

Multiple-track PhonoCardioGraphy (PCG) and Artificial Intelligence (AI) to Detect Heart Defects

Saif Khan¹, Ye Zhou², Rohitaa Ravikumar³, Jade Huang⁴

Dept. of Electrical and Computer Engineering, Rice University

¹sbk7@rice.edu, ²yz202@rice.edu, ³rr71@rice.edu, ⁴ch110@rice.edu

Abstract—Cairdio, a handheld device, is designed to enhance early detection of heart anomalies, a key factor in reducing heart disease mortality rates. It leverages PCG technology, equipped with a 4-track recorder, and is supported by an AI-powered mobile app. The app interprets heartbeats with high accuracy and communicates data via Bluetooth Low Energy (BLE). Cairdio aims to be accessible for lower trained staff or professionals, improving early heart disease detection and potentially lowering mortality rates.

I. INTRODUCTION

Heart disease, a leading global health crisis, accounts for approximately 17.9 million deaths annually, as per the World Health Organization (WHO)[1]. Early detection is key to reduce fatalities and improve patient outcomes. Cairdio is aimed at affordable health screening that attaches to the patient's chest at four critical areas: the aortic valve, pulmonic valve, right and left ventricular areas. It records the primary heart sounds, S1 and S2, using a microphone connected to an ADC converter, currently driven by a Nordic chip.

Analysis of heart sounds is crucial to overcome the limitations of traditional manual screening. This involves filtering out noise and interference before analysis. Techniques like Li et al.'s best subsequence selection algorithm, leveraging heart sound periodicity as a quality measure, is essential for this purpose. This approach automates the heart sound quality assessment.

Our paper focuses on the development of Cairdio's AI and Hardware. We delve into the AI's role in identifying corrupted signal segments and the hardware aspect, specifically the synergy between a microcontroller and an FPGA for efficient high-speed data sampling.

II. AI IMPLEMENTATION

A. Dataset

The CirCor DigiScope dataset comprises 5,282 Phonocardiogram (PCG) recordings obtained from four primary auscultation locations for a total of 1,568 patients [2]. These auscultation locations include the pulmonary valve (PV), aortic valve (AV), mitral valve (MV), tricuspid valve (TV), and other miscellaneous locations (Phc). Human experts manually labeled each recording to ascertain the presence, absence, or uncertainty of a cardiac murmur at each auscultation location, forming location-based murmur labels. Sixty percent of the CirCor dataset, constituting 942 patients and 3,163 recordings, was made publicly available as the training set.

B. Data Pre-processing

The preprocessing of heart sound signal includes key steps for enhancing signal quality. It starts with removing spikes that exceed thrice the average amplitude, followed by mitigating baseline wandering through a high-pass Butterworth filter with a 2 Hz cut-off frequency. The process concludes with normalizing the signal against its standard deviation, uniformly representing amplitudes and thereby improving the signal's quality and reliability for further analysis and applications.

C. Proposed Algorithm

1) Quality Index

Heart sounds, fundamental to cardiovascular diagnosis, result from heart valve closure, blood flow, and cardiac muscle contractions, repeating in each cardiac cycle. The quasi-cyclostationary nature of heart sound signals, affirmed by the consistent cycle duration, ensures that the heart rate undergoes gradual changes rather than abrupt shifts [1,2]. Represented as $x(t)$, a digital heart sound signal sequence that exhibits quasi-periodicity.

$$R_x(t, \tau) = \lim_{N \rightarrow -\infty} \frac{1}{2N + 1} \sum_{n=-N}^N x\left(t + \frac{\tau}{2} + nT\right) \\ \times x\left(t - \frac{\tau}{2} + nT\right) \quad (1)$$

$R_x(t, \tau)$ is a periodic function, i.e., $R_x(t, \tau) = R_x(t+T, \tau)$. We expand $R_x(t, \tau)$ using the Fourier series as

$$R_x(t, \tau) = \sum_{m=-\infty}^{\infty} R_x\left(\frac{m}{T}, \tau\right) e^{j2\pi \frac{mt}{T}} \quad (2)$$

where m is a real number, and m/T is called the cycle frequency, denoted as α . Equation 2 becomes

$$R_x(t, \tau) = \sum_{\alpha=-\infty}^{\infty} R_x(\alpha, \tau) e^{j2\pi \alpha t} \quad (3)$$

The coefficient of the Fourier series is given as

$$R_x(\alpha, \tau) = \langle x(t + \frac{\tau}{2})x(t - \frac{\tau}{2}) e^{-j2\pi \alpha t} \rangle_t \quad (4)$$

where the operator $\langle \rangle_t$ denotes the time average. $R_x(\alpha, \tau)$ is called the cyclic correlation function, which degenerates into a traditional correlation when the cycle frequency α is zero.

In the extreme case, the basic cycle frequency of the heart sound signal is $\alpha = 1/T$. $R_x(\alpha, \tau) \neq 0$ only if the cycle frequency is $k\alpha$, and $R_x(\alpha, \tau) = 0$ elsewhere, where k is an integer. However, the cycle duration of a normal heart sound signal is not fixed; it varies with time. This is known as heart rate variability (HRV). Thus, $R_x(\alpha, \tau) \neq 0$ even if α is any real number. $R_x(\alpha, \tau)$ can be transformed into the frequency domain via the Fourier transform. That is.

$$S_x(\alpha, f) = \int_{-\infty}^{\infty} R_x(\alpha, \tau) e^{-j2\pi f \tau} d\tau \quad (5)$$

$S_x(\alpha, f)$ is referred to as the cyclic spectral density. In any stochastic process for which $R_x(\alpha, \tau) \neq 0$ or $S_x(\alpha, f) \neq 0$, the process exhibits a certain degree of cyclostationarity at cycle frequency α . In this paper, the analysis in the cycle frequency domain is of primary interest. We can get the cycle frequency spectral density (CFSD) using the integral.

$$\gamma_x(\alpha) = \int_{-\infty}^{\infty} |S_{\alpha}x(f)| df \quad (6)$$

The relative rate defines the quality index for a heart sound signal.

$$d(\eta) = \frac{\gamma_x(\eta)}{\int_{\beta}^0 \gamma_x(\alpha) d\alpha} \quad (7)$$

where β is the maximum cycle frequency considered, and η is the basic cycle frequency indicated by the first peak location of $\gamma_x(\alpha)$. Noise and interference distort the heart sound signal, thus degrading the cyclostationarity. It is therefore reasonable to conclude that a heart sound sequence with less noise and interference will have a high degree of periodicity and thus high quality. The quality index thus has the ability to act as a quality score for heart sound signals.

2) Time-varying quality index

A sliding window approach is employed for data preprocessing on the heart sound signal. The signal is segmented into windows of a specified duration i.e., 8s with step size of 0.1s to capture local variations. The degree of periodicity is computed for each window. The resulting degree of periodicity values, along with the corresponding time points at the center of each window, are then collected for subsequent analysis. This preprocessing step enables the examination of localized changes in the degree of periodicity across the heart sound signal.

Figure 1 depicts the amplitude versus time graph of the preprocessed signal, while Figure 2 illustrates the degree of periodicity versus time graph. The latter employs the quality index and sliding window approach to capture localized variations in the signal's periodicity.

III. HARDWARE FOR NEXT PROTOTYPE

This project encompasses the utilization of an ESP32-S3-WROOM-2 module, which is anticipated to be integrated with an FPGA by a subsequent team. The primary objective

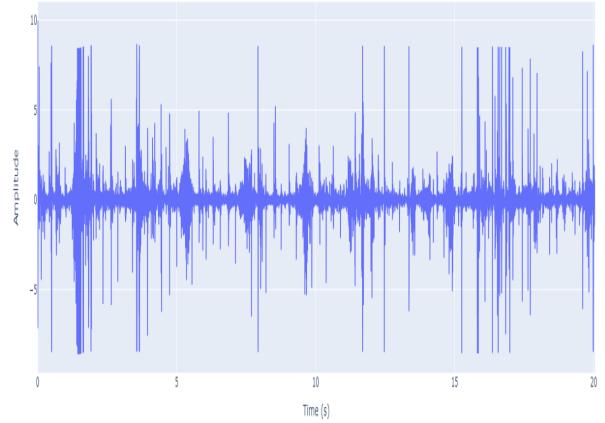


Fig. 1. Amplitude vs time graph of input signal

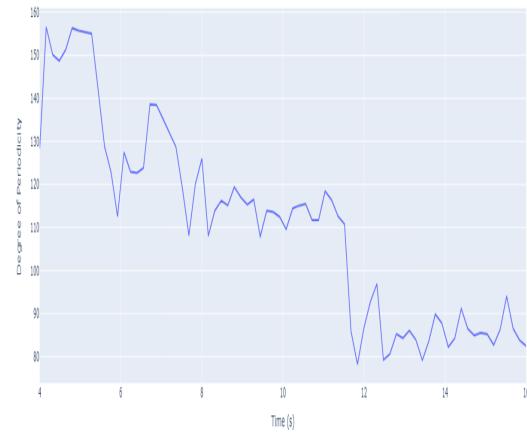


Fig. 2. Degree of Periodicity vs time graph of input signal

of this initiative is to examine the interfacing capabilities of the ESP32-S3 with the FPGA. The interface is meticulously engineered on a Printed Circuit Board (PCB) using Altium Designer [5], ensuring compatibility with a range of communication protocols such as SPI, I2C, USB OTG, and UART. This design choice allows for efficient and flexible communication with the Xilinx Zynq 7000 FPGA [6], with protocol selection being contingent upon the specific speed and distance requisites of the application.

Two versions of the PCB were designed, but only one was produced due to time constraints. Version 1 includes I2C, SPI, and USB OTG, while Version 2 adds UART and an external microSD card slot for additional data storage. This report will describe Version 2, but only Version 1 will be tested. The tests for Version 1 are considered representative for both versions. Due to the unavailability of the ESP32-S3-WROOM-2 chip, the ESP32-S3-WROOM-1N16R8 [8] was used for testing Version 1.

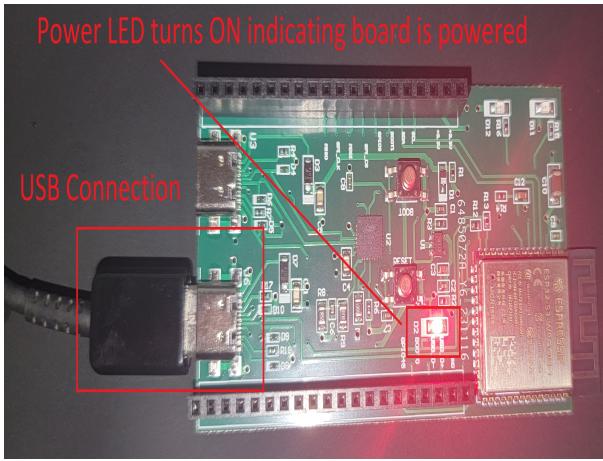


Fig. 3. ESP32-S3 Board Version 1

A. PCB Layout and Design

The project schematic is divided into 3 parts : Main IC , IC Peripherals and Connector Schematic.

Figure 4 is of Main IC Schematic. This schematic contains main ESP32-S3 Chip , USB OTG Type C connector connected directly to the ESP32 chip and two leds for debugging purposes.

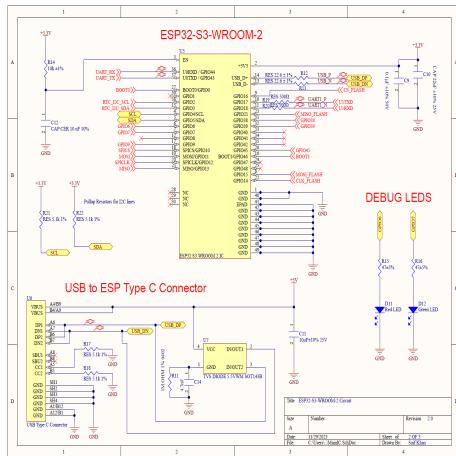


Fig. 4. ESP32-S3 Board Version 2 Main IC Schematic

Figure 5 is of Peripherals Schematic. This schematic contains the voltage regulator for power level shifting the voltage from 5V, obtained from USB connection to 3.3V to power the ESP32-S3 chip. The PCB contains USB to Serial which allows user to easily flash the micro-controller without any external JTAG interface only using USB Connection. The boot circuit allows the user to put the ESP32 into boot mode where a program can be flashed and then switch to Joint controller mode where the program can be read[8 ,Pg 14].

The design also contains two connectors J1 and J2 for GPIO, I2C, SPI and UART connections. The J3 is the MICRO SD Card connected to the ESP32-S3 Chip. These connectors

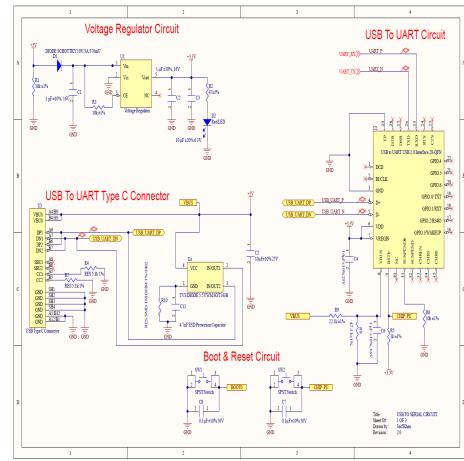


Fig. 5. ESP32-S3 Board Version 2 Peripherals Schematic.

are not shown here due to space constraint, please refer to the github [7].

The full layout is shown in the image below:

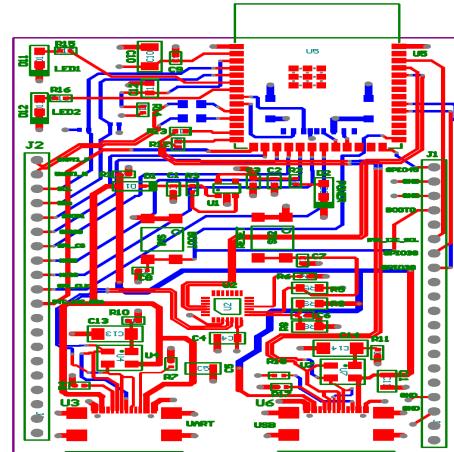


Fig. 6. Complete PCB layout in Altium.

This design has four layers: Layer 1 (TOP Signal), Layer 2 (GND), Layer 3 (POWER), and Layer 4 (BOTTOM). Layers 1 and 2 handle signal routing, with Layer 2 also featuring a GND polygon pour [9]. Layer 3 includes a +3.3V polygon pour for power and ground via connections. GPIO trace widths range from 12 to 15 mils, while power and ground traces are between 20 and 30 mils [9]. Vias have 31.5 mil pads and a 24 mil hole size.

Power traces are placed close to their respective components to reduce inductance and improve capacitor performance during transient events. The BOTTOM layer carries GPIO signal traces only, while the power traces (+3.3V and 5V) on the TOP layer are near the IC [10].

For testing, VS CODE with the ESP-IDF extension was used [12]. The board is connected via USB-A to USB-C cable (Figure 3). An issue was detected during flashing which was that the BOOT0 pushbutton was shorting the BOOT0 (GPIO0)

pin to ground when not pressed and when it is pressed it is still shorting it to the ground and this is because of a mistake made in the BOOT0 Pushbutton schematic and was removed from PCB instead a 11kohm resistor was used to pull the GPIO0 high to allow the microcontroller to boot the program stored in flash. The setup is shown in figure 10.

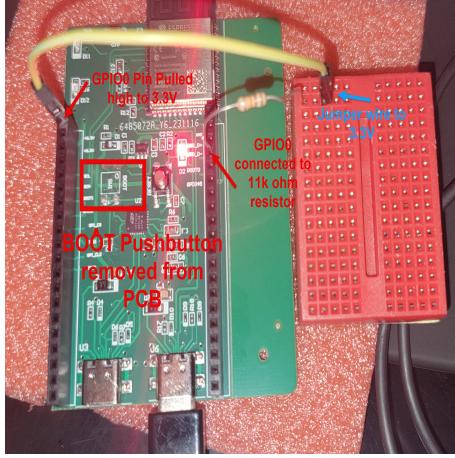


Fig. 7. BOOT Push-button removed.

Once the board is connected to a PC it will appear as COM port, an example program called Hello World found in VSCode ESP-IDF site [12] was used to flash to the microcontroller. Figure 9 first checks if the correct COM port is selected , then we select USB to UART communication is used to build, flash and monitor the output all using a single icon shown in figure 9.

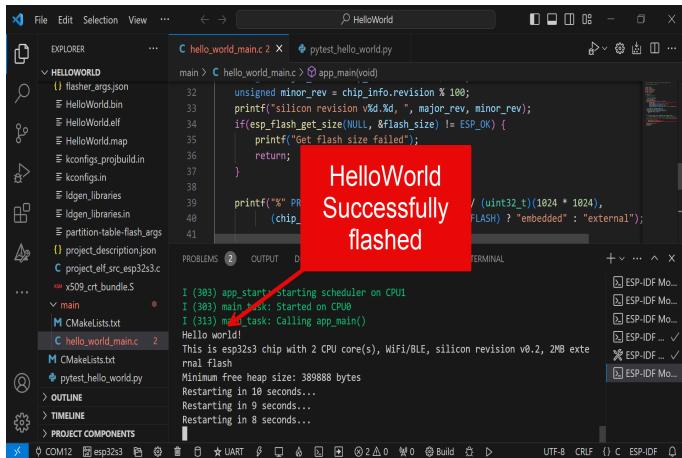


Fig. 8. Hello World Successfully Flashed onto PCB.

B. FPGA Implementation

Selecting an FPGA with sufficient processing power and resources is crucial for efficient audio data processing. The FPGA must also be compatible with the Xilinx Audio I2S interface for seamless integration. The ZedBoard Zynq-7000, meeting these requirements, is the ideal choice for the heart of

our audio processing system, where it will execute real-time audio algorithms.

In our system, the ADAU1761 audio Codec chip on the ZedBoard handles audio processing. Configured through the I2C bus, this chip uses its two Analog-to-Digital Converters (ADCs) to sample stereo audio at 48kHz. These digital samples are then sent to the Zynq chip via the I2S audio bus for additional processing.

The Zynq chip features a Numerically Controlled Oscillator (NCO) that generates sine wave samples at specific frequencies. These samples are superimposed on the incoming audio, creating a composite stream. This mixed audio is sent through the I2S bus to the Codec's DACs and played on a speaker or earphone connected to the chip's line-out port.

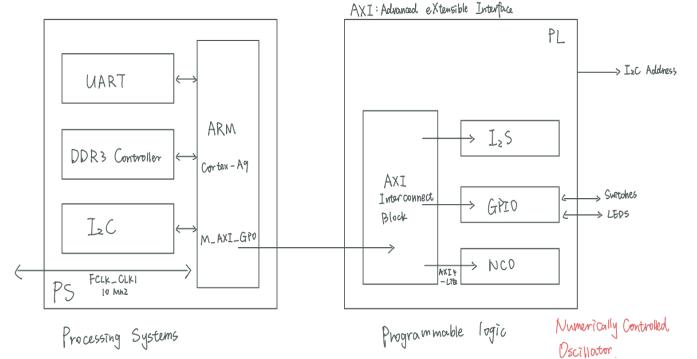


Fig. 9. Design Flow for FPGA Implementation.

1) Block Design in Vivado: The system features a ZedBoard with an ADAU1761 Codec chip, having 24-bit ADCs and DACs for audio, supporting 8kHz to 96kHz sampling. It samples stereo audio at 48kHz and sends it to the Zynq chip via 'zed-audio-ctrl' IP core. The Zynq chip uses an NCO to generate sine waves, merged with audio via ZedBoard switches. The mixed signal goes through the I2S bus to the Codec's DACs for output. It accepts inputs from microphones or analog sources.

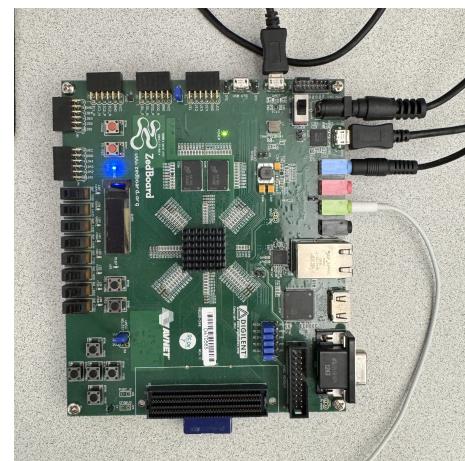


Fig. 10. Board Interface for FPGA Implementation.

2) *Audio Software Application in Vitis HLS*: We're creating a software that combines all our IP modules into a DSP system. The setup for the ZedBoard audio codec will be explained, involving hardware registers. Each IP peripheral added in IP Integrator automatically receives a base memory address, available in a Xilinx C header file created with Zynq Processing System designs[13]. Also, we've developed functions to initialize the audio codec and I2C interface in the Zynq PS.

IV. RESULTS

A. AI Results

Figure 11 exhibits the binary mask plot derived from the annotations of the input signal. Figure 12 showcases a binary mask plot generated from the degree of periodicity. In this binary mask, a value of 1 denotes a clean segment, while 0 signifies a noisy segment. The threshold for the degree peak values to generate the binary mask is set at 35%, where values below this threshold are considered noise, and those equal to or above 35% are regarded as the clean signal. A comparison between the binary mask plot derived from the annotations of the input signal and the one derived from the degree of periodicity reveals a close alignment. This alignment indicates the successful automation of the corrupted segment detection process using our approach.

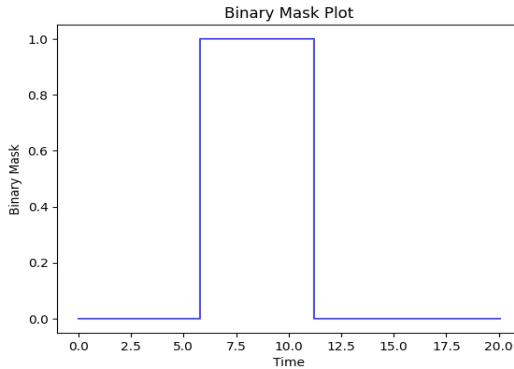


Fig. 11.

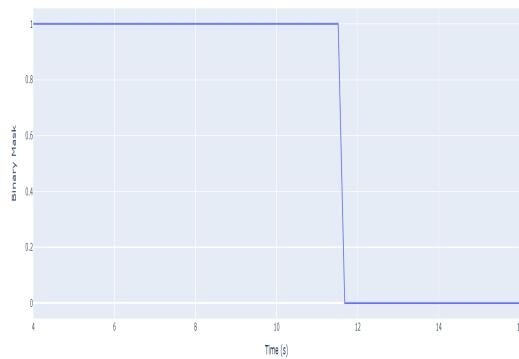


Fig. 12.

B. PCB Layout and Design

The overall design and testing was successful, Figure 11 shows an example hello world program that was successfully flashed onto the ESP32-S3 chip using USB Type C with the USB to UART module onboard the PCB. The BOOT pushbutton which was pulling the GPIO0 (BOOT0) pin to ground had to be desoldered from the PCB and pulled high using an external resistor to bring the micro-controller out of BOOT mode and this resolved the issue. Further testing is still required of the PCB.

C. FPGA Result

Using the Zedboard's switches, you can mix sine waves into audio played through a speaker. The Zynq chip processes this digital audio data, adding sine waves, and sends it back to the ADAU1761 Codec. Finally, the ADAU1761's DAC converter blends the original audio with the sine waves and outputs this mixed analog signal to the speaker or earphone.

V. DISCUSSION

A. AI Implementation

In summary, this paper introduces a method for heart sound signal quality assessment. Acknowledging the importance of signal quality in automatic heart sound signal analysis, the proposed approach provides a foundation for obtaining reliable results. With a focus on signal quality and automated detection of corrupted segments, the method offers a promising avenue for the development of more accurate and efficient diagnostic systems in cardiovascular health.

B. PCB Layout and Design

The ESP32-S3 contains all of the functionality to interface with an FPGA with an Onboard Bluetooth low energy (BLE) and WiFi Module. This custom PCB shows that an ESP32-S3 can be created to easily interface with an FPGA using I2C,SPI,UART and USB OTG (On-the-go). The USB to UART module onboard allows users to easily flash programs on to the board. This design can be used further along with the Xilinx FPGA to create a module to that can sample data ADC data at high frequencies and transfer data using WIFI and BLE.

C. FPGA Implementation

Configuring and communicating with the ZedBoard's ADAU1761 audio codec and custom peripherals in the PL involves identifying memory-mapped base addresses and offsets. This includes configuring the codec through control register addresses, managing audio samples via an audio controller block in the PL, and using generated software drivers for communication with these peripherals.

VI. ACKNOWLEDGMENT

We would like to thank Prof. Joseph Young, Rick Jones, Dr.Eric Welsh, Darya Yelshyna, Arnab Saha, Carlos Alejano, Annim Banerjee and the entire Healthseers team for their support.

REFERENCES

- [1] "Cardiovascular diseases," World Health Organization. [Online]. Available: www.who.int/health-topics/cardiovascular-diseases#tab_1. [Accessed: 01-Dec-2023]
- [2] Reyna, M. A., Kiarashi, Y., Elola, A., Oliveira, J., Renna, F., Gu, A., Perez-Alday, E. A., Sadr, N., Sharma, A., Mattos, S., Coimbra, M. T., Sameni, R., Rad, A. B., Clifford, G. D. (2022). Heart murmur detection from phonocardiogram recordings: The George B. Moody PhysioNet Challenge 2022. medRxiv, doi: 10.1101/2022.08.11.22278688
- [3] H.Tang, T.Li, T.Qiu, Noise and disturbance reduction for heart sounds in cycle frequency domain based on non linear time scaling, IEEE Trans. Biomed. Eng. 57(2)(2010)325–333
- [4] H.Tang, T. Li, Y.Park, T.Qiu, Separation of heart sound signal from noise in joint cycle frequency-time-frequency domains based on fuzzy detection, IEEE Trans. Biomed. Eng.57(10)(2010)2438–2447
- [5] Espressif (n.d.).ESP32-S3-WROOM-2-N32R8V. Digikey. Retrieved December 1, 2023, from www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-WROOM-2-N32R8V/15970964
- [6] Altium Designer (n.d.).Altium Designer. Retrieved December 1, 2023, from www.altium.com/
- [7] Digilent (n.d.). ZedBoard Zynq-7000 ARM/FPGA SoC Development Board. Retrieved November 30, 2023, from www.digilent.com/shop/zedboard-zynq-7000-arm-fpga-soc-development-board/
- [8] Github (n.d.). ELEC-594-MECE-Capstone-Project (Private).Retrieved December 1, 2023, from www.github.com/sbk7/ELEC-594-MECE-Capstone-Project.git
- [9] Espressif (n.d.).ESP32-S3-WROOM-2-N32R8V.Digikey.Retrieved December 1, 2023, from www.digikey.com/en/products/detail/espressif-systems/ESP32-S3-WROOM-2-N32R8V/15970964
- [10] Altium Designer (2020, April 5). Decoupling Capacitor and Bypass Placement Guidelines. Retrieved November 29, 2023, from www.resources.altium.com/p/bypass-and-decoupling-capacitor-placement-guidelines
- [11] E.S.(2023).ESP32-S3 Hardware Design Guidelines v3.3 (p. 23). Espressif. www.espressif.com/sites/default/files/documentation/esp32-hardware-design-guidelines-en.pdf
- [12] JLCPBCB (n.d.).PCB Manufacturing Assembly Capabilities. Retrieved November 30, 2023, from www.jlcpcb.com/capabilities/pcb-capabilities
- [13] Espressif (n.d.).Getting Started with VS Code IDE. Retrieved November 30, 2023, from www.docs.espressif.com/projects/esp-idf/en/v4.2.3/esp32/get-started/vscode-setup.html
- [14] "ZedBoard Audio in Vivado IP Integrator - the Zynq Book Tutorials - FPGAkey." Www.fpgakey.com, www.fpgakey.com/tutorial/section401. Accessed 1 Dec. 2023