Delft University of Technology
Master of Science Thesis in Embedded Systems

CardioSync: Heartbeat-Based BLE Synchronization for Batteryless IoT Devices

Arunjunai Rajan Senthil Kumar

CardioSync: Heartbeat-Based BLE Synchronization for Batteryless IoT Devices

Master of Science Thesis in Embedded Systems

Embedded Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Arunjunai Rajan Senthil Kumar

27 August 2023

Author

Arunjunai Rajan Senthil Kumar

Title

CardioSync: Heartbeat-Based BLE Synchronization for Batteryless IoT Devices

MSc Presentation Date

30 August 2023

Graduation Committee

Dr. Przemysław Pawełczak	Delft University of Technology
Dr. Nergis Tömen	Delft University of Technology
Dr. Jasper De Winkel	Delft University of Technology

Abstract

Batteryless Internet of Things (IoT) devices powered by energy harvesting enable sustainable and maintenance-free operation, but face challenges in achieving synchronised bidirectional communication between intermittently-powered nodes. This thesis presents CardioSync, a novel framework that leverages the human heartbeat as a shared clock to synchronise Bluetooth Low Energy (BLE) connections between battery less devices. CardioSync integrates a low-power optical heart rate sensor to capture real-time heartbeat data. A peak detection algorithm identifies distinct heart rate peaks, establishing synchronisation points for connection-setup events. By scheduling BLE advertising and scanning activities timed with detected peaks, CardioSync aligns connection attempts between intermittently-powered devices. This heartbeat-based synchronisation is integrated into the existing FreeBie architecture for intermittent BLE communication. Experimental evaluations demonstrate CardioSync's ability to successfully establish synchronised connections between batteryless nodes, reducing average connection setup time by up to 1.8x compared to an asynchronous FreeBie system. However, these gains incurred increased power consumption due to the integrated sensor. CardioSync enhances FreeBie's capabilities, enabling efficient intermittent-to-intermittent BLE connections. The proposed technique shows promise for advancing body sensor networks through sustainable and maintenance-free operation. Further work should optimise sensor utilisation and explore adaptive synchronisation to improve energy efficiency.

Preface

Motivation for this thesis stems from my academic journey, spanning from my undergrad in Electrical and Electronics Engineering at Thiagarajar College of Engineering, India, to my current pursuit of a Master's in Embedded Systems at Delft University of Technology, along with my professional experiences. Even during my undergraduate studies, I developed a curiosity for battery-free technologies, and my exposure to wireless networking embedded systems during my job before my Masters further fuelled my fascination for Embedded systems. So it was instantly serendipitous when I encountered the research works of Dr. Przemysław Pawełczak and Dr. Jasper De Winkel. Their profound insights into Battery Free Internet of Things (IoT) and Energy Harvesting technologies, along with their contributions to the development and enhancement of software for intermittent embedded systems, resonated with my interests. Engaging in discussions with them unveiled a world of untapped possibilities. Their pioneering research work of the FreeBie technology—an architecture enabling Bluetooth Low Energy (BLE) communication, propelled me to immerse myself in this exhilarating field. Despite beginning this journey with limited knowledge in this realm, the more I delved into the intricate workings of BLE communication and the nuances of the FreeBie architecture, the more impassioned I became. The opportunity to contribute to the advancement of FreeBie's capabilities and address its existing limitations was a clarion call that I couldn't resist.

With profound gratitude,
Arunjunai Rajan Senthil Kumar

Delft, The Netherlands
27th August 2023

Acknowledgement

I would like to extend my heartfelt gratitude to Dr. Jasper De Winkel and Dr. Przemysław Pawełczak for their invaluable guidance, unwavering support, and the exceptional opportunity to work on this thesis under their supervision. Your willingness to share your profound knowledge and insights has been instrumental in honing my skills and shaping the direction of this research. Dr. Jasper De Winkel, your regular meetings and patient assistance, including the skilful soldering sessions, have been indispensable, even if my soldering skills left much to be desired! Your guidance has been a lighthouse, steering this thesis toward completion.

I would also like to express my appreciation to Dr. Nergis Tomen for agreeing to be a committee member, contributing her time and expertise to review and assess this work. I am hopeful that she finds this thesis interesting and insightful.

To the distinguished faculty members in the Embedded Systems department, I am grateful for your dedication to education, for imparting knowledge and for evaluating my assignments, ultimately sharpening my skills during my academic journey.

My heartfelt thanks also go to my parents for their unwavering support, both morally and financially, throughout this academic pursuit. Your encouragement has been a pillar of strength.

I extend my sincere gratitude to all my friends, well-wishers, and particularly my roommates, whose camaraderie and assistance have been a source of comfort during these two years of my master's program. I am indebted to those peers and friends who generously shared their expertise and engaged in brainstorming sessions that fuelled my passion for completing this thesis.

AI Empowering Thesis

Various Artificial Intelligence (AI) tools that significantly empowered the completion of this thesis. These tools were as follows.

- **Chat-GPT**: A powerful AI language model, played a crucial role in refining the content, ensuring its coherence, and improving the overall quality of the thesis report.

- **GitHub Co-Pilot**: Integrated with VSCode, offered support by providing code suggestions, aiding in debugging, and enhancing error-code development efficiency. It also simplified the process of generating plots and data visualisations.

- **elicit.org**: A GPT-3 powered research tool, facilitated the literature survey by discovering and summarising relevant scientific papers and journals.

Contents

Chapter 1

Introduction

In today's interconnected world, the advancement in Internet of Things (IoT) devices is reshaping the way we interact with technology. These devices have become indispensable tools, enabling smart homes, industrial automation, healthcare monitoring, and more. As the demand for IoT devices continues to rise, so does the need for sustainable power solutions to ensure their seamless operation. About 78 million batteries powering IoT devices will be dumped globally every day by 2025 if nothing is done to improve their lifespan [13]. This dire statistic comes from EnABLES, an EU-funded project that's urging researchers and technologists to take action to ensure that batteries outlive the devices they power [13].

Battery Free IoT devices have emerged as a promising solution to address the challenges posed by finite battery life. These devices tap into ambient energy sources or energy harvesting techniques to power their operations, offering a sustainable and maintenance-free approach. This paradigm shift holds the potential to revolutionise various domains, ranging from healthcare to environmental monitoring, by enabling devices to function indefinitely without battery replacement.

Wireless sensor networks, a cornerstone of IoT infrastructure, stand to gain immensely from the advent of battery-less IoT devices. These networks, composed of interconnected sensors, offer unparalleled data collection and monitoring capabilities [2]. However, the reliance on traditional battery-powered sensors often limits their deployment due to the need for frequent maintenance and replacement. The integration of battery free IoT devices into wireless sensor networks presents a game-changing opportunity, promising extended operational lifetimes and reduced maintenance overhead.

Within this landscape, The FreeBie technology, which provides Bluetooth Low Energy (BLE) communication on intermittent battery-free IoT devices, has emerged as a trailblazing solution. FreeBie is the first battery-free active wireless system that sustains bi-directional communication on intermittently harvested energy. With this, FreeBie opens doors to novel applications and possibilities for battery-less IoT devices [12].

1.1 Problem Statement

Despite the immense potential it holds, a significant constraint is that FreeBie only supports intermittent-to-continuous device connections. That is in FreeBie, while the end device is intermittent-powered and battery-free, the BLE hosts to which FreeBie connects still requires continuous power. The central challenge revolves around achieving synchronisation within a fully battery-free architecture. This synchronisation is crucial for enabling the transition from connections with continuously powered hosts to connections with intermittently powered hosts.

1.2 Research Goals

Embracing this challenge, the core focus of this research is on devising a synchronisation mechanism that utilises a shared external signal. This method should seamlessly integrate into the existing FreeBie battery-free infrastructure, enabling intermittent-to-intermittent devices to synchronise and establish connections effectively.

In this thesis, we also introduce a novel concept of using human heart pulses as a shared external signal. This approach, while limited to human body wireless sensor networks, offers substantial benefits within healthcare applications.

In order to achieve this core objective, we outline three key contributions which collectively address the previously stated problem within the FreeBie architecture:

- We design a novel synchronisation algorithm capable of effectively detecting heart rate-based pulses and establish BLE connection.

- We integrate the devised framework into the FreeBie architecture as a supplementary enhancement. This enable Bluetooth end devices and hosts to synchronise using heart rate, significantly reducing connection time and energy.

- Validate the proposed system's effectiveness on intermittently-powered end devices and compare its performance to naive and state of the art systems.

1.3 Thesis Structure

The structure of this report is as follows. Chapter 2 presents related literature and explore them within the domain of battery-less IoT devices and wireless body sensor networks. Building upon the foundation laid out, Chapter 3 outlines the architecture and system design of the integrated solution. The subsequent Chapter 4 details the technical implementation of the proposed methodology. Chapter 5 showcases the results obtained from experimental evaluations and performance analyses. Chapter 6 highlights potential future research avenues that could lead to further advancements for FreeBie and the proposed system. Finally, Chapter 7 concludes this thesis by summarising contributions, and discussing the potential applications of the devised framework.

Chapter 2

Related Work

This chapter delves into an in-depth exploration of existing research and developments in the field of battery-less IoT devices, intermittent connectivity solutions, and the pivotal role of the FreeBie architecture. By examining advancements, challenges, and critical gaps in the literature, this section lays the foundation for understanding the significance of the proposed CardioSync framework.

2.1 Battery-Free IoT for Wireless Sensor Networks

The field of battery-less IoT has garnered significant attention, driven by the promise of sustainable, autonomous operation. This approach offers extended device lifetimes and reduced environmental impact. Researchers and innovators have actively explored diverse energy harvesting techniques to power these devices, enabling applications across various domains such as agriculture, logistics, and environmental monitoring. Early works like the development of wireless sensor networks using simple solar energy harvesters and cost-effective energy storage units [31] laid the groundwork for the subsequent advancements.

A notable milestone was achieved in 2017, the integration of solar energy harvesting chips and super capacitors in a battery-less sensor tag [9] showcased the successful integration of multiple sensors, including temperature, humidity, and gas sensors, with efficient data transmission through BLE communication. This advancement proved particularly promising for industrial applications, hinting at the viability of battery-less IoT in various sectors. In 2020 with the introduction of a system utilising ambient RF energy to power battery-less tags [8] highlighted the potential of harnessing ubiquitous energy sources for practical applications. Intriguingly, the employment of a piezoelectric converter as an energy harvester to power a Bluetooth board through low-voltage vibration electromagnetic conversion was explored [10]. This exploration validates the spectrum of solutions available to enhance the battery-less capability of IoT devices while still facilitating the formation of sensor networks.

A distinct stride was taken towards achieving batteryless communication

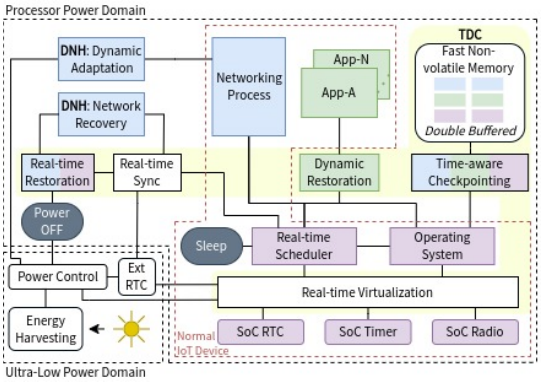


Figure 2.1: **FreeBie Architecture. Figure reproduced from [12].**

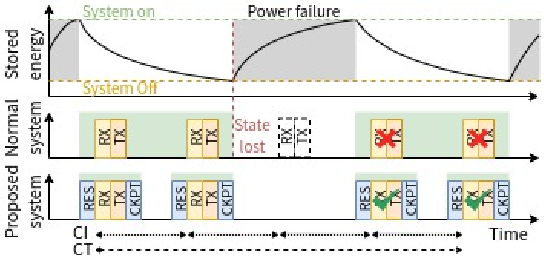


Figure 2.2: **Conceptual illustration of intermittently-powered communication device operation. In normal system, a power outage results in a disruption of the connection and necessitates additional handshakes to reestablish and configure the connection. The FreeBie system has a checkpointing and restoration mechanism that utilises nonvolatile memory for the purpose of storing network state and sustain connections, even in the event of a power outage. CT: connection timeout; CI: connection interval; RES: state restore; CKPT: state checkpoint; RX/TX: reception/transmission. Figure reproduced from [12].**

through the successful design and testing of a wireless LoRaWAN end sensor node [7]. This innovative approach demonstrates the feasibility of battery-less IoT even in long-range communication scenarios, further expanding the scope of its potential applications.

These strides exemplify only a subset of the numerous breakthroughs within the domain of battery-free IoT wireless communication. The works [30], [11], [14] offer additional evidence of the expanding landscape of battery-free IoT. Collectively, these advancements illuminate the dynamic landscape of battery-free IoT, highlighting its capacity to revolutionise myriad domains, from conventional industries to cutting-edge technologies.

2.1.1 The FreeBie

In the realm of intermittent connectivity solutions, the FreeBie architecture emerges as a pivotal contender, offering a distinctive approach to achieving Bluetooth Low Energy (BLE) communication on intermittently-powered wireless devices [12]. This architecture, shown in Figure 2.1, introduces an adaptive framework that tailors connection parameters according to the available harvested energy, facilitating efficient communication in resource-constrained environments. One of its significant features is to maintain BLE connections despite intermittent power, as illustrated in Figure 2.2. Its unique ability to enable bidirectional communication and dynamically manage network connections fills a critical gap in the domain of intermittently-powered devices. Also it supports preemptive scheduling, allowing network processes to take precedence over application or operating system processes.

These components within FreeBie architecture serve a critical role on achieving the battery less operation:

- **FRAM (Ferroelectric RAM):** FRAM is a non-volatile memory technology that offers high read/write speeds and low power consumption. It stores data and program code, ensuring persistence during power cycles. FRAM preserves memory sections during each checkpoint and stores context across separate allocated regions for OS processes, Network processes, and Application processes.

- **External RTC (Real-Time Clock):** The external RTC provides accurate timekeeping even during device power-off periods. This time reference is vital for synchronisation and event scheduling. The external RTC remains always powered through onboard capacitors and can control processor power domains using the Power switch and Power control module.

- **Time-Deterministic Checkpointing and Restore:** This component maintains data integrity by regularly capturing the device's state. In cases of power disruptions, the system can restore to a known state, preventing data loss and ensuring system consistency. It leverages External RTC and FRAM for uninterrupted atomic operations.

  - *Real-time RTC sync:* This subcomponent synchronises the Ext. RTC with the onboard RTC during each real-time operation resumption from power-off state. It uses FRAM to store the synchronising time T_{Sync} in the OS context.
  - *Real-time/Dynamic Restoration:* Upon resuming from power-off state, this component restores processes from FRAM for OS processes, network processes, and real-time application processes, as scheduled by the scheduler.
  - *Dynamic Handling of Network Connection:* Responsible for network recovery and dynamic network adaptation, this subcomponent ensures network recovery in unexpected power-offs and adapts to energy conditions while maintaining connection.

- **Power Control:** Governing power state transitions, power control mechanisms by managing power-on timings, task execution, and low-power modes.
- **Super Capacitors and Energy Harvesters:** Integral to FreeBie's energy autonomy, super capacitors and energy harvesters contribute to storing and harnessing energy from ambient sources.

However, since the architecture relies on external components such as FRAM and an external RTC, it impacts factors like system cost, size, and energy consumption. To address this, the authors suggest future exploration into developing a FreeBie version that integrates next-generation System on Chip (SoC) technology and leverage more energy-efficient harvesters.

Furthermore, the FreeBie architecture is designed to support intermittently-powered end devices, but a notable research gap lies in the absence of support for intermittently-powered hosts on both sides of BLE communication. While the architecture excels in enabling communication between intermittently-powered device and continuously powered hosts, there's potential for innovation in extending its capabilities to encompass two intermittently-powered end nodes. This extension could unlock novel use cases in wireless sensor networks, broadening the applicability of the FreeBie architecture.

2.2 Body Sensor Network

The evolution of Body Sensor Networks (BSNs) has marked a significant milestone in healthcare and wellness monitoring. These networks, composed of wearable sensors, offer real-time data collection, analysis, and transmission, empowering individuals and healthcare professionals with valuable insights into physiological and medical conditions. BSNs demonstrated remarkable potential in applications ranging from remote patient monitoring to sports performance analysis and beyond.

Early research in BSNs primarily focused on sensor integration and data aggregation techniques [20]. These studies paved the way for more advanced BSNs capable of real-time health monitoring. The advent of wearable devices with integrated physiological sensors has led to the development of innovative solutions for continuous monitoring of vital signs such as heart rate, temperature, and electrocardiogram (ECG) signals [24], [22]. These advancements enable early detection of anomalies and timely intervention in critical situations.

The energy requirements of BSNs vary based on the complexity of sensors and the data transmission frequency. Wearable sensors that capture high-resolution data, such as ECG signals, demand a continuous power source for accurate monitoring. However, battery limitations hinder the potential for uninterrupted data collection. This challenge becomes even more pronounced when considering the size and weight restrictions of wearable devices [32], [15], [18].

The integration of battery-free IoT devices within BSNs offers a promising solution to the energy challenge [21]. This innovation not only extends the operational lifetime of BSNs but also opens the door to continuous and sustainable monitoring. Also it could be affordable (less than US$2 each when manufactured in volume), disposable, small, and easy to use [21]. Although there are already some Battery free Wireless BSN solutions but they do use near-field-enabled clothing capable of establishing wireless power and data connectivity between multiple distant points around the body to create a network of battery-free sensors interconnected by proximity to functional textile patterns.[19]

In summary, the evolution of BSNs has been characterised by breakthroughs in sensor integration and wireless communication. However, the challenge of battery dependence has persisted. The emergence of battery-free IoT devices powered by energy harvesting techniques presents a transformative solution. As ongoing research continues to address energy challenges and optimise device performance, the future of BSNs holds the promise of continuous and sustainable healthcare monitoring.

Chapter 3

Architecture and System Design

This chapter aims to explore the architecture and design of the CardioSync framework, building upon the problem statement from Chapter 1.

In response to those challenges, this chapter outline the architectural evolution from FreeBie to CardioSync, highlighting the transformative integration of the framework. Also this chapter, we provide a thorough analysis of the proposed architecture, focusing on key components, interactions, and overall design concepts.

Furthermore, to aid in understanding the conceptual landscape, Figure 3.1 illustrates the high-level architecture of the CardioSync framework within the existing FreeBie model.

3.1 CardioSync Framework

The architecture of FreeBie, as described in Section 2.1.1, employs periodic advertisement for BLE connection establishment in an end device. However, this approach can lead to asynchrony between the end device's periodic advertisement and the BLE host's scan, resulting in a hindrance to establishing a successful connection when both are intermittently-powered.

Addressing this limitation, the CardioSync framework augments over the FreeBie architecture to orchestrate BLE connection setup events in synchronisation with the host's events through the utilisation of a shared external signal. By doing so, CardioSync enables bidirectional intermittent operation for FreeBie. Moreover, the final CardioSync architecture collaborates seamlessly with FreeBie for scheduling a wake-up and sleep cycle that aligns with synchronised connection setups and adapts to the dynamic power availability of intermittently-powered devices.

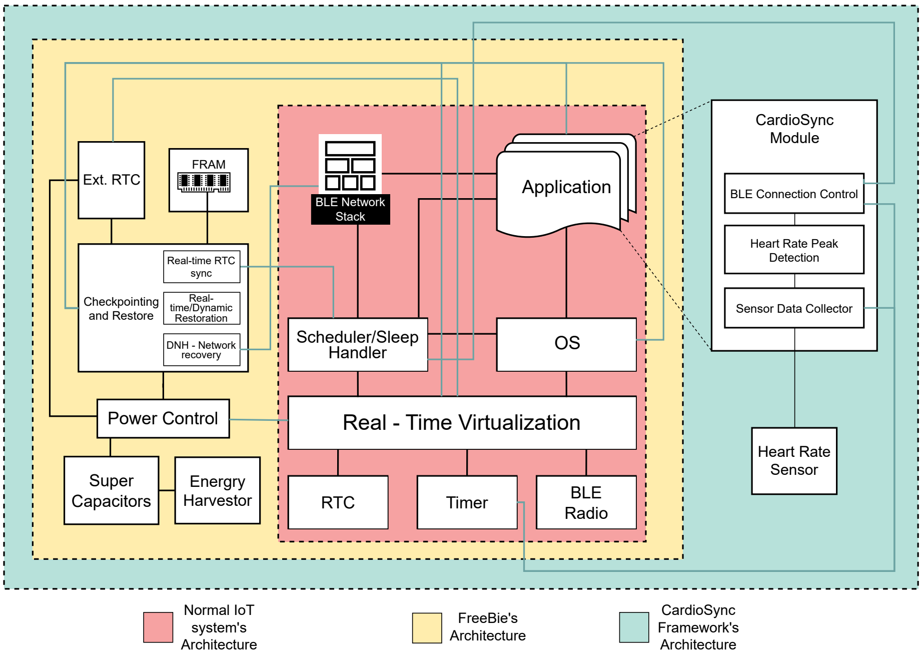
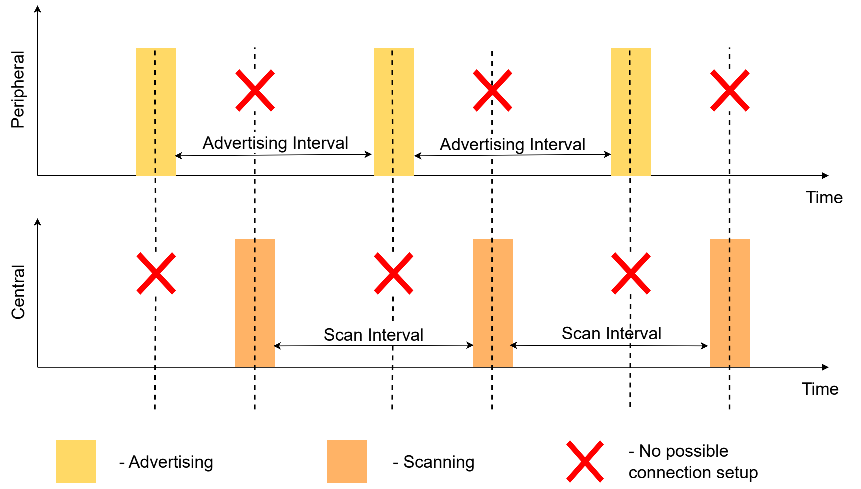
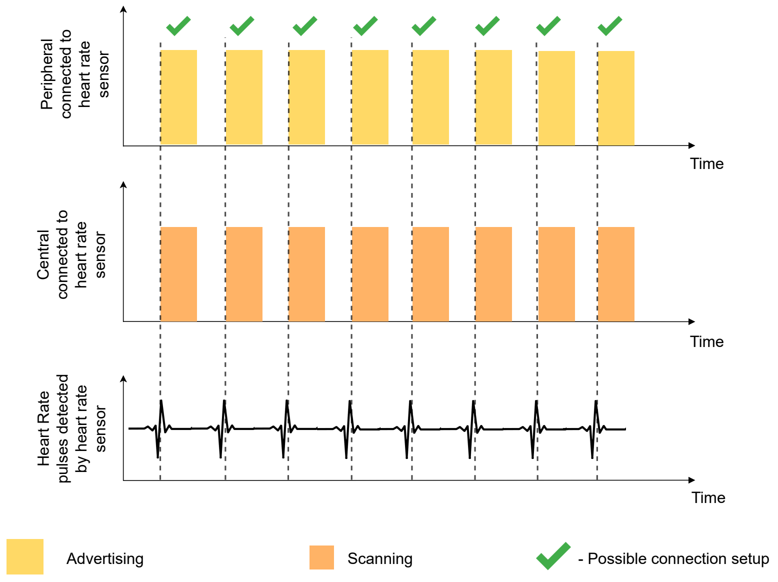


Figure 3.1: **High Level Architecture of the CardioSync framework.**



(a) **Failed asynchronous periodic connection setup attempts in a FreeBie system.**


Peripheral connected to heart rate sensor
Time
Central connected to heart rate sensor
Time
Heart Rate pulses detected by heart rate sensor
Time
Advertising
Scanning
- Possible connection setup

(b) **Successful synchronised connection setup attempts in a CardioSync system. Advertising of peripheral and scanning of central is synchronised by heart rate pulses detected by heart rate sensor connected to the devices.**

Figure 3.2: **Conceptual diagram illustrating the need and benefits of synchronisation by comparing the connection setup between two end devices in the asynchronous FreeBie system with the proposed CardioSync framework.**

Figure 3.2a highlights the difficulty encountered in establishing a connection between intermittent devices using the FreeBie architecture. Additionally Figure 3.2b showcases the planned approach used by CardioSync to address this obstacle via the utilisation of external clock as heart rate pulses.

The CardioSync framework is structured around the novel heart rate peak detection algorithm that detects peaks effectively based on raw data from a low power heart rate sensor. This algorithm serves as the basis for initiating synchronised connection events and also establishing synchronisation points which are strategically defined moments in time based on heart rate.

The CardioSync framework integrates into the FreeBie architecture, leveraging key components that contribute to the framework's enhanced functionality:

- **Low-Power Heart Rate Sensor:** The foundation of the CardioSync framework lies in a carefully selected low-power heart rate sensor. This sensor efficiently captures raw heart rate data, providing the essential input for synchronisation and connection setup activities.

- **CardioSync Real-Time Application:** This extension of the FreeBie architecture introduces the CardioSync module, encompassing several subcomponents responsible for different aspects of the synchronisation process.

  – **Sensor Data Collector:** This subcomponent interfaces with the low-power heart rate sensor, collecting accurate heart rate data. It ensures seamless data acquisition and serves as the initial step in the synchronisation process. It houses the sensor interfacing functionality along with data pre-processing as required for the algorithm.

  – **Heart Rate Peak Detection Algorithm:** At the heart of the CardioSync framework, this subcomponent processes the heart rate data collected by the sensor data collector. It intelligently identifies heart rate peaks, which serve as synchronisation triggers for connection setup events and establish synchronisation points. These synchronised trigger points are coupled with active sensor reads hence adhering to "Read and Synchronise" phase of CardioSync framework to establish a connection.

  – **Connection Control:** This subcomponent manages the setup of BLE connections via strategic scheduling. It leverages cues from the heart rate peak detection algorithm to time connection setup events, aligning them with heart rate interval. This integration utilises the Checkpointing and Restore component in the FreeBie architecture by effectively using Timers. Thereby ensuring the energy-efficient communication through "Sleep and Synchronise" phase of CardioSync framework.

These components collectively empower the CardioSync framework to synchronise communication activities with heart rate patterns, enabling bidirectional intermittent operation and efficient utilisation of intermittently-powered devices.

3.2 High-Level Operation Flow

The CardioSync framework operates through a systematic sequence of steps in integration with FreeBie architecture for establishing a connection

1. **Sensor Data Acquisition:** The process begins with the gathering of heart rate sensor data through dedicated interfaces.

2. **Heart Rate Peak Detection:** The acquired sensor data is processed by the heart rate peak detection algorithm. This algorithm identifies distinctive peaks that correspond to heart rate activity, simultaneously calculating the average time interval between these peaks. This interval defines the temporal spacing between synchronisation points.

3. **Connection Control:** Upon detecting a heart rate peak, the Connection Control component springs into action. It orchestrates the scheduling of BLE advertising or scanning events. Additionally, it optimally utilises the timers within the FreeBie architecture to initiate connection setup events that is precisely aligned with established synchronisation points.

4. **Time-Deterministic Checkpointing and Restore:** Operating within the FreeBie architecture, this module monitors periods of system inactivity, which in our case align with the intervals between synchronisation points set by the Connection Control component. It employs checkpointing to briefly pause the MCU's operations, reactivating only when there are pending tasks, such as the next connection setup event at next synchronisation point.

Chapter 4

Technical Implementation

This chapter delves into the practical realisation of the CardioSync framework, translating the architectural concepts and design discussed in the previous chapters into tangible code and functional components.

4.1 Hardware and Software Setup

This section outlines the necessary hardware setup and software environment required for the implementation of the framework.

4.1.1 Hardware Setup

Integral part of the devised framework is the heart rate sensor. After a comprehensive review of available low-power heart rate detection sensors in the market, the MAX30102 sensor from Maxim Integrated [17] was selected as the heart rate monitor for the CardioSync framework. The MAX30102 sensor, consuming as less as 1 mW and featuring low shutdown current around 0.7 μA, fits seamlessly with CardioSync's design goals [17].

The CardioSync framework is built upon the FreeBie model, integrating the MAX30102 sensor. The hardware setup comprises the following components:

- **FreeBie Mote:** This custom board in Figure 4.1 incorporates essential components, including the nRF52840 ARM-Based MCU module (*EY-SKBNZWB*) with BLE radio support, the External RTC and power management module (*AB1815*), and FRAM (*MB85RS4MT*) [12]. These components provide the foundation for the framework's intermittent operation.

- **MAX30102 Sensor Board:** The MAX30102 sensor board from Maxim Integrated, shown in Figure 4.2, serves as a critical element for heart rate monitoring. The sensor board features built-in pull-up resistors of 4.7kΩ in the I^2C pin outs [17]. However, we noticed even when MCU is in power save or sleep state, instead of near zero voltage and current, a considerable voltage difference of 1.6 Volts in $V_{\text{DD\_MCU}}$ and a current of 1.8mA is drawn by those external pull up resistors. For that reason, sensor's resistors were

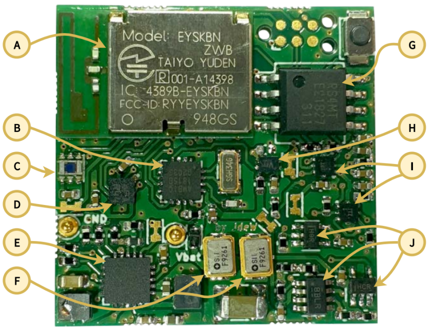


Figure 4.1: **FreeBie mote (front side). A total size is 1"×1". A: BLE ARM-based MCU - nRF52840, B: External RTC - AM1815, C: OPT3004 luminosity sensor, D: BMA400 accelerometer, E: BQ25570 energy harvester, F: Two parallel 7.5 mF capacitors, G: , MB85RS4MT fast non-volatile Ferroelectric Random Access Memory (FRAM), H: External power switch, I: SN74AUP2G79 flip-flop that allows the display to stay on when the MCU is off, J: Logic/switches prevent always-on signals from reaching the MCU when its OFF. Figure reproduced from [12].**

removed and instead the on-board pull up resistors of the nRF52840 were employed through Software configuration [26].

- **Connection Configuration:** Based on the pinout configuration of the FreeBie schematics [6], I^{2}C Clock Line (SCL) and I^{2}C Data Line (SDA) of the MAX30102 is connected to the respective GPIO pins. The MAX30102 is powered by the onboard super capacitors, connected to the V_{Batt} pinout of the FreeBie Mote.

4.1.2 Software Setup

The implementation of the CardioSync framework builds upon the open-source FreeBie source code, available from the TU Delft Sustainable Systems Lab repository [6]. The adaptation process involves customising the existing *"template"* application to accommodate the CardioSync system architecture - MAX30102 sensor and the Heart Rate peak detection algorithm.

Software Foundation and Customisation

The entire FreeBie source code relies on the PacketCraft BLE stack [1], which seamlessly integrates nRF5 SDK libraries for driver interfaces and peripheral interactions [27]. The code is specifically tailored for the nRF52840 MCU, utilising the GNU ARM Embedded toolChain, thus eliminating the need for additional build support or toolchain.

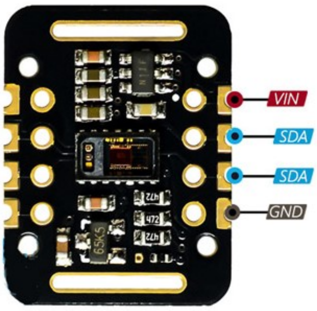


Figure 4.2: **MAX30102 sensor module (front side). The pin out marked are VIN: Module power supply 1.8V to 5V; GND: Ground; SCL: I2C clock bus; SDA: I2C data bus. Figure reproduced from [29].**

Sensor Interfacing

For interfacing with the MAX30102 sensor, nRFx driver for the Two Wire Interface - Master (TWIM) is employed. This involves enabling the necessary configurations in the *sdk_config.h* file of nRF5 SDK to activate the TWIM in the system. Subsequently, relevant APIs provided by the nRFx library are utilised for reading from and writing to the I^{2}C interface.

BLE Stack Configuration

In the context of the BLE stack, PacketCraft's CMake configurations take charge of initialising the MCU as either a Peripheral or Central device. Two specific CMake definitions, namely `INIT_PERIPHERAL` and `INIT_CENTRAL`, determine the operational mode. PacketCraft then leverages essential components like the *Device Manager* and the *Application framework main module* to initialise the BLE stack. This equips the framework with APIs that helps with the initiation and termination of advertising or scanning activities depending on whether the device is operating as a Peripheral or Central.

Timing and Scheduling

Both the FreeBie and CardioSync frameworks require precise timing and scheduling mechanisms. PacketCraft's Wireless Software Foundation (WSF) OS layer offers essential APIs for managing the AppTimer functionality. AppTimer is essentially a timer management module that allows to schedule and manage various timing events and callbacks within the application. It provides a way to create, start, stop, and manage timers, which can be crucial to perform real-time tasks at specific intervals.

In the context of your CardioSync framework, WSF AppTimer was utilised to handle timing-related operations such as to schedule the sampling of the heart rate sensor at the desired frequency. It also ensures that BLE connection setup occur at the right times and in synchronisation with each other.

Algorithm 1 MAX30102 Sensor Data Acquisition

```
Initialize 2D Buffer with maximum capacity of 2 x 500
Initialize Sliding Window size as 10

Start "MAX30102Timer" with 10 milliseconds period    ▷ After converting sampling rate: 100 Hz into milliseconds (1000/SamplingRate)

if MAX30102Timer expired then
    Read MAX30102 sensor's FIFO using "MAX30102_read_fifo()"  ▷ Called from Timer Expiry event handling
    Append IR value and corresponding RTC ticks to 2D Buffer and Sliding Window

    if Sliding Window is full and Average of sample data > 8000 then
        Call Heart Rate Peak Detection API on Sliding Window data ▷ calculate_heart_rate()
    end if

    Restart "MAX30102Timer"
end if
```

Additionally, these WSF AppTimer aid in the FreeBie architecture's ability to accurately track the timing of active scheduled tasks inside the queue. Consequently, the system is able to power down and enter a sleep state when there are no planned activities for the next period.

4.2 Sensor Interface and Peak Detection Algorithm

This section delves into the practical implementation and methodologies used for interfacing the MAX30102 sensor with the FreeBie architecture and the novel heart rate peak detection algorithm.

As outlined in the section 4.1.2, the existing "template" application within the FreeBie source code is extended to incorporate the MAX30102 sensor functionality into a new source file which houses all the necessary functions, declarations, and configurations.

4.2.1 MAX30102 Sensor Data Acquisition

Before starting the sensor sampling, sensor should be initialised and configured with settings that ensures optimal performance while minimising power consumption. Different configuration settings done for MAX30102 is described in detail in Section A.1 of Appendix A. With these settings, MAX30102 operates with less power consumption, yet capturing accurate blood flow for heart rate analysis.

This section outlines the methodology employed for data collection, using buffers and sliding window within the CardioSync framework. The process of the data acquisition is given in Algorithm 1.

The raw data is continuously read from the MAX30102 sensor's FIFO at a predetermined sampling rate of 100 Hz. This sampling rate of 100 Hz is chosen after thorough evaluation of sensor which will be covered in Section 5.1.

A two dimensional buffer named "ir_values" with a maximum capacity of 2 x 500 entries is employed to store the sensor data and their respective RTC ticks at which they are read. Also a sliding window named "window" of size 10 is chosen, to store the latest 10 sensor values at any instance.

Now that the initial setup and configurations are laid out, the data acquisition process is orchestrated by the following key steps:

1. **Start of the Sampling:** A WSF Apptimer named MAX30102Timer is started to start the sampling of MAX30102 sensor. The timer's period is determined to be 10 milliseconds i.e., 1000 divided by the chosen sampling rate of 100Hz.

2. **FIFO Read and Buffering:** When the timer expires, the sensor's FIFO is read using the "MAX30102_read_fifo()" function. This function relies on the nrfx_twim driver APIs to extract the real-time IR value that holds blood flow data for heart rate peak detection. Consequently, it is paired with the corresponding RTC ticks at the time which it was recorded and appended to the end of the 2-D buffer, i.e., ir_values buffer and also to the window buffer.

3. **Continuous Data Acquisition:** The "MAX30102Timer" is restarted and the sequence repeats till the timer is stopped or not started again. This ensures the raw data from sensor is continuously read and latest 10 values always stay in the window buffer. When the ir_values buffer becomes full, it undergoes a rotation where the samples are shifted to the left, causing the oldest value to be discarded. Subsequently, the newest value is appended to the right end of the buffer.

4. **Ready for peak detection:** This happens when the sliding window buffer - window is full. The data is subjected to heart rate peak detection algorithm only if the average of the data within the sliding window surpasses the threshold of 8000. This threshold was determined after analysing the sensor's raw data, which showed consistent changes when the sensor was in proximity to human skin compared to when it was not. After configuring the sensor, it recorded values above 8000 when in contact with human skin. Thus, this threshold ensures peak detection happens when the sensor reliably maintains contact with the skin.

The iterative nature of this process guarantees the availability of up-to-date sensor data for real-time heart rate analysis, ensuring the responsiveness and accuracy of the CardioSync system.

4.2.2 Heart Rate Peak Detection Algorithm

Initially, an attempt was made to utilise Maxim Integrated's reference heart rate peak detection algorithm [3] developed for their low power, optical heart-rate module - MAXREFDES117 [16]. This algorithm employed a method of peak identification by calculating a fixed threshold and detecting peaks above that threshold. However, this approach proved ineffective for the CardioSync

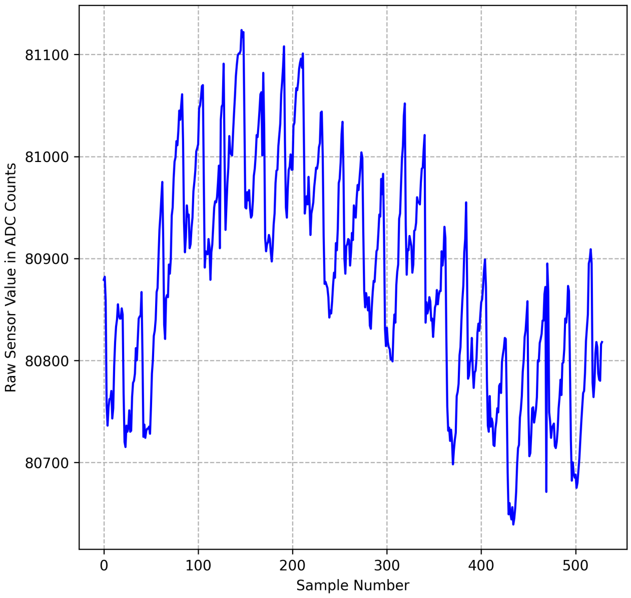


Figure 4.3: **Example measurement showing the dynamic and fluctuating nature of continuous sensor data readings from MAX30102. The Y-axis represents the sensor's raw data in ADC count units. These units indicate the magnitude of the analog signal measured by the sensor and converted into digital form by the analog-to-digital converter (ADC).**

framework due to the dynamic and varying nature of the continuous sensor data readings obtained from the MAX30102. As shown in Figure 4.3, the sensor data exhibited fluctuations that necessitated a more adaptive and dynamic threshold approach.

In light of these challenges, the decision was made to adopt an alternative approach that leverages the derivatives of the sensor data for peak detection. This approach was more suitable for handling the varying nature of the sensor readings and provided a robust means of detecting heart rate peaks.

The heart rate peak detection algorithm using derivatives operates as follows:

1. **Data Preprocessing:** The raw sensor data collected from Sliding window buffer, denoted as $\mathtt{a_{ir}}$, undergoes preprocessing steps. The algorithm removes the DC component by calculating the mean and subtracting it from the data. This step is essential to eliminate any static offset and focus on the variations caused by heartbeats.

2. **Moving Average:** The $\mathtt{a_{ir}}$ data is then updated using a moving average of two neighbouring samples on either side of each sample. This smoothing technique helps reduce noise and sharp variations in the data, facilitating more accurate peak detection.

3. **Derivative Calculation:** The first derivative of the $\texttt{a}_{\texttt{ir}}$ data, referred to as $\texttt{a}_{\texttt{ir\_diff}}$, is calculated. This derivative provides insights into the rate of change of the signal, which is particularly useful for identifying rapid transitions characteristic of heart rate peaks.

4. **Peak and Valley Detection:** The algorithm identifies peaks and valleys within the $\texttt{a}_{\texttt{ir\_diff}}$ data. Peaks correspond to points where the derivative changes from positive to negative, indicating a downward slope. Valleys, conversely, represent points where the derivative changes from negative to positive, indicating an upward slope. These points mark significant variations in the signal, which are indicative of heart rate peaks.

5. **Peak Validation and Removal:** Detected peaks that are in close proximity to valley indices are discarded. A maximum distance threshold `n_valley_distance` of value **4** is used to determine this proximity. Similarly, peaks that are too close to each other, within a distance threshold `n_peak_distance` of value **15**, are also removed. The selection of these thresholds is based on the retrospective analysis of visual patterns on the continuous sensor read. The visual analysis helped to determine the typical distances between peaks and the lengths between valleys and subsequent peaks. These additional validation steps help ensure that the detected peaks correspond to distinct heart rate events.

6. **Result Generation:** The algorithm generates a list of validated peak indices that represent heart rate peaks which likely to have occurred.

7. **Callback and Post Processing:** For each detected and validated peak index, the algorithm invokes the `sensor_callback` function. This function handles further processing, synchronisation, and scheduling tasks related to the CardioSync framework.

The constants `n_valley_distance`, and `n_peak_distance` are empirically determined based on the characteristics of the sensor data and the expected heart rate patterns. These values ensure the reliability and accuracy of the peak detection process. The algorithm is outlined in Algorithm 2

4.3 CardioSync Integration with FreeBie BLE

With the fundamental components of the CardioSync framework described, the next step involves integrating these elements into the FreeBie architecture. This section provides insight into how these various parts work together to create a functional system, ensuring accurate heart rate peak detection and synchronised BLE connection setup within a battery less system architecture.

An effective approach to elucidate the integration within FreeBie is to follow the chronological flow of steps required for establishing a synchronised BLE connection from a coding perspective. This approach is represented as a flowchart in Figure 4.4.

System Initialisation

Starting from the beginning of the execution process, depending on whether the device operates in a peripheral or central mode, the necessary BLE modules of the FreeBie architecture are initialised within the "template" application's stack initialisation. The initialisation of the CardioSync extension framework, which includes the MAX30102 sensor, is also integrated into the stack initialisation process.

The MAX30102 initialisation manages the setup and configuration of the sensor, as described in Section A.1 of Appendix A. Additionally, the essential WSF AppTimers as explained in Section 4.1.2, such as `MAX30102Timer` and `MAX30102RestartTimer`, are declared and initialised in this process. The specific roles and functions of these timers will be further explained.

Sensor Data Acquisition and Peak Detection

With all essential initialisation have been accomplished, the sampling of sensor is started through `MAX30102Start()` function, which kicks off the `MAX30102Timer` timer and starts the process of data acquisition and heart rate peak detection which was 4.2 and 4.2.2.

Read and Synchronise

With the successful identification of heart rate peaks, the post processing is rudimentary to distinguish between new peaks and peaks that linger within the rotating 2-D buffer. This differentiation is achieved by utilising the RTC ticks associated with the detected peak indices.

Additionally, as new peaks are identified, the post-processing routine updates the average time difference between peaks in a global variable - the *"Heart Rate Interval"*. This value later holds significant importance in establishing synchronisation points based on heart rate.

Depending on the operational mode of the system, either advertisement or scan initiation is executed as soon as a new peak is detected using PacketCraft's APIs, such as `AppAdvStart()` or `AppScanStart()`. These APIs use the BLE parameters that are strategically chosen based on the comprehensive evaluation of algorithm performance and system behaviour outlined in Table 5.3. It is noteworthy that this synchronised BLE connection setup operates simultaneously with continuous sensor read at a 100Hz sampling rate and heart rate peak detection.

Sleep and Synchronise

After the detection of three peaks (`MAX_PEAKS`), the `MAX30102Stop()` routine is invoked. It shuts down the sensor by configuring the power-saving mode and terminates `MAX30102Timer`, thereby suspending active sensor read and heart rate peak detection. BLE parameters are also then dynamically adjusted during runtime to ensure the scheduling of advertisement or scanning events aligned with the global variable *Heart Rate Interval*. The modified BLE parameters are:

- **BLE Advertisement Parameters (Peripheral):**
    - Advertising Interval is adjusted runtime with global variable *Heart Rate Interval.*
    - Advertising Duration is 10 seconds.
- **BLE Scan Parameters (Central):**
    - Scan Interval is modified in runtime with global variable *Heart Rate Interval.*
    - Scan Window is fixed as 100 milliseconds.
    - Scan duration is set as 10 seconds.

The incorporation of *Heart Rate Interval* in BLE parameters establishes the fixed synchronisation points for the system without any active sensor read. The scheduling of BLE connection events only during the established synchronisation points, results in the system being idle without any active tasks during the *Heart Rate Interval* period. The Checkpointing and Restore module, which is integrated into the FreeBie architecture, identifies system inactivity and utilises checkpointing to preserve the current state of the system while temporarily halting the operations of the MCU. The system resumes its operation at predetermined BLE synchronisation points and recovers its context, after which it returns to a sleep state for the upcoming heart rate period.

Adaptive Heart Rate Update

In cases where the BLE connection setup experiences prolonged delays, the continuity of the *Heart Rate Interval* may become compromised. In response, an adaptive iterative strategy is employed. A timer, `MAX30102RestartTimer`, restricts the "Sleep and Synchronise" phase to a 10-second duration. Upon the timer's expiration, `MAX30102Start()` is called again, triggering the "Read and Synchronise phase" which updates *Heart Rate Interval.* The iterative process between phases is stopped once a successful BLE connection is established. Additionally, when the connection is closed, `MAX30102Start()` is reinvoked to start "Read and Synchronise" phase.

Algorithm 2 Heart Rate Peak Detection Algorithm

```
function CALCULATE_HEART_RATE($a\_ir, callback$)
        ▷ $a\_ir$: array of sensor data from Sliding window buffer.
        ▷ $callback$: callback function to process the peaks detected by algorithm.

    Initialise $a\_ir\_clean[]$, $a\_ir\_diff[]$, $peak\_indices[]$, and $valley\_indices[]$
        ▷ $a\_ir\_clean[]$: array of sensor data after DC component is removed
        ▷ $a\_ir\_diff[]$: array of first derivative of sensor data
        ▷ $peak\_indices[]$: array of detected peak indices in the sensor data
        ▷ $valley\_indices[]$: array of detected valley indices in the sensor data

    $\mu = \frac{1}{n} \sum_{i=1}^{n} a\_ir[i]$        ▷ Calculate mean of raw IR sensor data
        ▷ n: size of $a\_ir$ array, which is 10 (size of sliding window buffer)

    for each data point $x$ in $a\_ir$ do        ▷ Removing DC component in sensor data
        $a\_ir\_clean[x] = a\_ir[x] - \mu$
    end for

    for each data point $x$ in $a\_ir\_clean$ do        ▷ Filtering out sensor data using moving average of two neighbouring data on both side of each data point
        $a\_ir\_clean[x] = \frac{1}{5} \sum_{i=x-2}^{x+2} a\_ir\_clean[i]$
    end for

    for each data point $x$ in $a\_ir\_clean$ do        ▷ First derivative of sensor data
        $a\_ir\_diff[x] = a\_ir\_clean[x] - a\_ir\_clean[x-1]$
    end for

    for each index $x$ in range 1 to $n-2$ do        ▷ Peak detection
        if $a\_ir\_diff[x-1] > 0$ and $a\_ir\_diff[x] < 0$ then
            $peak\_indices[] \leftarrow x$
        end if
    end for

    for each index $x$ in range 1 to $n-2$ do        ▷ Valley detection
        if $a\_ir\_diff[x-1] < 0$ and $a\_ir\_diff[x] > 0$ then
            $valley\_indices[] \geq x$
        end if
    end for

    Remove peaks close to each other using $n\_peak\_distance$ threshold
    Remove peaks close to valleys using $n\_valley\_distance$ threshold

    Call $callback$ function with detected peak indices to process them
end function
```

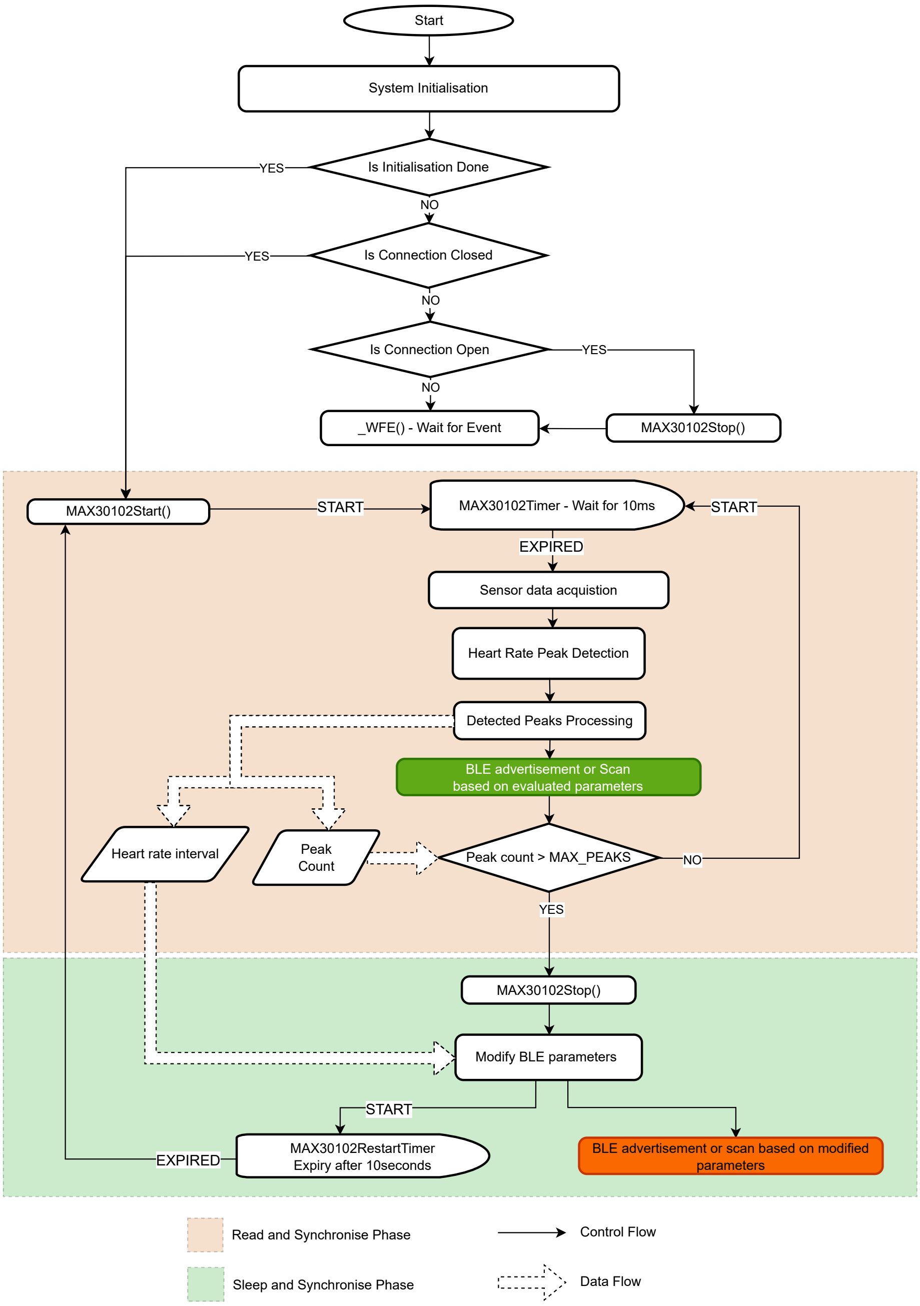


Figure 4.4: **Operation flowchart of the CardioSync framework implementation.**

Chapter 5

Results and Evaluation

This chapter presents the results of experimentation, and analysis of data obtained from the CardioSync Framework. Also it compares the key result outcomes of CardioSync framework, such as *BLE connection setup time, power and energy consumption* in comparison to a naive and reference FreeBie system that does not employ any synchronisation mechanism.

The evaluation of the CardioSync framework is structured into three distinct contexts, inline with the research objectives

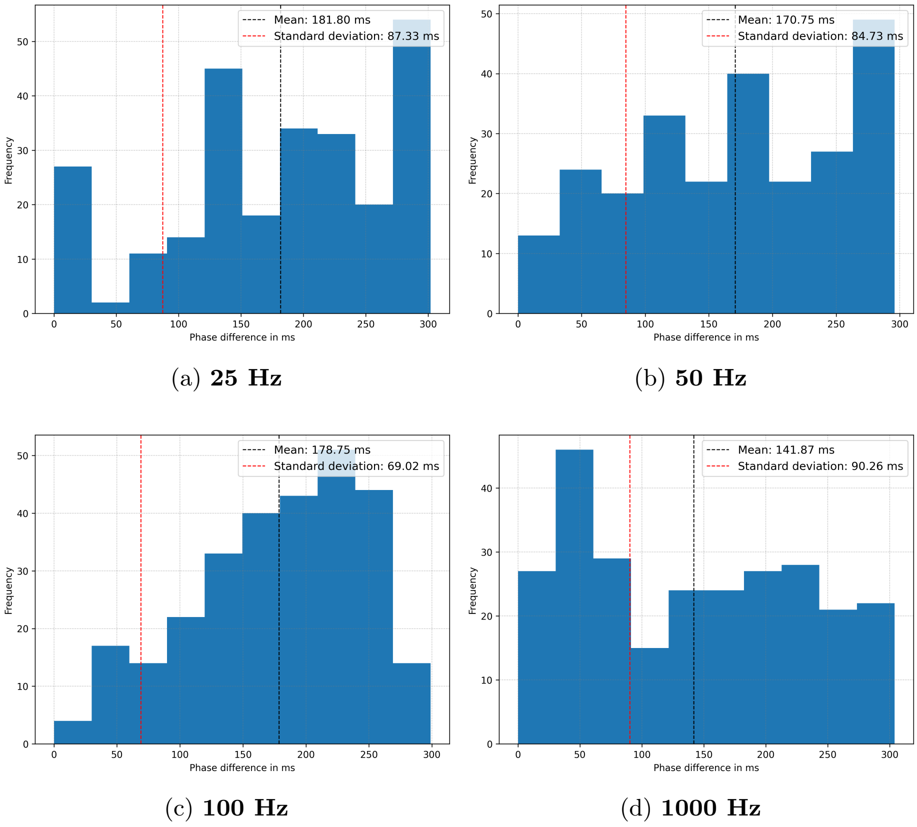
1. Heart Rate Sensor Accuracy and Peak detection Algorithm performance.
2. Verification of Integrated CardioSync framework in Continuous and Intermittent power.
3. Comparative analysis of the CardioSync framework versus Naive reference freebie model.

5.1 Evaluation of Heart Rate Sensor and Peak detection Algorithm

The evaluation in this section is split into two parts - *Sensor accuracy profiling* based on phase difference for a particular peak detected between two sensors and *BLE connection performance* at different sensor sampling rate.

5.1.1 Sensor Accuracy Profiling

Sensor accuracy for detecting heart beat is mainly based on the peak detection algorithm, and also on the sensor configurations. However, the sensor was intentionally designed to function within a fixed configuration, optimising power consumption (Refer to Section:A.1 in Appendix A). So to quantitatively assess accuracy of the algorithm, the phase difference for heart pulse detection between two sensors was selected as the focus. This parameter was examined at various sampling rates of the peak detection algorithm.



(a) **25 Hz**

(b) **50 Hz**

(c) **100 Hz**

(d) **1000 Hz**

Figure 5.1: **Histograms of phase differences between heart pulses detected by two sensors at different sensor read frequencies. At 25 Hz and 50 Hz, larger phase differences occur at more frequency than 100 Hz. At 100 Hz, phase differences are evenly distributed around mean. At 100 Hz, phases differences are more irregular.**

Experimental Setup

To measure the phase difference between the heart pulse detected by two different MAX30102 sensors and a nRF52840DK development board [25] was used. The measurement software was developed based on nRF5 SDK [27] to interface with two sensor boards using two-wire interface (TWI). The software would use the heart beat detection algorithm of the CardioSync system to independently identify heart pulse peaks for each sensor and log the corresponding detection time. Subsequently, the phase difference should be computed between the aforementioned peaks. After two sensors have recorded around 250 peak detection, an experiment is finished, and it is repeated four times, each time with different sensor sampling rates: *25 Hz, 50 Hz, 100 Hz, and 1000 Hz.*

Results Discussion

The histograms shown in Figure 5.1 illustrate the distribution of phase differences recorded for various sensor sampling rates. The mean, minimum, and

Sensor Sampling rate	Phase difference in milliseconds		
	Mean	**Min**	**Max**
25 Hz	181.80	0	301
50 Hz	170.75	0	296
100 Hz	178.75	0	299
1000 Hz	141.87	0	304

Table 5.1: **Phase difference statistics at different sensor sampling rates.**

maximum phase differences are summarised in Table 5.1. Analysing these distributions and their statistical characteristics reveals that phase difference in detection time between two sensors is inevitable. However, this data holds valuable insights for determining the optimal sensor read frequency for the CardioSync system.

At different sampling rates (25 Hz, 50 Hz, 100 Hz, and 1000 Hz), the distribution of phase differences between heart rate peaks reveals distinct characteristics. At 25 Hz and 50 Hz, a right-skewed distribution indicates a higher likelihood of larger phase differences, with about 33% and 35% of detected peaks respectively being more than 305ms out of phase. In contrast, a slightly right-skewed normal distribution at 100 Hz suggests that phase differences mostly occur around the mean value, with around 24% exceeding the 305ms threshold. A 1000 Hz sampling rate presents an irregular phase difference distribution, highlighting unpredictable occurrences.

From all these observations, it can be concluded that **Sampling rate 100 Hz** is more predictable in terms of phase difference and is more accurate in heart pulse peak detection between two sensors. Also, even the sensor settings are at 50 samples per second, thereby not losing any samples from the sensor.

5.1.2 BLE Connection Performance

The aim of this assessment is to analyse the performance of the novel CardioSync peak detection algorithm with synchronised Bluetooth Low Energy (BLE) connection at different sensor sampling rates. This evaluation is conducted in preparation for the integration of the algorithm into the FreeBie architecture. The analysis was based on the number of peaks needed for successful connection establishment.

Experimental Setup

To record the number of heart pulse peaks needed for successful BLE connection, two nRF52840DK development board [25] were used and each board was interfaced with one MAX30102 sensor [17] to its I2C GPIO pins. Measurement software was developed based on the PacketCraft BLE stack [1]. The devised measurement code calculates the average time interval between peaks detected by the algorithm in run time. Also once the connection is established, it records the number of peaks detected before connection.

Sensor Sampling rate	**Average number of peaks detected to connect**		**Average Connection setup time in seconds**	
	Peripheral	**Central**	**Peripheral**	**Central**
50 Hz	1.6	1.6	2.789	2.847
100 Hz	1.667	1.733	2.074	2.018
1000 Hz	1.8667	1.733	1.407	1.129

Table 5.2: **Comparison of BLE connection at different sensor sampling rates. Number of peaks detected before connection setup occurs is lesser at 100 Hz than 1000 Hz, even though the average connection setup time is lower at 1000 Hz.**

BLE Advertisement Parameters (Peripheral)	
Advertising Interval	50 milliseconds
Advertising Duration	200 milliseconds
BLE Scan Parameters (Central)	
Scan Interval	15 milliseconds
Scan Window	20 milliseconds
Scan Duration	200 milliseconds

Table 5.3: **Chosen BLE parameters for the CardioSync system based on chosen sampling rate 100 Hz.**

With these setup, 15 experiments were performed at different sensor sampling frequencies. Each experiment is considered complete once the connection is established between two nRF52840DK boards using the MAX30102 detected heart pulses.

Results Discussion

The results in Table 5.2 demonstrate that a sampling frequency of 1000 Hz results in a shorter average time interval between peaks, enhancing connection establishment time due to quicker peak detection. However, this frequency exhibited an inconsistent number of peaks between experiments for connection establishment, leading to a higher average number of required peaks. Consequently, balancing these findings with the earlier assessment of sensor accuracy, **the 100 Hz sampling rate was selected for integrating the CardioSync system**.

Furthermore, the selection of a sample frequency of 100 Hz has led to the determination of BLE parameters for the CardioSync system, backed up by the outcomes of this evaluation. In order to address the **average phase difference of 178.75 milliseconds** observed between two sensors and reducing the number of peaks necessary to establish a connection, the BLE advertising and scanning configurations have been chosen as outlined in Table 5.3.

Both the scan and advertising duration have been set at 200 milliseconds for each detected heart pulse. This allows for a sufficient window of time to synchronise and establish a connection, even in cases where there may be a significant disparity between the readings from two sensors.

5.2 Validation of Integrated CardioSync System

The purpose of the following section is to provide an illustration of the operational functionality of the integrated CardioSync framework to synchronise a BLE connection setup within sleep-wakeup principles of the FreeBie system architecture. The findings are presented for two types of input power: Continuous power and Intermittent power (which were emulated using a square wave and the parameters of the square wave are 75% duty cycle and total period of 8 seconds).

5.2.1 Experimental Setup

Hardware Setup

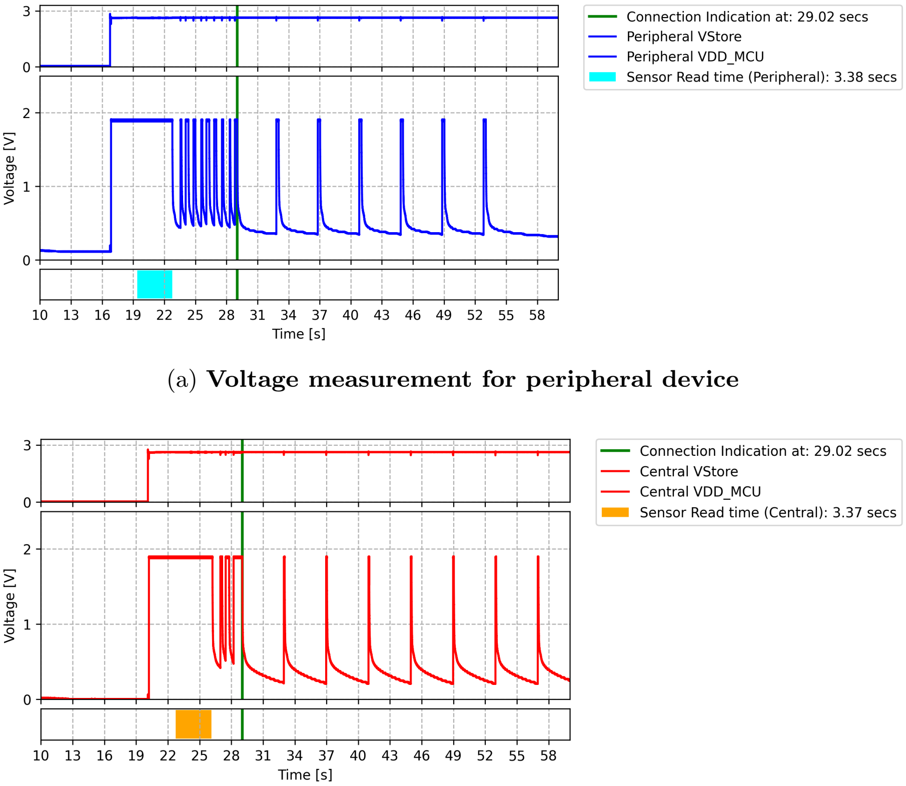
The experimental configuration comprises of two FreeBie boards [4], each equipped with a MAX30102 sensor. The voltage supplied to the sensor is derived from the $\mathrm{V_{Batt}}$ source on the FreeBie board. The FreeBie board receives its voltage supply from a voltage emulator kit called *DIPS* [5] connected to the $\mathrm{V_{Store}}$ pin. The DIPS system comes with Emulator Host software, which facilitates the configuration and simulation of the desired voltage supply for the FreeBie system [5].

The *Saleae logic analyser* [23] is employed to gather results that demonstrate the functionality of the system. In our experimental setup, we employed four channels from logic analyser for each of the FreeBie boards to measure the voltage values of V_{DD_MCU}*,* V_{Store}*, Sensor Read state(GPIO pin), and Connection Open indication (GPIO pin).*

The *nRF Power Profiler Kit II (PPK)* [28] is employed for the purpose of measuring energy consumption throughout the runtime by hooking it to the $\mathrm{V_{Store}}$ pin of the board. Additionally, it serves as a continuous source of power for the FreeBie board. Similar to the Saleae Logic analyser, the PPK is capable of capturing and recording current measurements of a device in real-time.

Software Setup

Regarding the experimental software setup, modifications have been made to the system described in Chapter 4. These modifications involve the activation of two GPIO pins to indicate the start and stop of the sensor read phase and the successful establishment of a BLE connection respectively. One of the FreeBie boards is programmed with the CardioSync Peripheral code, while the other is flashed with the CardioSync Central code.



(a) **Voltage measurement for peripheral device**

(b) **Voltage measurement for central device.**

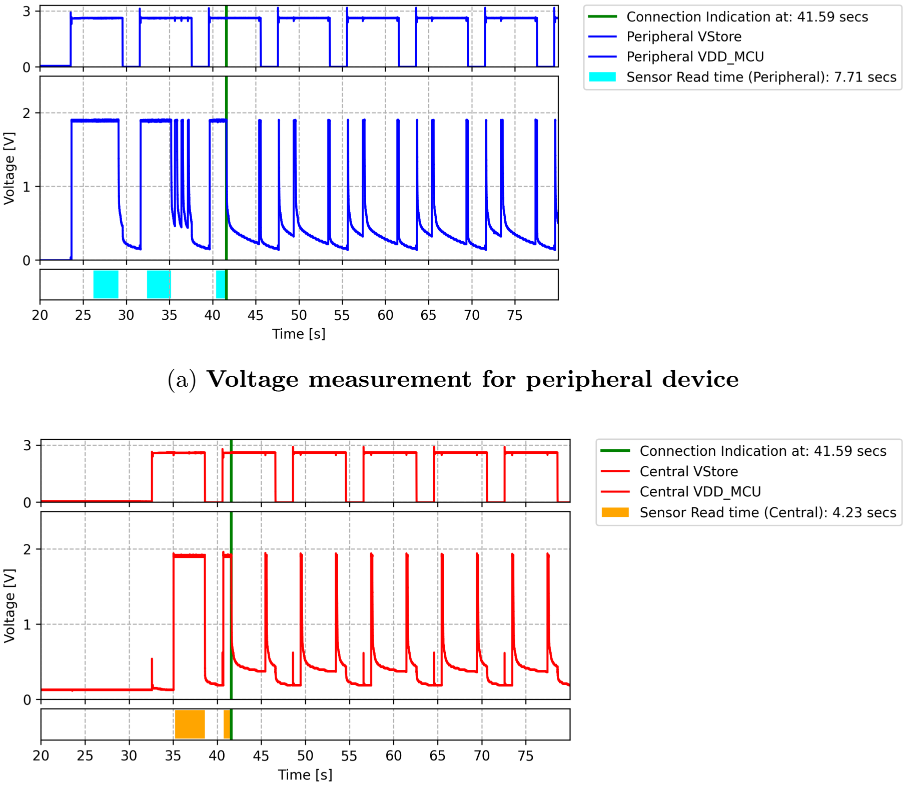
Figure 5.2: **Real-time voltage measurement for the CardioSync system *with continuous power supply.* Each of the three plot in two devices shows different measurement. From the top to bottom in each device, 1) Plot of V_{Store}: Supply Voltage, 2) Plot of V_{DD_MCU}: MCU voltage and 3) Plot of sensor read period indication. The green vertical line through all the plots indicates the successful BLE connection setup event**

5.2.2 Results Discussion

Validation of Working CardioSync

The time series plot depicted in Figure 5.2 illustrates the operation of CardioSync framework with continuous powered input. It can be asserted from the plots that there is the periodic shutdown of V_{DD_MCU} during times of inactivity, therefore corroborating the anticipated power-saving benefits associated with the integration of FreeBie.

As elucidated in Chapter 4, the system from Figure 5.2 demonstrates the anticipated functionality of "Read and Synchronise" for a period of *3.40 seconds* where the sensor is being actively read. Followed by that, we also see the "Sleep and Synchronise" functionality, where the system exclusively reactivates at the interval of each heartbeat pulse in order to initiate a connection. The connec-



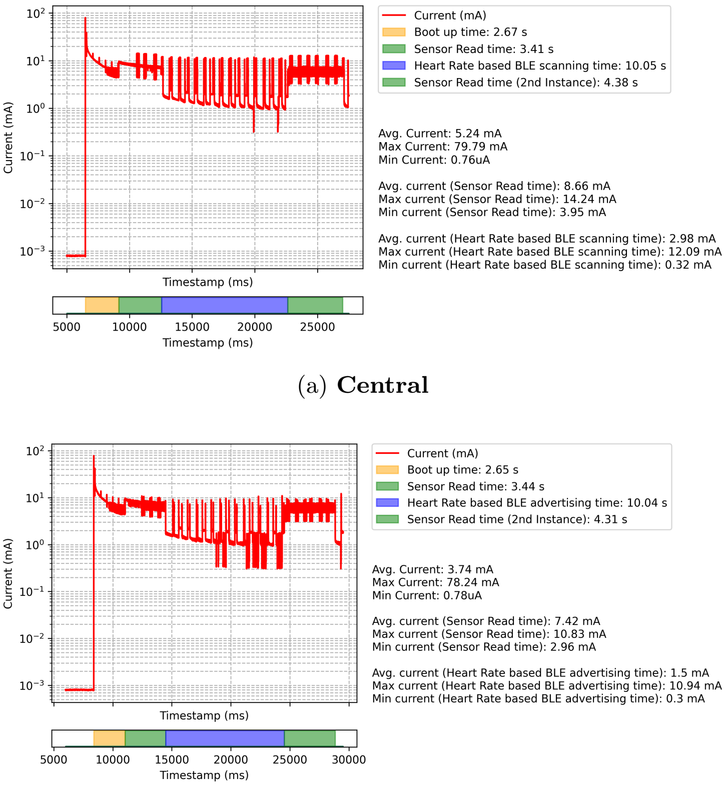
(a) **Voltage measurement for peripheral device**

(b) **Voltage measurement for central device**

Figure 5.3: **Real-time voltage measurement for the CardioSync system *with intermittent power supply*. Each of the three plot in two devices shows different measurement. From the top to bottom in each device, 1) Plot of V_{Store}: Supply Voltage, 2) Plot of V_{DD_MCU}: MCU voltage and 3) Plot of sensor read period indication. The green vertical line through all the plots indicates the successful BLE connection setup event.**

tion establishment event, which is closely synced with the heart rate pulses, is shown by the green vertical line in Figure 5.2.

A similar verification was carried out under intermittent power supply using a square wave signal with a duty cycle of 75% for a duration of 8 seconds. The time series plot (Figure 5.3) reveals the adaptability of the system to changing power availability, as connections continue to be established in coordination with heart beat, despite power fluctuations. Additionally, the CardioSync system was tested in various square wave periods with different duty cycles as shown in Table A.1 in Appendix A, demonstrating its adaptability to different supply configurations.



(a) **Central**

(b) **Peripheral**

Figure 5.4: **Current measurement in idle state without connection establishment for CardioSync systems. Average current consumed in real-time during different phases of the system are differentiated with different colours.**

Current Measurement of CardioSync system

Figure 5.4, gives us the current measurement for different phases of the CardioSync system operation to synchronise the BLE connection. The measured current values are afterwards utilised for comparative analysis against naive reference system.

It is important to note that the "Read and Synchronise" phase requires a higher current compared to the "Sleep and Synchronise" phase. This behaviour is supported by central and peripheral devices in the current measurement-driven evaluation. In the Central device, "Read and Synchronise" phase happens for around *3.41 seconds* and consumes an average current of approximately *8.66mA*. Similarly, the peripheral device registers *7.42mA* over a *3.44-second* interval of sensor read phase. Notably, during the "Sleep and Synchronise" phase of 10 seconds, both central and peripheral devices exhibit a substantially lower average current consumption—*2.98mA and 1.5mA*, respectively.

Configuration Name	Configured BLE parameters	
Low	*Advertising Interval*	6 seconds
	Scanning Interval	6 seconds
	Scanning Window	1 second
Medium	*Advertising Interval*	2 seconds
	Scanning Interval	2 seconds
	Scanning Window	500 milliseconds
High	*Advertising Interval*	500 milliseconds
	Scanning Interval	500 milliseconds
	Scanning Window	125 milliseconds

Table 5.4: **Chosen BLE parameters for Reference FreeBie system**

The values of these evaluation are afterwards utilised in the computation of the energy expended by the device for the purpose of synchronising a BLE connection.

5.3 Comparative Analysis against Naive Reference Design

In this section, we start by introducing a naive reference model of Battery-Free BLE. This model will be used as a yardstick to gauge the effectiveness of our synchronised CardioSync system.

5.3.1 Naive Reference FreeBie Model

To ensure a comprehensive comparison, the reference system used is the FreeBie system, which lacks any synchronisation method. For experimental purposes, both the Central and Peripheral components of the FreeBie reference model were intentionally set to have matching Advertising and Scanning intervals, preventing any permanent link failure.

To enrich our data pool, we selected three distinct BLE configurations for the FreeBie system, each characterised by varying energy consumption, as detailed in Table 5.4. To provide practical real-world context, each configuration represents a scenario within the realm of body sensor networks. These diverse scenarios provide a practical basis for comparing the performance of the CardioSync framework with the reference FreeBie system.

Configuration Name	*Average Current measured in mA*	
	Central	*Peripheral*
Low	*1.77*	0.07
Medium	*2.78*	0.16
High	*2.85*	0.6

Table 5.5: **Average current measurement using the experimental setup in Section 5.2.1 for each of the reference system configuration.**

- **FreeBie Low (Limited-Energy Scenario):** This scenario envisions a wearable sensor attached to a person's shoe sole, relying on minimal energy generated through piezoelectric energy harvesting from each step. In this challenging environment, efficient energy management during connection attempts is crucial.
- **FreeBie Medium (Balanced-Energy Scenario):** In this scenario, the device is part of a fitness tracker, like a wristband, accessing moderate energy from a solar cells. While energy constraints are less severe than in the Low scenario, efficient energy use remains vital.
- **FreeBie High (Abundant-Energy Scenario):** Two energy sources are considered, one on the wrist and one on the ankle, harnessing abundant energy from solar cells and motion harvesting, respectively. These devices can expend energy liberally for frequent connection attempts, emphasising connection efficiency.

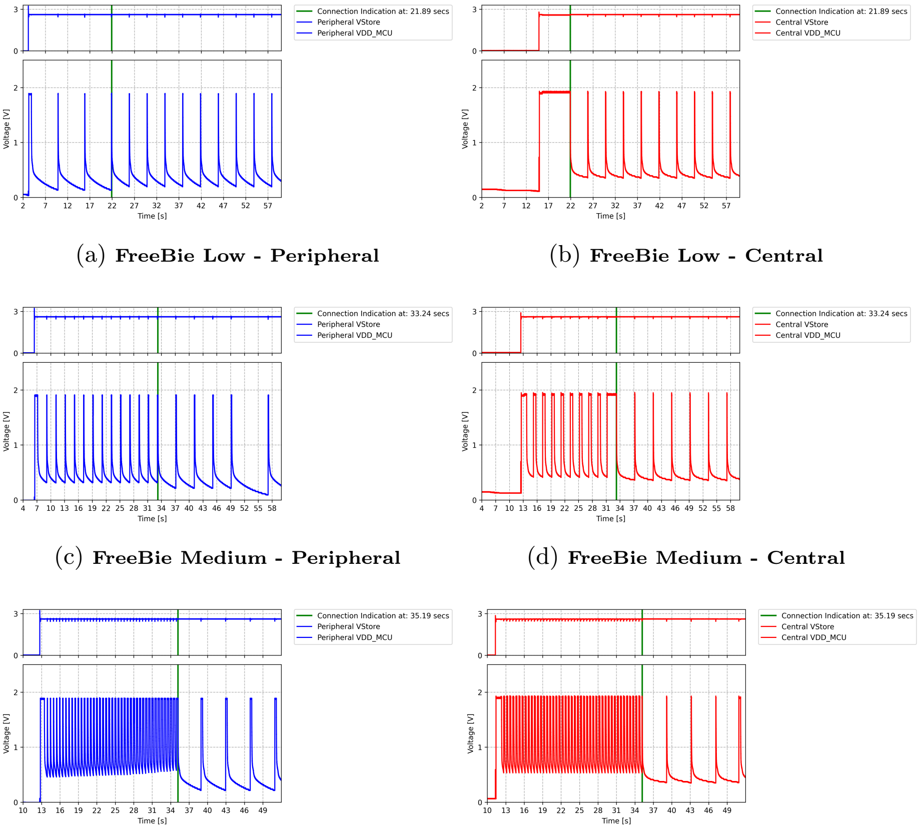
Time series voltage measurement plots were generated for each configuration using the same experimental setup discussed in Section 5.2.1, as shown in Figure 5.5. These plots capture connection establishment during periodic BLE events, with the device entering a sleep state between events, as evidenced by the $\mathrm{V_{DD\_MCU}}$ plot for both Central and Peripheral sides.

Additionally, current trend measured for each BLE configuration, as plotted in Figure 5.6, facilitated our comparative energy and power analysis. The average current required for BLE connection attempts under various configurations were calculated as illustrated in Table 5.5.

5.3.2 Experimental Setup

The experimental setup used in this study for comparative assessment is identical to the one stated in Section 5.2.1. The experiment recorded the first connection setup time since boot-up for both the CardioSync system and the reference system separately. The measurement of sensor read time is also recorded in the context of the CardioSync system. The experiment was conducted a total of 20 times using the CardioSync device. In the context of the reference system, the experiment is iterated six times for each of the three distinct BLE configurations.

After data collection, the quantification of energy consumption becomes a focal



(a) **FreeBie Low - Peripheral**

(b) **FreeBie Low - Central**

(c) **FreeBie Medium - Peripheral**

(d) **FreeBie Medium - Central**

(e) **FreeBie High - Peripheral**

(f) **FreeBie High - Central device**

Figure 5.5: **Real-time voltage measurement of FreeBie system. Each of the two plots in both devices for each configuration shows different measurements. From top to bottom in each device, 1) Plot of $\mathbf{V_{Store}}$: Supply Voltage and 2) Plot of $\mathbf{V_{DD\_MCU}}$: MCU voltage. The green vertical line through all the plots indicates the successful BLE connection setup event.**

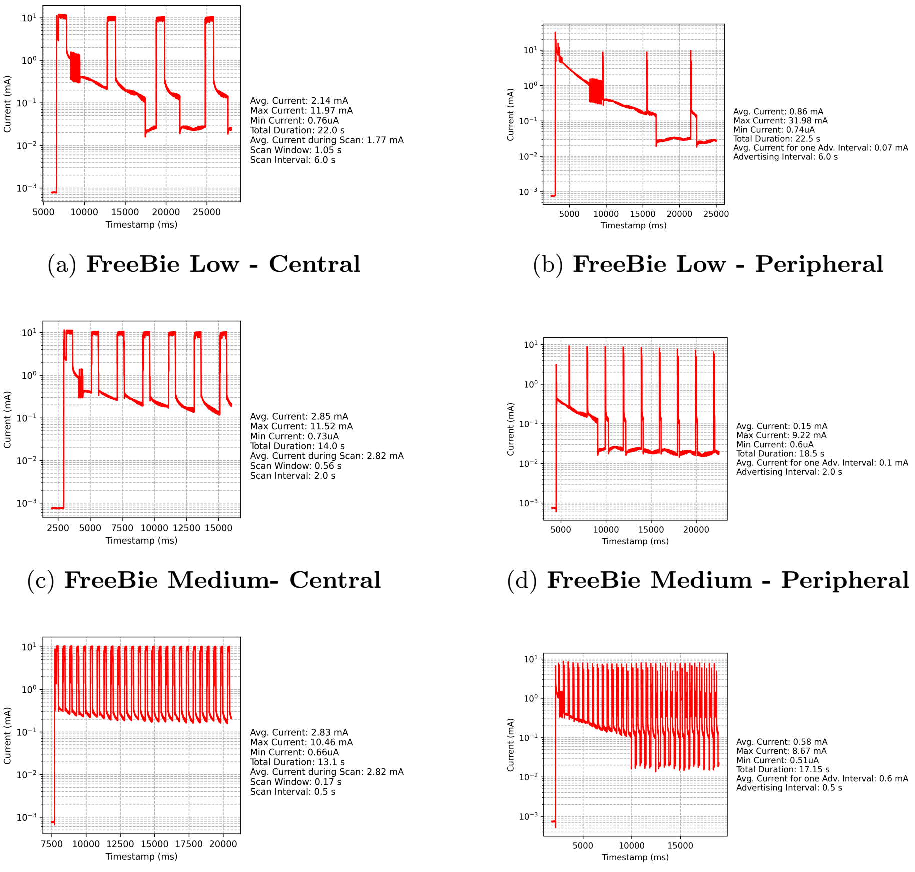
point. The computation of energy expended by the device during each experiment, leading up to connection establishment, is achieved using the formula

$$\mathrm{E} = \mathrm{V} \times I_{\text{avg}} \times T_{\text{Conn}},$$

where the average current I_{avg} is measured during distinct phases for both devices, the average voltage supply is 2.6 Volts (V) and T_{Conn} corresponds to the time taken to establish a BLE connection since boot up of the device.

5.3.3 Connection Setup Time Comparison

The bar plot in Figure 5.7 graphically represents the average connection duration across different systems. Though CardioSync exhibited a slightly longer average connection time compared to the low configuration of the reference FreeBie



(a) **FreeBie Low - Central**

(b) **FreeBie Low - Peripheral**

(c) **FreeBie Medium- Central**

(d) **FreeBie Medium - Peripheral**

(e) **FreeBie High- Central**

(f) **FreeBie High - Peripheral**

Figure 5.6: **Current measurement in idle state without connection establishment for different configurations of reference FreeBie system.**

system, this distinction remains negligible. In contrast, juxtaposing CardioSync with FreeBie Medium and FreeBie High configurations reveals a notable improvement in connection efficiency. Impressively, CardioSync achieves an average connection time **1.79 times faster than FreeBie medium and 1.65 times faster than FreeBie high**.

These insights offer valuable perspectives when considering practical real-world scenarios in the specific contexts chosen to represent the FreeBie reference systems:

- **Limited-Energy Scenario (FreeBie Low):** CardioSync's connection time efficiency, though slightly longer, positions it as a promising solution for scenarios with stringent energy constraints.
- **Balanced-Energy Scenario (FreeBie Medium):** CardioSync's significantly faster connection time underscores its potential for enhancing user

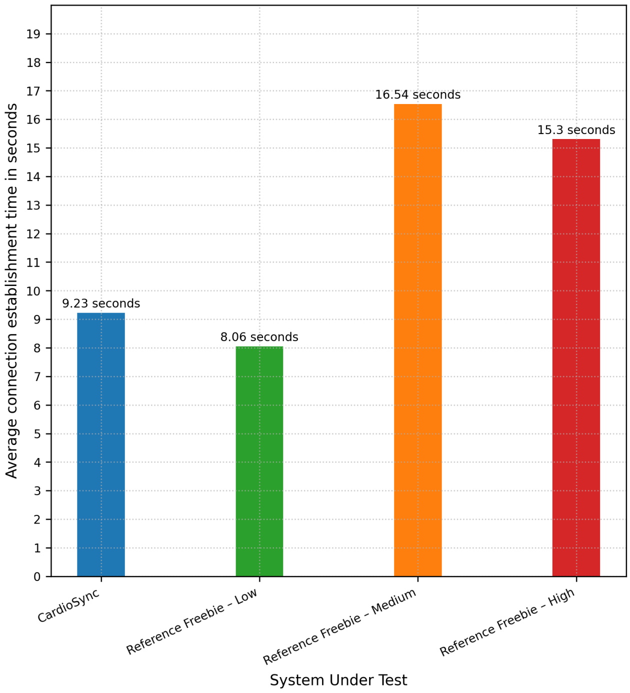


Figure 5.7: **Comparison of average BLE connection establishment time for each system.**

experiences in fitness tracking devices, where both energy resources and connection efficiency are crucial.

- **Abundant-Energy Scenario (FreeBie High):** CardioSync's impressive speed advantage makes it well-suited for scenarios prioritising connection efficiency due to surplus energy availability.

5.3.4 Energy Consumption Comparison

The bar chart depicted in Figure 5.8 provides an illustrative view of the average energy expended to establish a connection for each system. Notably, CardioSync records an average energy consumption of 144.8 mJ, encompassing the cumulative energy consumption from boot-up, sensor initialisation, the "Read and Synchronise" phase, and the "Sleep and Synchronise" phase. The partitioning of this energy consumption into distinct phases is depicted in the bar chart, showcasing the distribution of energy allocation for comprehensive insight. As anticipated, the "Sleep and Synchronise" consumes energy of 26.58 mJ, which is more similar to the energy expended by Naive FreeBie design with a low configuration. This confirms that sensor initialisation and active sensor reading, which employs a polling method, are active energy-consuming phases. In contrast, the reference FreeBie systems exhibit significantly lower energy expenditures—27.51 mJ for FreeBie Low, 41.64 mJ for FreeBie Medium, and 68.04 mJ for FreeBie High.

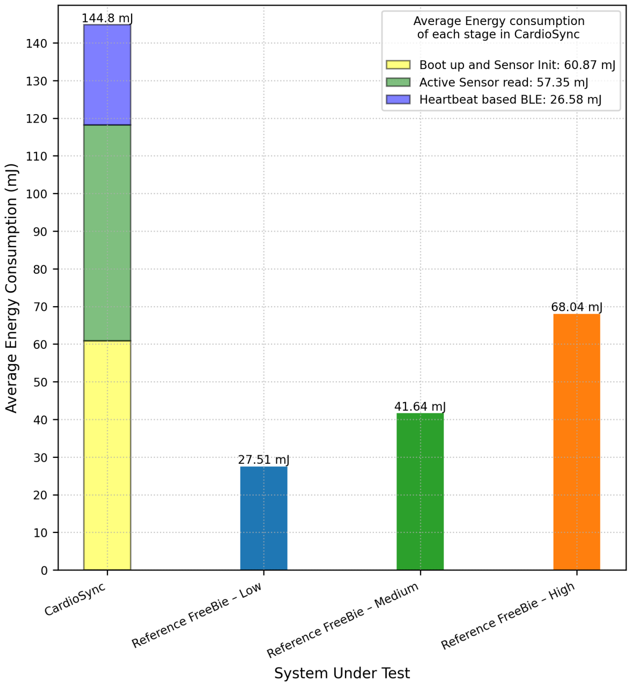


Figure 5.8: **Comparison of average energy consumed till connection setup for each system.**

Comparison of these results reveals that CardioSync consumes notably more energy compared to the reference FreeBie systems. This distinction is particularly evident when compared to FreeBie Low, where CardioSync's energy consumption is around **5.263 times higher**. Similarly, when contrasted with FreeBie Medium and FreeBie High, CardioSync's energy consumption is **3.48 and 2.13 times higher**, respectively.

This observed energy disparity aligns with the inherent characteristics of CardioSync, where the integration of the MAX30102 sensor introduces increased energy consumption as a trade-off for improved synchronisation and connection efficiency. Also in the context of the scenarios chosen for the FreeBie reference systems:

- **Limited-Energy Scenario (FreeBie Low):** CardioSync's higher energy consumption aligns with scenarios where energy constraints are paramount, raising questions about its feasibility in such settings.
- **Balanced-Energy Scenario (FreeBie Medium):** In scenarios like fitness trackers, where energy resources are limited, the energy efficiency of CardioSync becomes a crucial consideration for extended device functionality. This is particularly important when connection setups fail for extended periods, despite CardioSync demonstrating its ability to operate with intermittent power sources.

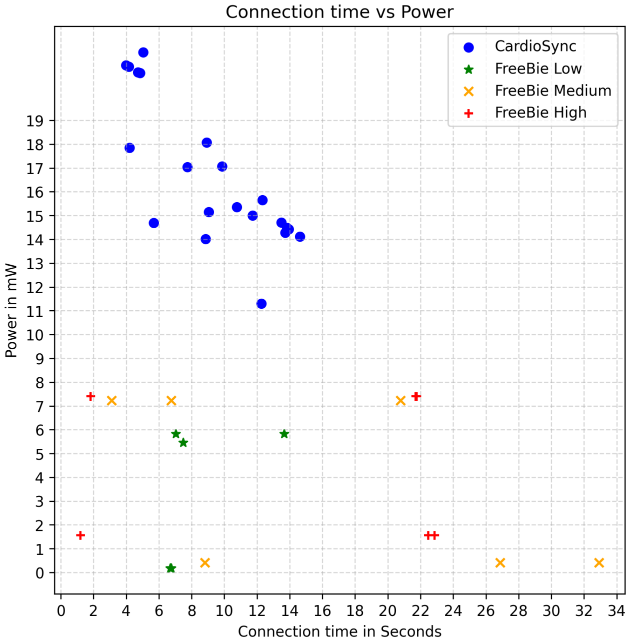


Figure 5.9: **Scatter plot showing connection time versus power.**

- **Abundant-Energy Scenario (FreeBie High):** While CardioSync's energy consumption is higher, its speed advantage suggests suitability for scenarios where connection efficiency is prioritised due to surplus energy availability.

5.3.5 Understanding Connection Setup versus Power

Figure 5.9 presents an insightful juxtaposition of connection time against power consumption for both the CardioSync and reference FreeBie systems. Notably, the distribution of experimental data points for the CardioSync system predominantly converges in the upper left quadrant of the graph. In contrast, the scatter plot depicting the reference FreeBie system showcases a more dispersed arrangement of data points, spanning across the lower half of the graph.

This disparity in the distribution of data highlights a pivotal distinction between the two systems. While the CardioSync system exhibits greater power consumption, it consistently achieves expedited connection times. In contrast, the reference FreeBie system, characterised by its asynchronous nature, demonstrates a wider range of connection times. This inconsistency in connection times, despite relatively lower power consumption, highlights the challenges inherent in achieving synchronisation within the battery-less FreeBie architecture. In essence, the scatter plot serves as a visual testament to the trade-offs between power consumption and connection time.

5.4 Discussion of Key Findings

In this section, we explore the implications and insights stemming from the comparative analysis between the CardioSync framework and the reference FreeBie system, shedding light on their performance and potential trade-offs.

- **Power and Energy Efficiency:** CardioSync demonstrated higher connection time efficiency, counterbalanced by increased power and energy consumption. This is due to the intentional inclusion of the MAX30102 sensor for synchronisation, which inherently demands additional energy. It is worth noting that the energy spent to establish a synchronised reconnection in case of connection timeouts is significantly lower in CardioSync compared to naive reference FreeBie systems.

- **Connection Time and Synchronisation:** CardioSync consistently achieved reduced connection times, while the reference FreeBie system exhibited a wider range of connection times, reflecting its asynchronous nature. Importantly, the readings from the reference system are after intentional synchronisation. Without this, the reference system's BLE advertising and scanning intervals remain asynchronous, rendering it unable to establish connections. In contrast, CardioSync achieves synchronisation effortlessly.

- **Trade-Offs and Implications:** CardioSync's higher energy usage, resulting in improved connection times, introduces trade-offs to consider in real-world applications. The dispersed connection times in the reference FreeBie system highlight the difficulty of achieving synchronisation.

In summary, the detailed evaluation of CardioSync against the reference FreeBie system aligns strategically with our research objectives and validates our research goals, effectively enhancing the discourse and possibilities within the battery-less domain.

Chapter 6

Future Work

While this thesis has presented an advancements in the field of battery-less embedded systems and synchronisation, there are several avenues for further exploration and improvement. The following outlines potential directions for future research

- **Fine-Tuning Sensor Utilisation**: Further optimising the utilisation of the heart rate sensor holds the potential for increased energy efficiency. The chosen sensor configurations detailed in Section A.1 are already finely balanced to conserve energy while excelling in heart rate detection. However, there remains room for additional optimisation through adjustments in duty cycle and the bit resolution of the ADC. Additionally, when measuring current while the sensor is interfaced, a noticeable residual current was detected in the Power Profiler Kit [28]. Addressing this could involve employing a power switch for the sensor's V_{DD} line, fully deactivating it when the MCU enters an OFF state. This approach has the potential to yield modest reductions in energy consumption.

- **Adaptive Synchronisation Strategies:** Developing dynamic synchronisation strategies that can adapt to connection setup failures by intelligently adjusting parameters such as widening the scan window or prolonging the advertising duration. This would involve monitoring for connection attempts and failure counts in post processing stage of algorithm and adjusting synchronisation intervals accordingly.

- **Exploring Different Sensor Platforms:** While the chosen heart rate sensor - MAX30102 has proven effective, considering alternative sensor platforms could provide insights into sensor energy efficiency and accuracy trade-offs. Exploring newer sensor technologies could yield novel opportunities.

- **Real-World Deployments and Practical Testing:** Conducting real-world deployments and testing CardioSync in various scenarios, including wearable applications or IoT systems, would validate its performance outside controlled environments. Practical challenges and optimisations can be explored.

Chapter 7

Conclusions

In this thesis, we have introduced the CardioSync framework, a novel solution that addresses a significant limitation in the state-of-the-art intermittent BLE device, FreeBie. The core challenge of achieving true intermittency between two end nodes and forming a reliable BLE network has been successfully overcome through CardioSync. This framework harnesses the capabilities of the low-power heart rate monitoring sensor, to synchronise connection setups between two battery-free nodes, employing the human heart rate as a shared clock pulse. By leveraging the inherent rhythm of the human heart, CardioSync achieves an average connection time that is nearly **1.8 times faster** compared to the FreeBie model's asynchronous periodic connection setup strategy. As promising as this is, it comes with an understandable trade-off of increased power demand for the connection setup due to sensor integration. The CardioSync framework demands **3.70 times more power** than reference FreeBie model.

This novel approach paves the way for a wide range of applications, especially in the field of Body Sensor Networks (BSN). The ability to establish connections between battery-free devices without relying on external power sources holds immense promise for healthcare monitoring, environmental sensing, and beyond. Looking ahead, the implications of the CardioSync framework on wearable health monitoring and IoT are significant. Its success opens avenues for more efficient and sustainable IoT deployments, reducing the environmental impact of battery disposal. With each synchronised connection established by CardioSync, we move closer to realising a more connected and sustainable world.

Bibliography

[1] packetcraft-inc . Github - packetcraft-inc/stacks: Packetcraft Open Source products: Controller, Host, Profile and Mesh. `https://github.com/packetcraft-inc/stacks`, 11 2020.

[2] Skip Descant . Battery-Less IoT Could Change How, When We Gather Data. `https://www.govtech.com/products/battery-less-iot-could-change-how-when-we-gather-data`, 5 2021.

[3] sparkfun . Sparkfun_MAX3010x_Sensor_Library/src/spo2_algorithm.cpp at master · sparkfun/SparkFun_MAX3010x_Sensor_Library. `https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/blob/master/src/spo2_{}algorithm.cpp`.

[4] TUDelftSSL . Researchers invent Bluetooth that keeps on working even if power runs out. `https://www.tudelft.nl/en/2022/tu-delft/researchers-invent-bluetooth-that-keeps-on-working-even-if-power-runs-out`.

[5] TUDSSL . Github - TUDSSL/DIPS. `https://github.com/TUDSSL/DIPS`.

[6] TUDSSL . Github - TUDSSL/FreeBie. `https://github.com/TUDSSL/FreeBie`.

[7] Stanislav Asenov and Dimitar Tokmakov. Battery free wireless lorawan end sensor node for iot applications. In *2020 28th National Conference with International Participation (TELECOM)*, 2020.

[8] Bansal Atul, Kumar Swarun, and Iannucci Bob. Does ambient rf energy suffice to power battery-free iot? In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, 2020.

[9] Jia-Hao Bai, Tzu-Hang Huang, Zen-Dar Hsu, Li-Ren Huang, Cihun-Siyong Alex Gong, and Chin Hsia. Battery-free sensor tag for iot applications. In *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2017.

[10] Sonia Bradai, Ghada Bouattour, Slim Naifar, and Olfa Kanoun. Electromagnetic energy harvester for battery-free iot solutions. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020.

[11] Thai-Ha Dang, Ngoc-Hai Dang, Viet-Thang Tran, and Wan-Young Chung. B2eh: Batteryless ble sensor network using rf energy harvesting. In *2023 IEEE Applied Sensing Conference (APSCON)*, 2023.

[12] Jasper de Winkel, Haozhe Tang, and Przemysław Pawełczak. Intermittently-powered bluetooth that works. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, New York, NY, USA, 6 2022.

[13] Cordis Europa. Up to 78 million batteries will be discarded daily by 2025, researchers warn. `https://cordis.europa.eu/article/id/430457-up-to-78-million-batteries-will-be-discarded-daily-by-2025-researchers-warn`, 5 2023.

[14] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, Luca Benini, and Rajesh Gupta. Pible: Battery-free mote for perpetual indoor ble applications: Demo abstract. In *Proceedings of the 5th Conference on Systems for Built Environments*, 2018.

[15] Mark A. Hanson, Harry C. Powell Jr., Adam T. Barth, Kyle Ringgenberg, Benton H. Calhoun, James H. Aylor, and John Lach. Body area sensor networks: Challenges and opportunities. *COMPUTER*, 2009.

[16] Maxim Integrated. Maxrefdes117 heart-rate and pulse-oximetry monitor. `https://www.analog.com/en/design-center/reference-designs/maxrefdes117.html`, 2016.

[17] Maxim Integrated. Max30102 datasheet. `https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf`, 2018.

[18] Blessy Johny and Alagan Anpalagan. Body area sensor networks: Requirements, operations, and challenges. *IEEE Potentials*, 2014.

[19] Rongzhou Lin, Han-Joon Kim, Sippanat Achavananthadith, Selman A. Kurt, Shawn C. C. Tan, Haicheng Yao, Benjamin C. K. Tee, Jason K. W. Lee, and John S. Ho. Wireless battery-free body sensor networks using near-field-enabled clothing. *Nat Commun 11*, 1 2020.

[20] Benny Lo and Guang-Zhong Yang. Architecture for body sensor networks. In *Perspectives in Pervasive Computing*, 2005.

[21] Soumyajit Mandal, Lorenzo Turicchia, and Rahul Sarpeshkar. A low-power, battery-free tag for body sensor networks. *IEEE Pervasive Computing*, 2010.

[22] Lingfei Mo, Shaopeng Liu, Robert X. Gao, and Patty S. Freedson. Energy-efficient and data synchronized body sensor network for physical activity measurement. In *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013.

[23] Saleae. Logic Analyzers from Saleae. `https://www.saleae.com/`.

[24] Aarti Sangwan and Partha Pratim Bhattacharya. Wireless body sensor networks: A review.

[25] Nordic Semiconductors. nrf52840 DK. https://www.nordicsemi.com/Products/Development-hardware/nRF52840-DK.

[26] Nordic Semiconductors. nrf52840 product specifications. https://infocenter.nordicsemi.com/index.jsp?topic=%2Fps_nrf52840%2Ftwim.html&cp=5_0_0_5_30_8&anchor=concept_twim_timings, 2021.

[27] Nordic Semiconductors. nrf5 SDK. https://www.nordicsemi.com/Products/Development-software/nRF5-SDK, 8 2023.

[28] Nordic Semiconductors. Power Profiler Kit II. https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2, 8 2023.

[29] Amir Mohammad Shojaei. Interfacing max30102 pulse oximeter heart rate module with arduino. https://electropeak.com/learn/interfacing-max30102-pulse-oximeter-heart-rate-module-with-arduino/.

[30] Ashish Kumar Sultania and Jeroen Famaey. Batteryless bluetooth low energy prototype with energy-aware bidirectional communication powered by ambient light. *IEEE Sensors Journal*, 2022.

[31] Kyriatzis Vasileios, Samaras Nicholas S., Stavroulakis Pavlos, Takruri-Rizk Haifa, and Tzortzios Stergios. Enviromote: A new solar-harvesting platform prototype for wireless sensor networks / work-in-progress report. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007.

[32] Yifei Wang, Huizhi Wang, Jin Xuan, and Dennis Y.C. Leung. Powering future body sensor network systems: A review of power sources. *Biosensors and Bioelectronics*, 2020.

Appendix A

APPENDIX

A.1 MAX30102 Sensor Configuration

The initialisation of the MAX30102 sensor happens through nRF's TWI driver. The nRF52840 board [25] supports two devices interfaced through TWI interface for I^{2}C or SPI communication. The MAX30102 sensor is assigned to *TWI instance 1* among the two TWI devices, as *TWI instance 0* is exclusively allocated for SPI communication with the external FRAM module in the FreeBie architecture. After successful enabling of TWI interface for the sensor, configuration of various registers of sensor is performed to ensure optimal performance while minimising power consumption. To achieve this, the `MAX30102_init()` function is implemented. Within this function, several critical registers are configured as follows:

- **REG_INTR_ENABLE_1 (0x02U):** This register is set with the value 0xC0 to enable the "A_FULL_EN" interrupt, which triggers when the FIFO is full and data is ready for reading.
- **REG_FIFO_CONFIG (0x08U):** The FIFO configuration register is set with settings such as averaging 4 samples per entry, setting FIFO size to 32 (which triggers Interrupt 1 when full), and disabling FIFO rollover.
- **REG_MODE_CONFIG (0x09U):** Configures the operational mode of the sensor. In the context of CardioSync, heart rate mode is selected while disabling Spo2 measurements.
- **REG_SPO2_CONFIG (0x0AU):** Sets parameters for data acquisition including a sampling rate of 50 samples per second, ADC range of 4096nA, pulse width of 118us, and 16-bit ADC resolution.
- **REG_LED1_PA(0x0CU), REG_LED2_PA (0x0DU):** Configures LED current; for CardioSync, 1mA is set for both LED1 and LED2.

By meticulously configuring these registers, the MAX30102 sensor is primed to operate within desired parameters and less power consumption, yet capturing accurate blood flow data for heart rate analysis.

A.2 Validation of CardioSync System with Intermittent Power

The CardioSync system was tested across various square wave periods with different duty cycles as shown in Table A.1, demonstrating its adaptability to different supply configurations.

Exp. number	Turn ON time in seconds	Turn OFF time in seconds	Total Period in seconds	Duty Cyle in percentage	Connection setup time since boot up in seconds
1	4	1	5	80.00%	11.718
2	4	2	6	66.67%	13.702
3	4	3	7	57.14%	29.337
4	4	4	8	50.00%	33.641
5	5	1	6	83.33%	9.271
6	5	2	7	71.43%	8.696
7	5	3	8	62.50%	9.19
8	5	4	9	55.56%	37.123
9	5	5	10	50.00%	14.254
10	6	1	7	85.71%	8.265
11	6	2	8	75.00%	8.993
12	6	3	9	66.67%	10.714
13	6	4	10	60.00%	10.986
14	7	1	8	87.50%	8.927
15	7	2	9	77.78%	9.881
16	7	3	10	70.00%	11.341
17	8	1	9	88.89%	11.388
18	8	2	10	80.00%	4.133

Table A.1: **Validation of CardioSync system across various square wave periods with different duty cycles. With turn OFF time more than 3 seconds and turn ON time less than 6 seconds, connection setup time is longer than other parameters of square wave supply.**