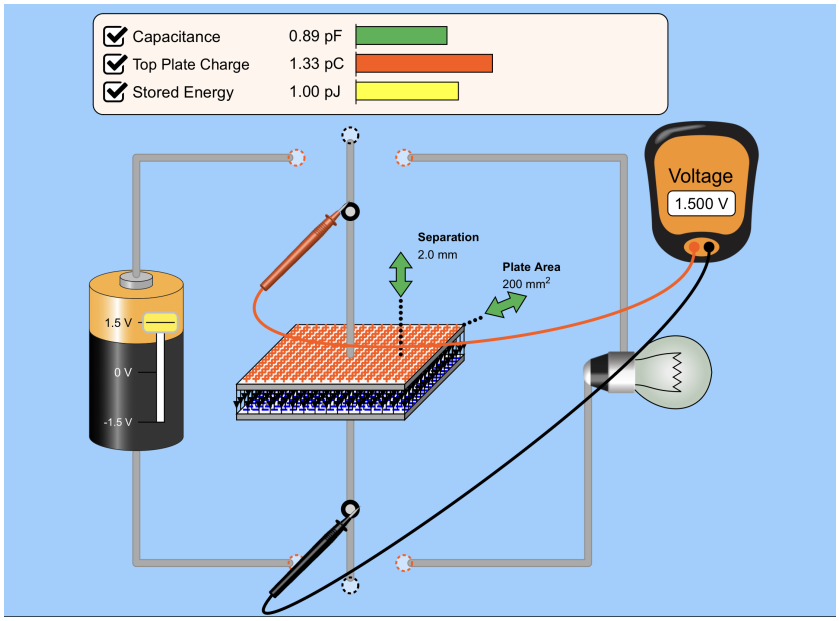

Figure 3.2: Investment Phase: The capacitor is isolated from the circuit (Image courtesy of the University of Colorado's PHET interactive simulations) [1]

2. **Harvesting Phase**: During this phase, the distance between the capacitor plates is increased, thereby decreasing the capacitance to C_{min} and causing a corresponding increase in voltage (V_{cv}) due to the conservation of charge ($Q = CV$).

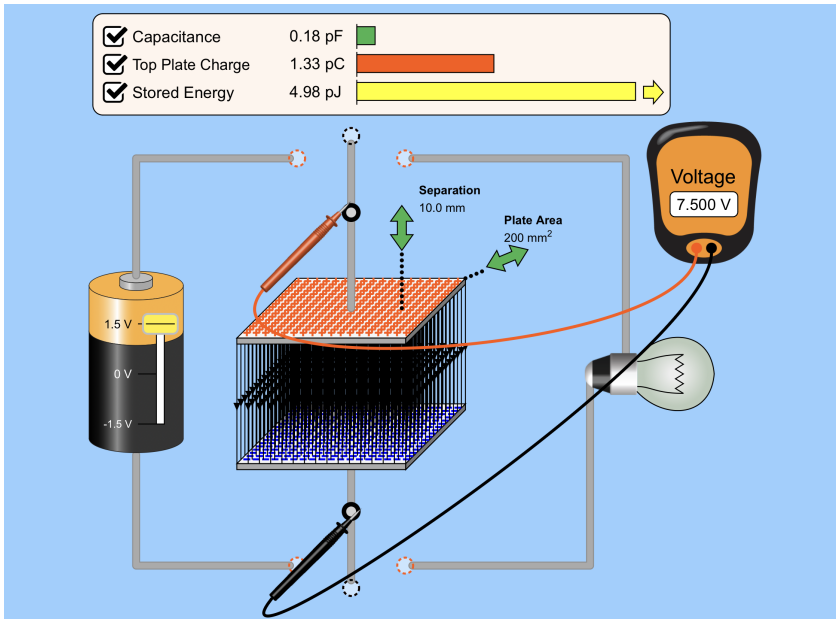

Figure 3.3: Harvesting Phase: Separation of the capacitor plates increases the voltage across them (Image courtesy of the University of Colorado's PHET interactive simulations) [1]

3. **Reimbursement Phase**: The switch is conencted to the load to allow the discharge of the accumulated charge into it, effectively converting the stored energy into current.

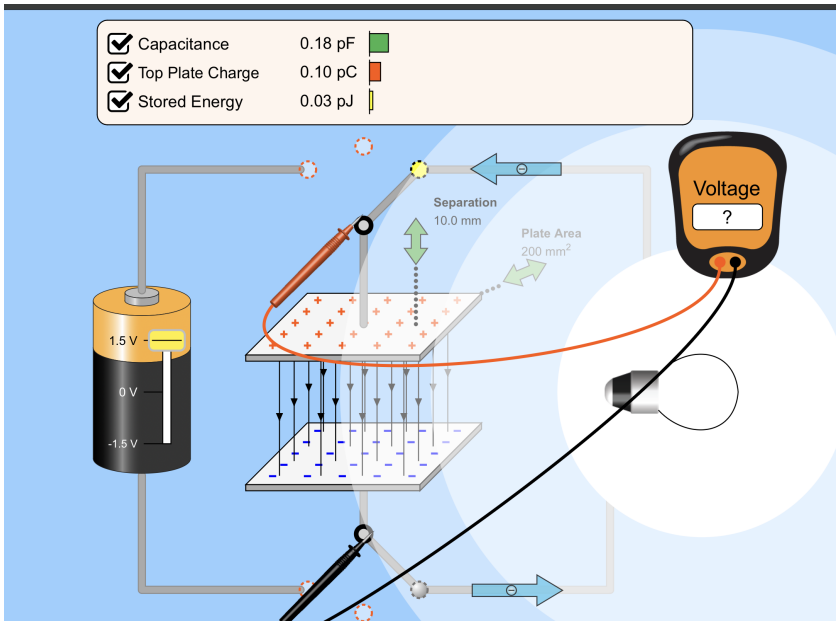

Figure 3.4: Reimbursement Phase: The capacitor discharges energy to a load (Image courtesy of the University of Colorado's PHET interactive simulations) [1]

4. **Recovery Phase**: The system resets as the capacitance returns to $C_{\max}$, preparing for the next energy harvesting cycle.

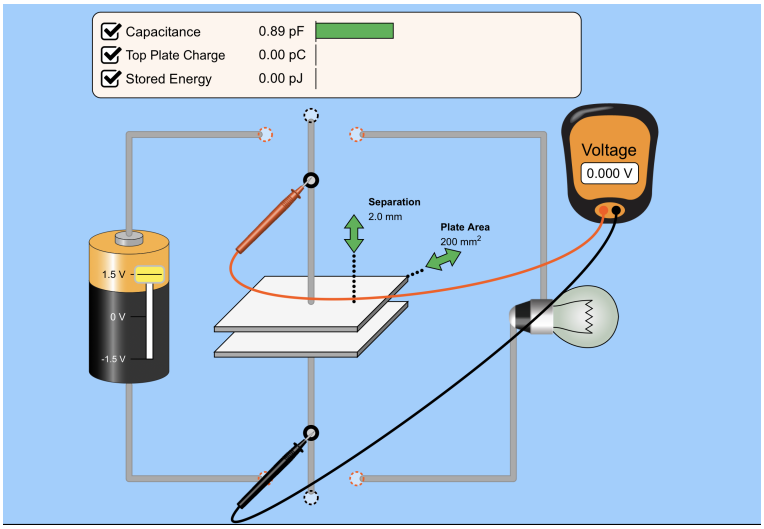

Figure 3.5: Recovery Phase: Capacitor plates are reset to start the next charging cycle (Image courtesy of the University of Colorado's PHET interactive simulations) [1]

3.1.1 Capacitance Variation

The capacitance of a parallel plate capacitor is given by the equation:

$$C = \frac{\epsilon_0 A}{d}$$

where:

- C is the capacitance,
- ϵ_0 is the permittivity of free space ($\approx 8.85 \times 10^{-12}$ F/m),
- A is the area of one of the plates,
- d is the distance between the plates.

Mechanical movements, such as breathing, alter the distance d between the capacitor plates, thereby changing the capacitance C.

During inhalation, the distance d between the plates decreases, leading to an increase in capacitance C. Conversely, during exhalation, the distance d increases, resulting in a decrease in capacitance C.

This variation in capacitance can be effectively utilized for energy harvesting. As the mechanical movement causes the plates to move, the changing distance directly impacts the capacitance according to the formula above.

By harnessing this change in capacitance due to mechanical motion, the system can generate and store electrical energy. This principle is the foundation of the energy harvesting mechanism employed in the design of the variable-capacitance electrostatic energy harvester.

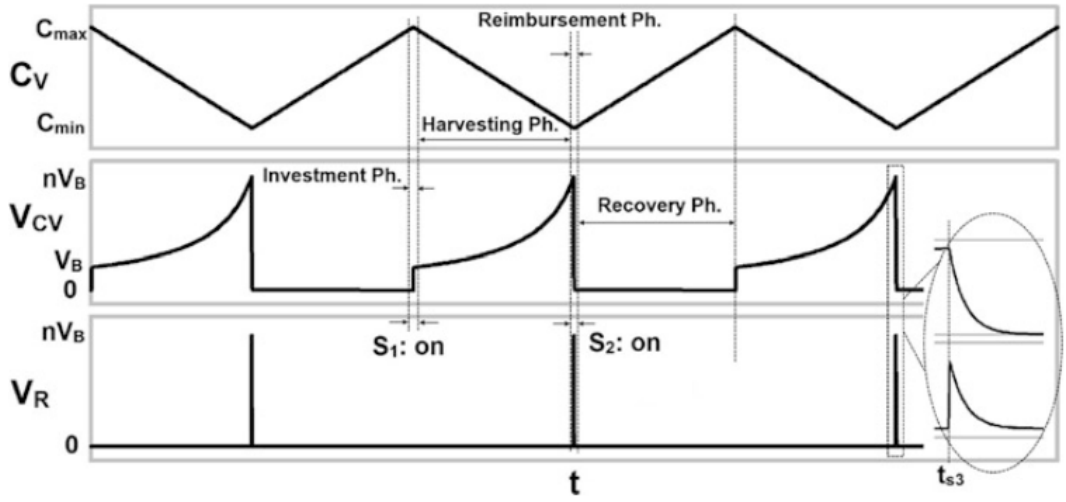

Figure 3.6: The changes in the capacitance of the variable capacitor and the voltages across CV and RL in the circuit [2]

3.1.2 Energy Conversion

The process of converting mechanical energy into electrical energy involves the four discussed phases [2]:

- **Investment Phase**: The capacitor is charged by a storage component. The energy received is:
$$E_{\text{vi}} = \frac{1}{2} C_{\max} V_r^2$$
- **Harvesting Phase**: Mechanical energy changes the capacitance from $C_{\max}$ to $C_{\min}$. The harvested energy is:
$$E_{\text{hC}} = \frac{1}{2} C_{\max} (n-1) V_r^2$$
where $n = \frac{C_{\max}}{C_{\min}}$.
- **Reimbursement Phase**: The stored energy is transferred to the load or storage component:
$$E_{\text{vr}} = \frac{1}{2} C_{\min} (nV_r)^2$$
- **Recovery Phase**: The capacitor is ready for the next cycle. The net deliverable energy is:
$$E_{\text{del}} = E_{\text{vr}} - E_{\text{vi}} = \frac{1}{2} C_{\min} (n-1) V_r^2$$

3.2 System Components

The system is comprised of several key components that collectively contribute to efficient energy harvesting. These components are designed to work in harmony, facilitating precise control and measurement capabilities.

- **Tunable Capacitors**: Utilized to dynamically adjust their capacitance value:
  - A tunable capacitor with a range from 300pF to 900pF for preliminary tests, allowing for basic functionality checks and system adjustments.
  - A precision-engineered tunable capacitor with parallel plates, designed for refined control and accurate measurements in advanced testing scenarios.
- **Measurement and Testing Tools**: Critical for assessing system performance and ensuring component integrity:
  - **LCR Meter**: Measures inductance (L), capacitance (C), and resistance (R), essential for tuning and calibrating the components within the system.
  - **Oscilloscope**: Used to monitor voltage fluctuations over time, providing a visual representation of the energy harvesting efficiency.
  - **Multimeter**: A versatile tool for various electrical measurements, indispensable for routine diagnostics and troubleshooting.
- **Power Supply**:
  - A 3.7V Lithium Ion Rechargeable Battery, supplying the necessary power for system operations and charging processes.
- **Key Electronic Components**: These elements are essential for managing power flow and enhancing circuit functionalities:
  - **Voltage Regulator**: The LD39015M18R is a low quiescent current, low noise voltage regulator that maintains a stable 1.8V output, ensuring consistent performance under varying load conditions.
  - **Switches**:
    - ADG6412 High Voltage Quad SPST Switch, designed for high-efficiency signal routing.
    - MIC4066B CMOS Quad Bilateral Switch, facilitating the seamless transmission or multiplexing of both analog and digital signals.

- **Amplifiers**: The TLV2372 operational amplifier, configured as a Unity Gain buffer, stabilizes signal strength to prevent distortion.
- **Diodes**:
  - BAT54CW,115: A diode array that protects against voltage spikes and power surges, enhancing system reliability.
- **Capacitors**:
  - Multilayer Ceramic Capacitors (47uF ±20% 25V X5R SMD 1206): Provide stable capacitance and reliability in high-frequency applications.

The control logic of the system is currently being developed and tested using a NucleoL412KB board. This setup allows for iterative programming and real-time adjustments before final integration with the STM32L412Kb microcontroller, ensuring that the system logic is fully optimized and reliable.

As part of the project's infrastructure, a pre-designed PCB was utilized, which had been developed prior to my involvement. This existing hardware setup included several Analog-to-Digital Converters (ADCs) that play a critical role in the system's measurement. However, the placement and configuration of these ADCs presented specific challenges that influenced the project's execution:

- **Suboptimal ADC Placement:** The positioning of ADCs on the PCB was not ideally suited for minimizing charge draw from the variable capacitors, which would completely discharge the capacitor in a matter of microseconds.
- **Configuration Limitations:** The ADC configuration, as inherited from the previous design, restricted the flexibility needed to adapt to the addition of new components to the system.

These inherited design aspects necessitated additional considerations and adjustments in the experimental approach to ensure accurate data collection and system functionality.

3.3 Circuit Design and Simulation

3.3.1 Circuit Diagrams

We present schematic diagrams of the circuits used in the system, detailed as follows:

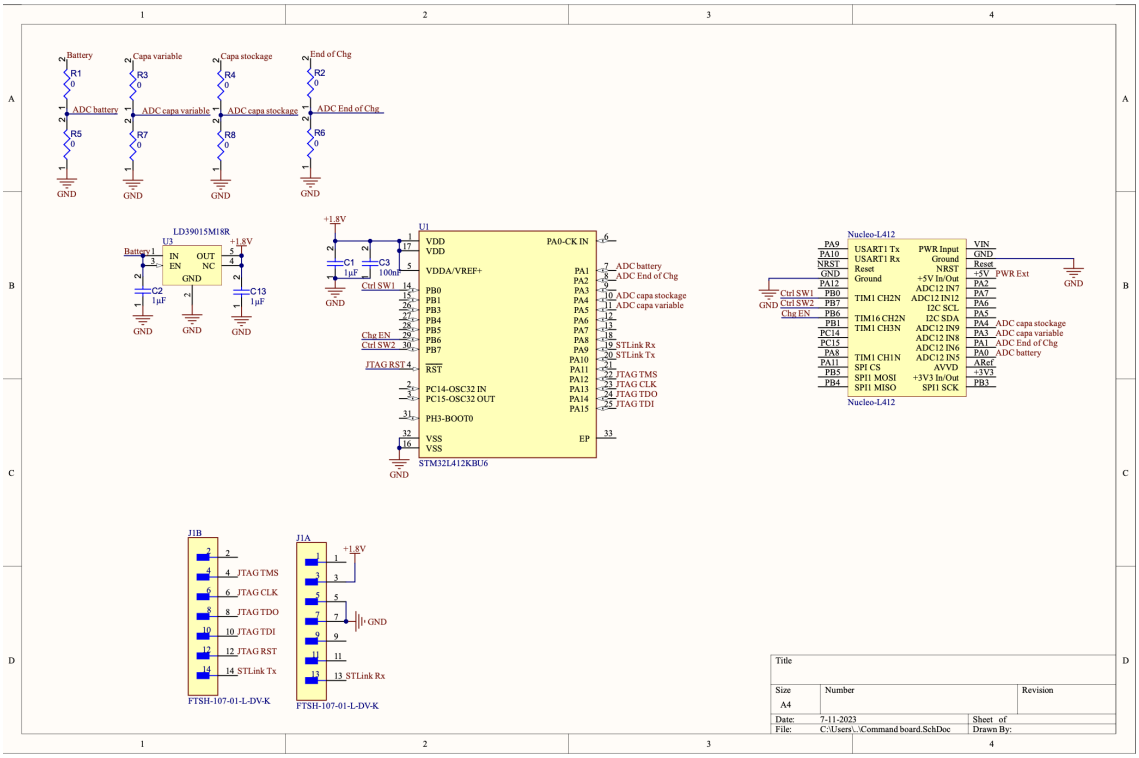

Figure 3.7: Left: Circuit to step down voltage from the battery to 1.8V; Center: Microcontroller indicating all its connections to the rest of the circuit; Right: Nucleo Board connected to the microcontroller for programming and debugging; Bottom Left: Connectors for programming and power.

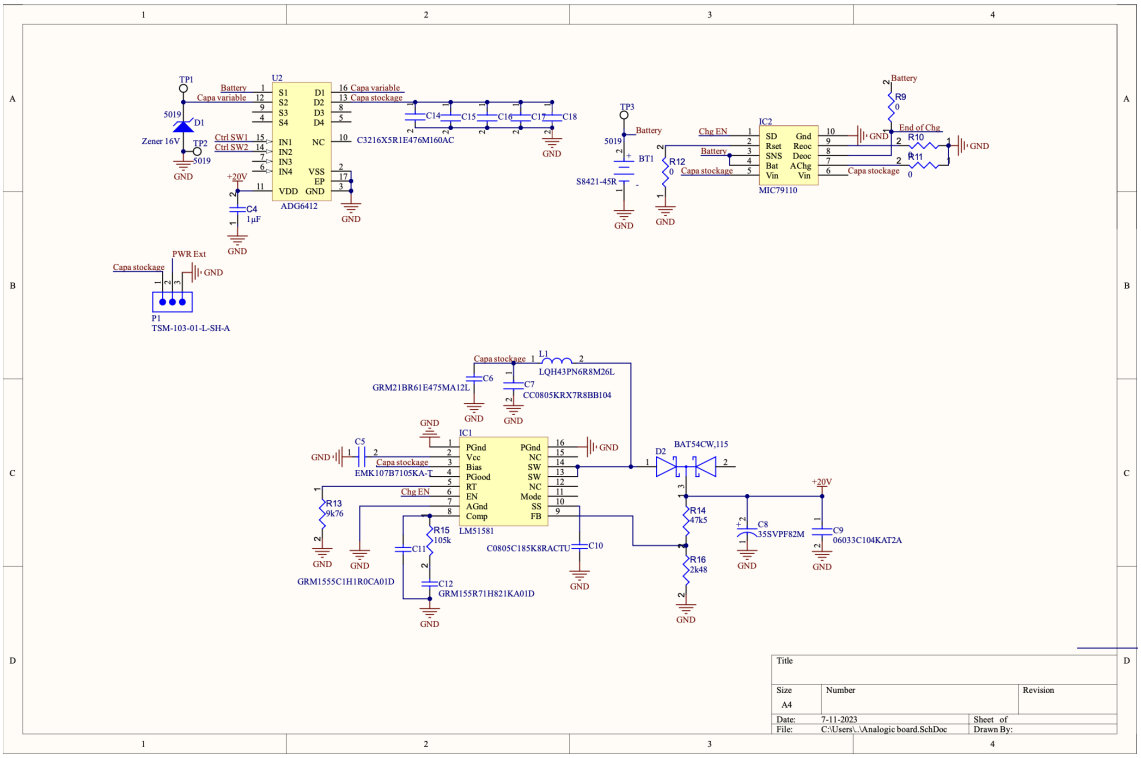

Figure 3.8: Top Left: Switching IC to control switches between the battery, variable capacitor, and capacitor storage; Top Right: Charger circuit for lithium-ion batteries; Bottom Center: Buck-Boost Converter to charge batteries at a higher voltage and current.

Voltage Step-Down

This section of the circuit converts the battery's voltage down to a stable 1.8V required by the microcontroller and other low-voltage components. This step-down is required for the microcontroller to operate in its lower power mode, optimizing energy consumption.

Microcontroller

The STM32L412KBU6 serves as the central processing unit for controlling various functions and processes within the circuit. It is perfect for its ultra low power functionality. The Microcontroller manages all computational tasks and controls peripheral devices. It is connected to the Nucleo board for programming and debugging, which facilitates software uploads and testing. The nucleo board `Nucleo L412KB`provides an interface for programming the STM32 microcontroller. It controls the switching IC, the battery charger and measures values across the system,

including the battery, capacitor storage, and variable capacitor.

Switch Control and Capacitor Management

The ADG6412 quad Single pole single throw (SPST) switch manages the routing of power between the battery, variable capacitor, and storage capacitors. It includes a Zener diode in parallel to the variable capacitor that breaks down at 16V to protect the circuit by flushing excess voltage to ground, preventing overvoltage damage. The **storage capacitors** store electrical energy temporarily in order to power the recharging circuit to recharge the battery through the battery charger circuit.

Battery Charger Circuit for Lithium-Ion Batteries

The MIC79110 charges the lithium-ion battery efficiently and safely, using storage capacitor power. It is a linear charger designed for lithium-ion batteries, providing necessary controls for safe charging, including current regulation and voltage monitoring.

Buck-Boost Converter The LM51581, a versatile buck-boost converter capable of handling higher voltages and currents, is used for maintaining stable operation across a range of input voltages. It supplies 20V to power the switching circuit to be able to accommodate higher voltages from the variable capacitor.

3.3.2 Simulation Model

To predict the behavior of the variable capacitor, a SPICE simulation model was developed using LTSpice, aiming to replicate the following circuit configuration:

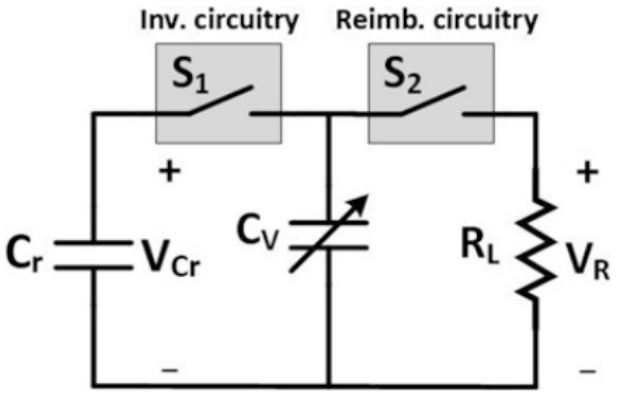

Figure 3.9: Investment and Reimbursement Circuitry

The schematic of the SPICE model is shown below, detailing the components and their interconnections:

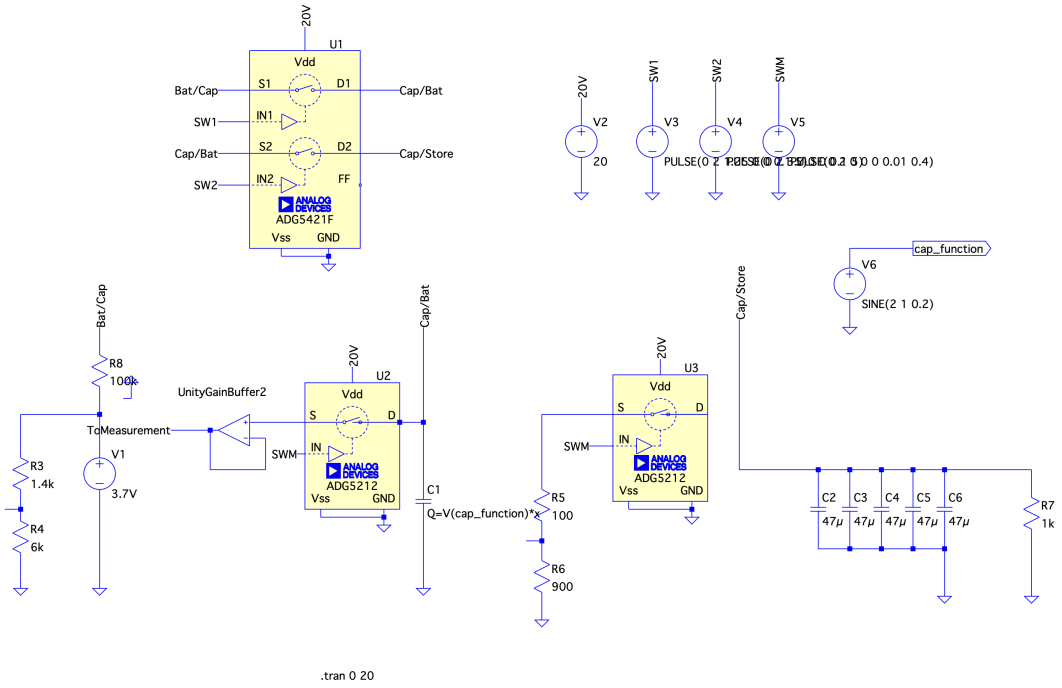

Figure 3.10: Circuit Simulation in SPICE

3.3.3 Simulation Results

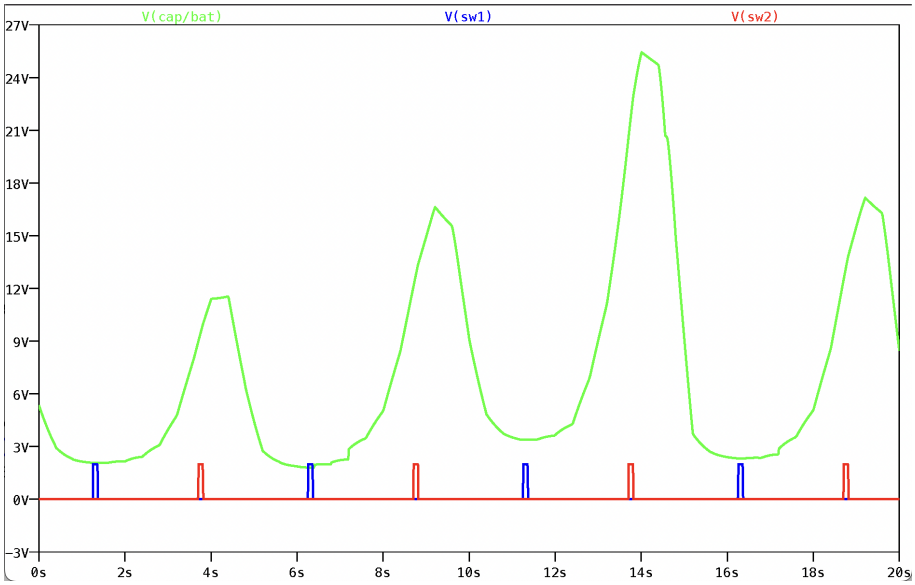

Figure 3.11: Simulation Result showing voltage levels: Switch 1 (blue), Switch 2 (red) and Voltage across the variable capacitor (green).

The simulation results confirmed that the circuit functions according to its design specifications. The key process observed involves the variable capacitor being isolated during the harvesting phase, during which its capacitance value reduces. As the capacitance of the variable capacitor reduces, its ability to store charge remains consistent, leading to an increase in voltage across its terminals. This behavior aligns with the expected inverse relationship between capacitance and voltage in a capacitor $V = \frac{Q}{C}$, where Q is the charge stored.

It should be noted that the observed variations in voltage peaks and behaviors during each harvesting phase are primarily attributed to the limitations of simulating variable capacitors in LTSpice and similar software. Accurately modeling the behavior of variable capacitances presents significant challenges, often resulting in disparate outcomes across simulations. Despite these inconsistencies, the depicted graph demonstrates a relative consistency in the pattern of voltage peaks, although each peak displays unique voltage measurements.

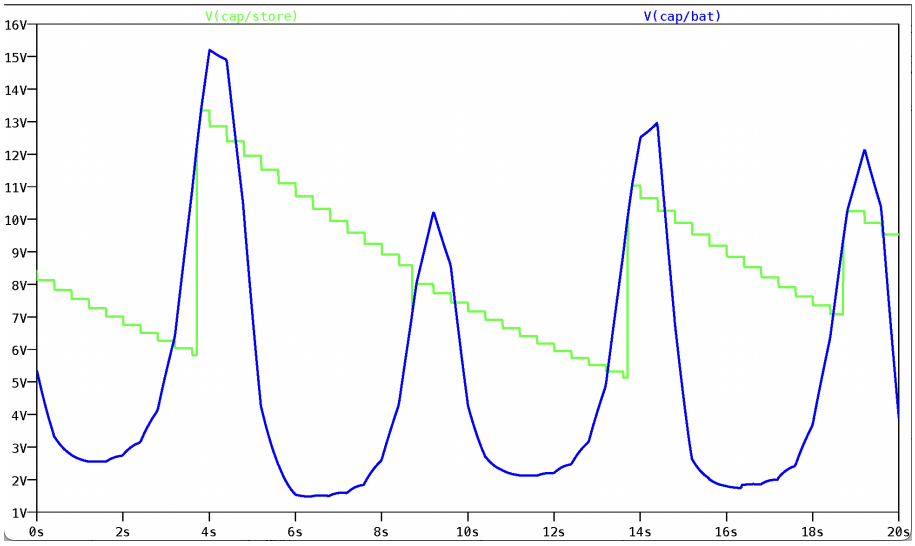

Figure 3.12: Simulation Result showing voltage levels: Voltage across the variable capacitor (blue) and the harvested voltage (green).

Subsequently, the circuit activates switch 2, transitioning the system into the reimbursement phase. During this phase, the variable capacitor discharges, effectively transferring its stored energy into the storage capacitors. Unexpectedly, the voltage across the storage capacitors exhibited a progressive decrease over time after each reimbursement from the variable capacitor, a phenomenon not initially anticipated in the circuit design parameters. This gradual voltage decline can be attributed to the measurement circuit, which draws a small but cumulative amount of power with each pulse. This continual draw on the storage capacitors leads to a substantial reduction in voltage across their terminals over time.

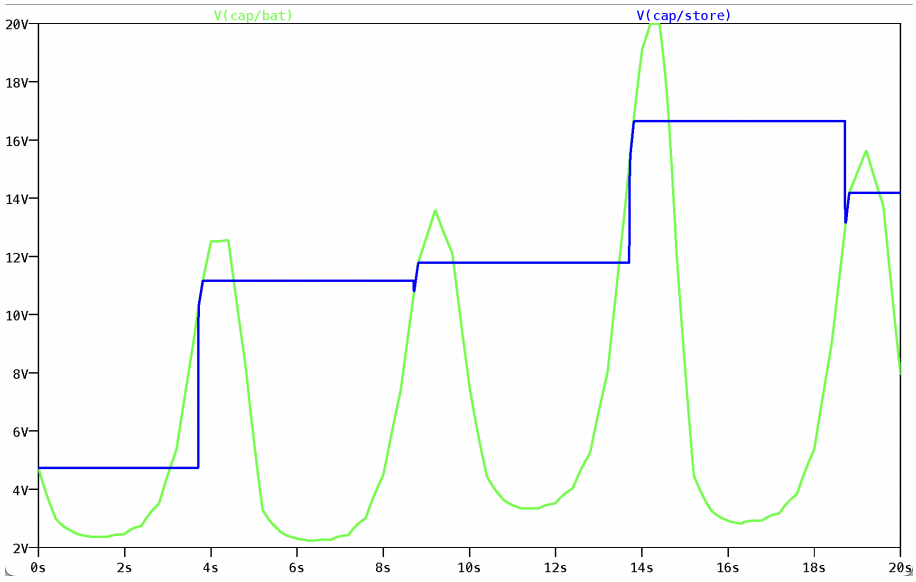

Figure 3.13: Simulation Result showing voltage levels: Voltage across the variable capacitor (green) and the harvested voltage (blue).

Upon isolating the measurement circuit from the system, it is evident that the storage capacitor maintains its charge effectively until it receives additional charge from the variable capacitor, at which point its charge notably increases. This observation confirms that in the absence of the measurement circuit, the system functions as intended. However, the absence of the measurement circuit makes it challenging to control the switching circuit accurately, as it is crucial to know the points of maximum and minimum capacitance for optimal operation.

To address this issue, the integration of a unity gain buffer is proposed to facilitate the measurement process without significantly impacting the voltage across the circuit. The buffer would ideally isolate the measurement circuit, minimizing the power draw during voltage sensing. Nonetheless, powering the unity gain buffer itself presents a challenge, particularly in terms of power conservation. Striking a balance between accurate measurements and minimal power consumption is essential, necessitating careful design consideration to optimize the buffer's impact on the system's overall energy efficiency.

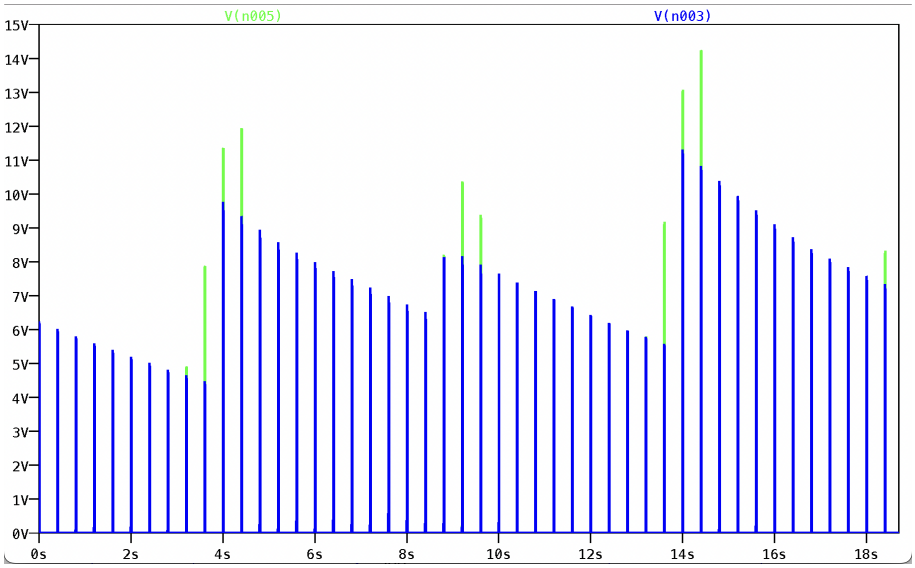

Figure 3.14: Simulation Result showing measured voltage levels: Variable Capacitor Voltage Divider (green) and Harvested Voltage Divider (blue).

To precisely monitor these voltage changes without excessively draining power from the energy harvesting circuit, the system employs a method of intermittent pulsing. This technique involves generating short, controlled pulses to periodically activate the measurement system. Between pulses, the system remains in a low-power or inactive state, thereby conserving energy and extending the operation time between necessary recharge cycles. The voltage levels are measured at the peak of each pulse, providing discrete snapshots of the system's dynamic behavior.

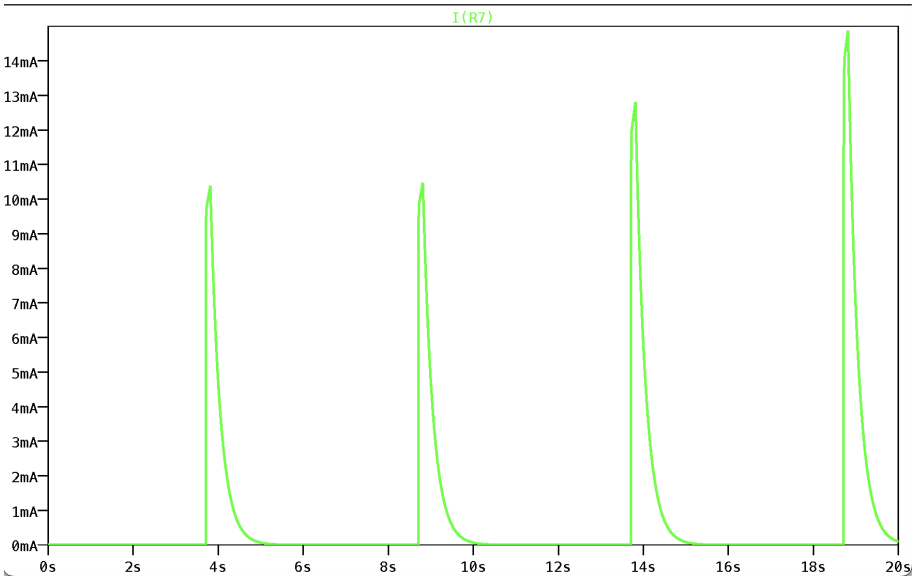

Figure 3.15: Simulation Result showing measured voltage levels: Variable Capacitor Voltage Divider (green) and Harvested Voltage Divider (blue).

To test the output, we placed a 1k resistor just after the storage capacitors and we were able to observe that we are able to obtain about 14mA at maximum of pulsing currents. In the actual system, the voltage is provided to a battery charger which aims to provide a more steady current supply in order to recharge the system. But from this we can obtain an output power estimate

Given:

- Peak current, $I_{\max} = 14\,\text{mA}$.
- Resistance, $R = 1\,\text{k}\Omega$.
- Duration of one pulse, $\tau = 0.2\,\text{s}$ (estimated from the graph).
- Pulse interval, $T = 5\,\text{s}$ (time between start of one pulse and the next).

Given the pulsing nature of the current, $P_{\text{inst}}(t)$ varies significantly over each cycle of the pulse. The waveform can be approximated as a triangle waveform.

A triangular pulse increases linearly to a maximum value and then decreases back to zero. This linear variation affects how the average power is calculated because the power depends on the square of the current (i^2) over time.

Mathematical Basis

For a triangular waveform, the mean of the square of the waveform, which is crucial for power calculations, is $\frac{1}{3}$ of the square of its peak value. This results from the integral of the square of the waveform over its period.

Let the triangular current pulse $i(t)$ rise linearly from 0 to $I_{\max}$ over a period from 0 to $T/2$ and then fall back to 0 in the next half. The calculation of the integral of $i^2(t)$ over a full period T is as follows:

For the increasing part from 0 to $T/2$:

$$\int_0^{T/2} \left(\frac{2I_{\max}}{T}t\right)^2 dt = \frac{4I_{\max}^2}{T^2}\int_0^{T/2} t^2\,dt = \frac{4I_{\max}^2}{T^2}\cdot\frac{(T/2)^3}{3} = \frac{I_{\max}^2 T}{12} \tag{3.1}$$

This integral is the same for the decreasing part, hence the total integral for the full period is:

$$\text{Total Integral} = 2\cdot\frac{I_{\max}^2 T}{12} = \frac{I_{\max}^2 T}{6} \tag{3.2}$$

Average of $i^2(t)$ over the period:

$$\text{Average of } i^2(t) = \frac{\frac{I_{\max}^2 T}{6}}{T} = \frac{I_{\max}^2}{6} \tag{3.3}$$

For the RMS (root mean square) value used in power calculations:

$$I_{\text{RMS}} = \sqrt{\frac{1}{T}\int i^2(t)\,dt} = \frac{I_{\max}}{\sqrt{3}} \tag{3.4}$$

Thus, the average power P_{avg} calculated using the RMS value is:

$$P_{\text{avg}} = R\left(\frac{I_{\max}}{\sqrt{3}}\right)^2 = \frac{RI_{\max}^2}{3} \tag{3.5}$$

Putting the values of I max and R:

$$P_{\text{pulse}} = \frac{RI_{\max}^2}{3} = \frac{1000\Omega\times(0.014\,\text{A})^2}{3} = \frac{1000\times 0.000196}{3} \approx 0.0653\,\text{W} = 65.3\text{mW}$$

Scaling to Total Interval

Number of pulses in 20 seconds:

$$n = \frac{20\,\text{s}}{5\,\text{s/pulse}} = 4\,\text{pulses}$$

Total energy contributed by all pulses in 20 seconds:

$$E_{\text{total}} = P_{\text{pulse}} \times n \times \tau = 0.0653\,\text{W} \times 4 \times 0.2\,\text{s} = 0.05224\,\text{J}$$

Average power over the 20-second interval:

$$P_{\text{avg}} = \frac{E_{\text{total}}}{20\,\text{s}} = \frac{0.05224\,\text{J}}{20\,\text{s}} \approx 0.002612\,\text{W} \approx 2.612\,\text{mW}$$

The average output power over a 20-second interval, with periodic triangular current pulses, is approximately 2.612 mW.

3.4 Design Specifications

3.4.1 Capacitor Specifications

The capacitance C of a parallel plate capacitor is calculated using the formula:

$$C = \frac{\epsilon A}{d}$$

where:

- C is the capacitance in farads (F),
- ϵ represents the permittivity of the medium between the plates, calculated as $\epsilon = \epsilon_0 \epsilon_r$. Here, ϵ_0 (the permittivity of free space) is approximately 8.85×10^{-12} F/m, and ϵ_r is the relative permittivity of the medium,
- A is the area of one of the plates in square meters (m^2),
- d is the distance between the plates in meters (m).

Given that the plate area A is 0.00070686 m^2, the equation above is used to calculate capacitance for varying plate distances influenced by breathing motions.

Experimental Setup and Measurements

The capacitance values were measured using an LCR meter as shown below, with the distance between the plates being adjusted using a precision-controlled test bench.

Figure 3.16: LCR Meter used for capacitance measurement

The physical setup of the test bench without any dielectric material is displayed below. This setup helps in understanding the capacitance variation with changes in the air gap due to mechanical movements influenced by breathing.

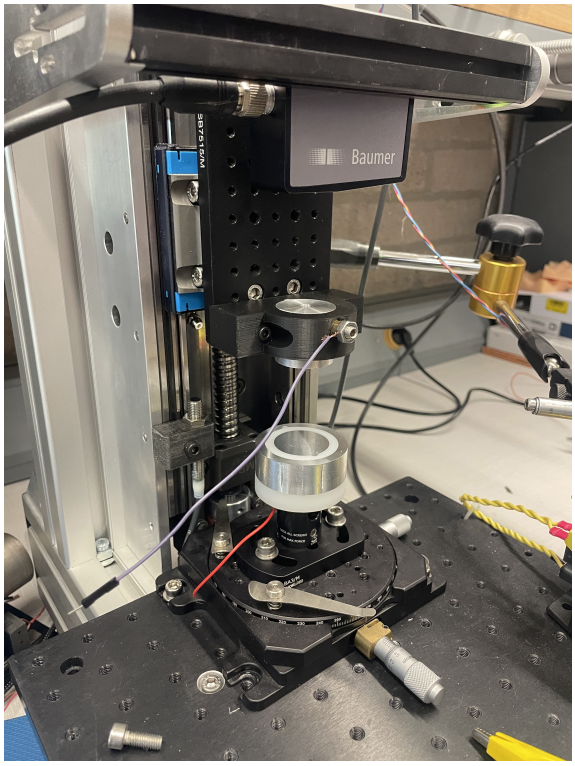

Figure 3.17: Test Bench Setup illustrating the parallel plates arrangement

The capacitance variation relative to the displacement of the parallel plates is detailed in Figure 3.18.

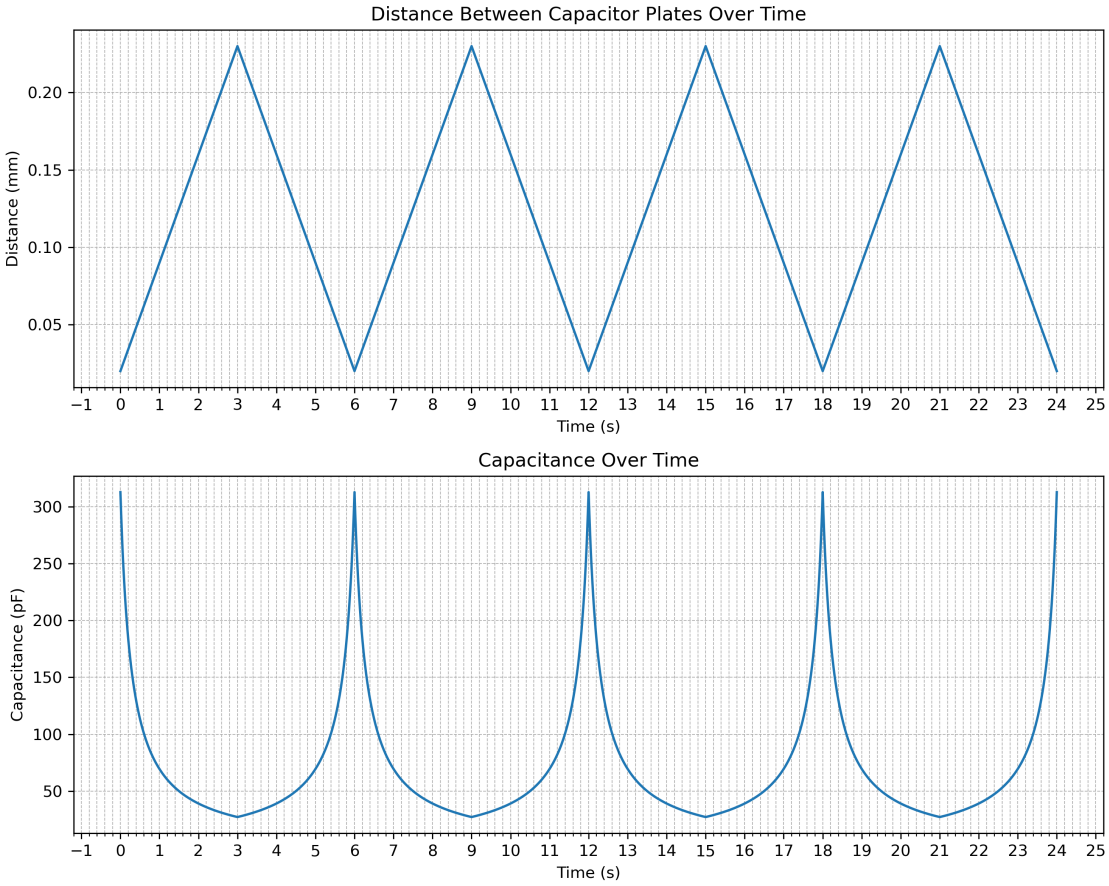

Figure 3.18: Graph of capacitance against time and displacement against time

Table 3.1: Capacitance vs. Distance for Variable Capacitor

Distance (μm)	**Capacitance (pF)**
20.000	312.785
43.333	144.363
66.667	93.836
90.000	69.508
113.333	55.197
136.667	45.773
160.000	39.098
183.333	34.122
206.667	30.270
230.000	27.199

System Configuration and Capacitance Range

The test bench configuration is programmed to accommodate the following settings:

- **Post Home Position:** 52.355 mm,
- **Displacement:** 0.63 mm,
- **Velocity:** 0.0372 mm/s.

Capacitance Range:

- **Minimum Capacitance:** 15 pF (at maximum plate separation),
- **Maximum Capacitance:** 325 pF (at minimum plate separation).

Using the capacitance formula, the plate separations for maximum and minimum capacitances are calculated as follows:

$$d_{\text{min}} = \frac{8.854187817 \times 10^{-12}\ \text{F/m} \times 0.00070686\ \text{m}^2}{325 \times 10^{-12}\ \text{F}} = 1.925 \times 10^{-5}\text{m} = 19.25\mu\text{m}$$

$$d_{\text{max}} = \frac{8.854187817 \times 10^{-12}\ \text{F/m} \times 0.00070686\ \text{m}^2}{15 \times 10^{-12}\ \text{F}} = 4.17 \times 10^{-3}\text{m} = 417\mu\text{m}$$

Therefore the variation in capacitance against distance for the project is shown in 4.2.

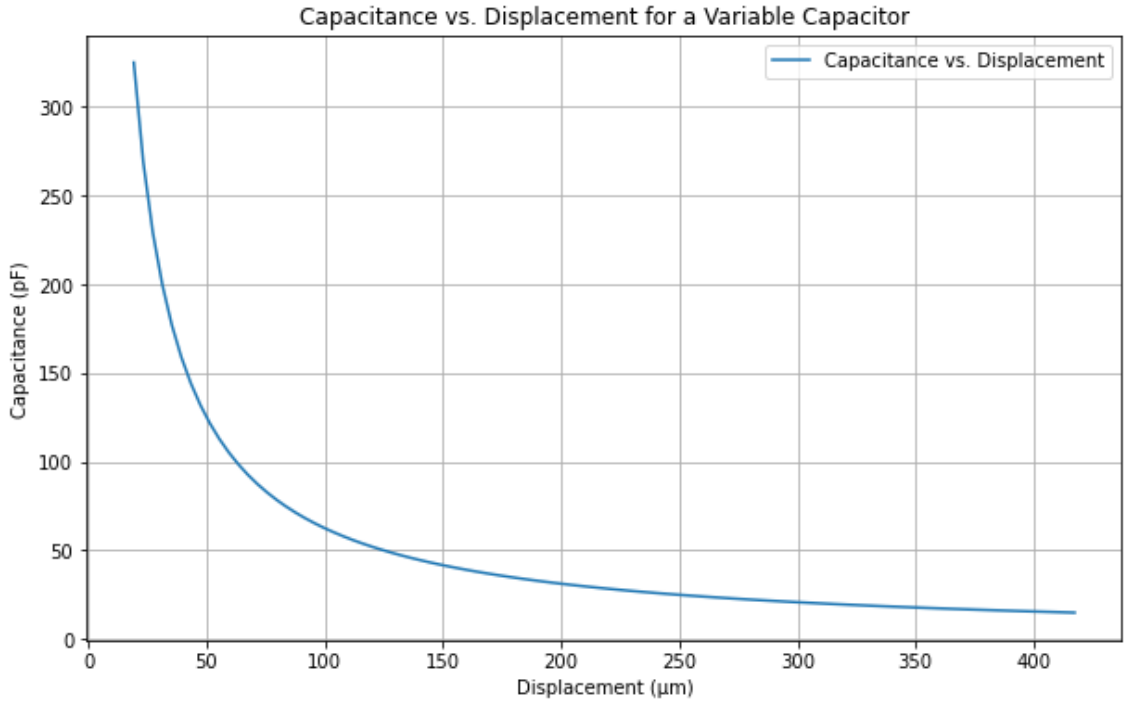

Figure 3.19: Capacitance in microFarads against displacement in microMeters

The distances between the plates could be reduced further, but the test bench in motion causes occasional contact between the plates, leading to short circuits.

3.4.2 Design of the Test Bench Commands and Specifications

Data Collection and Analysis

Data collected during the test phases were instrumental in understanding the dynamics of the system under real conditions. The following table summarizes some of the key measurements taken during the tests:

Table 3.2: Displacement Variations

Measurement	**Value (mm)**
Initial Displacement	100.6730
Up Displacement	100.6850
Down Displacement	101.2880
Variation of Distance (Up to Down)	0.6030

This table represents a snapshot of the mechanical settings and adjustments made during the testing phase, with value representing the displacement in millimeters.

3.5 Capacitance Measurement in Microcontroller STM32

Measuring the capacitance proved to be a very important aspect in the execution of this project. Knowing the capacitor values at any time helps to program the microcontroller to switch states optimally. Directly measuring capacitance with a microcontroller can be quite challenging. Several approaches to capacitance measurement include:

- Measuring the fall time of a charged capacitor, which necessitates precise timing control within the microcontroller.
- Comparing the capacitor's value against a preset threshold from a microcontroller pin using comparators. While effective, this method may compromise energy efficiency.

Given the constraints of these methods, a novel system was proposed to enhance both the accuracy and energy efficiency of the measurement process. This system operates by sending a 20-microsecond pulse through the capacitor every 500 milliseconds, corresponding to the low-frequency requirements of the application.

This method was tested using a vintage tunable capacitor from old radios, featuring capacitance values ranging from 300pF to 800pF, as illustrated in Figure 3.20.

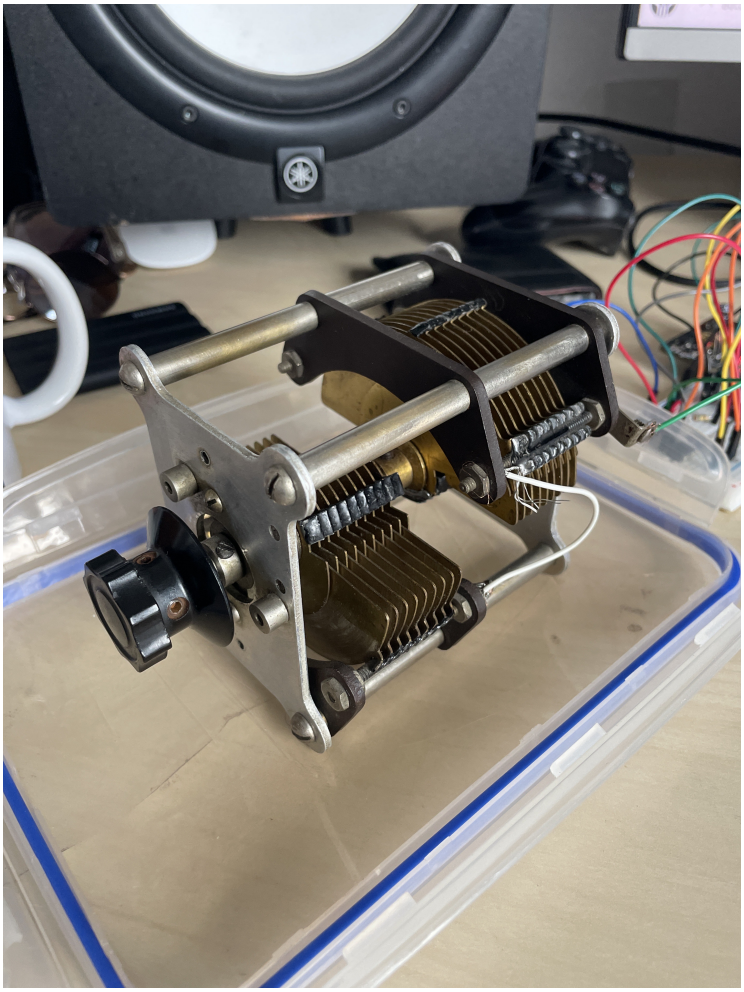

Figure 3.20: Tunable Capacitor with capacitance between 300pF and 800pF

The capacitor was connected to a Nucleo L412KB board, which supplied regular 20-microsecond pulses through a 100k resistor. The setup and measurement process is depicted in Figure 3.23.

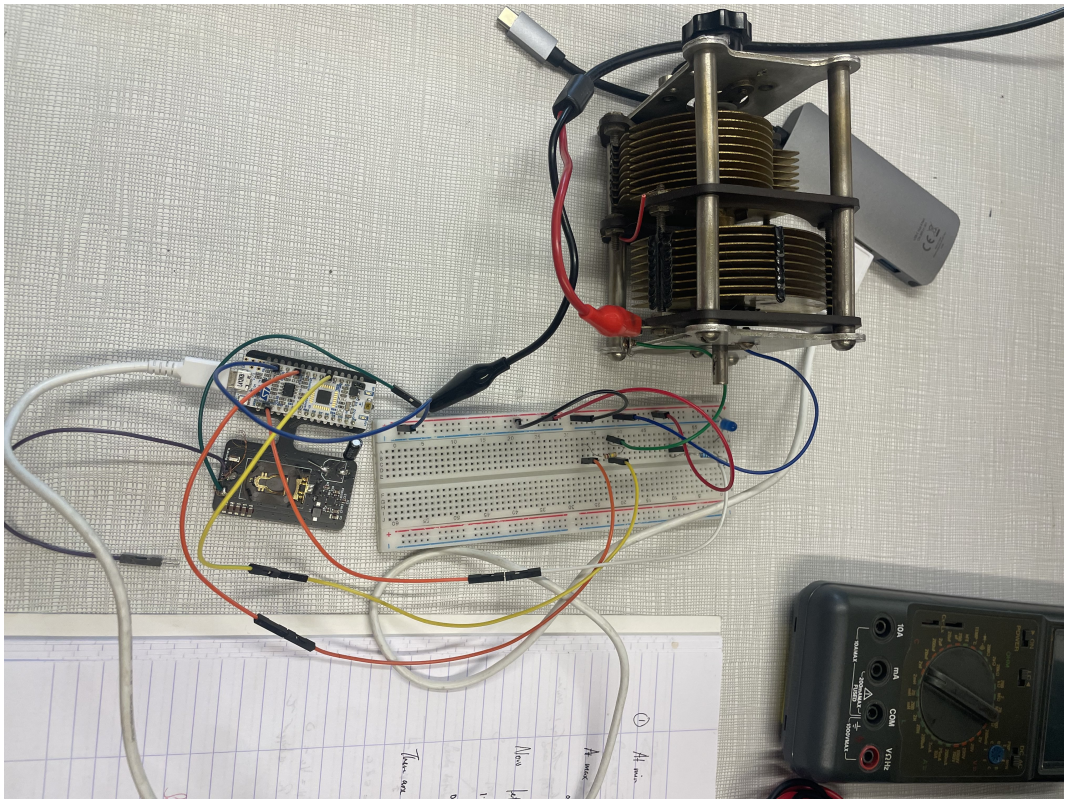

Figure 3.21: Measurement setup with tunable capacitor and Nucleo board, showing pulse application and voltage observation at various capacitance values using an oscilloscope

Initially, pulses were sent every 80 microseconds, which was too frequent for a low-frequency application and inefficient for continuous capacitance monitoring. The results from this setup, showing both the maximum and minimum voltage peaks, are presented in Figures 3.23 and 3.23.

Figure 3.22: Capacitance at minimum ofFigure 3.23: Capacitance at maximum of 300pF with a peak voltage of 1.8 Volts. 900pF with a peak voltage of 1.3 Volts.

To improve efficiency, the pulse frequency was adjusted to every 500 milliseconds. This modification yielded more consistent results, allowing for accurate measurements of peak voltage at varying capacitance levels.

Initially, measuring the spike directly with an ADC is convenient. However, when the capacitor needs to conserve the charge, direct discharge through the internal resistor or the pin of the microcontroller was problematic. To address this, a unity gain buffer with a very high input impedance was used to isolate the variable capacitor from the microcontroller ADC input, minimizing charge reduction while still allowing measurement.

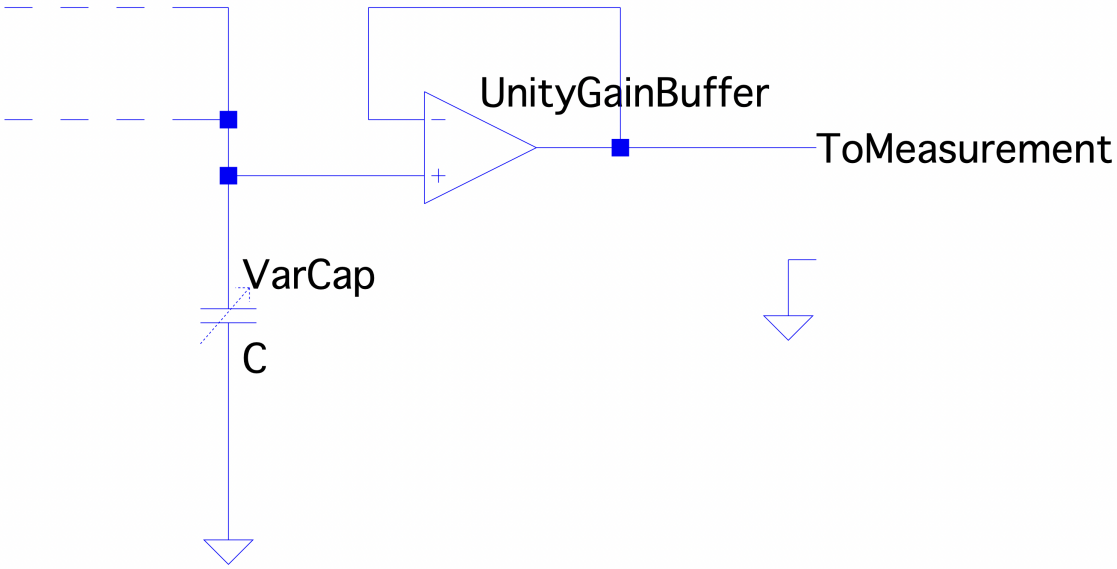

Figure 3.24: Measurement Circuit for ADC Input

Concerns related to the unity gain buffer involve powering it efficiently, possibly by following the switching nature of the measurement to conserve power. For the nature of testing the project, it is sufficient to power the Unity gain buffer directly and constantly, but for the real application this will not be possible.

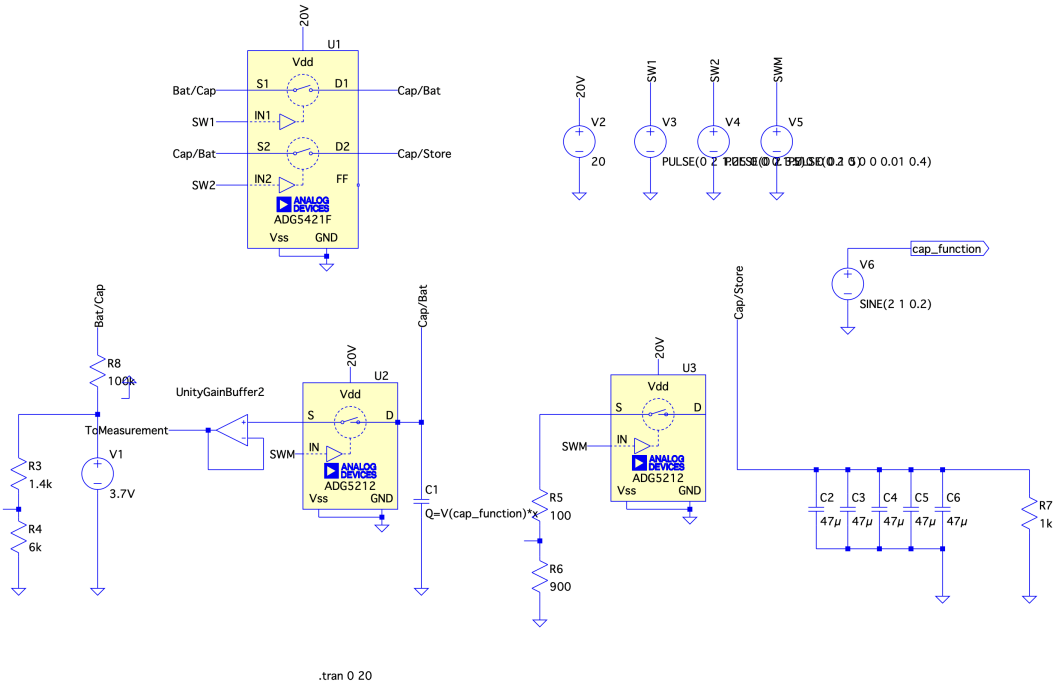

Figure 3.25: Circuit Diagram of Electrostatic Energy Harvesting indicating measurement with Unity Gain Buffer

3.6 Measurement and Operational Phases

The operational sequence of the system is delineated into the same four distinct phases:

1. **Recovery Phase:** In this initial phase, switches SWM and SW1 are engaged to initiate a pulse. This pulse serves to measure the peak voltage across the capacitor, essential for determining when the voltage begins to rise, indicating the capacitor has reached its maximum capacitance.

2. **Investment Phase:** Subsequently, SWM is temporarily deactivated while SW1 is then engaged for a longer duration, ensuring the capacitor is fully biased and charged to its maximal potential.

3. **Harvesting Phase:** During the harvesting phase, SW1 is deactivated, allowing the charged capacitor to maintain its voltage. SWM is toggled periodically to measure the voltage across the capacitor plates without causing significant discharge. This phase is sustained until the voltage plateaus at its peak, signaling that the capacitor is fully charged under current conditions.

4. **Reimbursement Phase:** Finally, SW2 is activated, enabling the charged capacitor to discharge its stored energy into a storage system, such as a battery or a network of capacitors. The process then recommences with the Recovery Phase.

Operational Dynamics and Voltage Behavior

A consistent pulse is transmitted at regular intervals throughout the operational cycle. Given the short duration of each pulse, the capacitor does not reach full charge. This phenomenon reflects an inverse relationship between capacitance and the observed voltage across the capacitor: a lower capacitance results in a quicker charge time and a higher voltage across the terminals, while a higher capacitance leads to a slower charge and consequently a lower voltage.

This dynamic is depicted in Figure 3.26, illustrating the expected variations in voltage during the charging cycle. This visual aids in understanding how changes in capacitance affect voltage responses under controlled charging conditions.

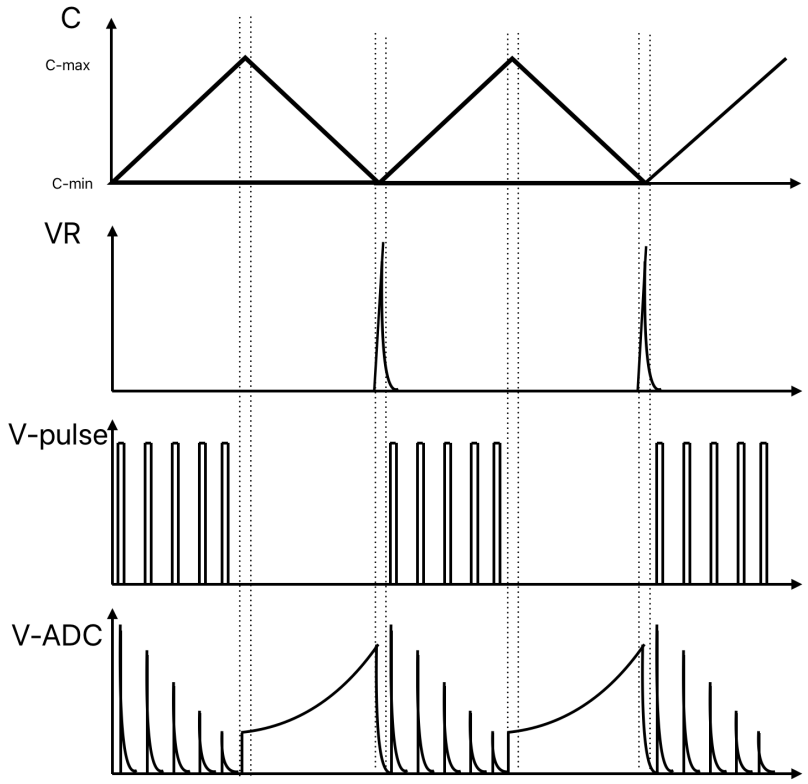

Figure 3.26: Expected Variations in the Voltage Across the Capacitor During Charging

3.7 Circuit Modification

To enhance measurement accuracy and maintain the integrity of the charge on the variable capacitor, the circuit design was updated to incorporate an analog switch (MCI4066B).

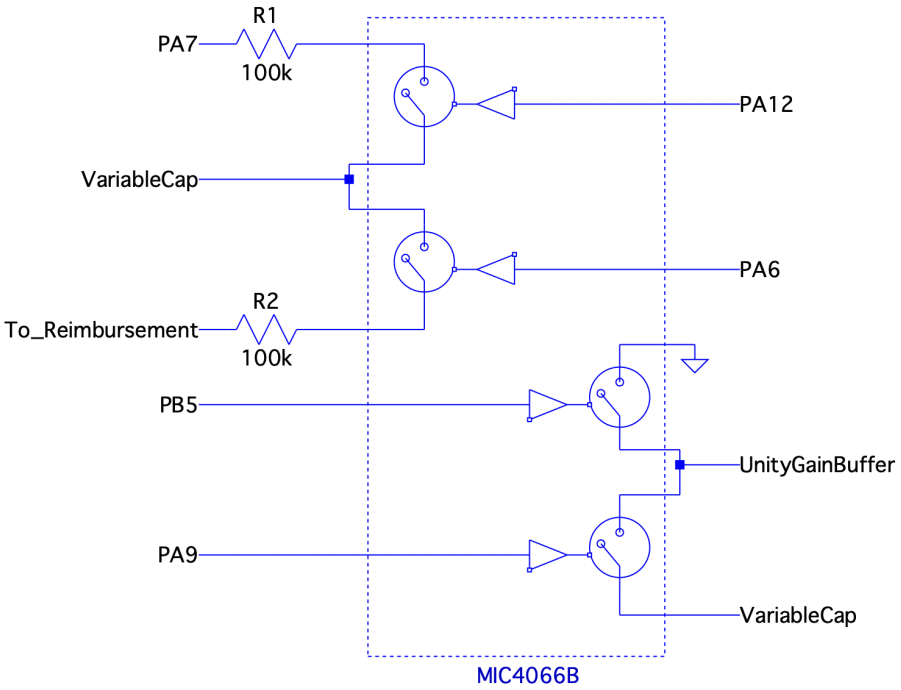

Figure 3.27: Analog Switch MIC4066B highlighting connections between components of the system

This switch facilitates selective interaction with the measurement system via pin PA9, effectively reducing charge loss through the ADC voltage divider, which was an issue in the original design. Additionally, the switch connects the unity gain buffer input to the ground when inactive, preventing output rail voltage. It also connects the variable capacitor to the pulse source at PA7, controlled by PA12, and to the reimbursement circuit, managed by PA6. This configuration enhances control and minimizes unwanted discharge.

The modified circuit was assembled on a stripboard, illustrated in figure 3.28.

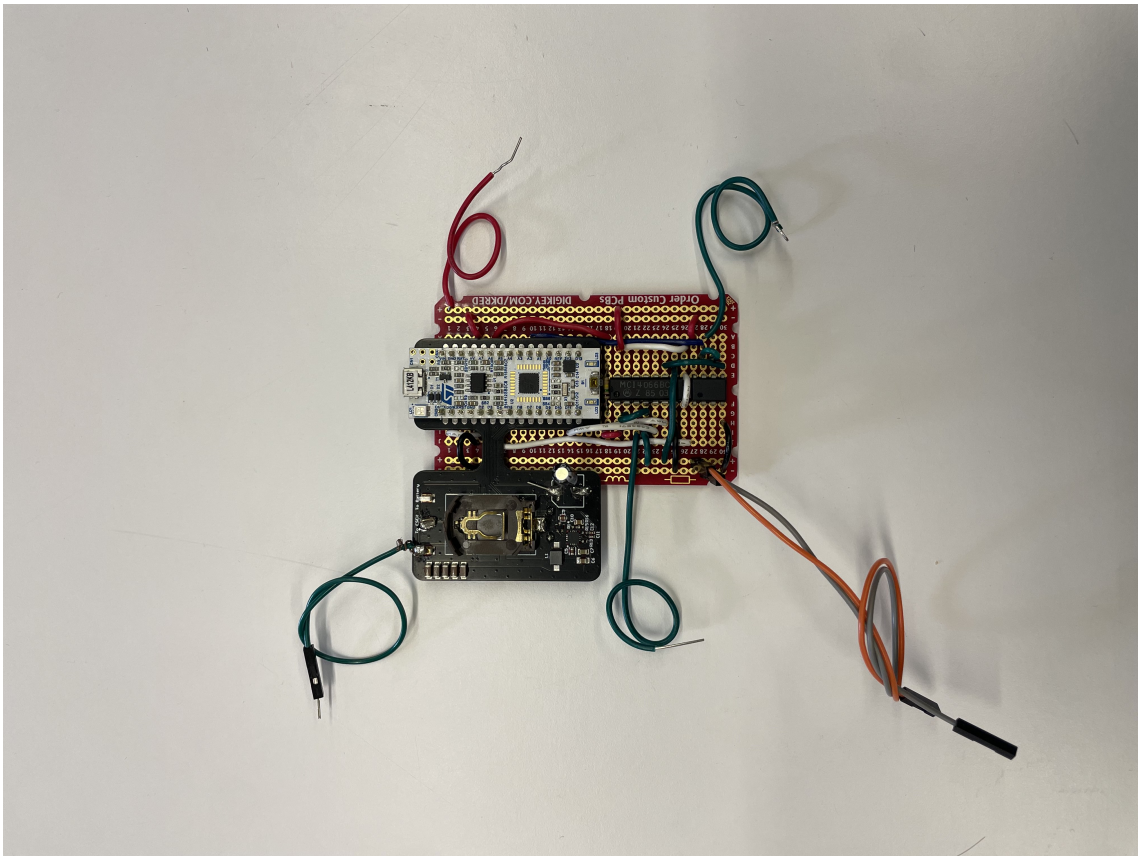

Figure 3.28: Updated Ciruit on Stripboard

The following details the connections and components used in the circuit:

- **Power Supply:** The red cable at the upper-left is connected to a 15V power supply, typically provided by a buck-boost converter that draws from a battery.
- **Output to ADC:** The upper-right green cable functions as the output to the analog-to-digital converter. This connection passes through a Unity gain buffer, enabling oscilloscope readings.
- **Ground Connection:** A ground test point is established via the lower-left black cable, which facilitates grounding to the oscilloscope.
- **Reimbursement Phase Output:** The lower-middle green cable represents the output for the reimbursement phase, handling voltage transitions.
- **Variable Capacitor Connections:** The variable capacitor is connected through two cables; the orange cable links to the upper terminal, while the

grey cable provides a ground connection.

Mathematical Rationale

The fundamental idea is that the capacitance value influences the charging peak during the brief pulse duration:

$$V(t) = V_{max}(1 - e^{-\frac{t}{RC}}) \tag{3.6}$$

where $V(t)$ is the voltage across the capacitor at time t, V_{max} is the maximum voltage, R is the resistance through which the capacitor is charged, and C is the capacitance.

For a pulse duration (Δt) of 20 microseconds:

- At minimum capacitance (C_{min}), the capacitor charges quickly, approaching V_{max}, making the observed voltage almost equal to the peak pulse voltage.
- At maximum capacitance (C_{max}), the same duration results in a significantly lower voltage, $V(t)$, due to the slower charging rate.

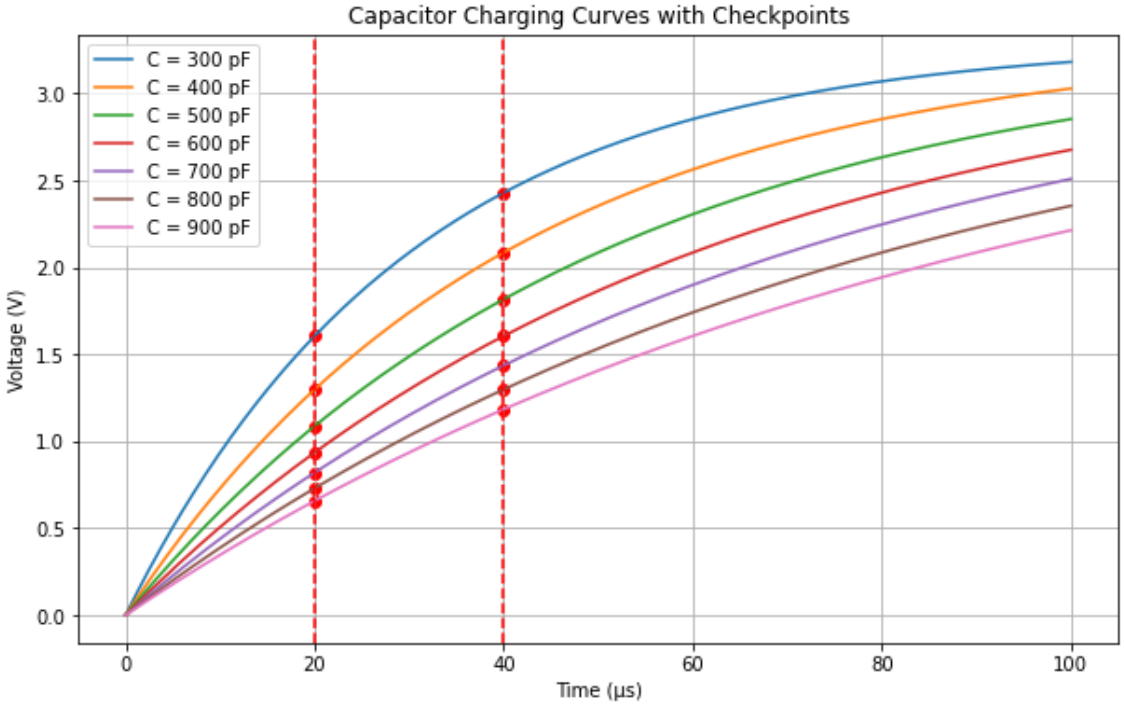

Figure 3.29: Peak Values of Different Capacitance Values at pulse times of 20 and 40 $\mu seconds$

Table 3.3: Capacitor Voltages at 20 μs and 40 μs for Various Capacitances

Capacitance (pF)	Voltage at 20 μs (V)	Voltage at 40 μs (V)
300	1.61	2.43
400	1.30	2.09
500	1.09	1.82
600	0.94	1.61
700	0.82	1.44
800	0.73	1.30
900	0.66	1.18

This charging behavior is predominantly linear along the initial, steeper part of the exponential charge curve, allowing for straightforward interpretation of the capacitor's state (maximum or minimum value).

To maximize energy efficiency, the pulse duration was meticulously set to 5 microseconds. The figure below illustrates the dynamic relationship between the voltage and capacitance during this brief energy pulse.

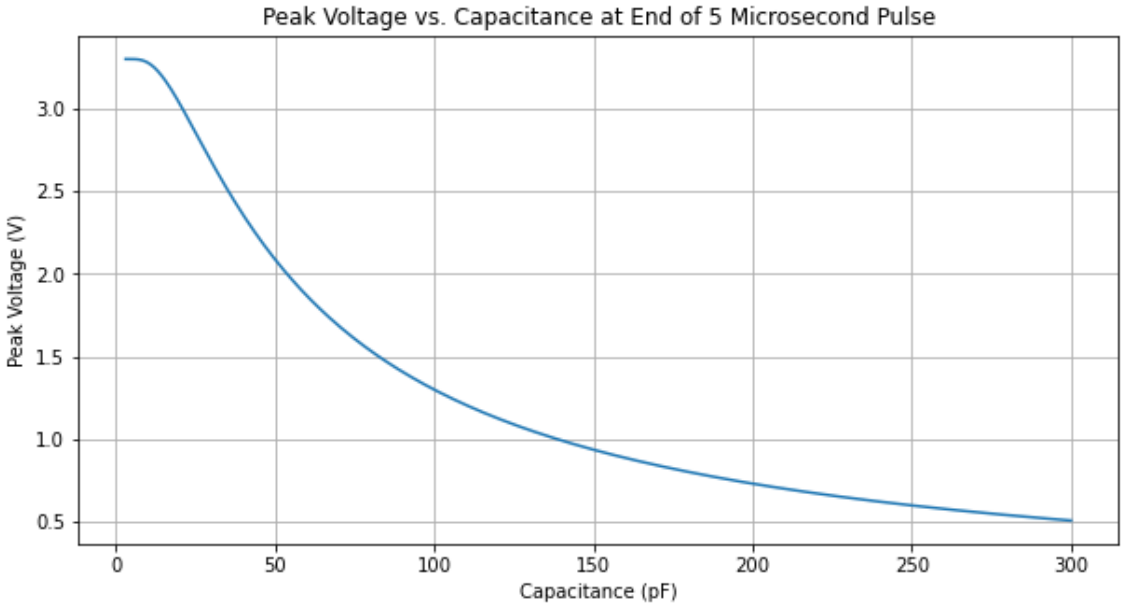

Figure 3.30: Dynamic Voltage-Capacitance Relationship During a 5 Microsecond Pulse

Measurement in the Microcontroller

To capture this dynamic, an ADC converter paired with Direct Memory Access (DMA) is configured to start just before the pulse generation:

- The ADC continuously converts the voltage across the capacitor during the pulse and stores these values directly into memory and then turns off until the next pulse to conserve power.
- The system then selects the maximum value from these samples, which corresponds to the peak or near-peak voltage, indicative of the capacitor's charge state at that moment.

This method not only simplifies the system by reducing the need for external timing or comparative threshold mechanisms but also enhances the energy efficiency of the measurement process by minimizing active measurement time and relying on passive data collection and analysis.

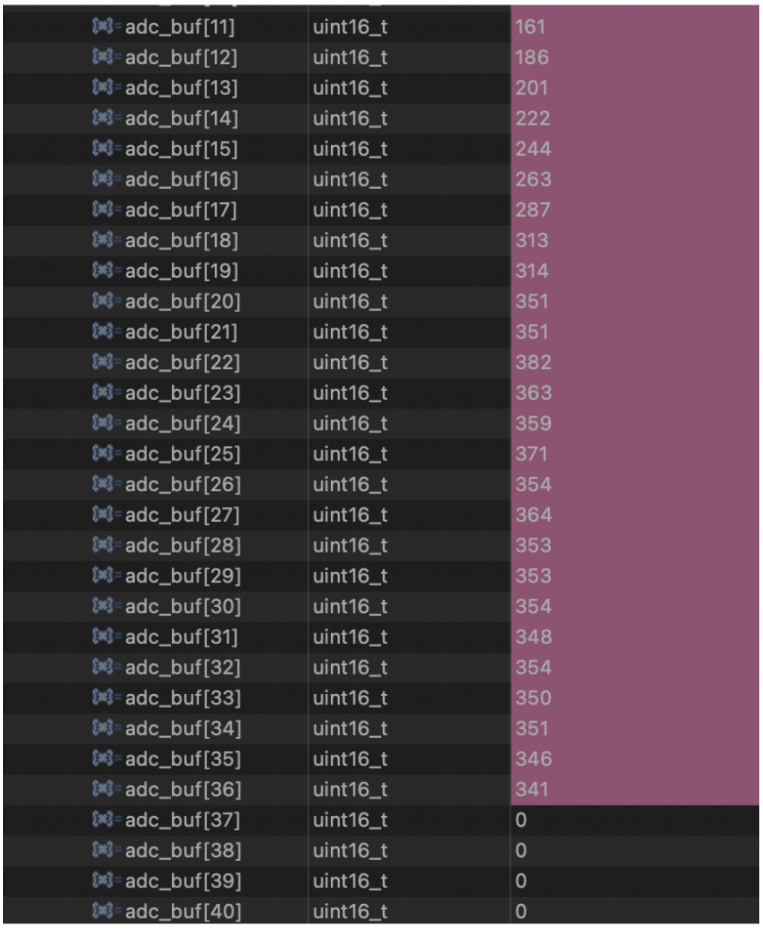

Figure 3.31: Collection of Buffer Measurements taken from the capacitance as viewed in CubeMX

3.8 Programming and System Configuration

The software controlling the system is developed in C, utilizing the HAL library for hardware abstraction. The code is structured to manage four primary states: INVESTMENT, HARVESTING, REIMBURSEMENT, and RECOVERY, which govern the operational phases of the energy harvesting process.

The configuration of peripherals includes ADC initialization, Direct Memory Access setup for continuous data transfer, and timer configuration for managing microsecond delays during pulse generation.

The ADC buffer length is defined to accommodate the data throughput required during the recovery phase, ensuring that no data is lost during conversion:

```
#define ADC_BUF_LEN 79 // Buffer size for ADC data
```

The code generally follows this principle shown in figure 3.32.

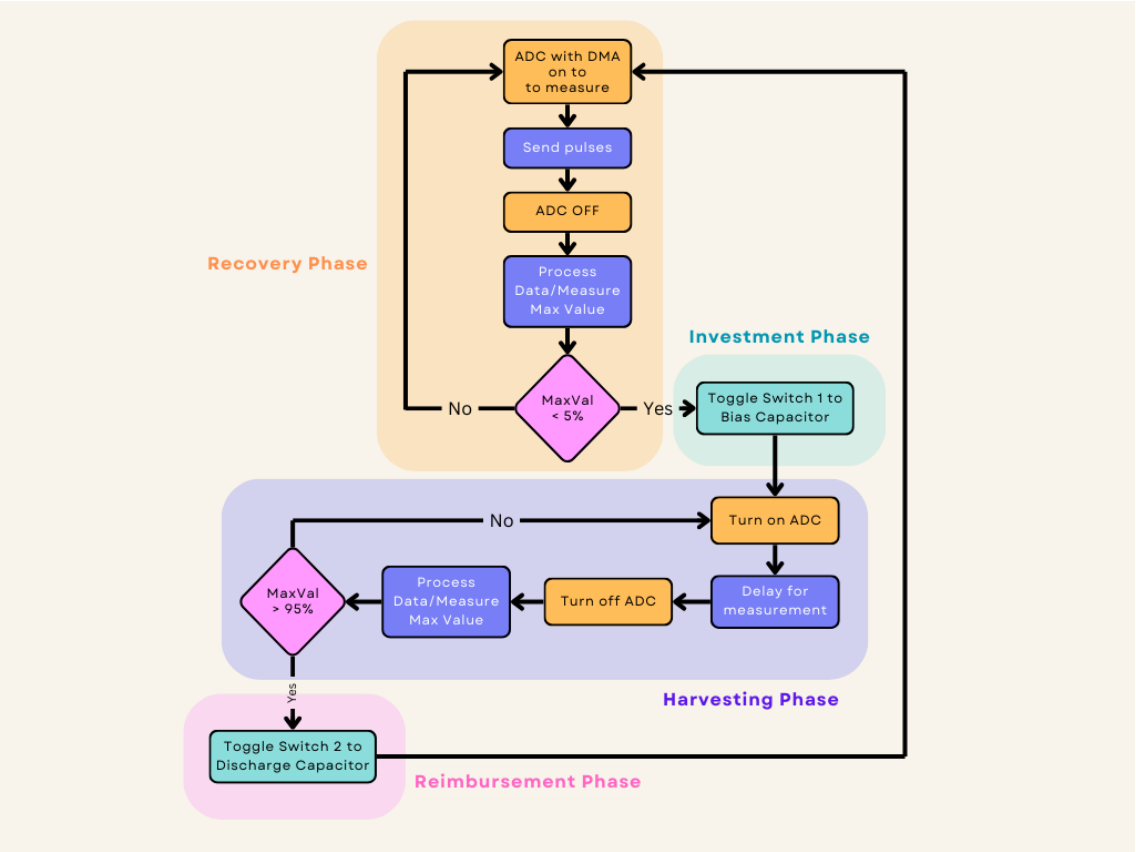

Figure 3.32: Pseudo Code for the Switching and Measurement Logic

Algorithm 1 Energy Harvesting System Phases

Start
Recovery Phase:
Send pulses
ADC with DMA to measure
ADC OFF
Process Data/Measure Max Value
if MaxVal < 95% **then**
    Go to Investment Phase
else
    Stay in Recovery Phase
end if
Investment Phase:
Toggle Switch 1 to Bias Capacitor
Turn on ADC
Delay for measurement
if MaxVal > 95% **then**
    Go to Harvesting Phase
else
    Go to Recovery Phase
end if
Harvesting Phase:
Process Data/Measure Max Value
if MaxVal > 95% **then**
    Stay in Harvesting Phase
else
    Go to Reimbursement Phase
end if
Reimbursement Phase:
Toggle Switch 2 to Discharge Capacitor
Return to Recovery Phase
End

Check appendix A for full implementation of the algorithm.

Challenges in System Development

One of the primary challenges we initially faced involved the methodology for measuring capacitance effectively. Initially, we employed an External Interrupt Service Routine (EXTI) to trigger an interrupt, signaling the microcontroller when the pulse began, followed by using a timer to measure the decay time. This approach,

initially suggested in the early stages of the project, introduced unnecessary complexity into the system architecture. In collaboration with teaching assistant Morgan Diepart, we explored alternative methods that would simplify the system. We decided to directly initiate a DMA ADC conversion just before generating the pulse and stop it immediately after a microdelay of 20 microseconds. This adjustment significantly streamlined the process by eliminating the need for an external interrupt and reducing the system's overhead.

Further into the project, we encountered issues with buffer management during continuous conversion mode, where the ADC was set to continuously convert the input signal over 20 microseconds. Initially, we had defined the buffer size as 4096 to maximize memory use. However, it became apparent that only the first 36 entries were utilized, making such a large buffer unnecessary. After several iterations, we adjusted the buffer size down to 79, which was just over twice the size of the entries used, and this helped in accurately capturing the data without overloading the buffer.

During the debugging phase in the lab, I confirmed that the buffer was updating values correctly, with variations in the ADC Data Register (DR) reflecting changes in the capacitor values accurately. However, we faced a significant challenge with the callback function intended to select and update the `maxValue` based on the largest value recorded. The system failed to enter this callback function, which was critical for processing the data accurately. We resolved this by measuring the value directly inside the main loop, allowing for global access to this crucial data without needing to rely on interrupt-driven callbacks.

Additionally, we considered using an Analog watchdog on a second ADC to keep the system in ultra-low power mode until the voltage across the capacitor surpassed a preset threshold, which would then trigger a switch to the reimbursement phase. However, introducing a second ADC caused unpredictable behavior in the predefined ADC configurations, leading to unrealistic data outputs. This erratic behavior led us to decide against using an analog watchdog. Instead, we modified our approach to measure the voltage across the capacitor similarly to the recovery phase. Rather than sending pulses, we introduced a switch to briefly activate the ADC, allowing for an accurate measurement without significantly discharging the capacitor.

Chapter 4

Results and Discussion

This chapter presents the experimental results obtained from a series of tests designed to explore the performance of an electrostatic energy harvester under varying electrical conditions. The primary focus was on measuring the voltage across the parallel plates of a variable capacitor during different phases of the energy harvesting cycle. To achieve accurate voltage measurements and minimize charge loss, a unity gain buffer in parallel with the capacitor and ground was utilized.

The experiments were structured to provide insights into the energy conversion efficiency and the dynamic behavior of the capacitor in response to controlled changes in voltage and plate separation. Each phase of the harvesting cycle—investment, harvesting, and reimbursement—was analyzed to understand the impact of operational parameters on the system's performance. Special attention was given to the role of the unity gain buffer in stabilizing the measurement setup and its effect on the overall system efficiency.

This investigation not only sheds light on the operational characteristics of the electrostatic energy harvester but also addresses the challenges encountered in real-world applications, such as voltage saturation and unexpected behavior of the operational amplifier. The following sections detail the setup, methodology, and outcomes of the experiments, providing a comprehensive overview of the findings and their implications for the design and optimization of energy harvesting systems.

Case 1: Measurement Circuit with Continuous Switch Activation, Excluding Reimbursement Phase

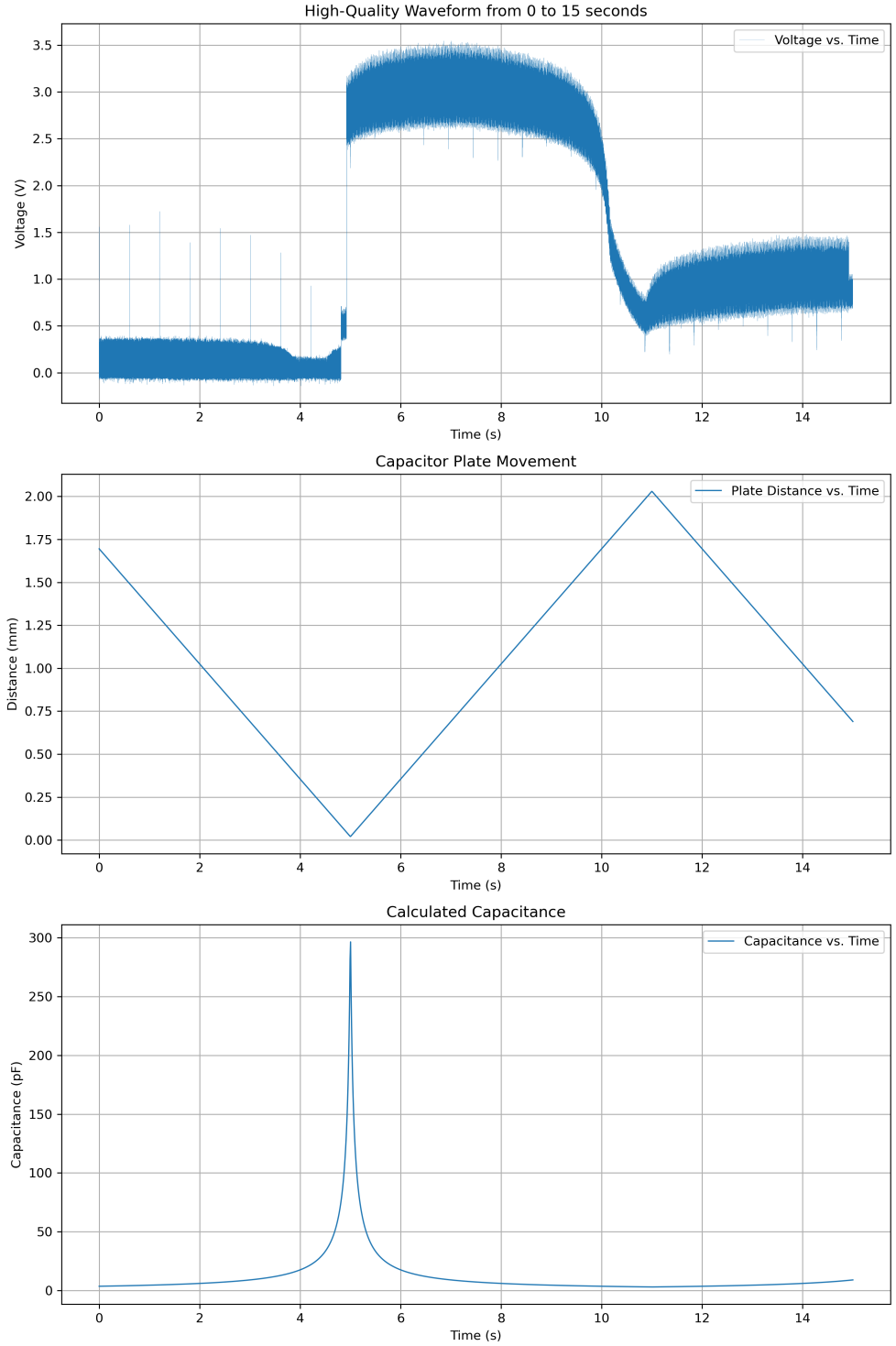

Figure 4.1: Observations of voltage, distance, and capacitance variations during the harvesting phase

This configuration excludes the reimbursement phase and incorporates the recovery, investment, and harvesting phases. The linear movement of the plates, as orchestrated by the test bench, causes an inverse variation in capacitance with decreasing distance between the plates.

Spikes noted from 0 to 5 seconds are due to microcontroller-initiated pulses designed to test capacitance by monitoring voltage peaks. Optimal capacitance is typically detected at 5 seconds when the voltage dips below a predefined threshold. However, the timing of the pulses every 500 ms may result in missing the precise moment of peak capacitance.

The contrast between the rapid change in capacitance as the plate distance reduces to 0.02mm and its slower decrease as it expands to 2.04mm could explain the abrupt rise in voltage to 2.8V around 4.8 seconds, likely influenced by mechanical movements of the test bench and the limitations of a breadboard setup.

From 5 to 11 seconds, as the distance between the plates linearly increases, the capacitance sharply falls from nearly 300pF to approximately 5pF, leading to a progressive voltage decline until the plates reach their furthest separation. As the plates reconverge, the voltage gradually increases, although it does not attain the previous maximum levels.

Concerns regarding the experimental setup include the 470 Ohms on-resistance of the analog switch and the unity gain buffer's $2V/\mu s$ slew rate, which may impact the peak voltage measurements.

To facilitate more linear changes in capacitance, it is necessary to modify the actuation pattern of the plates.

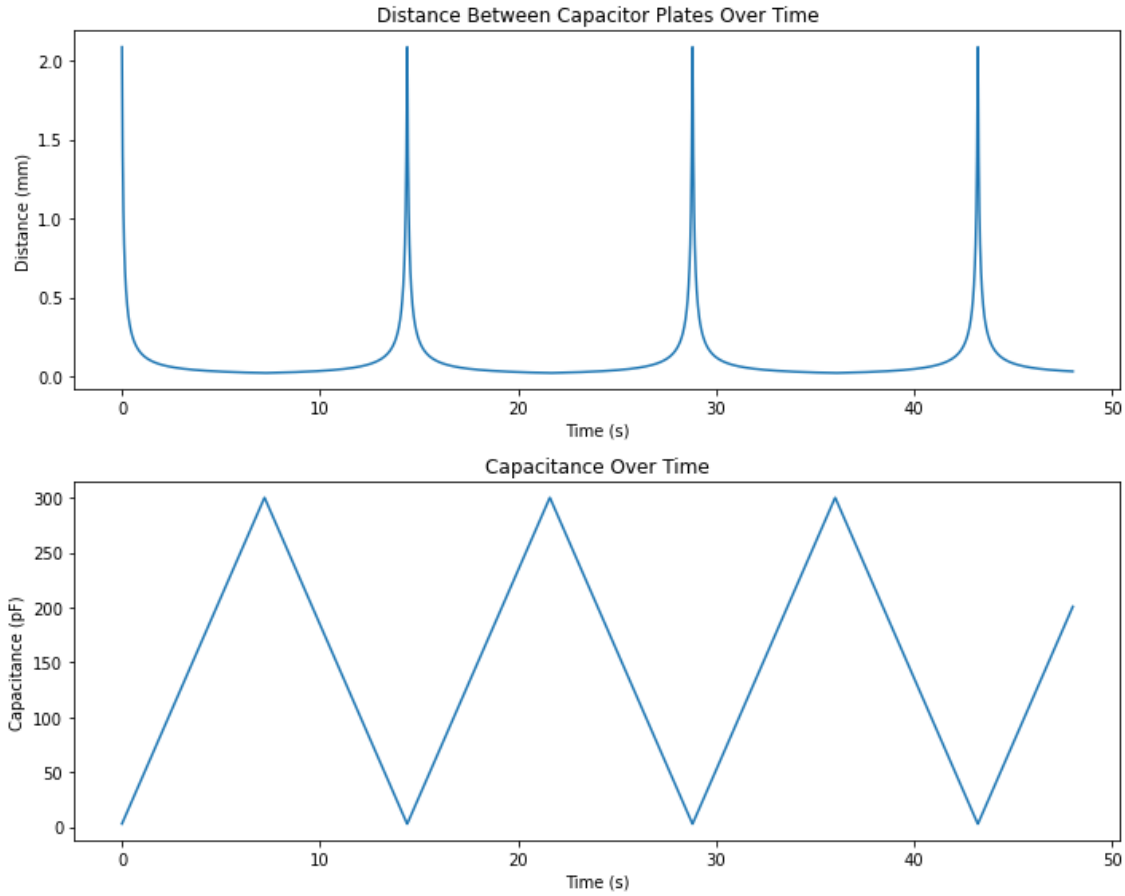

Figure 4.2: Graphical representation of capacitance versus time and displacement over time

Adopting a sinusoidal motion for the plates will provide a more accurate model for simulating natural processes such as breathing, enhancing our ability to predict the extremes in capacitance.

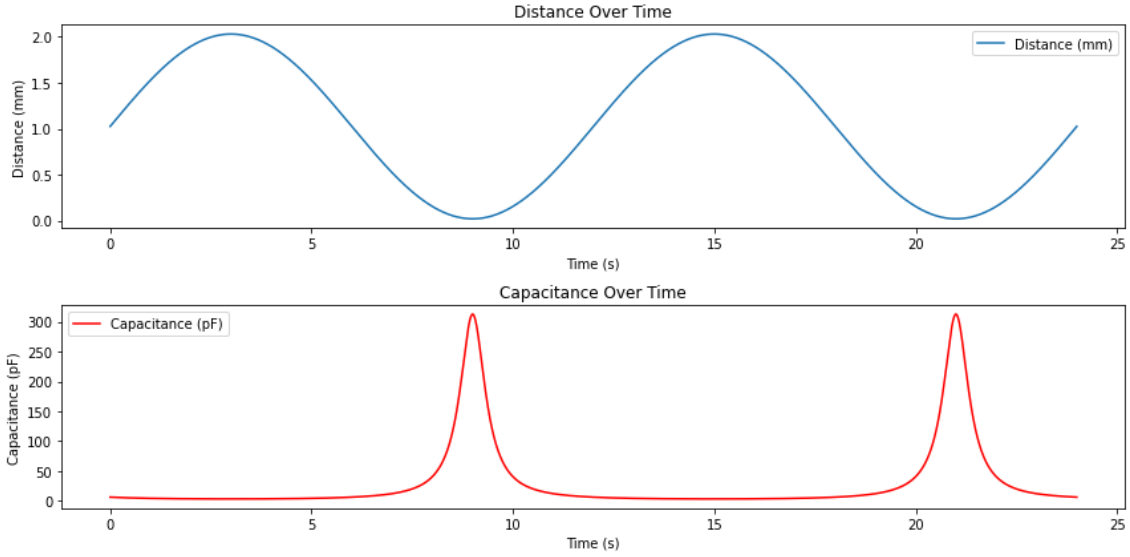

Figure 4.3: Illustration of sinusoidal plate movement over time

This approach allows for a slightly greater margin in predicting minimum and maximum capacitance values accurately. It is proposed that transitioning to the investment phase slightly before the capacitance reaches its lowest point could optimize energy harvesting. This adjustment might require initial calibration but could potentially be automated in future designs.

Case 2: Measurement Circuit with Constant Switch Activation, Targeting the Investment Phase

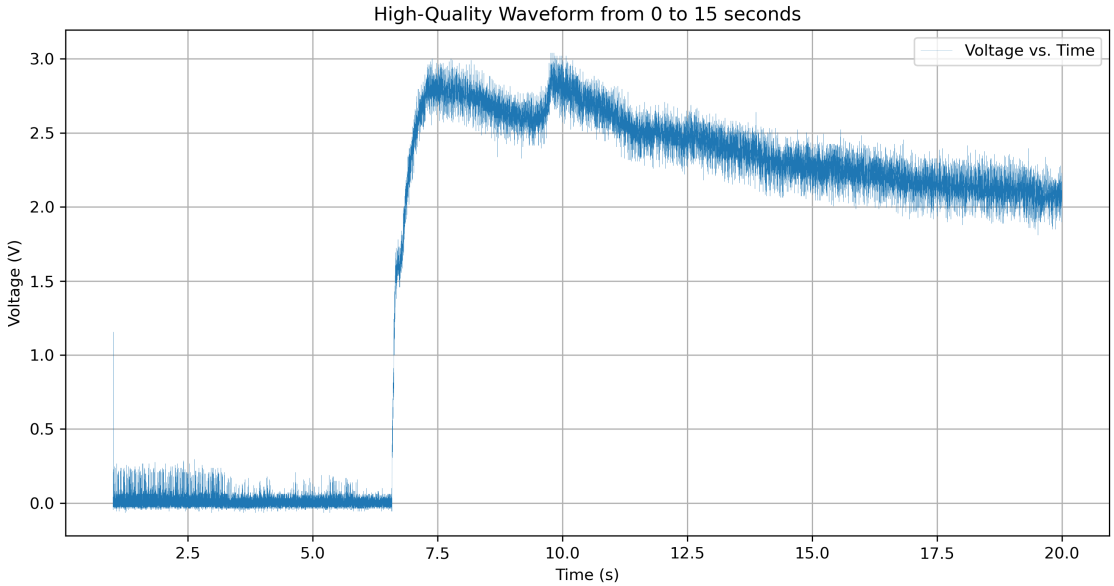

Figure 4.4: Voltage across parallel plates against time

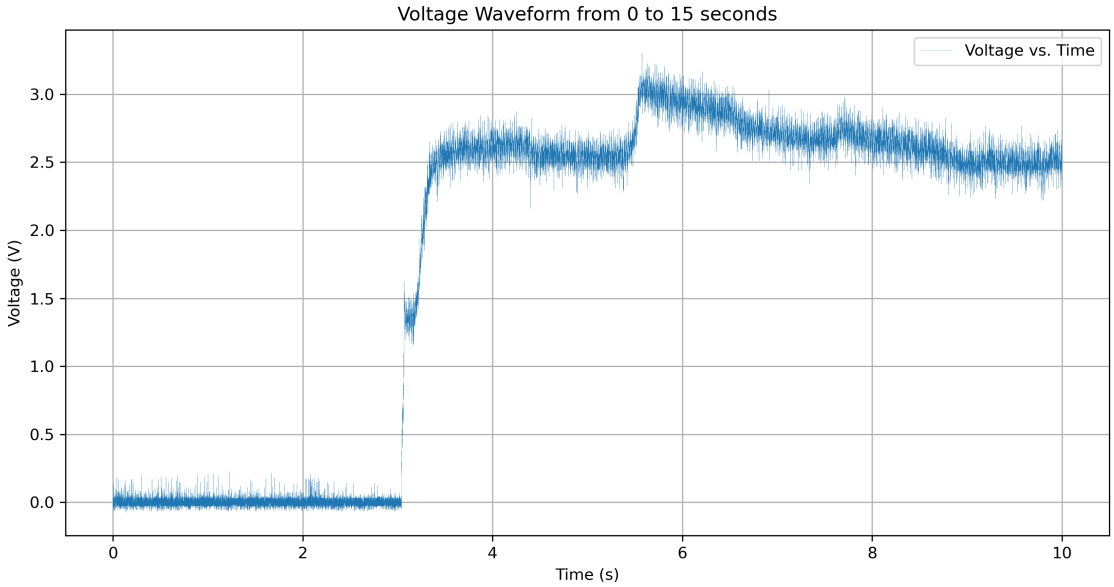

Figure 4.5: Re-measurement of voltage across parallel plates

In this case, the capacitor was intentionally biased 0.2 seconds before reaching its peak capacitance to leverage the rapid change in capacitance values. This strategic timing enabled the capacitor to be charged at approximately 6.7 seconds, initially reaching a voltage of about 1.7V. However, as the capacitance decreased, the harvested energy peaked at 2.8V. Notably, voltage fluctuations post-peak were observed, which were primarily influenced by the movement of the plates affecting energy retention and re-harvesting capabilities.

The amount of energy harvested in this case considering the primary charge input of 1.7V can be found using the expression:

$$E = \frac{1}{2}CV^2$$

where:

- E is the energy in joules (J),
- C is the capacitance in farads (F),
- V is the voltage across the capacitor in volts (V).

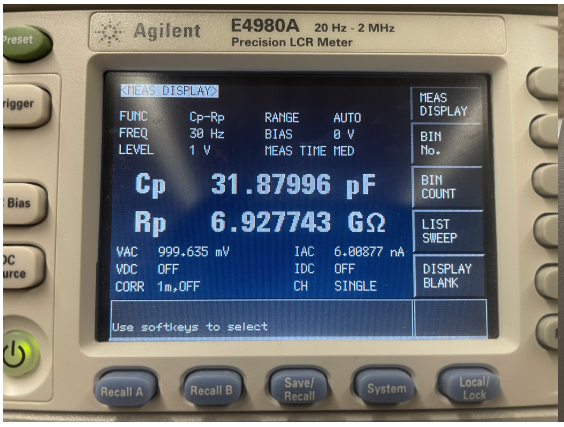

Figure 4.6: Minimum Capacitance value obtained at maximum plate distance

Figure 4.7: Maximum Capacitance value obtained at initial charging

Initial Energy Storage

$$E_{\text{initial}} = \frac{1}{2} \times 107.2194 \times (10)^{-}12 \times (1.7)^2$$

$$E_{\text{peak}} = 154.932\text{pJ}$$

Energy at Peak Voltage

$$E_{\text{peak}} = \frac{1}{2} \times 31.87996 \times (10)^{-}12 \times (2.8)^2$$

$$E_{\text{peak}} = 124.969\text{pJ}$$

Energy Harvested: The change in energy stored due to the change in capacitance and voltage.

$$E_{\text{harvested}} = E_{\text{peak}} - E_{\text{initial}}$$

$$E_{\text{harvested}} = 124.969\text{pJ} - 154.932\text{pJ}$$

$$E_{\text{harvested}} = -29.963\text{pJ}$$

This trial demonstrated poor energy harvesting, yet it also highlighted a limitation: the capacitor did not achieve the maximum charging voltage of 3.3V. This may be

attributed to the inherent resistance of the variable capacitor and the operational characteristics of the switch.

As the analog switch is powered with a 5V DC supply, the voltage was naturally upper limited by the supply of the switch forcefully preventing it from increasing beyond this value, the resistance within the switch, the resistance of the capacitor, parasitic capacitances and other elements potentially causing voltage drops.

Furthermore, powering the switch with a 15V supply unexpectedly resulted in a continuous 5V output from the Unity gain buffer, even when the input was supposed to be inactive. This issue likely stemmed from a floating input to the op-amp, causing it to output its rail voltage. To mitigate this, an additional terminal was added to the switch to connect the input of the Unity gain buffer to the ground, thus setting its input to 0V when not in use. Unfortunately, this modification did not resolve the issue as the voltage still slowly charged the capacitor on its own. This behavior is unusual and was not explained by the datasheet.

A potential reason could be **leakage currents** within the switch or the circuitry, which can occur due to imperfections in the semiconductor material or design.

Case 3: Measurement Circuit with Constant Switch Activation, Including Reimbursement Phase

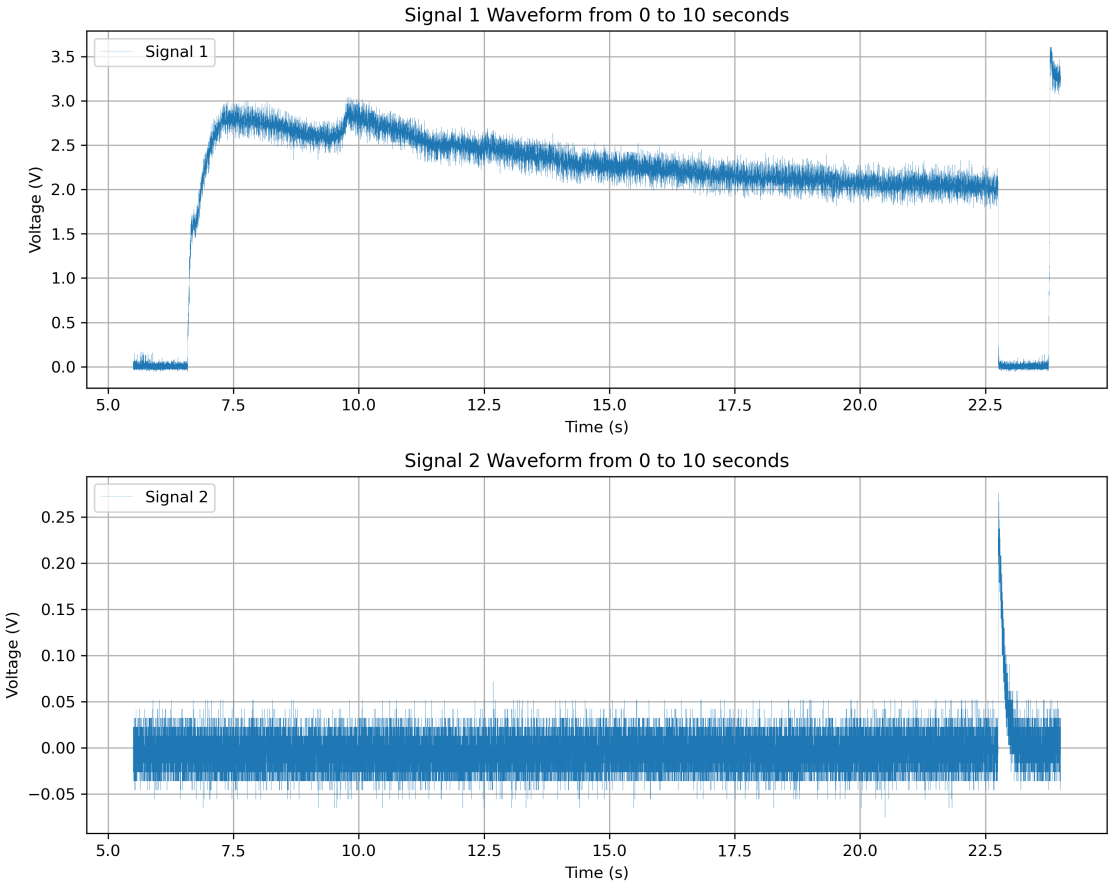

Figure 4.8: Voltage across parallel plates including reimbursement phase

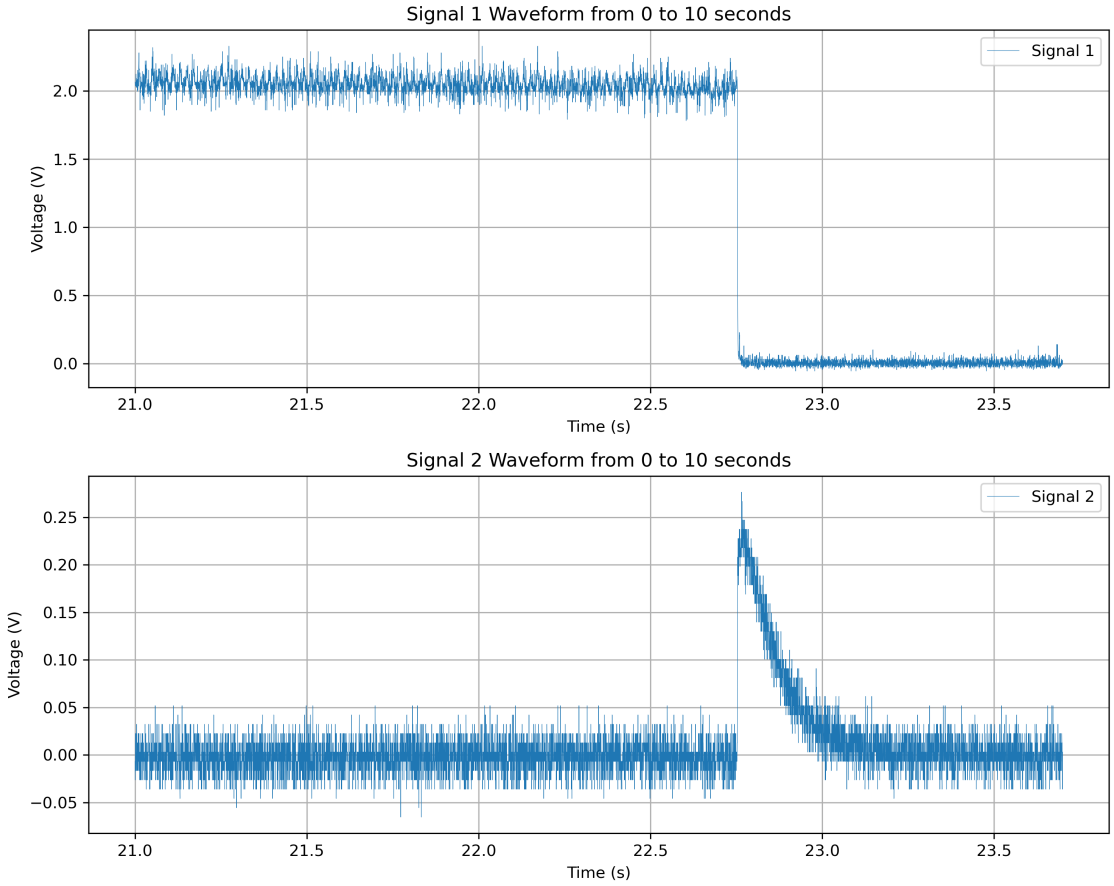

Figure 4.9: Voltage across parallel plates during reimbursement phase, re-measured

This experiment included a reimbursement phase, implemented after the harvesting phase, to examine its effectiveness. The reimbursement phase charged a 10nF capacitor, which was connected between the variable capacitor and the system ground. Typically, charging through a resistor is standard to ensure a rapid response and minimal energy loss.

The fact of not having a resistor to charge the capacitor can be seen as it does not appropriately charge, it only reaches a maximum voltage of 0.25V which is an inefficient transfer of energy.

This can be explained with **Voltage drop under load**: If the power supply cannot maintain its voltage under high current conditions, the voltage output might drop significantly when connected directly to the capacitor.

When you charge a capacitor directly from a voltage source without any series

resistance, the capacitor should theoretically charge up to the voltage of the source. The voltage across a capacitor C charged to a charge Q and voltage V is described by:

$$V = \frac{Q}{C}$$

When a voltage source is connected, it attempts to immediately establish its voltage across the capacitor. Since the initial voltage across the capacitor (uncharged) is 0V, and a capacitor resists sudden changes in voltage, the instantaneous current I that would theoretically flow is given by:

$$I = C\frac{dV}{dt}$$

Where $\frac{dV}{dt}$ is the rate of change of voltage across the capacitor. Without a resistor, the initial $\frac{dV}{dt}$ is theoretically infinite because the voltage source tries to instantly raise the capacitor's voltage from 0V to its own voltage. Thus, the initial current should also be theoretically infinite. In practice, this is limited by the internal resistance and inductance of the capacitor and the source's maximum current capability.

Case 4: Measurement Circuit with Constant Switch Activation, Excluding Reimbursement Phase, Targeting the Investment Phase 0.2 Seconds Before Peak Capacitance, 5V Switch Power Supply

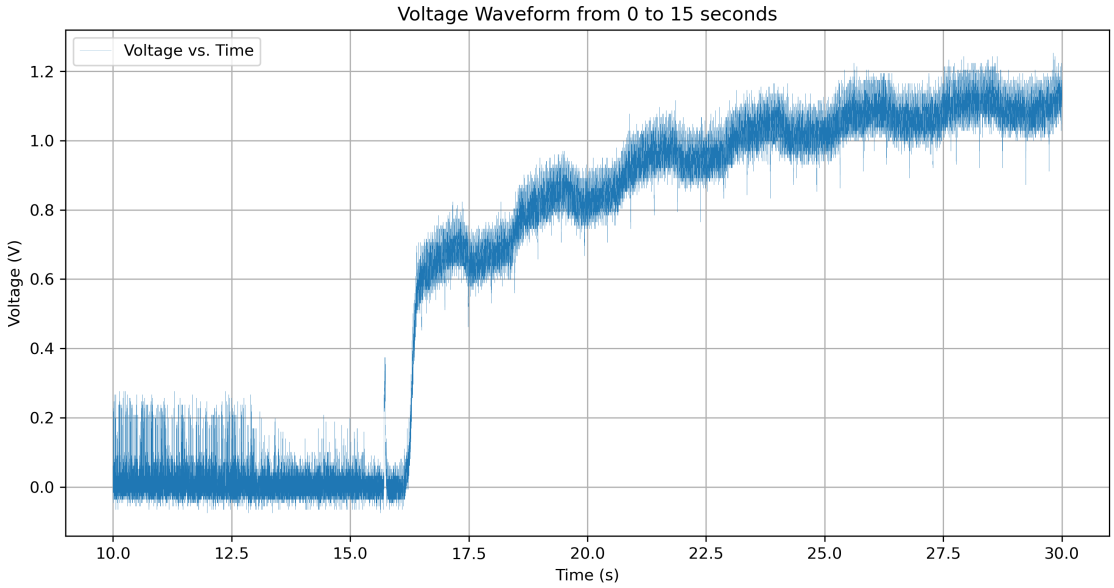

Figure 4.10: Voltage Across Parallel Plates Over Time

In this scenario, the analog switch's supply voltage (VDD) was set to 5V, corresponding to the maximum supply voltage of the microcontroller. This configuration initially limited the charging voltage to 0.6V, affecting the amount of energy harvested as the capacitance fluctuated between its maximum and minimum values. Notable energy gains were observed at regular intervals, with key transitions occurring at 17.5, 18.75, and 20 seconds.

Chapter 5

Conclusion

This research has culminated in several pivotal achievements and illuminated various challenges inherent in designing and implementing an electrostatic energy harvester (ESEH) suitable for implantable medical devices. The journey from concept to prototype has provided invaluable insights into the dynamics of energy harvesting and its application in biomedical engineering.

Achievements and Observations

A welcome achievement of this project was the development of an efficient method for measuring capacitance values that do not require the capacitor to be fully charged. This approach offers a rapid and effective solution for environments where capacitance values fluctuate moderately, making it superior to other methods in specific low-resolution applications.

During the testing phases, it was confirmed that the ESEH system functions as anticipated, capable of harvesting energy as described in the literature. However, the dynamic nature of the capacitance, especially its rapid inverse variation relative to the proximity of the plates, poses a challenge. This behavior suggests a high likelihood of missing the peak capacitance moment, potentially underutilizing the available energy differential from maximum to minimum capacitance values. This limitation can significantly impact the overall energy harvesting efficiency if the switching does not occur at the optimal moment.

Another challenge encountered was the operational amplifier's rail output, which restricted the testing of the energy harvester to its full potential. This limitation hindered the accurate measurement of the system's efficiency, as the harvester was

unable to extract as much energy as the configuration of the switch and unity gain buffer would permit.

Furthermore, when the operational amplifier output reached rail voltage, it inadvertently supplied the microcontroller's GPIO inputs with a damaging 10V—far exceeding their maximum tolerance of 3.3V. This excessive voltage not only destroyed the Nucleo board but also led to its complete disconnection from the Cube MX software during the experiment.

The issues were compounded by the constraints of the pre-existing PCB design, especially regarding the ADC configuration. The design's approach was to continuously monitor the charge in both the variable capacitor and the storage capacitor using a voltage divider circuit connected to the ADC. This setup resulted in significant charge dissipation from the capacitors.

To mitigate these problems, we introduced an additional switching and measurement circuit that activates only when necessary. This circuit utilizes a unity gain buffer to provide high input impedance and low output impedance. This adjustment allows for precise voltage measurements without significant charge depletion from the capacitors, thereby preserving the integrity of the energy harvesting process.

System Limitations and Design Constraints

One critical issue encountered was the limitation imposed by the switch's power supply, which acted as a ceiling, restricting the system's potential to harvest energy beyond a certain threshold. Attempts to increase this threshold to potentially boost the energy harvested led to unexpected outcomes, such as forcing the operational amplifier to output at rail voltage, which suggests possible damage or malfunction. This incident necessitated a complete system rebuild to continue the experiments.

Future Directions and Recommendations

Given the observations and challenges faced, future work should focus on redesigning the circuit to incorporate findings from this project. Adjustments should include a reliable method to power the operational amplifier for measuring the variable capacitor without inducing rail voltage outputs.

Further investigation is also required into the noisy signals introduced by the test bench, potentially caused by rapid switching of the stepper motor. This noise

interfered with accurate signal readings, and prevented smooth and neat plots.

Concluding Thoughts

While the system has not yet reached optimal efficiency, the potential for significant energy harvesting exists, as evidenced by the observed capabilities and system behavior under various test conditions. The journey of developing this ESEH has provided a solid foundation for future research and development in the field of energy harvesting for biomedical devices. The lessons learned from the challenges and the innovative solutions developed in response will undoubtedly contribute to advancing this technology towards practical implementation and eventual commercialization.

Addendum

Should further tests post-rebuild provide new insights or significantly different results, an addendum will be added to this thesis to reflect the updated findings, ensuring that the evaluation presented remains accurate and comprehensive up to the point of the oral defense.

Bibliography

[1] Capacitor lab basics simulation. `https://phet.colorado.edu/sims/html/capacitor-lab-basics/latest/capacitor-lab-basics_all.html`, 2024.

[2] S. H. Daneshvar, M. Yuce, and J.-M. Redouté. *Design of Miniaturized Variable-Capacitance Electrostatic Energy Harvesters*. Springer International Publishing, 2022.

[3] Mariachiara Di Cesare, Pablo Perel, Simon Taylor, Chodziwadziwa Kabudula, Honor Bixby, Thomas A. Gaziano, Derek V. McGhie, Jacqueline Mwangi, Brian Pervan, Jagat Narula, David Pineiro, and Fausto J. Pinto. The heart of the world. *Global Heart*, 19(1):11, 2024.

[4] Aleksandr Vostrukhin and Elena Vakhtina. Microcontroller measuring converter of capacitance based on transients in rc circuit. In *Proceedings of the International Scientific Conference "Engineering for Rural Development"*, pages 171–176, 2020.

[5] M. O'Hara, M. Ghoneim, J. Hinrichs, M. Mehta, and E. J. Wright. Psychological consequences of surgery. *Psychosomatic Medicine*, 51:356–370, 1989.

[6] F. Harerimana, H. Peng, M. Otobo, F. Luo, M. Gikunda, J. Mangum, V. LaBella, and P. Thibado. Efficient circuit design for low-power energy harvesting. *AIP Advances*, 10:105006, 2020.

[7] Shad Roundy, Paul Kenneth Wright, and Jan M. Rabaey. *Energy Scavenging for Wireless Sensor Networks: with Special Focus on Vibrations*. Springer, 2004.

[8] Paul D. Mitcheson, Eric M. Yeatman, G. Kondala Rao, Andrew S. Holmes, and Tim C. Green. Energy harvesting from human and machine motion for wireless electronic devices. *Proceedings of the IEEE*, 96:1457–1486, 2008.

[9] K. Konno and J. Mead. Measurement of the separate volume changes of rib cage and abdomen during breathing. *Journal of Applied Physiology*, 22:407–422, 1967.

[10] K. J. Sarro, A. Silvatti, A. Aliverti, and R. Barros. Proposition and evaluation of a novel method based on videogrammetry to measure three-dimensional rib motion during breathing. *Journal of Biomechanics*, 42:2494–2500, 2009.

[11] C. Dagdeviren, Z. Li, and Z. L. Wang. Energy harvesting from the animal/human body for self-powered electronics. *Annual Review of Biomedical Engineering*, 19:85–108, 2017.

[12] A. Erturk and D.J. Inman. Piezoelectric energy harvesting. *John Wiley Sons*, 2011.

[13] David Morgan. *Advanced Materials for Energy Harvesting Systems*. Academic Press, 2021.

[14] Hao Chang and W. Su. Design and development of a high-performance tensile-mode piezoelectric energy harvester based on a three-hinged force-amplification mechanism. *Smart Materials and Structures*, 2022.

[15] Z.L. Wang. Triboelectric nanogenerators as new energy technology. *Nano Energy*, 70:104201, 2020.

[16] V. Leonov, C. Van Hoof, and R.J.M. Vullers. Thermal energy harvesting for application in autonomous devices. *IEEE Transactions on Industrial Electronics*, 56(11):4306–4312, 2009.

[17] Charles Bradford and Samantha Lee. *Thermal Energy Harvesting for Wearable Technology*. Springer, 2022.

[18] S. Roundy, P.K. Wright, and J. Rabaey. Energy scavenging for wireless sensor networks with special focus on vibrations. *The Springer International Series in Engineering and Computer Science*, 792, 2005.

[19] J. Miao et al. Mems-based variable capacitors for energy harvesting: Non-resonant designs and applications. *Journal of Micromechanics and Microengineering*, 2006.

[20] J. Miao et al. Energy harvesting from mechanical vibrations using variable capacitors. *Journal of Micromechanics and Microengineering*, 2004.

[21] M. Seo et al. Advanced microfabrication techniques for mems-based energy harvesters. *Sensors and Actuators A: Physical*, 2015.

[22] H. Li et al. High dielectric materials in mems energy harvesters. *Advanced Functional Materials*, 2016.

[23] O. Sobianin et al. The efficacy of mems energy harvesters in medical devices. *Biomedical Microdevices*, 2022.

[24] N. Kaur et al. Design optimization of mems-based variable capacitors for energy harvesting. *Microsystem Technologies*, 2021.

[25] J.R. Miller and P. Simon. Materials science. electrochemical capacitors for energy management. *Science*, 321:651–652, 2008.

[26] Y. Zhu, S. Murali, M.D. Stoller, K.J. Ganesh, W. Cai, P.J. Ferreira, A. Pirkle, R.M. Wallace, K.A. Cychosz, M. Thommes, D. Su, E.A. Stach, and R.S. Ruoff. Carbon-based supercapacitors produced by activation of graphene. *Science*, 332:1537–1541, 2011.

[27] A. Amari, F. Touchard, C. Dieppedale, S. Grondel, and C. Bergaud. A review of power management strategies in hybrid energy storage systems. *Renewable and Sustainable Energy Reviews*, 105:138–151, 2019.

[28] A.D. Smith, S. Savagatrup, B.R. Watson, J.D. Azoulay, and T.M. Swager. Emerging applications for conjugated polymers in novel capacitor architectures. *Polymer Chemistry*, 3:2743–2751, 2012.

[29] S. H. Daneshvar, M. Maymandi-Nejad, M. Yuce, and J.-M. Redouté. A variable-capacitance energy harvester with miniaturized inductor targeting implantable devices. *IEEE Transactions on Industrial Electronics*, 69:475–484, 2021.

[30] N. Mendiratta, G. Singh, N. Chattoraj, and S. Kundu. Optimization of capacitor for piezoelectric energy harvesting. In *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–6, 2018.

[31] B. Bieske, G. Kropp, and A. Rolapp. Testing electrostatic energy harvesters: A new topology for accurate characterization. In *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 331–336, 2017.

[32] A. Stein, A. Sarin, and H. Hofmann. Theoretical comparison of energy harvesting methods for electret-free variable-capacitance devices. *Energy Harvesting and Systems*, 3:245–262, 2016.

Appendix A

Full Program Implementation

```
1
2
3
4       /* USER CODE BEGIN Header */
5   /* USER CODE END Header */
6   /* Includes
        ------------------------------------------------------------------*/
7   #include "main.h"
8
9   /* Private includes
        ----------------------------------------------------------*/
10  /* USER CODE BEGIN Includes */
11  #include <stdio.h>
12  #include <string.h>
13  #include <stdbool.h>
14  /* USER CODE END Includes */
15
16  /* Private typedef
        -----------------------------------------------------------*/
17  /* USER CODE BEGIN PTD */
18  // Enumerate system states
19  typedef enum {
20      INVESTMENT,
21      HARVESTING,
22      REIMBURSEMENT,
23      RECOVERY
24  } SystemState;
25  /* USER CODE END PTD */
26
27  /* Private define
        ------------------------------------------------------------*/
28  /* USER CODE BEGIN PD */
```

```
29 #define ADC_BUF_LEN 79 //Recovery phase buffer
30 #define MAX_CAP_THRESHOLD 3000  // Set based on observed values
31 #define MIN_CAP_THRESHOLD 200    // Set at 200 to switch just
      before reaching the maximum
32 #define EPSILON 20          // Small buffer to prevent frequent
      toggling near threshold values
33 //#define SAMPLES 5  // Number of samples for averaging
34
35 /* USER CODE END PD */
36
37 /* Private macro
      -------------------------------------------------------------*/
38 /* USER CODE BEGIN PM */
39 /* USER CODE END PM */
40
41 /* Private variables
      ---------------------------------------------------------*/
42 ADC_HandleTypeDef hadc1;
43 DMA_HandleTypeDef hdma_adc1;
44
45 TIM_HandleTypeDef htim2;
46 TIM_HandleTypeDef htim16;
47
48 UART_HandleTypeDef huart2;
49
50 /* USER CODE BEGIN PV */
51 uint16_t adc_buf[ADC_BUF_LEN];
52 volatile uint16_t maxValue;
53 volatile uint16_t maxVal;
54 volatile uint16_t HarvestVoltage;
55 volatile uint32_t smoothedValue;
56 SystemState system_state = RECOVERY;  // Initial state
57 uint32_t adcValues[SAMPLES];
58 uint8_t sampleIndex = 0;
59 bool bufferFull = false;
60
61 /* Variable to report ADC analog watchdog status:   */
62 /*   RESET <=> voltage into AWD window   */
63 /*   SET   <=> voltage out of AWD window */
64 volatile uint8_t  ubAnalogWatchdogStatus = RESET;  /* Set into
      analog watchdog interrupt callback */
65 /* USER CODE END PV */
66
67 /* Private function prototypes
      -----------------------------------------------*/
68 void SystemClock_Config(void);
69 static void MX_GPIO_Init(void);
70 static void MX_DMA_Init(void);
71 static void MX_USART2_UART_Init(void);
```

```
72 static void MX_TIM2_Init(void);
73 static void MX_TIM16_Init(void);
74 static void MX_ADC1_Init(void);
75 /* USER CODE BEGIN PFP */
76 /* USER CODE END PFP */
77
78 /* Private user code
   ---------------------------------------------------------*/
79 /* USER CODE BEGIN 0 */
80 void MicroDelay(uint16_t microseconds)
81 {
82     __HAL_TIM_SET_COUNTER(&htim16, 0);  // if htim16 is the timer
       instance
83     __HAL_TIM_ENABLE(&htim16);
84     while (__HAL_TIM_GET_COUNTER(&htim16) < microseconds);
85     __HAL_TIM_DISABLE(&htim16);
86 }
87
88 void GeneratePulse(void) {
89     // Start ADC conversion
90     HAL_ADC_Start_DMA(&hadc1, (uint32_t*)adc_buf, ADC_BUF_LEN);
91     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET); //Connect
       Op amp output to ADC
92     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET); // Start
       charging the capacitor
93
94       //Start the ADC and take many samples with the DMA
95     MicroDelay(10); // Charge for a very short time
96     //Check the values and choose the largest value
97     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET); //
       Discharge the capacitor
98     // Stop ADC conversion
99
100    HAL_ADC_Stop_DMA(&hadc1);
101    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
       //disconnect Op amp output to ADC
102    processData();
103    HAL_Delay(500); //Delay before sending the next pulse
104
105 }
106
107 void processData() {
108     maxVal = 0;
109     for (int i = 0; i < 38; i++) {
110         if (adc_buf[i] > maxVal) maxVal = adc_buf[i];
111
112     }
113    uint16_t max = 1 + ((maxVal - 1200) * (1000 - 1)) / (1800 -
       1200); //Values set based on observed maximum and minimum
```

```
114     if (max <= 0){
115         max = 0;
116     }
117 
118     maxValue = max;
119     if (maxValue > 4000){
120         maxValue = 9999; //For Debugging
121     }
122 
123 }
124 
125 
126 
127 void Update_GPIO_States(SystemState state) {
128     switch (state) {
129     //System states for use on the PCB directly. They are unused
        here as the PCB design is not optimal
130     case RECOVERY:
131         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
132         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
133         break;
134     case INVESTMENT:
135         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
136         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
137         break;
138     case HARVESTING:
139         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
140         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
141         break;
142     case REIMBURSEMENT:
143         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
144         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
145         break;
146     }
147 }
148 //Function to smooth out ADC values
149 uint32_t getSmoothedValue(uint32_t newValue) {
150     adcValues[sampleIndex] = newValue;
151     sampleIndex++;
152     if (sampleIndex >= SAMPLES) {
153         sampleIndex = 0;
154         bufferFull = true;
155     }
156 
157     uint32_t sum = 0;
158     uint32_t count = bufferFull ? SAMPLES : sampleIndex;
159     for (uint8_t i = 0; i < count; i++) {
160         sum += adcValues[i];
161     }
```

```
162
163     return sum / count;  // Return the average
164 }
165 /* USER CODE END 0 */
166
167 /**
168   * @brief  The application entry point.
169   * @retval int
170   */
171 int main(void)
172 {
173   /* USER CODE BEGIN 1 */
174
175   /* USER CODE END 1 */
176
177   /* MCU
          Configuration--------------------------------------------------------*/
178
179   /* Reset of all peripherals, Initializes the Flash interface
          and the Systick. */
180   HAL_Init();
181
182   /* USER CODE BEGIN Init */
183   /* USER CODE END Init */
184
185   /* Configure the system clock */
186   SystemClock_Config();
187
188   /* USER CODE BEGIN SysInit */
189   /* USER CODE END SysInit */
190
191   /* Initialize all configured peripherals */
192   MX_GPIO_Init();
193   MX_DMA_Init();
194   MX_USART2_UART_Init();
195   MX_TIM2_Init();
196   MX_TIM16_Init();
197   MX_ADC1_Init();
198   /* USER CODE BEGIN 2 */
199
200
201
202
203   /* USER CODE END 2 */
204
205   /* Infinite loop */
206   /* USER CODE BEGIN WHILE */
207   while (1)
208     {
```

```
209         switch (system_state) {
210             case RECOVERY:
211 
212                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5,
                        GPIO_PIN_RESET); //Keep reimbursement closed
213                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12,
                        GPIO_PIN_SET); // Switch for charging the
                        capacitor
214                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
                        GPIO_PIN_SET);  // Measurement is enabled
215 
216                 GeneratePulse();
217                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12,
                        GPIO_PIN_RESET); // Switch for charging the
                        capacitor
218                 if (maxValue <= (MIN_CAP_THRESHOLD + EPSILON)) {
219                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
                            GPIO_PIN_RESET); // Isolate measurement
                            circuit
220                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6,
                            GPIO_PIN_RESET); //Disconnect Opamp from
                            ADC to prevent damage
221                     system_state = INVESTMENT;
222                 }
223                 break;
224 
225             case INVESTMENT:
226 
227                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7,
                        GPIO_PIN_SET);  // Start biasing capacitor
228                 HAL_Delay(50);  // Ensure full charge
229                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12,
                        GPIO_PIN_RESET); // Isolate capacitor
230                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7,
                        GPIO_PIN_RESET);  // Turn off biasing
231 
232                 system_state = HARVESTING;
233                 break;
234 
235             case HARVESTING:
236 //                HAL_ADC_Start_DMA(&hadc1, (uint32_t*)adc_buf,
        ADC_BUF_LEN);
237                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
                        GPIO_PIN_SET);  // Enable measurement
238                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6,
                        GPIO_PIN_RESET); //Keep ADC from damage
239                 HAL_Delay(1000); //Wait for a while to harvest
240                 system_state = REIMBURSEMENT;
241                 break;
```

```
242
243             case REIMBURSEMENT:
244                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5,
                        GPIO_PIN_RESET); // Discharge into storage
                        capacitor
245                 HAL_Delay(100);  // Ensure discharge
246                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5,
                        GPIO_PIN_RESET); // Turn off discharge
247                 system_state = RECOVERY;
248                 HAL_Delay(1000);
249                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9,
                        GPIO_PIN_RESET); //Measurement is disabled
250                 break;
251         }
252
253         // General GPIO or control updates
254         Update_GPIO_States(system_state);
255
256     /* USER CODE END WHILE */
257
258     /* USER CODE BEGIN 3 */
259     }
260   /* USER CODE END 3 */
261 }
262
263 /**
264   * @brief System Clock Configuration
265   * @retval None
266   */
267 void SystemClock_Config(void)
268 {
269   RCC_OscInitTypeDef RCC_OscInitStruct = {0};
270   RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
271
272   /** Configure the main internal regulator output voltage
273   */
274   if
        (HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1)
        != HAL_OK)
275   {
276     Error_Handler();
277   }
278
279   /** Configure LSE Drive Capability
280   */
281   HAL_PWR_EnableBkUpAccess();
282   __HAL_RCC_LSEDRIVE_CONFIG(RCC_LSEDRIVE_LOW);
283
284   /** Initializes the RCC Oscillators according to the specified
```

```
     parameters
285    * in the RCC_OscInitTypeDef structure.
286    */
287    RCC_OscInitStruct.OscillatorType =
         RCC_OSCILLATORTYPE_LSE|RCC_OSCILLATORTYPE_MSI;
288    RCC_OscInitStruct.LSEState = RCC_LSE_ON;
289    RCC_OscInitStruct.MSIState = RCC_MSI_ON;
290    RCC_OscInitStruct.MSICalibrationValue = 0;
291    RCC_OscInitStruct.MSIClockRange = RCC_MSIRANGE_10;
292    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
293    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
294    {
295      Error_Handler();
296    }
297
298    /** Initializes the CPU, AHB and APB buses clocks
299    */
300    RCC_ClkInitStruct.ClockType =
         RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
301                                  |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
302    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_MSI;
303    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
304    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
305    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
306
307    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) !=
         HAL_OK)
308    {
309      Error_Handler();
310    }
311
312    /** Enable MSI Auto calibration
313    */
314    HAL_RCCEx_EnableMSIPLLMode();
315  }
316
317  /**
318    * @brief ADC1 Initialization Function
319    * @param None
320    * @retval None
321    */
322  static void MX_ADC1_Init(void)
323  {
324
325    /* USER CODE BEGIN ADC1_Init 0 */
326
327    /* USER CODE END ADC1_Init 0 */
328
329    ADC_MultiModeTypeDef multimode = {0};
```

```
330   ADC_ChannelConfTypeDef sConfig = {0};
331
332   /* USER CODE BEGIN ADC1_Init 1 */
333
334   /* USER CODE END ADC1_Init 1 */
335
336   /** Common config
337   */
338   hadc1.Instance = ADC1;
339   hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
340   hadc1.Init.Resolution = ADC_RESOLUTION_12B;
341   hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
342   hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
343   hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
344   hadc1.Init.LowPowerAutoWait = DISABLE;
345   hadc1.Init.ContinuousConvMode = ENABLE;
346   hadc1.Init.NbrOfConversion = 1;
347   hadc1.Init.DiscontinuousConvMode = DISABLE;
348   hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
349   hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
350   hadc1.Init.DMAContinuousRequests = ENABLE;
351   hadc1.Init.Overrun = ADC_OVR_DATA_PRESERVED;
352   hadc1.Init.OversamplingMode = DISABLE;
353   if (HAL_ADC_Init(&hadc1) != HAL_OK)
354   {
355     Error_Handler();
356   }
357
358   /** Configure the ADC multi-mode
359   */
360   multimode.Mode = ADC_MODE_INDEPENDENT;
361   if (HAL_ADCEx_MultiModeConfigChannel(&hadc1, &multimode) !=
        HAL_OK)
362   {
363     Error_Handler();
364   }
365
366   /** Configure Regular Channel
367   */
368   sConfig.Channel = ADC_CHANNEL_8;
369   sConfig.Rank = ADC_REGULAR_RANK_1;
370   sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
371   sConfig.SingleDiff = ADC_SINGLE_ENDED;
372   sConfig.OffsetNumber = ADC_OFFSET_NONE;
373   sConfig.Offset = 0;
374   if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
375   {
376     Error_Handler();
377   }
```

```
  /* USER CODE BEGIN ADC1_Init 2 */

  /* USER CODE END ADC1_Init 2 */

}

/**
  * @brief TIM2 Initialization Function
  * @param None
  * @retval None
  */
static void MX_TIM2_Init(void)
{

  /* USER CODE BEGIN TIM2_Init 0 */
  /* USER CODE END TIM2_Init 0 */

  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};
  TIM_IC_InitTypeDef sConfigIC = {0};

  /* USER CODE BEGIN TIM2_Init 1 */
  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 0;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 4294967295;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) !=
     HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_IC_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2,
     &sMasterConfig) != HAL_OK)
  {
    Error_Handler();
```

```
425   }
426   sConfigIC.ICPolarity = TIM_INPUTCHANNELPOLARITY_RISING;
427   sConfigIC.ICSelection = TIM_ICSELECTION_DIRECTTI;
428   sConfigIC.ICPrescaler = TIM_ICPSC_DIV1;
429   sConfigIC.ICFilter = 0;
430   if (HAL_TIM_IC_ConfigChannel(&htim2, &sConfigIC, TIM_CHANNEL_1)
          != HAL_OK)
431   {
432     Error_Handler();
433   }
434   /* USER CODE BEGIN TIM2_Init 2 */
435   /* USER CODE END TIM2_Init 2 */
436
437 }
438
439 /**
440   * @brief TIM16 Initialization Function
441   * @param None
442   * @retval None
443   */
444 static void MX_TIM16_Init(void)
445 {
446
447   /* USER CODE BEGIN TIM16_Init 0 */
448   /* USER CODE END TIM16_Init 0 */
449
450   /* USER CODE BEGIN TIM16_Init 1 */
451   /* USER CODE END TIM16_Init 1 */
452   htim16.Instance = TIM16;
453   htim16.Init.Prescaler = 7;
454   htim16.Init.CounterMode = TIM_COUNTERMODE_UP;
455   htim16.Init.Period = 65535;
456   htim16.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
457   htim16.Init.RepetitionCounter = 0;
458   htim16.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
459   if (HAL_TIM_Base_Init(&htim16) != HAL_OK)
460   {
461     Error_Handler();
462   }
463   /* USER CODE BEGIN TIM16_Init 2 */
464   /* USER CODE END TIM16_Init 2 */
465
466 }
467
468 /**
469   * @brief USART2 Initialization Function
470   * @param None
471   * @retval None
472   */
```

```
473 static void MX_USART2_UART_Init(void)
474 {
475
476   /* USER CODE BEGIN USART2_Init 0 */
477   /* USER CODE END USART2_Init 0 */
478
479   /* USER CODE BEGIN USART2_Init 1 */
480   /* USER CODE END USART2_Init 1 */
481   huart2.Instance = USART2;
482   huart2.Init.BaudRate = 115200;
483   huart2.Init.WordLength = UART_WORDLENGTH_8B;
484   huart2.Init.StopBits = UART_STOPBITS_1;
485   huart2.Init.Parity = UART_PARITY_NONE;
486   huart2.Init.Mode = UART_MODE_TX_RX;
487   huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
488   huart2.Init.OverSampling = UART_OVERSAMPLING_16;
489   huart2.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
490   huart2.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
491   if (HAL_UART_Init(&huart2) != HAL_OK)
492   {
493     Error_Handler();
494   }
495   /* USER CODE BEGIN USART2_Init 2 */
496   /* USER CODE END USART2_Init 2 */
497
498 }
499
500 /**
501   * Enable DMA controller clock
502   */
503 static void MX_DMA_Init(void)
504 {
505
506   /* DMA controller clock enable */
507   __HAL_RCC_DMA1_CLK_ENABLE();
508
509   /* DMA interrupt init */
510   /* DMA1_Channel1_IRQn interrupt configuration */
511   HAL_NVIC_SetPriority(DMA1_Channel1_IRQn, 0, 0);
512   HAL_NVIC_EnableIRQ(DMA1_Channel1_IRQn);
513
514 }
515
516 /**
517   * @brief GPIO Initialization Function
518   * @param None
519   * @retval None
520   */
521 static void MX_GPIO_Init(void)
```

```
522 {
523   GPIO_InitTypeDef GPIO_InitStruct = {0};
524 /* USER CODE BEGIN MX_GPIO_Init_1 */
525 /* USER CODE END MX_GPIO_Init_1 */
526
527   /* GPIO Ports Clock Enable */
528   __HAL_RCC_GPIOC_CLK_ENABLE();
529   __HAL_RCC_GPIOA_CLK_ENABLE();
530   __HAL_RCC_GPIOB_CLK_ENABLE();
531
532   /*Configure GPIO pin Output Level */
533   HAL_GPIO_WritePin(GPIOA,
      SW4_Pin|Pulse_Out_Pin|GPIO_PIN_8|Outswitch2_Pin
534                         |GPIO_PIN_10|OutSwitch1_Pin,
                             GPIO_PIN_RESET);
535
536   /*Configure GPIO pin Output Level */
537   HAL_GPIO_WritePin(GPIOB, SW1_Pin|LD3_Pin|SW3_Pin|CHRG_EN_Pin
538                         |SW2_Pin, GPIO_PIN_RESET);
539
540   /*Configure GPIO pins : SW4_Pin Pulse_Out_Pin PA8 Outswitch2_Pin
541                           PA10 OutSwitch1_Pin */
542   GPIO_InitStruct.Pin =
      SW4_Pin|Pulse_Out_Pin|GPIO_PIN_8|Outswitch2_Pin
543                         |GPIO_PIN_10|OutSwitch1_Pin;
544   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
545   GPIO_InitStruct.Pull = GPIO_NOPULL;
546   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
547   HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
548
549   /*Configure GPIO pins : SW1_Pin LD3_Pin SW3_Pin CHRG_EN_Pin
550                           SW2_Pin */
551   GPIO_InitStruct.Pin = SW1_Pin|LD3_Pin|SW3_Pin|CHRG_EN_Pin
552                         |SW2_Pin;
553   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
554   GPIO_InitStruct.Pull = GPIO_NOPULL;
555   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
556   HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
557
558   /*Configure GPIO pin : PA11 */
559   GPIO_InitStruct.Pin = GPIO_PIN_11;
560   GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
561   GPIO_InitStruct.Pull = GPIO_NOPULL;
562   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
563   GPIO_InitStruct.Alternate = GPIO_AF12_COMP1;
564   HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
565
566 /* USER CODE BEGIN MX_GPIO_Init_2 */
567 /* USER CODE END MX_GPIO_Init_2 */
```

```
568 }
569
570 /* USER CODE BEGIN 4 */
571 //Called when first half of buffer is filled
572 void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc){
573     if (hadc->Instance == ADC1) {
574         HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_10);
575       }
576 }
577
578
579
580 //Called when buffer is completely filled
581 void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc) {
582     if (hadc->Instance == ADC1) {
583         HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_10);
584     }
585 }
586
587 void HAL_ADC_LevelOutOfWindowCallback(ADC_HandleTypeDef* hadc)
588 {
589      /* Set variable to report analog watchdog out of window
           status to main      */
590      /* program.

           */
591      //HarvestVoltage = HAL_ADC_GetValue(&hadc2);
592      ubAnalogWatchdogStatus = SET;
593 }
594
595 /* USER CODE END 4 */
596
597 /**
598   * @brief  This function is executed in case of error occurrence.
599   * @retval None
600   */
601 void Error_Handler(void)
602 {
603   /* USER CODE BEGIN Error_Handler_Debug */
604   /* USER CODE END Error_Handler_Debug */
605 }
606
607 #ifdef  USE_FULL_ASSERT
608 /**
609   * @brief  Reports the name of the source file and the source
        line number
610   *         where the assert_param error has occurred.
611   * @param  file: pointer to the source file name
612   * @param  line: assert_param error line source number
```

```
613   * @retval None
614   */
615 void assert_failed(uint8_t *file, uint32_t line)
616 {
617   /* USER CODE BEGIN 6 */
618   /* USER CODE END 6 */
619 }
620 #endif /* USE_FULL_ASSERT */
```

Listing A.1: HAL C code written for STM32L412KB NucleoBoard for ElectroStatic Energy Harvesting