Politecnico di Torino

Politecnico di Torino

Master's Degree in Biomedical Engineering

A.a. 2025/2026

Graduation Session March 2026

Development and Implementation of an Audio Driver for a Multisensor Wearable Medical Device

Supervisors:

Prof. Luca Mesin

Ph.D. Paolo Motto Ros

Candidate:

María Sol Liendo

Abstract

Wearable multisensor devices enable continuous cardiac monitoring, but high-fidelity audio transmission poses challenges due to high data-rate requirements and low-power Bluetooth constraints.

This thesis presents the development and implementation of an embedded audio driver for real-time cardiac auscultation on an STM32U545 microcontroller. The system integrates a MEMS microphone through an I²S interface using DMA and double-buffering for low-latency and deterministic acquisition. A digital IIR filter, designed in MATLAB and implemented on the microcontroller using a CMSIS-DSP biquad cascade structure, preprocesses phonocardiographic (PCG) signals to enhance clinically relevant frequency bands (20–400 Hz for S1/S2 sounds and murmurs) with controlled gain while attenuating undesirable noise components.

Audio compression is performed using the LC3 codec (Bluetooth LE Audio standard) at 32 kbps, operating in real time with 10 ms frames, achieving ∼12:1 reduction (480 → 40 bytes/frame) while maintaining signal fidelity with SNR $>$ 25 dB and PRD $<$ 7% under controlled conditions. Bitrate-dependent characterization was conducted to evaluate compression efficiency, spectral distortion, and cardiac-band energy preservation.

Key developments include: (1) real-time audio pipeline implementation combining acquisition, digital filtering, and LC3 encoding with low computational overhead; (2) objective quality validation using SNR, PRD, spectral distortion, and band-specific energy preservation metrics (>95% retention within cardiac bands at 32 kbps); (3) experimental characterization through filter validation, codec performance analysis, and end-to-end acoustic testing.

The results demonstrate the feasibility of integrating an architecture capable of diagnostic-quality cardiac audio processing within a wearable multisensor platform, enabling efficient wireless transmission within BLE bandwidth constraints.

Keywords: cardiac auscultation, MEMS microphone, LC3 compression, Bluetooth LE Audio, embedded audio driver, wearable medical device.

Acknowledgements

Se mi guardassi indietro, a sei anni fa, non potrei nemmeno immaginare di essere dove sono ora. Questo lavoro riflette lo sforzo messo in questi anni di studio, che includono l'aver imparato un'altra lingua e il trovarsi a migliaia di chilometri di distanza da casa. Ma tutto questo non l'ho fatto da sola, per cui ci tengo a ringraziare di cuore tutte le persone che sono state presenti e che hanno reso possibile questo traguardo.

Vorrei esprimere la mia più profonda gratitudine ai miei relatori, Luca Mesin e Paolo Motto Ros, per aver seguito da vicino l'intero progetto e per i loro consigli. Un ringraziamento speciale va anche a Teoresi Group per avermi dato l'opportunità di partecipare a questo progetto. Porto a casa tantissime cose imparate. Grazie per lo spazio, la fiducia riposta in me, l'accoglienza, il tempo e la dedizione. In particolar modo, voglio ringraziare Guido Tiberi, che mi ha seguita ed è stato al mio fianco in ogni fase del lavoro, insegnandomi molto, valorizzando il mio contributo e portando tanta pazienza. Voglio inoltre ringraziare la mia università d'origine, la FCEFyN, e il Politecnico di Torino per aver reso possibile questo scambio di doppia laurea.

A tutte le persone meravigliose che ho conosciuto in questo periodo, che dal nulla sono entrate nella mia vita diventandone una parte fondamentale, e che spero vi rimangano per sempre. Un grazie speciale va ad Alessandro, che è stato al mio fianco in quest'ultimo capitolo della mia vita. Alle ragazze del CUS, che mi hanno subito accolta quando cercavo una squadra: insieme abbiamo creato un bellissimo spazio per staccare la spina quando ne avevo più bisogno, facendomi rivivere l'essenza dello sport e lo spirito di squadra con cui sono cresciuta.

A mi familia, por haberme bancado en absolutamente todo. Son mis pilares en el mundo; gracias por estar siempre cerquita y compartiendo cada logro, aunque físicamente estemos lejísimos. Gracias a ustedes soy la persona que soy ahora. A mis amigos del otro lado del charco, que también siempre estuvieron ahí apoyando, les estoy inmensamente agradecida por su compañía, son muy importantes para mí. Ustedes saben cuánto me cuesta terminar las cosas y cerrar capítulos. Imagínense ahora, que termina una de las etapas más maravillosas que pude haber vivido y que fue así porque los tuve acompañándome a cada paso. Pero mirándole el lado

positivo, esto no se termina, se sigue para adelante, con fuerza y ganas de descubrir qué nos espera en el futuro. Sea como sea, y esté donde esté, me los llevo a todos en el corazón conmigo siempre.

A las camadas Torino-FCEFyN, por haberse convertido en una segunda familia estando tan lejos de casa, y muy especialmente a Meli y a Sofi que ahora ya las considero prácticamente hermanas.

Creo que puedo decir que fueron los mejores años de mi vida, y lo haría mil veces más.

Table of Contents

List of Tables

List of Figures

Glossary

ANC	Active Noise Cancellation.
BLE	Bluetooth Low Energy.
CMSIS	Cortex Microcontroller Software Interface Standard.
CWT	Continuous Wavelet Transform.
DMA	Direct Memory Access.
DSP	Digital Signal Processing.
FIR	Finite Impulse Response.
FPU	Floating-Point Unit.
HAL	Hardware Abstraction Layer.
HR	Heart Rate.
HS	Heart Sounds.
I^2S	Inter-Integrated Circuit Sound.
IIR	Infinite Impulse Response.
LC3	Low Complexity Communication Codec.
LD-MDCT	Low-Delay Modified Discrete Cosine Transform.
LSD	Log-Spectral Distance.
MCU	Microcontroller Unit.
MDCT	Modified Discrete Cosine Transform.
MEMS	Micro-ElectroMechanical System.
PCG	Phonocardiogram.
PRD	Percentage Root-mean-square Difference.
PSD	Power Spectral Density.

PSNR	Peak Signal-to-Noise Ratio.
RMSE	Root Mean Square Error.
RMSSD	Root Mean Square of Successive Differences.
RTF	Real-Time Factor.
RZC	Ratio of Zero Crossings.
SAI	Serial Audio Interface.
SD	Spectral Distortion.
SNR	Signal-to-Noise Ratio.
SOS	Second-Order Sections.
SRAM	Static Random-Access Memory.
STFT	Short-Time Fourier Transform.
SWV	Serial Wire Viewer.

Chapter 1

Introduction

1.1 Context and Motivation

This thesis is developed within the context of *REMEDY* (*Remote personalized Modular monitoring for management of patients with non-communicable Diseases to improve physiological and social recovery*), a project led by Teoresi MedTech. The objective of REMEDY is the development of a multisensor wearable medical device designed for complex and elderly patients. Placed directly on the patient's chest, this device continuously acquires vital biometric data, including real-time cardiac auscultation (Phonocardiography or PCG), and transmits it via Bluetooth to a smartphone gateway for remote clinical analysis.

1.2 Problem Statement

While continuous cardiac auscultation provides invaluable diagnostic information, it introduces significant engineering challenges in a wearable context. Uncompressed audio data, even when sampled at the lower frequencies typical of cardiac bandwidths, generates a continuous high-bitrate stream.

For a battery-powered device communicating over Bluetooth Low Energy (BLE), transmitting raw audio creates a severe bottleneck. The high data volume rapidly depletes the battery, exceeds the reliable throughput limits of the BLE protocol, and leaves little bandwidth for the other vital sensors integrated into the REMEDY device. Furthermore, ambulatory PCG signals are highly susceptible to environmental noise and motion artifacts, requiring immediate conditioning before transmission.

Addressing the trade-off between diagnostic audio fidelity, strict energy constraints, and limited wireless bandwidth is the central challenge of this work.

1.3 Goal and Objectives

The primary goal of this thesis is the design and embedded implementation of a real-time, low-power audio driver for the REMEDY multisensor wearable device. To overcome the bandwidth and energy limitations of continuous auscultation, this work focuses on integrating the *Low Complexity Communication Codec (LC3)* directly on the microcontroller to reduce the amount of data to be sent.

To achieve this overarching goal, the following specific objectives were established:

- *Embedded Audio Acquisition:* Implement a reliable data pipeline using the I^2S protocol and Direct Memory Access (DMA) to capture high-quality digital audio from an on-board MEMS microphone without overloading the CPU.
- *Real-Time Signal Conditioning:* Design and deploy a lightweight digital filtering strategy (IIR) to attenuate out-of-band environmental noise and motion artifacts while preserving the morphological integrity of fundamental heart sounds.
- *Audio Compression:* Integrate the LC3 audio codec in a microcontroller to ensure real-time compression to reduce the transmission payload while maintaining medical-grade signal quality.
- *System Validation:* Evaluate the driver's performance in terms of compression efficiency, signal distortion, and computational load, ensuring it meets the stringent requirements of a wearable medical device.

Scope Limitations
While this thesis provides a robust embedded foundation for cardiac audio acquisition and compression, certain aspects remain outside its scope. Specifically, the development of a smartphone application frontend, the implementation of clinical validation with real patients are not addressed. The primary focus of this work lies strictly in the algorithmic processing, firmware integration, and hardware validation of the embedded system.

1.4 Thesis Structure

This work is organized as follows:

- *Chapter 2 (Background)* provides the theoretical foundation of the work, detailing the physiological characteristics of heart sounds, the principles of digital phonocardiography, signal processing techniques, and a comparative review of audio compression strategies for low-power wireless systems.

- *Chapter 3 (System Design and Materials)* describes the overall system architecture and the hardware platform, focusing on the selected microcontroller and the digital MEMS microphone used for acoustic acquisition. It also details the software architecture, the CMSIS and digital signal processing implementation, and the role of the LC3 codec.
- *Chapter 4 (Firmware Implementation)* details the implementation of the audio driver, including the DMA circular buffer and state machine. It also explains the real-time preprocessing on the MCU, the LC3 codec integration, and the data transmission and validation interface.
- *Chapter 5 (Experimental Setup and Results)* presents the experimental methodology and hardware characterization. It evaluates the system by analyzing the preprocessing validation, LC3 audio compression performance, and end-to-end system validation.
- *Chapter 6 (Discussion and Conclusions)* presents the interpretation of the results, discusses the limitations of the system, and outlines directions for future work.

Chapter 2

Background

2.1 Context of the Study

Wearable devices and smartphones are increasingly used as part of human-centric sensing systems, where sensing and computation are organized around the person rather than a single device. In mobile health, this typically means that a body-worn multisensor node acquires physiological signals, sends them via Bluetooth to a smartphone, and then forwards the data over the network to a remote analysis platform for storage, visualization, and decision support [1]. Figure 2.1 illustrates this pipeline for the specific case of cardiac auscultation.

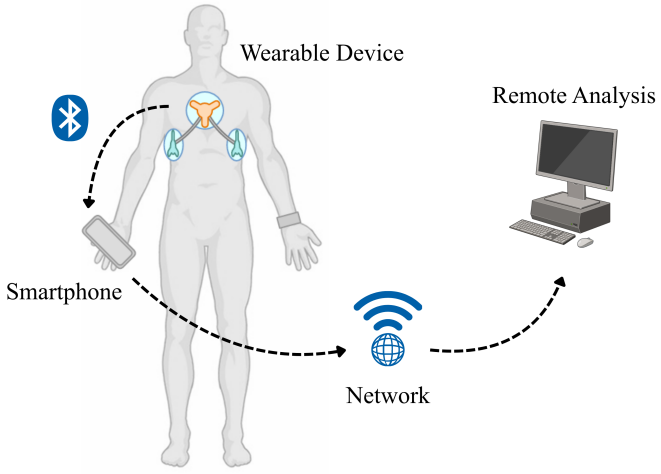


Figure 2.1: Human-centric sensing architecture for cardiac auscultation: a wearable device acquires heart sounds and sends them via Bluetooth to a smartphone, which acts as a gateway to the network and remote analysis platform.

In this architecture, the wearable device is responsible for continuous and non invasive signal acquisition under strict constraints of size, battery capacity, memory, and processing capability. The smartphone acts as a local gateway, managing Bluetooth Low Energy (BLE) communication and user interaction while relaying data to cloud or clinical infrastructures. Although this distributed approach enables long-term and out-of-hospital monitoring, it also imposes significant limitations on energy consumption and wireless bandwidth.

Data transmission is typically the dominant contributor to energy usage in remote healthcare monitoring systems, often exceeding sensing and local processing costs. Since battery-powered wearable devices must operate for extended periods, reducing the transmitted data volume is essential to ensure sustainable and reliable monitoring.

2.2 Cardiac Auscultation and Phonocardiography

Cardiac auscultation refers to the assessment of mechanoacoustic signals generated by cardiac mechanical activity and blood flow. Since the introduction of the acoustic stethoscope in 1816 (Figure 2.2a), it has remained a first-line diagnostic tool. This simple yet effective method is particularly relevant given the global prevalence of cardiovascular diseases (CVDs), which continue to be one of the leading causes of death worldwide [2]. Heart sounds often provide the earliest indications of valvular or hemodynamic abnormalities, guiding subsequent diagnostic evaluation.

From a physiological perspective, heart sounds carry critical information about valve motion, ventricular function, and hemodynamics. Abnormalities in these sounds, such as cardiac murmurs, serve as acoustic signatures of altered flow patterns and typically indicate valve-related pathologies. However, traditional acoustic auscultation is fundamentally limited by the subjectivity of human hearing, susceptibility to environmental noise, and the inability to store, share, or objectively reproduce the acoustic findings. To overcome these historical barriers, the digital stethoscope (Figure 2.2b) emerged, allowing for the electronic capture and amplification of bodily sounds.

Building upon this electronic foundation, Phonocardiography (PCG) addresses the core limitations of subjective auscultation by enabling the direct recording and visualization of heart sounds. As illustrated in Figure 2.3, PCG utilizes electronic sensors—such as digital microphones placed against the chest wall—to capture the acoustic waves propagating from the heart and transform them into a quantifiable visual waveform in the time domain.

This digital acquisition paradigm is transformative: it enables precise signal amplification, digital filtering to isolate cardiac bands, long-term storage, and

algorithmic analysis. Furthermore, PCG facilitates seamless integration into wearable monitoring systems and Internet of Medical Things (IoMT) frameworks [3], while also supporting multimodal diagnostics through synchronization with other physiological signals, such as electrocardiography (ECG).

(a) Traditional acoustic stethoscope. **(b)** Modern digital stethoscope.

Figure 2.2: Illustration of two types of stethoscopes: (a) traditional acoustic and (b) modern digital.

2.2.1 Wearable and Digital Auscultation Devices

Advances in sensor technology and embedded systems have enabled the transition from purely acoustic stethoscopes to digital and wearable auscultation devices. Unlike traditional stethoscopes, which transmit vibrations mechanically through tubing, electronic systems convert chest wall vibrations into electrical signals for amplification and processing.

Different sensing modalities have been employed in digital auscultation systems. Piezoelectric sensors offer robustness and mechanical sensitivity, electret condenser microphones provide wide bandwidth, and MEMS microphones combine compact size, reliability, and favorable frequency response characteristics for PCG applications. Accelerometers can also function as contact vibration sensors. In some

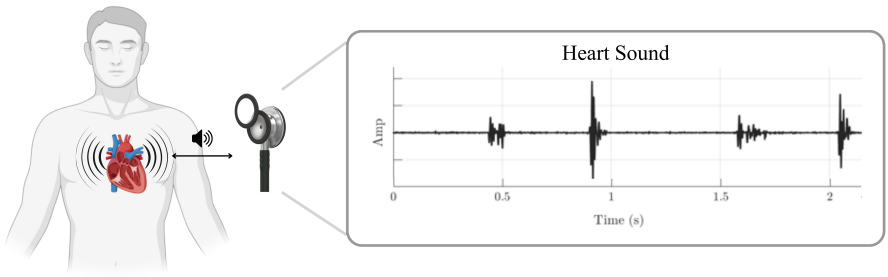


Figure 2.3: Phonocardiography (PCG). Acoustic vibrations generated by the mechanical activity of the heart (left) are captured by an electronic transducer on the chest wall and converted into a continuous digital waveform (right), allowing the objective visualization of the fundamental S_1 and S_2 cardiac events in the time domain.

designs, multiple modalities are combined to enhance signal robustness.

Electronic acquisition systems are capable of detecting and amplifying these low-amplitude and near-subaudible components with higher sensitivity and repeatability than purely acoustic stethoscopes. By employing suitable sensors, low-noise amplification, and digital filtering, wearable phonocardiographic systems can preserve diagnostically relevant information that might otherwise be attenuated or overlooked in traditional auscultation. Figure 2.4 highlights that diagnostically relevant heart sound components are concentrated in low-frequency regions, some of which lie near the lower threshold of human auditory sensitivity. The human ear is relatively insensitive below approximately 50–60 Hz, yet S3, S4, and certain pathological vibrations may contain useful information in this range.

Recent systems demonstrate the feasibility of wireless phonocardiography using MEMS microphones and Bluetooth transmission [5]. More advanced platforms integrate multimodal sensing, including accelerometers and digital MEMS microphones, to improve signal quality and diagnostic capability [6]. Smartphone-based solutions have also been explored, though they typically provide lower signal consistency compared to dedicated wearable devices [7].

Overall, these developments confirm the technical feasibility of portable and wearable cardiac acoustic monitoring, while highlighting ongoing challenges related to power efficiency, signal quality, and data transmission constraints.

2.2.2 Physiology and Spectral Characteristics of Heart Sounds

Heart sounds are low-amplitude, band-limited mechanical vibrations transmitted through the chest wall. The primary components are the fundamental sounds S1

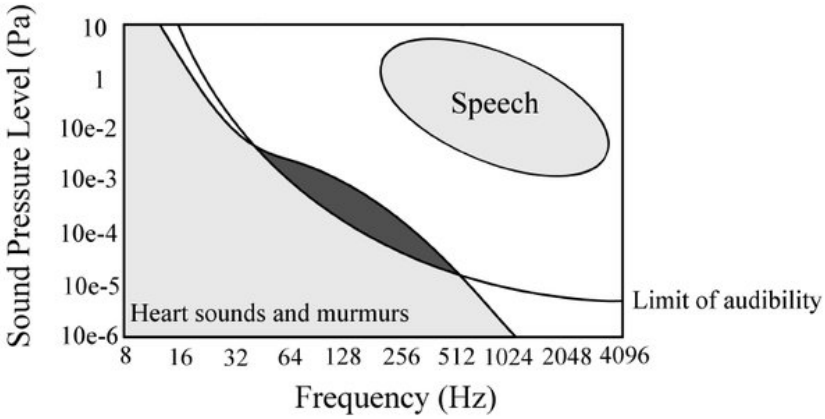


Figure 2.4: Frequency regions of heart sounds and murmurs relative to the human threshold of audibility. Clinically relevant components extend toward low-frequency bands near or below the most sensitive region of human hearing (adapted from [4]).

and S2. The first heart sound (S1) occurs at the beginning of the isovolumetric ventricular contraction and is associated with the closure of the atrioventricular (mitral and tricuspid) valves. The second heart sound (S2) marks the beginning of diastole and corresponds to the closure of the semilunar (aortic and pulmonic) valves [4]. Depending on specific physiological or pathological conditions, additional sounds may be produced by the mechanical activity of the heart. These include the third and fourth heart sounds (S3, S4), as well as murmurs, which are sustained acoustic phenomena caused by the turbulent, high-velocity flow of blood across structural defects or narrowed valves.

The spectral properties of heart sounds have been already studied and have been well described [8]. The clinically relevant energy is concentrated at relatively low frequencies, typically below 250 Hz for fundamental sounds, while murmurs and transient events may extend toward higher bands. Recordings are frequently contaminated by lung sounds, ambient noise, friction, and motion artifacts, motivating the need for appropriate preprocessing and filtering. Table 2.1 summarizes the typical frequency ranges for distinct cardiac events alongside common interference sources. The spectral overlap between cardiac murmurs and respiratory sounds underscores the critical need for careful acquisition design and robust digital signal conditioning in embedded implementations.

To illustrate this morphological diversity, Figure 2.5 presents representative 4-second temporal windows of phonocardiograms categorized into four distinct clinical classes. The figure contrasts the clean, rhythmic impulses of normal heart sounds against the sustained acoustic turbulence of murmurs, the presence of anomalous extra sounds, and the waveform degradation caused by broadband noisy

artifacts.

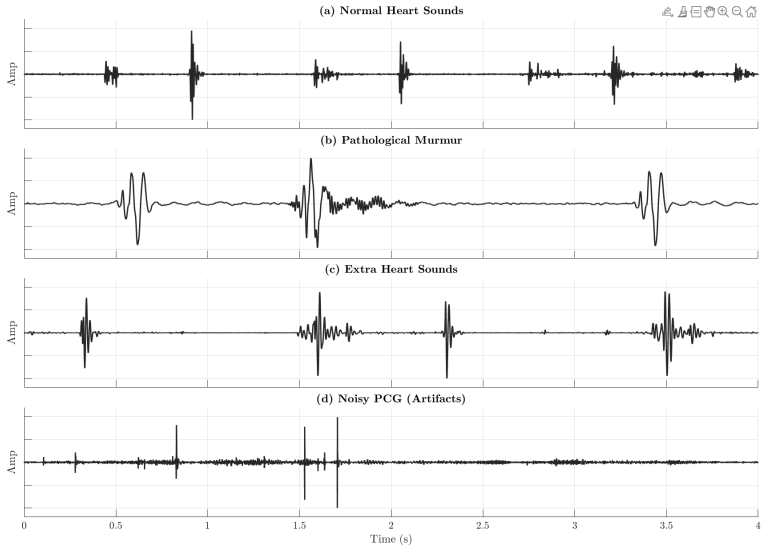


Figure 2.5: Representative time-domain waveforms of the four main PCG categories (normalized amplitude, 4-second windows). (a) Normal heart sounds exhibiting clear S1 and S2 peaks; (b) Pathological murmurs characterized by sustained, high-frequency turbulence between fundamental beats; (c) Extra heart sounds showing additional transient acoustic events within the cardiac cycle; and (d) Noisy PCG heavily degraded by broadband environmental and motion artifacts.

2.2.3 Auscultation Sites and Practical Acquisition

The audibility and spectral content of heart sounds vary across the precordium. Among the standard auscultation sites, Erb's point (left parasternal, third intercostal space) is frequently preferred because the first and second heart sounds (S1 and S2) are typically heard with comparable intensity. For wearable or sensor-based configurations, parasternal placement near the sternum offers stable mechanical coupling and consistent capture of the fundamental cardiac components.

In this work, the acquisition site follows the placement identified as P3 in [5], corresponding to the left parasternal region adjacent to the sternum. This location provides reliable signal quality and is well suited for practical phonocardiographic recording.

Table 2.1: Typical frequency content of heart sounds and common interference sources, including their physiological or physical origin.

Sound	Origin	Frequency Range
	Heart Sounds	
S1 (Lub)	Closure of mitral and tricuspid valves (onset of systole)	10–200 Hz
S2 (Dub)	Closure of aortic and pulmonary valves (end of systole)	20–250 Hz
S3, S4	Ventricular filling and atrial contraction phenomena	15–70 Hz
Murmurs	Turbulent blood flow across valves or septal defects	Up to 600 Hz
Opening snaps, clicks	Sudden valve opening or abnormal leaflet motion	100–800 Hz
	Interferences	
Lung sounds	Airflow in bronchial tree and alveoli	200–2000 Hz
Ambient noise	Environmental acoustic sources	20–2000 Hz

2.3 MEMS Microphones for Medical Applications

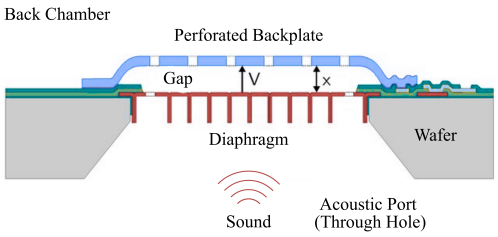
Micro-electro-mechanical systems (MEMS) have become a key enabling technology in modern medical devices. They exploit micro-scale mechanical structures and advanced fabrication processes to implement miniature sensors and actuators with high accuracy and reproducibility [9]. Compared with traditional electret condenser microphones (ECMs), MEMS microphones offer several advantages: smaller size, improved robustness to mechanical shocks, compatibility with high-temperature automated PCB assembly, and better control over device-to-device variability. Moreover, MEMS transducers can be co-packaged with CMOS electronics, reducing area and external component count while improving reliability and overall performance.

Beyond microphones, modern wearable medical systems commonly integrate multiple MEMS-based sensors—such as accelerometers, IMUs, and force sensors—together with a microcontroller in a compact form factor [1]. This high level

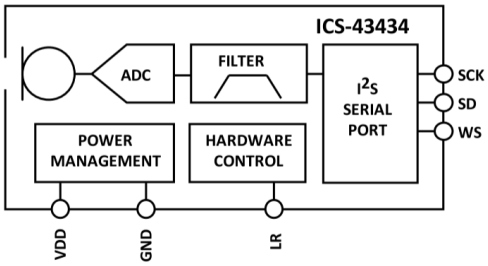
of integration enables real-time, on-device processing (edge computing) and creates the basis for remote, personalized monitoring solutions that can support earlier diagnosis and continuous follow-up using low-cost, mass-produced hardware.

A microphone is an electroacoustic transducer that converts incident sound waves into an electrical signal. Although different transduction principles exist—dynamic, piezoelectric, optical, or capacitive—the general mechanism is similar: an acoustic wave sets a mechanical element into motion, and this motion is converted into an electrical quantity [9]. In this work, a capacitive MEMS microphone is employed. As illustrated in Figure 2.6a, a compliant diaphragm and a rigid backplate form a variable capacitor; acoustic pressure causes small deflections of the diaphragm, which modulate the capacitance and generate the electrical signal.

In a typical digital MEMS microphone, shown in Figure 2.6b, the mechanical-to-electrical conversion is followed by an integrated analog front-end and an over-sampling sigma-delta analog-to-digital converter $\Sigma\Delta$. The converter produces a pulse-density modulated (PDM) bitstream, which is then decimated and filtered to obtain a pulse-code modulated (PCM) signal. This PCM output can be connected directly to a microcontroller through standard digital audio interfaces such as I^2S. The result is a compact signal chain that minimizes external analog circuitry and simplifies embedded audio acquisition.



(a) Schematic structure of a capacitive MEMS microphone.



(b) Digital MEMS microphone with on-board signal conditioning and PDM/PCM conversion. Adapted from InvenSense [10]

Figure 2.6: Conceptual structure and signal chain of a capacitive digital MEMS microphone.

The choice of a digital MEMS microphone in this project is motivated by its favorable signal-to-noise ratio in the cardiac band, relatively flat frequency response, and seamless integration with the on-board signal-processing pipeline. Comparative studies of microphone technologies for electronic stethoscopes have reported that digital MEMS-based solutions can provide both a wide usable bandwidth and high SNR, outperforming alternative transducer types in this application domain [11].

2.4 Audio Interfaces and Embedded Audio Acquisition (I2S - DMA)

The acquisition of digital audio in embedded systems relies on specialized serial protocols and hardware peripherals designed to handle high data rates with minimal CPU intervention. Inter-Integrated Circuit Sound (I²S) is the standard synchronous serial protocol for transmitting digital audio between integrated circuits (IC) [12] . An I²S bus consists of 3 lines:

- Serial Clock (SCK): continous clock that defines the bit rate.
- Word Select (WS): frame synchronization signal that selects the active channel.
- Serial Data (SD): data line carrying time-multiplexed audio samples.

The master device generates SCK and WS, while slaves synchronize to these signals. For stereo audio, WS operates at 50% duty-cycle: one half-period transmits the left channel and the other the right channel. Audio samples are transmitted in pulse-code modulation (PCM) format, encoded as two's complement, with the MSB (most significant bit) first and are aligned with SCK transitions as illustrated in Figure 2.7. In this project, a digital MEMS microphone provides a PCM stream that is captured by the microcontroller's Serial Audio Interface (SAI) peripheral, which implements the I²S standard.

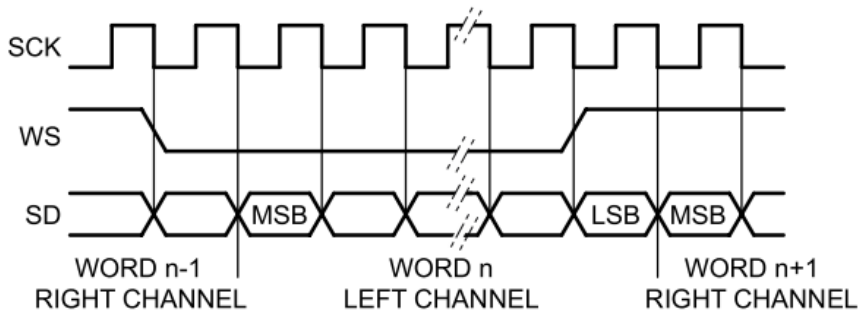


Figure 2.7: I²S timining diagram showing Serial Clock (SCK), Word Select (WS), and Serial Data (SD) for left and right channels.

2.5 Signal Processing Techniques

2.5.1 Signal Processing in Wearable Cardiac Monitoring

Biomedical signals such as phonocardiograms (PCG) are typically low-amplitude, non-stationary, and affected by multiple sources of interference. In the context of wearable cardiac monitoring, signal processing serves two main purposes. First, preprocessing techniques aim to attenuate noise and artifacts while preserving the intrinsic morphology of heart sounds. Second, analysis techniques—often performed offline or on more powerful processors—extract spectral, temporal, or statistical features for interpretation or classification. These operations can be carried out in the time domain, frequency domain, or time–frequency domain, each providing complementary information about the signal.

Given the strict energy and computational constraints of embedded wearable systems, the choice of processing techniques must balance diagnostic utility with implementation efficiency.

This section presents the theoretical foundations of the signal processing strategies employed in this work. The discussion is organized according to the processing domain: time-domain methods (digital filtering) for real-time embedded implementation, and frequency-domain/time-frequency methods for offline validation and analysis.

2.5.2 Noise Sources and Challenges in Ambulatory PCG Recording

Chest sound recordings acquired in ambulatory conditions are affected by multiple noise sources that compromise signal quality and diagnostic utility. These include:

- Environmental noise: Background sounds from conversations, air conditioning, and equipment operation
- Sensor-skin friction: Mechanical noise caused by movement or improper contact between the MEMS microphone and skin
- Motion artifacts: Low-frequency fluctuations resulting from patient movement or external disturbances
- Biological interference: Respiratory sounds, muscle contractions, and other physiological noises that spectrally overlap with heart sounds

A critical challenge in wearable cardiac monitoring is that these disturbances often overlap spectrally with the cardiac band (20–400 Hz), complicating simple frequency separation. Preprocessing must therefore attenuate out-of-band components

while preserving diagnostically relevant information with minimal computational overhead.

2.5.3 Preprocessing Strategies

The selection of a preprocessing strategy for wearable phonocardiography requires a careful balance between signal enhancement capabilities and computational complexity.

Wavelet-based methods, particularly the Discrete Wavelet Transform (DWT), represent a powerful approach for processing non-stationary biomedical signals. By decomposing the signal into time-localized frequency bands using mother wavelets (commonly Daubechies families such as *db4* or *db6* with decomposition levels between 4 and 7), DWT enables the selective thresholding of noise-dominated coefficients while preserving transient cardiac events such as S1 and S2 sounds [13]. Several studies report that wavelet-based denoising is particularly effective for mitigating motion artifacts and respiratory interference [5].

In this work, wavelet-based analysis (specifically the Continuous Wavelet Transform) is employed exclusively during offline signal evaluation, where computational resources are not constrained. This multiresolution framework allows for a detailed visual inspection of the non-stationary PCG signal and provides a robust baseline for validating the preprocessing performance. However, the computational and memory demands of wavelet transforms—requiring multiple filter bank stages, coefficient thresholding, and signal reconstruction—make them unsuitable for real-time execution on resource-limited microcontrollers.

Consequently, for the embedded implementation, the system adopts a digital band-pass Infinite Impulse Response (IIR) filter. As detailed in the subsequent sections, the IIR approach provides adequate frequency selectivity while maintaining an extremely low computational footprint. Relying on simple recursive difference equations with fixed coefficients, it ensures deterministic execution time, minimal CPU usage, and predictable memory consumption. These are fundamental requirements for wearable devices operating under strict power constraints. The designed filter emphasizes the principal cardiac frequency band (20–400 Hz), attenuating low-frequency motion artifacts and high-frequency environmental noise, thereby delivering a clean signal to the LC3 compression stage without imposing significant processing overhead.

2.5.4 General Principles of Biomedical Signal Processing

Time-domain processing directly manipulates the discrete-time signal samples acquired from the sensor. Among the most widely used operations in biomedical applications is digital filtering, which selectively attenuates or amplifies specific

frequency components of the signal.

A digital filter can be modeled as a linear time-invariant (LTI) system whose output $y[n]$ is obtained through convolution of the input signal $x[n]$ with a predefined impulse response[14].

Conceptually, the filter shapes the spectral content of a discrete-time signal according to the characteristics of the filter's impulse response to enhance diagnostically relevant components or suppress noise. In the time domain, filtering corresponds to a weighted combination of current and past samples of the input (and, in some cases, of the output), while in the frequency domain it is equivalent to multiplying the signal spectrum by a frequency-dependent transfer function $H(e^{j\omega})$.

Difference Equation, Impulse Response and Filter Order

A general causal digital filter is described by the linear difference equation

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k] - \sum_{k=1}^{N} a_k \, y[n-k], \tag{2.1}$$

where $x[n]$ and $y[n]$ denote the input and output signals, respectively, and a_k, b_k are real-valued *filter coefficients* that determine the behaviour of the system [15]. The integers M and N define how many past input and output samples contribute to the current output; the *filter order* is typically given by $\max(M, N)$ and is directly related to how sharp the transition between passband and stopband can be.

The corresponding *impulse response* $h[n]$ is the output of the system when the input is a unit impulse $\delta[n]$. For a given set of coefficients, $h[n]$ is completely determined and can be used to characterize the filter in both time and frequency domains [15].

Transfer Function and Frequency Response

Applying the z-transform to (2.1) yields the rational transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}, \tag{2.2}$$

whose poles and zeros determine stability and frequency selectivity. Evaluating (2.2) on the unit circle $z = e^{j\omega}$ gives the discrete-time frequency response

$$H(e^{j\omega}) = |H(e^{j\omega})| \, e^{j\phi(\omega)}, \tag{2.3}$$

where $|H(e^{j\omega})|$ is the magnitude response (gain as a function of frequency) and $\phi(\omega)$ is the phase response. In biomedical applications, the magnitude response is used to define the cardiac passband (e.g. 20–400 Hz for PCG), while the phase response is important to preserve the temporal morphology of heart sounds.

FIR vs IIR Filter Architectures

Finite Impulse Response (FIR) filters are obtained by setting $a_k = 0$ in (2.1), which removes the recursive part:

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k]. \tag{2.4}$$

The impulse response is simply $h[n] = b_n$ for $0 \leq n \leq M$ and zero otherwise. FIR filters are inherently stable and can be designed to have a linear phase [16]. The main drawback is that sharp transitions in the frequency response require a high order (many coefficients), increasing memory and computational cost, prohibitive for embedded real-time operation.

Infinite Impulse Response (IIR) filters retain the recursive feedback terms in (2.1), typically with M and N of low order:

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k] - \sum_{k=1}^{N} a_k \, y[n-k]. \tag{2.5}$$

Because past outputs are fed back into the system, the impulse response decays asymptotically but never becomes exactly zero, hence "infinite". IIR filters can approximate a desired magnitude response (e.g. Butterworth, Chebyshev, elliptic designs) with significantly fewer coefficients than an equivalent FIR filter, leading to much lower computational and memory requirements. This efficiency makes them attractive for embedded biomedical devices [16].

However, IIR filters have two important limitations. First, stability must be carefully checked to ensure that all poles of $H(z)$ lie inside the unit circle; otherwise the output may diverge. Second, their phase response is generally non-linear, introducing phase distortion and group-delay dispersion that can slightly alter the relative timing of S1, S2 and murmurs. In offline analysis this can be mitigated with zero-phase forward–backward filtering, but such techniques are not suitable for real-time embedded processing.

Cascaded Biquad Implementation

In practical applications, high-order IIR filters are rarely implemented as single transfer functions due to numerical instability from coefficient quantization sensitivity. Instead, the transfer function is decomposed into a product of K second-order sections (SOS), or *biquads*:

$$H(z) = \prod_{k=1}^{K} H_k(z) = H_1(z) \cdot H_2(z) \cdot \ldots \cdot H_K(z). \tag{2.6}$$

where each individual stage is defined by a biquadratic transfer function:

$$H_k(z) = \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}, \quad k = 1, \ldots, K. \tag{2.7}$$

The complete filter is realized by connecting these K sections in series, as illustrated in Figure 2.8.

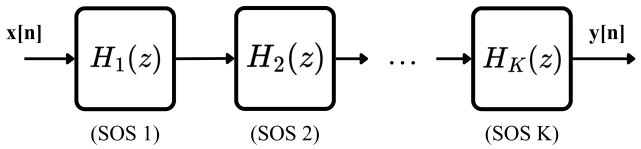


Figure 2.8: Cascaded biquad structure

Regarding the phase response, the total phase shift introduced by the cascade is the cumulative sum of the individual stage phases:

$$\phi_{\text{total}}(\omega) = \sum_{k=1}^{K} \phi_k(\omega) = \phi_1(\omega) + \ldots + \phi_K(\omega). \tag{2.8}$$

Similarly, the overall magnitude response of the system is the product of the individual magnitudes of each section:

$$|H(e^{j\omega})| = |H_1(e^{j\omega})| \cdot |H_2(e^{j\omega})| \cdot \ldots \cdot |H_K(e^{j\omega})|. \tag{2.9}$$

Expressing the magnitude in decibels (dB), the relationship becomes additive, just like the phase:

$$20 \log_{10} |H(e^{j\omega})| = \sum_{k=1}^{K} 20 \log_{10} |H_k(e^{j\omega})|. \tag{2.10}$$

The cascade structure offers decisive advantages for embedded implementation [16]:

- Numerical stability: Each second-order section localizes quantization errors, preventing propagation across the entire filter
- Reduced sensitivity: Poles and zeros are more robust to finite-precision arithmetic, critical for fixed-point microcontrollers
- Computational efficiency: Optimized libraries such as CMSIS-DSP exploit Cortex-M architecture for single-cycle multiply-accumulate operations on cascaded biquads [17]

This modular approach is the standard implementation strategy for practical IIR filters in resource-constrained environments.

2.5.5 Frequency-Domain and Time-Frequency Analysis for Offline Validation

Beyond preprocessing, frequency-domain analysis provides quantitative insight into the spectral distribution of a PCG signal energy for validation and interpretation. Because heart sounds are inherently non-stationary, containing transient components such as S1, S2, and short-duration murmurs, pure frequency-domain methods are often insufficient to capture their temporal evolution.

Time–frequency representations address this limitation by jointly describing how spectral content changes over time, enabling the characterization of transient phenomena and the identification of pathological components that may only appear during specific phases of the cardiac cycle. These analyses are computationally demanding and therefore performed offline during validation rather than on the microcontroller.

Fourier Transform and Spectral Representation

The theoretical basis of frequency-domain analysis is the Fourier transform, which decomposes a signal into sinusoidal components of different frequencies [14].

For a continuous-time signal $x(t)$, the Fourier transform is defined as:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt. \tag{2.11}$$

For discrete-time signals $x[n]$ sampled at frequency f_s, the Discrete-Time Fourier Transform (DTFT) is:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \tag{2.12}$$

where $\omega = 2\pi f/f_s$ is the normalized angular frequency.

In practical implementations, the Discrete Fourier Transform (DFT) is computed over N samples using the Fast Fourier Transform (FFT):

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}, \quad k = 0,1,\ldots,N-1. \tag{2.13}$$

The DFT provides spectral coefficients at discrete frequencies $f_k = kf_s/N$. For real-valued PCG signals, only the first $N/2+1$ bins contain unique information due to conjugate symmetry.

FFT analysis enables visualization of dominant frequency components, verification of filter behavior, and inspection of noise contributions outside the cardiac band.

Power Spectral Density Estimation via Welch's Method

The Power Spectral Density (PSD) describes how signal power is distributed across frequency. Direct periodogram estimation suffer from high variance, which limits their reliability for biomedical signals. Welch's method addresses this issue by segmenting the signal, applying windowing, and averaging multiple modified periodograms, thereby reducing variance without excessively degrading frequency resolution [18]:

1. Divide the signal into K overlapping segments of length L
2. Apply a window function (e.g., Hamming) to each segment
3. Compute the periodogram:

$$P_k(f) = \frac{1}{L}|FFT(x_k)|^2$$

4. Average across segments:

$$\hat{P}(f) = \frac{1}{K}\sum_{k=1}^{K} P_k(f)$$

In PCG analysis, Welch's method provides a robust spectral representation that highlights dominant components, reveals noise contributions, and supports objective validation of filtering and compression stages.

Time–Frequency Analysis: STFT and Wavelet Transforms

While the PSD provides information about the global distribution of signal energy across frequency, it does not capture temporal variations. Phonocardiogram (PCG) signals are inherently non-stationary: heart sounds (S1, S2) and murmurs are transient events whose spectral content evolves rapidly during the cardiac cycle. To jointly analyze time and frequency behavior, time–frequency representations are required.

Short-Time Fourier Transform (STFT) The Short-Time Fourier Transform (STFT) extends Fourier analysis by computing the spectrum within a sliding time window:

$$X(\tau, \omega) = \sum_{n} x[n]\, w[n-\tau]\, e^{-j\omega n}, \tag{2.14}$$

where $w[n]$ is a window function (e.g., Hamming) centered at time τ. The squared magnitude $|X(\tau, \omega)|^2$ generates the visual representation known as the spectrogram.

However, the STFT is fundamentally constrained by its use of a fixed window length, which imposes a constant time–frequency resolution. This leads to a Heisenberg-Gabor uncertainty trade-off: short windows provide excellent temporal precision but poor frequency resolution, whereas long windows yield high frequency resolution at the cost of temporal smearing. For PCG signals, this limitation is severely penalizing. Sub-audible and low-frequency components (20–150 Hz) require high frequency resolution, while high-frequency murmurs require precise temporal localization. A single fixed window cannot optimally represent both simultaneously, often leading to spectral smearing in the low-frequency cardiac band [15, 19].

Continuous Wavelet Transform (CWT) The Continuous Wavelet Transform (CWT) addresses the STFT's limitations through multiresolution analysis. Instead of a fixed window, CWT utilizes an 'elastic' approach by scaling and analyzes the signal using scaled and shifted versions of a mother wavelet:

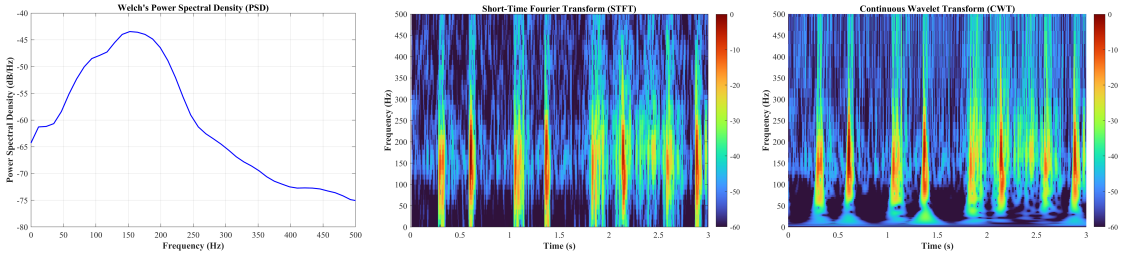
$$W(a,b) = \frac{1}{\sqrt{a}} \int x(t)\, \psi^* \left(\frac{t-b}{a} \right) dt, \tag{2.15}$$

where a is the scale parameter (inversely related to frequency), b is the time shift, and $\psi(t)$ is the mother wavelet.

This mathematical formulation allows the CWT to dynamically adapt its resolution. When capturing high-frequency components (small scale a), the wavelet compresses, acting as a narrow temporal window that offers extreme temporal precision—suitable, for instance, for identifying the exact instant of an aortic valve closure. In contrast, when capturing low-frequency components (large scale a), the wavelet stretches, enhancing frequency resolution and allowing the transform to accurately capture the spectral signature of deep, low-frequency cardiac vibrations.

In this work, the Analytic Morlet wavelet is employed for the analysis of cardiac acoustics. The Morlet wavelet—constructed as a complex exponential modulated by a Gaussian envelope—exhibits a morphology that closely resembles the damped, transient nature of human heart sounds. This structural affinity maximizes the correlation between the wavelet and the underlying PCG waveform, yielding a time–frequency representation that is both clean and minimally affected by artifacts.

Consequently, the CWT yields significantly sharper and more clinically informative visual representations of cardiac dynamics compared to the STFT [19]. To illustrate this critical difference, Figure 2.9 compares the PSD, the STFT spectrogram, and the CWT scalogram computed from the same PCG segment. This visual comparison highlights the blurring effect of fixed-resolution methods and the advantages of multiresolution wavelet analysis for inspecting low-frequency cardiac components.



(a) PSD **(b)** STFT Spectrogram **(c)** CWT Scalogram

Figure 2.9: Comparison of frequency-domain and time–frequency representations for a PCG signal. (a) Welch's PSD shows the global distribution of energy but lacks temporal tracking. (b) The STFT provides a time–frequency map but suffers from a rigid resolution trade-off, blurring low-frequency events. (c) The CWT offers multiresolution analysis, yielding unmatched frequency resolution at low bands and sharp temporal localization of transient acoustic impacts.

2.6 Audio Compression for Low-Power Wireless Systems

2.6.1 Audio sampling rates and need for compression

Audio is one of the most data-intensive biosignals, because clinically useful information is spread over relatively wide frequency bands and must be sampled at sufficiently high rates to avoid aliasing. According to the Nyquist theorem, a band-limited signal must be sampled at least at twice its highest frequency component in order to be perfectly reconstructable, as illustrated in Figure 2.10. If the sampling rate falls below this threshold, high-frequency components are erroneously reconstructed as lower-frequency waves, a destructive phenomenon known as aliasing.

For general audio applications, the upper limit of human hearing is often taken as 20 kHz, which motivates conventional sampling rates such as 44.1 or 48 kHz in music and streaming systems. Table 2.2 summarizes common audio bandwidth categories, from narrowband speech to full-band high-quality audio, and their approximate usable frequency ranges.

In phonocardiography, the bandwidth of interest is narrower but still non-negligible. Many studies assume that most heart sound and murmur energy lies below approximately 800 Hz, and therefore adopt sampling frequencies around 4 kHz with 16-bit resolution for PCG recording [20]. Even under these reduced conditions, such a stream still produces

$$f_s \times \text{bits} = 4{,}000 \times 16 \approx 64 \text{ kbit/s}$$

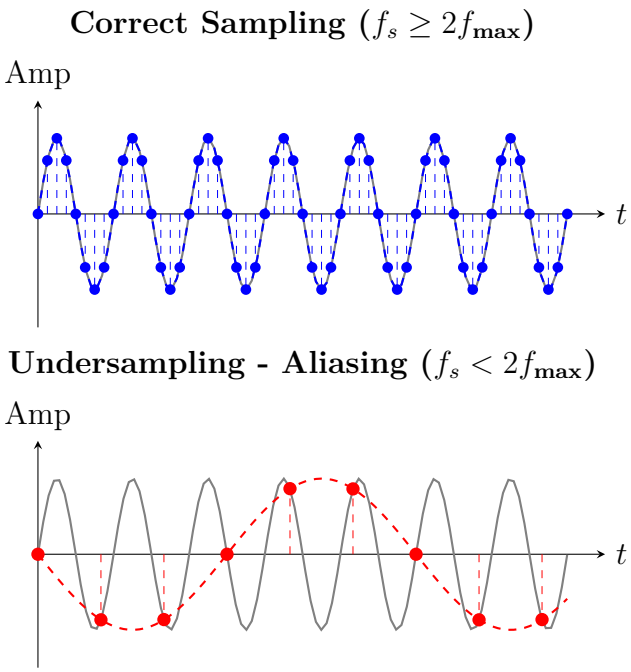


Figure 2.10: Visual representation of the Nyquist-Shannon sampling theorem. The upper plot shows correct signal reconstruction when the sampling rate (f_s) is sufficient. The lower plot demonstrates aliasing: undersampling causes the discrete points to form a false, lower-frequency alias of the original signal.

Table 2.2: Typical audio bandwidth categories and corresponding frequency ranges.

Category	**Approximate frequency range**
Narrowband (NB)	0–4 kHz
Wideband (WB)	0–8 kHz
Semi Super Wideband (SSWB)	0–12 kHz
Super Wideband (SWB)	0–16 kHz
Full Band (FB)	0–20 kHz

for a single channel, and higher sampling rates or wider bands increase this figure proportionally.

As established in Section 1, in battery-powered wearable devices, transmission energy typically dominates total energy consumption, and local memory is limited. Streaming raw audio at tens or hundreds of kilobits per second per channel rapidly

saturates the available BLE throughput and shortens battery life. As a result, on-device audio compression is mandatory to reduce the bitrate while preserving the diagnostic information contained in the PCG signal.

2.6.2 PCG audio compression techniques in the literature

A wide range of techniques has been proposed for compressing phonocardiogram (PCG) signals, typically aiming to balance compression ratio, signal fidelity, and energy consumption—three factors that are especially critical in wearable and low-power biomedical devices.

Dictionary- and codebook-based methods exploit the quasi-periodic structure of heart sounds. Tang et al. introduced a two-stage strategy combining sound repetition detection with vector quantization, where a dictionary of typical PCG patterns is learned and only dictionary indices plus a residual component are transmitted [20]. This approach achieves high compression with relatively low percentage root-mean-square difference (PRD). Similarly, Francescon et al. proposed a family of codebook-based algorithms for biomedical signals, designed specifically for resource-constrained wearable devices [1]. Recurring motifs are extracted and stored in a compact codebook, and new segments are encoded by matching them to the closest motif, reducing both memory footprint and transmission cost.

Transform-based compression is another widely explored direction. Wavelet- and DCT-based schemes project the PCG into a time–frequency domain and retain only the most significant coefficients, concentrating most of the signal energy into a compact representation [21]. In fetal PCG, more advanced moment-based techniques—such as fractional-order orthogonal moments or discrete Charlier moments—have been proposed to further reduce transmission time and energy in Internet-of-Healthcare-Things scenarios [22]. Several studies report that wavelet-based methods (e.g., DWT, WPT) often outperform DCT for preserving low-frequency cardiac components [13].

Compressive sensing and machine learning approaches represent a different paradigm. Ragusa et al. demonstrated that combining compressive sensing with lightweight neural networks running on STM32 microcontrollers enables direct classification of heart sounds from compressed measurements, bypassing explicit reconstruction while maintaining high F1-scores [3]. Similar architectures have been explored for multimodal wearable monitoring [23]. Deep learning–based autoencoders have also been proposed for PCG compression, using convolutional encoder–decoder structures to learn compact latent representations [19].

Overall, the literature shows that PCG signals can be compressed aggressively, particularly when the end goal is automated classification rather than high-fidelity reconstruction. However, many of these methods require custom decoders, specialized reconstruction algorithms, or dedicated neural networks on the receiver

side—factors that limit interoperability and increase system complexity.

In contrast, the present work adopts a *standardized audio codec*, LC3, which is part of the Bluetooth LE Audio ecosystem. LC3 offers a well-defined trade-off between compression efficiency and perceptual quality, can be decoded on off-the-shelf devices without custom software, and benefits from the robustness and widespread support of an established audio standard. This makes it particularly suitable for low-power wearable auscultation systems where interoperability and simplicity are essential.

2.6.3 The Low Complexity Communication Codec (LC3)

Wireless audio transmission in wearable medical devices must balance signal fidelity, transmission rate, power consumption, and latency. Although Bluetooth Low Energy (BLE) is widely adopted for low-power connectivity, its effective throughput is limited by protocol overhead. In practice, BLE supports up to 2 Mbit/s at the physical layer, but usable application throughput is typically around 1.4 Mbit/s or lower [24]. A 48 kHz, 16-bit mono audio stream requires approximately 1.5 Mbit/s, leaving insufficient margin for packet headers, retransmissions, and additional sensor data. As a result, audio compression is essential for reliable low-power transmission.

Earlier Bluetooth audio solutions relied on the Subband Codec (SBC), which provides acceptable quality at bitrates between 192 and 320 kbit/s but suffers from relatively high end-to-end latency and limited efficiency at lower bitrates [25]. Other codecs such as AAC or aptX target consumer multimedia applications and are not native to the BLE Audio framework, often requiring higher computational complexity or proprietary implementations.

Bluetooth LE Audio, introduced with Bluetooth 5.2, addresses these limitations through a redesigned architecture and the mandatory adoption of the Low Complexity Communication Codec (LC3) [24]. LC3 is a transform-based codec optimized for low computational complexity, short frame durations, and high perceptual quality at reduced bitrates [25]. Listening tests indicate that LC3 achieves quality comparable to or better than SBC while operating at approximately half the bitrate, as illustrated in Figure 2.11.

Unlike research-oriented PCG compression methods, LC3 can be decoded on off-the-shelf devices without custom software, ensuring interoperability and simplifying system design. This makes it an attractive choice for wearable auscultation systems, where the goal is to preserve diagnostic information while minimizing energy consumption and maintaining compatibility with standard BLE receivers.

LC3 supports a wide range of sampling rates and bitrates. In this work, a configuration of 16 kHz sampling rate, 32 kbps bitrate, and 10 ms frame duration was selected as a balance between diagnostic fidelity, latency, and computational

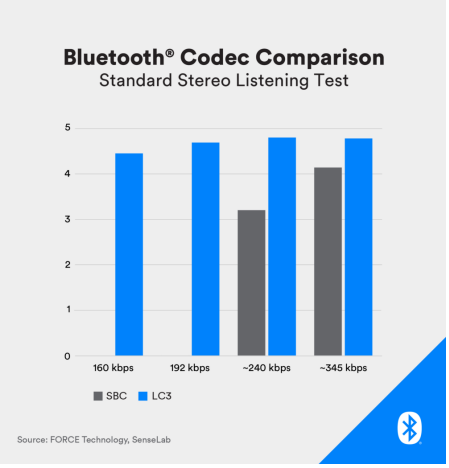


Figure 2.11: Comparison between Bluetooth audio codecs. LC3 provides improved perceptual quality compared to SBC at significantly lower bitrates [25].

load. Figure 2.12 illustrates the encoder–decoder pipeline used in the system, highlighting the shared parameters that must be identical on both ends of the link.

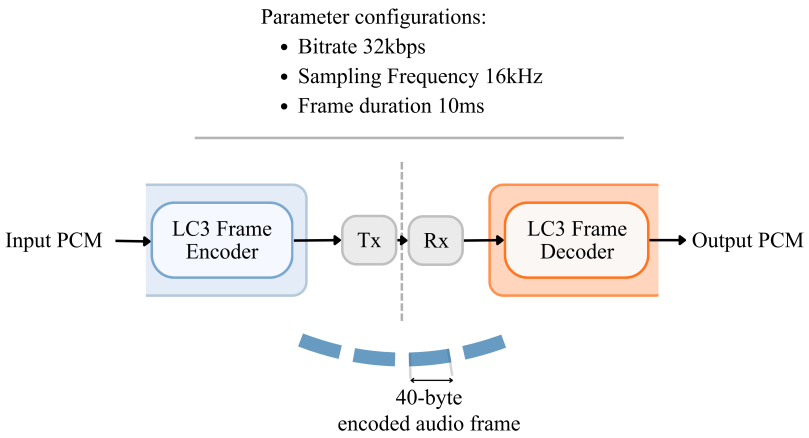


Figure 2.12: Conceptual LC3 encoding and decoding pipeline. The input PCM signal is segmented into 10 ms frames (40 bytes at 32 kbps), encoded, transmitted, and decoded back into PCM. Matching encoder–decoder parameters (sampling rate, bitrate, and frame duration) is essential for correct bitstream interpretation and reliable reconstruction.

This configuration enables low-latency transmission (20–30 ms end-to-end), reduces radio duty cycle, and can be executed in real time on resource-constrained microcontrollers such as the STM32U5. The theoretical foundations of LC3 are

discussed in Chapter 3, the embedded implementation is detailed in Chapter 4, and Chapter 5 validates that this operating point preserves the diagnostic features of the phonocardiogram.

LC3 in Medical Wearable Devices

LC3 has been widely adopted in modern hearing aids as part of the Bluetooth LE Audio ecosystem, replacing legacy analog or Bluetooth Classic-based solutions [26]. Its deployment in battery-constrained, continuously operating medical-grade devices demonstrates its maturity, reliability, and suitability for long-term audio streaming applications.

The selection of LC3 in this work is motivated by its standardization within the BLE Audio specification, its proven deployment in medical and assistive listening devices, and its balance between diagnostic signal fidelity, energy efficiency, and low latency. Unlike proprietary or research-oriented codecs requiring custom receivers [20], LC3 ensures interoperability with the broader LE Audio ecosystem while remaining computationally feasible for embedded microcontroller-based implementations.

Chapter 3

System Design and Materials

3.1 Overall System Architecture

The envisioned complete system is designed to acquire, enhance, compress, and transmit cardiac acoustic signals in a portable, low power, and efficient manner. The ideal architecture is organized into four functional layers, each addressing a specific stage of the signal chain:

- a *sensing layer* that captures heart sounds using a digital MEMS microphone,
- an *embedded processing layer* that runs on an STM32 microcontroller,
- a *wireless communication layer* intended to be based on Bluetooth Low Energy (BLE) in the final deployment,
- and a *remote analysis layer* responsible for decompression, visualization, and further signal analysis through a gateway and web-based application.

Figure 3.1 illustrates the block diagram of this complete architecture, highlighting the ideal signal flow from acquisition to remote analysis.

Within the embedded device, the audio firmware follows a precise processing pipeline as depicted in the diagram. First, the digital MEMS microphone converts acoustic pressure into a digital audio stream using an internal sigma-delta converter and decimation filters, providing a PCM signal over the I²S interface. Second, the acquired signal is preprocessed using a digital IIR filter that emphasizes the cardiac frequency band while attenuating undesired high-frequency components. Third, the filtered audio is compressed using the LC3 codec, which reduces the bitrate while preserving diagnostically relevant features.

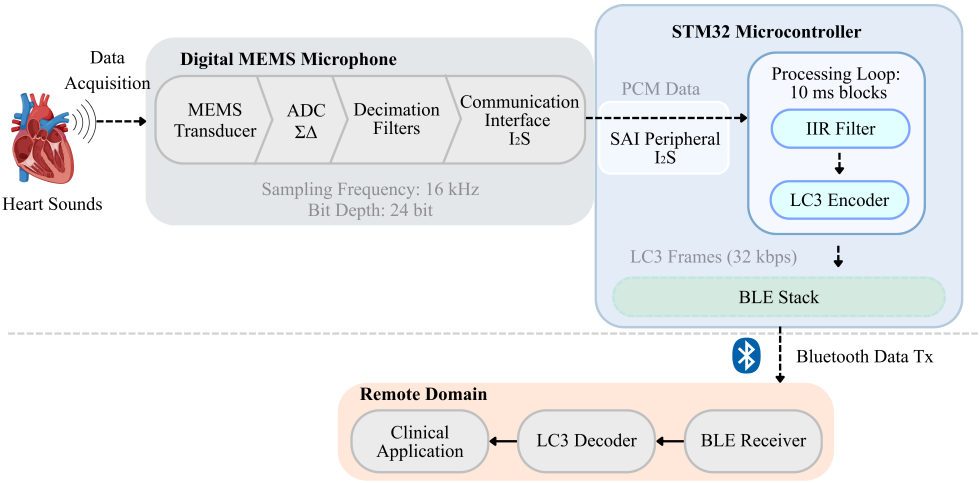


Figure 3.1: Block diagram of the proposed system architecture. *Note: While the envisioned system targets BLE transmission, the current prototype focuses on validating the embedded pipeline (preprocessing and LC3 compression), utilizing a generic wired data transmission link in place of the Bluetooth module.*

Finally, in the current implementation, these compressed frames are packetized and transmitted over a generic data link to a host computer (bypassing the BLE stage). On the receiving side, the LC3 decoder reconstructs the audio signal from the compressed stream, allowing it to be displayed, stored, and analyzed within a clinical software application.

The following sections provide a detailed description of the hardware and software components used to implement this architecture.

3.2 Hardware Platform

The hardware architecture is designed around three main pillars: the microcontroller, the digital MEMS microphone, and the wireless interface. While this work focuses on the implementation of the acquisition and compression stages, the system is designed to integrate a Bluetooth Low Energy (BLE) module as the primary communication link.

3.2.1 Microcontroller Unit (MCU)

The microcontroller is an STM32 device based on an Arm Cortex-M33 core. It provides sufficient computational performance and memory to support real-time audio processing under tight latency and power constraints. This capability is largely driven by its integrated single-precision Floating-Point Unit (FPU) and DSP

instruction set (SIMD), which drastically accelerate the complex mathematical operations required for IIR filtering and LC3 MDCT transformations.

Furthermore, the MCU architecture is highly receptive to aggressive C compiler build optimizations (e.g., the `-O3` flag) and features dedicated hardware such as the Filter Mathematical Accelerator (FMAC) and the Serial Audio Interface (SAI) with Direct Memory Access (DMA) support. This allows the system to offload data transfers and audio acquisition from the main CPU, maximizing the time spent in low-power sleep modes.

The selected model, the *STM32U545*, is specifically suited for wearable applications due to its ultra-low-power architecture. It offers a superior balance between energy efficiency and real-time processing capability. Additionally, it is one of the most cost-effective options in its family and provides a flexible set of peripherals, making it a practical choice for the proposed embedded system [27].

For the firmware development, prototyping, and experimental validation phases of this research, an official STMicroelectronics evaluation board housing the STM32U545 microcontroller was utilized. Relying on this evaluation platform facilitated rapid development and reliable hardware debugging. Specifically, it features an integrated ST-LINK programmer, which allows for seamless firmware flashing and high-speed data acquisition without requiring external programming hardware. This setup isolates the algorithmic validation of the audio driver from the hardware design complexities of a custom printed circuit board (PCB), which is reserved for future productization stages. Table 3.1 summarizes the MCU's main technical specifications.

Table 3.1: Technical specifications of the STM32U545 microcontroller.

Category	**Specification**
Processor Architecture	Arm® Cortex®-M33 with FPU and DSP
Performance	Up to 160 MHz
Supply Voltage	1.71 V to 3.6 V
Flash Memory	512 KB
SRAM	274 KB
Power Consumption	16.3 μA/MHz (Run) / 1.4 μA (Stop 3)
Digital Audio	2x SAI (I2S, PCM) with DMA support
Accelerators	CORDIC and FMAC (Filter Mathematical Accelerator)

3.2.2 MEMS Microphone

Acoustic sensing is performed using the *ICS-43434*, a digital MEMS microphone, manufactured by InvenSense. This model was selected for its high audio quality, low power consumption and seamless integration with the microcontroller via its direct I²S interface.

The microphone features a Signal-to-Noise Ratio (SNR) of 65 dBA, which is essential for capturing subtle cardiac sounds, and an Acoustic Overload Point (AOP) of 120 dB SPL to prevent distortion during environmental noise peaks. Internally, the device integrates a sigma-delta (Σ-Δ) modulator and a decimation filter that converts the raw analog signal into a 24-bit Pulse Code Modulated (PCM) stream. This internal path includes a high-pass filter with a fixed -3 dB corner at 24 Hz to eliminate DC offset and low-frequency motion artifacts. Additionally, a low-pass decimation filter scales with the sampling frequency (f_s), providing a flat passband up to $0.417 \times f_s$. As shown in Figure 3.2, the frequency response is nearly flat within the bandwidth of interest, translating into high linearity in the cardiac frequency range.

The microphone's power consumption is dynamically governed by the frequency of the incoming Serial Clock (SCK). It operates in High-Performance Mode (490 μA) for clock frequencies above 1.0 MHz and transitions to Low-Power Mode (230 μA) when the clock is between 400 kHz and 1.0 MHz. If the clock frequency falls below 3.125 kHz, the device automatically enters a 12 μA Sleep Mode.

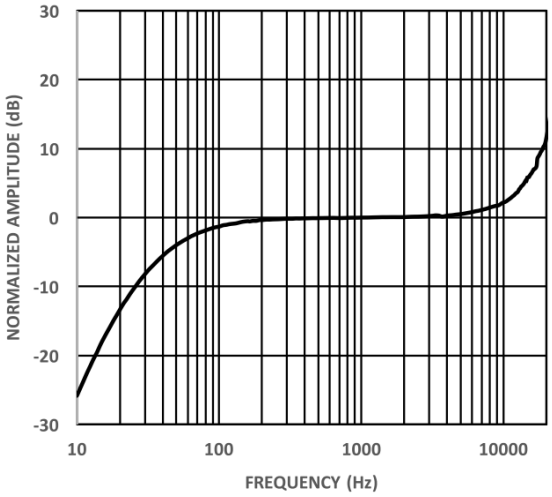


Figure 3.2: Frequency response of the ICS-43434 microphone. Adapted from Invensense datasheet[11]

Table 3.2: Technical specifications of the ICS 43434 digital MEMS microphone.

Category	Specification
Technology	MEMS sensor with digital I²S interface
Signal Integrity	65 dBA SNR and 120 dB AOP
Output Format	24-bit PCM, two's complement, MSB first
Dimensions	3.50×2.65×0.98 mm
Power Consumption	490 μA (High Performance) / 230 μA (Low-Power) / 12 μA (Sleep)
Operating Voltage	1.65 V to 3.63 V

3.2.3 Connectivity

Wireless Interface

The wireless communication layer is designed to utilize Bluetooth Low Energy (BLE) due to its optimized power profile for medical wearables. While the implemented system focuses on high-fidelity signal acquisition and LC3 compression within the *STM32U545* microcontroller, the system architecture is prepared for a seamless integration with a BLE communication via standard interfaces such as UART or SPI.

In the full-scale deployment, the wearable device acts as a BLE peripheral, and transmits the compressed LC3 frames encapsulated into BLE packets to a central gateway or smartphone, so audio data can be forwarded to a cloud backend or local storage for clinical analysis.

Real-time Debugging and Data Acquisition

For the validation and characterization of the system, the *Serial Wire Viewer (SWV)* interface was utilized as the primary data acquisition channel. Leveraging the Arm Cortex-M Instrumentation Trace Macrocell (ITM), the SWV provides a hardware-accelerated, high-speed debug channel capable of streaming data via the Serial Wire Output (SWO) pin. This signal is routed directly through the onboard ST-LINK programmer to the host PC, allowing continuous data transmission to the IDE console without interrupting the main CPU execution [28].

This mechanism enabled the real-time monitoring of internal variables and the direct streaming of processed audio frames to a console. The captured data was subsequently stored in raw format for offline processing and metric calculation. This approach ensured that the algorithmic integrity of the filtering and compression

stages could be rigorously verified, effectively isolating the embedded performance from the latency, bandwidth limitations, and packet loss overhead typical of early-stage wireless prototyping. The specific data formatting, transmission protocols, and payload reconstruction methodologies implemented over this interface are detailed in Chapter 4.

3.3 Software Architecture: Driver Layers

The firmware is organized according to a layered architecture that separates hardware-dependent code from high-level application logic. This structure improves modularity and maintainability, which is particularly important for embedded software in medical devices.

3.3.1 Layered Firmware Architecture

Conceptually, the firmware is divided into three main layers: the HAL/CMSIS and low-level driver layer, the middleware/driver layer, and the application layer. Figure 3.3 illustrates these layers and their interactions.

At the lowest software level, the Hardware Abstraction Layer (HAL), the CMSIS core, and the low-level (LL) device drivers interface directly with the STM32 hardware. They handle system clock configuration, SAI/I^2S setup for audio acquisition, DMA configuration for ping–pong buffering, timer and interrupt management, and control of the BLE peripheral. These components encapsulate register-level details behind standardized APIs, ensuring deterministic behaviour while shielding upper layers from device-specific implementation details.

Above this, the middleware or driver layer implements the audio-specific functionality. It includes modules for SAI/I^2S and DMA management, digital preprocessing using CMSIS-DSP IIR filters, LC3 encoding through a dedicated wrapper, and a transport interface that prepares compressed frames for the BLE stack. Each module exposes clear initialization and frame-processing functions, which simplifies reuse and testing.

The application layer contains the high-level control logic and state machines. It coordinates when acquisition starts and stops, reacts to DMA half-complete and complete callbacks, schedules LC3 encoding, and forwards compressed packets to the communication subsystem. This layer also provides hooks for integration with the rest of the multisensor platform, such as synchronization with other physiological signals or user-level commands.

The separation between application logic, middleware/driver components, and HAL/LL layers makes it easier to modify or extend individual parts of the system—for example, by replacing the filter design, changing codec settings, or adding

a different communication interface—without affecting the core acquisition infrastructure.

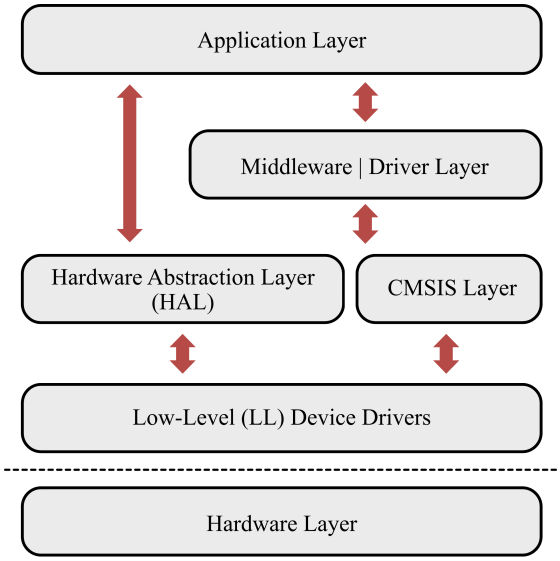


Figure 3.3: Layered firmware architecture illustrating the separation between application logic, middleware/driver components, HAL and CMSIS abstraction, low-level device drivers, and the underlying hardware.

3.3.2 Development Environment and Software Framework

The firmware was developed using STM32CubeIDE, which integrates peripheral configuration, code generation, compilation, and debugging. Peripheral initialization and clock tree configuration were generated with STM32CubeMX, ensuring consistent setup of the SAI/I^2S interface, DMA channels, timers, and GPIOs.

HAL drivers are functionality-oriented components associated with individual peripherals such as SAI/I^2S, DMA, timers, GPIO, and the BLE radio. Internally, they may rely on low-level drivers for finer control, but this remains transparent to the upper layers. This framework reduces development effort and improves portability across STM32 devices, while still allowing precise timing control when required by real-time audio processing.

3.3.3 Structured Design Approach

The firmware design follows structured programming principles. The implementation is decomposed into software items and units with clearly defined responsibilities and interfaces. The behavior of the system is controlled through explicit state

machines, which provide a predictable execution flow and simplify reasoning about timing, error handling, and test coverage. This approach is consistent with best practices for medical device software, where modular decomposition and clear traceability from requirements to implementation are essential for verification, validation, and long-term maintenance.

3.4 CMSIS and Digital Signal Processing Implementation

The Cortex Microcontroller Software Interface Standard (CMSIS) provides a standardized abstraction layer for Arm Cortex-M processors, including core register definitions, interrupt handling, and startup configuration [17]. In STM32 devices, CMSIS forms the foundation upon which the Hardware Abstraction Layer (HAL) and low-level drivers are built, enabling consistent peripheral configuration and predictable real-time behavior.

Beyond core abstraction, CMSIS includes the CMSIS-DSP extension, a collection of highly optimized digital signal processing routines tailored for Cortex-M architectures with optional floating-point and DSP instruction support. These routines are implemented with architecture-aware optimizations that exploit the single-cycle multiply–accumulate (MAC) instructions and floating-point unit (FPU) available in the Cortex-M33 core.

3.4.1 IIR Filter Implementation Using CMSIS-DSP

In the proposed system, CMSIS-DSP is used to implement the real-time preprocessing filter applied to the phonocardiogram signal. The filter is realized as a cascade of second-order sections (biquads), using the `arm_biquad_cascade_df1_f32` function provided by the library.

Each biquad stage implements the recursive difference equation:

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] - a_1y[n-1] - a_2y[n-2], \qquad (3.1)$$

where b_i are feedforward coefficients and a_i are feedback coefficients. The cascade structure adopted in this work is illustrated in Figure 3.4. Each second-order section is implemented using a Direct Form I realization. Cascading biquads instead of implementing a single high-order transfer function improves numerical stability compared to direct high-order implementations and reduces coefficient sensitivity.

CMSIS-DSP uses a Direct Form I structure with four state variables per section:

$$\{x[n-1], x[n-2], y[n-1], y[n-2]\}.$$

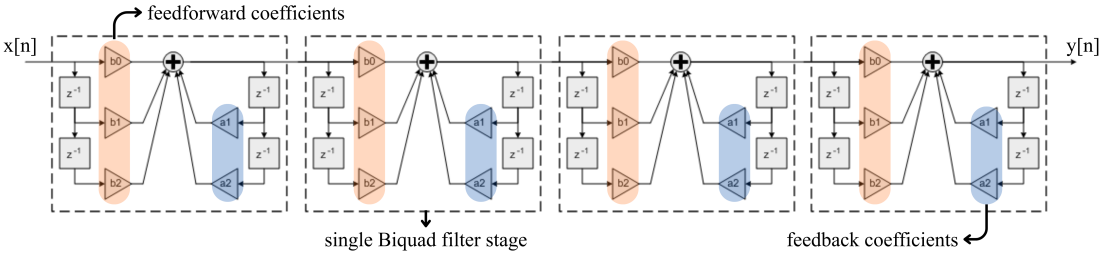


Figure 3.4: Structure of the IIR Biquad Cascade filter based on the Direct Form I (DF1) architecture. The diagram illustrates four cascaded stages, showing the feedforward (b_n) and feedback (a_n) coefficients, as well as the delay units (z^{-1}) for each single Biquad filter stage.

These state variables are stored in a dedicated state buffer that is updated after each processing block.

3.4.2 Using CMSIS-DSP

The decision to use CMSIS-DSP instead of a custom filter implementation was driven by several considerations. First, CMSIS-DSP offers strong numerical reliability, as its routines are extensively validated and widely used in industrial and medical embedded systems. In addition, the library is architecturally optimized for Cortex-M processors, resulting in a significantly lower cycle count compared to generic C implementations. This reduction in computational load directly improves energy efficiency, which is essential in continuous acquisition scenarios where dynamic power consumption must be minimized. Another important factor is portability: the same filtering code can be reused across different STM32 devices or other Cortex-M platforms without requiring modification.

From a system-level perspective, the CMSIS-DSP filter provides the necessary spectral conditioning of the PCG signal while maintaining minimal computational overhead. This ensures that sufficient CPU headroom remains available for LC3 encoding and BLE communication tasks, preserving overall system responsiveness and efficiency.

3.5 Low Complexity Communication Codec (LC3)

3.5.1 Role in the System

As introduced in Chapter 2, LC3 is the mandatory codec of Bluetooth LE Audio and is specifically designed for low-bitrate, low-latency wireless audio transmission. In the proposed wearable auscultation system, LC3 acts as the core compression

engine enabling continuous PCG streaming over a Bluetooth Low Energy (BLE) link under strict power constraints.

Within the system architecture, the LC3 encoder executes on the STM32U545 microcontroller immediately after digital preprocessing. The encoder receives filtered PCM frames and produces fixed-size compressed frames that are transmitted over BLE. Decoding is performed on the receiving device (smartphone, gateway, or PC), where the reconstructed signal can be played back, stored, or subjected to further analysis.

This section focuses on the internal operation of the LC3 encoder and its algorithmic structure, with emphasis on aspects relevant to embedded implementation. Concrete configuration parameters and firmware integration details are described in Chapter 4.

3.5.2 Encoder Architecture

LC3 is a low-delay transform codec operating on short frames (typically 10 ms, with optional 7.5 ms frames). Its architecture is optimized to balance compression efficiency, perceptual quality, and computational complexity [29].

At a high level, the encoder consists of four principal stages:

1. time–frequency transformation using a Low-Delay Modified Discrete Cosine Transform (LD-MDCT);
2. spectral envelope estimation and band-wise normalization;
3. perceptual bit allocation and quantization;
4. entropy coding and fixed-size frame packing.

Additional auxiliary modules refine the perceptual performance while maintaining low complexity. A block diagram of the encoder is shown in Figure 3.5.

Internally, LC3 is a spectral transform coder. A segment of the time-domain signal is converted into a compact spectral representation, which is then perceptually shaped and efficiently quantized before bitstream generation.

3.5.3 Time–Frequency Transformation

The input PCM signal is processed in frames of length N, determined by the sampling rate and chosen frame duration. For example, at $f_s = 16$ kHz and a 10 ms frame, $N = 160$ samples. To avoid blocking artefacts, LC3 employs a 50 %-overlapped *Low-Delay Modified Cosine Transform (LD-MDCT)*. The transform operates on windowed samples formed from the current half-frame and the previous half-frame, producing N spectral coefficients per frame. This provides

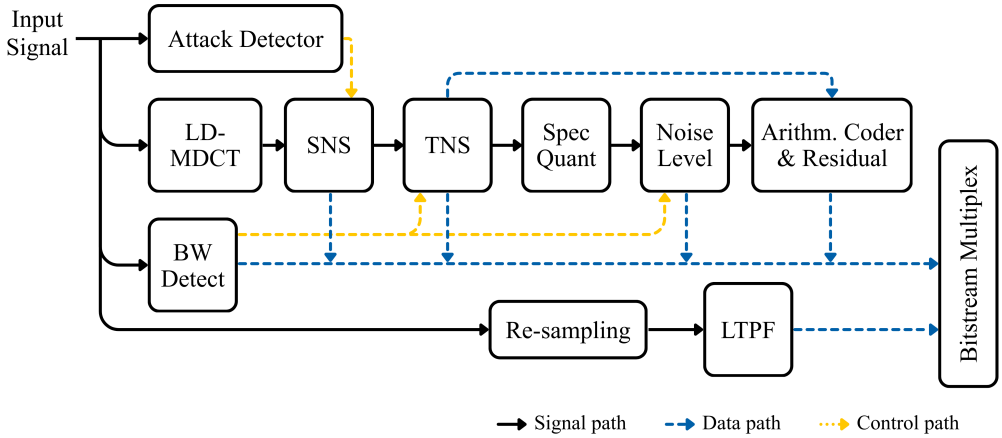


Figure 3.5: High-level block diagram of the LC3 encoder. Adapted from [29].

a compact representation in which much of the signal energy is concentrated in relatively few coefficients, facilitating efficient compression.

The LD-MDCT length scales with the sampling rate (e.g., 640 points at 48 kHz, shorter sizes at 16–24 kHz). For PCG signals, whose useful bandwidth is well below 1 kHz, operating at 16 or 24 kHz keeps the transform size moderate and reduces computational cost without losing relevant information.

The use of the LD-MDCT introduces a fixed mathematical latency known as algorithmic delay. For the chosen frame duration of 10 ms, the transform requires an additional 2.5 ms of lookahead for the overlap-add operation. Consequently, the intrinsic algorithmic delay of the LC3 codec in this configuration is strictly bounded to 12.5 ms, independent of the hardware execution speed.

3.5.4 Spectral Noise Shaping and Envelope Coding

After transformation, the spectral coefficients are processed to reduce perceptual quantization artifacts. LC3 employs two shaping mechanisms:

- *Spectral Noise Shaping (SNS)*, which adjusts the spectral envelope using vector quantization techniques;
- *Temporal Noise Shaping (TNS)*, which redistributes quantization noise along the time axis to reduce pre-echo artifacts.

In the SNS stage, coefficients are grouped into frequency bands that approximately follow perceptual resolution. For each band b, a scale factor S_b is computed based on band energy. These scale factors represent the coarse spectral envelope and are themselves quantized and transmitted as side information.

The SNS module uses a two-stage vector quantization structure: an initial split vector quantizer followed by a low-complexity pyramid vector quantizer. This hierarchical structure captures dominant spectral patterns efficiently while maintaining low computational cost.

TNS further refines the perceptual quality by shaping quantization noise in the temporal domain. For signals containing transient components—such as the sharp onset of S1 and S2—this mechanism helps reduce audible distortion after decoding.

3.5.5 Bit Allocation and Quantization

Once the spectral envelope is determined and coefficients are normalized, LC3 performs scalar quantization with a dead-zone characteristic. The *quantization* step size in each band is controlled by a bit-allocation algorithm driven by:

- the total bit budget available for the frame (set by the target bitrate),
- the band energies reflected by the scale factors,
- a simplified perceptual model estimating tolerable distortion.

Bits are iteratively distributed among bands until the predefined bitrate constraint is satisfied. Bands with higher energy or greater perceptual relevance receive finer quantization, while less significant bands are coded more coarsely.

Spectral coefficients quantized to zero are not left empty at the decoder. Instead, a noise substitution mechanism reconstructs shaped noise whose level is determined by a dedicated *Noise Level estimator*. This approach reduces artificial spectral holes and improves subjective quality.

A *Bandwidth (BW) detector* module identifies unused high-frequency regions and limits processing to active spectral areas, further reducing complexity and improving coding efficiency.

3.5.6 Entropy Coding and Frame Formatting

The quantized spectral indices and the quantized scale factors are further compressed using *entropy coding* (Huffman and run-length coding with tables tuned to typical audio statistics). Together with side information such as bit-allocation parameters and a frame header, they are packed into a fixed size compressed frame. For a given bitrate R and frame duration T, the frame size in bytes is

$$\text{Frame size (bytes)} = \frac{R \times T}{8}.$$

For example, at $R = 32$ kbit/s and $T = 10$ ms, each compressed frame contains exactly 40 bytes. This constant frame size greatly simplifies BLE packetization and ensures deterministic timing behavior in the embedded firmware.

3.5.7 Decoder Operation

The LC3 decoder can be configured to output linear PCM audio at various bit depths depending on the specific requirements of the final end-user application. However, a resolution of 16 bits per sample is highly recommended. As the standard format in the audio industry, a 16-bit output ensures seamless compatibility with a wide range of receiving devices and software applications, while providing a dynamic range that is more than sufficient to preserve the diagnostic fidelity of the PCG.

The LC3 decoder performs the inverse sequence of operations: it parses the incoming frame, decodes the entropy-coded fields, reconstructs the quantized scale factors and spectral indices, applies inverse quantization to obtain normalized MDCT coefficients, multiplies by the scale factors to recover the MDCT spectrum, and finally applies the inverse MDCT with overlap-add to reconstruct the time-domain PCM signal. Because the decoder uses the same window and overlap rules, the signal is perfectly reconstructed up to quantization error. The computational cost of the decoder is comparable to that of the encoder and is easily handled by smartphones or gateway devices.

In addition to inverse quantization and inverse MDCT reconstruction, the decoder optionally applies a Long-Term Post Filter (LTPF). The LTPF is a pitch-based enhancement stage operating at a fixed internal sampling rate (12.8 kHz), which reinforces periodic components of voiced signals by exploiting long-term correlations. The associated pitch parameters are estimated in the encoder and transmitted as side information.

Although originally designed to enhance speech signals, the LTPF may also slightly reinforce quasi-periodic components in heart sounds. However, since it operates exclusively on the decoder side, it does not increase computational load on the wearable device.

3.5.8 Suitability for Phonocardiography

Although LC3 was originally designed for speech and music, several features make it particularly suitable for phonocardiography:

- *Low bitrate operation*: LC3 maintains good perceptual quality down to 16–32 kbit/s per channel, which is well matched to the relatively narrow bandwidth of PCG signals.

- *Short frame length*: 10 ms frames keep end-to-end latency below about 30 ms, which is important for interactive remote auscultation and synchronisation with other modalities.

- *Low complexity*: the encoder runs in real time on the STM32U545 while leaving CPU margin for preprocessing, BLE stack, and other tasks.
- *Standardised bitstream*: interoperability with any LE Audio-capable device removes the need for proprietary decoders and simplifies integration into telehealth platforms.

The underlying idea of spectral quantisation with perceptual bit allocation is conceptually similar to dedicated PCG compression schemes based on codebooks and residual coding [20, 1], but LC3 offers the additional benefit of being an open, widely adopted standard.

3.6 Datasets and Evaluation Metrics

To assess the performance of the proposed audio driver, both public phonocardiogram datasets and custom recordings are used. The datasets include examples of normal heart sounds as well as murmurs and other pathological findings, under both relatively clean and noisy conditions. This variety allows evaluation of the system's ability to preserve diagnostic information across different acoustic scenarios.

The recorded audios used for the analysis were taken from the public heart sound database (Bentley et al., 2011) [30]. This dataset was selected because, among the available collections, it provides recordings at the highest sampling rate (44 kHz). A higher sampling rate ensures greater temporal and spectral resolution, which is essential for performing accurate frequency-domain analysis of cardiac sounds. [1]

3.6.1 Performance indicators

The performance of the audio pipeline is evaluated using a set of quantitative indicators that cover signal fidelity, compression efficiency, computational cost, and clinically relevant spectral changes. Let $x[n]$ denote the original PCG signal and $\hat{x}[n]$ the processed or reconstructed signal, both of length N.

Signal fidelity metrics

Classical waveform-based metrics quantify how close the processed or compressed signal is to the original PCG. The signal-to-noise ratio (SNR) compares the energy of the reference signal to the energy of the error:

[1] `https://istethoscope.peterjbentley.com/heartchallenge/index.html`

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{n=0}^{N-1} x[n]^2}{\sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2} \right) \text{ dB} \tag{3.2}$$

The root mean square error (RMSE) between $x[n]$ and $\hat{x}[n]$ is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2} \tag{3.3}$$

The Peak Signal-to-Noise Ratio (PSNR) provides a measure of the maximum possible signal-to-noise ratio, based on the peak signal value:

$$\text{PSNR} = 20 \log_{10} \left(\frac{\max |x|}{\text{RMSE}} \right) \text{ dB}, \tag{3.4}$$

where $\max |x|$ is the maximum absolute value in the original signal $x[n]$, and RMSE is the root mean square error already defined above. PSNR normalizes distortion by peak amplitude rather than total energy (unlike SNR), making it suitable for signals with known dynamic range.

The Percent Root-mean-square Difference (PRD) [20] expresses the distortion as a percentage of the original signal energy:

$$\text{PRD} = 100 \sqrt{\frac{\sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2}{\sum_{n=0}^{N-1} x[n]^2}} \, \%. \tag{3.5}$$

Envelope correlation To assess morphological preservation independent of amplitude scaling and phase shifts, the Pearson correlation coefficient (r) is computed between the Hilbert envelopes of the original and processed signals. The Hilbert envelope is computed after bandpass filtering (20–400 Hz) to isolate cardiac components, with a subsequent low-pass filter (cutoff 20 Hz) to obtain the smooth envelope $e[n]$. The envelope $e[n]$ is obtained as:

$$e[n] = |x[n] + j \cdot \mathcal{H}\{x[n]\}| \tag{3.6}$$

where $\mathcal{H}\{\cdot\}$ denotes the Hilbert transform. The envelope correlation is then:

$$r_{env} = \frac{\sum_{n=0}^{N-1} (e_x[n] - \bar{e}_x)(e_{\hat{x}}[n] - \bar{e}_{\hat{x}})}{\sqrt{\sum_{n=0}^{N-1} (e_x[n] - \bar{e}_x)^2} \sqrt{\sum_{n=0}^{N-1} (e_{\hat{x}}[n] - \bar{e}_{\hat{x}})^2}} \tag{3.7}$$

Values close to 1.0 indicate perfect preservation of temporal morphology, even when sample-by-sample alignment is imperfect.

Morphological quality metrics

Following Mei et al. [31], the Ratio of Zero Crossings (RZC) and the Root Mean Square of Successive Differences (RMSSD) are used to assess basic signal quality and morphology. Let $A[n]$ be the heart-sound signal (or a low-frequency wavelet approximation).

The RZC is defined as the ratio of zero crossings to the signal length:

$$\mathrm{RZC} = \frac{1}{N-1} \sum_{i=1}^{N-1} \Big[\mathrm{sgn}(A[i] \cdot A[i+1]) \cap |A[i+1] - A[i]| > 0 \Big] \tag{3.8}$$

where $\mathrm{sgn}(x) = 1$ if $x \leq 0$ and 0 otherwise [31]. The RZC reflects the rate at which the signal's sign changes due to heart beats. High values indicate excessive high-frequency noise.

The RMSSD captures the root mean square of successive differences:

$$\mathrm{RMSSD} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2} \tag{3.9}$$

where x represents the approximate coefficients of the two-level wavelet decomposition sequence [31]. Segments with RZC and RMSSD outside typical ranges are considered degraded (e.g. dominated by noise or excessively smoothed) and can be excluded from further analysis.

Spectral fidelity metrics

Log-Spectral Distance (LSD) The Log-Spectral Distance measures the difference between the power spectra of two signals on a logarithmic scale, aligning with human auditory perception. For power spectral densities $P_x(f)$ and $P_{\hat{x}}(f)$ estimated via Welch's method, LSD is defined as:

$$\mathrm{LSD} = \sqrt{\frac{1}{M} \sum_{k=0}^{M-1} \left[10 \log_{10} \frac{P_x(f_k)}{P_{\hat{x}}(f_k)} \right]^2} \quad \mathrm{dB} \tag{3.10}$$

where M is the number of frequency bins. Lower LSD values indicate better spectral preservation.

Spectral Distortion (SD) The Spectral Distortion quantifies the root-mean-square deviation between logarithmic power spectra:

$$\mathrm{SD} = \sqrt{\frac{1}{M} \sum_{k=0}^{M-1} \left(10 \log_{10} P_x(f_k) - 10 \log_{10} P_{\hat{x}}(f_k) \right)^2} \quad \mathrm{dB} \tag{3.11}$$

Magnitude-Squared Coherence The coherence function $C_{xy}(f)$ measures the linear relationship between two signals as a function of frequency:

$$C_{xy}(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)} \tag{3.12}$$

where G_{xy} is the cross-spectral density and G_{xx}, G_{yy} are the auto-spectral densities. Coherence ranges from 0 (no correlation) to 1 (perfect linear relationship). The band-averaged coherence $\bar{C}_{xy}$ is computed as the mean over clinically relevant frequency ranges.

Temporal and clinical metrics

Heart rate estimation error The relative error in heart rate (HR) estimation between original and processed signals:

$$\mathrm{HR}_{\mathrm{error}} = 100 \times \frac{|\mathrm{HR}_{\mathrm{original}} - \mathrm{HR}_{\mathrm{processed}}|}{\mathrm{HR}_{\mathrm{original}}} \quad \% \tag{3.13}$$

Heart rates are computed from the median inter-beat intervals detected via peak finding on the Hilbert envelope.

S1–S2 interval error The relative error in the temporal separation between consecutive S1 and S2 sounds:

$$\text{S1-S2}_{\mathrm{error}} = 100 \times \frac{|\Delta t_{\text{S1-S2, orig}} - \Delta t_{\text{S1-S2, proc}}|}{\Delta t_{\text{S1-S2, orig}}} \quad \% \tag{3.14}$$

Compression and computational efficiency

The compression efficiency is computed as the ratio between the total number of bits required to transmit the original signal divided by those required for the transmission of the compressed one. Compression Ratio (CR) is:

$$\mathrm{CR} = \frac{\textit{original signal bits}}{\textit{compressed signal bits}} \tag{3.15}$$

Following [21], a simple Quality Score (QS) combines CR and PRD:

$$\mathrm{QS} = \frac{\mathrm{CR}}{\mathrm{PRD}} \tag{3.16}$$

Higher QS values indicate better compression efficiency for a given distortion level.

If T_{sig} is the signal duration and T_{comp} the time required to compress it, the Real-Time Factor (RTF) is:

$$\text{RTF} = \frac{T_{\text{comp}}}{T_{\text{sig}}} \tag{3.17}$$

An RTF below 1 indicates that the encoder can run faster than real time.

The computational efficiency (CE) is defined as:

$$\text{CE} = \frac{\text{CR}}{T_{\text{comp}}} \tag{3.18}$$

which measures how much compression is obtained per unit of computation time [21]. At the system level, metrics include end-to-end latency, CPU cycles per process on the STM32U545, memory footprint, and energy consumption estimated following approaches used in previous work on lightweight biometric compression, from number of operations and cycles required by the algorithms using Cortex-M33 references [1].

CPU load and throughput The CPU utilization percentage for real-time processing:

$$\text{CPU}_{\text{load}} = 100 \times \frac{T_{\text{processing}}}{T_{\text{frame}}} \quad \% \tag{3.19}$$

where T_{frame} is the audio frame duration (e.g., 10 ms) and $T_{\text{processing}}$ is the measured execution time.

Memory footprint Static RAM usage for algorithm state (buffers, coefficients) and stack requirements during execution, reported in kilobytes (kB).

Latency / Algorithmic Delay End-to-end algorithmic latency, defined as the delay from input sample availability to output sample production, measured in milliseconds (ms). The total latency is composed of:

$$T_{\text{total}} = T_{\text{frame}} + T_{\text{lookahead}} + T_{\text{processing}} \tag{3.20}$$

where T_{frame} is the frame duration (10 ms), $T_{\text{lookahead}}$ is the MDCT overlap (2.5 ms for LC3), $T_{\text{processing}}$ is the measured execution time.

3.6.2 Spectral analysis in clinically relevant bands

Previous studies have demonstrated that the energy distribution across specific frequency bands serves as a robust feature set for classifying heart sounds [32]. Building upon this principle; beyond global distortion measures, this thesis proposes a more clinically oriented evaluation based on the spectral content of heart sounds. The goal is to verify whether LC3 compression preserves the energy distribution in frequency regions that are known to carry diagnostic information for phonocardiography.

Table 2.1 (see Chapter 2) summarizes typical frequency ranges for heart sounds and interference sources [13, 33]. Building on this, the PCG is analyzed in several bands associated with different components of the cardiac cycle:

- Very low frequencies (below 20 Hz): baseline drift, motion artefacts, and subaudible components.
- S1 low band (20–70 Hz): main low-frequency content of the first heart sound.
- S1/S2 mid band (70–150 Hz): additional energy of S1 and S2, often used for timing and envelope detection.
- High S2 and soft murmurs (150–400 Hz): higher-frequency components of the second heart sound and soft systolic or diastolic murmurs.
- Murmur band (400–1000 Hz): broadband turbulent flow components associated with pathological murmurs.
- Out-of-band region (above 1000 Hz): frequencies that are usually not clinically relevant for PCG and can reveal codec artefacts or aliasing.

For each band, the power spectral density (PSD) of the original and reconstructed signals is estimated using Welch's method. The following band-wise indicators are then computed:

- *Band energy* E_b: integrated PSD within band b.
- *Relative band energy change* ΔE_b (in %): difference between original and reconstructed band energies, normalized by the original energy.

$$\Delta E_b = 100 \times \frac{E_b^{\text{processed}} - E_b^{\text{original}}}{E_b^{\text{original}}} \quad \% \tag{3.21}$$

where $E_b = \int_{f_1}^{f_2} P_{xx}(f)\, df$ is the integrated PSD in band $b = [f_1, f_2]$.

- *Spectral distortion in band b*: average deviation in dB between the two PSDs over the band.

From a clinical perspective, small changes in ΔE_b in the S1/S2 bands indicate that the main heart sounds are preserved, while significant losses in the murmur band (400–1000 Hz) would suggest a risk of missing pathological murmurs. In contrast, an increase in the out-of-band energy may reveal codec-induced artifacts that could confuse automated analyzes or human interpretation.

This band-wise spectral analysis is used in Chapter 5 to complement global metrics such as PRD and SNR, and represents a key contribution of this thesis towards a more clinically meaningful evaluation of audio codecs for cardiac auscultation.

Table 3.3: Performance metrics used for audio pipeline evaluation.

Category	**Metric**	**Equation**	**Reference**
Signal fidelity	SNR	(3.2)	[21, 22, 34, 35]
	RMSE	(3.3)	[1, 35]
	PSNR	(3.4)	[21, 22]
	PRD	(3.5)	[20, 21, 36, 22, 35]
	Envelope Corr.	(3.7)	Thesis contribution
Morphology	RZC	(3.8)	[31]
	RMSSD	(3.9)	[31]
Spectral	LSD	(3.10)	[37]
	Spectral Distortion	(3.11)	Standard definition
	Coherence	(3.12)	Standard definition
	ΔE_b	See Sec. 3.6.2	Thesis contribution
Temporal	HR Error	(3.13)	Thesis contribution
	S1-S2 Error	(3.14)	Thesis contribution
Compression	CR	(3.15)	[21, 20, 22, 3]
	QS	(3.16)	[21, 22]
	RTF	(3.17)	[3]
	CE	(3.18)	[21, 22]
	CPU Load	(3.19)	Standard definition
	Memory	kB (reported)	Standard practice

Chapter 4

Firmware Implementation

This chapter details the software architecture and the specific firmware routines developed for the STM32U545 microcontroller. As summarized in Figure 4.1, the embedded application is designed around a sequential data processing pipeline: continuous audio acquisition from the MEMS sensor, real-time digital preprocessing, and LC3 data compression, concluding with the formatting of the payload for external transmission.

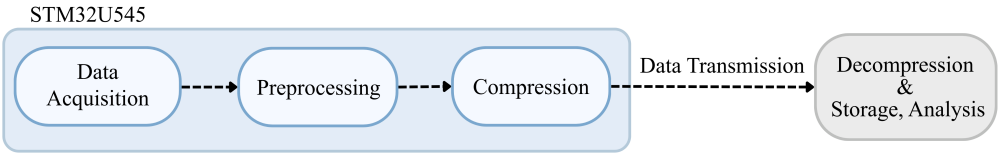


Figure 4.1: Simplified block diagram of the current implemented pipeline, focusing on Data Acquisition, Preprocessing, and Compression on the STM32U545, followed by generic data transmission for remote Decompression, Storage, and Analysis.

4.1 Audio Driver Implementation: Double Buffering and State Machine

The audio front-end operates as a deterministic state machine with three primary states: `IDLE`, `MIC_ON`, and `MIC_OFF`. In the `IDLE` state, the MCU configures the clock tree, initializes the Serial Audio Interface (SAI), and sets up the associated Direct Memory Access (DMA) channel. Once initialization is complete, the system transitions to `MIC_ON`, where audio acquisition runs continuously through SAI using the I^2S protocol and DMA transfers. In this state, the MCU also performs real-time processing and LC3 encoding of the incoming audio stream. For testing

purposes, the `MIC_OFF` state is used to stop acquisition after a predefined amount of data has been captured, whereas in the real-time application the system remains continuously in `MIC_ON`.

4.1.1 DMA Circular Buffer and Ping-Pong Strategy

To minimize CPU load during acquisition, samples are routed from the SAI data register directly to the RAM using DMA. By operating in circular mode, the DMA controller continuously refills the buffer: upon reaching the end of the memory block, the pointer automatically wraps around to the beginning, preventing acquisition gaps.

With DMA, the CPU is not involved in per-sample transfers and only reacts to DMA events, so to safely process the data while it is being continuously written, the receive buffer is logically divided into two equal halves, implementing a "ping-pong" (double-buffering) scheme. The firmware relies on two hardware-triggered DMA interrupts:

- *Half-Transfer Callback:* Indicates that the first half of the buffer is full and ready for processing.
- *Transfer-Complete Callback:* Indicates that the second half of the buffer is full and ready.

This mechanism allows the MCU's processor core to execute the DSP and encoding algorithms on one half of the buffer while the DMA hardware concurrently fills the other half. Figure 4.2 illustrates this concurrent acquisition and processing loop.

4.1.2 Timing and Real-Time Constraints

The system processes audio in frames of 10 ms. As shown in Figure 4.3, the half-transfer and transfer-complete callbacks occur periodically and provide a deterministic schedule for block-based processing. To guarantee real-time operation without buffer overruns, the total execution time for preprocessing and encoding must be strictly less than the 10 ms frame duration.

Beyond simply avoiding buffer overruns, the system must also adhere to strict end-to-end latency requirements dictated by the clinical application. In interactive remote auscultation, the acoustic feed is often accompanied by a real-time visual representation, such as an active electrocardiogram (ECG) trace or a live video feed of the patient. In such multimodal scenarios, audiovisual synchronization is critical. According to the International Telecommunication Union (ITU-R BT.1359-1) standards for the relative timing of sound and vision, the human threshold of

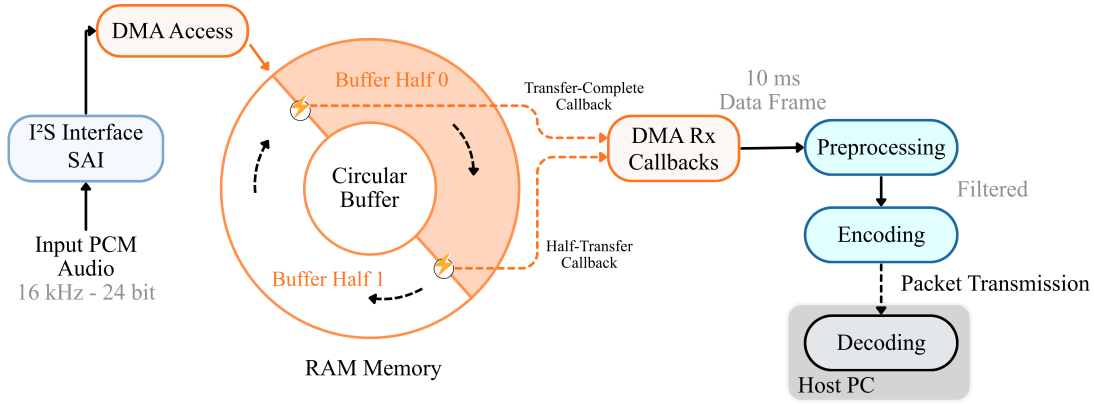


Figure 4.2: Detailed firmware memory architecture for real-time audio acquisition and processing. The diagram illustrates the core of the ping-pong strategy: while the DMA controller actively fills Buffer Half 1 with incoming I2S data, the MCU simultaneously reads and processes Buffer Half 0 to generate the current encoded audio frame, which is then available for transmission to the Host PC.

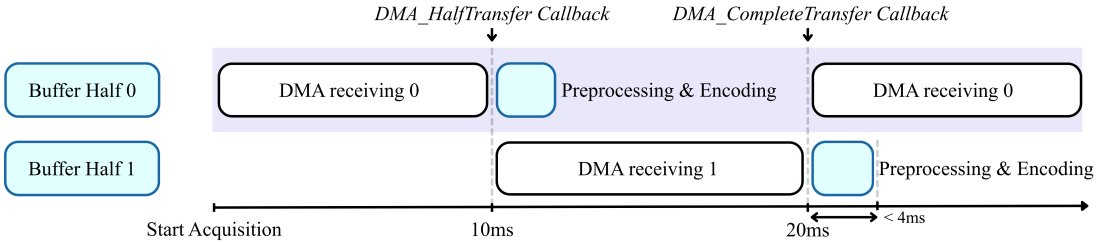


Figure 4.3: Timing diagram of the DMA-based ping-pong buffering mechanism. The DMA fills one half-buffer (hardware task) while the MCU processes the previous frame in the other half-buffer (software task).

detectability for audio delay is approximately 45 ms [38]. If the audio delay exceeds this limit, the user begins to perceive a noticeable desynchronization between the visual event and the corresponding sound, severely impairing the diagnostic experience.

Therefore, the system is designed with a strict latency budget, targeting a conservative threshold of 30 ms to ensure it remains comfortably within the undetectable plateau. In this implementation, the combined preprocessing and LC3 encoding pipeline completes in less than 4 ms, which is safely below the 10 ms frame interval. This highly efficient execution not only prevents data loss but also ensures that the total on-device processing delay—even when accounting for the codec's intrinsic algorithmic latency—remains well below the required 30 ms

perceptual threshold, leaving ample margin for subsequent Bluetooth packetization and air-interface transmission.

4.1.3 System Configuration

Sampling Frequency Selection The sampling frequency selection involves balancing signal fidelity against computational load and transmission bandwidth. According to the Nyquist-Shannon sampling theorem, the sampling rate must be at least twice the highest frequency component of interest to avoid aliasing. Based on the frequency characteristics of heart sounds, a sampling rate of 16 kHz was selected. This frequency, with a Nyquist frequency of 8 kHz, ensures the system captures all clinically relevant cardiac components with substantial margin. Moreover, LC3 supports 16 kHz natively, eliminating the need for computationally expensive sample rate conversion algorithms. Higher rates (such as 32 kHz or 48 kHz) would drastically increase the DMA bandwidth, memory footprint, and encoder workload without adding any diagnostic value for phonocardiography.

Furthermore, the bandwidth provided by the selected sampling frequency extends well beyond the requirements for cardiac auscultation. This wide bandwidth ensures that the hardware is inherently capable of capturing respiratory and lung sounds, whose frequency content typically ranges from 200 Hz to 2000 Hz. Consequently, the system can be adapted in the future for multimodal cardiopulmonary monitoring simply by modifying the digital filtering stage and software parameters, without requiring any hardware modifications.

Clock System and I²S Configuration The STM32U545 clock tree is configured to operate at its maximum core frequency (160 MHz) to minimize per-frame processing time. The SAI peripheral derives its clock from a dedicated PLL, precisely tuned to generate the 16 kHz sampling frequency required by the MEMS microphone. The ICS-43434 microphone outputs 24-bit samples in a left-justified format (MSB first). Accordingly, the SAI data format is configured with a 32-bit frame length, 32-bit slot size, and 24-bit data size. This alignment preserves the full 24-bit dynamic range of the acoustic sensor while maintaining standard I²S protocol compatibility.

Buffer Sizing The DMA buffer length is explicitly sized to match the LC3 codec framing requirements. At a 16 kHz sampling rate, a 10 ms audio frame contains:

$$N = f_s \cdot T = 16000 \cdot 0.01 = 160 \text{ samples}$$

To trigger an interrupt exactly every 10 ms using the ping-pong strategy, the total circular buffer is sized to hold two complete frames:

$$\text{Buffer Size} = 2N = 320 \text{ samples}$$

4.2 Preprocessing on MCU (CMSIS-DSP)

4.2.1 Filter Requirements and Target Response

Before LC3 compression, the raw PCM signal undergoes digital preprocessing to enhance the fundamental cardiac components (S1 and S2 sounds, typically between 20 and 250 Hz) while attenuating high-frequency ambient noise.

The preprocessing stage follows the typical Infinite Impulse Response (IIR) workflow: raw PCM samples enter the filter, a cascade of biquads applies the frequency shaping, and the resulting signal is passed to the LC3 encoder. The goal is to apply a targeted gain within the 10–400 Hz band while suppressing out-of-band noise. The design must also respect the constraints imposed by the real-time system: the group delay must remain compatible with the 10 ms frame structure, and the implementation must be computationally lightweight to ensure that filtering, encoding, and BLE communication fit comfortably within the CPU processing window.

4.2.2 Filter Design in MATLAB

The desired frequency response was modeled in MATLAB. Based on the physiological targets, a custom magnitude curve was defined using specific normalized frequency points and their corresponding linear gains. The cardiac band (10–400 Hz) was targeted with gain, while the sub-audible DC region and the high-frequency transition band (above 1000 Hz) were heavily attenuated.

The `yulewalk` function was employed to design an 8th-order IIR filter that optimally approximates this arbitrary magnitude response. An 8th-order filter was selected as the ideal compromise: it provides sufficient spectral selectivity without introducing excessive computational overhead for the microcontroller. The theoretical stability of the design was verified by computing the filter's poles and confirming that they all lie strictly inside the unit circle. A comprehensive evaluation of the filter's frequency response and its clinical neutrality on actual PCG signals is detailed in Chapter 5.

Conversion to Second Order Sections (SOS)

Directly implementing an 8th-order transfer function using a single difference equation is highly susceptible to numerical instability and quantization noise, particularly in embedded systems. To guarantee absolute stability, the 8th-order polynomial was decomposed into a cascade of four independent second-order filters (biquads).

This conversion was performed in MATLAB using the `tf2sos` function, generating a matrix of Second Order Sections (SOS). This mathematical transformation

is not only an engineering best practice but a strict requirement for utilizing the optimized biquad cascade functions within the CMSIS-DSP library.

Mapping to CMSIS Array Format

To implement the filter on the STM32U545, the MATLAB SOS coefficients had to be mapped to the memory layout expected by the CMSIS-DSP library. For a Direct Form I biquad implementation, CMSIS-DSP requires the coefficients of each stage to be stored sequentially in a 32-bit floating-point array in the following specific order: $\{b_0, b_1, b_2, -a_1, -a_2\}$

The SOS matrix was extracted, the global gain was applied to the first section, the signs of the denominator coefficients were inverted, and the output was formatted as a C-compatible `float32_t` array ready to be pasted into the STM32 firmware. (The complete MATLAB filter design and exportation script is provided in Appendix A).

4.2.3 Firmware Integration and Real-Time Execution

Library Importation

The default STM32Cube firmware package does not include the latest DSP optimizations required for the Cortex-M33 architecture. Therefore, the most recent version of the CMSIS-DSP library was manually downloaded from the official repository and integrated into the STM32CubeIDE project. The source files (`.c` and `.h`) were organized into a dedicated workspace folder, and the build paths were updated to ensure the compiler could correctly link the highly optimized `arm_biquad_cascade` functions.

Filter Implementation Comparison: DF1 vs. DF2T vs. Fixed-Point

To select the most efficient and accurate filter implementation for the STM32U545, a performance evaluation was conducted directly on the target hardware . Three different algorithms from the CMSIS-DSP library were evaluated for the 8th-order cascade:

- `arm_biquad_cascade_df1_f32`: Direct Form I structure using 32-bit floating-point math. It requires 4 state variables per biquad stage.
- `arm_biquad_cascade_df2T_f32`: Direct Form II Transposed structure using 32-bit floating-point math. It is more memory-efficient, requiring only half the state variables of DF1.
- `arm_biquad_cascade_df1_q31`: Direct Form I structure using 32-bit fixed-point math (Q31 format).

The results, summarized in Table 4.1, provide insights into execution time, memory usage, and numerical behavior under real-time constraints.

Table 4.1: Performance benchmark of IIR filter implementations on the STM32U545.

Metric	**df2T_f32**	**df1_f32**	**df1_q31**
Real-time Performance			
Avg. Exec / Frame (μs)	218.01	218.56	1023.02
CPU Load (%)	2.18	2.19	10.23
Filter Benchmark			
Avg. Cycles / Frame	34,881	34,970	163,683
Cycles / Sample / Stage	54.50	54.64	255.75
Memory & Format			
RAM Usage (Bytes)	112	144	112
Coefficient Format	float32	float32	q31

The benchmarking demonstrated that the floating-point implementations (`_f32`) executed nearly five times faster than the fixed-point (`_q31`) version. This is because the STM32U545 features a hardware Floating-Point Unit (FPU), allowing it to process native `float32` instructions in a single cycle, whereas the Q31 math required heavier software emulation or complex instruction handling. Consequently, the `q31` variant was discarded to preserve CPU headroom.

While both `df1_f32` and `df2T_f32` showed nearly identical computational speed, a rigorous time-domain analysis of the filtered outputs revealed a critical difference. As shown in Figure 4.4, the DF2T structure introduced a slight temporal distortion and phase delay compared to the original signal dynamics. The classical Direct Form I (DF1) implementation, however, maintained excellent morphological fidelity without phase artifacts. Therefore, `arm_biquad_cascade_df1_f32` was selected as the optimal preprocessing function.

Block Processing and Persistent State

The chosen CMSIS-DSP function, `arm_biquad_cascade_df1_f32`, is designed for block-based processing, aligning perfectly with the DMA ping-pong strategy. Instead of filtering sample-by-sample, the MCU waits for the DMA half-transfer interrupt and processes an entire 10 ms block (160 samples) in a single function call, reducing CPU context-switching overhead.

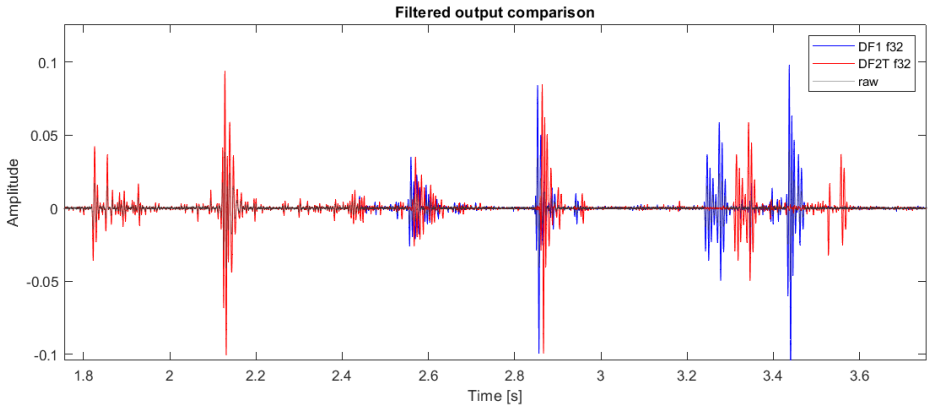


Figure 4.4: Time-domain comparison between the DF1 and DF2T filter implementations. The DF2T structure introduced undesirable temporal distortion, leading to the selection of the DF1 architecture.

Because audio is a continuous stream divided into discrete blocks, the filter must remember its mathematical state (the previous inputs and outputs) across block boundaries. CMSIS-DSP manages this using a dedicated state vector (`pState`). For a Direct Form I structure, each biquad requires 4 state variables. Thus, for the 4-stage cascade, a 16-element floating-point array is allocated in RAM. This state vector is preserved between function calls, ensuring that the filter output remains mathematically continuous and free of click artifacts at the 10 ms frame boundaries.

4.3 LC3 Codec Implementation and Integration

4.3.1 Library Importation and PC-Side Compilation

To integrate the LC3 codec into the embedded firmware, the official reference implementation provided by Google (`liblc3`) was utilized. The source code was imported into the STM32CubeIDE project within a dedicated *liblc3* directory, keeping the core algorithmic files completely decoupled from the microcontroller's HAL drivers.

To establish a mathematical ground truth for subsequent validation, the same C-based source files were compiled into standalone executables on a desktop PC using GCC via Git Bash. This allowed for off-target encoding and decoding of reference `.wav` files directly from the command line interface. This PC-side implementation ensured that the codec's intrinsic algorithmic distortion could be isolated and analyzed independently of any embedded hardware constraints.

4.3.2 Integration into the Audio Driver

In the proposed system, the LC3 encoder is invoked immediately after the preprocessing stage. The audio driver maintains a ping-pong buffer of PCM samples; whenever a full frame (e.g. 160 samples at 16 kHz) becomes available, it is passed to the LC3 encoder library. The encoder returns a compressed frame of predetermined size (for example, 40 bytes at 32 kbit/s and 10 ms), which is then forwarded to the BLE transport layer. On the receiving side, the LC3 decoder reconstructs the PCM stream, which can be played back or stored for clinical review. The exact configuration parameters (sampling rate, frame duration, bitrate) are chosen based on the trade-offs between audio fidelity, power consumption, and latency, and are detailed in Chapter 5.

4.3.3 LC3 Encoder Configuration on MCU

For the embedded implementation, the LC3 encoder was initialized using the official C API provided by the `liblc3` library. The algorithm was configured to process audio at a sampling rate of 16 kHz (`srate_hz = 16000`) with a frame duration of 10 ms (`frame_us = 10000`). The High-Resolution mode was explicitly disabled (`hrmode = false`), as the standard LC3 resolution is perfectly adequate for the frequency range of cardiac acoustics.

Because the ICS-43434 microphone natively outputs a 24-bit data stream, the encoder was configured to accept 24-bit PCM input data (`pcm_fmt = LC3_PCM_FORMAT_S24`) over a single mono channel (`nchannels = 1`). Even though the initial input is a 24-bit high-resolution stream, allowing the system to capture the maximum dynamic range, the resulting compressed payload can be reconstructed at different resolutions on the decoder side depending on the specific needs of the end application. Nonetheless, a standard 16-bit reconstruction is typically recommended (as detailed in Section 3.5.7).

To ensure deterministic execution and avoid memory fragmentation during real-time operation, dynamic memory allocation (`malloc`) was avoided. Instead, the persistent state required by the LC3 encoder (including the MDCT overlap buffers) was statically allocated in the MCU's SRAM as a dedicated byte array (`encoder_memory`). This array was explicitly aligned to 4-byte boundaries using the GCC `__attribute__((aligned(4)))` directive, guaranteeing memory-safe and highly efficient 32-bit memory accesses by the Cortex-M33 core.

The integration with the audio driver is seamless: upon receiving a DMA callback, the 160-sample preprocessed PCM frame is passed directly to the LC3 encode function. The algorithm applies the MDCT overlap-add logic, executes the perceptual bit allocation, and outputs a highly compressed binary payload.

An Important Parameter: The Bitrate

The most critical setting in the LC3 configuration is the target bitrate, as it defines the strict "bit budget" the encoder is allowed to spend on each audio frame.

For this system, a nominal operating bitrate of 32 kbps was selected. This means that for every 10 ms frame, the encoder is restricted to a budget of exactly 320 bits (40 bytes). The encoder must intelligently allocate these 320 bits to represent the most perceptually important frequencies of the original signal.

To put this in perspective, the original raw frame contains 160 samples of 24-bit audio, which equals 3840 bits of data. By restricting the output to 320 bits, the LC3 encoder achieves a 12:1 compression ratio. This specific payload size is strictly bounded to exactly 40 bytes per frame, making it ideal for encapsulation into standard Bluetooth Low Energy (BLE) packets. The comprehensive clinical and mathematical justification for selecting 32 kbps as the optimal operating point for preserving diagnostic phonocardiogram features is extensively detailed in the experimental results in Chapter 5.

4.4 Data Transmission and Validation Interface

4.4.1 Data Transmission from Microcontroller to PC

During the prototyping phase, extracting the encoded LC3 frames from the microcontroller to the host PC was essential for algorithmic validation. Standard interfaces like UART lacked sufficient bandwidth and presented hardware pin conflicts, while implementing a full USB peripheral stack introduced unnecessary firmware complexity.

To bypass these limitations, the Serial Wire Viewer (SWV) interface was utilized as a high-speed data streaming channel. Instead of transmitting raw binary bytes—which are prone to corruption and parsing errors in the ITM console under heavy load—the LC3 payloads were converted on-the-fly into an ASCII hexadecimal representation on the MCU side. As visually summarized in Figure 4.5, this text stream was captured directly in the IDE's console and processed offline, enabling bit-exact verification of the embedded pipeline while completely avoiding the latency and packet-loss behavior of early wireless prototypes.

4.4.2 LC3 Bitstream Formatting and Decoder Requirements

For offline validation, the compressed audio was decoded on the PC using the reference LC3 decoder (`dlc3.exe`). Because this decoder requires explicit metadata to understand the raw payloads, the STM32 firmware was programmed to format

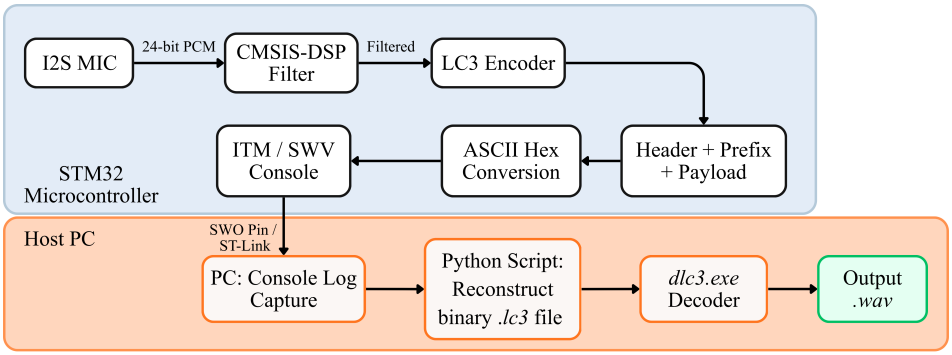


Figure 4.5: Data transmission and processing pipeline, illustrating the on-device formatting and the offline PC reconstruction workflow.

the transmitted bitstream with three mandatory components, as illustrated in Figure 4.6:

1. Global Header (18 bytes): Sent only once at startup. It contains a synchronization word (`0xCC1C`) and the decoding parameters (16 kHz, 32 kbps, 10 ms frames, mono).

2. Frame Size Prefix (2 bytes): Sent immediately before every audio frame, indicating the exact payload length (e.g., `28 00` in little-endian for a 40-byte frame).

3. Frame Payload (N bytes): The actual LC3-compressed audio data.

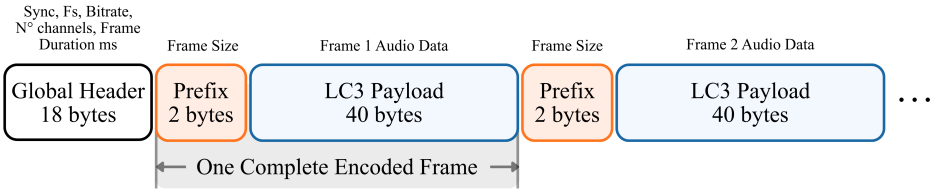


Figure 4.6: Structural breakdown of the custom `.lc3` bitstream format utilized for PC-side decoding.

Figure 4.7 visually breaks down an excerpt of the resulting SWV console stream, showing how the global header is immediately followed by consecutive LC3 audio frames and their prefixes.

Once captured on the host PC, a custom Python script (provided in Appendix B) parses this hexadecimal log, converts the tokens back into contiguous binary bytes, and reconstructs valid `.lc3` and `.wav` files. This streamlined methodology

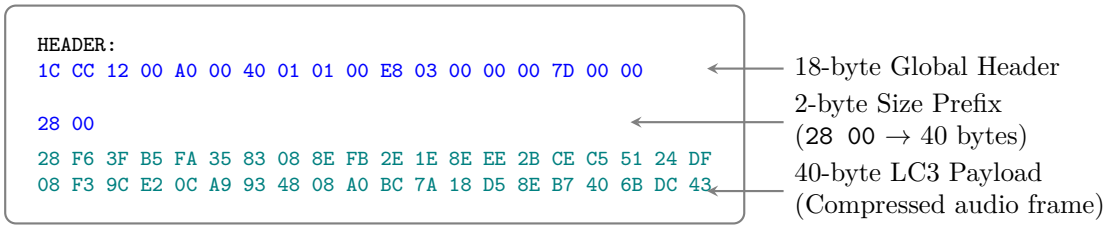


Figure 4.7: Visual breakdown of the ASCII hexadecimal stream captured via the SWV console.

guaranteed seamless compatibility with the reference decoder, allowing for robust algorithmic evaluation.

Chapter 5

Experimental Setup and Results

This chapter presents a comprehensive experimental validation of the proposed wearable phonocardiography system. The evaluation follows a progressive structure designed to verify each stage of the signal pipeline:

1. *Hardware characterization:* To quantify the baseline noise and acoustic limitations of the physical capture setup.
2. *Preprocessing validation:* To ensure the digital filtering stage enhances signal quality without altering clinical timing or morphological features.
3. *LC3 compression analysis:* To validate that the audio codec does not degrade diagnostically relevant spectral or temporal information.
4. *End-to-end system verification:* To confirm that the complete pipeline preserves diagnostic integrity under realistic acquisition conditions.

5.0.1 Test Platforms

Two complementary platforms were employed:

- Offline platform: The C-based codec (Google's open-source `liblc3` reference implementation) was compiled from source and executed to perform the encoding and decoding of the audio files allowing algorithmic characterization isolated from hardware constraints. MATLAB R2024a was utilized to perform numerical validation, signal analysis, and metric computation.
- Embedded platform: The STM32U545 Nucleo-U545RE evaluation board interfaced with the target MEMS microphone. End-to-end validation used

a controlled acoustic setup with a loudspeaker placed at a 5 cm separation distance in a quiet environment.

5.0.2 Dataset Composition

To evaluate the system across different clinical scenarios, a dataset of representative PCG signals was utilized. Table 5.1 details the distribution of the signals. All metrics reported in subsequent sections are computed per signal category and reported as mean ± standard deviation.

Table 5.1: Dataset composition and signal categorization for system validation.

Category	**Count**	**Diagnostic Purpose**
Clean PCG (high quality)	1	Baseline morphological preservation
Acceptable normal PCG	1	Mild variability and real-world robustness
Murmur PCG	1	High-frequency diagnostic content preservation
Extra heart sounds	2	Transient event preservation
Noisy PCG / Artifacts	2	Noise robustness and out-of-band rejection
White noise	1	Worst-case spectral behavior stress test

5.0.3 Evaluation Metrics

To rigorously assess the system's performance, a comprehensive set of evaluation metrics was computed systematically across the dataset. The metrics (defined mathematically in Chapter 3) are grouped into four primary categories based on their analytical objective:

Signal Fidelity

- Percentage Root-mean-square Difference (PRD)
- Signal-to-Noise Ratio (SNR)
- Peak Signal-to-Noise Ratio (PSNR)
- Root Mean Square Error (RMSE)
- Envelope correlation (Pearson's r)

Morphological Signal Quality

- Ratio of Zero-Crossings (RZC)
- Root Mean Square of Successive Differences (RMSSD)

Spectral Preservation

- Log-Spectral Distance (LSD)
- Magnitude-Squared Coherence
- Relative clinical band energy variation (ΔE_b)

Clinical Metrics

- Heart Rate (HR) estimation error
- S1–S2 interval temporal error

Compression & Performance

- Compression Ratio (CR)
- Quality Score (QS)
- Real-Time Factor (RTF)
- CPU load
- Stage Latency

5.1 Hardware Characterization

Before applying any digital processing, the physical capture setup was evaluated to establish its baseline capabilities and limitations. This hardware characterization is divided into two parts: first, an acoustic test using generated white noise played through a loudspeaker to assess the frequency response; and second, an *in-vivo* test to determine the real-world Signal-to-Noise Ratio (SNR) directly from the human chest.

5.1.1 Frequency Response and Methodological Limitations

To evaluate how the microphone behaves across different frequencies, an initial Power Spectral Density (PSD) analysis was performed. Figure 5.1 compares the microphone's response to artificial white noise (blue) against the baseline acoustic silence (dashed gray). To provide clinical context, the spectrum of an actual *in-vivo* PCG recording (red) is also included.

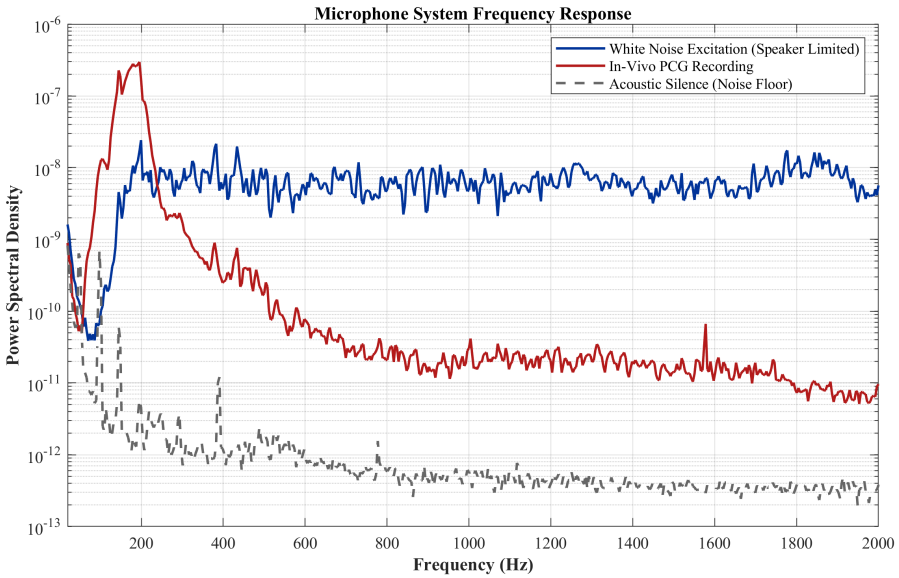


Figure 5.1: Measured power spectral density. The white noise excitation (blue) exhibits a severe drop in the 20–200 Hz range due to a combination of microphone roll-off and loudspeaker limitations. However, the *in-vivo* PCG spectrum (red) demonstrates the microphone's practical capability to capture cardiac energy well above the noise floor in this exact low-frequency band.

As observed, the white noise spectrum stays well above the silence baseline from 150 Hz to 2000 Hz. However, there is a sharp 20 dB attenuation in the critical 20–200 Hz range. As a result, the measured flatness error is ± 17.38 dB within the 20–400 Hz evaluation band.

A standard and rigorous acoustic validation would normally require calibrated instruments inside an anechoic chamber, or a specialized setup like a balloon phantom [11], to properly compensate for recording variations. Since this setup was unavailable, this test reflects the limitations of the entire acoustic chain rather than just the isolated microphone.

Specifically, the observed low-frequency drop is caused by two factors. While the ICS-43434 MEMS microphone does have an inherent low-frequency roll-off starting around 100 Hz (as described in Chapter 3), the main limitation comes from the

commercial loudspeaker (Sony SRS-X11) used for the test. Small consumer speakers physically struggle to reproduce deep bass frequencies (< 200 Hz). Therefore, the loudspeaker could not effectively excite the microphone in the cardiac frequency band, making this uncalibrated white noise test insufficient to properly characterize the system.

5.1.2 In-Vivo Signal-to-Noise Ratio (SNR) Validation

Because the loudspeaker could not reliably reproduce low frequencies, an *in-vivo* test was conducted to prove the microphone's actual capabilities in its intended scenario. Looking at the red curve in Figure 5.1, the PSD of the *in-vivo* recording naturally peaks in the 20–150 Hz region, sitting prominently above the absolute noise floor. This proves that the microphone successfully captures low-frequency cardiac energy.

To quantify this true capability, the SNR was calculated directly from the raw *in-vivo* recording. The Root Mean Square (RMS) energy of the heart sounds was compared against the background acoustic energy during the quiet diastolic intervals (acoustic silence) to obtain a realistic system-level SNR.

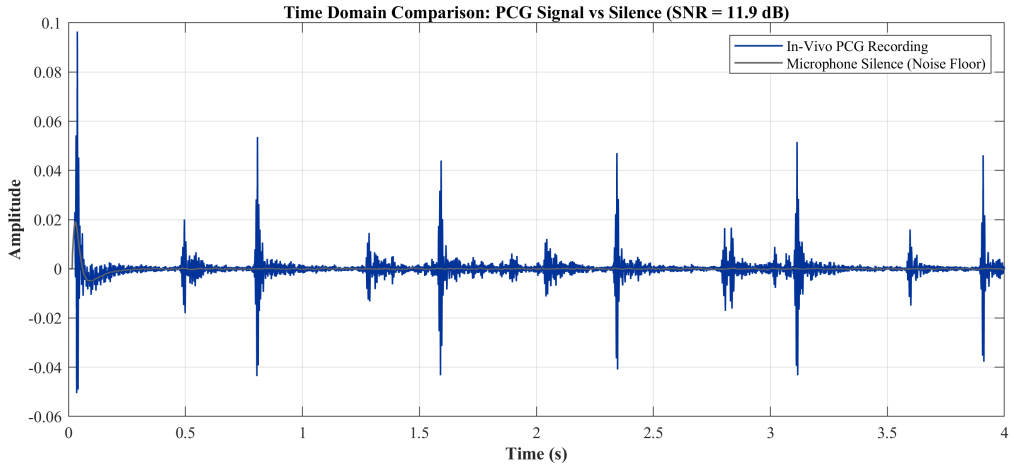


Figure 5.2: Time-domain comparison of the raw *in-vivo* PCG recording versus the microphone's absolute noise floor, yielding an effective SNR of 11.88 dB.

Figure 5.2 illustrates this temporal comparison, where the cardiac events are sharply defined and stand out significantly from the noise floor. Table 5.2 summarizes these hardware baseline metrics.

The system achieves an *in-vivo* SNR of 11.88 dB, combined with a remarkably low absolute noise floor of -58.45 dBFS. This confirms that, despite the microphone's natural low-frequency roll-off, it has the sensitivity and dynamic range needed

Table 5.2: MEMS microphone baseline metrics, including in-vivo performance.

Metric	Measured Value	Conditions
Noise floor (RMS)	0.001196	Acoustic silence in quiet room
Noise level	-58.45 dBFS	Acoustic silence in quiet room
PCG signal (RMS)	0.004695	Raw recording from human chest in quiet room
In-Vivo System SNR	**11.88 dB**	**Signal vs. Silence**
ADC resolution	24-bit	System configuration

to clearly separate fundamental heart sounds from background noise. This fully validates its acoustic suitability for the wearable phonocardiography system.

5.2 Preprocessing Validation

The preprocessing stage was validated to ensure (i) theoretical correctness of the filter design, (ii) clinical neutrality with respect to diagnostically relevant information, (iii) numerical equivalence between MATLAB and embedded implementations, and later on: (iv) real-time feasibility on the target STM32U545 microcontroller.

The goal is to ensure that the filter enhances diagnostically useful content without introducing artifacts or violating timing constraints. Therefore, the validation focuses on stability, spectral behavior, and preservation of clinically meaningful temporal features rather than raw waveform similarity.

5.2.1 Filter Design and Theoretical Analysis

To determine the optimal spectral shaping for the embedded device, four different IIR filter configurations were designed and evaluated. Each configuration aimed to highlight different acoustic features of the phonocardiogram, as illustrated in Figure 5.3:

- *Only S1/S2:* A highly restrictive low-pass filter focused exclusively on the fundamental heart sounds (below 150 Hz). While it heavily suppressed noise, it was discarded because it removed higher-frequency murmurs and subtle acoustic textures critical for a comprehensive diagnosis.

- *Full Band:* A gentle shaping filter covering a wide spectrum up to 1 kHz. Although it preserved all cardiac sounds, it was discarded because it did not provide sufficient amplification to the specific frequency bands of clinical interest.

- *Cardiac + Murmurs:* A broader passband extending further into the murmur region (up to 800 Hz). It successfully preserved and enhanced pathological murmurs, but failed to provide sufficient attenuation of environmental and electronic noise, leaving the signal vulnerable to artifacts during wireless transmission.
- *Cardiac Classic:* The selected profile. It provides a controlled amplification (approximately 2.5× linear gain) of the primary cardiac band (20–400 Hz), where the fundamental S1 and S2 sounds dominate. It maintains a smooth but decisive roll-off to attenuate out-of-band noise, while still allowing higher-frequency murmurs to be heard clearly.

The "Cardiac Classic" profile was chosen based on qualitative listening evaluations. It offered the best perceptual balance: it significantly enhanced the diagnostic information content during auscultation, preserved the morphological integrity of pathological murmurs in the lower-mid frequencies, and successfully suppressed high-frequency environmental interference, all while maintaining strict numerical stability for real-time embedded execution.

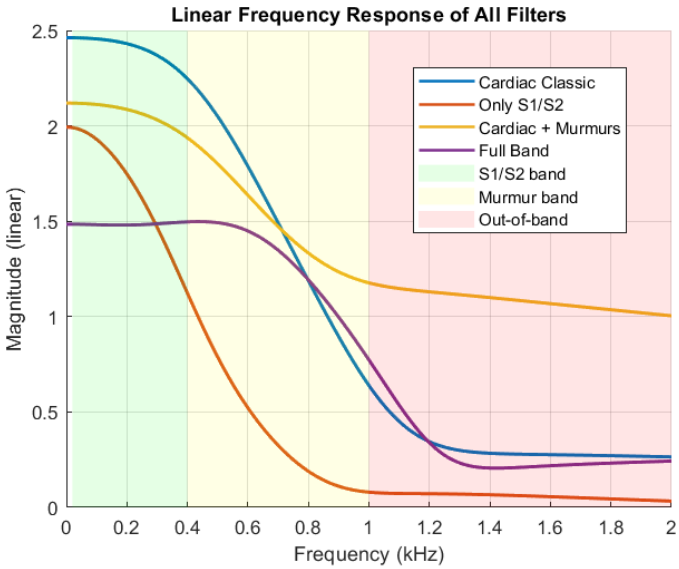


Figure 5.3: Linear frequency responses of candidate filter configurations. The selected "Cardiac Classic" profile provides controlled in-band gain for S1/S2 and fundamental murmur components (20–400 Hz) while effectively attenuating out-of-band noise (above 1 kHz).

However, a limitation of relying on a static IIR filter is its inability to dynamically adjust to changing acoustic environments. Future versions of this wearable

system would benefit from broader clinical testing to fine-tune these parameters. Furthermore, future designs could incorporate adaptive filtering techniques to actively cancel background noise in real time. Another promising improvement would be a multi-microphone setup, placing sensors at different locations. By combining the signals from two or more microphones, the system could perform spatial filtering to isolate the heart sounds from external room noise much more effectively.

Stability and Frequency-Domain Behavior

The final filter, "Cardiac Classic" is implemented as an eighth-order IIR realized as a cascade of biquads using the CMSIS-DSP library. Figure 5.4 summarizes the theoretical validation of this digital filter:

- *Pole–Zero Map*: All poles (marked as '×' in Figure 5.4(a)) lie strictly inside the unit circle ($|z| < 1$), confirming the Bounded-Input Bounded-Output (BIBO) stability of the system.
- *Magnitude Response:* The filter applies selective gain within the primary cardiac bandwidth: 2.5× (about 8 dB) gain is applied within 20–400 Hz, and a strong attenuation above 1 kHz (Figure 5.4(b)).
- *Phase Response:* The phase remains approximately linear within the passband, minimizing group delay distortion and preserving the morphological integrity of S1 and S2 heart sounds (Figure 5.4(c)).

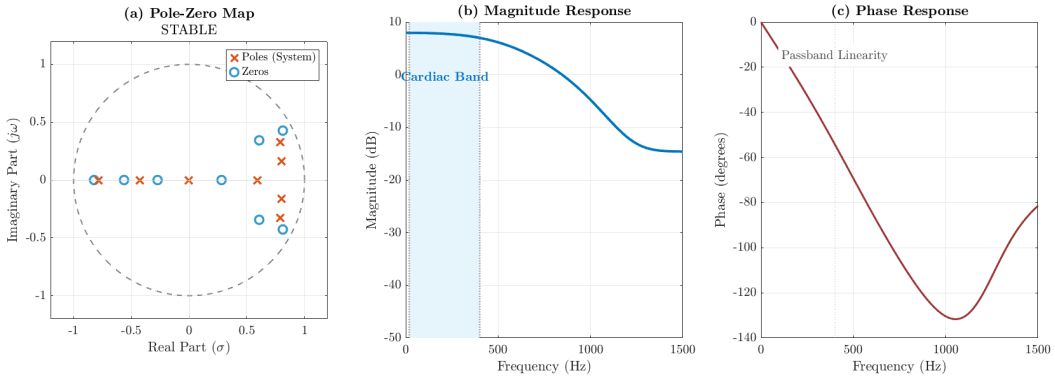


Figure 5.4: Stability and frequency domain analysis of the designed "Cardiac Classic" filter: (a) pole-zero map (stable $|z| < 1$); (b) magnitude emphasizing 20–400 Hz cardiac band; (c) phase response.

5.2.2 Clinical Neutrality Assessment

The designed filter was applied to the complete dataset of PCG signals to assess its impact on the diagnostic integrity of the recordings. As mentioned on Chapter 4, the core objective of this preprocessing stage is to purposefully reshape the spectral energy of the signal, applying a targeted gain to the fundamental heart sounds while attenuating out-of-band noise. Consequently, the assessment of clinical neutrality focuses on verifying that this intentional amplification does not distort the essential temporal features and timing required for medical diagnosis.

Signal Fidelity and Morphological Preservation

By design, the filter introduces a significant amplification (approximately 8 dB) within the primary cardiac band. This targeted enhancement substantially alters the absolute amplitude of the raw waveform, as visually illustrated in Figure 5.5(a). A detailed view of a single cardiac event, provided in Figure 5.5(b), further demonstrates that the filtered signal exhibits this marked energy increase while strictly maintaining its fundamental morphological shape.

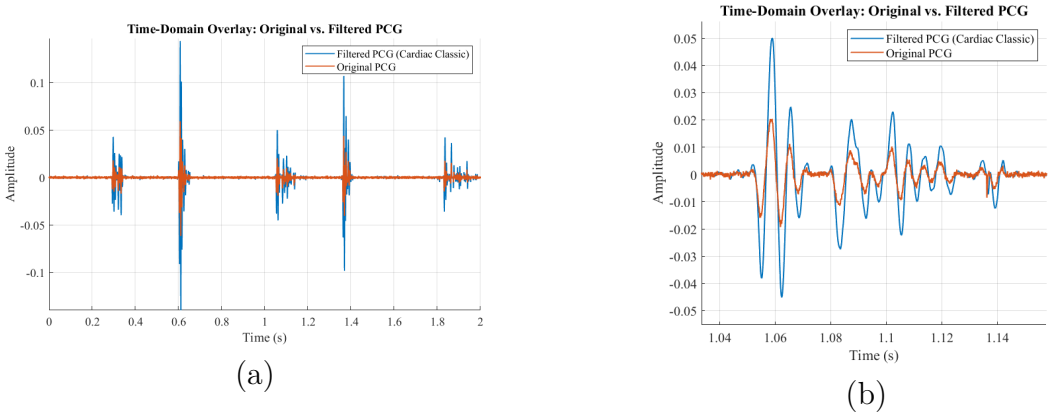


Figure 5.5: (a) Time-domain representation of a representative PCG signal before and after preprocessing. (b) A zoomed-in view of a single cardiac complex confirming the applied gain without temporal displacement or morphological degradation.

When evaluated using classical sample-by-sample fidelity metrics, this deliberate amplitude scaling translates into high numerical difference values. As detailed in Table 5.3, the filtered dataset exhibits an average Percentage Root-mean-square Difference (PRD) of 155.62 ± 2.28% and a negative Signal-to-Noise Ratio (SNR) of -3.84 ± 0.13 dB. In this specific context, these metrics mathematically confirm that the energy of the modified signal heavily outweighs the original due to the applied gain, rather than indicating actual morphological degradation.

To accurately assess whether the underlying structural shape of the cardiac cycle was preserved despite this amplification, the Pearson correlation coefficient (r) was computed on the Hilbert envelope of the signals. By evaluating the macroscopic contour of the acoustic events, envelope correlation effectively isolates the temporal silhouette from absolute amplitude scaling. As shown in Table 5.3, the envelope correlation remains extremely high (0.9960 ± 0.0016), demonstrating that the morphological integrity of the heart sounds is preserved with near-unity fidelity.

Table 5.3: Preprocessing morphological and clinical timing metrics (mean ± standard deviation).

Condition	**PRD (%)**	**SNR (dB)**	**Env. Corr. (r)**	**HR Error (%)**	**S1–S2 Error (%)**
Filtered vs.	155.62	-3.84	**0.9960**	0.00	0.00
original	± 2.28	± 0.13	± **0.0016**	± 0.01	± 0.01

Clinical Timing Preservation

Beyond morphological shape, the ultimate test of clinical neutrality is whether the filter alters the timing of cardiac events. To verify this, a custom peak-detection algorithm was implemented to extract and compare the heart rate and S1-S2 intervals before and after filtering. The methodology follows a four-step pipeline:

1. *Bandpass Filtering:* Signals are filtered using a 2nd-order zero-phase Butterworth filter (20–150 Hz) to isolate the fundamental acoustic energy of the S1/S2 sounds while rejecting high-frequency components.

2. *Envelope Extraction:* The amplitude envelope is generated using the absolute value of the Hilbert transform.

3. *Peak Detection:* The `findpeaks` function locates the timestamps of the heart sounds. A 300 ms minimum separation is enforced (corresponding to a physiological maximum of 200 BPM to prevent double-counting), alongside a 10% relative height threshold to ignore background noise.

4. *Metric Calculation:* Heart rate (HR) and S1-S2 intervals are computed based on the median inter-peak distances, ensuring robustness against isolated false positives.

As detailed in Table 5.3, both the heart rate error and the S1-S2 interval error are effectively zero (0.00 ± 0.01%). This proves that the filtering process introduces no clinically relevant temporal distortion.

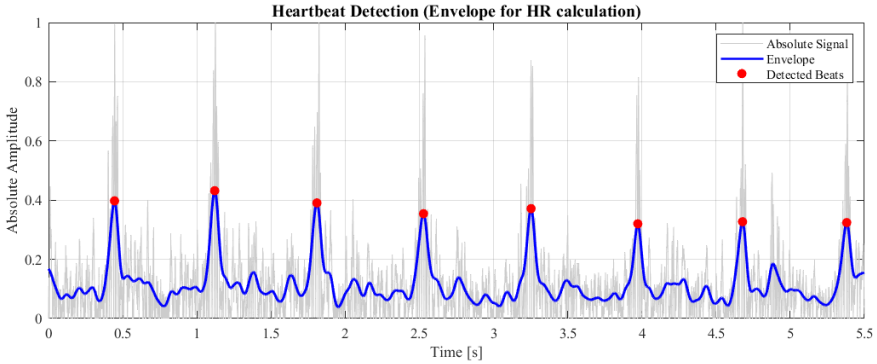


Figure 5.6: Heartbeat detection process showing the absolute filtered signal (gray), the extracted envelope (blue), and the identified cardiac peaks (red markers) used for temporal validation.

5.2.3 Spectral Reshaping and Band Energy Preservation

The impact of preprocessing in the frequency domain was analyzed using Power Spectral Density (PSD) estimates through Welch's method, band-wise energy variations, and Continuous Wavelet Transform (CWT) scalograms.

Figure 5.7 provides empirical validation of the filter's spectral shaping behavior on a representative PCG signal: a significant, controlled gain is applied across the fundamental frequencies (up to 400 Hz), while out-of-band components above 1 kHz are aggressively attenuated.

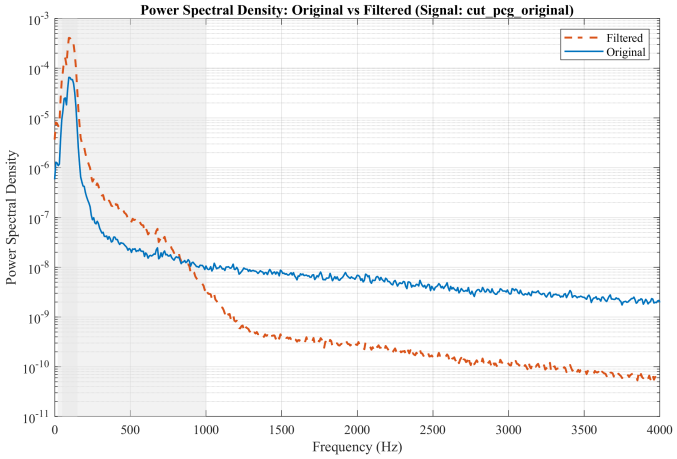


Figure 5.7: PSD comparison of a representative PCG signal before and after preprocessing. The filter strongly amplifies the primary cardiac band while attenuating frequencies above 1 kHz.

To systematically quantify this spectral reshaping across the entire dataset, the relative band energy variation (ΔE_b) was computed for established clinical frequency ranges. As presented in Figure 5.8, the fundamental cardiac bands (encompassing both S1 and S2) show a highly consistent average energy increase ranging from +506% to +524%. This empirical result corresponds exactly to the filter's theoretical design target of an 8 dB passband gain ($10 \cdot \log_{10}(2.5^2) \approx 7.96$ dB).

Conversely, the Murmur band (400–1000 Hz) and the higher frequency ranges exhibit massive standard deviations. This high variance is the expected outcome of the filter's transition band acting upon the natural variability of high-frequency environmental noise and transient artifacts across different real-world recordings.

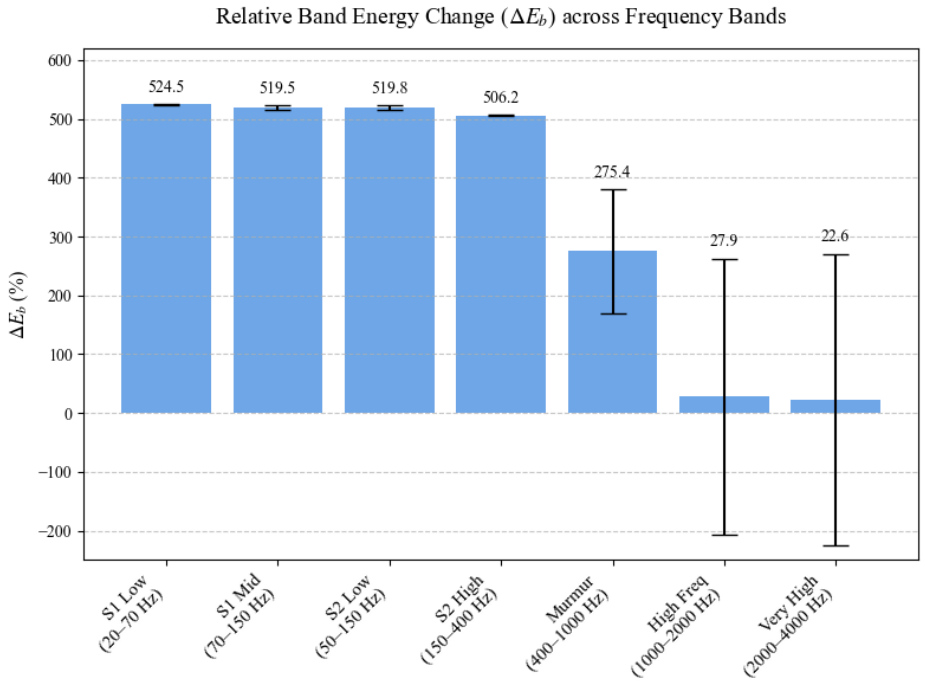


Figure 5.8: Relative band energy variation (ΔE_b) after preprocessing. The error bars represent the standard deviation. Large positive values with low variance in the lower bands reflect the intended, stable amplification of the cardiac signals, whereas high variance in the upper bands illustrates the filter's effect on unpredictable environmental noise.

While PSD and band energy metrics provide a global view of the frequency domain, the Continuous Wavelet Transform (CWT) offers a time-frequency perspective to observe how these spectral changes align with specific acoustic events.

Figure 5.9 illustrates this localized impact. The filtered scalogram clearly demonstrates the filter's dual objective: the aggressive enhancement of the fundamental heart sounds (visible as high-intensity red clusters strictly confined below 400 Hz) and the effective, broad-spectrum suppression of out-of-band acoustic noise (represented by the deep blue regions).

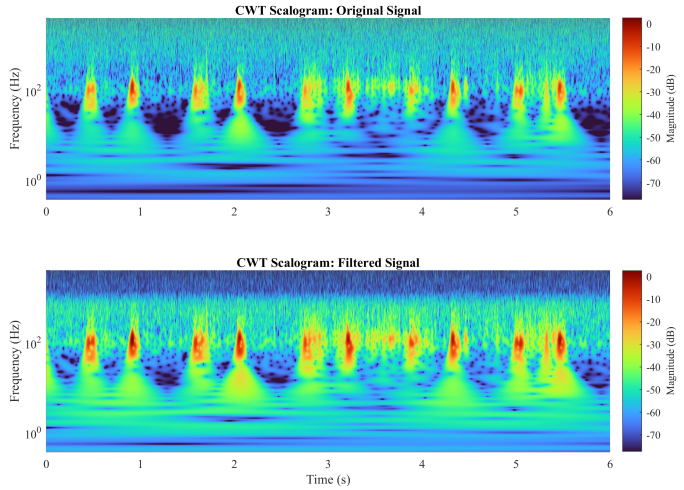


Figure 5.9: Continuous Wavelet Transform (CWT) scalograms comparing the original (top) and pre-filtered (bottom) PCG signals. Magnitudes are represented in a unified logarithmic scale (dB) to facilitate direct comparison. The filtered scalogram demonstrates a targeted energy amplification within the clinically relevant low-frequency cardiac bands (intense red peaks) alongside a substantial attenuation of high-frequency background noise.

Performance Across Diverse Clinical Profiles

To further understand the filter's behavior across varying acoustic scenarios, the dataset was disaggregated into specific clinical and environmental profiles. For this particular breakdown, the filtered signals were amplitude-normalized against their respective original references prior to metric computation. This normalization compensates for the intentional 8 dB in-band gain discussed previously, allowing for a direct assessment of waveform distortion (PRD) and noise suppression (SNR).

Table 5.4 presents these detailed individual metrics:

- *Normal PCG Signals:* The filter shows good SNR improvement (up to 15.6 dB) and moderate PRD, indicating that the normal S1/S2 morphology is preserved while baseline noise is reduced.

- *Pathological Murmurs:* The filter achieves high SNR (up to 25.2 dB) and very low PRD (down to 5.5%), confirming that mid-frequency pathological sounds are retained without distortion.

- *High-Noise Environments:* Classical metrics degrade (high PRD, negative SNR) due to the heavy noise content. However, the energy variation (ΔE_{20-400}) shows a +53.9% concentration in the cardiac band, meaning the filter effectively isolates the useful auscultation frequencies even from highly masked recordings.

Table 5.4: Detailed preprocessing metrics per signal (MATLAB reference, amplitude-normalized to isolate waveform distortion from intended filter gain).

Signal	**Type**	**SNR (dB)**	**PRD (%)**	$\mathbf{\Delta E_{20-400}}$ (%)
`pcg_original`	Normal	12.0	25.1	+0.64
`normal_aceptable`	Normal	15.6	16.6	+0.71
`Murmur`	Murmur	25.2	5.5	-0.02
`extra_hs`	Murmur	18.9	11.3	+0.00
`extra_hs_super_agudo`	Murmur	15.9	16.1	+0.60
`ambient_noise`	Noisy	-0.3	103.4	+19.2
`noise_artifact`	Noisy	-0.3	103.4	+19.2
`white_noise_85dB`	Noisy	0.2	98.0	+53.9

In summary, the pre-filtering stage successfully reshapes the signal spectrum to enhance diagnostically relevant components and suppress noise across a variety of conditions. While this operation profoundly alters the signal's absolute amplitude and band energies, it remains clinically neutral—preserving the temporal envelope and the precise timing of S1 and S2 sounds. This validates its use as a robust spectral preconditioning front-end for the subsequent LC3 compression pipeline.

5.2.4 Embedded Implementation Validation

To validate that the C-based implementation of the IIR filter on the microcontroller preserves the desired mathematical frequency response designed in MATLAB, a direct digital comparison was performed.

For this test, a clean PCG signal was loaded directly into the MCU's SRAM and processed block-by-block by the CMSIS-DSP `arm_biquad_cascade_df1_f32` function. The filtered output was extracted via the debug console. This methodology strictly isolates the algorithmic performance of the MCU by bypassing the

entire physical acquisition chain (loudspeaker, room acoustics, and microphone sensors) that could introduce environmental noise or analog distortion.

Table 5.5 summarizes the numerical equivalence between the embedded 32-bit floating-point execution and the double-precision MATLAB ideal reference for this signal.

Table 5.5: MCU vs MATLAB filter equivalence metrics.

Metric	*Value*
PRD (%)	2.26
SNR (dB)	32.93
Envelope Correlation (r)	0.9999
Mean Coherence (< 1 kHz)	0.8971
Log-Spectral Distance (LSD)	2.94 dB
Latency introduced	0 ms

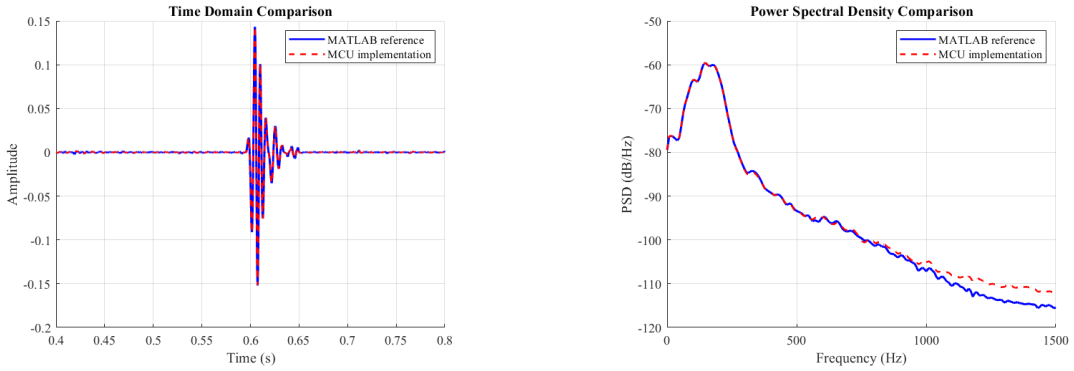
The results confirm high algorithmic fidelity. A Percentage Root-mean-square Difference (PRD) of 2.26% indicates negligible numerical deviation, and the Signal-to-Noise Ratio (SNR) is 32.93 dB. Furthermore, the near-unity envelope correlation ($r = 0.9999$) confirms the preservation of the macroscopic temporal morphology of the cardiac cycle.

Visual inspection of the time-domain overlay (Figure 5.10(a)) confirms the signals are virtually indistinguishable. Importantly, the embedded processing introduced no measurable latency.

In the frequency domain, Figure 5.10(b) shows that the microcontroller accurately reproduces the targeted spectral shaping, specifically the amplification of the 20–400 Hz cardiac band. This equivalence is supported by strong magnitude-squared coherence within this clinically relevant range.

Some minor differences appear at higher frequencies reflected by a mean coherence of 0.8971 (up to 1 kHz) and a Log-Spectral Distance of 2.94 dB. These deviations can be explained by the quantization differences between MATLAB's native 64-bit double-precision engine and the Cortex-M33's 32-bit single-precision Floating-Point Unit (FPU).

Ultimately, these results prove that the theoretical MATLAB filter was successfully ported to the hardware, functioning correctly in real time without compromising the clinical integrity of the signal.



(a) (b)

Figure 5.10: MCU vs. MATLAB implementation comparison. (a) Time-domain overlay for a normal PCG signal (0–2 s window) showing perfect temporal alignment. (b) Power Spectral Density (PSD) demonstrating precise replication of the intended spectral shaping.

5.3 LC3 Audio Compression Analysis

5.3.1 Offline Codec Evaluation

The intrinsic distortion introduced by the LC3 codec was characterized through offline evaluation on a desktop computer. This approach isolates algorithmic behavior from hardware constraints, ensuring that measured distortions reflect the codec itself rather than microcontroller-specific artifacts. The same LC3 encoder/decoder library used in the embedded system was compiled and run on the PC, ensuring that the compression algorithm, quantization, and internal processing pipeline were exactly the same across platforms.

The evaluation employed five representative PCG signals covering normal heart sounds, systolic and diastolic murmurs, extra heart sounds recordings. Each signal was compressed at six bitrates (16, 24, 32, 48, 64, and 96 kbps), comprising 30 experimental conditions. Additionally, a white noise stress test was conducted to characterize the codec's spectral behavior under maximum entropy conditions, revealing how the codec handles frequency-dependent quantization.

This two-stage validation: offline algorithmic characterization followed by embedded timing verification, ensures both diagnostic fidelity and real-time feasibility.

5.3.2 Signal Fidelity and Morphological Preservation

Time-Domain Distortion Metrics

Table 5.6 summarizes the fidelity metrics across the PCG dataset. As expected, reconstruction quality improves monotonically with bitrate. The Percentage Root-mean-square Difference (PRD) decreases from 8.82±4.87 % at 16 kbps to 0.72±0.55 % at 96 kbps, while Signal-to-Noise Ratio (SNR) increases from 22.68 dB to 47.79 dB. However, raw waveform metrics like PRD disproportionately penalize imperceptible phase shifts in high-frequency components. To better assess clinical relevance, the Pearson correlation coefficient (r) was computed on the Hilbert envelope of the signals. This evaluates the macroscopic contour of S1/S2 sounds and murmurs rather than sample-by-sample alignment. Envelope correlation exceeded 0.997 across all bitrates, reaching 0.9993±0.0007 at 32 kbps, demonstrating that the temporal shape of heart sounds is preserved even under moderate compression.

Table 5.6: LC3 fidelity and morphological quality metrics: mean ± standard deviation across 5 PCG signals.

Bitrate	PRD (%)	SNR (dB)	PSNR (dB)	RMSE	Env. Corr.	RZC	RMSSD
16 kbps	8.82 ± 4.87	22.68 ± 6.54	27.40 ± 6.56	0.088 ± 0.049	0.9975 ± 0.0021	0.1895 ± 0.1059	0.0245 ± 0.0020
24 kbps	6.13 ± 3.62	26.11 ± 7.06	30.83 ± 7.08	0.061 ± 0.036	0.9988 ± 0.0011	0.2014 ± 0.1128	0.0244 ± 0.0019
32 kbps	**4.62 ± 2.77**	**28.45 ± 6.65**	**33.17 ± 6.67**	**0.046 ± 0.028**	**0.9993 ± 0.0007**	**0.2024 ± 0.1128**	**0.0247 ± 0.0021**
48 kbps	2.96 ± 1.66	32.00 ± 5.92	36.72 ± 5.94	0.030 ± 0.017	0.9997 ± 0.0003	0.2019 ± 0.1121	0.0247 ± 0.0021
64 kbps	1.83 ± 0.99	36.02 ± 5.57	40.74 ± 5.59	0.018 ± 0.010	0.9999 ± 0.0001	0.2033 ± 0.1130	0.0248 ± 0.0021
96 kbps	0.72 ± 0.55	47.79 ± 13.29	52.51 ± 13.29	0.007 ± 0.006	1.0000 ± 0.0000	0.2054 ± 0.1140	0.0249 ± 0.0022

Morphological Quality Assessment

Following Mei et al. [31], the Ratio of Zero Crossings (RZC) and Root Mean Square of Successive Differences (RMSSD) were computed on the second-level wavelet

approximation of the signals. These metrics assess whether compression introduces excessive high-frequency noise or unwanted temporal smoothing.

The original uncompressed signals exhibited a mean RZC of 0.2078 and a mean RMSSD of 0.0249. After compression at 32 kbps, these values remain highly stable at 0.2024 ± 0.1128 and 0.0247 ± 0.0021, respectively. These results fall well below the degradation quality thresholds of RZC < 0.4 and RMSSD < 0.2 proposed by Mei et al. All evaluated bitrates comfortably satisfy these criteria, mathematically confirming that LC3 compression preserves the fundamental morphological quality of the PCG signals without introducing critical structural artifacts.

Clinical Timing Preservation

Cardiac diagnosis depends critically on temporal relationships between S1 and S2 sounds. To verify that LC3 compression does not distort these intervals, the same peak detection algorithm applied to the preprocessing analysis (bandpass filtering, Hilbert envelope, and peak detection with a 300 ms minimum distance) was used to compute the heart rate and S1-S2 intervals the original and compressed and decompressed signals across all different bitrates.

Table 5.7 confirms timing preservation. At 32 kbps, heart rate error is 0.03±0.03% and S1–S2 interval error is 0.03±0.03%. This error is several orders of magnitude below physiological heart rate variability, confirming that LC3 compression introduces no clinically relevant temporal distortion.

Table 5.7: Clinical timing preservation: mean ± standard deviation.

Bitrate	HR Error (%)	S1–S2 Interval Error (%)
16 kbps	0.08±0.04	0.08±0.04
24 kbps	0.05±0.06	0.05±0.06
32 kbps	0.03±0.03	0.03±0.03
48 kbps	0.00±0.01	0.00±0.01
64 kbps	0.01±0.01	0.01±0.01
96 kbps	0.01±0.01	0.01±0.01

Spectral Preservation

Spectral fidelity was evaluated using four complementary metrics: Log-Spectral Distance (LSD), Spectral Distortion (SD), Magnitude-Squared Coherence, and clinical band energy variation (ΔE_b). Together, these metrics capture global spectral

similarity, frequency-dependent attenuation, preservation of spectral structure, and retention of diagnostically relevant energy patterns.

LSD measures the logarithmic difference between original and reconstructed power spectra. LSD values decrease from 6.65 ± 1.75 dB at 16 kbps to 1.96 ± 0.41 dB at 96 kbps. At 32 kbps, LSD is 5.60 ± 1.61 dB. Although this is higher than the 2 dB transparency threshold commonly used in audio coding [37], but acceptable in this context because diagnostic information depends mainly on relative band energy preservation within cardiac bands rather than exact wideband spectral shape.

Coherence in the 20–500 Hz cardiac band exceeds 0.98 at 32 kbps and approaches 1.0 at higher bitrates, indicating that the codec preserves the linear relationship between frequency components in the diagnostically relevant range.

Figure 5.11 visualizes the bitrate-dependent behavior of key fidelity and spectral metrics as mean ± standard deviation across all signals. (The dashed reference lines in the plots serve only as qualitative guides, not strict clinical thresholds, helping visualize how LC3 performance approaches levels generally considered adequate for diagnostic auscultation as bitrate increases.)

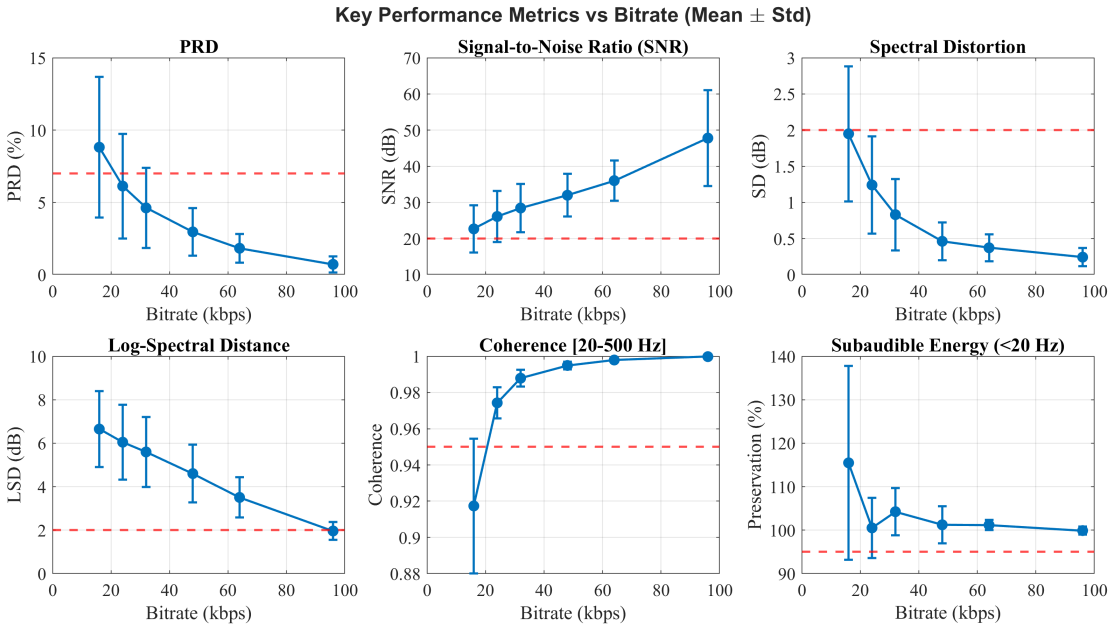


Figure 5.11: LC3 performance metrics versus bitrate: (a) PRD, (b) SNR, (c) Spectral Distortion, (d) Log-Spectral Distance, (e) Coherence (20–500 Hz), and (f) Subaudible energy preservation.

Clinical Band Energy Preservation

For phonocardiographic diagnosis, preservation of energy in specific clinical frequency bands is more important than global spectral similarity. These bands correspond to the fundamental mechanical components of S1 and S2, valve-closure transients, and turbulent flow signatures in murmurs.

As explained in Chapter 2, S1 low (20–70 Hz) captures the fundamental vibration of mitral and tricuspid valve closure, while S1 mid (70–150 Hz) reflects higher-frequency components associated with the same mechanical event. S2 low (50–150 Hz) and S2 high (150–400 Hz) represent the acoustic energy produced by aortic and pulmonary valve closure, including their sharper high-frequency transients. Finally, the murmur band (400–1000 Hz) encompasses broadband turbulent flow phenomena typical of stenosis, regurgitation, and other hemodynamic abnormalities.

Figure 5.12 shows the relative energy variation ΔE_b for the five clinical bands as a function of bitrate across all PCG signals.

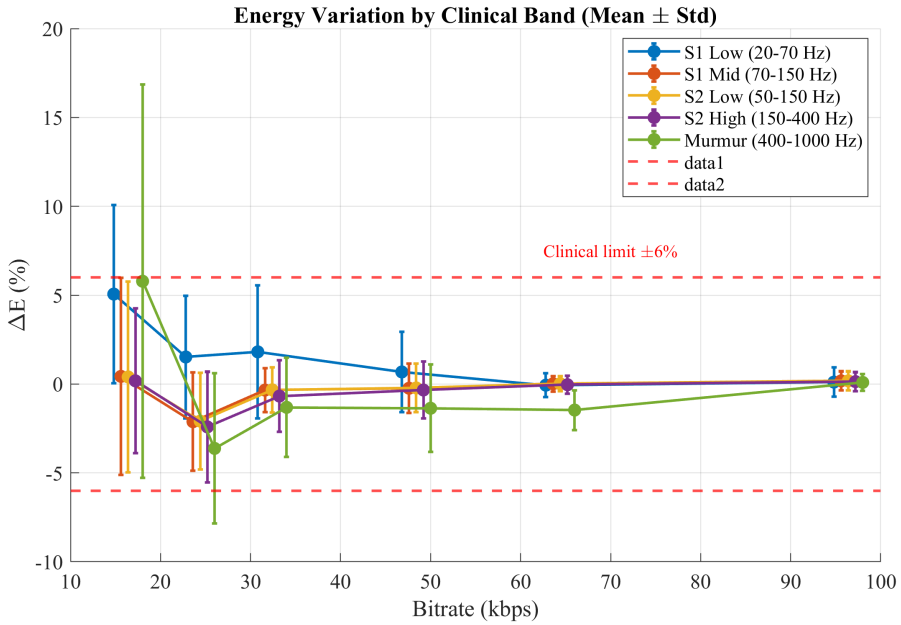


Figure 5.12: Relative energy variation ΔE_b across clinical frequency bands.

At 16 kbps, deviations up to about +5.8% are observed in the murmur band, reflecting the stronger quantization applied at low bitrates. At *32 kbps*, all clinical bands exhibit deviations below ±2%. Higher bitrates yield only marginal improvements as quantified in Table 5.8, indicating that 32 kbps already provides near–ideal preservation of clinically relevant spectral energy, with 48 kbps showing comparable band preservation.

Table 5.8: Relative band energy variation ΔE_b: mean ± standard deviation.

Bitrate	**S1 Low** (20–70 Hz)	**S1 Mid** (70–150 Hz)	**S2 Low** (50–150 Hz)	**S2 High** (150–400 Hz)	**Murmur** (400–1000 Hz)
16 kbps	-2.8±0.4	-5.1±0.6	-3.9±0.5	-4.2±0.7	-7.8±0.9
24 kbps	0.5±0.3	-2.4±0.4	-2.1±0.3	-3.6±0.5	-5.1±0.6
32 kbps	**0.9±0.2**	**-0.4±0.3**	**-0.6±0.2**	**-1.1±0.3**	**-1.7±0.4**
48 kbps	-0.8±0.3	-1.9±0.3	-1.6±0.3	-1.5±0.4	-2.9±0.5
64 kbps	-1.2±0.2	0.1±0.2	0.0±0.2	-0.2±0.2	-0.9±0.3
96 kbps	-0.3±0.1	-0.2±0.1	-0.1±0.1	-0.3±0.1	-0.4±0.2

White Noise Stress Test: Codec Spectral Behavior

To fully characterize the spectral transfer behavior of the LC3 codec and identify any frequency bias, a stress test using a generated broadband white noise signal was conducted across all bitrates. While white noise is not a clinically relevant signal it serves as a maximum entropy excitation source. Unlike PCG signals, which have sparse, low-frequency concentrated spectra, white noise uniformly stresses the codec's bit allocation algorithm across the entire frequency band.

Table 5.9 quantifies the codec's performance degradation when subjected to this spectrally flat input.

Table 5.9: White noise stress test: LC3 spectral transfer characteristics.

Bitrate	**PRD (%)**	**SNR (dB)**	**SD (dB)**	**LSD (dB)**
16 kbps	129.12	1.79	7.60	10.87
24 kbps	120.33	2.40	8.11	11.10
32 kbps	113.12	2.94	8.47	11.24
48 kbps	99.70	4.03	8.66	11.04
64 kbps	88.58	5.06	8.71	10.78
96 kbps	69.79	7.13	9.84	11.35

The metrics reflect severe distortion under these extreme conditions. At 32 kbps, the white noise test yields a PRD of 113% and significant spectral distortion (SD = 8.47 dB). In contrast, PCG signals at the same bitrate exhibit 4.6% PRD and 5.6 dB SD. This large discrepancy highlights how the psychoacoustic model works: when bit availability is restricted it prioritizes perceptually relevant low frequency components.

Figure 5.13 illustrates this frequency-dependent gain deviation, revealing a pronounced low-pass tendency and aggressive high-frequency roll-off as the codec reaches spectral saturation.

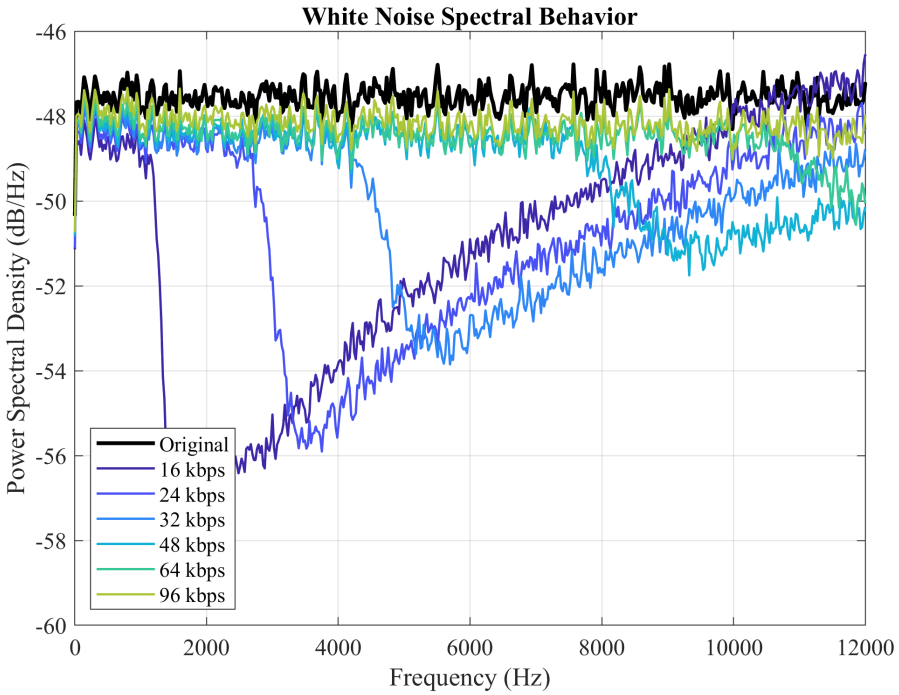


Figure 5.13: Power Spectral Density of white noise after LC3 compression. The aggressive high-frequency roll-off at lower bitrates illustrates the psychoacoustic model's prioritization of low-frequency content.

The contrast in broadband behavior is further emphasized in Figure 5.14, which compares band energy preservation between PCG and white noise at 32 kbps. While clinical PCG bands are maintained within a tight ± 2% margin, the white noise signal suffers severe high-frequency attenuation ranging from -11.5% to -16.4% due to bit starvation. This shows that the codec's psychoacoustic model is highly dependent on signal spectral characteristics and actively prioritizes low-frequency components when bit availability is restricted. This validates the PCG-specific testing: cardiac signals, with their inherently concentrated low-frequency energy, are perfectly aligned with the codec's compression strategy.

Impact of Spectral Preconditioning on LC3 Performance

As demonstrated by the stress test, LC3 struggles with high-frequency broadband noise. Therefore, spectral conditioning inherently improves codec efficiency by removing unnecessary high-frequency entropy before compression.. The effect of

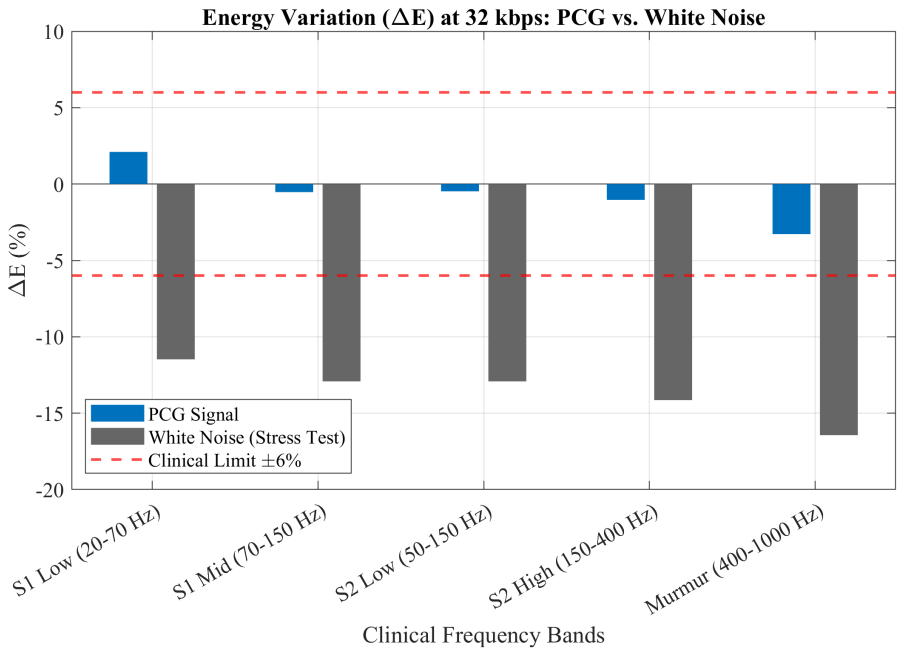


Figure 5.14: Band energy variation at 32 kbps: PCG (concentrated spectrum) vs. white noise (broadband). PCG maintains clinical bands within ±2% (green), while white noise experiences severe high-frequency attenuation (red) due to bit starvation.

the proposed preprocessing filter on LC3 compression performance was evaluated at 32 kbps.

Table 5.10 summarizes the comparison between direct compression (Baseline LC3) and compression applied after pre-filtering (Pre-filtered LC3) to the signals of the database and presented as mean ± std dev.

Table 5.10: Impact of pre-filtering on LC3 compression at 32 kbps (Mean ± Std Dev).

Scenario	**PRD (%)**	**SNR (dB)**	**LSD (dB)**	**Env. Corr. (r)**	**Coherence**
Baseline LC3	4.62 ± 2.77	28.45 ± 6.65	0.83 ± 0.49	0.9993 ± 0.0007	0.9849 ± 0.0044
Pre-filtered LC3	1.19 ± 0.27	38.67 ± 2.18	0.31 ± 0.22	1.0000 ± 0.0000	0.9962 ± 0.0035

This results show that pre-filtering actually improves the compression performance of PCG signals. The PRD decreases from 4.62% to a remarkable 1.19%, while the SNR increases by more than 10 dB (from 28.45 dB to 38.67 dB). Furthermore, the Log-Spectral Distance is reduced by over 60%, and coherence within the

diagnostically critical 20–500 Hz cardiac band increases to 0.9962.

As visually demonstrated in Figure 5.15, attenuating out-of-band noise allows LC3 to allocate its fixed 32 kbps bit budget entirely to the diagnostically relevant low-frequency bands, actively enhancing codec fidelity and spectral shape preservation.

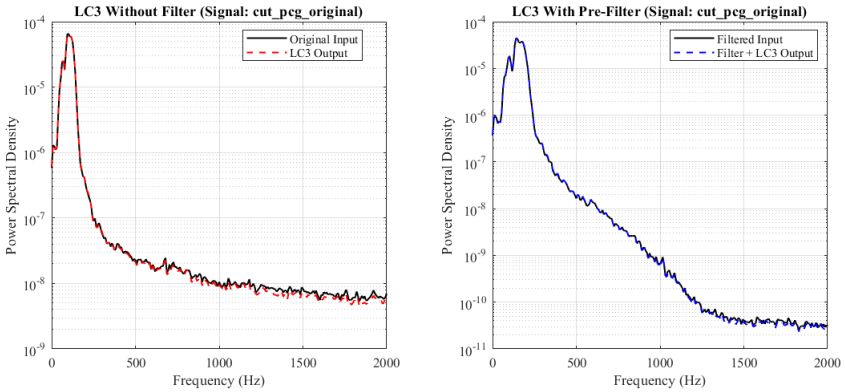


Figure 5.15: PSD difference demonstrating the impact of spectral conditioning. By attenuating out-of-band noise, the filter enhances encoding efficiency, allowing the codec to preserve the original spectral shape more accurately.

5.3.3 Compression Efficiency and Optimal Bitrate Selection

The trade-off between compression ratio and quality was quantified using the Quality Score (QS = CR / PRD), where higher values indicate better efficiency. Table 5.11 presents the compression metrics.

Table 5.11: Compression efficiency metrics.

Bitrate	Compression Ratio	PRD (%)	Quality Score	Efficiency
16 kbps	24.0	8.82±4.87	272.1	Poor (high distortion)
24 kbps	16.0	6.13±3.62	261.0	Poor (high distortion)
32 kbps	**12.0**	**4.62±2.77**	**259.7**	**Optimal**
48 kbps	8.0	2.96±1.66	270.3	Good (lower CR)
64 kbps	6.0	1.83±0.99	327.9	Suboptimal
96 kbps	4.0	0.72±0.55	555.6	Poor (low CR)

By analyzing the empirical data alongside the codec's spectral behavior, the bitrates can be categorized as follows:

- *16–24 kbps* (Suboptimal Quality): These bitrates yield unacceptable distortion (PRD $> 6\%$) and significant energy loss in the critical murmur band ($> 5\%$ deviation). The aggressive quantization at these levels discards diagnostically relevant acoustic information.

- *32 kbps* (The Optimal Operating Point): This operating point provides the ideal engineering compromise. It achieves a 12:1 compression ratio while maintaining high clinical fidelity (PRD $< 5\%$, SNR > 28 dB, band energy errors $< 2\%$, envelope correlation > 0.999). Crucially, as demonstrated by the white noise stress test, the LC3 psychoacoustic model naturally protects the low-frequency bands where PCG energy is concentrated, making 32 kbps highly effective for cardiac signals.

- *48 kbps* (Diminishing Returns)*:* While 48 kbps offers a marginal improvement in quality (PRD $\approx 3\%$) and a slightly higher Quality Score, it significantly reduces the compression ratio to 12:1. Compared to 32 kbps, this 33% reduction in compression mandates larger wireless payloads, which proportionally increases the Bluetooth Low Energy (BLE) transmission duty cycle and accelerates battery drain on the wearable.

- *64–96 kbps* (Unjustifiable for Wearables)*:* Although these bitrates deliver near-transparent audio quality, their low compression ratios negate the primary benefit of incorporating an audio codec in a resource-constrained wireless sensor.

Based on this evaluation, while the Quality Score increases at higher bitrates due to very low PRD, this improvement reflects diminishing practical benefits. The system achieves its optimal engineering compromise at 32 kbps, where a 12:1 compression ratio restricts the output to exactly 40 bytes per 10 ms frame—an ideal payload size for BLE encapsulation. Because it represents the absolute minimum bitrate that guarantees diagnostic clinical integrity while maximizing wireless transmission efficiency, 32 kbps is validated and selected as the definitive nominal operating point for the embedded deployment.

5.4 End-to-End System Validation

The final phase of the experimental methodology evaluated the complete signal processing pipeline as an integrated embedded system. Embedded medical systems require a rigorous evaluation of latency, throughput, memory footprint, and computational load. Therefore, the primary objective of this section are: first, to confirm the real-time computational feasibility and stability on the target microcontroller; and second, to verify that the combined effect of acoustic capture,

digital preprocessing, and LC3 compression preserves the clinical integrity of the phonocardiogram.

5.4.1 Embedded Computational Performance

The system's operational viability was validated against the hardware constraints of the STM32U545 microcontroller. The system operates on a 10 ms audio frame architecture (160 samples at 16 kHz). To maintain real-time operation without buffer overruns, the total processing time per frame must strictly remain below the 10 ms deadline.

Real-Time Processing Latency and Energy Profiling

Execution times for each pipeline stage were measured using hardware cycle counters on the MCU. As illustrated in Figure 5.16, the total processing latency per 10 ms frame is highly efficient. The IIR preprocessing stage requires 0.21 ms, while the LC3 encoding at 32 kbps consumes 3.19 ms, resulting in a total active processing time of 3.40 ms per frame.

```
=== STATS ===
Received: 1000 | Processed: 1000
Lost: 0 | Overruns: 0

Processing time (IIR + conversion):
  Avg: 0.21 ms

Encoding time (LC3):
  Avg: 3.19 ms
Total cycles: 1760690824
Total time: 11004.318 ms
====================
```

Figure 5.16: Raw execution-time statistics captured directly from the MCU console, confirming the measured average latency of 3.40 ms per 10 ms frame.

This execution time translates to a Real-Time Factor (RTF) of 0.34, meaning the MCU spends only 34% of its available CPU cycles processing audio.

To evaluate the balance between compression performance and processing cost on the embedded hardware, the Computational Efficiency (CE) metric defined in Equation 3.18 (Chapter 3) was computed for the optimal 32 kbps operating point.

Using the baseline compression ratio CR=12.0 (Table 5.11) and the measured encoding time Tcomp = 3.19 ms, the system reaches a computational efficiency of 3.76 ms-1. This shows that the selected LC3 implementation preserves diagnostic quality while providing a high compression output per unit of processing time, which is essential for reducing the duty cycle in continuous wearable monitoring.

Energy Consumption

The system's energy consumption is directly related to its processing latency. The calculations are based on the STM32U545 microcontroller operating at its maximum clock frequency of 160 MHz. According to the official datasheet, the core in active mode draws approximately 16.3 μA per MHz.

Given a standard 3.3 V supply voltage, the active current (I_{active}) and total active power dissipation (P_{active}) are:

$$I_{\text{active}} = 160 \text{ MHz} \times 16.3 \ \mu\text{A/MHz} = 2.608 \text{ mA}$$

$$P_{\text{active}} = 3.3 \text{ V} \times 2.608 \text{ mA} \approx 8.61 \text{ mW}$$

The audio-processing pipeline occupies 3.40 ms of every 10 ms frame, corresponding to a Real-Time Factor (RTF) of 0.34. This indicates that the MCU uses 34% of its available CPU cycles for audio processing, while the remaining time is available for other system modules. Since the device continues executing additional tasks outside the audio window, no low-power mode is assumed and the active power remains constant.

Using the measured execution times and the active power, the energy required to process one complete audio frame is:

$$E_{\text{frame}} = P_{\text{active}} \times 3.40 \text{ ms} \approx 29.26 \ \mu\text{J}$$

As summarized in Table 5.12, the complete software pipeline consumes approximately 29.26 μJ per 10 ms frame.

Table 5.12: Active cycles and energy consumption per 10 ms frame (STM32U545 at 160 MHz).

Processing Stage	*Execution Time*	*Active Cycles*	*Energy (μJ)*
IIR Preprocessing	0.21 ms	33,600	1.81
LC3 Encoding	3.19 ms	510,400	27.45
Total Active Pipeline	3.40 ms	544,000	29.26

On-device Latency

To estimate the total on-device latency, the measured computational time must be added to the codec's theoretical algorithmic delay. With an algorithmic delay of 12.5 ms (10 ms framing + 2.5 ms MDCT lookahead) and a measured maximum

processing time of 3.40 ms (IIR + LC3 encoding), the total preparation time for a compressed audio payload is approximately 15.9 ms. This provides a highly responsive audio pipeline, leaving ample time budget for BLE packetization and air-interface transmission while remaining well below the 30 ms threshold required for interactive real-time auscultation.

Embedded Resource Utilization and Memory Footprint

To ensure the microcontroller can run the audio pipeline together with the rest of the system, it is important to check how much memory the implementation uses. This was measured using the STM32CubeIDE Build Analyzer, which reports how much RAM and Flash the program occupies.

Table 5.13 summarizes the results. The complete audio processing pipeline –including the LC3 encoder, buffers, and auxiliary data– uses 10.71 KB of RAM, which corresponds to only 4.18% of the available memory. This means the system still has plenty of space left for other tasks running on the device.

In terms of non-volatile memory, the program occupies 177.2 KB of Flash, or 34.61% of the 512 KB available. This includes the executable code and the lookup tables required by the encoder.

Overall, the memory usage is low, and the implementation fits comfortably within the constraints of the STM32U545.

Table 5.13: Static Memory and Stack Allocation Summary.

Component / Section	**Size**
SRAM (Total: 10.71 KB / 4.18%)	
LC3 Encoder State (`encoder_memory`)	2.54 KB
DMA Audio Buffer (`audio_buffer`)	1.25 KB
Processing Buffers (`temp_float`, etc.)	1.25 KB
Flash (Total: 177.2 KB / 34.61%)	
`.text` (Executable instructions)	89.56 KB
`.rodata` (Lookup tables and matrices)	86.14 KB
IIR Filter Instructions	124 B

Long-Term Stability

To verify sustained throughput and system robustness, the system was subjected to a continuous 15-minute operational stress test. Operating at the required throughput of 100 frames per second, the device successfully acquired, processed,

and encoded 90,000 consecutive audio frames. Throughout the testing period, the system exhibited no buffer overruns, memory leaks, or temporal drift. These results confirm that the proposed architecture provides a stable real-time processing platform for continuous wearable auscultation.

5.4.2 Integrated Signal Pipeline Validation

After confirming the real-time feasibility of the embedded implementation, the next step was to evaluate the fidelity of the complete signal processing chain. To thoroughly assess the system, two complementary testing methodologies were employed: a digital injection test to isolate and extract precise numerical metrics regarding the internal behavior of the embedded algorithms across various signal types, and an in-vivo acoustic evaluation to validate the morphological and clinical integrity of the signals under real-world conditions.

Emulated Acoustic End-to-End Validation

To approximate a wearable deployment scenario and observe the signal evolution across the entire system, reference PCG signals were reproduced through a loudspeaker positioned 5 cm from the MEMS microphone. The captured audio was processed by the embedded IIR filter, encoded with LC3 at 32 kbps, transmitted, and decoded on a host PC.

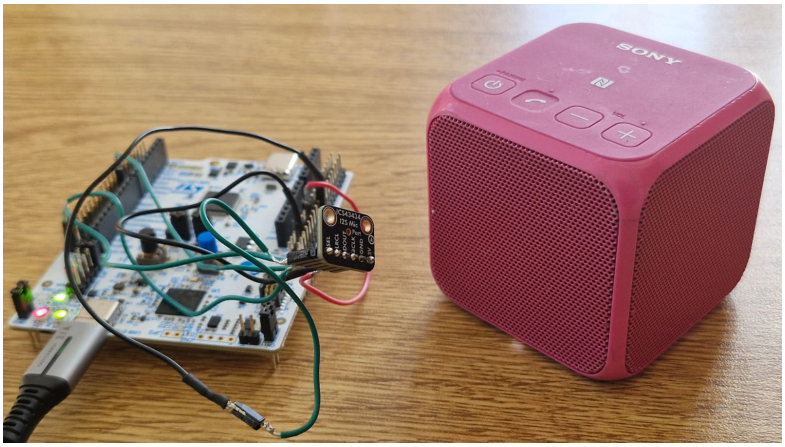


Figure 5.17: Acoustic end-to-end validation setup.

The experimental architecture is defined by the block diagram in Figure 5.18, which establishes five specific extraction points (a through e) for comparative analysis.

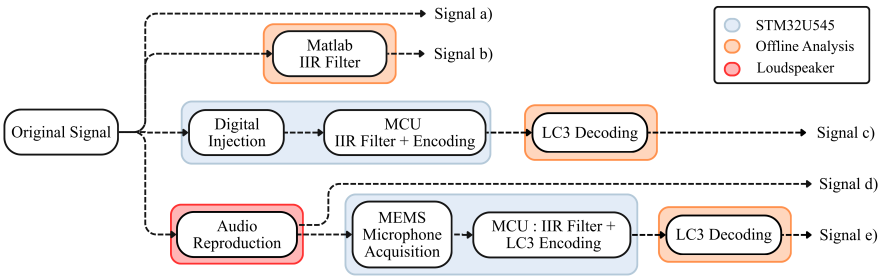


Figure 5.18: Block diagram of the end-to-end signal validation pipeline, illustrating the five extraction points for comparative analysis: (a) original reference, (b) software-filtered reference, (c) hardware-filtered via digital injection, (d) raw acoustic capture, and (e) hardware-filtered acoustic capture.

To visually assess the impact of each stage, Figure 5.19 presents the time-domain waveforms and Continuous Wavelet Transform (CWT) scalograms, strictly mapped to the extraction points of the block diagram:

- *Panel 1 (Extraction point a):* The original reference PCG signal.
- *Panel 2 (Extraction point b):* The ideal MATLAB-filtered reference.
- *Panel 3 (Extraction point c):* The hardware-filtered signal via digital injection (MCU performance).
- *Panel 4 (Extraction point d):* The raw, unfiltered acoustic capture via the MEMS microphone.
- *Panel 5 (Extraction point e):* The final embedded output: acoustic capture after MCU filtering and LC3 decoding.

The digital stages (Panels 1–3) reveal that the MATLAB and MCU filtered signals exhibit the exact same attenuation of high-frequency components and enhancement of the cardiac band, confirming that the embedded filter reproduces the ideal mathematical response with negligible deviation.

The acoustic stages (Panels 4–5) show the expected coloration introduced by the loudspeaker–air–microphone chain, including excess mid-frequency energy and reduced low-frequency content. After MCU filtering (Panel 5), however, the acoustic signal recovers the expected band-limited morphology, demonstrating that the embedded filter effectively suppresses both environmental noise and transducer-induced artifacts.

To complement the time–frequency analysis, Figure 5.20 compares the Power Spectral Density (PSD) of the same five extraction points. The MATLAB and MCU filtered spectra overlap almost perfectly across the entire cardiac band. The

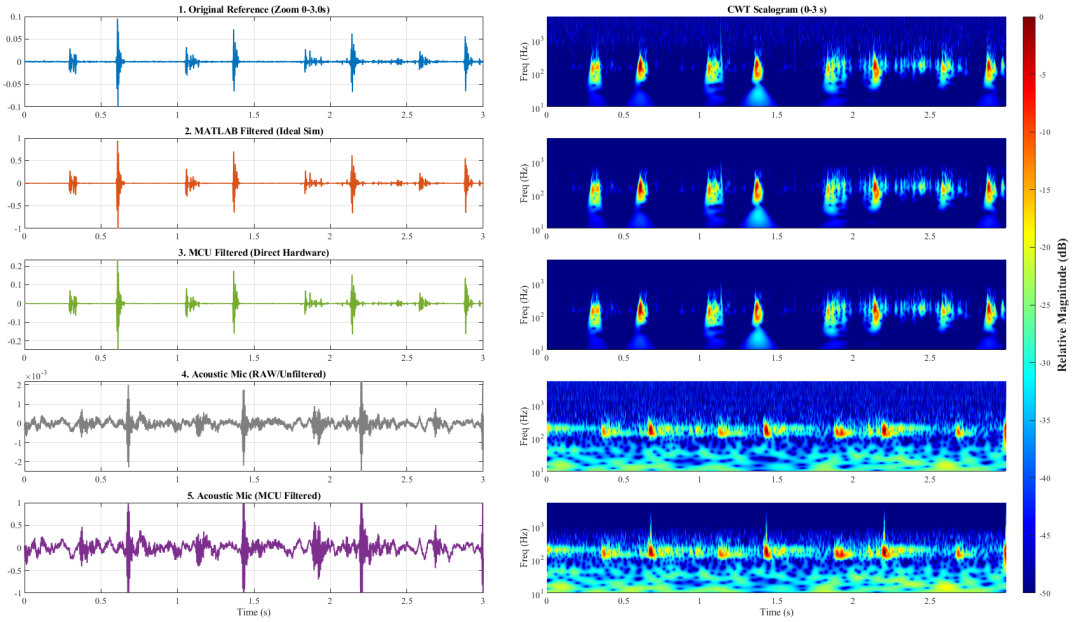


Figure 5.19: Time-domain waveform and CWT scalograms demonstrating the signal evolution. Panels 1 to 5 directly correspond to extraction points (a) to (e) from Figure 5.18.

acoustic recordings exhibit characteristic distortions (attenuation below 150 Hz) consistent with the non-flat low-frequency response of the consumer loudspeaker. Importantly, despite these acoustic limitations, the filtered acoustic signal still follows the expected spectral envelope within the passband, indicating that the MCU is able to correctly process the captured signal.

Digital Injection Validation

To isolate the performance of the embedded algorithms from acoustic distortions, reference PCG signals were digitally injected into the microcontroller's memory and processed entirely on-device. The MCU applied the IIR filter and the LC3 encoding, producing three signal versions for analysis: the original reference (x_{orig}), the MCU-filtered signal (x_{filt}), and the MCU-decoded signal (x_{dec}).

Initial comparisons between MATLAB simulations and the embedded outputs confirmed that the CMSIS-DSP implementation introduces negligible numerical deviation. Likewise, the encoded and decoded signals match the reference LC3 behavior, with distortion levels strictly consistent with the codec's expected performance at 32 kbps.

Figure 5.21 provides a comprehensive visual summary of this process for a

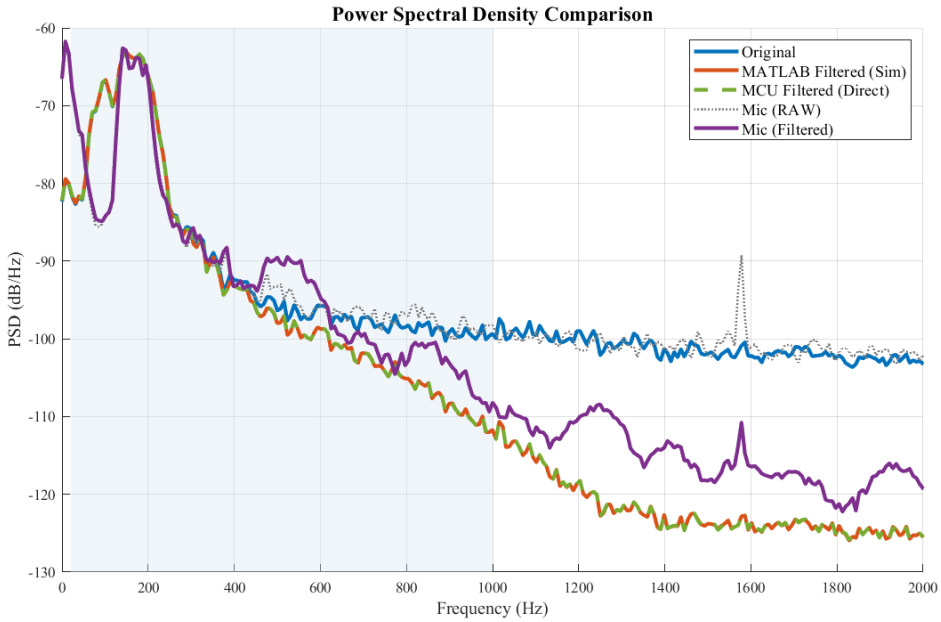


Figure 5.20: Power Spectral Density across the five extraction points: (a) original reference, (b) MATLAB-filtered, (c) MCU-filtered, (d) raw microphone capture, and (e) microphone capture after MCU filtering.

representative PCG signal. Panels A–C display the macroscopic time-domain waveforms, while the 50 ms zoomed segments (Panels D–F) demonstrate that the temporal morphology of the cardiac cycle is preserved with near-perfect phase alignment, particularly between the filtered and decoded stages (Panel E). Furthermore, the Power Spectral Density (Panel G) and the Continuous Wavelet Transform scalograms (Panels H–I) confirm that the decoded signal retains the characteristic spectral envelope and time-frequency energy distribution of the fundamental heart sounds.

To thoroughly assess the clinical viability of the pipeline, three pairwise comparisons were performed across four representative signals: a normal PCG, a pathological murmur, an extra heart sounds recording, and an ambient noise stress test. Table 5.14 details these quantitative results.

Original vs. Filtered For the original vs. filtered comparison, classical error metrics (SNR and PRD) are mathematically degraded due to the intentional ∼8 dB gain applied by the filter in the cardiac band. However, the envelope correlation exceeds 0.997 for all cardiac signals, and the energy increase in the 20–400 Hz band (ΔE_{20-400}) ranges from +511% to +523%, precisely matching the designed spectral

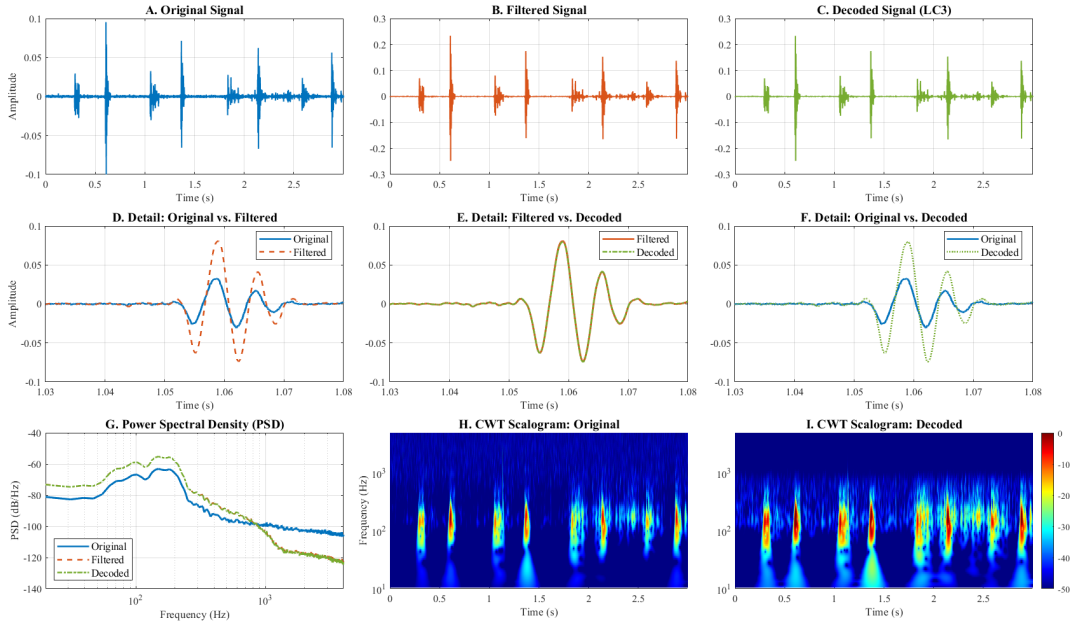


Figure 5.21: End-to-end digital injection validation for a representative PCG signal. (A–C) Time-domain waveforms for the original reference, MCU-filtered, and LC3-decoded stages. (D–F) Detailed 50 ms temporal overlays illustrating phase alignment and amplitude modifications. (G) Power Spectral Density (PSD) comparison across the three processing stages. (H–I) Continuous Wavelet Transform (CWT) scalograms demonstrating the preservation of clinically relevant time-frequency diagnostic features after on-device filtering and decoding.

shaping without introducing temporal distortion.

Filtered vs. Decoded For the filtered vs. decoded comparison, the metrics demonstrate exceptional codec transparency. The PRD remains below 1.6%, SNR exceeds 36 dB, and envelope correlation is above 0.9999. This confirms that LC3 at 32 kbps behaves almost losslessly on the MCU-filtered signals.

Original vs. Decoded Finally, the original vs. decoded comparison evaluates the end-to-end performance. Despite the filter's amplification, envelope correlation remains highly stable (≥ 0.997) for all clinically relevant signals, and the coherence in the cardiac band exceeds 0.99. Heart rate and S1–S2 timing errors remain effectively zero, except for the ExtraHS signal where the complex morphology leads to a slight increase (still below 2%).

Overall, these results validate that the complete embedded implementation is mathematically faithful to the reference models. When combined with the acoustic

Table 5.14: Digital injection test: comprehensive pairwise comparison metrics for the embedded pipeline.

Comparison	Signal	SNR (dB)	PRD (%)	Env. Corr.	LSD (dB)	Coh. (<500Hz)	ΔE_{20-400} (%)	HR err (%)	S1-S2 err (%)
Orig vs. Filt	Normal PCG	-3.35	147.03	0.9988	16.38	1.000	+511.85	0.00	0.00
	Murmur	-3.50	149.70	1.0000	7.78	1.000	+523.27	0.00	0.00
	Extra HS	-3.37	147.39	0.9976	15.54	1.000	+515.25	0.01	0.01
	Ambient noise	-1.20	114.84	0.7887	17.63	1.000	+472.99	—	—
Filt vs. Dec	Normal PCG	38.84	1.14	1.0000	0.65	0.998	+0.22	0.00	0.00
	Murmur	36.35	1.52	0.9999	0.12	0.998	-0.04	0.00	0.00
	Extra HS	36.18	1.55	0.9999	0.56	0.997	-0.08	1.71	1.68
	Ambient noise	25.21	5.49	0.9981	3.75	0.998	-0.51	—	—
Orig vs. Dec	Normal PCG	-3.34	146.92	0.9988	16.61	0.998	+513.17	0.00	0.00
	Murmur	-3.50	149.66	0.9999	7.79	0.997	+523.04	0.00	0.00
	Extra HS	-3.36	147.27	0.9975	15.79	0.997	+514.78	1.70	1.67
	Ambient noise	-1.13	113.85	0.7929	20.94	0.998	+470.09	—	—

Note: For 'Orig vs Filt' and 'Orig vs Dec', the negative SNR and high PRD reflect the intentional gain applied by the IIR filter, not signal degradation. The slightly higher HR/S1-S2 errors in the ExtraHS signal are due to its transient, non-periodic morphology.

validation, they confirm that the full acquisition, processing, and transmission chain preserves the diagnostic integrity of phonocardiogram signals while operating in real time on the STM32U545.

In-Vivo Acoustic Clinical Validation

The loudspeaker-based test confirmed that the system could process captured sound along the entire hardware path, but it also introduced unavoidable coloration from the speaker itself. To verify the system's true diagnostic performance under real clinical conditions, a final *in-vivo* evaluation was performed. Unlike previous test that relied on a pre-recorded reference signal for direct comparison, this evaluation utilized the embedded system's direct recording, capturing an acoustic signal directly from a human subject, fully bypassing the loudspeaker response.

The results of this direct hardware processing, shown in Figure 5.22, demonstrate that the system preserves both the structural and spectral characteristics of the natural cardiac signal.

The time-domain waveform shows that the processed signal maintains a clear and stable morphology. The single-window PSD and the CWT scalograms confirm that the main acoustic energy remains within the cardiac frequency band, while high-frequency artifacts are effectively suppressed. The CWT representation aligns with the temporal waveform and clearly highlights the rhythmic energy bursts associated with the S1 and S2 heart sounds.

The lower section of the dashboard illustrates the heartbeat detection algorithm applied to the hardware-filtered signal. The temporal waveform and Hilbert envelope provide an accurate heart rate estimation (85.1 bpm) and reliable intervals.

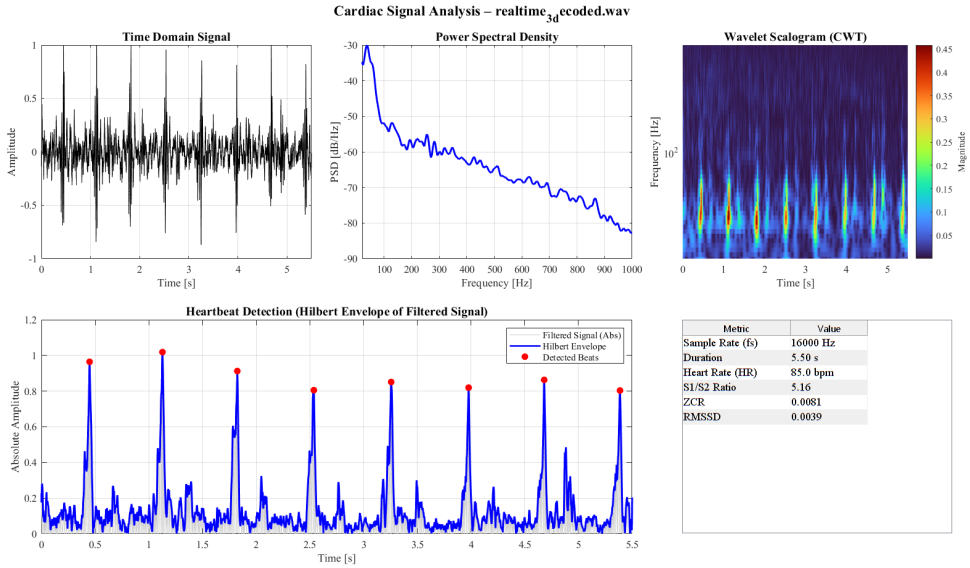


Figure 5.22: Comprehensive clinical analysis of the real audio capture after embedded processing. The dashboard shows the time-domain morphology, Power Spectral Density, 2D Wavelet Scalogram, and the heartbeat detection envelope used to compute clinical metrics. The low ZCR and RMSSD values confirm the absence of high-frequency noise and amplitude artifacts.

The peaks in the envelope (red trace) correspond to the energy peaks observed in the CWT scalogram, clearly identifying the S1 and S2 beats.

Other metrics further confirm the high morphological quality of the captured signal. The very low Zero-Crossing Rate (ZCR = 0.0081) indicates a smooth waveform with minimal high-frequency noise, which would otherwise increase zero-crossing events. Likewise, the low Root Mean Square of Successive Differences (RMSSD = 0.0039) shows that the amplitude baseline is stable and free from sudden fluctuations.

Furthermore, the 3D topological scalogram in Figure 5.23 highlights the periodic energy bursts corresponding to S1 and S2 across time and frequency. Together, these results demonstrate that the acoustic-to-digital chain preserves the essential time-frequency features required for reliable clinical auscultation.

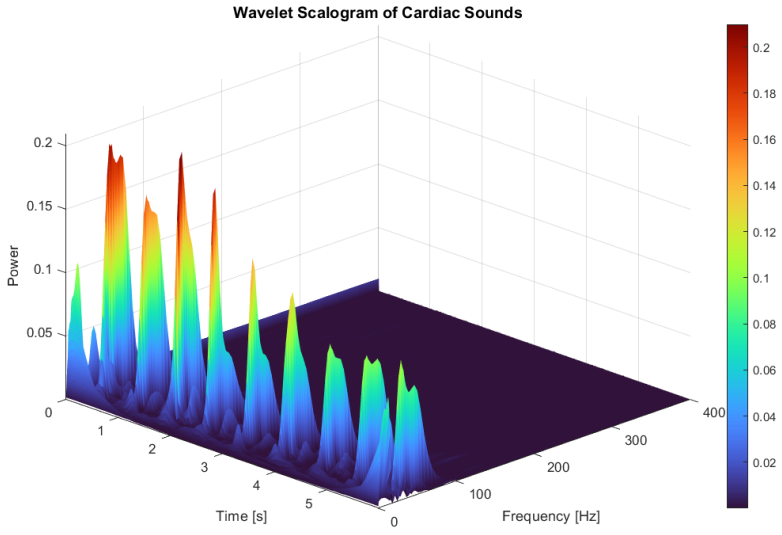


Figure 5.23: 3D Wavelet scalogram of the embedded output from an in-vivo auscultation, showing clear energy peaks associated with S1 and S2 heart sounds.

Chapter 6

Conclusions and Future Work

6.1 Discussion and Conclusions

The main objective of this thesis was to design and implement an audio driver for a wearable device capable of performing cardiac auscultation for phonocardiography. The results presented in this work demonstrate that the proposed architecture successfully achieves real-time acquisition, preprocessing, and compression on an ultra-low-power STM32U545 microcontroller, proving to be both computationally efficient and clinically viable. Table 6.1 summarizes the key performance indicators of the final integrated system.

Table 6.1: End-to-end system performance summary.

Component	Key Metric	Achieved Result
Acoustic Capture (MEMS)	*In-vivo* SNR	11.88 dB
IIR Preprocessing (MCU)	PRD vs. MATLAB reference	2.26 %
LC3 Compression (32 kbps)	Cardiac band energy preservation	> 98.0 %
Integrated Digital Pipeline	End-to-end Envelope Correlation (r)	> 0.99
Real-Time Execution	RTF (10 ms frame pipeline)	0.34
Overall Latency	End-to-end processing delay	≈ 15.9 ms

A primary contribution of this work is pioneering the application of the new Bluetooth LE Audio standard (specifically the LC3 codec) for a medical-grade wearable device. The successful integration of LC3 into a phonocardiography pipeline challenges the traditional paradigm of remote auscultation. While previous

studies have proposed custom compression algorithms for PCG signals—such as dictionary-based methods [20] or proprietary wavelet transforms—these approaches often require specialized decoders on the receiver side, severely limiting clinical interoperability. This thesis validates that a standardized perceptual audio codec, originally designed for consumer audio, can meet the stringent fidelity requirements of cardiac diagnostics.

When compared to existing literature, the proposed pipeline demonstrates highly competitive performance. Custom PCG compression schemes typically report Percentage Root-mean-square Difference (PRD) values between 2% and 7% for compression ratios ranging from 8:1 to 15:1 [20]. In this work, the baseline LC3 codec evaluation achieves a 12:1 compression ratio (at 32 kbps) with an average PRD of 4.62%. However, the complete proposed pipeline–which integrates a targeted digital pre-filtering stage–significantly enhances encoding efficiency, driving the average end-to-end PRD down to an impressive 1.19%. Furthermore, unlike custom algorithms that demand significant computational resources for vector quantization or complex mathematical transforms, the LC3 encoder executes highly efficiently on the ARM Cortex-M architecture.

The embedded audio pipeline—data acquisition, digital filtering, and LC3 compression—executes in only 3.4 ms per 10 ms frame, corresponding to a 34% CPU load. By relying on hardware accelerators and DMA transfers, the system maintains stable continuous operation while leaving a 66% idle margin. This headroom ensures that the microcontroller can handle wireless communication, interrupts from other sensors, and low-power modes. When combined with the LC3 algorithmic delay of 12.5 ms, the total end-to-end latency is approximately 15.9 ms—well below the 45 ms threshold required for smooth remote auscultation.

A key finding of this work is the strong synergy between the spectral conditioning stage and the LC3 compression algorithm. The real-time IIR filter successfully reshapes the acoustic spectrum, amplifying the clinically relevant cardiac bands by 8 to 12 dB while attenuating the high-frequency noise floor. By attenuating out-of-band noise before encoding, the filter removes spectral entropy irrelevant for diagnosis. This enables the codec to allocate its entire bit budget to the frequencies that matter, compressing the audio stream at 32 kbps without losing morphological fidelity.

Most importantly, the validation results show that the entire acoustic-to-digital chain preserves the diagnostic integrity of the phonocardiogram. Classical fidelity metrics are often insufficient in this context as they penalize intentional filtering gain. However, clinically-oriented evaluation confirms that the temporal morphology of the cardiac cycle, quantified by an envelope correlation above 0.999, is preserved with near-unity fidelity. The precise timing of cardiac events remains undistorted, with a mean S1-S2 interval error of just 0.03%.

Finally, the architecture relies exclusively on commercial off-the-shelf components

and standardized communication protocols. This design choice validates that high-fidelity continuous cardiac monitoring does not require expensive, proprietary digital signal processors or custom receiver applications, laying the groundwork for proactive telemedicine, computer-assisted diagnosis at the edge, and overall healthcare cost reduction.

6.2 Limitations and Future Work

While the proposed system demonstrates strong performance under controlled conditions, certain limitations highlight clear directions for future development to prepare the device for a final wearable implementation.

Currently, the digital preprocessing relies on a static IIR band-pass filter. While effective for stable, stationary noise, it cannot dynamically adapt to the natural variability of different patients or unpredictable motion artifacts. In a realistic wearable scenario, triboelectric noise caused by clothing friction will introduce low-frequency interference that overlaps with the 20 to 150 Hz cardiac band. Therefore, a primary focus for future work involves improving *robustness against motion artifacts and environmental interference*. This will require adaptive filtering strategies that adjust based on real-time spectral analysis or the continuously detected heart rate. Combining microphone data with the onboard accelerometer could enable motion-aware filtering and Active Noise Cancellation (ANC), while moving to a small microphone array would open the possibility of spatial beamforming.

Beyond signal conditioning, the available computational headroom offers opportunities for more advanced processing. Lightweight *edge-AI* algorithms could support basic on-device classification of normal versus abnormal heart sounds. Similarly, features derived from S1 and S2 may eventually support cuffless blood-pressure estimation [2]. Furthermore, as justified during the system configuration, the hardware is already equipped to capture pulmonary acoustics. Future firmware updates could implement parallel filtering pipelines to extract and analyze both cardiac and respiratory components simultaneously, unlocking true multimodal cardiopulmonary monitoring.

From a hardware perspective, all experimental validations were conducted using a microcontroller evaluation board and a raw breakout module. Consequently, the system must be *validated in its final physical form*. The next step involves transitioning to a custom printed circuit board sealed within a designed wearable enclosure. Because physical constraints, casing resonance, and skin-to-device coupling impedance will inevitably alter the frequency response, the system's acoustic front-end will require specific characterization using precision anechoic instruments.

Once this hardware is finalized, comprehensive clinical validation will be required.

The initial characterization was limited by the frequency response of a consumer loudspeaker, and while algorithmic equivalence was proven, large-scale *in-vivo* clinical trials across a diverse patient population and in real-world ambulatory environments are needed to statistically confirm diagnostic accuracy against gold-standard echocardiography. Crucially, this clinical phase must include a perceptual tuning of the IIR filter. While the current filter meets theoretical targets, its coefficients should be iteratively adjusted based on clinician feedback to match the acoustic profile of traditional stethoscopes, maximizing both diagnostic accuracy and perceptual comfort for the physician. This final phase will provide the evidence needed for regulatory approval and clinical adoption.

Appendix A

Firmware Implementation Details

A.1 Audio Acquisition Pipeline and State Machine

The processing pipeline implements continuous MEMS PCG acquisition through DMA-driven SAI/I²S, real-time IIR filtering, and LC3 compression within 10 ms frame deadlines.

The function `I2S_Mic_Mng()` implements a three-state sequencer for microphone data: `MIC_IDLE` (initialisation), `MIC_ON` (acquisition, preprocessing and LC3 encoding), and `MIC_OFF` (termination and statistics).

```
void I2S_Mic_Mng(void) {
    switch(mic_status) {
    case MIC_IDLE:
        MX_DSP_Init();                              // IIR
   coefficients
        enc = lc3_hr_setup_encoder(...);            // LC3 state init
        HAL_SAI_Receive_DMA(&hsai_BlockA1,          // <-- START DMA
                            (uint8_t*)audio_buffer, I2S_BUFFER_SIZE
   );
        frames_processed = 0;
        mic_status = MIC_ON;
        break;

    case MIC_ON:
        if(mic_dr_state != MIC_DR_NONE) {
            mic_dr_state = MIC_DR_NONE;

            Preprocess_Frame();                     // 0.28 ms
```

```
17              Encode_Frame();                          // 3.2 ms
18              frame_cycles_total += DWT->CYCCNT - start_frame;
19
20              if(++frames_processed >= TOTAL_FRAMES) {
21                  HAL_SAI_DMAStop(&hsai_BlockA1);      // <-- STOP
    DMA
22                  mic_status = MIC_OFF;
23                  print_statistics();
24              }
25          }
26          break;
27      case MIC_OFF: break;
28      }
29  }
```

Code A.1: DMA-driven Microphone manager state machine.

DMA callbacks ensure sub-frame latency by processing halves of the circular buffer:

```
1  void HAL_SAI_RxHalfCpltCallback(SAI_HandleTypeDef *hsai) {
2      if(hsai->Instance == SAI1_Block_A) {
3          memcpy(audio_buffer_lc3, audio_buffer, FRAME_BYTES);
4          mic_dr_state = MIC_DR_1ST_HALF;
5          I2S_Mic_Mng();
6      }
7  }
8
9  void HAL_SAI_RxCpltCallback(SAI_HandleTypeDef *hsai) {
10     if(hsai->Instance == SAI1_Block_A) {
11         memcpy(audio_buffer_lc3, &audio_buffer[I2S_BUFFER_SIZE/2],
       FRAME_BYTES);
12         mic_dr_state = MIC_DR_2ND_HALF;
13         I2S_Mic_Mng();
14     }
15 }
```

Code A.2: SAI1 DMA half/full-complete callbacks.

A.2 Real-Time Preprocessing (CMSIS-DSP)

The real-time preprocessing stage is implemented as a 4-stage cascaded biquad IIR filter using the CMSIS-DSP `arm_biquad_cascade_df1_f32()` routine. Coefficients are derived in MATLAB (Yule-Walker design) to emphasize the 20-400 Hz cardiac band.

```
1  void MX_DSP_Init(void) {
```

```
    arm_biquad_cascade_df1_init_f32(&S, 4, filter_coeffs,
    filter_state);
}

void Preprocess_Frame(void) {
    int32_t *input = &audio_buffer_lc3[0];

    /* S24->float32: normalize by 2^23 */
    for(uint32_t i = 0; i < FRAME_SAMPLES; i++)
        temp_float[i] = (float32_t)input[i] * INV_2P23;

    /* 4-stage cascade: 20-400 Hz emphasis */
    arm_biquad_cascade_df1_f32(&S, temp_float, temp_float,
    FRAME_SAMPLES);

    /* float32->S24: denormalize + saturation */
    for(uint32_t i = 0; i < FRAME_SAMPLES; i++)
        input[i] = __SSAT((int32_t)(temp_float[i]*8388608.0f), 24)
    ;
}
```

Code A.3: IIR initialization and per-frame processing.

A.3 LC3 Compression Pipeline

The `Encode_Frame()` function provides a thin wrapper around the LC3 reference encoder, operating on the preprocessed frame.

```
void Encode_Frame(void) {
    lc3_encode(enc,                     // Encoder state
               LC3_PCM_FORMAT_S24,      // Input format
               &audio_buffer_lc3[0],    // Preprocessed frame
               1,                       // nchannels
               48,                      // block_bytes (Output size
    )
               lc3_bytes);              // Bitstream
}
```

Code A.4: High-level LC3 encoding wrapper.

The LC3 encoder follows a standard audio codec pipeline: PCM loading → MDCT analysis → perceptual quantization → arithmetic coding and bitstream packing.

```
int lc3_encode(struct lc3_encoder* enc, enum lc3_pcm_format fmt,
               const void* pcm, int nbytes, void* out) {
    struct side_data side;

    load[fmt](enc, pcm, 1);             // S24->normalized float
```

```
    analyze(enc, nbytes, &side);          // MDCT+LTPF+TNS
    encode(enc, &side, nbytes, out);      // Quantize+pack
    return 0;
}

static void analyze(struct lc3_encoder* enc, int nbytes, struct side_data* side) {
    lc3_mdct_forward(enc->dt, enc->sr_pcm, enc->sr, xs, xd, xf);
    float e[LC3_MAX_BANDS];
    bool nn_flag = lc3_energy_compute(enc->dt, enc->sr, xf, e);
    side->bw = lc3_bwdet_run(enc->dt, enc->sr, e);     // BW est.
}
```

Code A.5: LC3 reference encode pipeline (excerpt).

Appendix B

Data Extraction and Validation Scripts

B.1 MATLAB Filter Design and Export Script

The following MATLAB script demonstrates the design of the 8th-order Yule-Walker filter used for PCG preprocessing. It defines the target frequency response, verifies system stability, converts the filter to Second Order Sections (SOS), and formats the coefficients into a C-compatible array structure required by the CMSIS-DSP library.

```
%% PCG EQUALIZER FILTER DESIGN - 16kHz, 10ms blocks
% Preprocessing: boost 10-400Hz cardiac band, strong HF
    attenuation
clear; close all; clc;

%% SYSTEM PARAMETERS
fs = 16000;                % Hz
Nyquist = fs/2;            % 8kHz
block_ms = 10;             % ms
block_samples = round(block_ms * fs / 1000);  % 160

%% 1. DESIGN H(f) WITH YULEWALK (custom magnitude curve)
f_norm = [0,    10/Nyquist,  400/Nyquist,  1000/Nyquist,  1];
m_gain = [0.1,     2.5,          2.5,           0.2,        0.05];
orden_yule = 8;  % Low order for STM32 (4 biquads)
[b_yule, a_yule] = yulewalk(orden_yule, f_norm, m_gain);

%% 2. EVALUATION: Stability
if all(abs(roots(a_yule)) < 1)
    fprintf('Filter is STABLE (All poles inside unit circle)\n');
else
```

```
    warning('Filter is UNSTABLE (Poles outside unit circle)!');
end

%% 3. CONVERT TO CMSIS-DSP BIQUADS (STM32 ready)
fprintf('\n=== CMSIS BIQUADS EXPORT ===\n');
[sos_yule, g_yule] = tf2sos(b_yule, a_yule);  % Second Order Sections

% Apply global gain ONLY to the first biquad
sos_yule(1,1:3) = sos_yule(1,1:3) * g_yule;
num_stages = size(sos_yule,1);

% CMSIS coefficients: b0, b1, b2, -a1, -a2 per stage (NOTE: a0=1 always)
coeffs_cmsis = [sos_yule(:,1:3), -sos_yule(:,5:6)];

fprintf('float32_t coeffs[%d] = {\n', 5*num_stages);
fprintf('%.6f, %.6f, %.6f, %.6f, %.6f,  %% stage 1\n', coeffs_cmsis(1,:));
for i=2:num_stages-1
    fprintf('%.6f, %.6f, %.6f, %.6f, %.6f,  %% stage %d\n', coeffs_cmsis(i,:), i);
end
fprintf('%.6f, %.6f, %.6f, %.6f, %.6f  %% stage %d\n};\n', coeffs_cmsis(end,:), num_stages);
```

Code B.1: PCG Equalizer Filter Design (Yule-Walker)

B.2 LC3 File Header Reconstruction (Python)

As detailed in Chapter 4, a live BLE stream only transmits the raw audio payload to conserve bandwidth. Therefore, to validate the embedded compression on a PC, the `.lc3` file header must be artificially reconstructed. This Python script parses the hex dump from the SWV console and prepends the exact 18-byte `lc3bin` header (plus optional flags) required by the `dlc3.exe` reference decoder.

```
#!/usr/bin/env python3
import struct
import subprocess

def create_lc3bin_header(frame_us, srate_hz, hrmode, bitrate, nchannels, nsamples):
    """Generates the exact lc3bin_header (20 bytes) + 2 bytes hrmode"""
    hdr_hrmode = 1 if hrmode else 0

    hdr = struct.pack('<HH H H H H H H H',  # little-endian
```

```
        0xCC1C,                     # file_id = 0x1C | (0xCC << 8)
        22,                         # header_size = 20 + 2
        srate_hz // 100,            # srate_100hz
        bitrate // 100,             # bitrate_100bps
        nchannels,                  # channels
        frame_us // 10,             # frame_10us
        0,                          # epmode = 0
        nsamples & 0xFFFF,          # nsamples_low
        nsamples >> 16              # nsamples_high
    )
    extra = struct.pack('<H', hdr_hrmode)
    return hdr + extra

# MCU Parameters
frame_us = 10000      # 10ms
srate_hz = 16000      # 16kHz
hrmode = False
bitrate = 32000       # 32 kbps
nchannels = 1
nsamples = 160        # 1 frame at 16kHz

header = create_lc3bin_header(frame_us, srate_hz, hrmode, bitrate,
    nchannels, nsamples)
frame_prefix = struct.pack('<H', len(frame_data)) # Frame size
   prefix

lc3_file = header + frame_prefix + frame_data
with open('reconstructed_audio.lc3', 'wb') as f:
    f.write(lc3_file)

# Call reference decoder
subprocess.run(["./bin/dlc3.exe", "reconstructed_audio.lc3", "
   decoded_output.wav"])
```

Code B.2: LC3 Header Generation and Decoding Script

Appendix C

Hardware Configuration

C.1 SAI1 Peripheral Configuration

The SAI1 peripheral is configured as a 16 kHz I^2S master receiver with 32-bit slots and a single active channel, used to interface the MEMS microphone.

Table C.1: SAI1 configuration: 16 kHz MEMS microphone interface.

Parameter	**Configuration**
Audio mode	`SAI_MODEMASTER_RX`
Protocol	I^2S Standard
Frame length	32 bits
Data size	24 bits (left-justified)
Sample rate	16 kHz
FIFO threshold	`SAI_FIFOTHRESHOLD_EMPTY`
DMA mode	Circular, double-buffered
Clock source	PLL1Q (HSE-derived)

Bibliography

[1] Roberto Francescon, Mohsen Hooshmand, Matteo Gadaleta, Enrico Grisan, Seung Keun Yoon, and Michele Rossi. «Toward lightweight biometric signal processing for wearable devices». In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2015, pp. 4190–4193. DOI: 10.1109/EMBC.2015.7319318 (cit. on pp. 4, 10, 23, 40, 44, 46).

[2] Rafi u.Shan Ahmad, Muhammad Shehzad Khan, Mohamed Elhousseini Hilal, Bangul Khan, Yuanting Zhang, and Bee Luan Khoo. «Advancements in wearable heart sounds devices for the monitoring of cardiovascular diseases». In: *SmartMat* 6 (1 Feb. 2025). ISSN: 2688819X. DOI: 10.1002/smm2.1311 (cit. on pp. 5, 97).

[3] Domenico Ragusa, Rens Baeyens, Danilo Pau, Elisa Marenzi, Jan Steckel, Walter Daems, Francesco Leporati, and Emanuele Torti. «Ultra-Efficient Compressed Phonocardiogram Classification on a Custom Embedded Neural Accelerator». In: *IEEE Internet of Things Journal* (2025). ISSN: 23274662. DOI: 10.1109/JIOT.2025.3635785 (cit. on pp. 6, 23, 46).

[4] Chengyu Liu et al. «An open access database for the evaluation of heart sound algorithms». In: *Physiological Measurement* 37 (12 Nov. 2016), pp. 2181–2213. ISSN: 13616579. DOI: 10.1088/0967-3334/37/12/2181 (cit. on p. 8).

[5] Akkarapol Sa-ngasoongsong, Jakkrit Kunthong, Venkatesh Sarangan, Xinwei Cai, and Satish T.S. Bukkapatnam. «A low-cost, portable, high-throughput wireless sensor system for phonocardiography applications». In: *Sensors (Switzerland)* 12 (8 Aug. 2012), pp. 10851–10870. ISSN: 14248220. DOI: 10.3390/s120810851 (cit. on pp. 7, 9, 14).

[6] Dennis Laurijssen, Toon Stas, Rens Baeyens, Kris Ides, Peter Delputte, Stijn Verhulst, Walter Daems, and Jan Steckel. «ModAu: Modernized Auscultation». In: *Proceedings of IEEE Sensors*. Institute of Electrical and Electronics Engineers Inc., 2023. ISBN: 9798350303872. DOI: 10.1109/SENSORS56945.2023.10325258 (cit. on p. 7).

[7] Hongxing Luo, Pablo Lamata, Salome Bazin, Thea Bautista, Natsuki Barclay, Mehrdad Shahmohammadi, Jolijn M. Lubrecht, Tammo Delhaas, and Frits W. Prinzen. «Smartphone as an electronic stethoscope: Factors influencing heart sound quality». In: *European Heart Journal - Digital Health* 3 (3 Sept. 2022), pp. 473–480. ISSN: 26343916. DOI: 10.1093/ehjdh/ztac044 (cit. on p. 7).

[8] Lukasz J. Nowak and Karolina M. Nowak. «Sound differences between electronic and acoustic stethoscopes». In: *BioMedical Engineering Online* 17 (1 Aug. 2018). ISSN: 1475925X. DOI: 10.1186/s12938-018-0540-2 (cit. on p. 8).

[9] Piero Malcovati and Andrea Baschirotto. «The evolution of integrated interfaces for MEMS microphones». In: *Micromachines* 9 (7 June 2018). ISSN: 2072666X. DOI: 10.3390/mi9070323 (cit. on pp. 10, 11).

[10] InvenSense. *ICS-43434: Multi-Mode Microphone with I2S Digital Output*. TDK Group. 2017. URL: https://invensense.tdk.com/wp-content/uploads/2016/02/DS-000069-ICS-43434-v1.1.pdf (cit. on p. 11).

[11] Matteo Zauli, Lorenzo Mistral Peppi, Luca Di Bonaventura, Valerio Antonio Arcobelli, Alberto Spadotto, Igor Diemberger, Valerio Coppola, Sabato Mellone, and Luca De Marchi. «Exploring Microphone Technologies for Digital Auscultation Devices». In: *Micromachines* 14 (11 Nov. 2023). ISSN: 2072666X. DOI: 10.3390/mi14112092 (cit. on pp. 11, 30, 62).

[12] NXP Semiconductors. *UM11732: I2S bus specification*. 2022. URL: https://www.nxp.com/docs/en/user-guide/UM11732.pdf (cit. on p. 12).

[13] Rene Jaros, Jiri Koutny, Martina Ladrova, and Radek Martinek. «Novel phonocardiography system for heartbeat detection from various locations». In: *Scientific Reports* 13 (1 Dec. 2023). ISSN: 20452322. DOI: 10.1038/s41598-023-41102-8 (cit. on pp. 14, 23, 45).

[14] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing*. 4th. Pearson, 2007. DOI: 10.1007/978-1-4615-0553-5 (cit. on pp. 15, 18).

[15] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd. Pearson, 2010. DOI: 10.1007/978-1-4615-0553-5 (cit. on pp. 15, 20).

[16] G. Bertoni and F. Grossi. *Capitolo 8: La compressione dei segnali audio (Elaborazione per il corso di Complementi di Misure Elettriche)*. http://at.dii.unipd.it/renato.gobbo/didattica/corsi/complementi_misure_elettriche/2012_13/elaborazione_bertoni_grossi/cap8.pdf. Università degli Studi di Padova. Accessed: 2026-02-23. 2013 (cit. on pp. 16, 17).

[17] Arm Ltd. *CMSIS-DSP: Arm Cortex Microcontroller Software Interface Standard*. Version 1.15.0, Online documentation. 2023. URL: https://developer.arm.com/Architectures/CMSIS (cit. on pp. 17, 34).

[18] Dah-Jing Jwo, Wei-Yeh Chang, and I-Hua Wu. «Windowing Techniques, the Welch Method for Improvement of Power Spectrum Estimation». In: *Computers, Materials and Continua* 67.3 (2021), pp. 3983–4003. ISSN: 1546-2218. DOI: https://doi.org/10.32604/cmc.2021.014752. URL: https://www.sciencedirect.com/science/article/pii/S152614922000168X (cit. on p. 19).

[19] Youngsin Kim, Mihyung Moon, Seokwhwan Moon, and Wonkyu Moon. «Effects of precise cardio sounds on the success rate of phonocardiography». In: *PLoS ONE* 19 (7 July 2024). ISSN: 19326203. DOI: 10.1371/journal.pone.0305404 (cit. on pp. 20, 23).

[20] Hong Tang, Jinhui Zhang, Jian Sun, Tianshuang Qiu, and Yongwan Park. «Phonocardiogram signal compression using sound repetition and vector quantization». In: *Computers in Biology and Medicine* 71 (Apr. 2016), pp. 24–34. ISSN: 0010-4825. DOI: 10.1016/J.COMPBIOMED.2016.01.017 (cit. on pp. 21, 23, 26, 40, 41, 46, 96).

[21] Gaber Hassan, Khalid M. Hosny, Mostafa M. Fouda, and Islam S. Fathi. «Efficient Compression of Fetal Phonocardiography Bio-Medical Signals for Internet of Healthcare Things». In: *IEEE Access* 11 (2023), pp. 122991–123003. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3329889 (cit. on pp. 23, 43, 44, 46).

[22] Islam S. Fathi, Mohamed Ali Ahmed, and M. A. Makhlouf. «An efficient compression technique for Foetal phonocardiogram signals in remote healthcare monitoring systems». In: *Multimedia Tools and Applications* 82 (13 May 2023), pp. 19993–20014. ISSN: 15737721. DOI: 10.1007/s11042-022-14259-z (cit. on pp. 23, 46).

[23] S. Bharadwaj et al. «PhysioEdge: Multimodal Compressive Sensing Platform for Wearable Health Monitoring». In: *arXiv preprint arXiv:2507.07645* (2025) (cit. on p. 23).

[24] Bluetooth SIG. *Bluetooth Low Energy Primer*. Accessed: February 15, 2026. Bluetooth Special Interest Group. 2024. URL: https://www.bluetooth.com/bluetooth-le-primer/ (cit. on p. 24).

[25] Mark Powell. *A Technical Overview of LC3*. Accessed: February 15, 2026. Bluetooth SIG Blog. 2020. URL: https://www.bluetooth.com/blog/a-technical-overview-of-lc3/ (cit. on pp. 24, 25).

[26] Lalit Kumar Baghel, Radhika Raina, and Suman Kumar. «Evolution of Bluetooth Classic Audio towards Bluetooth LE Audio: Challenges and Road Ahead». In: *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*. 2023, pp. 459–464. DOI: 10.1109/ICIIS58898.2023.10253545 (cit. on p. 26).

[27] STMicroelectronics. *STM32U535/545 - STMicroelectronics*. Accessed: 2024-05-20. 2024. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32u535-545.html (cit. on p. 29).

[28] STMicroelectronics. *AN4989 Application note: STM32 microcontroller debug toolbox*. Application Note Rev 4. STMicroelectronics, 2021. URL: https://www.st.com/resource/en/application_note/an4989-stm32-microcontroller-debug-toolbox-stmicroelectronics.pdf (cit. on p. 31).

[29] Bluetooth SIG. *Low Complexity Communication Codec (LC3) Specification v1.0.1*. Bluetooth Special Interest Group. Jan. 2024. URL: https://www.bluetooth.com/specifications/specs/low-complexity-communication-codec-1-0-1/ (cit. on pp. 36, 37).

[30] P. Bentley, G. Nordehn, M. Coimbra, and S. Mannor. *The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results*. http://www.peterjbentley.com/heartchal (cit. on p. 40).

[31] Na Mei, Hongxia Wang, Yatao Zhang, Feifei Liu, Xinge Jiang, and Shoushui Wei. «Classification of heart sounds based on quality assessment and wavelet scattering transform». In: *Computers in Biology and Medicine* 137 (Oct. 2021). ISSN: 18790534. DOI: 10.1016/j.compbiomed.2021.104814 (cit. on pp. 42, 46, 75).

[32] Nicholas E. Singh-Miller and Natasha Singh-Miller. «Using spectral acoustic features to identify abnormal heart sounds». In: *2016 Computing in Cardiology Conference (CinC)*. 2016, pp. 557–560 (cit. on p. 45).

[33] Parastoo Dehkordi et al. «Comparison of Different Methods for Estimating Cardiac Timings: A Comprehensive Multimodal Echocardiography Investigation». In: *Frontiers in Physiology* 10 (Aug. 2019). ISSN: 1664042X. DOI: 10.3389/fphys.2019.01057 (cit. on p. 45).

[34] Junbin Zang, Qi An, Bo Li, Zhidong Zhang, Libo Gao, and Chenyang Xue. «A novel wearable device integrating ECG and PCG for cardiac health monitoring». In: *Microsystems and Nanoengineering* 11 (1 Dec. 2025). ISSN: 20557434. DOI: 10.1038/s41378-024-00858-3 (cit. on p. 46).

[35] Awad Al-Zaben, Amjad Al-Fahoum, Muhannad Ababneh, Bassam Al-Naami, and Ghadeer Al-Omari. «Improved recovery of cardiac auscultation sounds using modified cosine transform and LSTM-based masking». In: *Medical and Biological Engineering and Computing* 62 (8 Aug. 2024), pp. 2485–2497. ISSN: 17410444. DOI: 10.1007/s11517-024-03088-x (cit. on p. 46).

[36] Manuel Blanco-Velasco, Fernando Cruz-Roldán, J. Ignacio Godino-Llorente, Joaquín Blanco-Velasco, Carlos Armiens-Aparicio, and Francisco López-Ferreras. «On the use of PRD and CR parameters for ECG compression». In: *Medical Engineering and Physics* 27 (9 Nov. 2005), pp. 798–802. ISSN: 13504533. DOI: 10.1016/j.medengphy.2005.02.007 (cit. on p. 46).

[37] Peter Jax and Peter Vary. «Bandwidth extension of speech based on vector quantization of subband spectra». In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* 1 (2004), pp. I–369 (cit. on pp. 46, 77).

[38] International Telecommunication Union. *Recommendation ITU-R BT.1359-1: Relative timing of sound and vision for broadcasting*. Tech. rep. ITU Radiocommunication Sector (ITU-R), 1998 (cit. on p. 49).