A Real-time 12-lead Electrocardiogram Remote Patient Monitoring and Analytics Framework

by

Ahmed Badr

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical and Computer Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

October 2022

THESIS EXAMINATION INFORMATION

Submitted by: **Ahmed Badr**

Doctor of Philosophy in **Electrical and Computer Engineering**

Thesis title:
A Real-time 12-lead Electrocardiogram Remote Patient Monitoring and Analytics Framework

An oral defense of this thesis took place on October 17, 2022, in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Sanaa Alwidian
Research Supervisor	Dr. Khalid Elgazzar
Examining Committee Member	Dr. Qusay Mahmoud
Examining Committee Member	Dr. Akramul Azim
University Examiner	Dr. Carolyn McGregor
External Examiner	Dr. Abdallah Shami – University of Western Ontario

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

This work presents XBeats: A machine learning-based framework for real-time electrocardiogram monitoring and analysis that uses edge computing and data analytics for early anomaly detection. The framework encompasses a data acquisition ECG patch with 12 leads to collect heart signals, perform on-chip processing, and transmit the data to healthcare providers in real-time for further analysis. The ECG patch provides a dynamically configurable selection of the active ECG leads for transmission to the backend monitoring system. The selection ranges from a single ECG lead to a complete 12-lead ECG testing configuration. XBeats implements a lightweight binary classifier for early anomaly detection to reduce the time to action should abnormal heart conditions occur. This initial detection phase is performed on an edge node and alerts can be configured to notify designated healthcare providers. Further deep analysis can be performed on the full-fidelity 12-lead data sent to the backend. A fully functional prototype of the XBeats is implemented to demonstrate the feasibility and usability of the proposed system. XBeats can achieve up to 95.30% detection accuracy for abnormal conditions while maintaining a high data acquisition rate of up to 480 samples per second. Besides a systematic energy consumption profiling criteria is provided for evaluating participating hardware components in the XBeats ECG patch. We isolate each hardware component to find power-intensive processes, discover energy consumption patterns, and measure voltage, current, power, and energy consumption for a given period. The proposed optimization techniques demonstrate significant improvements to the hardware components. The results show that optimizing the data acquisition process saves 8.2% compared to the original power consumption and 1.62% in data transmission over BLE, thus extending the lifetime of the device. Lastly, we optimize the data logging operation to save 54% of data initially written to an external drive. Moreover, the analytical results of the energy consumption profile show that the ECG patch provides up to 37 hours of continuous 12-lead ECG acquisition.

Keywords: remote patient monitoring; electrocardiogram; telemedicine; cardiovascular diseases; real-time streaming

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize the University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Ahmed Badr

STATEMENT OF CONTRIBUTIONS

- Badr, A., & Elgazzar, K. (2022). A Framework for Real-Time Remote ECG Monitoring and Diagnoses", *(Submitted ICCSPA 2022).*

  I performed the majority of the conceptualization, methodology, software implementation, validation, investigation, writing of original draft, preparation and writing the manuscript.

- Badr, A., Rashwan, A., & Elgazzar, K. (2022, September). An Application-Specific Power Consumption Optimization for Wearable Electrocardiogram Devices," The 47th IEEE Conference on Local Computer Networks (LCN). IEEE.

  I performed the majority of the conceptualization, methodology, software implementation, validation, investigation, testing, writing of original draft, preparation and writing the manuscript.

- Badr, A., Badawi, A., Rashwan, A., & Elgazzar, K. (2022, May). 12-Lead ECG Platform for Real-time Monitoring and Early Anomaly Detection. In 2022 International Wireless Communications and Mobile Computing (IWCMC) (pp. 973-978). IEEE.

  I performed the majority of the conceptualization, methodology, software implementation, validation, investigation, testing, writing of original draft, preparation and writing the manuscript.

- Badr, A., Badawi, A., Rashwan, A., & Elgazzar, K. (2022). XBeats: A Real-Time Electrocardiogram Monitoring and Analysis System. Signals, 3(2), 189-208.

  I performed the majority of the conceptualization, methodology, software implementation, validation, investigation, testing, writing of original draft, preparation and writing the manuscript.

- Ouda, H., Badr, A., Rashwan, A., Hassanein, H. S., & Elgazzar, K. (2022, May). Optimizing Real-Time ECG Data Transmission in Constrained Environments. In ICC 2022-IEEE International Conference on Communications (pp. 2114-2119). IEEE.

  I performed the conceptualization, investigation, empirical testing, validation, writing the original draft, preparation and editing of the manuscript.

- Badawi, A. A., Badr, A., & Elgazzar, K. (2021, June). ECG Real-time Monitoring and Heart Anomaly Detection Reimagined. In 2021 IEEE 7th World Forum on Internet of Things (WF-IoT) (pp. 326-331). IEEE.

  I performed the conceptualization, investigation, software implementation, validation, testing and editing of the manuscript.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervisors for their continuous help in my work and guidance in completing this thesis. I would also like to thank them for their valuable advice in different contexts of this document.

My heartfelt thanks to Dr. Khalid Elgazzar for his great help, support, and patience during the time of this work. And many thanks to Dr. Abdulmonem Rashwan for his valuable remarks, efforts and guidance during the various stages of this thesis work.

Finally, I would like to express my great appreciation to my dear family and friends for their continuous care, patience, encouragement, and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AFE	Analog Frontend
ECG	Electrocardiogram
IoT	Internet of Things
BLE	Bluetooth Low Energy
RTOS	Realtime Operating System
MQTT	Message Queueing Telemetry Transport
CVD	Cardiovascular Diseases
ML	Machine Learning
CNN	Convolutional Neural Network
ADC	Analog to Digital Converter
MCU	Microcontroller Unit
TI	Texas Instruments
RPI	Raspberry PI
WHO	World Health Organization
RF	Random Forest
SVM	Support Vector Machine
ET	Extra Tress
KNN	K-Nearest Neighbors
LR	Linear Regression

Chapter 1. Introduction

1.1 Heart Diseases and Statistics

According to the Centers for Diseases Control and Prevention [1], in the United States, 20.1 million adults were diagnosed with cardiovascular diseases (CVDs) in 2020. The World Health Organization (WHO) in 2019 identified CVDs as the number one cause of death [2]. Over 17.9 million people died globally from CVDs in 2019, representing 32% of all global deaths. In 2020, nearly 697,000 people died because of heart disease, which made it one of the leading causes of death in the U.S that year [1]. Similarly, according to the public health agency of Canada [3], about 2.4 million Canadian adults aged 20 and over are diagnosed with heart disease. CVDs are identified as a group of irregular heart rhythms called arrhythmia. Arrhythmia includes coronary heart disease, cerebrovascular disease, rheumatic heart disease, and other conditions.

People with previous CVDs or generally irregular heart rhythms visit hospitals to monitor their heart conditions. However, in low or middle-income countries, where three-quarters of CVD deaths occur, hospitals struggle to provide long-term heart monitoring [2]. Therefore, it is essential to monitor heart activity regularly to prevent CVDs and minimize complications like premature death. Most importantly, during challenging times like the period of the COVID-19 pandemic, hospitals aim to reduce the number of in-person visits to reduce possible infections and strive to find alternatives to serve patients in dire need remotely. Therefore, there is an urgent need for remote ECG monitoring for long-term cardiac diagnoses. As a result of these demands, a shift in the healthcare landscape is taking place where everyone is moving toward preventive care in an age where nearly everyone is digitally connected [4], [5]. Therefore, the healthcare industry is seeing a lot of connected health devices and remote patient monitoring technologies that enable physicians to monitor patients without having to come into contact with them. Furthermore, a recent study [6] emphasizes the positive impacts of continuous remote monitoring in helping people manage chronic conditions.

1.2 Heart Monitoring and Electrocardiogram

Electrocardiogram (ECG) is the oldest and most used test by cardiologists for understanding heartbeat "rhythms" [7]. The heart assembles an electrical activity for every beat captured in an ECG test. Cardiologists build their diagnoses on heart conditions by analyzing the activity of the heart that is rendered into a waveform format. Diagnosing heart disease is commonly a manual process. ECG refers to a 12-lead ECG test (or EKG) and is defined as a non-invasive diagnostic test that evaluates the electrical pulses of the heart to assess possible heart conditions [7]. In order to carry out an ECG test, ten cables are attached to a patient's body to obtain 12 electrical views of the heart. The electrical pulses of the heart are received via flat metal electrodes placed on the patient's body to detect electrical charges generated by the heart as it beats. Then, signals are relayed via wires to a device that encodes the received analog signals and displays the signals on a monitor [8]. The term electrode is used to express the ten cables, and the term lead represents the 12 electrical views of the heart. Each ECG lead shows the heart from a specific angle, where the combination of the 12 views constitutes the standard 12-lead ECG test. The standard ECG 12-leads are obtained from four electrodes attached to the patient's limbs (i.e., right and left arm, right and left leg). At the same time, the rest are obtained from six electrodes attached to the patient's chest), labelled from V1 to V6. The exact orientation of each lead can be found in [7]. Furthermore, each lead of an ECG test constitutes a standard pattern comprising three wave components named P, QRS and T or the PQRST feature points [9].

Holter monitor is a widely known device used by practicing specialists in hospitals and cardiology clinics. Those devices use wired electrodes to connect directly to monitors for rendering the received signals. Usually, Holter monitors and similar devices come with diagnostic tools to analyze heartbeats in real-time. Notably, similar devices provide similar functionality to Holter monitors, also called event monitors [10]. Event monitors are portable devices someone can carry around day and night until symptoms occur. The main difference is that Holter monitors record ECG signals regardless of the heart conditions during the recording period. In contrast, event monitors are designed to be automatically activated when symptoms are experienced and then record current heartbeats.

Alternatively, the patient can manually activate them if an abnormal heart rhythm is felt.

The process of identifying the heart's diagnosis is known to be a manual process carried out by cardiologists [7]. Furthermore, heart conditions could be misdiagnosed (or an anomaly entirely missed) due to intermittent heart irregularity (aka. arrhythmia) [8]. Accordingly, recent event monitors can now be configured to be worn continuously, and others are applied to the skin and activated automatically when symptoms are experienced [10]. Event monitors overcome the Holter monitor shortcomings as they can provide heart monitoring for extended periods that can last for weeks. However, event monitors are not designed to trigger an emergency response for life-threatening arrhythmia due to a processing lag of several minutes or lack of connectivity. Given the abovementioned requirements, patients are advised to stay in bed and be hospitalized for the test period. However, the recent COVID-19 pandemic tested existing healthcare infrastructures, revealing numerous operating challenges, including a limited number of beds, intensive care units and device management. Consequently, remote patient monitoring (RPM) should be embedded within hospitals and healthcare providers as a primary service instead of considering RPM as a secondary service with independent management systems,

1.3 ECG Remote Patient Monitoring

The provision of RPM has become intrinsic in changing traditional healthcare services and abilities to monitor and manage patients remotely [11]. Patients with chronic diseases (i.e., cardiovascular (CVDs), diabetes and cancer) require ongoing medical attention and limited activities in everyday routines. Chronic diseases last for long periods that can be in years. RPM gives healthcare providers access to their patients without requiring their patients to visit hospitals to perform ECG testing. Standard ECG testing can be performed remotely with the help of a reliable ECG acquisition device and an RPM framework [12]. There are multiple configurations of ECG tests using Holter monitors, like the number of leads or data acquisition period. The selection between the common two to three leads or the standard 12 leads is left to the cardiologist's discretion or as the patient's heart condition develops [8]. The 2- to 3-lead Holter monitors are used for detecting heart rate and its rhythm.

Conversely, a standard 12-lead ECG would be needed to screen patients for possible cardiac ischemia and help healthcare providers quickly identify patients who have ST-elevation myocardial infarction (i.e., heart attack) and perform the appropriate medical intervention in time [13]. A standard 12-lead ECG requires a Holter monitor that can be installed at hospitals or carried by patients. Usually, Holter monitors installed at hospitals come with diagnostic tools to analyze heartbeats in real-time, in contrast to the devices carried by patients, which only provide offline ECG data logging. Moreover, previous ECG charts are not available for comparison with ECG current signals to observe any potential correlation between previous ECG charts and current ECG charts. Furthermore, heart conditions could be misdiagnosed (or an anomaly entirely missed) due to intermittent heart irregularity (aka arrhythmia) [13].

Practically, long-term ECG monitoring and measurement devices are intended to be standalone, lightweight, wearable, flexible, and facilitate seamless integration with the electrodes attached directly to the chest. However, many of the recent wearable and lightweight ECG measurement devices developed for continuous measurement of the ECG signals lacks in providing one or many of the features mentioned above. Consequently, in response to the rapid digital revolution and the COVID-19 pandemic, the healthcare landscape has rapidly shifted from physical to virtual care and telemedicine. The provision of remote patient monitoring has changed the traditional healthcare abilities to monitor and manage patients [14]. Despite the significant research efforts brought by MedTech companies and the research community, continuous remote ECG monitoring still lacks comprehensiveness and completeness compared to the services offered in hospitals and clinics [14], [15].

1.4 Thesis Objectives and Contributions

This thesis proposes a new framework to provide unbounded continuous remote ECG monitoring using a lightweight 12-lead ECG smart patch that integrates intelligent signal analysis and offers two heartbeat classification phases. The framework encompasses two major components:

(1) The hardware component is responsible for the data acquisition represented in the proposed smart patch (i.e., XBeats) for ECG monitoring; The smart patch supports backward compatibility with various combinations of ECG leads, not just the standard 12-lead ECG testing. This feature enables healthcare providers to customize the number of enabled ECG leads during the ECG test according to the developing health conditions of their patients. Furthermore, the hardware enables wireless connectivity using low-power communication modules to facilitate seamless, remote, long-term cardiac monitoring and diagnoses. Similarly, the hardware maintains a continuous live log of ECG signals collected by the hardware on local storage, serving as a backup in events when wireless connectivity to the internet gateway is interrupted.

(2) The software component is responsible for the data streaming and analytics implemented at the backend. The backend gathers all the data acquired by the ECG smart patch, stores it in a high-performance database, and trains machine learning algorithms to perform real-time data diagnosis and predictions. Data streaming from the patch is carried out by Apache Kafka [16], a high-performance open-source real-time streaming engine. Kafka supports unbounded data streams with a latency of less than ten milliseconds and allows the integration of distributed computing

frameworks to carry out advanced classification models. The integration of the latest IoT communication protocols (e.g., Bluetooth Low Energy (BLE) and MQTT (Message Queueing Telemetry Transport)) enables the connection between participating components in the framework. Furthermore, the proposed framework includes a modular frontend user interface for displaying the real-time ECG stream, preliminary diagnosis, records access and management, and robust notification service that interfaces with smart home devices.

1.4.1 Thesis Contributions

(1) Improve remote real-time ECG monitoring for long-term cardiac diagnoses by developing a lightweight and wearable ECG device (i.e., XBeats) to not interrupt the everyday lifestyle of the patient. At the same time, we are supporting standard 12-lead ECG recording using low-power hardware components for data acquisition, transmission, and logging.

(2) Optimize data transfer between the smart patch and the backend system based on developing conditions while utilizing BLE communication protocols in receiving data and controlling the device. Healthcare providers may configure the system using three modes of operations: a) **continuous mode**, where ECG signals are lively streamed regardless of heart conditions; b) **triggered mode**, where the patch only sends a beaconing alive signal during normal conditions and only sends ECG signals when a

possible abnormality is detected, it also supports varying fidelity transmission operation, where low fidelity ECG signals (i.e., a few leads at a low sampling rate) are sent on normal conditions and high fidelity ECG signals (i.e., all 12-leads at high sampling rate) are sent when abnormal arrhythmia is detected; c) **disconnected mode** when the patch is completely disconnected from all bonded Bluetooth devices or wireless gateways, it applies the same logic as in the triggered mode, but additional local logging to a flash storage function is enabled to keep records of the entire disconnectivity period until a bonded Bluetooth device becomes in range.

(3) Develop an RPM framework enabling real-time streaming and analysis of ECG data through a backend architecture that precisely process ECG signals received from the ECG patch and predicts a broad spectrum of possible developing conditions. The backend utilizes scalable, fault-tolerant, and secure streaming engines to accommodate the high volume of streamed ECG and vital information. The framework uses advanced Machine Learning (ML) algorithms and Convolutional Neural Networks (CNN) to perform deep analytics and build a correlation between real-time and historical data for better analysis and predictions.

(4) Build a power consumption benchmark for continuous real-time 12-lead ECG acquisition devices with BLE connectivity, which includes investigating the power consumption profile of the proposed ECG patch.

The investigation involves three main components in the ECG patch hardware design: the analog to digital data converter, the communication module represented in the main controller and the local SD storage.

(5) Optimize the power consumption profile of the ECG patch concerning each component under investigation. Besides, evaluate the impact of the applied optimization techniques regarding the resultant power consumption profile and operation lifetime of the device.

1.5 Thesis Outline

The thesis is organized into six chapters as follows:

1. **Chapter 1** introduces the topic and outlines the main objectives of this research and the thesis organization.
2. **Chapter 2** reviews related work in the RPM domain for remote ECG testing and real-time analytics. It categorizes the literature into two main categories according to works introduced in the research community and the industry (i.e., commercialized solutions). Likewise, each category is divided into subcategories accordioning to the number of ECG leads (i.e., single ECG lead, two or more ECG leads and standard 12-lead ECG).
3. **Chapter 3** presents XBeats, a patent-pending lightweight 12-lead ECG smart patch for long-term cardiac diagnoses. The chapter is divided into two main sections. The first section gives details about the XBeats ECG patch hardware components and prototyping. The second section builds on the

XBeats proposed hardware design and explores ways for energy consumption reduction, prolonging the expected battery operation time.

4. **Chapter 4** presents the proposed XBeats RPM framework architecture and design to operate the XBeats ECG patch. It demonstrates a comprehensive end-to-end solution for real-time ECG monitoring and analytics.
5. **Chapter 5** provides the implementation and prototyping details for the proposed hardware for the ECG patch and the real-time RPM framework setup.
6. **Chapter 6** discusses the performance evaluation and experimental results on each component utilized in the proposed end-to-end real-time standard 12-lead ECG data monitoring and analytics framework.
7. **Chapter 7** provides concluding remarks and future directions.

Chapter 2. Background and Related Work

In efforts to provide an in-depth review of the literature for this thesis, the literature review is organized to evaluate the works proposed in achieving remote ECG testing using wearable devices. Then we evaluate the works concerning ECG data acquisition devices concerning the number of provided ECG leads, utilized communication technologies, modes of operation, and hardware components. The ECG acquisition devices presented in the literature share standard hardware components and designs which can be categorized by the number of ECG leads used. Consequently, we provide a detailed overview evaluating the works introduced in the industry from a commercial perspective in providing remote ECG monitoring services and diagnoses. Furthermore, we review research works proposed concerning remote patient monitoring systems and underlying architectures. The review highlights the gaps, disconnectivity and overlapping components presented in the presented works in enabling remote ECG data streaming and analytics. Furthermore, it outlines the shortcomings that face those solutions if they were to operate in the future of e-health and preventive care in the healthcare industry. Lastly, section 2.4 provides a summary of the chapter.

2.1 Introduction

Due to the intensive amount of daily health-related data, there is a dire need for efficient data analysis techniques to process data in real-time and empower the predictive capability of healthcare applications. Furthermore, the prevalence of chronic illnesses is increasing globally. Currently, wearable sensors and communication protocol developments contribute in ways that will soon transform remote healthcare monitoring services. The first of these improvements is remote patient monitoring (RPM). RPM systems collect vital signs from patients by non-invasive procedures and their real-time transmission to healthcare providers. The information collected by RPM devices may assist clinicians in making the best choice possible at the appropriate moment. Therefore, many efforts have been conducted to advance remote ECG monitoring systems to match or exceed the performance of the ECG testing administered at hospitals and healthcare facilities [17].

This chapter intends to identify research gaps in defining the lifecycle required to perform a standard ECG monitoring system and highlight existing solutions introduced in the literature. Then we illustrate in detail the critical design advantages and shortcomings of the discussed ECG solutions and efforts regarding the primary functions of standard remote ECG testing for patients with chronic heart diseases. The growing interest in the research community concerning remote health and patient monitoring has resulted in a multitude of proposed systems, many of which focus on ECG monitoring. Therefore, the literature review presented in this chapter shows the prevalence of real-time ECG monitoring and diagnosis. While real-time ECG monitoring and diagnosis is receiving significant attention from numerous affiliations, the topic has been introduced as separate pieces and components of a complete and comprehensive remote ECG testing framework. Therefore, remote ECG monitoring has been the focus of the research community for many years and is divided into three branches: (1) hardware development for remote and wearable ECG acquisition devices; (2) leveraging the advancements in arithmetic intelligence and machine learning algorithms in performing automated diagnoses and predictions concerning heart condition of patients; and (3) software development and integrations in providing an enabling infrastructure for unbounded streams of ECG data and real-time event processing and analytics on the received data.

2.2 Remote ECG Monitoring Devices

The delivery of RPM services requires a reliable data acquisition service to con-verge vital medical charts and information directly from the patient. Data acquisition services require the presence of wearable wireless sensors (i.e., Apple Watch, QardioCore, Kardia). However, wearable wireless medical devices entail a strict set of requirements to be considered for medical applications and critical patient conditions. This set of requirements is translated to a group of high-level hardware components that include but are not limited to high-resolution data acquisition modules or sensors. It also includes reliable communication modules (e.g., Bluetooth Low Energy (BLE), Zigbee) and low-power processing units and storage units (e.g., SD Cards, Flash Storage) [10, 11]. Due to some medical conditions (i.e., ECG test), it acquires vital signals or information without interruptions during data acquisition [12, 13]. Moreover, RPM systems enable medical acquisition devices (i.e., Electrocardiogram (ECG) and heart monitoring) to transmit vital information continuously to the healthcare provider.

2.2.1 Single Lead ECG

Single Lead ECG devices are very common today. A single ECG lead covers limited heart regions, making it suitable for heart activity monitoring. One-lead ECG can help improve arrhythmia diagnosis but discriminating P-waves may be challenging and insufficient for correct diagnosis of sinus rhythm [18]. Accordingly, these devices are generally used to record long recordings for up to 14 days or a recording of merely a few seconds (e.g., Apple Watch Series 7). The authors in [19] fused the machinal pattern of the heart using seismocardiography (SCG) signal with ECG data. The fusion method is based on the Naïve Bayes probabilistic model to extract the PQRST annotations from the raw ECG signal. Then, they calculate the duration between different subsets of the PQRST data vector, which can indicate the presence of abnormalities in the ECG signals. The ECG processing is performed on a host desktop, transmitting the data through a synchronous data logger. This process is decoupled from the data acquisition process as it runs using a data logger, which is not designed for real-time operations.

Several research works have proposed transmitting ECG signals in real-time to a backend server for processing. The authors in [20] proposed integrating wireless communications (e.g., BLE and WIFI) for transmitting the acquired data. The actual design and hardware prototype did not include wireless communication modules. Therefore, the device is connected to high-performance data acquisition hardware. Then, the data are transferred to the host computer for storage, processing, analysis, and visualization. Klum et al. [21] present a fusion algorithm between multiple sensors, including a single-lead ECG sensor to measure lead I or II and a stethoscope to obtain an acoustic insight of the heartbeats. Similarly, [22] introduced a remote single-lead ECG powered by a coin battery. ECG signals are transferred to a mobile device in real-time and then to a backend system for further analysis. Although the device logs offline ECG data, it does not analyze the collected data for abnormal heart conditions to notify the patients or the healthcare provider to take necessary actions.

A different line of research develops porotypes that use fog computing to enable remote ECG monitoring. The authors in [23] introduced a single lead ECG monitoring system to provide a telemedical solution for rural areas with the help of fog computing. The system utilizes the ESP-32 module as the main microcontroller (MCU) for processing the collected ECG data, then sends the collected data in one-minute intervals, not on a real-time basis. This limitation is due to the bandwidth restriction of the LoRa communication link (i.e., 0.3–50 Kb/s). Moreover, the authors did not consider discontinuity scenarios where no wireless connectivity is in range since the device does not offer offline data logging. Accordingly, this solution is not optimized for long-term ECG monitoring. Similarly, Ahsanuzzaman et al. [24] developed a single-lead ECG acquisition hardware using Arduino Uno for acquiring ECG signals and sending them to a Raspberry PI (RPI) for further analysis and classification. However, the system has an overhead in the design since the Analog to Digital Converter (ADC) is interfaced with an Arduino Uno via Serial Peripheral Interface (SPI). The Arduino Uno device is connected serially to the RPI for signal analysis. This design overlap could have been avoided by directly connecting the ADC module to the RPI. Moreover, the RPI has an embedded BLE module that can communicate with a mobile phone instead of using the extra HC-05 Bluetooth hardware.

The integration of the four modes of operations: (1) serial cable transmission, (2) offline data logging to a local flash drive, (3) transmission to a mobile phone via Bluetooth, and (4) transmission to a desktop via Bluetooth provides better integration and operability. The authors in [25] proposed a single-lead ECG signal acquisition prototype using one of the four given modes of operation. Using the first two operation modes, the system can detect the QRS feature points from a complete cardiac cycle. However, this function is only available through the first operation mode. Besides, the device does not provide an automated handover between operation modes. For example, when a discontinuity event occurs, the device does not switch to the offline operation mode.

Furthermore, the authors in [26] presented a single lead ECG with multiple wireless protocols for data transmission and communication. The solution presented by the authors in [26] provided a vertical IoT system for remote ECG monitoring, but the system lacks multiple functionalities essential in the remote ECG testing eco-system. The ECG device works only on demand when connected to a gateway device with internet connectivity, which violates the founding definition of remote ECG monitoring. Moreover, the device does not provide means of storage on the ECG device to maintain a continuous log of ECG data in case of intermittent disconnectivity with the gateway device. On the other hand, the proposed IoT solution in [26] lacks core components in enabling continuous ECG data streaming since the system utilizes legacy client-server architectures. Most importantly, the authors [26] claim that the ECG device operates using ZigBee, Bluetooth, or WiFi wireless technologies; however, the underlying protocols used to transmit data are not discussed. Similarly, the authors in [27] presented a single lead ECG device for remote monitoring using dry Fabric electrodes, which provides additional flexibility to patients who wear the device for long-term monitoring. However, the device operates as a passive ECG signal collection device with no active components for signal detection or analysis.

2.2.2 Two ECG Leads or More

Multiple Leads ECG provides a better view of the heart condition for long-term cardiac diagnoses. Further, additional ECG leads are required to diagnose sinus rhythm or arrhythmias. In [28], the authors developed a 3-lead ECG wearable device that transfers the acquired ECG data to a mobile device through BLE communications, then relays the data to a backend system through mobile connectivity. Their hardware is based on the RPI development board as the main MCU, and a breakout Printed Circuit Board (PCB) was designed with an analog frontend for acquiring the ECG signals. The system featured a backend component to store collected ECG data. Despite the RPI's compact size, multiple communication protocols, and computation capabilities, it is not optimized for wearable technologies since it is not designed for low-power computing. Yuan et al. [29] developed a 3-lead ECG acquisition device to acquire ECG signals during pregnancy. The signals are transmitted to an Android application using BLE. Then, the application performs real-time filtering to remove baseline drift and high-frequency interference. The sample entropy algorithm is then used for fetal ECG extraction performed offline. To that extent, Wang et al. [30] designed an intelligent vest using non-adhesive electrodes to collect 3-lead ECG signals transmitting data to a mobile device to display the received signals. The system acts as a Holter monitor, where no data analytics or classification is done on the collected ECG data. This component is needed in critical heart conditions.

Physicians recommend increasing the number of active ECG leads helps understand heart conditions. To that extent, Abtahi et al. [31] developed a 5-lead ECG acquisition hardware using the ADAS1000 chip as an Analog Frontend (AFE) on a custom-designed PCB interfaced to an RPI module using the serial SPI. The system featured a backend component to store the collected ECG data. However, the hardware is not optimized for wearable technologies since the RPI is not designed for low-power computing. Moreover, the device does not offer offline ECG data recording in disconnected scenarios when Bluetooth connectivity is unavailable. Lastly, the authors [32] introduced an early detection module to notify patients about abnormal heart conditions. However, the module relies on getting the classification results (e.g., Mild and Severe over a certain period) using a script running on a host desktop.

2.2.3 Standard 12-Lead ECG

Standard 12-Lead ECG provides a complete and comprehensive analysis of the heart and enables a better thorough diagnosis that cannot be obtained otherwise. The authors in [33] presented a 12-lead module to capture ECG vital signals and transmit them to a backend server for analysis via a mobile phone connected to the ECG device through Bluetooth. However, the Bluetooth link used the classical Bluetooth protocols to communicate with a mobile device. It did not optimize the BLE protocol stack for sending the data, which increases the packet size and overhead of the transferred data rather than using BLE standard services and characteristics. Besides, the device depends on the results coming from the backend regarding the heart condition of the patient because it only works if the device is connected to a mobile phone with internet connectivity. Moreover, the Bluetooth module is not optimized for low power consumption, reducing the device's operation time and compromising the patient experience. Accordingly, their system mainly focuses on the data acquisition part of the ECG test. In contrast, the authors in [34] use a similar monitoring architecture but add active electrodes attached to the patient's body without adhesive materials. However, the device only supports offline ECG monitoring. The data is stored on a flash drive and then transferred to a PC for further processing and analysis of the collected ECG data.

While in [35], the authors developed a prototype for EEG/ECG data acquisition, the hardware utilizes the TI ADS1298/9 Analog Frontend chip for digitizing analog EEG/ECG signals. The authors incorporated multiple components to make it suitable for remote EEG/ECG monitoring like Bluetooth, WIFI, and local storage (i.e., SD) modules. However, the platform for operating these modules to perform ECG/EEG monitoring is not mentioned. Besides, the authors were interested in validating the integrability and operability of the hardware modules in terms of the available bandwidth between the different components to perform a successful EEG/ECG test. Lastly, the authors in [15], [17] conclude that these systems still lack completeness and comprehensiveness with the massive diversity of work proposed in ECG monitoring systems. Despite the significant overlapping in the literature about the number of recorded ECG leads, accuracy, usability, and mobility, these systems cannot provide real-time data streaming and heart condition diagnosis.

2.2.4 Commercial ECG Devices

ECG devices must be approved by respective health authorities in countries to be used on patients or spread commercially. Many devices are currently available in the market that perform remote ECG monitoring. However, most of them have not been approved for medical use by respective authorities in targeted countries, like the Medical Device Bureau in Canada or the Food and Drug Administration (FDA) in America. AliveCor is one of the commercial devices that the FDA has cleared as a clinical-grade consumer product. The device offers real-time diagnoses of heart conditions. It only detects recurrent atrial fibrillation and provides real-time readings for a short time. Also, patients have to stand still and limit their movement during the data acquisition period. The SEEQ^{TM} sensor by Medtronic Inc. [36], the ZIO^{XT} Patch by iRhythm Technologies Inc. [37], and the wearable biosensor by Philips [38] all share the same features of continuous ECG data collection from 7 up to 14 days of recorded data. However, the collected data is only for one ECG lead that can detect limited heart diseases. Additionally, the SEEQ^{TM} and the ZIO^{XT} are “single-use” devices.

While Savvy ECG [39] addresses the shortcomings of the previously mentioned devices like reusability (rechargeable battery) and real-time streaming to a mobile device, the solution does not offer diagnostic information about the heart conditions of the patient. Moreover, the device cannot determine various heart diseases since it only supports single-lead ECG acquisition. To that extent, a similar device was recently proposed for continuous monitoring called ECG247 [40] with a single ECG lead. However, the device addresses the limitation of Savvy ECG [39] and introduces a post-processing arrhythmia analyzer to detect abnormalities in heart activity.

Similarly, BioTelemetry [41] introduced the MCOT wireless ECG that provides two channels of ECG data acquisition with up to 30 days of data storage. The device allows for wireless data transmission on demand. Besides, the device cannot be used multiple times. Once the battery is exhausted, it cannot be used again. Moreover, the battery life lasts one year, and once the expiration date has expired, the device is no longer usable and needs to be replaced with a new device.

In comparison, a recent device introduced by QTMedical [42] incorporated the full scale of a standard 12-lead ECG testing as a wearable RPM device. However, the device does not provide continuous real-time ECG monitoring. It only collects ECG data when a patient activates the recording from a mobile device using a customized mobile application. Even with the manual activation, the ECG data collection period is limited to ten seconds, and then the patient must activate the data collection again whenever required. On the other hand, [43] delivers continuous wireless ECG monitoring. However, the device captures only one single ECG channel. Moreover, the device is not designed for long-term cardiac diagnoses since the device's working time lasts for up to a day, then the device would need to be recharged.

2.3 ECG Remote Patient Monitoring and Diagnoses Platforms

Remote health monitoring and related technologies are being standardized and integrated into the healthcare domain from only being used in wellness and lifestyle activities. ECG is one of the major health applications widely investigated by the research community and invested by the industry. RPM and real-time data analytics have significantly contributed to enhancing ECG monitoring and enabling healthcare providers to gain 24/7 access to their patients remotely, especially for patients with coronary ECG diseases. Therefore, the definition of remote patient monitoring spans more than just providing data collection and visualization infrastructures. It necessitates data streaming processing, analytics and notification systems. Accordingly, these systems combined provide a comprehensive ECG PRM for real-time data collection and diagnosis. However, the significant expansion in ECG remote monitoring systems has created overlapping and disconnected pieces in the provided solutions. Besides, many of the proposed systems fail to consider interoperability and integration with existing and state-of-the-art technologies and frameworks. Therefore, it is becoming more challenging for researchers to compare or utilize available methods for practical applications.

2.3.1 Real-time Streaming and Event Processing

Due to the massive amount of health-related data produced daily, there is an urgent need for reliable real-time data streaming and processing engines to empower remote patient monitoring in healthcare applications. The authors in [44] introduced a general platform to ingest real-time data using Kafka from different sources. The platform extracts useful information about patients using Artificial Intelligent (AI) to predict chronic diseases. However, this work only introduces a theoretical design of the system without empirical results. Sandha et al. [45] use Apache Kafka and Apache Spark to build a system that helps measure stress and predict heart attack risks. Their system uses a dataset from Physionet that contains ECG and blood pressure signals and a simple Naive Bayes algorithm to predict the heart failure risk. However, according to the paper, the practical evaluation of the use cases was out of their study scope. Pomprapa et al. [46] discuss a methodology to detect obstructive sleep apnea, a severe sleep disorder, from multiple sensors. They build a multi-sensor prototype that collects data, passes it to a hybrid deep learning model, and processes it in real-time using Kafka. Even though the study shows promising results, most of the data has been collected from simulated scenarios built by the authors. Moreover, the system is built to accommodate that specific case with no discussions on scalability or interoperability with various components in RPM like a notification system in case of abnormal activities.

Healthcare and medical applications ideally share a real-time data processing layer that enables the system to classify signals, recognize diseases, organize them into records on the cloud or on-premises, and notify healthcare providers during emergencies. Furthermore, the platforms differ in the feedback they receive from patients and health care providers and how they communicate. Also, the technical challenges in these platforms cannot be overlooked from the amount of data, privacy, and ability to store data while offline. The authors in [47] used fog-based computing to reduce latency, make decisions quickly, improve energy consumption, and reduce network congestion during computation and analytics. However, the platform did not provide an outline for data processing and analytics, which constitute essential components in healthcare applications. On the other hand, cloud computing constitutes an ideal choice for data processing, as it provides processing capabilities higher than the edge or fog nodes. These requirements match the demands of healthcare applications for real-time data processing and analytics. The authors in [48] present a portable real-time ECG device using Raspberry pi to receive the ECG signal. They created a Wi-Fi access point between Raspberry Pi and a heart rate sensor to collect ECG data and transmit it to a mobile application. Then the mobile application uploads the collected ECG data to the cloud to perform deep learning using neural networks. The system proposed in [48] comes with certain limitations in the deep learning approach due to limitations inherited from the used dataset for training the deep neural network. Besides, the system uses non-standard methods for converting and transmitting the ECG signals to the cloud.

Meanwhile, Alfian et al. [49] propose monitoring patients with diabetes using a web-based application that utilizes the Kafka streaming engine. The authors implemented two classification algorithms: one is based on MLP to classify diabetes patients and an LSTM network to predict blood glucose levels. However, the reported accuracy of the MLP algorithm is 77.083%, which is less significant to be used in monitoring critical health conditions. Moreover, the data acquisition (e.g., sampling rate) and prepossessing methods were not discussed. The system also lacks proper notification channels like sending SMS, dispatching an ambulance, or contacting the healthcare provider in case of abnormal data detection. Lastly, a recent review [14] compares more than 280 references for related work in the healthcare monitoring domain, showing that given the amount of work presented in the ECG monitoring systems, these systems are like vertical silos of various IoT applications. These systems lack interoperability on a horizontal scale to cover the developing needs of the healthcare domain [50].

To that extent, the authors in [51] proposed a horizontally scalable system to cover the previously mentioned shortcomings in the literature. The system is based on the Industrial Internet Reference Architecture (IIRA) three-tier model [52]. The first tier represents the edge tier representing sensor nodes for data collection, and the second tier is the platform tire, which represents the service an IoT platform provides, like data transformation and basic data analysis. The third tier is the enterprise tier serving as a user interface for data visualization and displaying medical charts. The system shows advanced levels of interoperability, leveraging the services of the ThingsBoard IoT platform [53]. It is worth noting that the system in [51] utilizes Apache Spark for big data analysis but missed to explain how to scale Apache Spark to accommodate various data analytics functions and algorithms. Therefore, the system in [51] lacks an essential component needed by the healthcare providers for a comprehensive RPM system with a real-time data analytics and predictions.

2.3.2 ECG Heart Condition Diagnoses and Analytics Benchmark

The diagnosis and classification of heart diseases are common areas that researchers seek to improve their efficiency. Deep learning classifications constitute significant importance in diagnosing vital heart conditions, therefore contributing to saving patient lives. To that extent, the authors in [54], [55] performed a set of experiments utilizing the PTB-XL [56] dataset. The experiments were carried out in a three-level hierarchical structure following the dataset presentation. The first layer included two classes: normal and abnormal heart conditions, where all heart diseases in the dataset are aggregated into the abnormal class. The second level included five classes (Normal ECG, Conduction Disturbance (CD), Myocardial Infarction (MI), Hypertrophy (HYP) and ST/T change (STTC)). Each class is composed of multiple subclasses of heart diseases (i.e., arrhythmias), which are: (CD has 8 arrhythmias, MI has 4 arrhythmias, HYP has 5 arrhythmias, and STTC has 5 arrhythmias). Therefore, the overall classes available for classification are 23 classes.

Consequently, in [55], the authors used the PTB-XL dataset and extracted the R-peak and the QRST component of the ECG signals for the classification process. They compared the Few-Shot Learning neural network and the SoftMax-based network in classifying ECG signals into normal and abnormal classes. The Few-Shot Learning neural network achieved better detection accuracy of 93.2%. Similarly, the authors applied the same classification techniques to classify five and 20 classes of ECG arrhythmias. The detection accuracy was 80.2% in the case of five classes and 24.9% in the case of 20 classes. However, the SoftMax neural network performed better than the Few-Shot Learning neural network when five or 20 classes of ECG arrhythmias were included. They said the reason for the decrease in accuracy is due to the large number the categories and imbalanced data distribution in the data sets (e.g., data with a label "normal" represents about one-third of the data while some diseases have few samples in the dataset) to train the models used to perform the classification. The same experiments were applied in [54], utilizing the same dataset but using a different classification technique, the FSL neural network. The results of the detection accuracy were 90.8% for the two-class classification (i.e., normal and abnormal heart conditions), 79.1% when classifying between the five superclasses and 70.1% for 20 classes of arrhythmias.

On the other hand, the authors in [57] established a benchmark and a comparison of different 12-lead ECG signal classification algorithms. They categorized heart diseases into nine categories and compared seven classification algorithms (i.e., inception1d, xresnet1d101, resnet1d_wang, fcn_wang, stm_bidir, LSTM, and Wavelet+NN). They achieved a maximum accuracy of 92.5% in classifying heart diseases into one of the defined categories using the inception1d algorithm. The algorithm dealt with the ECG signals in their original format (i.e., one-dimensional array of ECG data measured in millivolts), which constructs the waveform of an ECG signal. While in [58], the authors transformed ECG raw data from its original format into 2D images to identify heart diseases. This step imitates the process used by cardiologists in interpreting and diagnosing ECG tests. Therefore, the authors compared the 2D image representation of ECG signals for classification to one-dimensional ECG signals representation. The MIT-BIH dataset was used to compare, where five arrhythmias were only included in the classification in both cases. The 2D image representation of ECG signals architecture improved the detection accuracy of the classification process. The one-dimensional approach achieved 94.5% detection accuracy, while the 2D images approach achieved higher accuracy of 98%. Moreover, the classification methods only considered a single ECG lead instead of a standard 12-lead ECG which reduces the capability of detecting a broader range of heart diseases (e.g., arrhythmias).

Accordingly, existing solutions are designed to fit a limited number of heart diseases without a balanced dataset with an equal number of samples in each disease. ECG classifications have evaluated ECG data using Lead II signals only as an input to the classification functions. As a result, these functions are insufficient to classify a wide range of heart diseases. Likewise, some heart diseases (e.g., Anterior and Posterior heart diseases [59]) cannot be identified using Lead II only, which justifies decisions taken by cardiologists in some cases asking patients to perform standard 12-lead tests to diagnose the patient’s heart conditions fully. ECG signals classification would benefit by including more ECG leads as input to the classification functions. Therefore, more heart diseases can be found from a single test depending on the number of ECG leads; thus, the confidence in classification results increases.

2.4 Summary

Several solutions exist proposed by the research community or the healthcare industry, changing the status quo of traditional ECG Holter monitoring systems and offering portable and wearable ECG monitoring. Patients can use these solutions anywhere, not just in hospitals or healthcare facilities. The literature review presented in this chapter concludes that with the massive diversity of ECG monitoring platforms, these platforms still lack comprehensiveness and essential features for effective remote ECG testing. Moreover, the overlapping is significant between existing solutions in the literature concerning the number of recorded ECG leads, accuracy, usability, and mobility. Furthermore, the healthcare industry is witnessing an explosion of companies proposing solutions, creating a fragmented landscape within the e-health architectures. As a result, this fragmentation is one of the significant obstacles hindering the widespread integration of health systems. To that extent, the literature is divided into three major topics concerning remote ECG monitoring: wearable e-health devices development, automated diagnosis on collected e-health data using AI and ML techniques and real-time streaming coupled with event processing engines. Many of the proposed systems lack the ability to provide real-time data streaming and heart condition diagnosis. None of them also offer an informative correlation between current ECG data being collected and previously collected ECG data and/or clinical charts.

We conclude that pieces of a complete comprehensive ECG testing exist or have been proposed in the literature. Inspired by such findings, in the following chapters, a device for ECG data collection and monitoring named XBeats is presented, which utilizes a suitable combination of solutions and methodologies for effectively enabling remote ECG monitoring. Then a design of a comprehensive RPM framework is presented to complete the ECG monitoring lifecycle and provide an end-to-end real-time and remote ECG data monitoring and analytics framework for healthcare providers.

Chapter 3. XBeats ECG Patch Hardware Design

This chapter introduces XBeats, a patent-pending wearable device for remote ECG data collection and monitoring. The XBeats ECG patch addresses the limitations observed in the literature review and provides a standard 12-lead ECG device that is fully autonomous. The device utilizes BLE communication technologies for connectivity and data transmission, making the device fully connected through paired internet-enabled gateways. Moreover, XBeats features various operation modes to cover possible scenarios and give healthcare providers complete control over the ECG data acquisition process. The ECG data acquisition process on XBeats features a fully configurable ECG leads selection starting from a single lead to a standard 12-lead ECG test. Furthermore, the XBeats hardware ships with an embedded storage module that ensures uninterrupted ECG data logging and serves as a backup to the wireless data transmission operations. Therefore, healthcare providers are guaranteed to maintain full access to the ECG recordings of their patients at any given time.

3.1 Introduction

The provision of remote patient monitoring is imperative in changing traditional healthcare services and abilities to monitor and manage patients remotely [11]. Patients with chronic diseases, especially cardiovascular (CVD require ongoing medical attention, which limits their activities and everyday routines. Chronic diseases last for long periods that can be in years [7]. The XBeats design leverages the means of edge computing in performing a machine learning classification technique on the collected ECG data in real-time. The classification technique serves as a binary classifier categorizing ECG signals into two classes: regular and irregular ECG signals. The firmware on the XBeats ECG patch uses the classification results from the edge device and updates the currently active mode of operation on the patch. This way, XBeats offers autonomously configurable modes of operation. For example, in the case of an irregular ECG signal classification result, the ECG patch dynamically changes the operation mode to the continuous modes of operation for a standard 12-lead acquisition.

Accordingly, we propose a comprehensive remote patient monitoring framework for ECG testing, as shown in Figure 3.1. The objective of the framework is to perform ECG testing and monitoring remotely without hindering the daily activities of the patients using an easy-to-use wearable ECG patch. The design of XBeats addresses the limitations of existing solutions by providing a BLE-connected, real-time, and comprehensive ECG monitoring system. The architecture comprises a wearable and unobtrusive real-time ECG device which can be configured with three operation modes: (a) continuous mode, (b) triggered mode, and (c) offline mode.

Furthermore, guarantee unbounded real-time connectivity to the healthcare provider for monitoring the heart conditions and vitals of the patient while maintaining prompt responses should irregular heart conditions develop. The framework comprises two stages: the first stage constitutes the hardware components responsible for the ECG data collection and delivery to the backend system of the RPM framework. The second stage constitutes the software and system design of an RPM framework to provide the intended services of providing continuous real-time ECG monitoring and analytics.

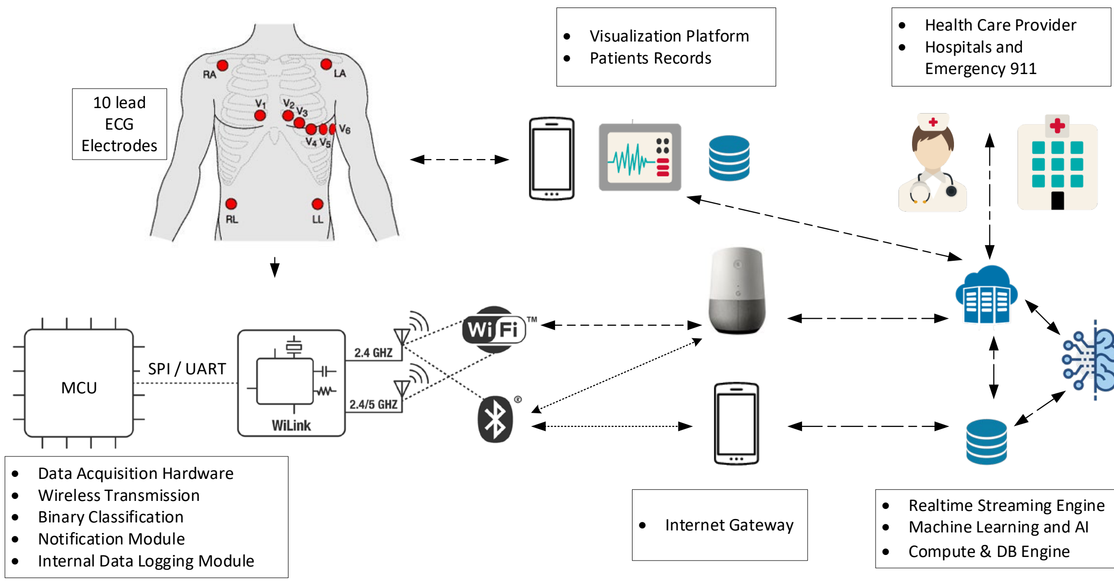


Figure 3.1. High-level architecture of the XBeats ECG RPM framework.

3.2 XBeats Hardware Components

The XBeats hardware represents the data acquisition device (i.e., ECG patch) for acquiring ECG signals. XBeats constitutes a wearable, unobtrusive and connected real-time ECG device. Besides, the main focus of the hardware design is to deliver the intended functionalities of a standard 12-lead ECG acquisition process. It considers the mechanics of wearable devices (e.g., the fitting of the ECG leads on various body masses) as topics of its own. The device comes with configurable modes of operation that are adjusted upon directions from the healthcare provider or developing heart conditions. The delivery of RPM services requires a reliable device to converge vital medical charts and information directly from the patient. Data acquisition services require the presence of wearable wireless sensors (i.e., Apple Watch). However, wearable wireless medical devices entail a strict set of requirements to be considered for medical applications and critical patient conditions. This set of requirements is translated to a group of high-level hardware components that includes:

1. High-resolution data acquisition modules or sensors to guarantee reliable and representative medical data collection [60]–[62].
2. Reliable communication modules (e.g., BLE, Zigbee) for data transmission between the ECG patch and the receiving device (e.g., smartphone) [60]–[62].
3. Low-power processing units
4. High-speed and low latency storage units (e.g., SD Cards, Flash Storage) [60]–[62].

Due to the nature of ECG testing, it acquires vital signals or information without interruptions during the acquisition period [63], [64]. Lastly, RPM devices enable the acquisition of medical and vital data for ECG and heart monitoring while transmitting data continuously to the healthcare provider.

The design of the XBeats implements a modular architecture focusing on the specifications and functionalities of the hardware components needed for the operation of the device. Therefore, the proposed architecture is not limited to a specific hardware component or vendor, where the proposed hardware components can replace other components with similar capabilities. Accordingly, the hardware components will be discussed based on functionality with candidate components for prototyping to show the feasibility of the proposed architecture. The main building blocks of the XBeats ECG patch architecture are shown in Figure 3.2, with the following functionalities:

1. Collect the standard 12-lead ECG test data in real-time using a lightweight and unobtrusive body sensor not to interrupt the patient's everyday lifestyle.
2. Provide flexible modes of operation to accommodate various heart conditions and enable healthcare providers to control the ECG patch remotely.
3. Transmit ECG data to internet-enabled gateways (e.g., mobile devices) using low-energy communication protocols and standards.
4. Log collected ECG data using timestamps on local storage attached to the ECG patch serving as a backup service routine.

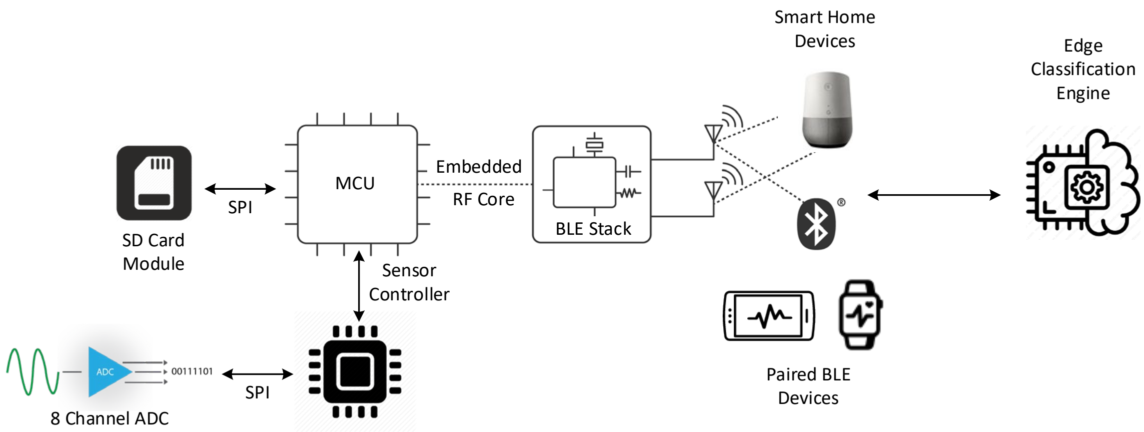


Figure 3.2. High-level design of the XBeats ECG patch hardware components.

3.2.1 ECG Data Acquisition

The data acquisition component is the first step of the data pipeline in XBeats and is designed to match the operation modes of a typical ECG test performed in hospitals [7], [8], [13]. Accordingly, the data collection pipeline in XBeats starts with the data acquisition phase. Data is captured from the patient's chest using special electronic pads (electrodes). A full ECG test necessitates ten electrodes to be attached to the patient's chest. The ten electrodes obtain 12 views of the heart, referred to as the 12-lead ECG test. ECG signal data are received as a sequence of analog voltage data. The received signals are usually accompanied by noise and distortions due to motion artifacts and lead misplacement. Therefore, we process the ECG signals in three sequential steps to complete the data filtering process. The first step uses a bandpass filter to filter unwanted frequencies [65]; this removes powerline noise, muscle noise, and electrode contact noise while acquiring real-time ECG signals. The second step starts by buffering the ECG signals based on the minimum and maximum heart rates recorded between 30 and 240 beats per minute (bpm) [65].

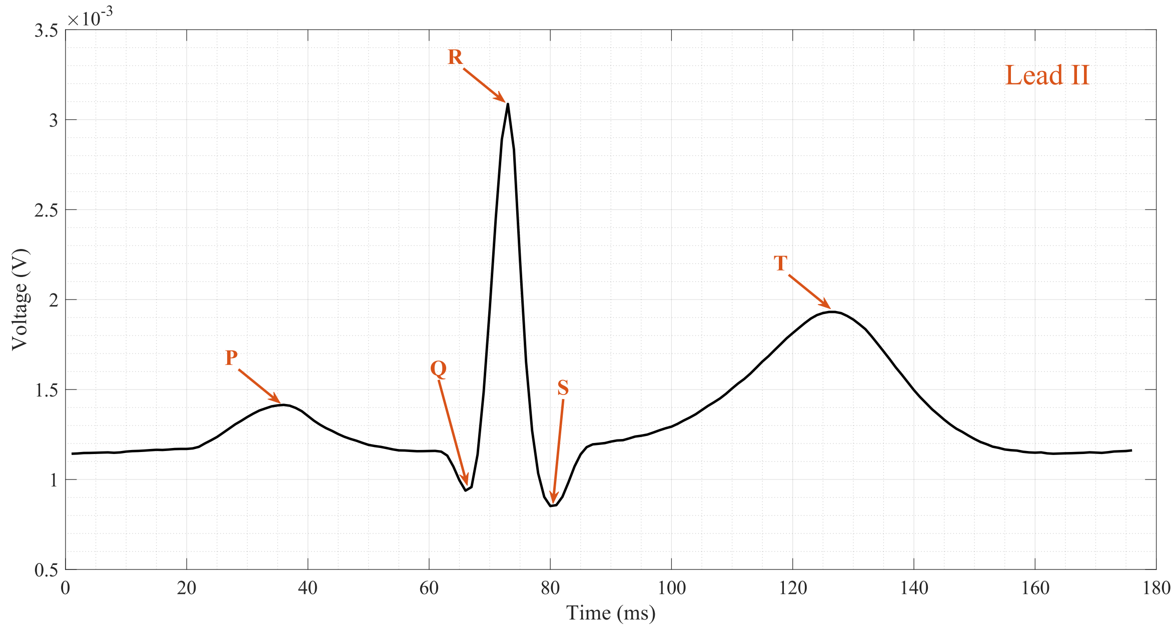


Figure 3.3. Single heartbeat signal with PQRST feature points.

A sliding window of two-second intervals is used to capture at least one complete heartbeat signal. At 30 bpm, one complete cardiac cycle is guaranteed to be captured in a two-second interval and four complete cardiac cycles at a heart rate of 240 bpm. The third step identifies ECG signal features; there are two features: intra-beat and inter-beat features. The intra-beat features resemble prominent points in each cardiac cycle; these points are P, Q, R, S, and T, as illustrated in Figure 3.3. Accordingly, identifying the PQRST feature points in one signal implies the occurrence of a complete cardiac cycle. The PQRST feature points of each cardiac cycle are selected based on the values from Table 3.1. On the other hand, inter-beat features are derived from intra-beat features such as RR-interval, which is the interval between two consecutive R peaks.

Table 3-1. Standard ECG PQRST features points intervals for normal heart conditions.

Normal Heat Rate	60 - 100 bpm
PR interval	0.12 - 0.20 s
QRS interval	≤ 0.12 s
QT interval	< half RR interval (males < 0.40 s; females < 0.44 s)
P wave amplitude (in lead II)	≤ 3 mV (mm)
P wave terminal negative deflection (in lead V1)	≤ 1 mV (mm)
Q wave	< 0.04 s (1 mm) and < 1/3 of R wave amplitude in the same lead

To that extent, the data acquisition module in Figure 3.4 is set to acquire the following leads purely in the analog format: two of the limb leads and the six chest leads (i.e., Leads I, II, V1, V2, V3, V4, V5 and V6). Besides, in a typical implementation of the 12-lead ECG, the augmented leads (i.e., aVR, aVL, and aVF) and Lead III are computed digitally [66]. The data acquisition module converts analog ECG signals (i.e., voltage) to a digitized format. However, the received signals are usually accompanied by noise and distortions due to motion artifacts and lead misplacement.

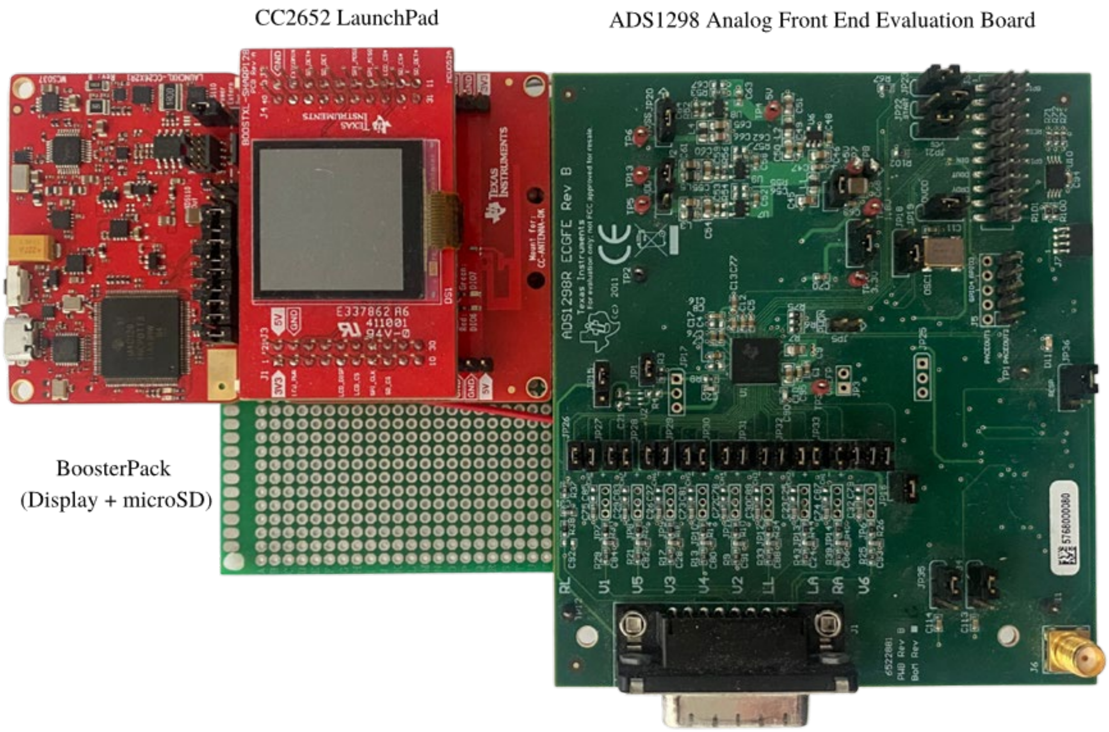


Figure 3.4. XBeats hardware development kits.

XBeats features three modes of operation to provide an ultimate user experience and high operability under various operation scenarios. Accordingly, the data acquisition process is governed by the proposed three modes of operations explicitly developed for the XBeats ECG patch. Moreover, the backend of the XBeats RPM framework is designed to support these modes of operation. The introduced modes of operations are as follows:

1. The **continuous mode** provides an unbounded real-time, high-resolution data stream of the 12-lead ECG data transmitted directly to the backend of the RPM system. Physicians sometimes require this mode of operation if abnormal heart conditions are detected or the patient's medical condition

requires 24/7 monitoring. However, this mode has a significant power consumption profile that affects the battery lifetime of the device due to the continuous transmission of the collected 12-lead ECG raw data from the analog-to-digital module wirelessly to the backend system via a communication gateway.

2. The **offline mode** records the 12-lead ECG data on an attached multimedia storage unit when no paired BLE device is nearby to connect to the ECG patch. This mode is enabled for the entire data acquisition period until a paired BLE device connects to the patch and synchronizes the data transfer to the backend system.
3. The **triggered mode** is optimized for power saving. The device sends keep-alive signals in normal heart conditions and only transmits ECG signals when a potential heart abnormality is detected. XBeats chooses from three data acquisition settings where the number of leads is configurable. The default setting for this operation mode is three ECG leads (e.g., Lead I, II and V1), which can be changed dynamically in real-time. The patient and health care provider can reconfigure the number of enabled ECG leads through a paired BLE-enabled device or the backend system. Accordingly, the backend system is designed to support these modes of operation.

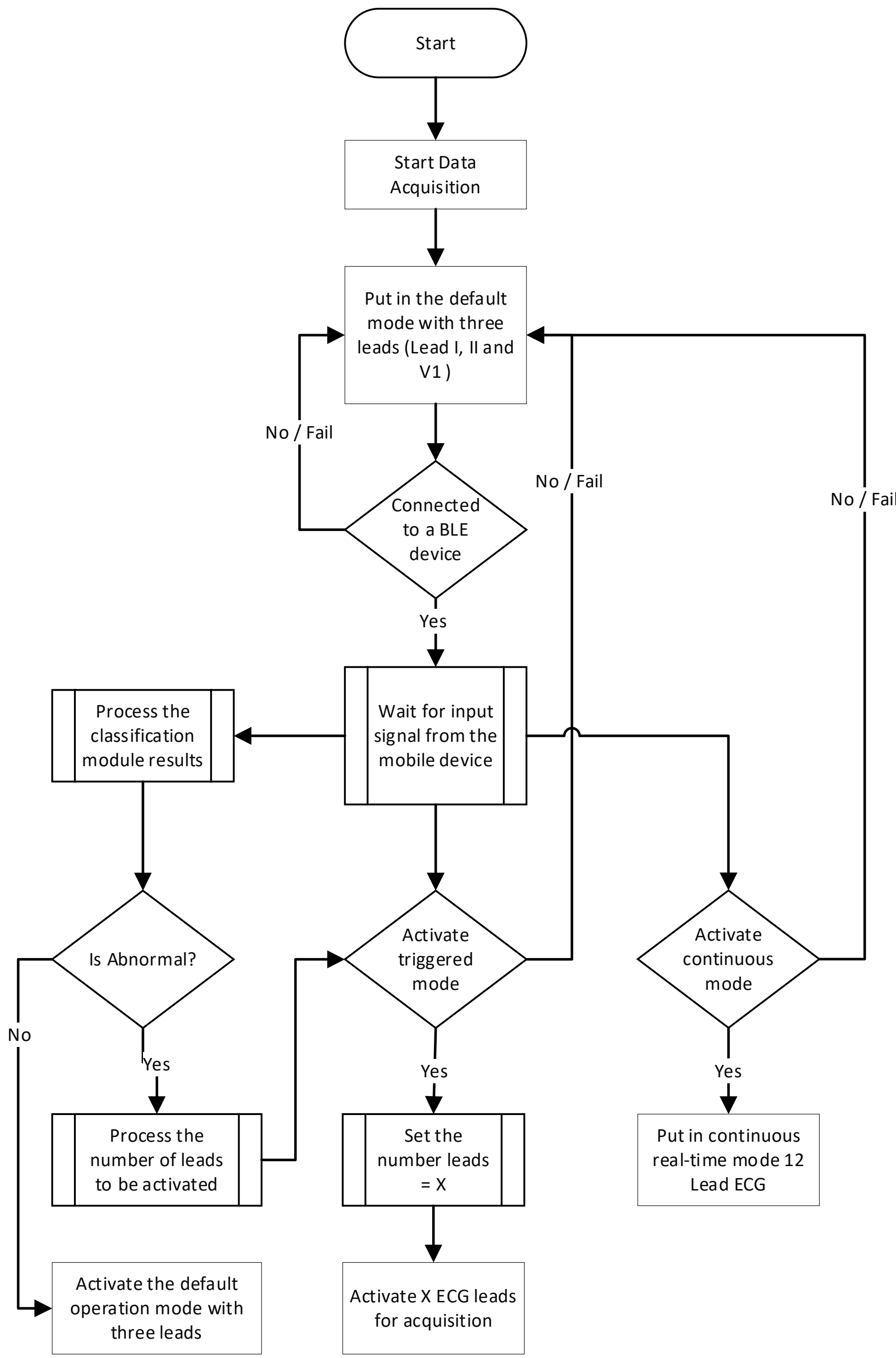


Figure 3.5. Data Acquisition flowchart of the XBeats modes of operation.

Figure 3.5 shows the data acquisition flow diagram integrated into the firmware of the proposed ECG patch. The device runs the default operation mode until a new instruction is received from a connected paired BLE device. Once the connection is established, the instructions are received through the connected BLE device. Then, XBeats starts a service routine that listens for new instructions to update the current operation mode. The service routine expects one of three instructions to be received at a time. One of the expected instructions is to activate the continuous mode to perform a 12-lead ECG test. This instruction comes directly from the user through the mobile application or as instructed by the healthcare provider through the backend system. The second instruction involves handling the results of the classification module. The classification module analyzes the ECG data continuously on an edge device upon receiving it from the ECG patch. The classification results are transmitted to the mobile application, and then the mobile application sends instructions to the ECG patch with the classification results. The ECG patch will continue in the triggered operation mode with three leads (i.e., Leads I, II and aVF) when the heart activity is normal. On the other hand, in case of abnormal heart conditions, the classification module will return the number of leads to be activated and start the triggered mode.

3.2.2 ECG Data Transmission Using BLE Technologies

Data transmission constitutes the second step of the data pipeline in the XBeats ECG patch while continuously exchanging data with the backend.

Furthermore, patients and their healthcare providers can display related vital information about the heart from the recorded ECG signals. The ECG patch can also connect to smart home devices to establish a direct communication channel with the backend notification modules when necessary to send alerts to physicians or healthcare providers if abnormal heart conditions are detected. The system can be configured to call the emergency in extreme cases when the patient is at imminent risk.

The data transmission hardware features a multi-standard wireless microcontroller (MCU) with a radio frequency (RF) core that fully supports BLE version 5.2. The MCU enables seamless data transmission in real-time to the backend system. The. BLE version 5.2 upgrades the data transfer rate two times the rate of the previous Bluetooth version 4.2, from 1-Mbps to 2-Mbps. Moreover, the BLE version 5.2 protocol stack can be integrated into the Realtime Operating System (RTOS) [67] as an additional software layer. Accordingly, in RTOS, all BLE-related functions run in a separate task. The communication between BLE-based sensors and smart devices is defined using Generic Access Profile (GAP) and Generic Attribute (GATT) protocols [68].

GATT defines how two BLE devices exchange data, while GAP constitutes a standard way for BLE devices to communicate with the outside world. A complete GATT transaction constitutes three high-level objects Profiles, Services and Characteristics. For instance, the Heart Rate Profile (HRP) is used in devices measuring heart rate [69]. The HRP profile has two mandatory services: (1) heart rate and (2) device information services. Supplementary services can also be added, for instance, Battery Service (BAS), to indicate the battery's current level. Accordingly, BLE provides a versatile set of full-stack solutions to meet the needs for low-power wireless connectivity. However, a Profile [68] has to be defined to exchange data between BLE-enabled peripherals. According to Bluetooth SIG [70], a profile with its related services can be selected from the Bluetooth SIG's predefined profiles and services if they match the application's specifications or develop custom services to match the application's specifications [71]. There is no standard BLE profile to deliver the standard 12-lead ECG services [70]. Therefore, we create a customized BLE profile to enable our proposed to provide the designed services in our framework (i.e., operation modes and ECG data streaming).

To that extent, the proposed ECG BLE "Profile" describes the number of GATT "Services" and GATT "Characteristics" that should be used to achieve the addressed functionalities of the proposed ECG patch. The smallest addressable unit of data used by the ECG BLE profile is called an "Attribute". Thus, a set of defined "Attributes" constitutes a BLE "Characteristic". One "Characteristic" consists of an at least value attribute and a declaration attribute which describes whether the value attribute can be read or written. Consequently, a collection of "Characteristics" constitutes a BLE service, while one or more services define a BLE "Profile" [71]. The BLE profile describes how services deliver the application's intended functionalities. Therefore, we design the communication and data transmission protocols XBeats ECG services using BLE standards and APIs.

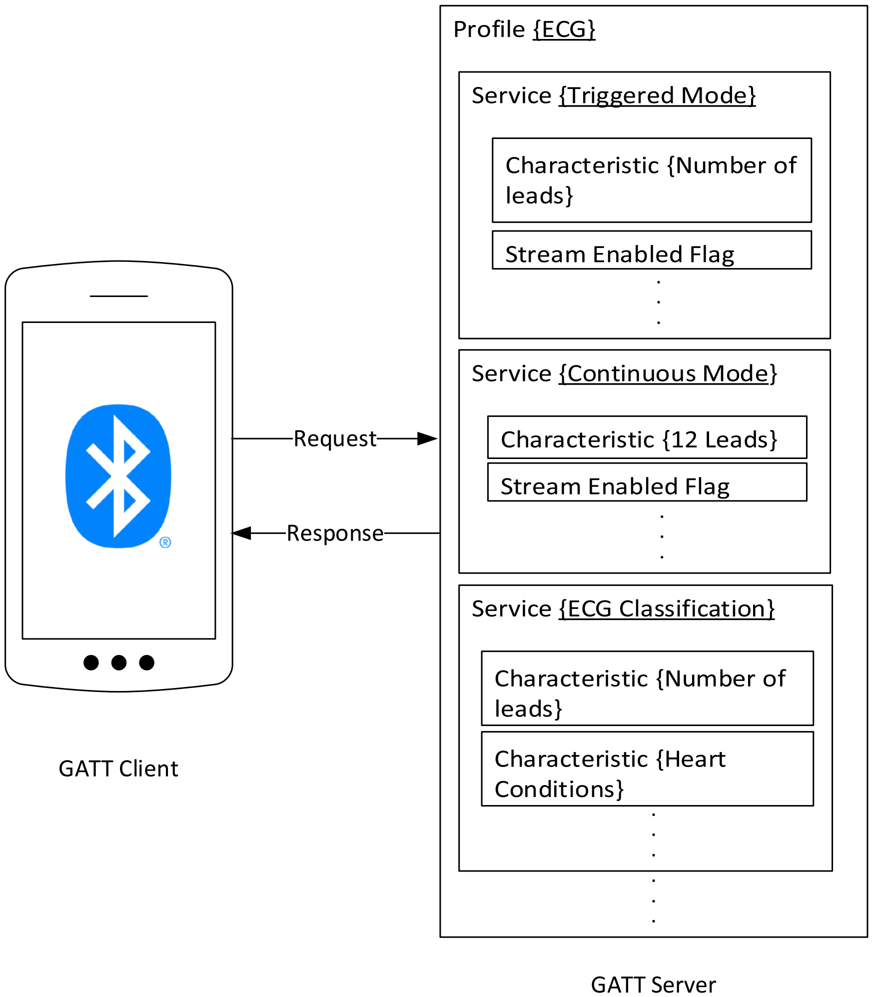


Figure 3.6. The Proposed BLE Profile for XBeats communications and data transmission protocols.

As shown in Figure 3.6, the designed "ECG Profile" has three services that correspond to operation modes defined on the ECG patch and the edge classification service. The first service in the ECG profile is the triggered mode containing three characteristics. The first characteristic holds information about the active number of leads. In contrast, the second Characteristic is a flag that refers to the status of real-time streaming, whether it is currently enabled or disabled. The second "Service" is for the continuous operation mode where ECG leads are fixed to a 12 leads configuration, and the stream flag characteristic is enabled by default. Lastly, we implement the "ECG Classification" service for our novel adaptive modes of operation. The "ECG Classification" service contains two characteristics. The first characteristic holds the current heart condition results from the classification implemented on the edge device. The other "Characteristic" carries the recommended ECG leads to be activated based on the results from the classification function on the edge device.

3.2.3 ECG Data Logging Operations

Data logging and local storage are essential components in a critical safety system where backtracking is vital for life-saving decisions. The proposed system integrates a high-performance and reliable Multimedia Card (MMC) for continuous data logging. The data logging process is encapsulated in an independent task running simultaneously with the data acquisition task. A data retention service routine is also integrated within the system for extended periods, as cardiologists recommend. Consequently, the device monitors all incoming ECG signals for critical heart condition detection. After this grace period, newly received data is not recorded until previously saved data is downloaded. This is another system design decision with an alternative option that could be overwriting older data with recent ECG signals. Cardiologists recommend our design choice [72].

3.2.4 ECG Edge Data Classification

XBeats integrates a binary classifier implemented on the edge to perform preliminary analysis on the collected raw ECG signal for anomaly detection. The classifier can be either integrated into a mobile application or an edge device with enough resources. Moreover, the objective of the edge classification module is to optimize data transmission to the backend system compared to the continuous mode of operation. The objective of the classification protocol is to classify incoming ECG signals in real-time into two classes: "normal" and "abnormal". The classification module operates as a binary classifier, performing preliminary analysis on the collected raw ECG signal to determine whether it is normal or abnormal. The correlation between two consecutive PQRST vectors facilitates interpreting the patient's heart condition and discovering abnormalities if they exist. Therefore, when an irregular heartbeat is detected, the ECG patch shall notify the patient or the healthcare provider through the nearest paired mobile device or internet-connected gateway.

However, data preprocessing is applied to the collected ECG data before we apply the classification task. The preprocessing phase extracts the PQRST features from the ECG waveform using the Pan Tompkins algorithm [73]. The PQRST feature points constitute one single heartbeat. The correlation between two consecutive PQRST vectors facilitates interpreting the patient’s heart condition and discovering abnormalities if they exist. The preprocessing step includes detecting multiple R points (R-R interval), which helps measure the heart rate. Also, the PR interval, QRS duration, or QT interval contributes to revealing significant heart condition information.

Accordingly, data cleaning and preprocessing are applied to the collected ECG data before the classification task. The preprocessing phase extracts the PQRST features from the ECG waveform. The PQRST feature points constitute one single heartbeat. The correlation between two consecutive PQRST vectors facilitates interpreting the patient's heart condition and discovering abnormalities if they exist. The preprocessing step includes detecting multiple R points (R-R interval), which helps measure the heart rate. Also, the PR interval, QRS duration, or QT interval contributes to revealing significant heart condition information. That, in return, adds more confidence to the binary classification detection task. The binary classification module (e.g., integrated into the mobile application) classifies the signals into just two categories; normal, which includes one type of signal, and abnormal, which represents all other types of signals.

The binary classification of ECG signals utilizes the online PTB-XL arrhythmia database [56] to train the model on larger populations. The implemented model uses lead II and resamples the ECG data to 300 Hz to match the minimum sampling rate of the signals extracted from our ECG patch. The binary classifier classifies the signals into just two categories; normal, which includes one type of heartbeats, and abnormal, which represents all other types of irregular heartbeats. Model training and verification are performed according to the proposed approach in [74], using 44 records for training and 22 for testing. The preprocessing phase includes an algorithm to extract 85 features from the ECG waveforms mentioned in [74]. Accordingly, we propose a feature selection technique called mutual information (MI) ranking criterion to select the ten most informative features to obtain high accuracy. This approach suits our system needs due to the power and time limitation in our system with high accuracy. Moreover, we only need to extract meaningful information about the heart condition, which adds more confidence to the binary classification task. The extracted information is translated into ten features: a collection of R–R-intervals, HBF (Hermite basis functions), and time-domain morphology features explained in [74]. We test several algorithms that yield better results, including Random Forest, Support Vector Machine (SVM), Decision Tree, k-nearest neighbours (Knn), Logistics Regression, and Extra Trees.

3.3 XBeats Power Consumption Analysis

The power consumption analysis of the XBeats ECG patch follows systematic power consumption profiling steps on the hardware components to study potential optimizations and extend the lifetime of the battery on the device. Each hardware component is analyzed individually to isolate the controlling parameters and provide a deep insight into the power consumption behaviour. The hardware components under investigation are the ECG analog to digital front end (AFE) sensor, the wireless communication module, the sensor controller, the MCU, and the local storage unit. The power investigation of the ECG patch spans the hardware components mentioned earlier, from which a suitable optimization approach is selected. The optimization approaches work in tandem with the firmware operating the ECG patch to provide the highest efficiency possible regarding power consumption while maintaining the expected operational functionalities of the device. The power consumption profiling order follows the direction of the data acquisition pipeline on XBeats, as illustrated in the following points:

1. The first step evaluates the power consumption profile of the data acquisition process performed by the analog to digital converter implemented on the ECG patch.
2. The second step evaluates the data transmission process on XBeats using BLE communication links. This process implicitly evaluates the overall performance of the main microcontroller since the communication core enabling BLE is an integrated component shipped inside the MCU.
3. The third step investigates various techniques to optimize the read/write operations performed on the local storage module.

The firmware on XBeats constitutes a significant role in optimizing the power consumption behaviour on the device. To that extent, the investigation process includes steps to optimize the operation of hardware components controlled by the firmware on XBeats. The power consumption profile of each component is investigated individually to isolate all related parameters concerning the power consumption.

3.3.1 XBeats Power Analysis for Data Acquisition

Two hardware components are utilized on XBeats to acquire ECG signals. The first component is the ADS1298 chip [75], a low-power AFE used for medical applications, ECG precisely. The second component is the sensor controller, which forwards the digitized ECG signals by the ADS1298 chip to the MCU. The ADS1298 chip comes with eight ADC channels responsible for collecting the analog ECG voltages from the patient body.

The following parameters contribute to measuring the power consumption on the ADS1298 and the sensor controller at a base voltage $V_{src} = 3V$ operational voltage:

1. The power consumed per each active channel P_{ch} Where each channel consumes 818 uW.
2. The ECG patch uses the internal clock of the ADS1298 chip; therefore, the usage of the internal oscillator power P_{clk} adds approximately 120 uW.
3. The power dissipation $\mathrm{P_{dis}}$ when the device is operating in the high-resolution mode at 500 samples per second (SPS) has a maximum power of 9.5 mW.
4. Power consumption of the Right Leg Driver and the Wilson Terminal, respectively named P_{RLD} and P_{WCT}.

5. Power consumption of the sensor controller in the active mode $P_{SC\,|\,Active}$ equals 90.3 uW at 2 MHz clock frequency.

The overall power consumption by the ADS1298 chip is calculated using the following equation:

$P_{ADS} = n_{ch} \times P_{ch} + P_{clk} + P_{dis} + P_{RLD} + P_{WCT}$, where $\mathrm{n_{ch}}$ is the number of active *channels* $\in [0:8]$

The firmware of XBeats implements three modes of operation, as explained in Section 3.2.1. The three modes of operation are: continuous, triggered and disconnected modes. The continuous mode assumes unbounded 12-lead ECG data collection during the period when the operation mode is enabled. Therefore, the total power consumption is calculated by substituting the n_{ch} by eight

$$P_{ADS\,|\,Cont.} = (8 \times P_{ch}) + P_{clk} + P_{dis} + P_{RLD} + P_{WCT}$$

And the total power consumed over time $T_{cont.}$

$$P_{ADS\,|\,Cont.} \times T_{cont.}$$

While in the triggered mode, the enabled channels $\mathrm{n_{ch\,|\,Trigg.}} \in \{1, 3\}$

$$P_{ADS\,|\,Trigg.} = (n_{ch\,|\,Trigg.} \times P_{ch}) + P_{clk} + P_{dis} + P_{RLD} + P_{WCT}$$

And the total power consumed over time $\mathrm{T_{Trigg.}}$

$$P_{ADS\,|\,Trigg.} \times T_{Trigg.}$$

In the case of disconnected mode, the device maintains the same operation mode (i.e., number of active channels) just before a disconnectivity occurs. Therefore, the total power consumed by the ADS1298 chip and the sensor controller during the data acquisition processes is calculated using the following equations:

$$P_{ADS} = P_{ADS\,|\,Cont.} + P_{ADS\,|\,Trigg.} + P_{ADS\,|\,disc.} + (T_{Total} \times \; P_{SC\,|\,Active})$$

The objective of the power consumption profiling of the ADS is to minimize the time $T_{cont.}$ in which the device operates in the continuous mode. Therefore, we override the original firmware operational flowchart to limit the continuous mode data acquisition to ten-second intervals and then revert to the triggered mode with three enabled channels.

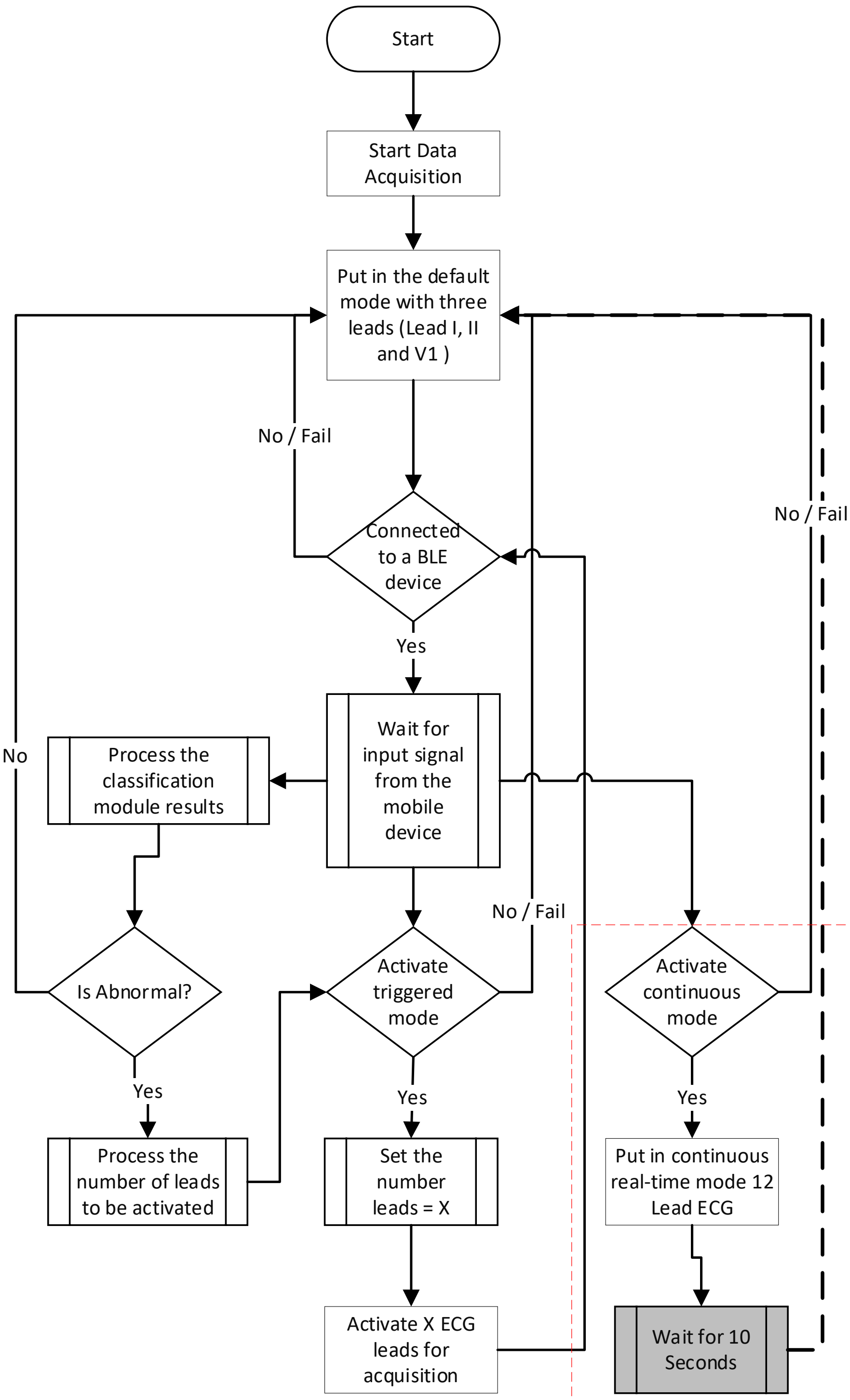


Figure 3.7. The modified data acquisition flowchart of the XBeats modes of operation.

Accordingly, reducing the operation time of some modes of operation (i.e., the continuous mode) contributes to reducing the current consumption. To that extent, Figure 3.7 shows the modified data acquisition and modes of operations on the ECG patch with a new subroutine that automatically switches to the triggered mode each time the continuous mode is activated after ten seconds.

3.3.2 XBeats Power Analysis for Data Transmission over BLE

The power consumption profile of BLE data transmission varies depending on the available uplink/downlink throughputs (i.e., 1Mbps and 2Mbps) and payloads (i.e., 27 bytes and 251 bytes). Besides, connection events, connection intervals, slave latency and the supervision time-out of a BLE session are the main controlling parameters which dictate the power consumption profile of the BLE communication module. A connection event is when a peripheral and central device sends and receives data from one another on a specific channel at a particular time. The connection event takes place periodically every time interval. The time interval is measured in units of 1.25 ms. The minimum time interval in a standard BLE connection is six units (7.5 ms) and a maximum of 3200 units (4 seconds). The power consumption is inversely proportional to the connection interval in a BLE connection. If the connection interval is reduced, the connected BLE devices attempt to exchange connection events data more frequently; thus, additional current is consumed during the process. However, increasing the connection interval period reduces the connection throughput and the time for sending data in either direction increases.

Furthermore, the slave latency parameter allows the peripheral device to skip several consecutive connection events. The connection event skipping gives the slave device a chance to stay idle or sleep for a time equal to the number of skipped connection events (i.e., slave latency) multiplied by the default connection interval period. The slave latency values range from 0 (i.e., slave latency is disabled) to 499. However, the slave latency parameter shall not exceed the following:

$$Supervision\ Timeout\ /\ 2$$

The Supervision Time-Out is calculated in units of 10 ms. The minimum supervision time-out is 100 ms (10 units), and the maximum value equals 32.0 seconds (3200 units). In the scenario where the slave latency is enabled, the connection interval period becomes a factor of the effective connection interval, which is calculated using the following equation:

$$EffectiveConnectionInterval = (ConnectionInterval) * (1 + [SlaveLatency])$$

Moreover, the time the BLE device spends communicating with the connected device is called the airtime T_{Air} . Therefore, the power consumed by the communication module P_{BLE} is calculated using the following equation:

$$P_{BLE} = T_{Air} \times I_{Tx} \times V_{src}$$

3.3.3 XBeats Data Logging & Storage Optimizations

Data logging is an essential component of the ECG patch serving as a requirement for critical safety systems and life-saving conditions. Therefore, the ECG patch integrates a high-performance and reliable Multimedia Card (MMC). The objective is to optimize the number of times the data logging process on the ECG patch invokes the write operation on the MMC. The writing operation has two controlling parameters: the written data type (e.g., styled string, hexadecimal, or raw binary data) and the sector size on the MMC (data is written in multiples of the sector size).

3.4 XBeats RPM Framework

Remote health monitoring and related technologies are being standardized and integrated into the healthcare domain, changing the status quo from only being used in wellness and lifestyle activities. ECG testing is one of the major health applications widely investigated by the research community and invested by the industry. RPM and real-time data analytics have significantly contributed to enhancing ECG monitoring and enabling healthcare providers to gain 24/7 access to their patients remotely, especially for patients with coronary diseases. This section presents a comprehensive RPM framework for real-time telehealth operations with scalable data monitoring, real-time analytics and decision making, fine-grained data access and robust notification mechanisms in emergencies and critical health conditions. We focus on the overall framework architecture, integration of enabling technologies on the system level and deployment options. While we provide a use case application for patients with chronic heart conditions as they require continuous ECG monitoring in real-time. We are releasing the framework as open-source software to the active research community.

3.4.1 The Promise of Remote Patient Monitoring

In the last decade, RPM in the healthcare sector has witnessed an increasing number of enabling technologies due to the proliferation of the Internet of Things services and standard [60], [76]. Moreover, the advancements in electronics to the finest granularity enabled the development and manufacturing of a diversified range of wearable medical devices serving the RPM ecosystem [77]. In contrast, the establishment and availability of high-speed internet and communication protocols worldwide facilitated the rapid spread and acquisition of RPM services in the healthcare sector. Similarly, the emergence of artificial intelligence (AI) and machine learning (ML) extend the collection of raw data from heterogeneous sources and provide quality insights and diagnostic information related to the collected data [60], [78]. Data classifications and predictions enabled by AI and ML contribute to saving patients with critical health conditions by continuously analyzing their vitals and providing physicians with insights [12].

Most importantly, real-time data analytics contributes to automating emergency responses (i.e., call 911 or dispatch an ambulance) in critical conditions (i.e., heart attack). Lastly, data virtualization comes into place, representing a gateway to the users (i.e., healthcare providers or patients), giving them valuable insights over the patient health conditions and control (e.g., adjusting the operation mode of the intended acquisition device or tuning ML parameters) [79]. To that extent, a typical RPM implementation constitutes various services like online resources, tracking, communication, automated analysis, diagnoses, and notification systems. Figure 3.8 shows the core elements of the XBeats RPM framework divided into four layers data acquisition, streaming, analytics and visualization.

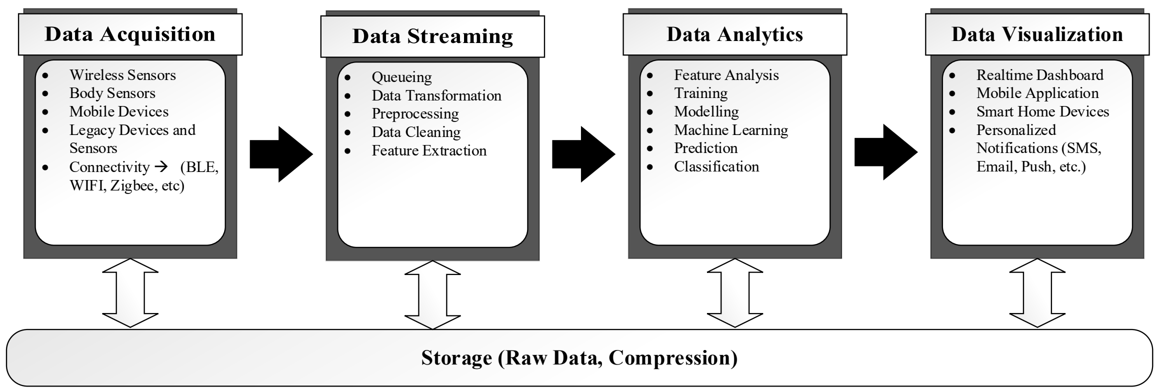


Figure 3.8. XBeats RPM framework life cycle.

3.4.2 Data Acquisition as Enabling Technologies in RPM Systems

RPM and wearable medical devices entail a strict set of requirements to be considered for medical applications and critical patient conditions. ECG data acquisition devices enable the delivery of RPM services by providing reliable data acquisition services to converge vital medical charts and information directly from the patient. Wearable ECG devices run on batteries, and due to the nature of an ECG test, an ECG device continuously acquires ECG signals without interruptions during the acquisition period [10], [63], [64]. Contrary to other wearable RPM devices, the data collection occurs intermittently over long periods. In the past few years, there has been an expansion in the introduced wearable ECG and heart rate monitoring devices that serve various purposes. However, there are still gaps in the devices that support and provide a standard 12-lead ECG testing remotely. Therefore, this work introduces the XBeats ECG patch in Chapter 3 to address the limitations and overlapping components introduced in the literature and the healthcare industry. XBeats ECG patch provides standard 12-lead remote ECG monitoring for long-term heart monitoring using wearable and low-power technologies.

3.4.3 Leveraging Data Streaming and Event Processing in RPM Systems

Data streaming and event processing are the enabling technologies for RPM systems (i.e., on-demand e-health services) due to their efficiency and reliability in ingesting unbounded streams of data (e.g., ambulatory and intensive care units). The streaming engine transforms medical data collected in the data acquisition stage using standard IoT communication protocols (i.e., Message Queuing Telemetry Transport [80] (MQTT)) for data collection, which is suitable to serve and accommodate the needs of the healthcare systems. Streaming engines enable many-to-many communication channels between the data acquisition services and services provided by the backend of an RPM system (i.e., data analytics and storage). The many-to-many communication architecture makes the data available to the backend services in real-time, regardless of the number of the enabled services [81]. A typical scenario is when a standard ECG test is performed remotely, where an ECG patch is attached to the patient's body, transmitting the collected ECG data to the streaming engine in real-time using the appropriate communication protocol (i.e., MQTT). This scenario can be extended to a larger global scale where more than one patient utilizes remote ECG testing services.

On the other hand, once the data is transmitted to the streaming engine, the streaming engine makes the data available to the intended ECG services at the backend component of the RPM system. The services include heart rate calculation, ECG signal features extraction, ECG signals classification and visualization services. The services above would run independently in isolated environments by ingesting the data streamed to the dedicated streaming channels for ECG data. The separation between services using the many-to-many architecture implies micro-services architecture design [82]. Contrary to the client-server architectures [82], [83], where many-to-one communications are used, many devices or sensors (i.e., clients) transmit data to the service provider (i.e., server). Moreover, client-server architectures are centralized; for example, the whole network will be disrupted if the main server fails. Therefore, client-server architectures lack robustness in regard to failure optimizations. Accordingly, modern service providers utilize object-oriented architectures and micro-services [84] due to increased maintainability, scalability, and fault tolerance techniques.

3.4.4 Real-time Data Analytics in RPM Systems

The data acquisition and streaming layers are considered passive data collection layers due to the absence of intelligence or decision-making. In contrast, the data analytics layer permits healthcare providers to focus on diagnosing, educating, and treating patients, theoretically improving the productivity and efficiency of the care provided. The data analytics layer in an RPM encompasses the life cycle of machine learning or deep learning processes starting from training, modelling, classification and prediction. A use case of the data analytics stage is performing analytics on ECG data to diagnose heart conditions from various candidate heart diseases detected during the classification process. Due to parallel classifications, the analytics stage can report more than one disease from a single input ECG signal. The capabilities of the streaming engine enable the introduced concept of parallel classifications by providing unbounded streaming channels to the installed classification functions. The installed classification functions are design decisions as they are trained and modelled based on the available data. This feature defines a new benchmark, in contrast to ECG analytics in the existing benchmarks [49, 60, 61], where only a single class of heart diseases is detected during the classification process.

Furthermore, the new feature can provide informative reports (e.g., ECG reports) with multiple candidate diseases (e.g., heart diseases) that facilitate the decision-making process by physicians. For example, two or more heart diseases detected from the same ECG test add more confidence to the diagnostic results of the heart condition of the patient. Moreover, the design decision can also include deploying multiple classifications and prediction models depending on availability.

3.4.5 Data Visualization and Notifications in RPM Systems

Data visualization and notifications simplify the sophistication of the previous layers of an RPM system. The virtualization layer includes various technologies for displaying medical information collected and processed in previous stages to the end-user. The healthcare provider gets full access to medical charts and vital information relevant to the patients in real-time through a user-friendly interface which can be a web interface through the Internet or mobile devices. A practical use case would be the applications of remote ECG monitoring, where a patient who wears an ECG patch can collect ECG signals in real-time using the XBeats ECG patch. Then, the visualization layer in an RPM system displays the ECG signals in real-time using dedicated graphical user interfaces (GUI). This also entitles healthcare providers and doctors to visualize the patient's current health conditions in real-time and retrieve historical data when needed.

On the other hand, the RPM framework provides application programming interfaces (APIs) for smartphones to display ECG signals and vital information about the patient. Then utilize notification features embedded in smart home devices to issue alerts to the patient when abnormal health conditions are detected (e.g., heart attacks). Similarly, the visualization layer allows broadcast notifications through paired smart home devices in case of emergencies and quickly dispatching emergency responses.

3.5 XBeats RPM Framework and Software Specifications

This section explains the design specifications of the XBeats RPM framework for unbounded ECG data acquisition, streaming and real-time analytics. The software specifications of a typical RPM framework constitute five layers, as explained in Section 3.4.1. The five layers are applied to the XBeats RPM framework, as shown in Figure 3.9 and discussed in the following sections.

The data acquisition layer complements the operation of the XBeats ECG patch as it provides unbounded data streaming channels to all connected XBeats devices. Then, the streaming engine ingests the incoming ECG data from all connected devices in real-time. Similarly, the streaming engine creates outbound communication channels for the data analytics, storage and visualization layers in the framework. The data streaming engine receives the incoming ECG data and performs deep analytics using machine learning techniques while building correlations with previous health charts concerning each patient. To that extent, the data storage layer logs the ECG collected ECG signals in raw format for further diagnoses by healthcare providers when needed. In addition, the storage layer saves all related patient vital and health information (i.e., analytics results, previous health conditions and diagnoses). Lastly, the visualization layer is a crucial component in the proposed framework as it acts as the first line of interaction with the framework. Healthcare providers and physicians utilize the visualization layer to display ECG signals in real-time, analytics results, heart conditions and relevant health information. Similarly, patients and healthcare providers receive notifications when irregular heart activities are detected. Accordingly, each component is explained in detail in the following subsections.

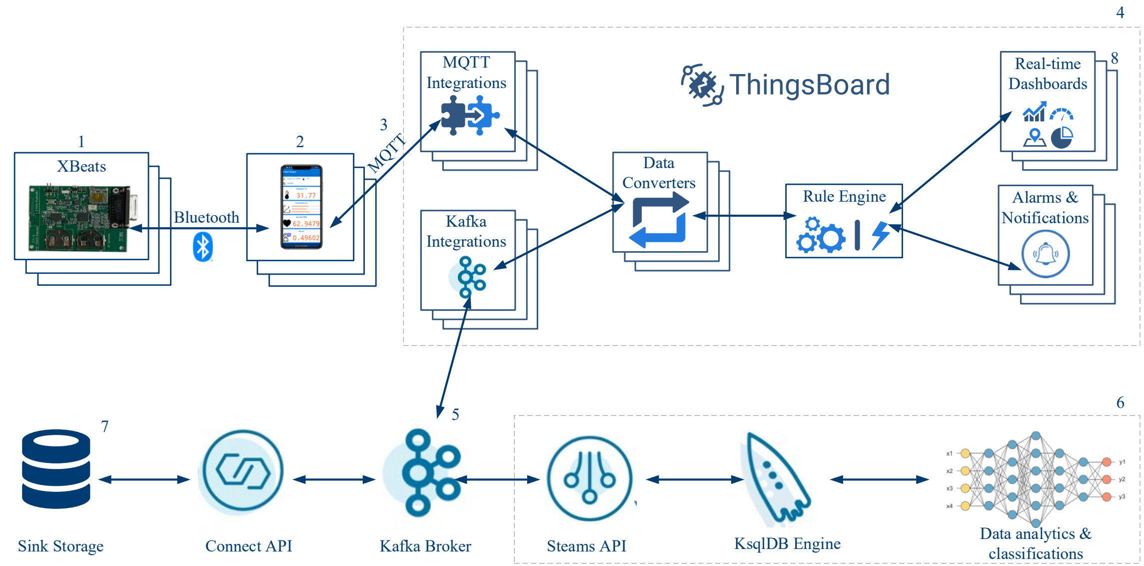


Figure 3.9. A high-level architecture design of the XBeats RPM framework.

3.5.1 XBeats Real-time Data Streaming

The XBeats RPM framework design enables real-time data streaming and decision-making for healthcare providers. The framework incorporates the latest data streaming and pipelining technologies while utilizing reliable and lightweight protocols for communication between all functional components. The backend system utilizes a scalable, fault-tolerant, and fast streaming engine to accommodate the high volume of streamed ECG data and vital information. The backend system gathers all the data acquired by the ECG patch, stores it in a high-performance database, and trains machine learning algorithms to perform real-time data diagnosis and predictions. The connection between the XBeats ECG patch and the backend system in the RPM framework is enabled through the integration of the latest Internet of Things (IoT) communication protocols (i.e., BLE and MQTT [80]). Data streaming from the ECG patch is carried out by Apache Kafka [16], a high-performance open-source real-time streaming engine. Kafka supports unbounded data streams with a latency (i.e., less than 10 milliseconds) and allows the integration of event processing frameworks like KsqlDB [85] to carry out advanced classification processes. Moreover, the proposed framework includes a modular frontend user interface for displaying the real-time ECG stream, preliminary diagnosis, records access and management, and robust notification service that interfaces with smart home devices.

3.5.2 XBeats Real-time Data Analytics

The real-time ECG data analytics layer is an integrated part of the XBeats RPM framework. It applies event processing techniques to the received ECG data and provides accurate diagnoses of the current heart conditions. While the streaming engine processes the ECG data, the data analytics component investigates each ECG lead individually (up to 12 leads) to increase confidence in the classification results. The framework builds a confidence level in the classification process by leveraging the capabilities of the ECG patch in collecting

ECG data using multiple ECG leads up to the standard 12-lead configurations. The confidence levels constitute the fusion of diagnostic results from multiple classification algorithms applied to the same data while trying various combinations of ECG leads. Then the results are sent to the healthcare providers as a report for each patient. Therefore, the data analytics component can classify multiple heart diseases if existent from the same ECG test, which imitates the ECG diagnosis process performed by cardiologists.

The XBeats framework comprises eight integrated system-level components, including the XBeats ECG patch, message queueing systems, real-time streaming engine, event processing engine, data storage, IoT platform, data analytics, and notification systems shown in Figure 3.9. The following section explains the integration process of the components while highlighting the objective of each component in delivering a comprehensive RPM framework for ECG data monitoring and analytics.

3.6 System Integration Steps of the XBeats RPM Framework

The first step involves developing a custom embedded BLE library on the ECG patch to utilize the BLE communication module on the ECG patch. The BLE library provides access to the real-time digitized ECG data acquired by the XBeats ECG patch to the nearest paired mobile device [86]. The library controls the operation modes (i.e., continuous, triggered, and offline) enabled on XBeats, which involves the number of enabled ECG leads activated during the acquisition period.

Moreover, the library sets a feedback communication between the activated operation mode and the results of the analytics modules. The analytics results include the classification of the current ECG signals (e.g., regular or irregular heartbeats) accompanied by the number of ECG leads to be activated in the case of the triggered mode of operation. Besides providing an interface to control the modes of operation on the ECG patch, the library utilizes BLE GATT services with two characteristics: the configuration characteristics (i.e., operation modes controller, number of activated leads) and the data stream characteristic. The data stream characteristic is used to transmit the collected ECG data during the data acquisition process on the ECG patch to the nearest paired mobile device.

The second step includes a custom mobile application implemented using the Android operating system that works as an internet gateway for the ECG patch and a user interface for the end-user to display vital information and ECG signals. The ECG waveforms (i.e., signals) are displayed in real-time on the mobile application as received from the ECG patch. Consequently, once the mobile application receives and verifies the data, it publishes the data directly to the backend of the RPM framework in real-time. The mobile application comprises three main screens (i.e., activities). The first activity handles connectivity and pairing to the nearest ECG patch for first-time connections. Then the second activity comes in place, enabling the user to select the intended mode of operation (i.e., continuous, triggered, and offline) and the number of activated ECG leads. The third activity displays the ECG signals in real-time along with the results of the performed analytics of the ECG data by the data analytics step of the RPM framework. Similarly, the mobile application process notification received from the backend concerning the current heart conditions of the patient and provides a gateway to broadcast notifications using smart home devices. On the other hand, a background service starts independent of the selected activity; this service connects (i.e., subscribes) to an MQTT topic available through the RPM framework. The MQTT topic is where the analytics results and notifications generated by the backend system are forwarded to the intended (i.e., subscribed) end-user.

The third step requires the presence of an MQTT message broker (e.g., HiveMQ [87]) for handling communication the gateway for the ECG patch (i.e., mobile device or smart home device) and backend system reliably and securely using an IoT standard protocol (i.e., MQTT). The MQTT protocol provides bi-directional communications between the mobile device and the backend system using the publish/subscribe architecture. The publish/subscribe architecture uses the term "Topic" in defining data pipelines for exchanging messages between the backend system and mobile devices. As mentioned earlier, the mobile application developed for the ECG patch uses the subscribe services provided by the MQTT server to receive messages and notifications from the RPM framework. While the MQTT server provides communication channels (i.e., MQTT topics) for publishing mainly ECG signals and related patient information.

The fourth step builds an API bridge between core framework features, connecting three layers in the XBeats RPM framework: the data between the acquisition, the streaming, and the visualization layers. The framework utilizes the ThingsBoard [53] platform in enabling the integration between the messaging broker (i.e., MQTT server) in the acquisition layer and the streaming engine layer (e.g., Apache Kafka). The choice of ThingsBoard is a design decision selection that can be replaced with another platform that provides the same functionalities. The MQTT API bridge allows users to connect to external MQTT brokers, subscribe to data streams from the external brokers and convert any payload from the connected ECG patch to ThingsBoard message format. This feature utilizes connections to external MQTT brokers or connectivity providers with an MQTT-based back-end, making the RPM framework independent of specific providers. Moreover, the MQTT API bridge enables connectivity to more than one broker or provider, giving the RPM framework the edge to expand horizontally. Similarly, the API allows the integration of external Kafka brokers for data streaming services and connectivity to the ECG data pipeline in the framework. Moreover, ThingsBoard provides a highly customizable rule engine that enables the proposed RPM framework to process complex events. The rule engine has various filters applied to inbound/outbound messages or events between the backend system and connected devices. The proposed RPM framework leverages the rule engine to filter messages between participating entities (i.e., ECG patch, mobile applications, and the backend system). Moreover, we use ThingsBoard to send notifications (e.g., push notifications, emails, messages) using various cloud service providers (e.g., AWS, GCP).

The fifth step utilizes the services of streaming engines (e.g., Kafka Confluent [88]), where medical data is processed and sent to the data analytics layer. The streaming engine uses microservices concepts as it allows the data analytics layer in the proposed framework to deploy and update new models independently without rebuilding the entire framework. The framework utilizes the microservices architecture instead of the traditional 3-tire application architecture referring to the client-server architecture [83]. The three-tier structure includes a user interface for visualization, business logic and data access control. However, this structure is outdated since it was initially intended for application development before the era of cloud and public computing. Therefore, it is challenging to use the three-tier architecture, as each component becomes large and complex to manage over time.

In contrast, the microservices architecture designs complex applications as a collection of services that are fully decoupled from the application. This collection of services and applications can be implemented in different programming languages and frameworks. Moreover, they communicate using different protocols where each microservice is only responsible for a specific purpose or task. This way, microservices abstract away implementation details and only expose the application through application programming interfaces APIs. Accordingly, the framework utilizes Apache Kafka for implementing the streaming engine. Kafka provides data pipelines for streaming and event processing and allows the integration of distributed processing frameworks like the ksqlDB [85]. The Kafka ecosystem services are delivered through five core APIs [16], as shown in Figure 3.10: Producer, Consumer, Streams, Connect, and Admin. These APIs assume microservices architecture, providing three main functionalities: publish/subscribe operations, process streams in real-time, and store received records using fault-tolerant and scalable methodologies. Moreover, Kafka uses the term "Topic" to define streaming channels between all participating entities.

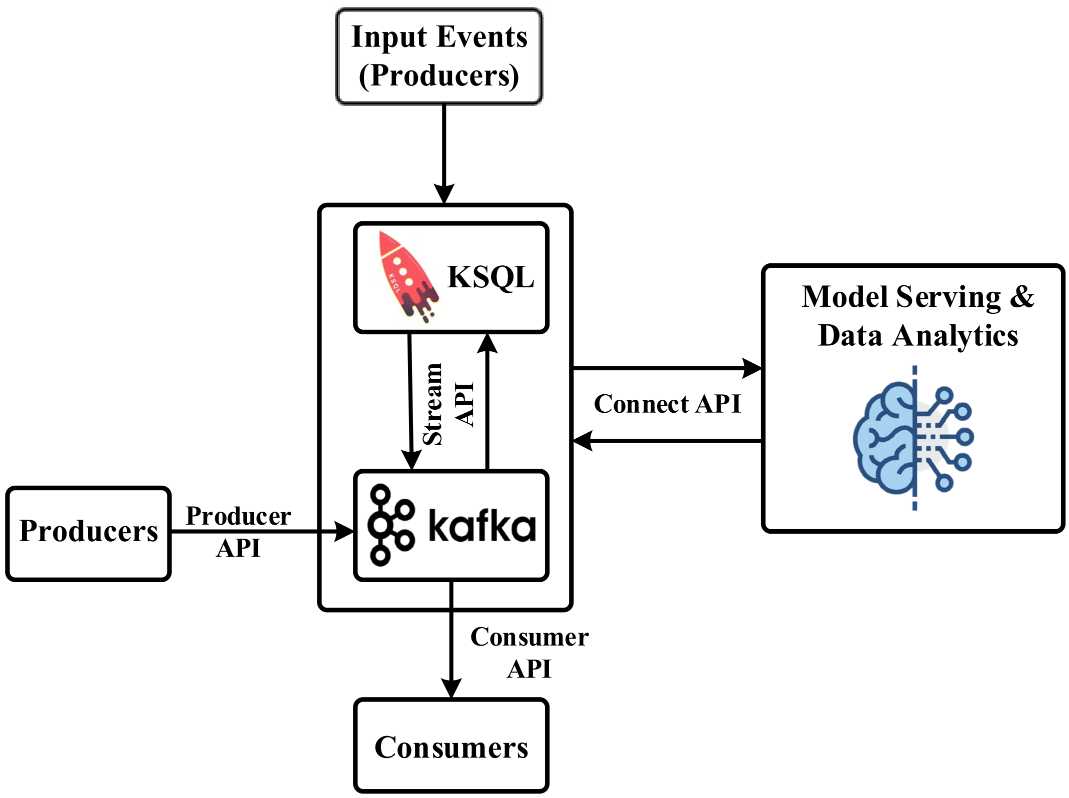


Figure 3.10. Kafka APIs Integration.

The Kafka APIs deliver three main functionalities, publish and manage subscriptions to streaming engines, process streams in real-time, and store received records using fault-tolerant and scalable methodologies. The Producer API allows publishing data streams to the Kafka engine to the designated topics. Similarly, patients using the XBeats ECG patch and healthcare providers utilize the Produce API to publish medical information and historical results as much as needed. The Consumer API allows consuming messages from the designated Kafka topics to end-users by integrating different telemetry and communication protocols (e.g., MQTT, CoAP). The Streams API gives access to the streaming processor in Kafka, consuming ECG data streams and producing an output stream of the processed data to one or more output topics. Effectively the Streams API transforms the input streams into output streams essentially for data analytics engines like Keras and TensorFlow libraries to apply deep learning algorithms. The Connect API provides an agile interface that continually pulls data directly from a data source into Kafka or pushes from Kafka into a sink system or an application. Moreover, the Kafka engine integrates the ksqlDB [85] to build a robust streaming engine for Apache Kafka, which leverages the power of stream processing using just a few SQL statements and familiarity with building traditional applications on a relational database. The Admin API simplifies monitoring all parts contributing to building Kafka's ecosystem as it allows managing and inspecting topics, brokers, and other Kafka objects.

The sixth step implements the Streams API of the Kafka system into the proposed framework, giving access to a streaming processor that consumes inbound streams. Then effectively transforms the streams into useful data used by the advanced Machine Learning (ML) algorithms and Convolutional Neural Networks (CNN) applied in the sixth component. The selected streaming processor in the proposed framework is ksqlDB. ksqlDB processes the data and sends it to the backend analytics component using RESTful APIs. Therefore, the framework integrates the features of Kafka and ksqlDB to apply advanced ML algorithms and CNN techniques. Then performs deep learning and analytics on the body sensor data collected by the ECG patch and draws correlations between real-time data and previous charts to predict some events. In contrast, the streaming engine processes ECG data, and the data analytics component investigates each ECG lead individually (up to 12 leads) to increase confidence in the classification results. We build the confidence level in the classification process by leveraging the capabilities of XBeats. The confidence levels constitute the fusion of diagnostic results from each ECG lead individually sent to the healthcare providers in unified reports for each patient. Therefore, the data analytics component can classify multiple heart diseases if existent from the same ECG test, which imitates the ECG diagnosis process performed by cardiologists.

The seventh step implements the Connect API of the Kafka system into the proposed framework to build persistent connections with database management systems like MongoDB. It provides a data pipeline that works as a sink offloading data from Kafka topics to the database. Therefore, the Connect API, in this case, is used to establish a connection with a database server instance (e.g., MongoDB) for data exchange. Then, the raw data collected by the ECG patch received by the streaming channels (i.e., Kafka Topics) on Kafka is transferred to the connected database instance. Consequently, this feature enables the framework to retrieve previous data charts of the user either to display the data on a dashboard or to be used by the healthcare provider for further analysis.

The eighth step designs and implements the user-interface and user-experience stages, where the user or the healthcare provider interacts with the framework. A dashboard allows the healthcare provider to access all information related to the patients assigned to them and navigate through their records. Moreover, it notifies the healthcare provider if an abnormal activity by the patient has been detected. Abnormal health activities information comes directly from the analytics done on the collected ECG data by the data analytics and prediction layers. On the other hand, healthcare providers receive notifications regarding their patients' vital information and heart conditions—another real-time dashboard for patients to access their information. To that extent, notifications come into place as the framework utilizes cloud notification services using an email service provider, push notifications over MQTT using publish/subscribe to notification channels, and connectivity to smart home devices. Smart home devices provide a new edge in providing notification and alerting patients as these devices can broadcast alerts when a patient is enduring abnormal heart conditions. Moreover, a smart home device can start a conversation using AI with the patient and through customized dialogues. The dialogues would contain questions asking patient-specific questions that confirm certain conditions discovered by the data analytics layer at the backend layer of the framework. The framework automatically dispatches emergency responses if the real-time data analytics layer detects abnormal activities.

3.7 Summary

This chapter proposes XBeats, an ECG patch for real-time and remote monitoring. The XBeats framework utilizes a suitable combination of solutions and methodologies for effectively performing a standard 12-lead remote ECG testing reproducing the same operations performed at hospitals and healthcare providers. While outlining the XBeats hardware components, we emphasize how they can help with providing remote reliable ECG testing at various levels, including high-resolution data acquisition and reliable communication protocols. Through the use of BLE version 5.2, the XBeats ECG patch can stream ECG signals seamlessly to a nearby paired smartphone or smart home device. A custom BLE Profile is developed, named ECG BLE Profile; the custom BLE Profile encapsulates the functionalities embedded on the ECG patch in the form of services the user can subscribe to using their mobile device. Then we present the design and system integrations to build the XBeats RPM framework for remote patient monitoring and real-time data analytics. Accordingly, we discuss the specifications for a comprehensive RPM framework along with the design goals. Besides, we emphasize the objectives of our proposed framework and the flexibility of our design as our architecture is not restricted or dependent on any service provider where all components can be replaced with other components that provide similar functionalities.

Chapter 4. XBeats Hardware Prototype and Supporting RPM Framework Implementations

This chapter describes the software and hardware components used in building the XBeats ECG patch hardware prototype and the RPM framework to provide comprehensive real-time ECG data collection and analytics. The implementation steps are presented in the logical flow of operations and processes required to perform a standard 12-lead ECG test remotely using the proposed RPM framework. Therefore, the high-level order of operations of the proposed framework starts with the hardware prototype represented in the XBeats ECG patch. Then, the next level includes the operations at the backbone of the framework, where the core operations of real-time data streaming and analytics take place, as presented in Chapter 3. To that extent, the selection of software tools utilized in implementing the proposed framework are considered design decisions that can be replaced with other software or tools that provide the same functionalities.

4.1 XBeats Hardware Prototype

The printed circuit board (PCB) prototype of the XBeats ECG patch in Figure 4.1 constitutes four major components: A microcontroller with a wireless core for BLE communications, an analog frontend interface (AFE) for acquiring analog

ECG signals using a high-performance analog to digital converter and a micro-SD card module for data logging.

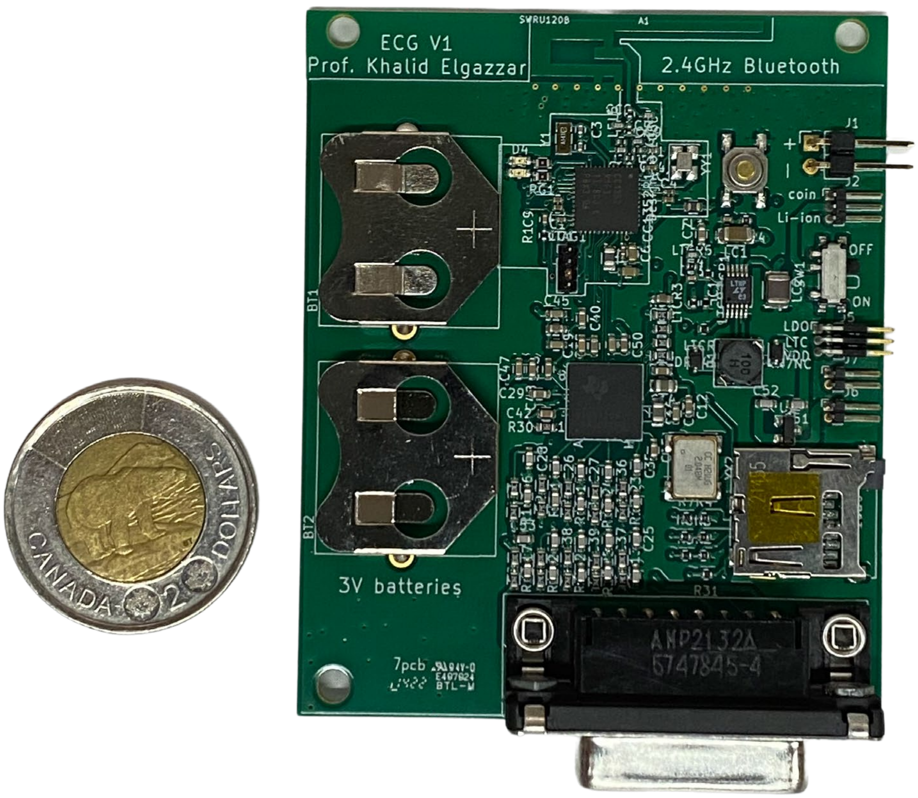


Figure 4.1. XBeats PCB Hardware Prototype.

4.1.1 XBeats Data Acquisition and Logging Implementation

The prototype of XBeats is powered by the SimpleLink 32-bit Arm MCU (CC2652R), which runs as the central processor. The CC2652R chip can handle real-time operations needed for critical safety systems. The MCU features an ultra-low power sensor controller powered by an Arm Cortex-M0 processor that offloads simple tasks like sensor readings of the main MCU. These features make the selected MCU well-suited for high data acquisition applications. Moreover, the ECG patch utilizes a specialized medical AFE to take electrical signals from the heart and digitize them. The ADS1298 chip is integrated into the patch to digitize the acquired ECG signals. It comes with a 24-bit status word that reflects the status of the electrodes (i.e., probably attached, whether connected or not). Moreover, it offers two sampling modes: high-resolution and low-power modes.

XBeats utilizes the proprietary Texas Instruments Real-Time Operating System (TI-RTOS) [67] to run the intended functionalities. TI-RTOS provides tools for managing and scheduling tasks using application layer functions. The TI-RTOS and ultra-low-power MCU combination supports longer battery life and makes applications more adaptable for real-time monitoring systems and wearable devices. The firmware on XBeats encapsulates the operations conducted by the MCU into tasks using the scheduling APIs of the onboard RTOS. Data acquisition, logging, and transmission are each encapsulated in separate tasks. Each task is ranked based on a pre-configured priority. The data acquisition task is set to receive the highest priority in the firmware operating on the hardware. While the data logging task has the second-highest priority after the data acquisition task, then the data transmission task.

The data acquisition task on XBeats runs at a sampling rate ranging between 250 to 500 SPS. In the case of the 250 SPS, the acquisition task captures one sample every four milliseconds (250 samples/1000 ms = 4 ms). The data acquisition task itself requires 1ms to capture each sample. Therefore, the remaining time available for other tasks (e.g., compression, logging, and transmission) to operate after each sample acquisition equals 3 ms (4 ms - 1ms). This period is down to 1 ms at 500 SPS. However, in the proposed ECG patch, samples are processed in batches every second, in which the logging and transmission tasks are executed. Accordingly, a total of 750 ms are available for these two tasks when the sampling rate is 250.

In comparison, at the 500 rates, a total of 500 ms is available for running these tasks. This setup is a stringent time constraint in our system, and the data logging and transmission tasks will have to be completed during this time interval. Otherwise, it will be interrupted by the data acquisition task since it has the highest priority according to the intended setup to ensure data consistency and integrity. Then the classification service comes in handy as it enables the ECG to dynamically configure the number of active ECG leads in the acquisition service.

4.1.2 Data Transmission Implementation on XBeats

Data transmission represents the main bottleneck in XBeats operations, found in the communication link using BLE as a transmission protocol. The ECG patch uses the low power mode of the BLE stack instead of the classical BLE mode. Therefore, implementing the required functionalities from XBeats constitutes developing a custom BLE profile to maximize data exchange and handling between the ECG patch and BLE-enabled smart devices. To that extent, the custom BLE profile; developed for the ECG patch; describes the number of

GATT services, and GATT characteristics should be used to achieve intended functionalities.

The smallest addressable unit of data used by the ECG BLE profile is called an Attribute. Each Attribute has a 16-bit handle used when accessed via the Attribute protocol [71]. The Attribute "Type" field is identified using Universally Unique Identifiers (UUID), and it determines the kind of data present in the value of the attribute (e.g., Profile UUID, Service UUID, Characteristic UUID). Besides, the Attribute "Value" field carries data up to 512 bytes which are interpreted differently depending on the UUID type defined by the Bluetooth SIG or by the peripheral designers for custom applications [70]. Accordingly, a set of defined Attributes constitutes a BLE Characteristic. One Characteristic consists of at least value and declaration attribute. In contrast, the declaration attribute always comes before the value attribute, as shown in Table 4.1. It describes whether the value attribute can be read or written and contains the UUID of the Characteristic and the handle of the Characteristic Value attribute.

Table 4-1. A BLE "Attribute" Data Type Definition.

Handle	Type (UUID)	Value (Data)	
16 bits	16 or 128 bits	1 to 512 bytes	
	Example		**Definition**
30	0x2800	F0:FF	ECG BLE Profile Declaration
33*	0x2803	02:22:00:AA:CC	Characteristic Declaration

34	0xCCAA	ECG BLE Service	Characteristic Value

Table 4-2. An Example of a BLE "Characteristic" Declaration.

Bytes	Definition	Value	Meaning
0	Char Value Permissions	02	Permit Read on Characteristic Value
1-2	Char Value's ATT handle	22:00	0x0022 = 33
3-n	Characteristic UUID	AA:CC	0xCCAA

The value of the Characteristic Declaration attribute with handle 33 is interpreted in Table 4.2. Noting that the value of the attribute value "ECG BLE Service" with handle 34 is up to the system how the value is interpreted because it is not defined by the Bluetooth SIG [70]. Consequently, a collection of Characteristics constitutes a BLE service, while one or more services define a BLE Profile. The BLE profile describes how services can deliver the intended functionalities of the application.

Table 4.3 shows the attributes table of one of the ECG services available on XBeats; this service constitutes the continuous operation mode with 12-lead enabled. Services are shown in black; characteristics are bold, and characteristic values and descriptors are shown in grey. The other two services: "triggered mode service" and "ECG classification service" are designed similarly to the attributes in Table 4.3. Consequently, the combination of the three services allows XBeats to deliver a dynamic way of configuring the settings of the device in real-time as heart conditions develop. We leverage the new features introduced in BLE version 5.2 over earlier versions of BLE (e.g., BLE version 4.1). The first feature is the improved transmission rate (i.e., LE 2M PHY), allowing BLE-enabled devices to transfer data at a symbol rate of 2Mbps. This means we can transmit each bit in half the time compared to earlier BLE PHY, allowing a symbol rate of 1Mbps.

Moreover, the Data Length Extensions (DLE) [91] enable the packet to carry a significantly larger payload (Up to 251 bytes vs. 27 when disabled), as introduced in BLE version 4.2. The integration of DLE has increased the size of data sent in a single packet and reduced the number of the mandatory Interframe Space (IFS) delays (i.e., 150μs) between each packet sent. Accordingly, the ECG patch can transmit more data in significantly less time.

Table 4-3. The Proposed BLE Profile for XBeats: 12 Leads ECG service Attributes.

Handle	Type	Type	Hex / Text Value (default)	GATT Server Permissions	Notes
0x10	0x2800	GATT_PRIMARY_SERVICE_UUID	0xBA55 (ECG_SERV_UUID)	GATT_PERMIT_READ	Start of ECG Profile Service
0x11	0x2803	**ECG_PROFILE_CHARACTER1_UUID**	**12 00 (handle: 0x0012)** **AD 2B (UUID: 0x2BAD)**	**GATT_PERMIT_READ**	**Characteristic 1 declaration**
0x12	0x2BAD	FULL_ECG_12LEAD_UUID	00::00 (224 bytes)	GATT_PERMIT_READ |	ECG data value

				GATT_PERMIT_NOTIFY	
0x13	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)	GATT_PERMIT_READ | GATT_PERMIT_WRITE	BLE characteristic notifications enable/disable
0x14	0x2901	GATT_CHAR_USER_DESC_UUID	"ECG Data Stream" (15 bytes)	GATT_PERMIT_READ	Characteristic 1 user description
0x15	0x2803	**ECG_PROFILE_CHARACTER2_UUID**	**16 00 (handle: 0x0016)** **AD 3B (UUID: 0x3BAD)**	**GATT_PERMIT_READ**	**Characteristic 2 declaration**
0x16	0x3BAD	ECG_NUM_CHANS	0x08 (1 byte)	GATT_PERMIT_READ	Number of ECG Channels
0x17	0x2901	GATT_CHAR_USER_DESC_UUID	"Number of ECG Channels" (22 bytes)	GATT_PERMIT_READ	Characteristic 2 user description
0x18	0x2803	**ECG_PROFILE_CHARACTER3_UUID**	**19 00 (handle: 0x0019)** **CD 2B (UUID: 0x2BCD)**	**GATT_PERMIT_READ**	**Characteristic 3 declaration**
0x19	0x2BCD	ECG_STREAM_FLAG_COMMAND	0x00 (1 byte)	GATT_PERMIT_READ | GATT_PERMIT_WRITE	"01:00" to enable / "00:00" to disable
0x1A	0x2901	GATT_CHAR_USER_DESC_UUID	"Stream Flag Status" (18 bytes)	GATT_PERMIT_READ	Characteristic 3 user description

4.2 XBeats RPM Prototype Implementation

This section lays out the tools and software used in building the proposed RPM framework, as shown in Figure 4.2. The core of the proposed framework leverages the Confluent platform [88], which complements Kafka with additional features and integration tools using the concept of microservices. The Confluent Platform is used as an assimilation layer working with all data stream pipelines. The framework design focuses on the functionalities and services provided to the end user. Therefore, we present the tools and software used to implement the proposed framework; however, the selected tools and software are not limited to specific software as they can be replaced by other software with similar functionalities.

4.2.1 XBeats RPM Framework Infrastructure

Two types of implementations are used to deploy the functional components of the framework, which are container-based and native-based installations. A container-based installation is a trending approach in software development and operations (i.e., DevOps). Docker represents one of the famous container providers with open-source distributions. Moreover, native-installation methods are utilized to install other software components. Our implementation of the RPM framework provides a use-case application for real-time ECG heart monitoring and analytics. The deployment of the RPM framework takes place on a managed virtual machine (VM). The VM comes with an Ubuntu Server (Version 22.04 LTS) as the operating system, where the VM comes with 64 gigabytes (GB) of RAM, 32 virtual cores and 160 GB of storage. While we install the Docker engine and Docker compose library to run multi-container Docker applications. The streaming engine utilizes the Confluent Kafka platform, and the event streaming engine provided by ksqlDB are configured and deployed using Docker compose commands. While the message queuing broker for MQTT (i.e., HiveMQ) is installed using the native library for Unix-based systems. HiveMQ [87] is a java-based open-source MQTT broker providing a reliable messaging platform and implements all MQTT protocol standard features.

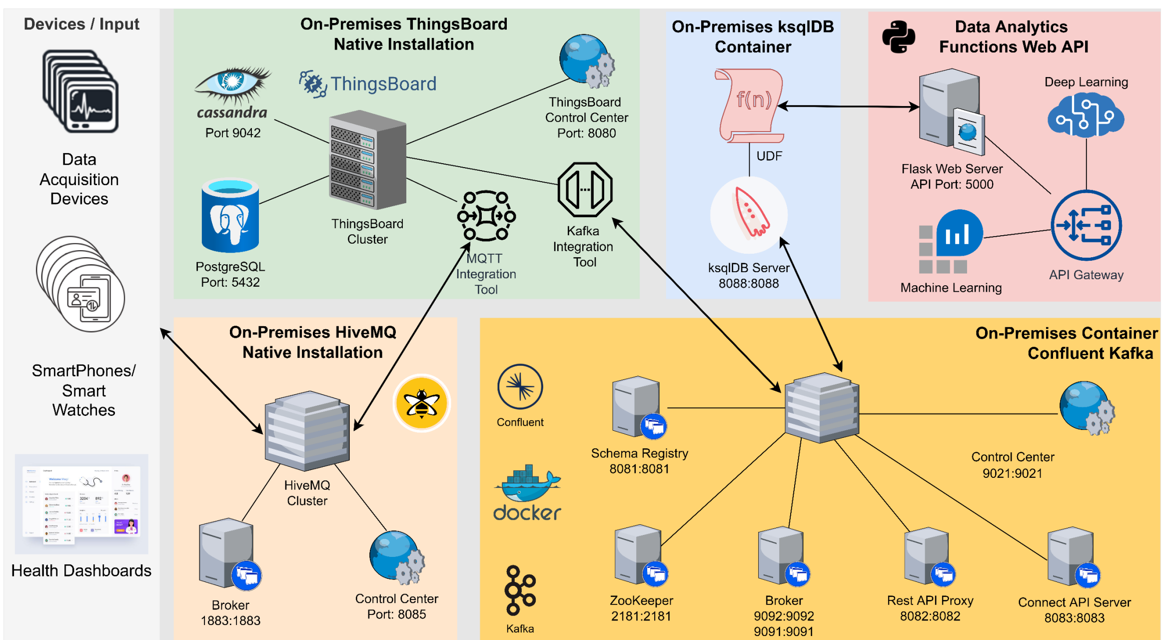


Figure 4.2. A high-level implementation of the XBeats RPM framework.

Similarly, we install ThingsBoard [53] using the on-premises installation distribution for Ubuntu 20.04 LTS. Accordingly, three different types of databases are installed. Two databases explicitly serve ThingsBoard using a hybrid approach, PostgreSQL database to store all entities' information (e.g., users, devices, assets and dashboards). On the other hand, the Cassandra database is used to save time-series data. Cassandra database provides a NoSQL database management system designed to handle large amounts of real-time data across distributed servers. Therefore, this setup avoids single-point failures at different framework levels, where all services use distributed architectures with clusters of at least two brokers for each service.

4.3 Data Flow and Configurations for XBeats RPM Framework

We give an example of how a healthcare provider can use the XBeats RPM framework to perform remote ECG monitoring for patients with chronic heart diseases. ECG testing collects data at significantly high acquisition rates (e.g., 500 samples per second). The ECG test requires reliable unbounded streaming services to handle large streams of incoming data in real-time. To this extent, we explain the data pipeline and workflow as shown in Figure 4.3 for an ECG test using the XBeats RPM framework. Each step in the data pipeline is labelled 1, 2a, 3a, and 3b. The initial stage starts by collecting ECG signals using the XBeats ECG patch. Once the ECG patch is switched on, the device connects to the nearest paired mobile device and sends the collected information as a data stream using BLE. Then, the smartphone runs our custom-designed mobile application to process the ongoing ECG data streams from the ECG patch and forwards them to the backend while using the MQTT protocol provided by the message queueing server. If the device is connecting to the backend for the first time, we enable automated device provisioning with the help of the MQTT topic filter feature. The topics are designed using the wildcard feature in MQTT, allowing the mobile application to subscribe to multiple topics simultaneously. Therefore, this feature enables the framework to register a new device if the topic the mobile device tries to subscribe to is not already registered.

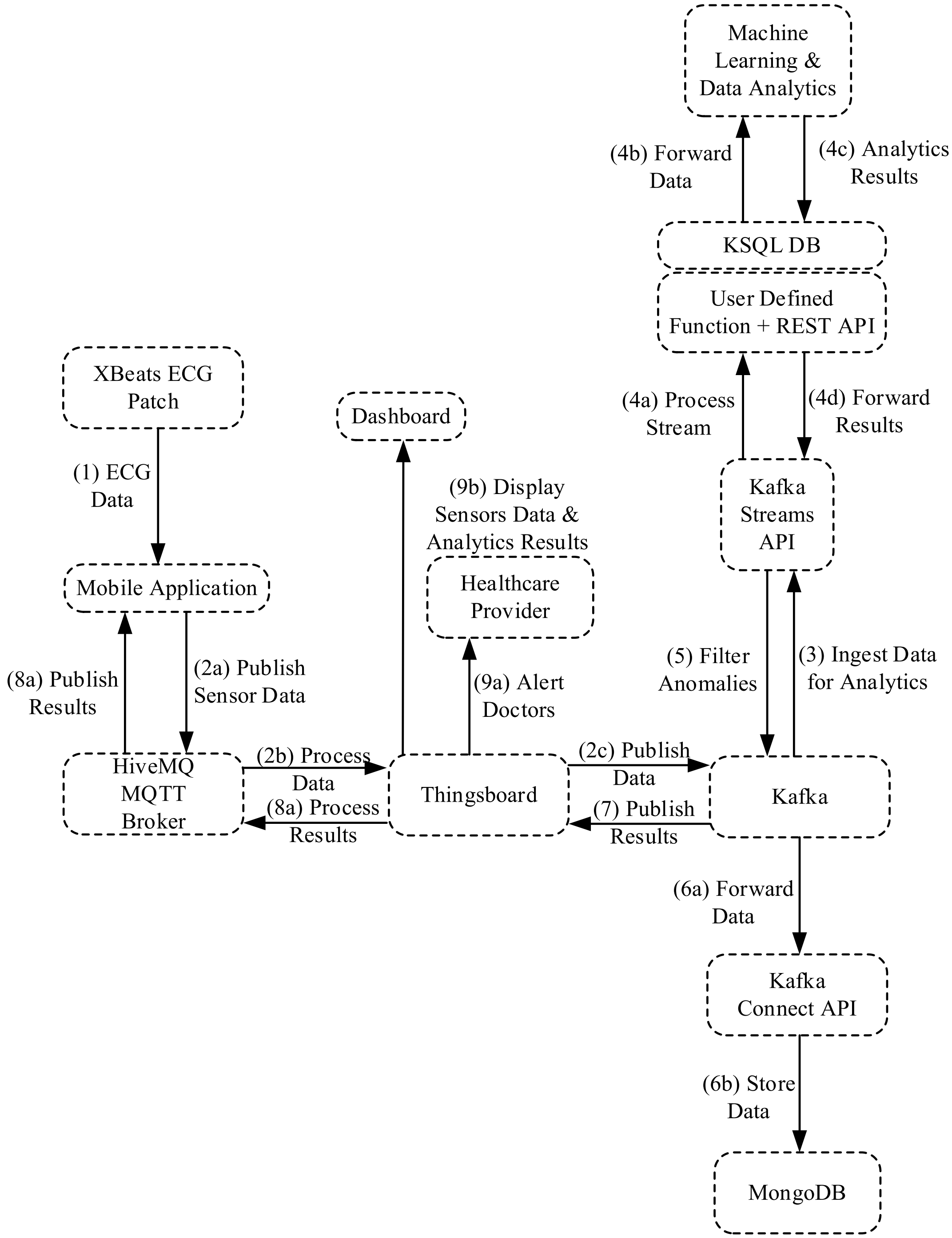


Figure 4.3. Data Pipeline and Workflow of the XBeats RPM Framework.

The following represents the topic definition used by our mobile application to register and subscribe to the ECG services provided by the XBeats RPM framework:

```
Topic: health/mqtt-integration/sensors/ecg/+(device_id)/data
```

To that extent, the mobile application (step-2a) publishes the collected sensor data in patches every 10 seconds to the backend system through an MQTT connection established with the MQTT broker. We use the following data structure in each patch of data published by the mobile application:

```
{{"device_{id}": "SN-002", "data": %s}}
```

The ThingsBoard MQTT Integration acts as an MQTT client. It subscribes to topics and converts the data into telemetry and attribute updates. The MQTT broker transfers the data from each topic to the middleware layer represented in the ThingsBoard platform. We process the ECG data received on ThingsBoard (step-2b) via the MQTT integration tool by defining filters on the topics using the device ID. Consequently, we publish (step-2c) the filtered MQTT data to the Kafka streaming engine, where we perform data analytics operations on the ECG data.

Consequently, event and stream processing techniques are applied to create dedicated streams for data analytics. The implementation utilizes the Kafka Streams API to establish a connection to the ksqlDB server; ideally, this step is considered the entry point to ingest (step-3) the collected data for the classification and analytics components of the proposed framework. Accordingly, we use our custom user-defined functions (UDF) to make automated callings to the ECG data analytics API. Python Flask is used to implement a web API to standardize the usage of the data analytics functions. The analytics functions utilize machine learning (ML) and deep learning algorithms. Likewise, the web API enables the framework to seamlessly add new analytics functions without disturbing the operation lifecycle of the framework. The current web API implementation integrates two functions. The first function integrates a binary classification ML function to classify ECG data into normal or abnormal signals. The second function detects the PQRST feature points from each ECG signal and calculates the heart rate. Then, the ksqlDB CLI is used to create (step-4a) a persistent query that generates a new Kafka topic aggregating the received ECG data every 60 seconds, as illustrated in Figure 4.4.

```
CREATE STREAM ECG_RAW_DATA_AGGREGATED WITH
    (KAFKA_TOPIC = 'ECG_RAW_DATA_AGGREGATED',
     PARTITIONS = 1, REPLICAS = 1)
AS SELECT
     ECG_RAW_DATA_STREAM.DEVICE_ID,
COLLECT_LIST (ECG_RAW_DATA_STREAM.DATA) KSQL_COL_0
     FROM ECG_RAW_DATA_STREAM
WINDOW TUMBLING (SIZE 60 SECONDS)
GROUP BY
     ECG_RAW_DATA_STREAM.DEVICE_ID
HAVING (COUNT (ECG_RAW_DATA_STREAM.DATA) = 5)
EMIT CHANGES;
```

Figure 4.4. ksqlDB Query for Creating a New Data Streaming Topic with Data Aggregation every 60 Seconds.

The query automatically generates a new stream with the topic name ”ECG_RAW_DATA_AGGREGATED” that aggregates the received ECG data to one data entry every 60 seconds.

```
CREATE STREAM ECG_CLASSIFICATION WITH
     (KAFKA_TOPIC = 'ECG_CLASSIFICATION',
     PARTITIONS=1, REPLICAS = 1)
AS SELECT
     ECG_RAW_DATA_AGGREGATED.DEVICENAME,
     BINARY_CLASSIFY_ECG_LEADS
       (ECG_RAW_DATA_AGGREGATED.DEVICENAME,
       ECG_RAW_DATA_AGGREGATED.DATA)->UDF_MESSAGE_ECG,
     ECG_RAW_DATA_AGGREGATED.TS
FROM ECG_RAW_DATA_AGGREGATED
EMIT CHANGES;
```

Figure 4.5. ksqlDB Query for Creating a New Data Streaming Topic Calling the ML UDF on the Aggregated ECG Data Stream.

Furthermore, the query in Figure 4.5 creates another data streaming topic ready for data analytics which executes the ”BINARY_CLASSIFY_ECG_LEADS” UDF. Then, the UDF function waits (step-4c) for the analytics results from the web API and then forwards (step-4d and -5) the results to a new Kafka topic. The topic carries the analytics results in a data tuple that references the original data used to perform the analytics. Moreover, we use the Kafka Connect API to establish a link with the MongoDB server and save (step-6a and -6b) the ECG data and the analytics results once a new data entry is added to the queue of any Kafka topic. Simultaneously, we use the Kafka integration tool in Things-Board to consume the (step-7) analytics results.

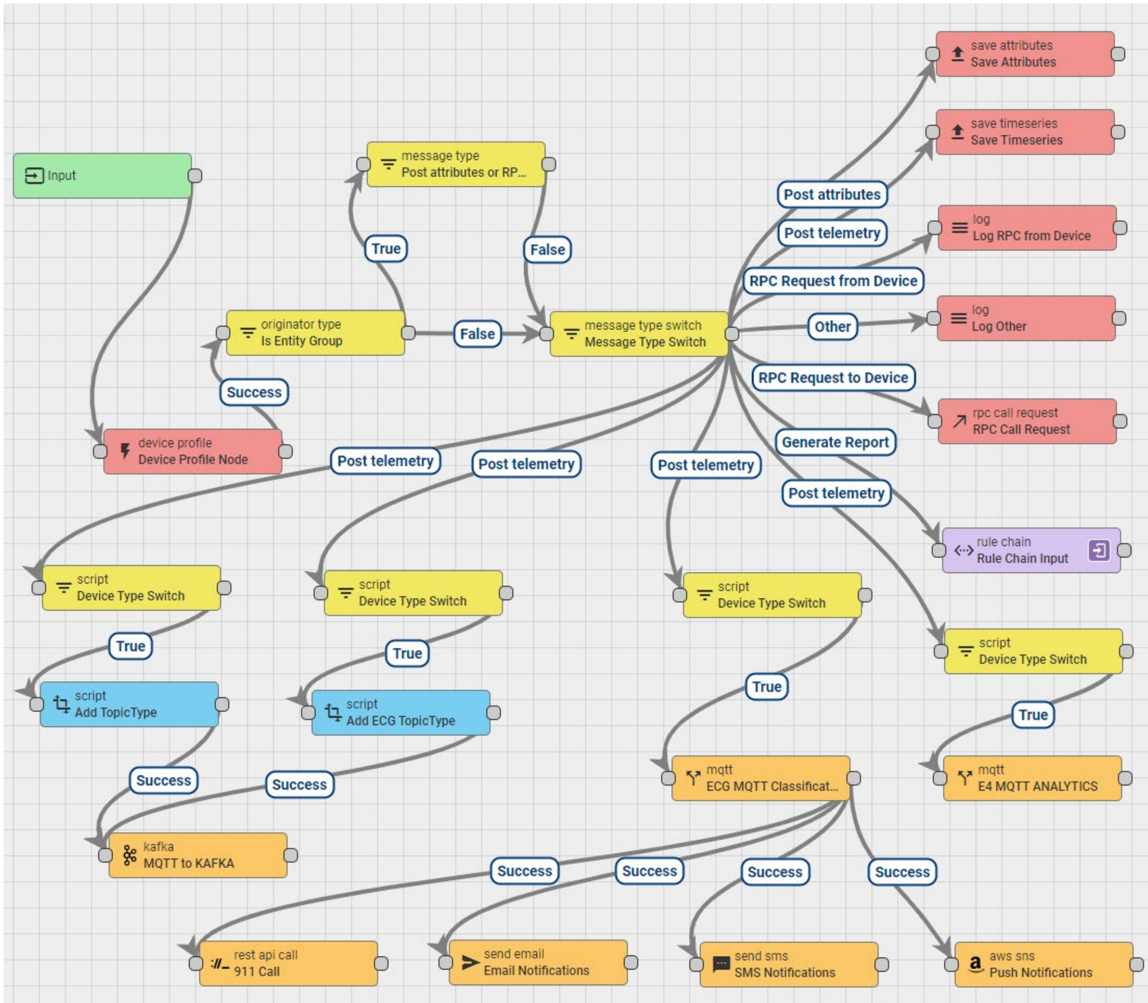


Figure 4.6. XBeats RPM Framework ThingsBoard Rule-chain Implementation.

Consequently, a custom rule engine using ThingsBoard APIs is implemented, as shown in Figure 4.6, to prepare the telemetry ECG data and analytics results and route them to the intended destinations. Moreover, the rule-engine script checks (step-8a) the received analytics results for anomalies; if an anomaly is detected, a notification is published (step-8b) to the healthcare provider through push notifications to the developed mobile application. Likewise, notifications are sent like emails, text messages, push notifications, and emergency calls (step-9a) to the healthcare provider in similar scenarios. Lastly, the healthcare provider gets access to a web-based dashboard that displays all vital information in real-time for all the patients who belong to the same healthcare provider. The healthcare provider can filter the data by the device ID or display ECG charts for specific periods.

4.4 Summary

This chapter discusses the directions and steps for the XBeats hardware prototype and implementation of the underlying RPM framework supporting the operations of the XBeats ECG patch. The initial prototype of the XBeats ECG patch is provided, where we explain the reasons for selecting the presented hardware components. Then emphasize the technical specifications required to guarantee the intended functionalities for performing a standard 12-lead ECG testing. Furthermore, we highlight the design of the proposed BLE Profile, enabling the full potential of the ECG patch for a seamless wireless data exchange.

In the quest to design a horizontally scalable RPM framework, we provide detailed instructions on the environment and tools needed to set up the framework. The instructions include a use-case scenario installing and integrating various framework components using containerized and native deployments techniques. Moreover, we present the required settings to utilize data streaming services for event processing and preparing the data for real-time analytics. The provided queries work in tandem with the custom UDFs proposed to utilize the web-based data analytics APIs, emphasizing the ability to scale vertically concerning the enabled analytics functions.

Chapter 5. Experimental Evaluation and Results

This chapter illustrates the performance evaluation steps used to evaluate the proposed XBeats ECG patch in providing continuous 12-lead ECG monitoring. Section 6.1 highlights the objectives of the conducted experiments in evaluating each component of XBeats while performing the intended remote ECG testing functionalities. The remainder of this chapter is organized as follows: Section 5.2 briefly explains the data sets used in evaluating the XBeats ECG patch. The first data set is created by collecting data from the TechPatient CARDIO V4 heart simulator, while the other data set is the PTB-XL, the most extensive 12-lead ECG data set. Section 5.3 evaluates the data acquisition functionality on the XBeats ECG patch. The evaluation includes setting a benchmark for each data collection concerning the sampling rate and ECG signal quality, followed by evaluating the actual sampling rates offered by XBeats under various modes of operation using BLE for data transmission. Furthermore, we discuss the proposed binary classification module implemented on an edge node and evaluate the classification accuracy in detecting irregular heartbeats. Section 5.4 focuses on the power consumption analysis of XBeats and provides power optimization methods to save the battery lifetime of the ECG patch. Finally, the chapter is summarized in Section 5.5.

5.1 Experiments Objectives

The evaluation techniques and testing scenarios are presented to verify and validate functional components in the proposed ECG framework. These include measurements of the operation modes, useful sampling rates, and energy consumption analysis concerning the hardware constraints. The hardware constraints in each of the following experiments are mainly related to the data acquisition time constraints, the quality and correctness of the acquired ECG signals, the accuracy of the classification algorithm and the overall battery lifetime. The experiments are organized in the same order the proposed framework is presented in terms of the functional components. Therefore, the first experiment evaluates the ECG data acquisition under different operation modes while running the data logging subroutine, followed by data transmission using BLE. The second evaluates the ECG data classification module. Then, we analytically calculate the energy consumption of the ECG patch over time. The experimental setup consists of the following steps:

1. Collect ECG data in real-time using the prototype hardware of the ECG patch. The experiment includes acquiring ECG data at different modes of operation: standard 12-Lead ECG data under the "continuous" mode of operation; One and three ECG leads under the triggered mode of operation; standard 12-Lead ECG data under the offline mode of operation.

2. Evaluate the effective ECG data sampling rate compared to the theoretical data acquisition values provided by the analog to digital converter. The evaluation is performed on each of the operation modes above.
3. Evaluate the proposed ECG classification service implemented on an edge node. The evaluation steps include comparing the accuracy and processing time of six different techniques, which is concluded by the selected classification techniques for our edge classification service.
4. Calculate the power consumption footprint and the energy-saving of applying the triggered operation mode while activating the edge classification service.

5.2 Data Sources Description

The TechPatient CARDIO V4 heart simulator generates standard 12-lead ECG data, where the ECG is connected via 12-lead ECG cables. Then the ECG patch collects ECG data to build an ECG dataset that is used in later stages for testing. Also, it removes the need for connecting the prototyped hardware to actual patients in this early research stage. The simulator can generate real-time ECG waveforms for different cardiac conditions and support two modes of operation: ECG mode and Rhythmic mode. The ECG mode provides realistic 12-lead ECG waveforms. The rhythmic mode simulates 45 predefined arrhythmias or heart diseases, such as ventricular tachycardia and ventricular fibrillation. The device can be configured in the ECG mode in 1 beat per minute (BPM) increments from

20 to 240 BPM and 2 BPM increments from 240 to 300 BPM. Therefore, we created two datasets using the ECG Simulator: normal and abnormal ECG signals to test the ECG patch.

On the other hand, a larger dataset (i.e., PTB-XL [56]) is used for training and validating the developed classification algorithms and the dataset created from the TechPatient CARDIO V4 heart simulator [92] using the ECG patch. The purpose of using the PTB-XL is to extend the developed classification model to cover a broader range of heart diseases and provide better accuracy. The PTB-XL dataset [56] provides a freely accessible ECG dataset of unprecedented size hosted by PhysioNet. The dataset comprises 21837 clinical 12-lead ECG records measured simultaneously, representing the conventional 12-lead ECG records. The length of each record is 10 seconds. The tests were performed on 18885 subjects (52 % were male, and 48 % were female). The digitized signals are available at two sampling frequencies, 500 and 100 samples per second. Each record in the dataset has an attached header file describing each subject's demographic information, health conditions, doctor's comments, age, gender, diagnoses, number of records, number of samples, and the sampling rate. The records are categorized into five superclasses (NORM: normal ECG, CD: conduction disturbance, MI: myocardial infarction, HYP: hypertrophy, and STTC: ST/T changes) from which a 24-subclasses are derived, forming a multitude of diverse ECG data as a resource for ECG analysis algorithms. Accordingly, we aggregate all the abnormal signals (24 different abnormal heartbeat classes) into one category to make up a binary data set for normal and abnormal conditions. The size of the PTB-XL dataset makes it a valuable asset in machine learning and deep learning applications.

5.3 XBeats ECG Data Acquisition Evaluation

A sequence of experiments is performed to validate the resultant sampling rate and the correctness of the digitized ECG data. Consequently, the TechPatient CARDIO V4 heart simulator is connected to the ADS1298 evaluation hardware. The purpose of this step is to create a benchmark for the ECG patch when evaluating the ECG patch while acquiring ECG data in real-time. Furthermore, these experiments are implemented using the TechPatient CARDIO V4 simulator to simulate the heart's electrical activity, as, during the time this research was conducted, we did not have the required licenses to perform clinical trials on patients or FDA approvals.

5.3.1 ECG Data Collection Benchmark

We use the ADS1298 ECG frontend evaluation software built on top of LabVIEW libraries. The initial setup includes generating square signals using a square generator. Following that step, we start to capture the signals using the evaluation software using the ADS1298 development kit. Then use the ECG patch to collect the same signals and compare them to the benchmark signals collected by the evaluation software. In the second step, we use the evaluation software to collect ECG data through the ADS1298 development kit simulated directly from the TechPatient CARDIO V4 heart simulator. Then we compare the collected ECG data by the evaluation software to the ECG data collected by the proposed ECG patch. The signals displayed in Figure 5.1 and Figure 5.2 represent a full set of 12-lead for a standard ECG test, where Figure 5.1 shows the ECG limb leads, and Figure 5.2 shows the ECG chest leads. The data retrieval process was accomplished by selecting from two methods: Read Data Continuous (RDATAC) and Read Data (RDATA). The RDATAC method sets the device to continuously read data without sending any subsequent commands or further configuration.

In contrast, the RDATA reads data output from the output register once triggered by the data-ready flag. After a successful data cycle, 216 bits of data are available to read from the output register. The 216 bits (27 bytes) are formatted as follows: 24 status bits + 24 bits of data per channel x 8 channels = 216 bits. The low-power mode starts at 250 SPS, generating 6750 bytes every second, while the high-resolution mode starts at 500 SPS, generating 13.5 Kbytes every second. Table 5.1 shows the size of the ECG data acquired and saved on the internal storage attached to the ECG patch. The digitized ECG signals are derived using the formulas illustrated in Table 5.2.

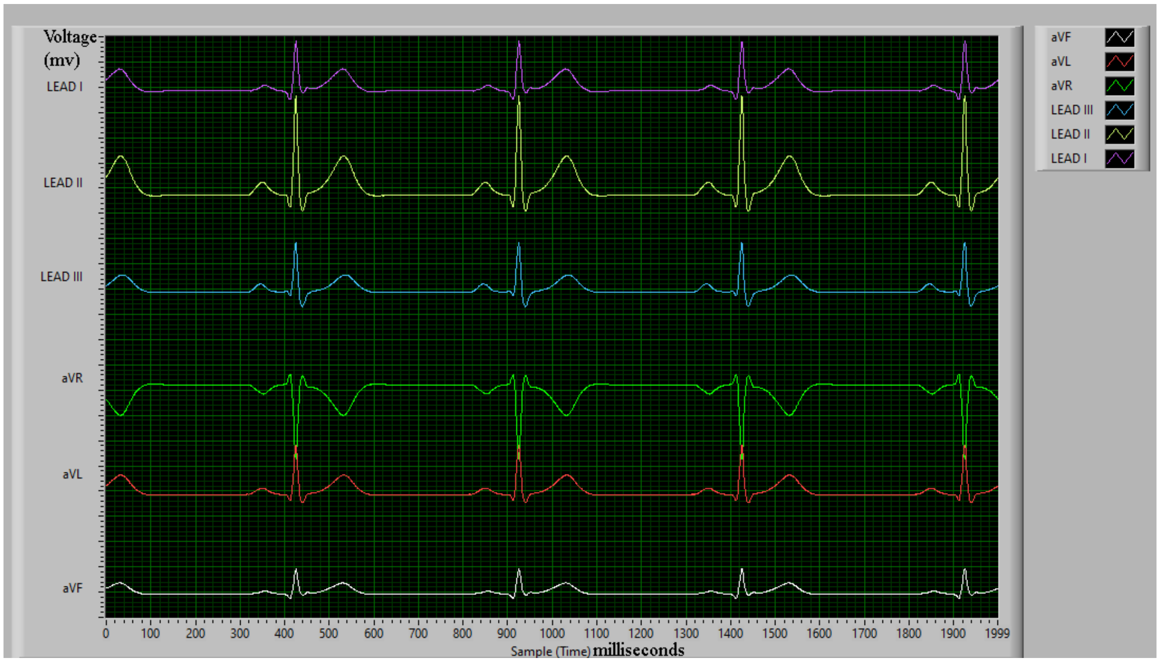


Figure 5.1. A sample of the collected ECG data using the ADS1298 TI evaluation software using a maximum sampling rate of 500 samples/sec for four seconds: Limb leads corresponding to the first group of leads (i.e., I, II, III, aVR, aVL, aVF).

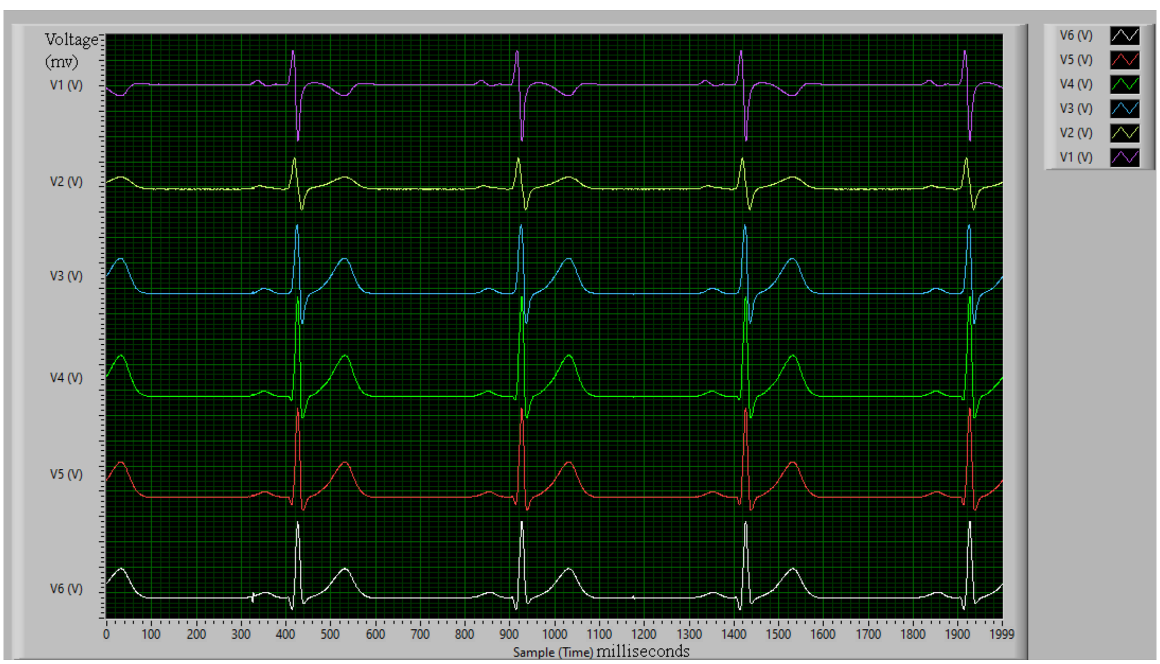


Figure 5.2. A sample of the collected ECG data using the ADS1298 TI evaluation software using a maximum sampling rate of 500 samples/sec for four seconds: Chest leads corresponding to the second group of leads (i.e., V1, V2, V3, V4, V5, V6).

Table 5-1. Collected ECG Data Size over Different Periods.

Sampling Rate Time Interval	250 SPS (Low-Power)	500 SPS (High - Resolution)
1 Second	6.75 Kilobyte (KB)	13.5 KB
1 Minute	405 KB	810 KB
1 Hour	24.3 Megabytes (MB)	48.6 MB
24 Hours	583.2 MB	1.1664 Gigabyte (GB)

Table 5-2. ECG 12-Lead Derivations.

Analog Input	Derived Lead	Polarity	Digitally Generated Leads
Channel 1	V6 = V6 - WCT	Unipolar	Lead III = Lead II – Lead I
Channel 2	Lead I = LA(1) – RA(2)	Bipolar	aVF = (Lead II + Lead III) / 2
Channel 3	Lead II = LL(3) – RA	Bipolar	-aVR = (Lead I + Lead II) / 2
Channel 4	V2 = V2 – WCT (*)	Unipolar	aVL = (Lead I – Lead III) / 2
Channel 5	V3 = V3 – WCT (*)	Unipolar	(*) WCT = (LA + RA + LL) / 3
Channel 6	V4 = V4 – WCT (*)	Unipolar	(1) Left Arm Electrode (2) Right Arm Electrode
Channel 7	V5 = V5 – WCT (*)	Unipolar	(3) Left Leg Electrode * Wilson Center Terminal
Channel 8	V1 = V1 – WCT (*)	Unipolar	

* Wilson Center Terminal

5.3.2 XBeats Data Acquisition Hardware Prototype Evaluation

Following the benchmark setup, we evaluate the XBeats hardware prototype by applying the same configurations used in the benchmark experiment. The ADS1298 chip on the XBeats hardware prototype applies the RDATAC methods to collect ECG data continuously at a default sampling rate of 500 SPS. This setup is fixed during the whole experiment. We ran the experiment five times under different operation modes, as shown in Table 5.3. The evaluation criterion is based on the resultant data rate of the XBeats hardware prototype, which varies according to the selected operation mode. The first operation mode evaluated is the "offline mode", in this operation mode, the wireless communication module (i.e., BLE) is switched off completely. The ECG patch is programmed to enable the 12-lead ECG acquisition in the "offline mode" or the "disconnected mode". We allow this feature to guarantee that the ECG patch does not miss any vital information about the heart conditions during disconnectivity. The resultant ECG data acquisition rate in the "offline mode" is 480 SPS. In contrast, the" Disconnected mode" provided a resulting sampling rate of 370 SPS. The observations from this experiment noted that the communication module on the ECG patch enters the advertising mode [93]. A BLE device uses advertisements to broadcast packets to BLE-enabled devices around it. Then the receiving devices can act on this information or connect to receive more information. When the BLE module on the ECG patch is in advertising mode, advertising packets are sent periodically on each advertising channel to update the presence of the ECG patch to the surrounding devices until it matches with a paired BLE device and establishes a new connection. This operation adds an overhead to the data acquisition task and, thus, the reduced sampling rate.

5.3.3 XBeats Data Transmission over BLE Evaluation

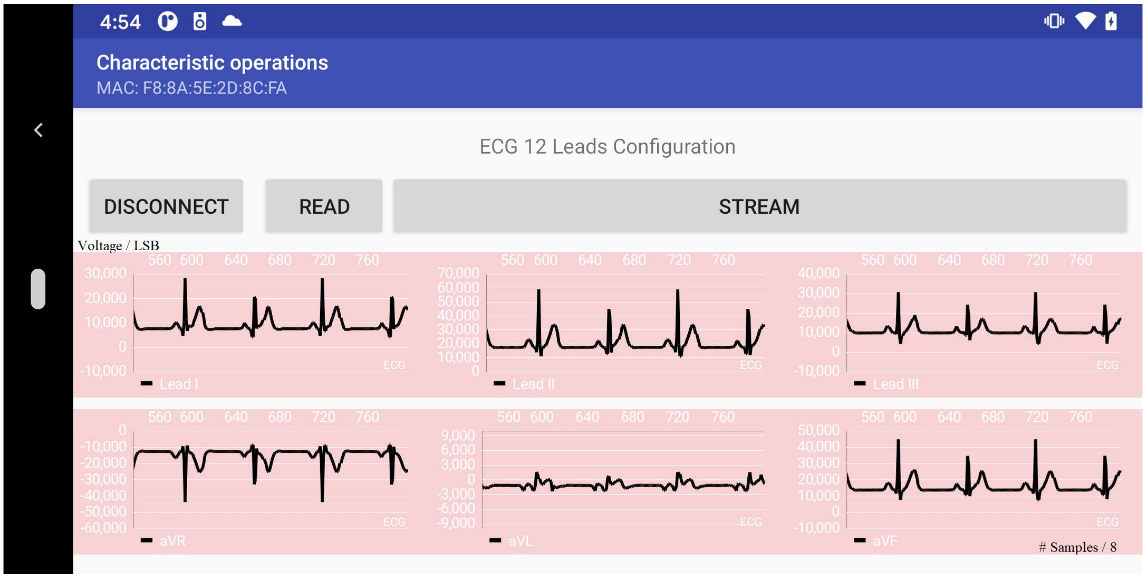
The second patch of tests includes the operation modes that rely on the BLE wireless connectivity to transfer the acquired ECG data to the paired BLE-enabled mobile device. This experiment requires a BLE-enabled mobile device to run our customized mobile application. We use the Google Pixel 3 smartphone to install our ECG patch mobile application. The smartphone supports the latest BLE version 5.2 allowing our application to utilize the Data Length Extensions, and the LE 2M PHY features provided by BLE version 5.2. Our mobile application automatically sets the physical layer to the 2 MB/s physical configurations and updates the maximum payload to 251 bytes. The payload of one successful BLE packet at the "continuous mode" contains seven samples, where each sample carries values from the digitized channels. We encapsulate the 24 bits received for each channel into an unsigned integer object. The total payload size inside on BLE packet applies the following formula:

$$Number\ of\ Channels * Object\ Size * Number\ of\ samples.$$

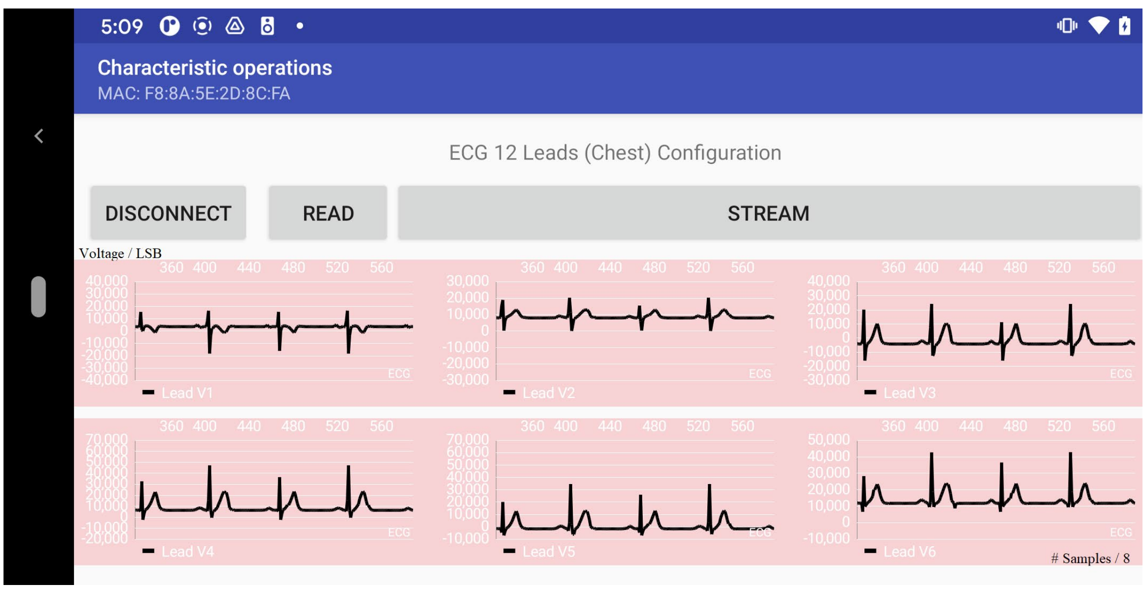
Consequently, we evaluate the "continuous mode" on the ECG patch prototype and display the streamed ECG in real-time, as shown in Figures 5.3a and 5.3b. The displayed ECG leads are computed similarly using the lead derivation in Table 5.2. The useful acquisition rate at the "continuous mode" is 343 SPS. Likewise, we apply the same setup on the "triggered mode" with one and three lead configurations. The observed useful data acquisition rate at both operation mode configurations was the same (i.e., 441 SPS).

Table 5-3. The useful sampling rate of the ECG patch over various operation modes.

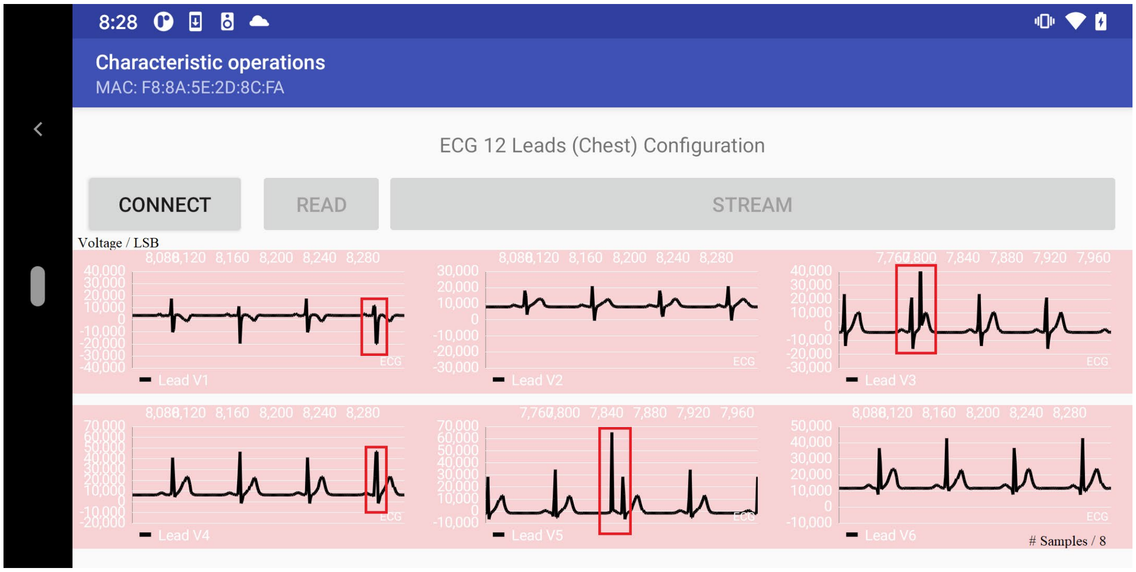
Operation Mode	Number of Channels	Number of ECG Leads	Samples / BLE Packet	Payload / BLE Packet	Acquisition Rate
Offline	8	12	N/A	N/A	480 SPS
Disconnected Mode	8	12	N/A	N/A	370 SPS
Continuous Mode	8	12	7	8 (CH) * 4 (Bytes) * 7 (Samples) = 224 Bytes	343 SPS
Triggered Mode -1	1	1 (i.e., Lead II)	56	1 * 4 * 56 = 224 Bytes	441 SPS
Triggered Mode -2	2	3 (i.e., Leads I, II, aVF)	28	3 * 4 * 28 = 224 Bytes	441 SPS


4:54
Characteristic operations
MAC: F8:8A:5E:2D:8C:FA
ECG 12 Leads Configuration
DISCONNECT
READ
STREAM
Voltage / LSB
Lead I
Lead II
Lead III
aVR
aVL
aVF
ECG
Samples / 8

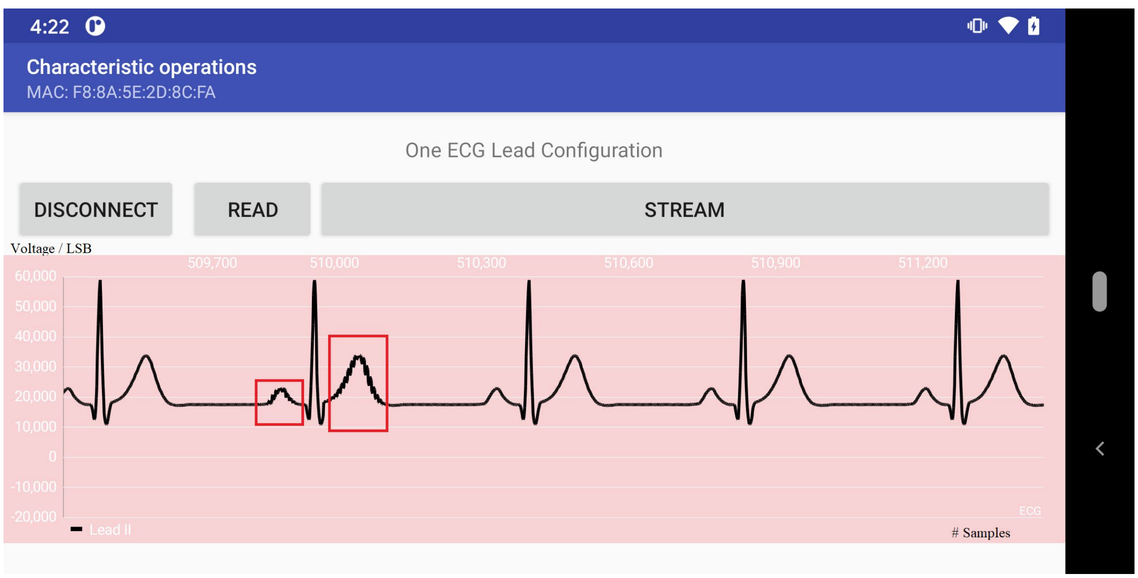
(a)


5:09
Characteristic operations
MAC: F8:8A:5E:2D:8C:FA
ECG 12 Leads (Chest) Configuration
DISCONNECT
READ
STREAM
Voltage / LSB
Lead V1
Lead V2
Lead V3
Lead V4
Lead V5
Lead V6
ECG
Samples / 8

(b)



(c)



(d)

Figure 5.3 A Sample of the streamed ECG data over BLE in real-time using the ECG patch: (a) Chest leads; (b) Limb Leads; (c) Chest leads with noise and outliers; (d) One ECG lead with outliers.

We noticed some noise and outliers in the ECG singles acquired by our ECG patch prototype, as shown in Figures 5.3c and 5.3d, while operating in the "continuous mode" and "triggered mode". We found that the primary source of noise and outliers results from sudden movements and motion artifacts on the wires connecting the electrodes to the ECG patch prototype.

5.3.4 XBeats Edge Signal Detection and Classification

The signal detection and classification represent one of the novel integrations of this work as it enables the dynamic configurations of operation modes in real-time. Moreover, it provides an additional safeguard to the ECG monitoring framework as we bring the classification closer to the patient. This feature is considered the first step of a two-phase ECG data classification of the ECG monitoring framework. The binary classification module provides a faster response by classifying the ECG data to normal heartbeat or irregular heartbeats. The second phase is performed at the backend level, where we perform deep analytics and build a correlation between real-time and historical data for better analysis and predictions. To that extent, we use the real-time ECG dataset acquired from the simulator using the proposed ECG patch to evaluate the performance of our classification algorithm. In contrast, we use the PTB-XL datasets for training the proposed classification module. To select the best classifier for our application, we compare six different algorithms, namely, random forest (RF), support vector machine (SVM), K-nearest neighbours (KNN), Decision tree (DT), logistic regression (LR), and Extra Trees Classifier, which is the ensemble learning method of the decision trees method as recommended by [39].

Table 5-4. The top five informative features used to classify the ECG signals.

Rank	Feature	Definition
1	RR_0/RR_{avg}	The current R-R interval divided by the average of the last 32 beats
2	RR_{+1}/RR_0	The next R-R interval divided by the current R-R interval
3	RR_{-1}/RR_0	The previous R-R interval divided by the current R-R interval
4	RR_{+1}/RR_{avg}	The next R-R interval divided by the average of the last 32 beats
5	hbf_3	The coefficients of fitting Hermite basis functions with polynomials degree = 3

Table 5-5. The six classification techniques accuracy and processing time.

	RF	SVM	KNN	LR	DT	Extra Trees
Accuracy	95.20%	94.19%	94.05%	93.60%	91.56%	95.30%
Processing Time	44.54 s	89.13 s	1.84 s	0.857 s	3.98 s	5.78 s

We evaluate the proposed system by calculating the performance metrics of the classification model, such as accuracy, precision, recall, and F1-score, as presented in Table 5-6. A True Positive (TP) and a True Negative (TN) refer to the numbers of correctly classified ECG signals for the normal and abnormal categories, respectively. In comparison, a False Negative (FN) and a False Negative (FP) refer to the numbers of misclassified signals for normal and abnormal conditions, respectively. Furthermore, we compare the detection response time for machine learning and deep learning algorithms on the selected datasets to evaluate the processing time of these algorithms.

Table 5-6. Performance metrics for the proposed classification models.

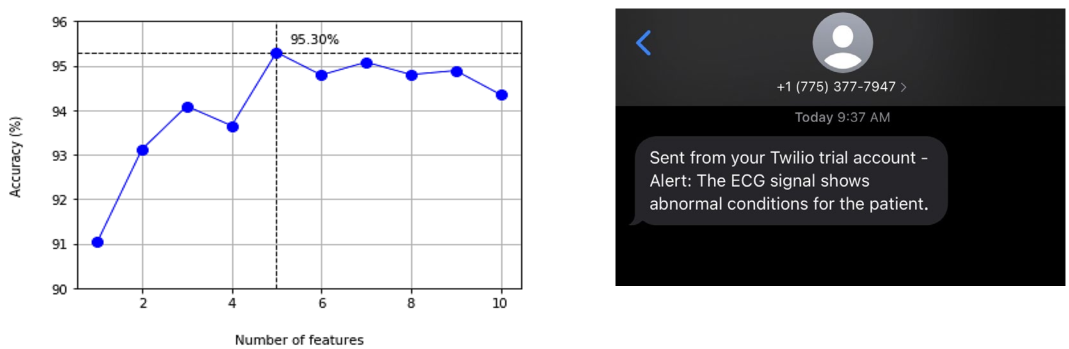
Performance Metric	Formula
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F1-score	2 × (Precision × Recall)/(Precision + Recall)

Table 5-7. Performance report obtained from the six classification techniques for normal and abnormal ECG signals.

		RF	SVM	KNN	LR	DT	Extra Trees
N	Precision	96.11%	95.26%	96.64%	94.73%	95.75%	96.17%
	Recall	98.27%	98.38%	96.68%	98.27%	94.73%	98.63%
	Fl-score	97.19%	96.79%	96.66%	96.47%	95.23%	97.38%

	Precision	83.33%	82.06%	72.98%	79.87%	60.61%	85.96%
ABN	Recall	70.31%	60.30%	72.73%	55.60%	65.85%	68.13%
	Fl-score	76.27%	69.51%	72.86%	65.56%	63.12%	77.03%

The objective of our model is to achieve the best accuracy with the minimum processing time to fit the limitations of our hardware system. We calculate the F1-score (F1), Precision, Recall, overall accuracy, and the processing time to fit the trained model of each classification technique. The training set performance is calculated using a K-fold cross-validation splitting strategy with ten folds. The results are collected using only five features from the top ten to minimize our processing time, as shown in Table 5-4. We used Python and Scikit-learn for implementation. Table 5-5 compares the six algorithms from the accuracy and time perspectives. We observe that the Extra Trees Classifier achieves the best combination of accuracy and time with the highest accuracy of 95.3% and only 5.78 seconds to classify the ECG signal. Logistic regression performs the best processing time with 0.857 seconds but with an accuracy of 93.6%, which is considered the lowest accuracy out of all classifiers. Worth noting that the processing time in Table 5-5 represents the overall processing time of the corresponding algorithm over the selected dataset. The following experiment evaluates the actual time when implementing the best algorithm in real-time on an edge device. Table 5-7 shows the performance details obtained from the test dataset for the normal signals (N) and abnormal signals (ABN). We observe that almost all the best values result from the Extra Trees classifier, which concludes that it is the best model to adopt in our system. Furthermore, we investigate the accuracy of the Extra Trees classifier with the number of features, as shown in Figure 5.4a It can be observed that the highest performance is accomplished with only five features (accuracy = 95.30%), and after that, the accuracy decreases with the growth of the number of features.



(a) (b)

Figure 5.4. (a) Accuracy of extra trees classifiers with a varying number of the top ten mutual information ranked features; (b) SMS message by Twilio sent to the healthcare provider to alert of any abnormal heartbeats.

As an edge device, we deploy the classification module on a Raspberry Pi 3B+ board. MQTT is the underlying communication protocol between our mobile application and the edge device. The mobile application is designed to publish the ECG data in patches every second. This system design decision can be changed to alternative options with 5- or 10-second intervals. Therefore, the integration of the MQTT protocol provided a pipeline for our mobile application to publish the

ECG signals collected by the ECG patch in 0.12 seconds. The edge device continuously receives ECG data until an abnormal heartbeat is detected; the system simultaneously sends the signal to different services to alert caregivers and/or healthcare providers. Figure 5.4b shows a screenshot of an SMS message sent to alert the healthcare provider of irregular heartbeats as part of the notification service provided by the proposed framework. The average processing time for ECG signal detection is 0.29 seconds. If an abnormal heart condition is detected, a message is sent out immediately to caregivers in a range of 0.57 to 0.77 seconds, which is quick enough for healthcare providers to take necessary actions. Moreover, the XBeats framework presents the classification results as recommendations to the healthcare provider and doesn’t take decisions.

5.3.5 XBeats RPM Framework Evaluation

The XBeats RPM framework can scale horizontally, allowing it to adapt to growing data volumes and changing environments. The initial setup includes two Confluent Kafka clusters with 100 MB/s Ingress/Egress data pipelines. Two connectors are configured, MongoDBSink to handle the connection to MongoDB Atlas cloud solution [89] and MQTTSourceConnector to facilitate the connection to HiveMQ MQTT broker. On top of that, a TensorFlow Python library is used to load existing analytics models and apply the loaded analytics functions on all received ECG data for deep analytics and update detection models in real-time. According to the latest benchmark performed in [90], the following upper limits were set per

CKU (Confluent Unit for Apache Kafka): The maximum number of simultaneously connected clients equals 1000 and 3000 maximum number of partitions per topic. Noting that, increasing the allocated CKUs will linearly increase the upper limits. The system achieved 16 milliseconds average latency at the Producer: 25 MB/sec and a maximum latency of 1851 milliseconds for one cluster. While extending the test to multiple clusters, the system achieved the maximum 2-CKU at bandwidth: 100 MB/s for producers and 300 MB/s for consumers bandwidth, more information about the test setup is presented in [90].

5.4 XBeats Power Consumption Evaluation

This section applies the optimization techniques introduced in Section 3.3 to each component individually. The main goal of carried-out experiments is to find the optimum values for the controlling parameters of each component (i.e., data acquisition, transmission, and storage modules). The experiments are carried out using DMM6500 6.5 Digit multimeter to measure current consumption. The multimeter is configured to collect 200K samples per second with a continuous buffer saving the data directly to external storage. We used the current digitization function on the DMM6500 that automatically calibrates the current range and adjusts the amplitude resolution. We also use MATLAB for processing and displaying the collected data by the multimeter.

On the other hand, we use the Energy Trace tool integrated into the Code Composer Studio IDE to measure the current consumption by the microcontroller while debugging different stages of the operating system using the Runtime Object View tool. Consequently, we divide the experiments into three phases. The first phase evaluates the optimization parameters on the ADS1298 data acquisition module. The second phase considers the CC1552 microcontroller responsible for the data transmission and processing on the XBeats ECG patch. The third phase evaluates the applied optimization techniques for data logging and writing on the SD card.

5.4.1 Data Acquisition Evaluation

This experiment aims to study the effective current consumption of the ADS1298 chip. We evaluate the module responsible for the data acquisition process in the XBeats patch regarding the number of enabled channels during the acquisition period. Then, we make a decision regarding the activation period of each operation mode with respect to the modified data acquisition procedure proposed in Figure 3.7. Observing the time of each operation mode when enabled is crucial in optimizing the power consumption on the ECG patch. Accordingly, reducing the operation time of some modes of operation (i.e., the continuous mode) contributes to reducing the current consumption, as proposed in Section 3.2.1. To that extent, Figure 3.7 shows the modified data acquisition and modes of operations on the ECG patch with a new subroutine that automatically switches to the triggered mode each time the continuous mode is activated after ten seconds. The expected current consumption is shown in Figure 5.5 at a sampling rate of 250 SPS. The current consumption follows a periodic pattern with respect to the configured sampling rate. If the device is configured at a sampling rate of 500 SPS, then the periodic events witnessed in Figure 5.5 shall take place every 2 ms instead of 4 ms in the case of 250 SPS.

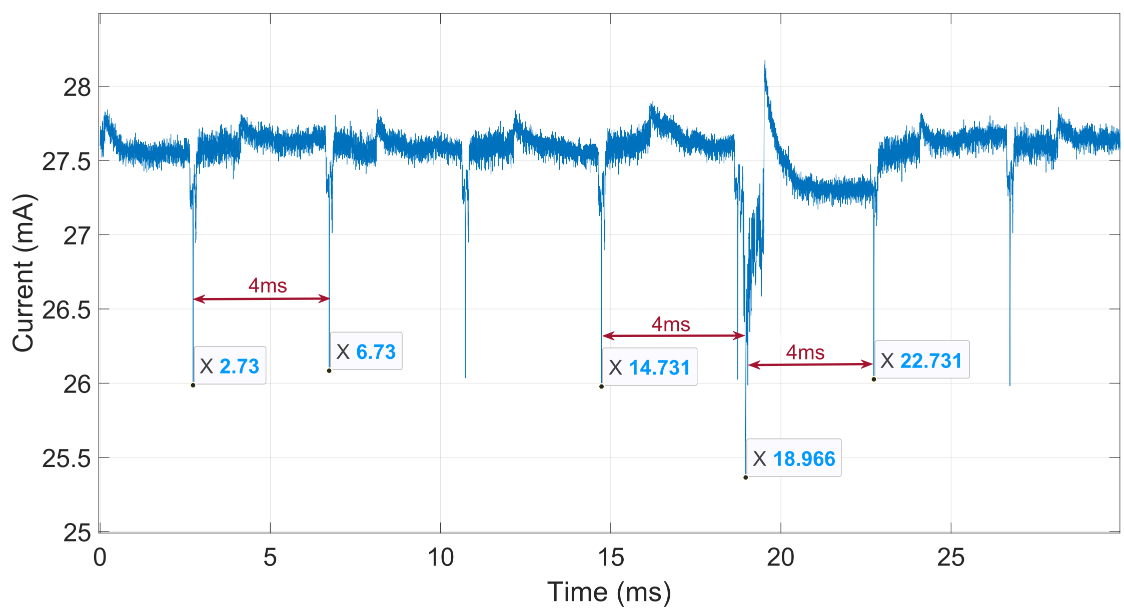


Figure 5.5. The Base current consumption profile of the ADS1298 in low power mode at a sampling rate of 250 SPS.

The current consumption is expected to be high if the device operates using the continuous operation mode. Contrary to the case when the device operates under the triggered mode, two channels are activated while the rest are disabled. The goal of the proposed optimization technique is to minimize the time the device spends profile since the device consumes operating under a high-power consumption profile (i.e., the continuous operation mode). Then the device reverts to the low power consumption profile afterwards (e.g., from continuous to triggered mode). Table 5-8 shows the underlying experiments developed for evaluating the operation and current consumption of the data acquisition module. The default setting for the sampling rate in the three experiments is 250 samples per second. The first two experiments represent the Triggered mode of operation on the ECG patch with one and three enabled channels. While the third experiment includes activating all available channels (i.e., eight channels) on the ADS1298 chip.

The results in Figure 5.5 show a recognizable drop in the current consumption versus the number of enabled channels during data acquisition. The third experiment shows the highest current consumption with an additional 2.72 mA compared to the second experiment when two channels are activated. Similarly, a 3.74 mA difference is observed when only one channel is activated. We observed that the longer the device operates in the continuous operation mode, the operation time (i.e., battery life) of the device decreases. Therefore, the device can significantly reduce the current consumption and extend the battery lifetime by applying the modified version of operations modes.

Table 5-8. The Data acquisition experiments versus the number of enabled channels and the consumed current.

Trial	Number of Enabled Channels	Average Current (mA)	Maximum Current (mA)	Minimum Current (mA)
#1	1	23.82	24.92	21.98
#2	3	24.84	25.47	22.66
#3	8	27.56	28.17	25.36

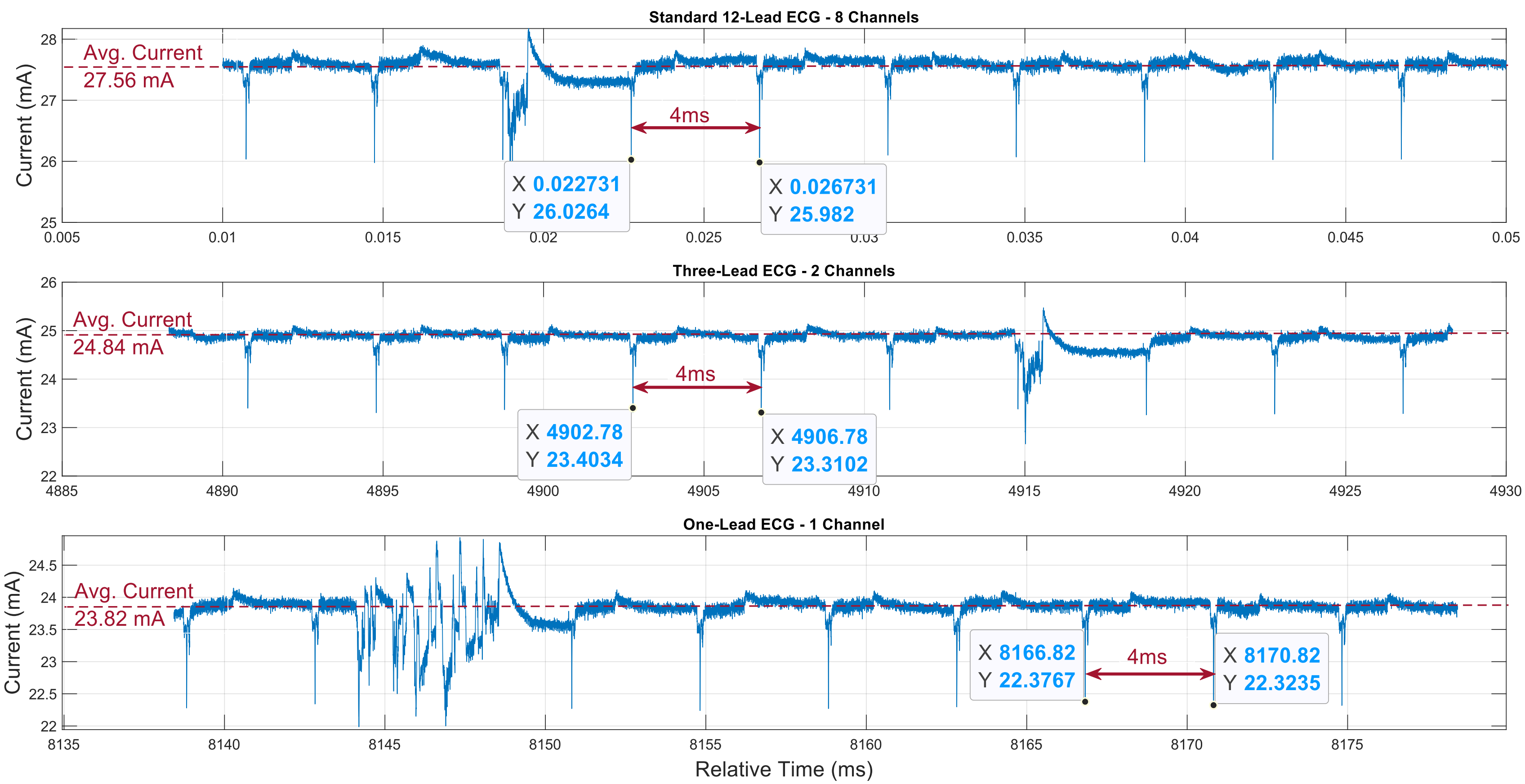


Figure 5.6. The current consumption profile of the ADS1298 chip with 8, 2 and 1 enabled channels, respectively.

Table 5-9 shows the impact of applying the modified operation modes on the power consumption profile on the ECG patch. The ECG patch saves approximately 408 mW (i.e., 8.2% prior to optimizing the operation modes) every 60 seconds when applying the proposed optimization technique during the data acquisition process.

Table 5-9. Power consumption optimization scenarios on the ADS1298 chip.

Scenario # 1 – The Continuous mode of operation applied for 60 seconds			
Time	***Average Current***	***Power Supply***	***Total Energy***
60 Seconds	27.56 mA	3 V	4960.8 mW
Scenario # 2 – The Continuous mode of operation applied for 10 seconds and the Triggered mode with two channels for 50 seconds			
Time	***Average Current***	***Power Supply***	***Energy***
10	27.56 mA	3 V	826.8 mJ
50	24.84 mA	3 V	3.726 J
Total Energy	826.8 + 3726 = 4552.8 mJ		

5.4.2 Data Transmission Evaluation

We evaluate the data processing and transmission module on the ECG patch powered by the SimpleLink microcontroller CC1352. The experiments involve tuning the parameters governing a standard BLE communication. Prior to these experiments, we analyze the time between two connection events which is known as a connection interval, as explained in Section 3.3.2. Figure 5.7 shows the periodic connection events of a standard BLE connection between the XBeats ECG patch and a mobile device. The default maximum connection interval between two consecutive connection events is 45 ms.

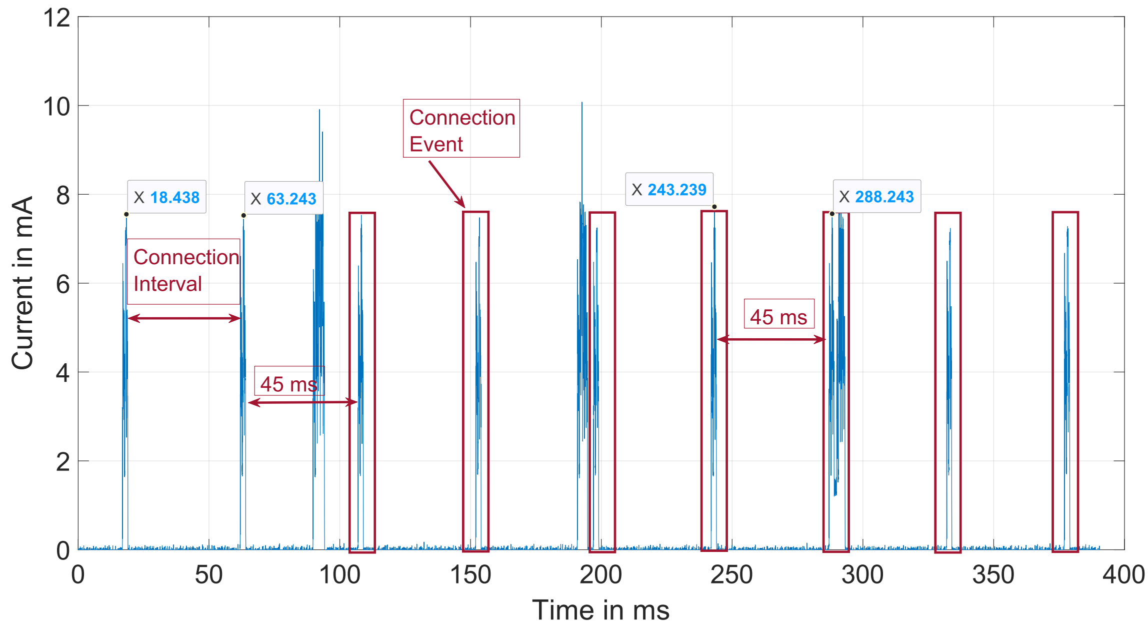


Figure 5.7. The base current consumption of the ECG patch for a BLE connection before exchanging data.

Figure 5.8 shows the current consumption of the ECG patch during data acquisition and transmission over BLE to a mobile device. The connection interval observed in Figure 5.8 has decreased compared to the connection interval observed in Figure 5.7. When the slave latency is enabled, the connection interval period becomes a factor in the effective connection interval, as explained in Section 3.3.2. Therefore, the slave latency parameter minimizes the number of connection events in a BLE connection. Therefore, the aim is to find the optimum connection interval to reduce power consumption while maintaining sufficient throughput during the connection period to maximize data transfer between the intended devices. The current $\mathbf{I_{Tx}}$ consumed by the BLE communication core module on the ECG patch equals 7.3 mA while communicating (e.g., exchanging data) with a corresponding mobile device.

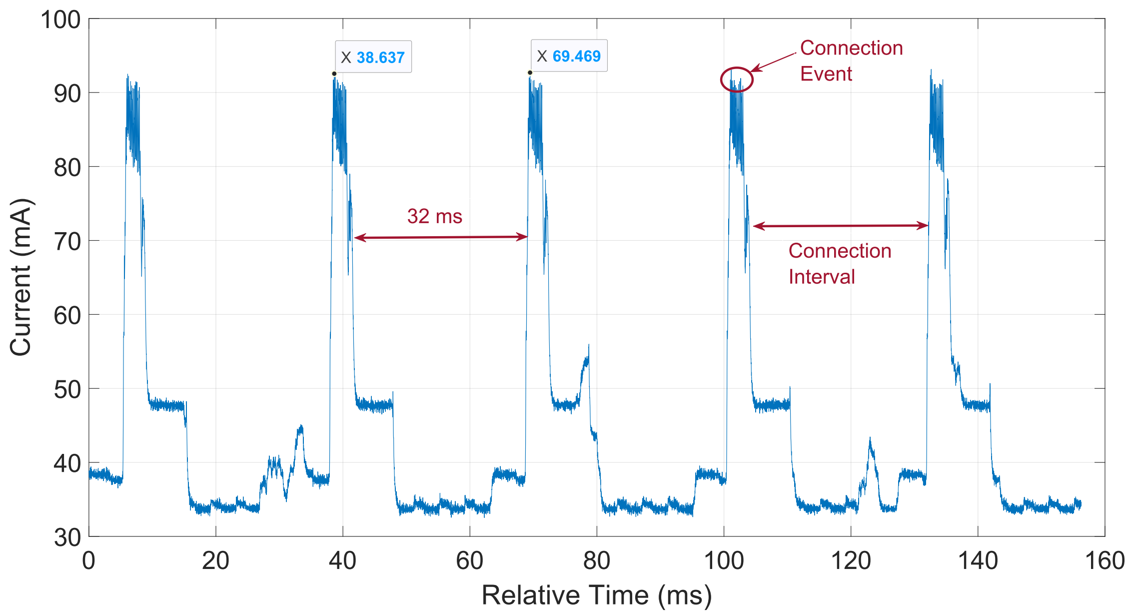
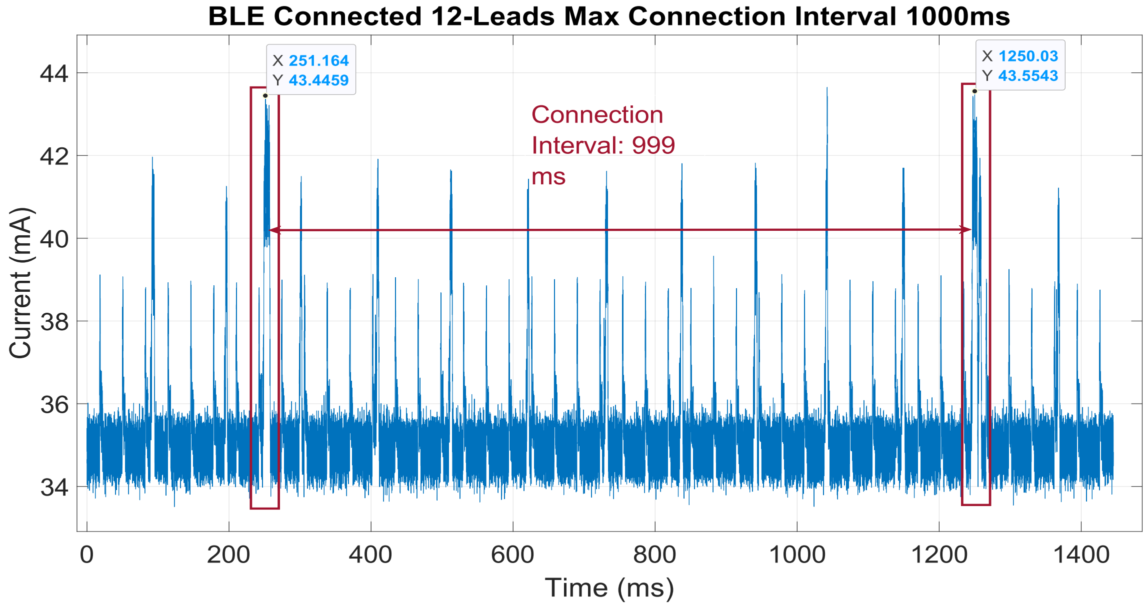


Figure 5.8 The Base current consumption of the ECG patch for a BLE connection during data exchange.

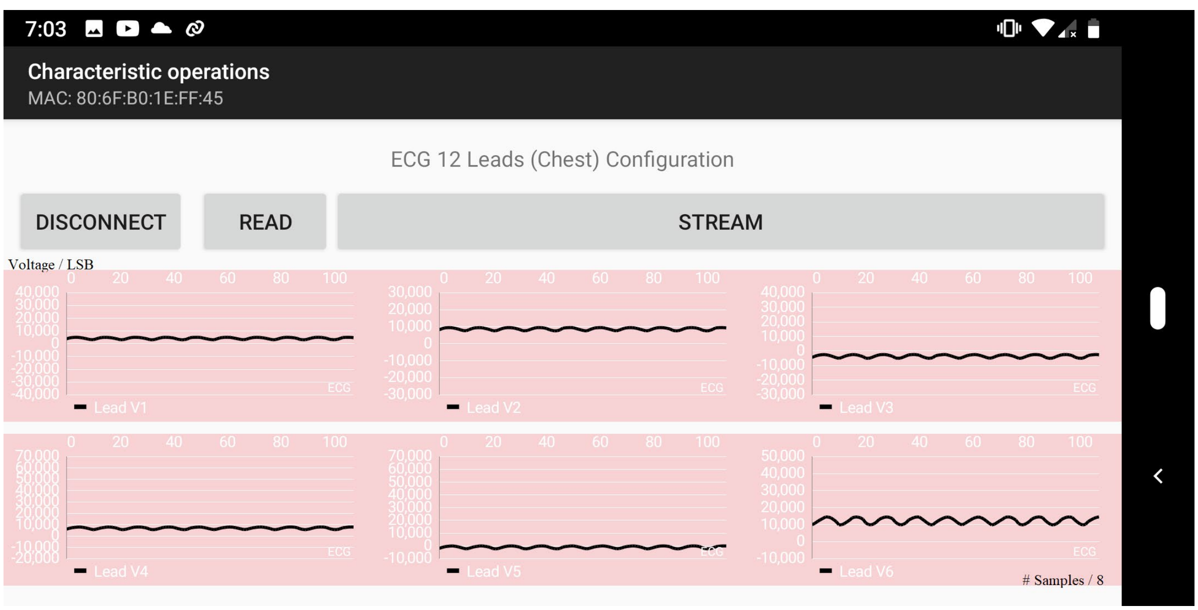
Table 5-10 lists the experiments designed for the data transmission optimization regarding the connection interval, the slave latency and the effective connection interval. The slave latency is set to zero in the four experiments to reduce the complexity of the evaluation. Moreover, the slave latency is mainly used in applications where the connectivity with the central devices is flexible, skipping many connection events. Contrary to the ECG patch, which is considered a time-stringent application, the acquired ECG data must be delivered in time. Moreover, the slave latency would incur a delay in transmitting the collected ECG data to the mobile device. This may lead to disrupting the order of the received ECG signals. Accordingly, the effective connection interval, in this case, is always equal to the defined maximum connection interval.

Table 5-10. Parameter configurations for the BLE optimization Experiments with Zero Slave Latency = 0.

Experiment	Minimum Connection Interval	Maximum Connection Interval	Effective Connection Interval
#1	500 ms	1000 ms	1000 ms
#2	250 ms	500 ms	500 ms
#3	125 ms	250 ms	250 ms
#4	100 ms	150 ms	150 ms
#5	50 ms	75 ms	75 ms

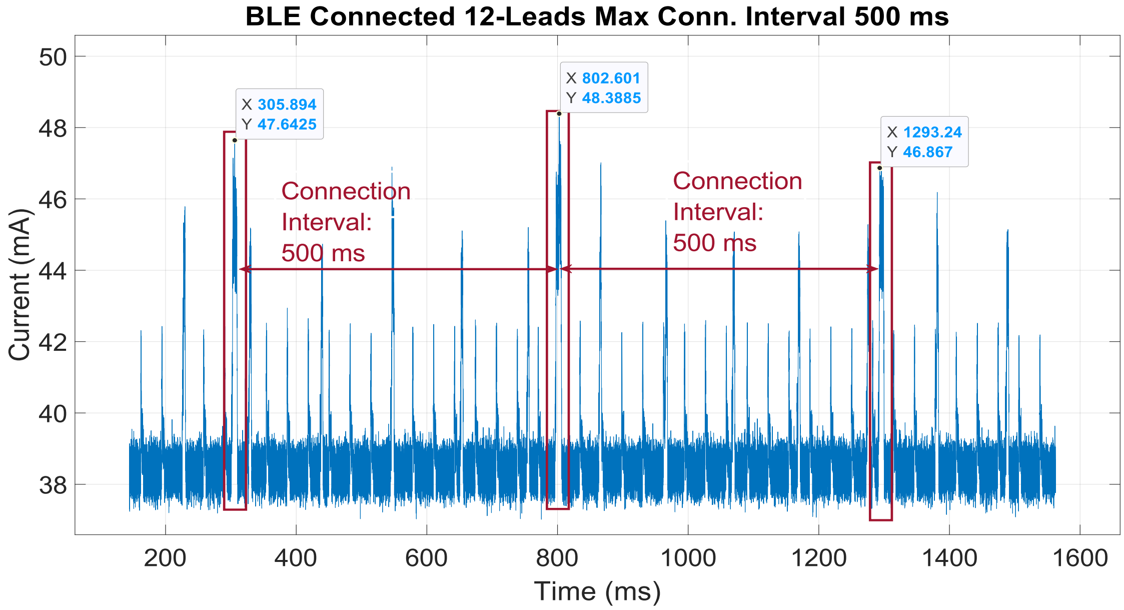


(a)

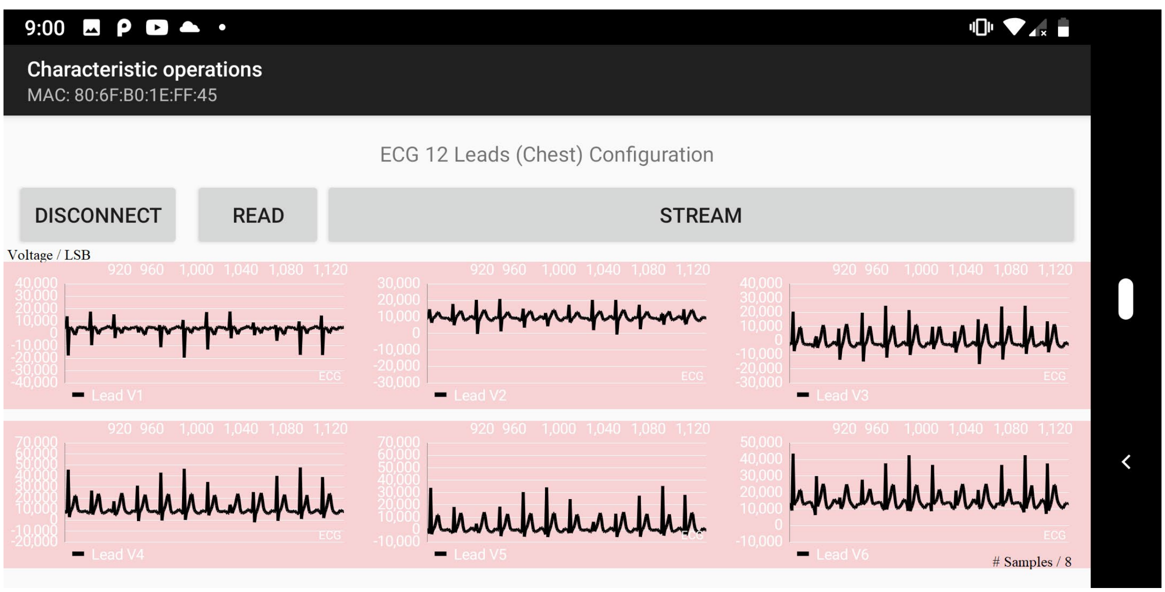


(b)

Figure 5.9. Experiment #1: Data transmission using BLE with a Maximum Connection Interval of 1000 ms.

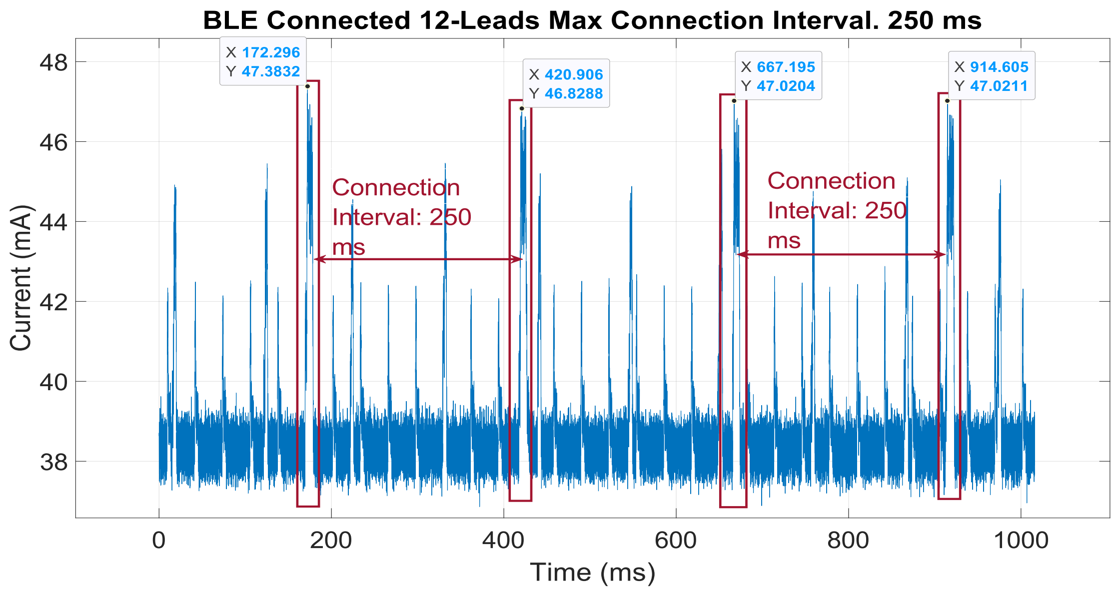


(a)

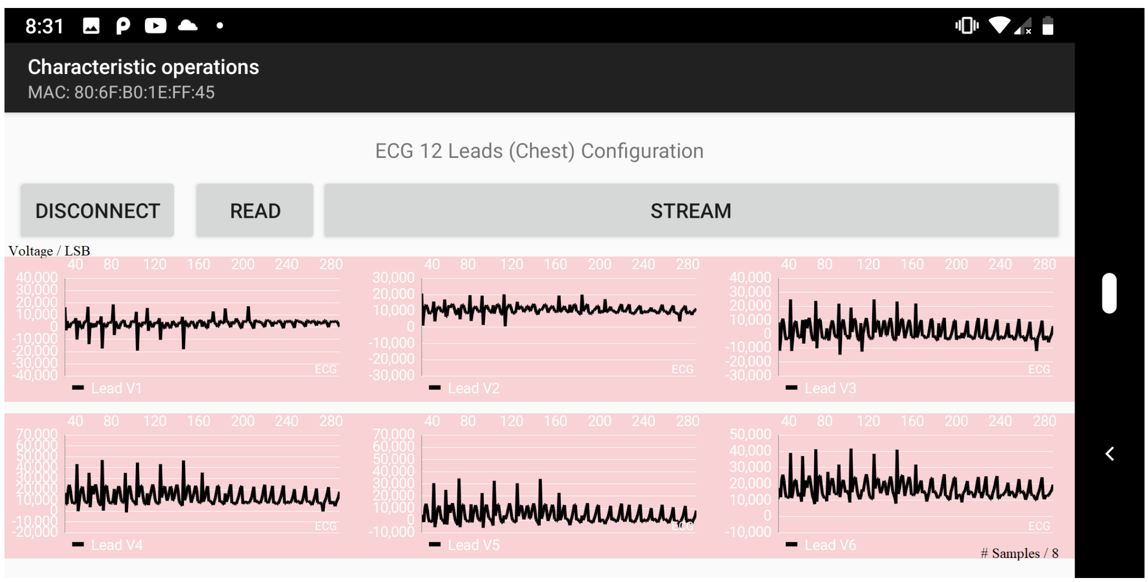


(b)

Figure 5.10. Experiment #2: Data transmission using BLE with a Maximum Connection Interval of 500 ms.

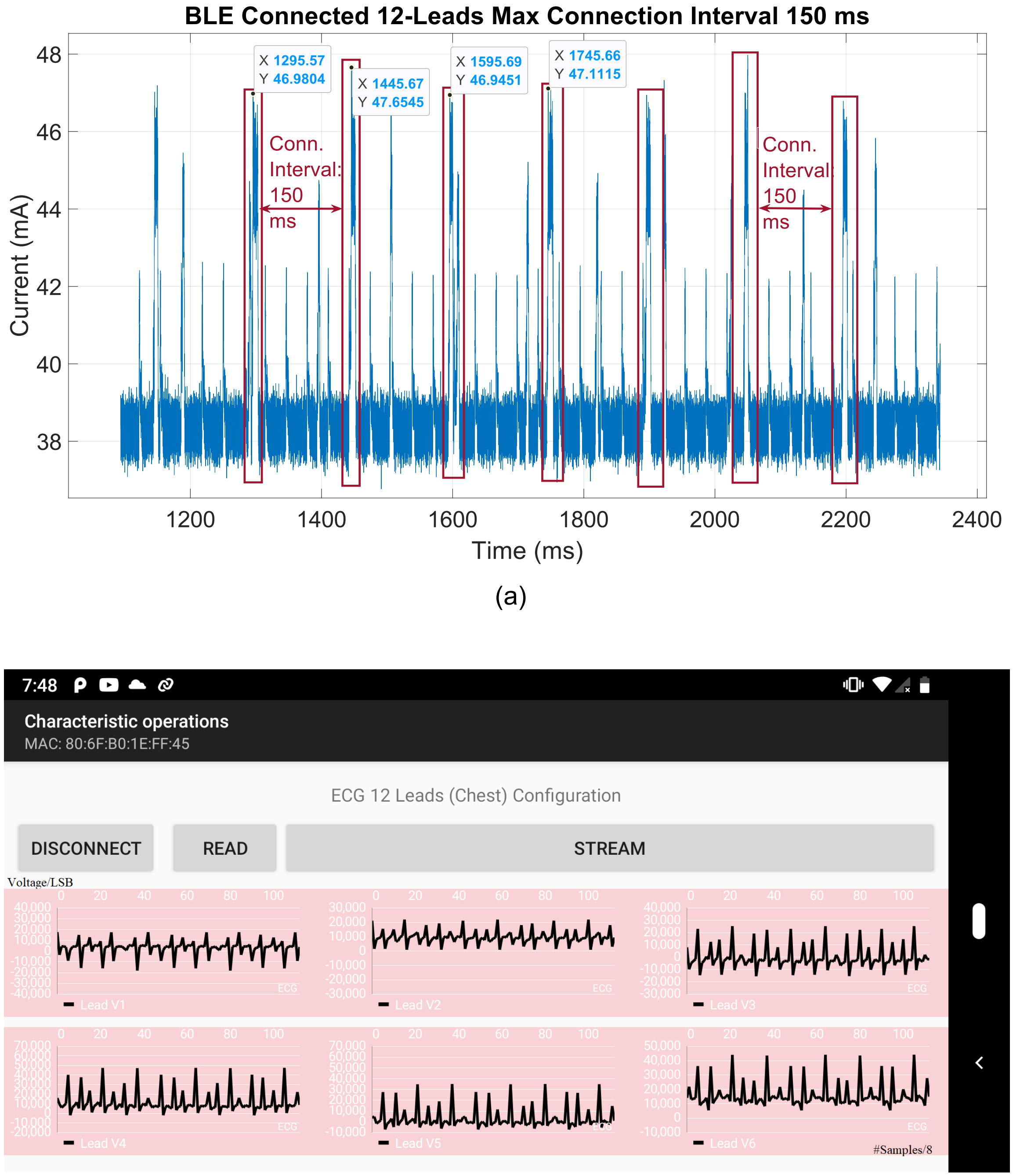


(a)



(b)

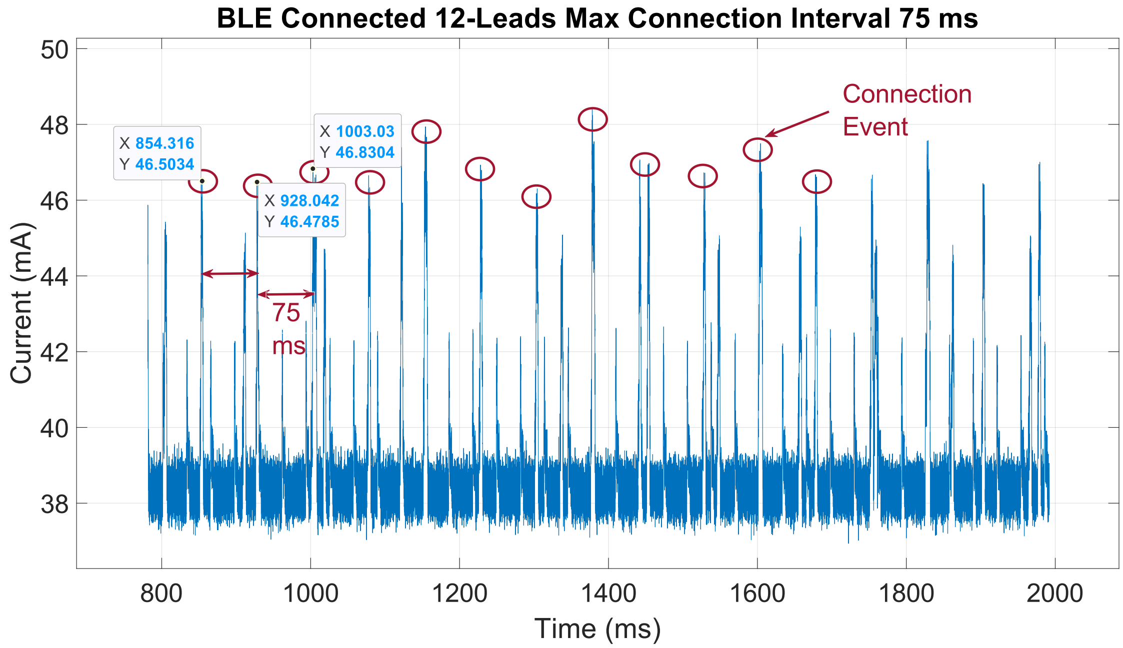
Figure 5.11. Experiment #3: Data transmission using BLE with a Maximum Connection Interval of 250 ms.



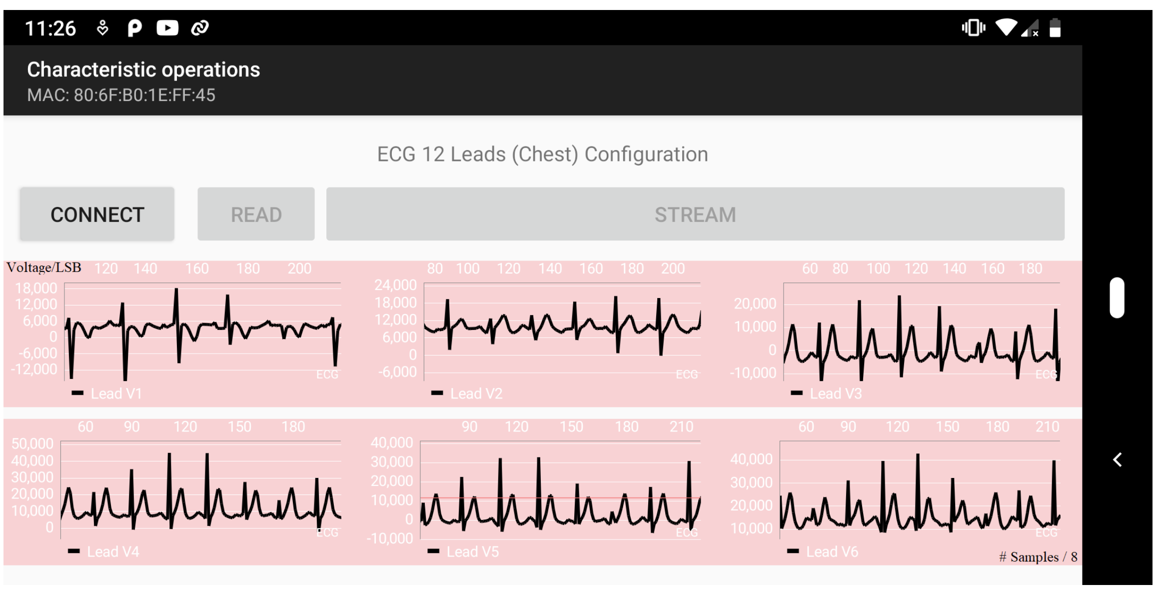
(a)

(b)

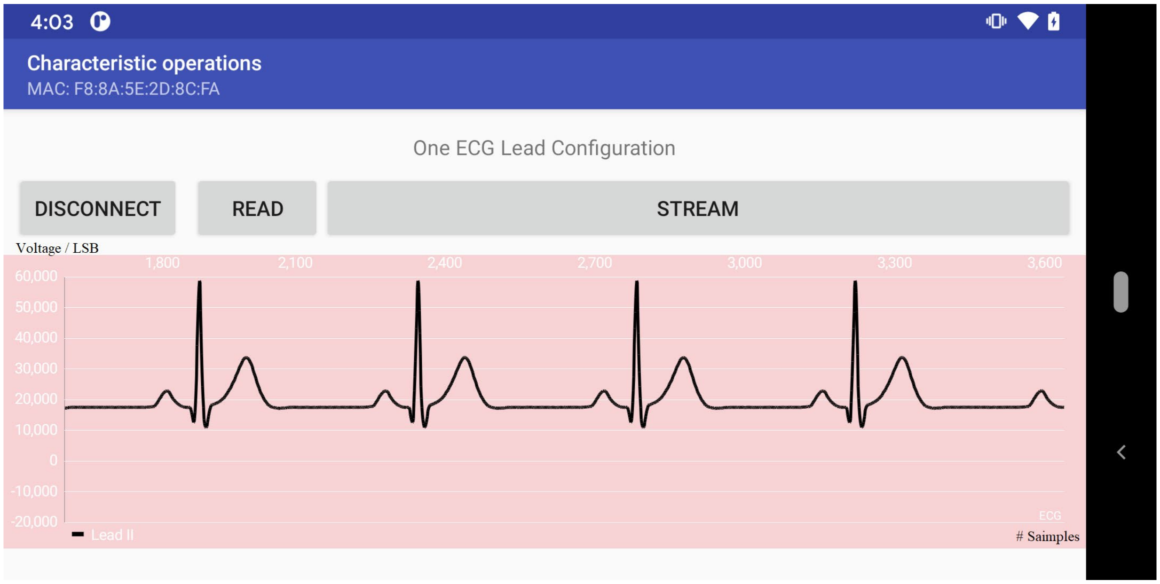
Figure 5.12. Experiment #4: Data transmission using BLE with a Maximum Connection Interval of 150 ms.


BLE Connected 12-Leads Max Connection Interval 75 ms
Current (mA)
Time (ms)
Connection Event
75 ms
X 854.316
Y 46.5034
X 928.042
Y 46.4785
X 1003.03
Y 46.8304

(a)


11:26
Characteristic operations
MAC: 80:6F:B0:1E:FF:45
ECG 12 Leads (Chest) Configuration
CONNECT
READ
STREAM
Voltage/LSB
Lead V1
Lead V2
Lead V3
Lead V4
Lead V5
Lead V6
Samples / 8

(b)



(c)

Figure 5.13. Experiment #5: Data transmission using BLE with a Maximum Connection Interval of 75 ms.

In the previous graphs, the value of the Least Significant Bit (LSB) is used to determine the voltage value of the acquired ECG signals. One LSB represents the voltage weight of one code (i.e., one digitized sample (24 bits) acquired by the ADS1298 chip). Accordingly, the full-scale range (FSR) of the ADC divided by the total number of codes yields the LSB size.

$$LSB = \frac{FSR}{(2^n - 1)}, \qquad where\ n\ is\ the\ ADC\ resolution\ (24\ bits)$$

In the ADS1298, the differential input voltage to each ADC can range from $-V_{REF}\ to\ +V_{REF}$Therefore, the full-scale range = 2 $X\ V_{REF}$.

$$LSB = 2\ X\ V_{REF} /\ Gain\ /\ 2^{24} - 1, \qquad where\ Gain\ \in [1, 2, 3, 5, 6, 8, 12]$$

Table 5-11. Experiment results of the data transmission over BLE on the ECG patch.

Experiment (10 seconds)	Maximum Current	Average Current	Energy	Sampling Rate
#1	~ 47 mA	36.63 mA	1209.152 mJ	35 SPS
#2	~ 47 mA	36.98 mA	1220.926 mJ	70 SPS
#3	~ 48 mA	36.97 mA	1220.252 mJ	147 SPS
#4	~ 49 mA	37.15 mA	1225.9 mJ	238 SPS
#5	~ 49 mA	38.14 mA	1227.71 mJ	383 SPS

The experiment results shown in Table 5-11 are evaluated with respect to the effective sampling rate of the ECG data by the ECG patch. The first three experiments showed a decrease in the current consumption. However, the sampling rate decreased significantly, negatively impacting the ECG monitoring process [94]. Moreover, the components (i.e., PQRST [7]) of the ECG signals were not detected in the first three experiments, as shown in Figures 9b, 10b and 11b. The PQRST components of an ECG signal are essential in diagnosing the signals and heart attack detection [7]. On the other hand, the last two experiments showed better performance and higher sampling rates for the ECG signals. Conclusively, the experiments showed significant changes in the effective sampling rates. However, the power consumption had a mostly negligible impact on the operation lifetime of the ECG patch. Therefore, stretching the connection interval of a BLE connection did not help with the total energy consumption as it did not lead to less current consumption. However, the sampling rate is hindered due to the connection interval stretching.

5.4.3 Data Logging & Storage Evaluation

The writing operation on an MMC is considered the most energy-expensive operation. The MMC is a strictly 3V device, and the current consumption can reach up to 100mA or more, necessitating an efficient 3V power supply for the card. The optimization in this phase involves the functions used to write data on the MMC. We use the Generic FAT Filesystem Module [95] as an API providing various filesystem functions to interact with the MMC on the ECG patch. The candidate function (i.e., f_printf()) is evaluated to find the optimum operation performance. The original implementation of the XBeats firmware used the f_printf() function to data to the MMC as floating-point decimals. The floating-point decimal takes from 1 to 31 characters for each floating-point number (e.g., each data point in an ECG signal is measured in millivolts). The accuracy of the ADS1298 is 24-bit. Therefore, a single ECG data point value would be similar to “0.001152849”. If we write the number using the floating-point decimal representation would take 11 characters which is equal to 11 bytes. Accordingly, the total bytes needed for one ECG sample (e.g., eight channels are activated) equals 11 bytes × 8 channels = 88 bytes, and 22 Kbytes at 250 SPS.

Alternatively, if we use the unsigned hexadecimal, which only takes 1 to 8 characters to write a respective ECG data point on the MMC. A single ECG data point is presented in the following format “0070DF“ (i.e., six characters in total). Then, the total bytes needed to write one ECG sample equals 6 bytes × 8 channels = 48 bytes, and 12 Kbytes at 250 SPS. Therefore, the transformation to the unsigned hexadecimal saves 45% of the data written to the MMC.

5.4.4 XBeats Battery Energy Consumption and Operation Period

We use the Energy Trace tool to determine the analytical energy consumption of the XBeats ECG patch over time. The Energy trace tool is used in the free-run mode, where the sampling frequency is approximately 4.2 kHz. The energy consumption evaluation tests performed on the ECG prototype assumed the device to be in the “continuous” operation mode. The continuous mode uses the standard 12-lead ECG data collection and transmits the collected signals over BLE in real-time to a smartphone. This means the MCU is always in active mode, and no power-saving protocols are applied other than the default settings on the MCU. Consequently, the analytical results show that the ECG patch can deliver a continuous real-time 12-lead ECG for approximately 37 hours using a rechargeable lithium-ion battery with a capacity of 2000 mAh; details are shown in Table 5-12.

Table 5-12. Analytical power consumption of the ECG patch hardware prototype.

Power Consumption Results	Value
Mean, Min, and Max	157.73 mW, 91.69 mW, and 364.133 mW
Average Voltage	3.3 V
Battery Capacity	2000 mAh
Total Operation time	1 Day, 13 h approximately

5.5 Discussions

We start by creating a benchmark for ECG data collection to evaluate the ECG data acquisition performed by the XBeats ECG patch concerning the correctness and quality of the acquired signals. Then, the three operation modes are evaluated regarding the useful acquisition rate. XBeats can perform standard 12-lead ECG testing while continuously streaming the collected signals over BLE at a rate of SPS. Furthermore, the device supports a maximum sampling rate of 480 SPS while operating under the offline mode when the communication module is turned off. Interestingly, when the communication module is turned on, but no connectivity is established, the sampling rate decreases to 370 SPS. The experiment showed a drop in the sampling since BLE enters the advertisement mode looking for potential connections when no device is connected to the ECG patch.

Moreover, we evaluate the proposed ECG signal classification based on six machine learning classifiers, and the Extra Trees achieves the maximum accuracy of 95.30%. This accuracy is accepted as an initial classification phase to support just-in-time patient/healthcare providers notifications while logging ECG data. This phase only classifies heart conditions into normal and abnormal classes for faster actions and low power consumption.

On the other hand, we study the power consumption behaviour of the XBeats ECG patch and apply the proposed optimization techniques to each component in the system. The applied optimization techniques are first applied to the data acquisition process and improve the related power consumption profile. The improvement included using a modified version of the firmware on the device, allowing the device to dynamically switch from the high consumption profile to the low power consumption profile. This operation yielded 8.2% in saved energy. Furthermore, we studied the power consumption profile of the BLE communication module. The results of the communication module had minor impacts on the power consumption profile, as it witnessed about a 1.6% reduction in the initially consumed power. The stretching of the connection interval led to a significant decrease in the sampling rate contrary to marginal optimization in the consumed energy that is almost negligible (i.e., 2 mJ in a 10-second interval). Lastly, we studied the patterns of ECG data collected and written on the external storage attached to the ECG patch. We optimized 54% of the data by writing the data using its hexadecimal representation instead of using a floating-point decimal format.

The proposed optimization methods assisted in analyzing and optimizing the power consumption in the XBeats ECG patch and extending the battery lifetime.

5.6 Summary

This chapter provides a systematic performance evaluation of the XBeats ECG patch. A benchmark for a standard 12-lead ECG data monitoring and acquisition is provided. The modes of operation implemented on the XBeats ECG patch are evaluated with respect to the expected sampling rates and power consumption. The experiments showed that the device could support a maximum sampling rate of 480 SPS while operating under the offline mode when the communication module is turned off and 370 SPS when the communication module is turned on. Then we evaluated the proposed ECG signal classification based on six machine learning classifiers, and the Extra Trees achieved a maximum accuracy of 95.30%. A systematic power consumption evaluation is provided to optimize the power consumption profile of the XBeats ECG patch concerning various scenarios and modes of operation. Since XBeats is a wearable device operated by a battery, power consumption profiling and optimization are essential to utilize the device and provide maximum operation time.

Chapter 6. Conclusions and Future Directions

Although the ECG test is a 100-year-old technology, it remains scientifically challenging and attractive to research to unleash the full potential of information technology and IoT in this domain. We demonstrate the need for a compact wearable ECG monitoring system from the literature. We introduce XBeats, a novel ECG patch for continuous real-time monitoring. The XBeats ECG patch supports dynamic modes of operations that are actively configured when the heart conditions of the patient change. The device carries out all primary operations: data acquisition, logging, and transmission at an acquisition rate of up to 480 samples per second with significantly low latency. The proposed framework integrates fog computing data analytics to perform binary ECG signal classification. The classification algorithm achieved a maximum detection accuracy of 95.30% based on the Extra Trees machine learning classifier. This accuracy is accepted in our proposed framework as an initial phase of classification to support in-time notifications to the patient/healthcare providers.

6.1 Conclusions

The XBeats RPM framework for real-time ECG monitoring and diagnosis demonstrates exemplary performance and can send immediate messages when irregular heartbeats are detected to patients or healthcare providers. It can also support long-term medical diagnosis for ECG signals in real-time. The results achieved in the prototype development allow us to conclude that high-quality real-time remote 12-lead ECG monitoring is achievable through our robust framework design and selected hardware components.

In Chapter 2, a detailed investigation is provided concerning efforts toward providing standard 12-lead ECG testing remotely while outlining characteristics, features, and requirements for having continuous remote ECG testing in real-time. We also categorized the presented solution in the literature according to the number of ECG leads offered by the acquisition devices. Besides, a dedicated section is added to review the existing commercial ECG monitoring devices while presenting the advantages and shortcomings of the provided solutions. We remarked that although many discussed solutions provide enough functionalities for ECG testing, they are accompanied by inherited system gaps that prevent a comprehensive framework for remote 12-lead ECG monitoring and diagnoses. Not to mention that a significant number of the presented literature focus on a limited number of ECG leads, up to a maximum of 5 leads compared to a standard 12-lead ECG.

In Chapter 3, the design and implementation of XBeats, a flexible 12-lead/Holter real-time ECG prototype. The design of the ECG patch utilizes BLE standardization in the ECG patch by creating custom profiles and services. Consequently, the services provided by XBeats are enabled through the integration of the XBeats custom ECG Profile for BLE. Furthermore, the ECG BLE profile gives the patient and the healthcare provider complete control over the XBeats ECG patch remotely, as they can control the modes of operation and the number of selected ECG leads. To that extent, XBeats incorporates three modes of operation to accommodate various use cases and health conditions, which allows healthcare providers to configure the device concerning their patients' heart conditions. Moreover, the MCU on the ECG patch utilizes the sensor controller addon to run the ECG data acquisition and logging tasks simultaneously besides the data transmission task. Accordingly, it allowed the MCU to carry out all primary operations: data acquisition, logging, and transmission at an acquisition rate of up to 480 samples per second with significantly low latency.

Furthermore, we present an AI-powered system for ECG signal classification based on machine learning and real-time streaming. We implemented several machine learning algorithms to classify and detect anomalous heart conditions, including Logistic Regression, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Extra Tree. Our findings suggest that Extra Trees outperforms other techniques with acceptable real-time performance. We trained the Extra Trees on a publicly available dataset to classify the signal into normal and abnormal categories. Consequently, we introduce systematic energy consumption profiling criteria for evaluating participating components in an RPM device. Each hardware component is isolated and evaluated individually to find power-intensive processes in the XBeats system, discover energy consumption patterns, and measure voltage, current, power, and energy consumption for a given period. The acquisition module power consumption is analyzed by studying the controlling parameters like the sampling frequency, the number of ECG leads and the operation period. Likewise, the power consumption behaviour of the communication module is analyzed to find the optimum configuration settings for the XBeats ECG patch to reduce power consumption and maintain the integrity of the collected ECG signals during data transmission.

Driven by the demand and importance of an efficient remote cardiovascular monitor for virtual care, Chapter 3 presents a framework that enables remote ECG testing and provides ubiquitous data access to patients and their healthcare providers. The framework utilizes XBeats, a patent-pending 12-lead data acquisition ECG patch for long-term cardiac monitoring and diagnoses. The XBeats framework provides a comprehensive RPM system for real-time ECG monitoring and data analytics. The framework gives an edge to healthcare providers to continuously monitor their patients remotely without requiring patients to visit hospitals or healthcare facilities. Moreover, through the event processing and data analytics technologies integrated into the framework, healthcare providers get prompt notifications in the event of irregular or abnormal heart conditions. Furthermore, the framework can dispatch an ambulance or a 911 if the system detects severe heart conditions like a heart attack. Accordingly, the XBeats ECG patch and the proposed RPM framework provide an end-to-end solution for long-term remote cardiac monitoring.

To reproduce the proposed XBeats framework, we present an implementation of the framework in Chapter 3. The framework is deployed as a use case application for remote ECG testing and diagnoses. At the same time, the framework integrates the latest technologies in distributed systems, containerized deployment, communication protocols, data streaming platforms, data access and storage. We use microservices and web services APIs to continuously integrate new features and services without disturbing the standard services of the framework. The standard services defined in the framework are data collection, streaming, stream processing, storage, reporting and visualization. Moreover, we use the MQTT protocol and REST web services to exchange data between participating parties and establish communication pipelines. Besides, the framework establishes direct communication with doctors or the healthcare provider through notifications when a patient is undergoing an abnormal medical condition so proper actions and emergency procedures can be activated to ensure the patient's well-being.

We list the objectives and benefits of integrating RPM systems into the existing healthcare infrastructure and how it facilitates healthcare providers to accommodate the needs of their patients in real-time. The new e-health and telemedicine era emphasizes the importance of RPM when hospitals strive to provide patients with essential medical needs. The key objective of this chapter is the design of a comprehensive framework for RPM, focusing on the system integration of the enabling technologies and software. The distributed nature of the design architecture of the framework facilitates horizontal scalability to accommodate the increasing growth of connected RPM devices and services.

In Chapter 5, we evaluate the proposed ECG signal classification based on the Extra Trees machine learning classifier, which achieved a maximum accuracy of 95.30%. This accuracy is accepted in the XBeats RPM framework as an initial classification phase to support just-in-time notifications to the patient/healthcare providers while logging ECG data. This phase only classifies heart conditions into normal and abnormal classes for faster actions and low power consumption. However, due to time and memory constraints on the MCU, the classification technique significantly disrupted the primary operations on the ECG patch. Therefore, we deployed the proposed classification component on an edge device using Raspberry bi 3 B+. The average processing time for ECG signal detection is 0.29 seconds. If an abnormal heart condition is detected, a message is sent out immediately to caregivers in a range of 0.57 to 0.77 seconds, which is quick enough for healthcare providers to take necessary actions.

The results show that optimizing the data acquisition process saves 8.2% compared to the original power consumption and 1.62% in data transmission over BLE, thus extending the device's lifetime. Also, we optimize the data logging operation to save 54% of data initially written to an external drive. Lastly, the analytical energy results yield up to 37 hours of continuous 12-lead ECG streaming using a 2000 mAh rechargeable lithium-ion battery. The results achieved in the prototype development allow us to conclude that high-quality real-time remote 12-

lead ECG monitoring is achievable through our robust framework design and selected hardware components. At the same time, there remain a few open issues due to the significance of remote ECG in saving patients with chronic heart diseases. Some of those issues can be extremely challenging to address, such as privacy and security, due to performing almost all operations remotely. Other issues, such as handling design diversity concerning the modularity of the XBeats ECG patch, in accommodating the needs of the patients as they wear the device daily. Another issue concerns the battery lifetime and providing reliable remote ECG testing for extended periods.

6.2 Possible Future Directions

This thesis introduces the XBeats ECG patch architecture and designs by presenting a working hardware prototype for successful remote ECG testing and diagnoses. Although, the XBeats ECG patch hardware prototype is still in the early stages of research. The experiments showed promising results concerning the power consumption of XBeats using li-ion batteries which constitutes potential challenges in charging the replacing the batteries. Therefore, as a future direction, we are looking into various sets of batteries like coin or Lithium polymer batteries. Similarly, investigate the ability to integrate a hot-swap feature while changing the batteries, so the primary operations running on the ECG patch are not interrupted.

On the other hand, privacy and security are significant roadblocks to successful integrations of the XBeats RPM framework since the current design considered privacy and security topics of their own. Privacy is a serious topic, especially in the healthcare domain; therefore, we consider enforcing necessary privacy policies to match existing standards and requirements by healthcare providers and involved authorities. The current design of the XBeats framework utilizes security and privacy-ready components to facilitate future research and developments in handling privacy concerns and better securing the framework.

Finally, we plan to extend the proposed framework to incorporate a broader range of ECG data analytics and deep learning tools at the backend for various abnormal heart conditions to support better diagnoses. Moreover, we consider modifying the classification technique to work with aggregated ECG data in real-time as they arrive with predefined windows for the aggregation process. This way, the framework shall avoid false positives regarding the patient's heart conditions when classifying the heartbeats individually.

References

[1] Centers for Disease Control and Prevention (Accessed: 2022, September), Heart Disease Facts. [Online]. https://www.cdc.gov/heartdisease/facts.htm.

[2] World Health Organization (Accessed: 2022, September), Cardiovascular diseases. [Online] https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[3] Government of Canada (Accessed: 2022, September), Heart Disease in Canada. [Online]. https://www.canada.ca/en/public-health/services/publications/diseases-conditions/heart-disease-canada.html

[4] Woods, M., & Miklencicova, R. (2021). Digital Epidemiological Surveillance, Smart Telemedicine Diagnosis Systems, and Machine Learning-based Real-Time Data Sensing and Processing in COVID-19 Remote Patient Monitoring. American Journal of Medical Research, 8(2), 65-78.

[5] Rogers, R. (2021). Internet of Things-based Smart Healthcare Systems, Wireless Connected Devices, and Body Sensor Networks in COVID-19 Remote Patient Monitoring. American Journal of Medical Research, 8(1), 71-81.

[6] Rios A. (Accessed: 2022, September) Sony's mSafety™. [Online]. https://bwnews.pr/3f0L7xt

[7] Ashley, E. A., & Niebauer, J. (2004). Cardiology explained, 15-33.

[8] Galli, A., Ambrosini, F., & Lombardi, F. (2016). Holter monitoring and loop recorders: from research to clinical practice. Arrhythmia & electrophysiology review, 5(2), 136.

[9] Ramkumar, M., Ganesh Babu, C., Manjunathan, A., Udhayanan, S., Mathankumar, M., & Sarath Kumar, R. (2021). A graphical user interface based heart rate monitoring process and detection of pqrst peaks from ecg signal. In Inventive Computation and Information Technologies (pp. 481-496). Springer, Singapore.

[10] Chua, S. K., Chen, L. C., Lien, L. M., Lo, H. M., Liao, Z. Y., Chao, S. P. & Chiu, C. Z. (2020). Comparison of arrhythmia detection by 24-hour Holter and 14-day continuous electrocardiography patch monitoring. Acta Cardiologica Sinica, 36(3), 251.

[11] Casale, P. N., Vyavahare, M., Coyne, S., Kronish, I., Greenwald, P., Ye, S. & Fleischut, P. M. (2021). The promise of remote patient monitoring: lessons learned during the COVID-19 surge in New York City. American Journal of Medical Quality, 36(3), 139.

[12] Kichloo, A., Albosta, M., Dettloff, K., Wani, F., El-Amir, Z., Singh, J. & Chugh, S. (2020). Telemedicine, the current COVID-19 pandemic and the future: a narrative review and perspectives moving forward in the USA. Family medicine and community health, 8(3).

[13] Mubarik, A., & Iqbal, A. M. (Accessed: 2022, September) Holter monitor. In StatPearls [Online]. http://www.ncbi.nlm.nih.gov/books/NBK538203/

[14] Serhani, M. A., T. El Kassabi, H., Ismail, H., & Nujum Navaz, A. (2020). ECG monitoring systems: Review, architecture, processes, and key challenges. Sensors, 20(6), 1796.

[15] Alshamrani, M. (2021). IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey. Journal of King Saud University-Computer and Information Sciences.

[16] (Accessed: 2022, September) Apache Kafka: an open-source distributed event streaming platform. [Online]. https://kafka.apache.org/.

[17] Faruk, N., Abdulkarim, A., Emmanuel, I., Folawiyo, Y. Y., Adewole, K. S., Mojeed, H. A. & Katibi, I. A. (2021). A comprehensive survey on low-cost ECG acquisition systems: Advances on design specifications, challenges and future direction. Biocybernetics and Biomedical Engineering, 41(2), 474-502.

[18] Kwon, S., Lee, S. R., Choi, E. K., Ahn, H. J., Song, H. S., Lee, Y. S., & Oh, S. (2021). Validation of adhesive single-lead ECG device compared with holter monitoring among non-atrial fibrillation patients. Sensors, 21(9), 3122.

[19] Sahoo, P. K., Thakkar, H. K., Lin, W. Y., Chang, P. C., & Lee, M. Y. (2018). On the design of an efficient cardiac health monitoring system through combined analysis of ECG and SCG signals. Sensors, 18(2), 379.

[20] Sahoo, P. K., Thakkar, H. K., & Lee, M. Y. (2017). A cardiac early warning system with multi channel SCG and ECG monitoring for mobile health. Sensors, 17(4), 711.

[21] Klum, M., Leib, F., Oberschelp, C., Martens, D., Pielmus, A. G., Tigges, T. & Orglmeister, R. (2019, July). Wearable multimodal stethoscope patch for wireless biosignal acquisition and long-term auscultation. In 2019 41st Annual

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 5781-5785). IEEE.

[22] Rashkovska, A., Depolli, M., Tomašić, I., Avbelj, V., & Trobec, R. (2020). Medical-grade ECG sensor for long-term monitoring. Sensors, 20(6), 1695.

[23] Rincon, J. A., Guerra-Ojeda, S., Carrascosa, C., & Julian, V. (2020). An IoT and fog computing-based monitoring system for cardiovascular patients with automatic ECG classification using deep neural networks. Sensors, 20(24), 7353.

[24] Ahsanuzzaman, S. M., Ahmed, T., & Rahman, M. A. (2020, June). Low cost, portable ECG monitoring and alarming system based on deep learning. In 2020 IEEE Region 10 Symposium (TENSYMP) (pp. 316-319). IEEE.

[25] Gao, Z., Wu, J., Zhou, J., Jiang, W., & Feng, L. (2012, May). Design of ECG signal acquisition and processing system. In 2012 international conference on biomedical engineering and biotechnology (pp. 762-764). IEEE.

[26] Spanò, E., Di Pascoli, S., & Iannaccone, G. (2016). Low-power wearable ECG monitoring system for multiple-patient remote monitoring. IEEE Sensors Journal, 16(13), 5452-5462.

[27] Masihi, S., Panahi, M., Maddipatla, D., Hanson, A. J., Fenech, S., Bonek, L. & Atashbar, M. Z. (2021). Development of a flexible wireless ECG monitoring device with dry fabric electrodes for wearable applications. IEEE Sensors Journal.

[28] Klum, M., Urban, M., Tigges, T., Pielmus, A. G., Feldheiser, A., Schmitt, T., & Orglmeister, R. (2020). Wearable cardiorespiratory monitoring employing a multimodal digital patch stethoscope: estimation of ECG, PEP, LVET and respiration using a 55 mm single-lead ECG and phonocardiogram. Sensors, 20(7), 2033.

[29] Yuan, L., Yuan, Y., Zhou, Z., Bai, Y., & Wu, S. (2019). A fetal ECG monitoring system based on the android smartphone. Sensors, 19(3), 446.

[30] Wang, L. H., Zhang, W., Guan, M. H., Jiang, S. Y., Fan, M. H., Abu, P. A. R., & Chen, S. L. (2019). A low-power high-data-transmission multi-lead ECG acquisition sensor system. Sensors, 19(22), 4996.

[31] Abtahi, F., Snäll, J., Aslamy, B., Abtahi, S., Seoane, F., & Lindecrantz, K. (2014). Biosignal PI, an affordable open-source ECG and respiration measurement system. Sensors, 15(1), 93-109.

[32] Ozkan, H., Ozhan, O., Karadana, Y., Gulcu, M., Macit, S., & Husain, F. (2019). A portable wearable tele-ECG monitoring system. IEEE Transactions on Instrumentation and Measurement, 69(1), 173-182.

[33] Pineda-López, F., Martínez-Fernández, A., Rojo-Álvarez, J. L., García-Alberola, A., & Blanco-Velasco, M. (2018). A flexible 12-Lead/Holter device with compression capabilities for low-bandwidth mobile-ECG telemedicine applications. Sensors, 18(11), 3773.

[34] Boehm, A., Yu, X., Neu, W., Leonhardt, S., & Teichmann, D. (2016). A novel 12-lead ECG T-shirt with active electrodes. Electronics, 5(4), 75.

[35] Uktveris, T., & Jusas, V. (2018). Development of a modular board for EEG signal acquisition. Sensors, 18(7), 2140.

[36] Medtronic (Accessed: 2022, September) Cardiac Monitors. [Online]. https://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/cardiac-monitors.html

[37] Zio Monitoring (Accessed: 2022, September) Easy EKG Interpretation: Zio XT. [Online] https://www.irhythmtech.com/products-services/zio-xt.

[38] Philips Healthcare (Accessed: 2022, September) Wearable biosensor Wireless remote sensing device. [Online]. https://www.philips.ie/healthcare/product/HC989803196871/wearable-biosensor-wireless-remote-sensing-device.

[39] (Accessed: 2022, September) Savvy ECG. [Online]. http://www.savvy.si/en/Savvy_1/ (accessed Jul. 25, 2022).

[40] Jortveit, J., & Fensli, R. (2022). Remote ECG monitoring by ECG247 smart heart sensor. International Journal of Telemedicine and Applications, 2022.

[41] BioTelemetry (Accessed: 2022, September) ePatch. [Online]. https://www.gobio.com/epatch/

[42] QT Medical (Accessed: 2022, September) Portable 12-lead ECG Machine, Medical ECG Device. [Online]. https://www.qtmedical.com/

[43] QardioCore (Accessed: 2022, September) Smart Wearable ECG EKG Monitor. [Online] https://www.getqardio.com/en/qardiocore-wearable-ecg-ekg-monitor-iphone/

[44] Kaur, J., & Mann, K. S. (2017, December). AI based healthcare platform for real time, predictive and prescriptive analytics using reactive programming. In Journal of Physics: Conference Series (Vol. 933, No. 1, p. 012010). IOP Publishing.

[45] Sandha, S. S., Kachuee, M., & Darabi, S. (2017). Complex event processing of health data in real-time to predict heart failure risk and stress. arXiv preprint arXiv:1707.04364.

[46] Pomprapa, A., Sayani, M. S., Anwar, T., Stollenwerk, A., Kowalewski, S., von Platen, P., & Leonhardt, S. (2018). Apnea Detection in a Contactless Multisensor System using Deep Learning Algorithm. XIII RGC, 86.

[47] Padmavathi, S. (2020). A fog-based approach for real-time analytics of IoT-enabled healthcare. In Internet of Things Use Cases for the Healthcare Industry (pp. 43-66). Springer, Cham.

[48] Yeh, L. R., Chen, W. C., Chan, H. Y., Lu, N. H., Wang, C. Y., Twan, W. H. & Chen, T. B. (2021). Integrating ECG monitoring and classification via IoT and deep neural networks. Biosensors, 11(6), 188.

[49] Alfian, G., Syafrudin, M., Ijaz, M. F., Syaekhoni, M. A., Fitriyani, N. L., & Rhee, J. (2018). A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing. Sensors, 18(7), 2183.

[50] Desai, P., Sheth, A., & Anantharam, P. (2015, June). Semantic gateway as a service architecture for iot interoperability. In 2015 IEEE International Conference on Mobile Services (pp. 313-319). IEEE.

[51] Nguyen, Q. H., Dang, Q., Pham, C., Nguyen, T. D., Nguyen, H., Setty, A., & Le, T. Q. (2020, December). Developing an Architecture for IoT Interoperability in Healthcare: A Case Study of Real-time SpO2 Signal Monitoring and Analysis. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 3394-3403). IEEE.

[52] Lin, S. W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R. & Crawford, M. (2019). The Industrial Internet of Things Volume G1: Reference Architecture, 2017. Industrial Internet Consortium.

[53] ThingsBoard (Accessed: 2022, September) ThingsBoard Professional Edition Documentation. [Online]. https://thingsboard.io/docs/pe/

[54] Śmigiel, S., Pałczyński, K., & Ledziński, D. (2021). Deep Learning Techniques in the Classification of ECG Signals Using R-Peak Detection Based on the PTB-XL Dataset. Sensors, 21(24), 8174.

[55] Pałczyński, K., Śmigiel, S., Ledziński, D., & Bujnowski, S. (2022). Study of the Few-Shot Learning for ECG Classification Based on the PTB-XL Dataset. Sensors, 22(3), 904.

[56] P. Wagner *et al.*, "PTB-XL, a large publicly available electrocardiography dataset," *Scientific Data*, vol. 7, no. 1, p. 154, Dec. 2020, doi: 10.1038/s41597-020-0495-6.

[57] Strodthoff, N., Wagner, P., Schaeffter, T., & Samek, W. (2020). Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. IEEE Journal of Biomedical and Health Informatics, 25(5), 1519-1528.

[58] Wu, Y., Yang, F., Liu, Y., Zha, X., & Yuan, S. (2018). A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. arXiv preprint arXiv:1810.07088.

[59] Jaffe, A. S. (2013). Third universal definition of myocardial infarction. Clinical biochemistry, 46(1-2), 1-4.

[60] Albahri, O. S., Albahri, A. S., Mohammed, K. I., Zaidan, A. A., Zaidan, B. B., Hashim, M., & Salman, O. H. (2018). Systematic review of real-time remote health monitoring system in triage and priority-based sensor technology: Taxonomy, open challenges, motivation and recommendations. Journal of medical systems, 42(5), 1-27.

[61] Liu, Y., Wang, H., Zhao, W., Zhang, M., Qin, H., & Xie, Y. (2018). Flexible, stretchable sensors for wearable health monitoring: sensing mechanisms, materials, fabrication strategies and features. Sensors, 18(2), 645.

[62] Malasinghe, L. P., Ramzan, N., & Dahal, K. (2019). Remote patient monitoring: a comprehensive study. Journal of Ambient Intelligence and Humanized Computing, 10(1), 57-76.

[63] El-Rashidy, N., El-Sappagh, S., Islam, S. R., M. El-Bakry, H., & Abdelrazek, S. (2021). Mobile health in remote patient monitoring for chronic diseases: Principles, trends, and challenges. Diagnostics, 11(4), 607.

[64] Barrett, P. M., Komatireddy, R., Haaser, S., Topol, S., Sheard, J., Encinas, J. & Topol, E. J. (2014). Comparison of 24-hour Holter monitoring with 14-day

novel adhesive patch electrocardiographic monitoring. The American journal of medicine, 127(1), 95-e11.

[65] Chen, C. L., & Chuang, C. T. (2017). A QRS detection and R point recognition method for wearable single-lead ECG devices. Sensors, 17(9), 1969.

[66] Texas Instruments (Accessed: 2022, September) ADS1298RECGFE-PDK Evaluation board. [Online]. https://www.ti.com/tool/ADS1298RECGFE-PDK

[67] Texas Instruments (Accessed: 2022, September) TI-RTOS-MCU Operating system. [Online]. https://www.ti.com/tool/TI-RTOS-MCU

[68] Townsend, K., Cufí, C., & Davidson, R. (2014). Getting started with Bluetooth low energy: tools and techniques for low-power networking. " O'Reilly Media, Inc.".

[69] Bluetooth Special Interest Group (Accessed: 2022, September) Heart rate profile. [Online]. http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=239865.

[70] Bluetooth Special Interest Group (Accessed: 2022, September) GATT Specifications. [Online]. https://www.bluetooth.com/specifications/gatt/

[71] Townsend, K., Cufí, C., & Davidson, R. (2014). Getting started with Bluetooth low energy: tools and techniques for low-power networking. " O'Reilly Media, Inc.".

[72] Yiadom, M. Y. A., Baugh, C. W., McWade, C. M., Liu, X., Song, K. J., Patterson, B. W. & Storrow, A. B. (2017). Performance of emergency department screening criteria for an early ECG to identify ST-segment elevation myocardial infarction. Journal of the American Heart Association, 6(3), e003528.

[73] Pan, J., & Tompkins, W. J. (1985). A real-time QRS detection algorithm. IEEE transactions on biomedical engineering, (3), 230-236.

[74] Saenz-Cogollo, J. F., & Agelli, M. (2020). Investigating feature selection and random forests for inter-patient heartbeat classification. Algorithms, 13(4), 75.

[75] Texas Instruments (Accessed: 2022, September) ADS1298 datasheet. [Online]. https://www.ti.com/product/ADS1298

[76] De Guzman, K. R., Snoswell, C. L., Taylor, M. L., Gray, L. C., & Caffery, L. J. (2022). Economic Evaluations of Remote Patient Monitoring for Chronic Disease: A Systematic Review. Value in Health.

[77] Mohammed, K. I., Zaidan, A. A., Zaidan, B. B., Albahri, O. S., Alsalem, M. A., Albahri, A. S. & Hashim, M. (2019). Real-time remote-health monitoring systems: a review on patients prioritisation for multiple-chronic diseases, taxonomy analysis, concerns and solution procedure. Journal of medical systems, 43(7), 1-21.

[78] Ullah, F., Haq, H. U., Khan, J., Safeer, A. A., Asif, U., & Lee, S. (2021). Wearable iots and geo-fencing based framework for COVID-19 remote patient health monitoring and quarantine management to control the pandemic. Electronics, 10(16), 2035.

[79] Balakrishnan, L. (2021, May). An Internet of Things (IoT) Based Intelligent Framework for Healthcare–A Survey. In 2021 3rd International Conference on Signal Processing and Communication (ICPSC) (pp. 243-251). IEEE.

[80] (Accessed: 2022, September) MQTT - The Standard for IoT Messaging. [Online]. https://mqtt.org/

[81] Sharvari, T., & Sowmya Nag, K. (2019). A study on modern messaging systems-kafka, rabbitmq and nats streaming. CoRR abs/1912.03715.

[82] De Lauretis, L. (2019, October). From monolithic architecture to microservices architecture. In 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 93-96). IEEE.

[83] Gallaugher, J. M., & Ramanathan, S. C. (1996). Choosing a client/server architecture: a comparison of two-and three-tier systems. Information Systems Management, 13(2), 7-13.

[84] Zaragoza, P., Seriai, A. D., Seriai, A., Shatnawi, A., Bouziane, H. L., & Derras, M. (2022). Materializing Microservice-oriented Architecture from Monolithic Object-oriented Source Code. In International Conference on Software Technologies (pp. 143-168). Springer, Cham.

[85] (Accessed: 2022, September) ksqlDB - event streaming database. [Online]. https://docs.ksqldb.io/en/latest/

[86] Badr, A., Badawi, A., Rashwan, A., & Elgazzar, K. (2022). XBeats: A Real-Time Electrocardiogram Monitoring and Analysis System. Signals, 3(2), 189-208.

[87] HiveMQ (Accessed: 2022, September) Enterprise ready MQTT. [Online]. https://www.hivemq.com/

[88] Confluent (Accessed: 2022, September) Apache Kafka & Event Streaming Platform. [Online]. https://www.confluent.io/product/confluent-platform

[89] (Accessed: 2022, September) MongoDB Atlas. [Online]. https://docs.atlas.mongodb.com/getting-started

[90] S. H. AnnaPovzner (Accessed: 2022, September) Benchmark Your Dedicated Apache Kafka® Cluster on Confluent Cloud. [Online]. https://assets.confluent.io/m/2d7c883a8aa6a71d/original/20200501-WP-Benchmark_Your_Dedicated_Apache_Kafka_Cluster_on_Confluent_Cloud.pdf

[91] Texas Instruments (Accessed: 2022, September) LE Data Length Extension. [Online]. http://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.35.00.33/exports/docs/ble5stack/ble_user_guide/html/ble-stack/data-length-extensions.html

[92] HE Instruments (Accessed: 2022, September) Patient Simulators. [Online]. https://www.heinstruments.com

[93] Bluetooth® Technology (Accessed: 2022, September) Bluetooth Low Energy - It Starts with Advertising. [Online]. https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising

[94] Bailey, J. J., Berson, A. S., Garson Jr, A., Horan, L. G., Macfarlane, P. W., Mortara, D. W., & Zywietz, C. (1990). Recommendations for standardization and specifications in automated electrocardiography: bandwidth and digital signal processing. A report for health professionals by an ad hoc writing group of the Committee on Electrocardiography and Cardiac Electrophysiology of the Council on Clinical Cardiology, American Heart Association. Circulation, 81(2), 730-739.

[95] (Accessed: 2022, September) FatFs - Generic FAT Filesystem Module. [Online]. http://elm-chan.org/fsw/ff/00index_e.html.