DE-41 (MTS) Shahzaib, Rao, Muhammad

Doppler Ultrasound Fetal Heart Rate Instrument with AI Interface

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY RAWALPINDI

2023

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DE-41 MTS
PROJECT REPORT

DOPPLER ULTRASOUND FETAL HEART RATE INSTRUMENT WITH AI INTERFACE

Submitted to the Department of Mechatronics Engineering
in partial fulfillment of the requirements
for the degree of
Bachelor of Engineering
in
Mechatronics
2023

Submitted By:
Shahzaib Ali
Rao Abdul Sami
Muhammad Afzal

ACKNOWLEDGMENTS

First, we would like to thank Allah Almighty for helping us produce effective results. We thank our supervisor, Dr. Hamid Jabbar and Dr. Mohsin Tiwana, for guiding us with different research papers, and helping us in ordering the hardware for us. Without their support, we could never have managed to have the assistance of medical doctors which had been a crucial stone in the building. We thank Dr. Arshia Kanwal from Benazir Bhutto Hospital Rawalpindi for providing us with assistance in getting a data set of fetal heart signal from the hospital.

ABSTRACT

The aim of this project is to design and develop a fetal heart rate monitor with an artificial intelligence (AI) interface. The proposed device utilizes a non-invasive technique, piezo electric transducer to accurately measure the fetal heart rate, providing real-time data to medical professionals and parents-to-be.

This project presents a novel approach for fetal heart rate monitoring utilizing the piezoelectric material Doppler effect, combined with an artificial intelligence (AI) interface. By harnessing the Doppler effect, the system accurately detects and analyzes fetal heart signals, providing real-time information on the baby's well-being. The signal from the fetus is filtered using bandpass filters. The filtered signal is used to calculate beats per minute and is fed into a trained neural network to classify them as normal or abnormal signals.

The development of this fetal heart rate monitor with an AI interface is a significant contribution to the healthcare industry. It will provide a non-invasive, cost-effective, and accurate method of monitoring fetal heart rate, enabling early detection of potential health risks, and ultimately improving the quality of care for expectant mothers and their unborn children.

Table of Contents

List of Figures

List of Tables

List of Symbols

1. **FHR: Fetal Heart Rate Monitor**
2. **BPM: Beats Per Minute**
3. **ESI: Edge Impulse Studio**
4. **ML: Machine Learning**
5. **TinyML: Tiny Machine Learning**
6. **PCG: Phonocardiography**
7. **CTG: Cardiotocography**

Chapter 1 - INTRODUCTION

In 2015, the population distribution in Pakistan showed that 61.19% of the total population resided in villages. The country's total fertility rate, estimated in 2016, was 2.68 children born per woman. Unfortunately, healthcare facilities in Pakistan are known to be very poor. The country had the highest rate of stillbirths (fetal deaths after 22 weeks (about 5 months) of gestation) among 186 countries in 2015, making it the worst performer. Globally, the average rate of stillbirths is 18.4 deaths per 1,000 total births [1]. In Pakistan, the rate of stillbirths is significantly higher at 43.1 deaths per 1,000 total births, surpassing other countries like Nigeria, Chad, Niger, Guinea-Bissau, and Somalia, which are also among the bottom 10 performers [2].

Given the alarmingly high fetal mortality rate, there is a pressing need for a solution that is reliable even in the absence of a doctor, portable, and cost-effective. Regular and frequent fetal health examinations are essential, requiring an accessible and affordable solution.

In our Final Year Project, we have developed a prototype machine aimed at addressing this challenge. It is equipped with a classification system that can distinguish between normal and abnormal fetal heart signals, enabling it to function effectively even without a doctor present. This machine complements the primary diagnostic devices, such as CTG, by aiding in the decision-making process.

1.1 Overview

Fetal mortality is a global issue and an area where experts are working to reduce the mortality rate. Pakistan needs to look at and research this problem more does due to i insanely high mortality rate. As we worked and progressed way into the project, it was found that the current solution to the problem is the cardiotocography (CTG) machine as shown in Figure 1. CTG machine is not available in all parts of Pakistan, especially

in the rural areas. Secondly, it is an expensive machine for a country like Pakistan to import. Thirdly, it requires the availability of a doctor to interpret the results. The results of the study required a solution that would be:

- **Portable.**
- **Cost effective.**
- **Equipped with a decision support system so it would suggest a decision itself, which would complement the primary decision.**

Figure 1- A 10.4 inch Portable Touch Screen Fetal Monitor CTG Machine Maternal Pregnant Woman Baby Heart Rate Detector with printer

Chapter 2 - BACKGROUND AND LITERATURE REVIEW

2.1 The Phenomenon of doppler effect:

The Doppler effect is a phenomenon observed in waves, particularly in sound and light waves, where the frequency and wavelength of the waves change relative to an observer when the source of the waves is in motion [3]. It was named after the Austrian physicist Christian Doppler, who first described the effect in 1842.

The Doppler effect also finds application in medical science, particularly in obstetrics, where it is used for fetal Doppler monitoring. Fetal Doppler monitoring involves the use of a handheld device called a Doppler ultrasound, which emits high-frequency sound waves and detects their reflection off moving objects, such as blood cells.

When the blood cells in the fetal heart move towards the Doppler device, the frequency of the reflected sound waves appears higher, resulting in an audible higher-pitched sound [4]. Conversely, when the blood cells move away from the Doppler device, the frequency of the reflected sound waves appears lower, resulting in an audible lower-pitched sound. By analyzing the changes in frequency, healthcare professionals can monitor the fetal heart rate and assess the well-being of the fetus.

Fetal Doppler monitoring is commonly used during prenatal check-ups and can provide valuable information about the fetal heart rate, rhythm, and overall cardiovascular health. It is non-invasive and safe procedure that allows healthcare providers to gather important data about the developing fetus, ensuring timely interventions if any issues are detected [5].

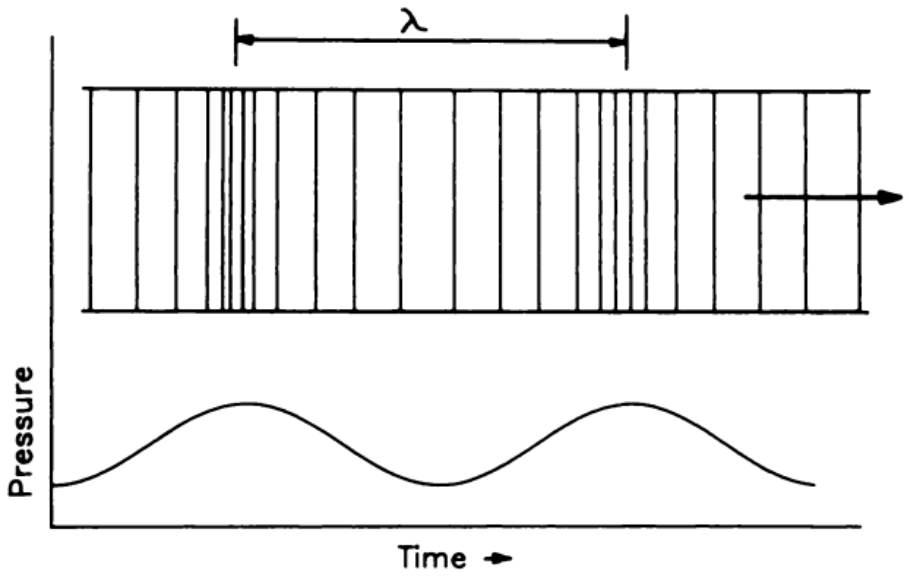


Figure 2 Time-varying changes in localized pressure due to passing wave. High pressure produces areas of compression and low pressure produces areas of rarefaction. [3]

However, it's worth noting that while Doppler ultrasound is a useful tool for fetal monitoring, it should be performed by trained healthcare professionals who can accurately interpret the results and provide appropriate medical advice based on the findings.

2.2 Fetal Heart Rate Significance:

The fetal heart rate (FHR) is a crucial indicator of the well-being and development of the fetus during pregnancy. It refers to the number of times the fetal heart beats per minute (bpm) and is typically monitored using a device called a Doppler or an electronic fetal monitor. The significance of fetal heart rate lies in its ability to provide valuable information about the fetus's oxygenation, overall health, and response to external stimuli. Here are a few key points regarding its importance:

2.2.1 Health Assessment:

FHR monitoring helps assess the fetal well-being and identify any potential issues. A normal FHR pattern indicates that the fetus is receiving adequate oxygen and nutrients, which is essential for its growth and development.

Changes in the FHR can indicate the fetus's response to stress or insufficient oxygen supply. For example, a decreased or irregular FHR may suggest fetal distress or hypoxia (low oxygen levels), potentially signaling a need for medical intervention [6].

2.2.2 Monitoring during Labor:

FHR monitoring is especially critical during labor to evaluate the fetus's response to contractions and ensure its safety. Abnormal FHR patterns during labor could indicate fetal distress and may prompt interventions like changing the mother's position, administering oxygen, or considering an emergency cesarean section if necessary.

2.2.3 Indicating Fetal Health Conditions:

Variations in the FHR can provide clues about certain fetal health conditions. For instance, an abnormally high FHR (tachycardia) may indicate maternal fever or fetal infection, while a consistently low FHR (bradycardia) could be associated with congenital fetal heart block or other abnormalities [[6].

2.2.4 Monitoring Growth and Development:

By assessing the FHR over time, healthcare providers can track the fetus's growth and development. Deviations from the expected FHR patterns can help identify potential issues, such as fetal growth restriction or anomalies, prompting further investigation or intervention.

It's important to note that FHR monitoring is typically performed by healthcare professionals trained in interpreting the data accurately. They consider numerous factors, including the gestational age, maternal health, and other clinical parameters, to make

informed decisions regarding the well-being of the fetus and the appropriate course of action.

2.3 Available devices for Fetal Heart Rate Monitoring:

2.3.1 Pinard Stethoscope (Fetoscope)

The Pinard stethoscope, also known as a fetoscope, is a traditional, low-cost, and non-electronic device used to auscultate and monitor the fetal heart rate (FHR) during pregnancy[7]. Named after French obstetrician Adolphe Pinard, who introduced it in the late 19th century, the Pinard stethoscope has remained a valuable tool in obstetric care, particularly in resource-limited settings. This literature review aims to explore the historical background, design, clinical applications, advantages, limitations, and current use of the Pinard stethoscope.

The Pinard stethoscope was first introduced in the late 1800s as a replacement for the more invasive and risky procedures of that time. Adolphe Pinard's innovation provided a simple, direct, and safe method to listen to the fetal heart sounds. Over the years, it gained popularity due to its affordability, ease of use, and non-invasive nature.

Figure 3 Pinards stethoscope [8]

The Pinard stethoscope is a conical, wooden or metal device with a trumpet-shaped end that amplifies sound. It is held against the mother's abdomen, usually over the fetal back, to detect and listen to the fetal heart sounds [7]. The practitioner places one end of the

stethoscope against the abdomen while the other end is held to their ear, allowing them to hear the amplified sound of the fetal heartbeat.

2.3.1.1 Fetal Heart Rate Monitoring:

The primary purpose of the Pinard stethoscope is to monitor the FHR during prenatal care and labor. It allows healthcare providers to assess fetal well-being, detect abnormalities, and identify signs of distress.

2.3.1.2 Localization of Fetal Position:

The Pinard stethoscope can also help in determining the fetal position within the uterus. By listening to the heart sounds at various locations, practitioners can estimate the position of the back or the location of the maximum intensity of the heartbeat [9].

2.3.1.3 Limitations:

Interpreting the fetal heart sounds using the Pinard stethoscope requires a trained healthcare provider with a keen ear and experience. The accuracy of FHR assessment can be influenced by individual interpretation and expertise.

The Pinard stethoscope provides qualitative information about the presence and quality of fetal heart sounds but does not provide precise quantitative measurements, such as beats per minute.

2.3.2 Fetal heart rate monitoring:

A microphone fetal monitor fills the same niche as a Doppler monitor. The major difference between a microphone-based monitor and a Doppler monitor lies in the method by which the signal is collected. Where a Doppler monitor actively probes for a signal using ultrasound techniques, a microphone-based monitor passively receives a signal through electro-acoustic means [7], [9], [10]. This offers a major advantage in terms of cost of manufacture – a microphone-based probe is significantly less expensive than one reliant on ultrasound.

2.4 Cardiotocography:

2.4.1 Fetal Heart Rate Monitoring:

Cardiotocography (CTG) machines encompass a wide range of skin tones in their devices, along with other favorable features valued by clinicians. The upper line of the CTG graph displays the fetal heart rate over time, with the x-axis representing elapsed time and the y-axis representing the instantaneous fetal heart rate [11]. The lower line illustrates uterine contractions, enabling clinicians to identify accelerations and decelerations. This functionality allows for the diagnosis of life-threatening conditions like hypoxia and acidosis (Lawrence, 2012). However, these additional features come at the expense of affordability and portability. CTG machines often exceed $500 in cost, with newer models reaching over $5,000 when purchased from medical device suppliers. Moreover, their size, which is typically comparable to a desktop computer tower unit, makes them impractical for long-distance transportation, hindering clinicians making house calls in developing countries. Additionally, CTG machines require a power source and cannot be used in clinics lacking electricity.

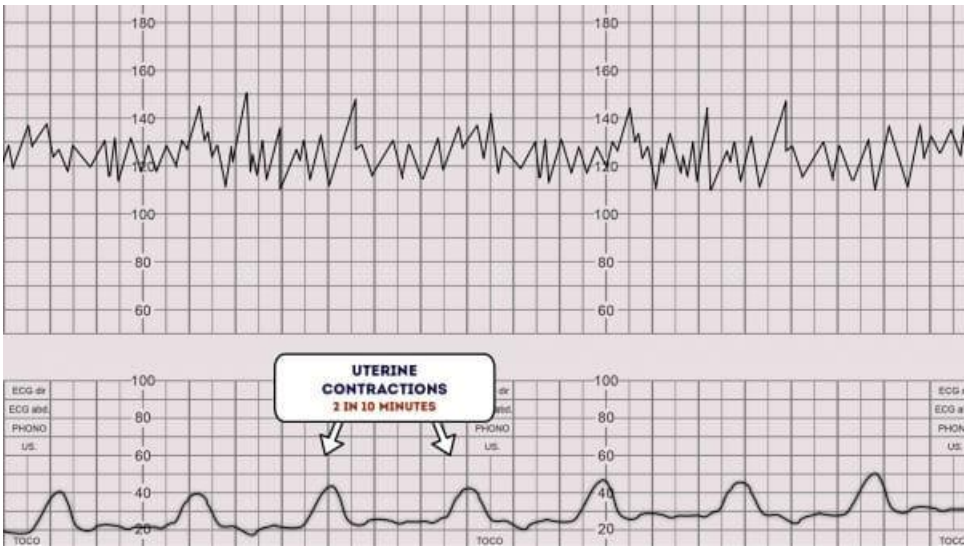


Figure 4 Output of CTG Machine

On the other hand, an electrocardiogram (ECG) can detect the electrical potential generated by cardiac muscle contractions. It measures the heart rate and identifies heart damage by collecting and analyzing electrical signals through electrodes. In the case of adults, electrodes are placed on the skin above the heart. While invasive electrode ECGs have been used to measure the fetal heart rate by passing an electrode through the cervix to the fetus' scalp, noninvasive measurement using electrodes on the abdomen is more challenging. The fetal ECG signal has a significantly lower signal-to-noise ratio compared to the maternal ECG[5]. The presence of tissue between the fetal heart and the abdominal electrode, the small size of the fetal heart, and interference from electrical brain activity and maternal muscle contractions contribute to noise in the fetal heart signal. It is difficult to isolate the fetal heart signal from other bio signals based on time, space, frequency, or amplitude.

2.5 Piezoelectric Materials

2.5.1 Definition of piezoelectricity

An exceptional property of piezoelectric materials is their capacity to transform mechanical energy into electrical energy and vice versa. The atoms are arranged in these materials' natural crystal structures, which causes the phenomenon known as the piezoelectric effect.

2.5.2 Types of piezoelectric materials

There are several varieties of piezoelectric materials, each with special qualities and traits. Ceramics, polymers, and single crystals are the three primary categories of piezoelectric materials.

2.5.3 PZT-4 Properties

PZT-4 is made up of lead zirconate ($PbZrO_3$) and lead titanate ($PbTiO_3$) in a solid solution. Typically, the composition is represented by a chemical formula, such $(1-x)Pb(Zr_{0.52}Ti_{0.48})O_3 - xPb(Zn_{1/3}Nb_{2/3})O_3$, where x is the amount of $Pb(Zn_{1/3}Nb_{2/3})O_3$ that is added to the composition to increase specific features. The

effectiveness of converting energy between mechanical and electrical forms depends on the piezoelectric coefficients, which are present in high concentrations in PZT-4[12], [13].

PZT-4 can withstand mechanical stress, vibrations, and outside forces thanks to its good mechanical strength. This quality is essential for applications where the material must survive challenging conditions or repeated mechanical stresses. PZT-4 has a low dielectric loss, which means there is not much energy lost when an electric field is applied to the material. In situations where the substance is utilized as a capacitor or energy transducer, low dielectric loss boosts efficiency and improves performance[14].

Material Properties	Elastic Constants $\times 10^{10}$ (N/m^2)		Piezoelectric Constants (C/m^2)		Dielectric Constants $\times 10^{-9}$ (C/Vm)	
PZT – 4	c_{11}	13.9	e_{31}	-6.98	p_{11}	6.00
	c_{12}	7.78	e_{33}	13.84	p_{33}	5.47
	c_{13}	7.43	e_{15}	13.44		
	c_{33}	11.3				
	c_{44}	2.56				
	c_{55}	0.775				

Figure 5 Properties of PZT-4

2.5.4 Piezoelectric properties of PZT-4

PZT-4 is a ceramic material made of lead zirconate titanate (PZT), which has exceptional piezoelectric characteristics. Its use in sensors, actuators, transducers, and other devices that depend on the conversion of mechanical to electrical energy requires these characteristics. The following are PZT-4's main piezoelectric characteristics:

Piezoelectric Coefficient (d33): The capacity of a material to transform mechanical stress or strain into an electrical charge is indicated by the piezoelectric coefficient (d33). Due to its high d33 coefficient and effective translation of mechanical energy into electrical impulses, PZT-4 is extremely sensitive to mechanical stimuli [13], [15].

Piezoelectric Voltage Coefficient (g33): The piezoelectric voltage coefficient (g33), which characterizes a material's capacity to produce voltage when subjected to mechanical stress, is a measurement of this capacity. Due to its high g33 coefficient, PZT-4 can produce electric signals in reaction to outside mechanical forces.

Dielectric Constant: PZT-4's dielectric constant explains how much electrical energy it can store when an electric field is applied. Increased energy storage capacity is made possible by a greater dielectric constant. PZT-4 generally has a reasonably high dielectric constant, allowing for effective polarization and charge storage.

Mechanical Quality Factor (Qm): This is a measure of mechanical quality. The energy lost during mechanical vibrations in a material is represented by Qm. The high Qm of PZT-4 is well recognized for its low energy dissipation and great mechanical energy storage efficiency. The capacity of the material to sustain resonance and generate powerful, sustained vibrations depends on the Qm value.

PZT-4's Curie temperature (Tc): It is the temperature at which the substance goes through a phase change and loses its piezoelectric characteristics. PZT-4 can function at high temperatures without significantly degrading its piezoelectric ability because of its comparatively high Curie temperature [15].

2.6 Piezoelectric Disc Manufacturing Process

Material Preparation: Preparation of the PZT-4 material is the first step in the procedure. This entails precisely weighing out and combining the necessary amounts of lead zirconate and lead titanate powders, as well as any extra dopants or additives. To create a uniform composition, the powders are generally ground and combined.

Forming: The PZT-4 material must next be shaped into a disc as required. There are other ways to make an object, such as tape casting, dry pressing, or isostatic pressing. PZT-4 slurry is placed on a level surface during tape casting, dried, and then cut into the required form. PZT-4 powder is compressed into a die using high pressure in a process

known as dry pressing. The PZT-4 powder is subjected to consistent pressure during isostatic pressing from all sides[12], [14].

Sintering: After being formed, the PZT-4 disc goes through the sintering process. To encourage densification and particle bonding, sintering requires heating the produced material at high temperatures. To get the necessary density and electrical characteristics for the PZT-4 disc, the sintering temperature and duration are carefully regulated.

Poling: Poling plays a key role in increasing the PZT-4 disc's piezoelectric characteristics. It entails exposing the sintered disc to a strong electric field at a high temperature. By aligning the electric dipoles within the material, this method produces the necessary piezoelectric effect. A high-voltage power supply or a poling machine may be necessary for the poling procedure.

Electrode Deposition: On the surfaces of the PZT-4 disc, electrodes are deposited to make electrical connections easier. Typically, silver or platinum metal electrodes are screen-printed or spattered onto the surfaces. The active region of the piezoelectric disc is defined by the patterning of the electrodes, which guarantees appropriate electrical contact.

Quality Control and Testing: Quality control procedures are used to guarantee the PZT-4 disc's dimensional correctness, electrical characteristics, and general performance throughout the production process. The completed product's quality and functioning are verified using tests, including dimensions measurements, impedance analysis, and electrical characterization.

Assembly and Integration: Lastly, depending on the PZT-4 piezoelectric disc's intended use, it may be incorporated into a larger system or device. Incorporating the proper signal conditioning or control circuitry, putting the disc into a housing, attaching electrical leads to the electrodes, and mounting the disc into a housing are some possible steps in this process[15].

2.7 Key findings from previous research on PZT-4-disc manufacturing

Sintering Optimization: Researchers have explored various sintering parameters, such as temperature, duration, and atmosphere, to optimize the densification and electrical properties of PZT-4 discs. It has been found that sintering at higher temperatures and longer durations can lead to improved density, reduced porosity, and enhanced piezoelectric properties.

Poling Effects: Poling, the process of aligning the electric dipoles in the PZT-4 material, has been investigated extensively. Researchers have focused on understanding the impact of poling conditions, such as electric field strength, duration, and temperature, on the resulting piezoelectric properties. Studies have shown that appropriate poling parameters can significantly enhance the piezoelectric coefficients and improve the performance of PZT-4 discs[15].

Dimensional Control: Precise dimensional control of PZT-4 discs has been highlighted as a critical factor in achieving desired performance. Researchers have investigated fabrication techniques, such as tape casting, dry pressing, and isostatic pressing, to control the dimensions and thickness of the discs. Studies have emphasized the importance of accurate dimensions in achieving the desired resonance frequency, sensitivity, and integration within the overall system.

Defects and Quality Control: Researchers have studied the influence of defects, such as porosity, cracks, and impurities, on the electrical and mechanical properties of PZT-4 discs. Techniques for quality control, such as impedance analysis, dielectric measurements, and mechanical testing, have been explored to assess the integrity and reliability of the manufactured discs. Efforts have been made to minimize defects and ensure consistent and high-quality production[12], [13], [15].

Chapter 3 - METHEDOLOGY

3.1 Data Collection

Fetus Doppler data is required for designing digital filters and training of machine learning models. Here are the few days through which we collected data as per our requirement.

3.2 Online Dataset

Thanks to Shiraz university for providing a large data set of Fetus heart sounds. The Shiraz University (SU) fetal heart sounds database (SUFHSDB) contains fetal and maternal phonocardiogram (PCG) recordings from 109 pregnant women in single and twin pregnancies. The data set is available in wave file format. This format can be directly used in Lt. spice and proteus software. This data is somewhat similar to the signal provided by our 3-megahertz probe, but it was having a lot of noise as compared to our probe signal. But this data can be used to design filters and amplifiers. This filtered data can be further used to train ML models.

To use this data in Lt. spice you need to use a DC voltage source with value written in the following format:

wavefile = "(your directory) \(filename).wav"

Thus, your wave file will now be able to be directly accessible Lt spice software. Here is how I use this:

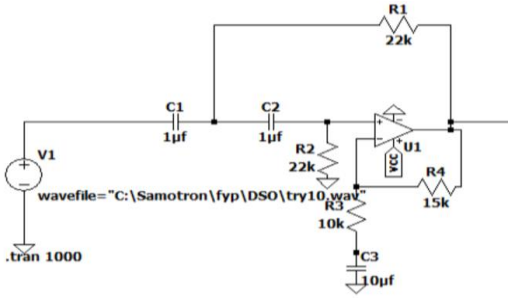


Figure 6 Wave file imported in Lt spice

Similarly, to import this data in proteus software use a DC generator probe double click on it and select the “audio” under the analog types of panels. From there you can browse and select any wave file. Thus, your signal will now be loaded in proteus software.

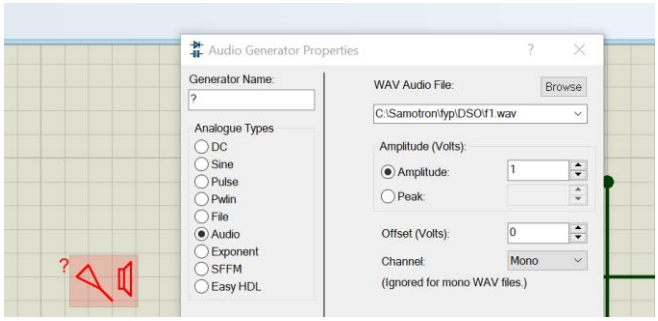


Figure 7 Wavefile Import in Proteus software.

3.2.1 Data from the 3Mhz Probe

We used our probe and connected its output to the lab oscilloscope model UTD2052CL. This oscilloscope can store waveforms displayed on its screen. We connected our USB to this oscilloscope and copied the stored waveforms from oscilloscope to our USB storage device. This oscilloscope stores data in three formats. Dat file, SAV file, and Bitmap file format. The Dat file and SAV file are coded in hex format and cannot be directly opened in any software like MATLAB, proteus, and Lt. Spice. To decode this file, we used a hex file opener software named **HxD**. This software decodes the hex file and presents the data in a tabular form. We copied this data into an Excel file and then imported it into MATLAB. From there you can either process it or convert it into a wave file format. There is also one oscilloscope in Micro-Nano lab of Mechatronics department that directly stores the data in a text file or Excel file atere is also a PC oscilloscope by Hantek available in our department. This oscilloscope is made to be used with laptops and computers.

Another way we used it was that we read the values from the probe in ADC of our Pico RP2040 microcontroller. And we saved these values and imported them into Matlab.

3.3 Hardware Design

The hardware comprises of:

a. 30mm (about 1.18 in) diameter piezoelectric disk

b. PCB for filtering the data and sending it to Microcontroller and a display screen.

3.3.1 3.3 The 30mm PiezoElectric Disk Selection

The choice of PZT-4 as the piezoelectric material for a specific application is typically based on several factors. Here are some rationales for choosing PZT-4:

- **High Piezoelectric Coefficients:**

  PZT-4 exhibits high piezoelectric coefficients, such as the d33 and g33 coefficients, which indicate its high sensitivity to mechanical stimuli and efficient conversion of mechanical energy to electrical signals. This makes PZT-4 suitable for applications where precise sensing or actuation is required.

- **Excellent Mechanical Strength:**

  PZT-4 is known for its good mechanical strength, enabling it to withstand mechanical stress, vibrations, and external forces. This characteristic is crucial for applications where the material needs to operate in demanding environments or undergo repetitive mechanical loads.

- **Low Dielectric Loss:**

  PZT-4 exhibits low dielectric loss, indicating minimal energy dissipation in the material when an electric field is applied. Low dielectric loss leads to higher efficiency and improved performance in applications where the material is used as a capacitor or energy transducer.

- **High Curie Temperature:**

  PZT-4 has a high Curie temperature, allowing it to operate at elevated temperatures without significant degradation in its piezoelectric properties. This characteristic is advantageous for applications that involve high-temperature environments or require stable performance across a wide temperature range.

- **Wide Frequency Response:**

  PZT-4 offers a wide frequency response, making it suitable for applications requiring operation at various frequencies. This characteristic is beneficial for applications such as ultrasonic imaging, where a broad range of frequencies needs to be detected or generated.

- **Commercial Availability:**

  PZT-4 is commercially available and widely used in various industries. Its availability, along with established manufacturing processes and suppliers, makes it a convenient choice for many applications.

d. Disc Design

The disc is designed using PZT-4 material.

Physical and Piezoelectric Properties of T&P Materials

Material:	T&P-4	T&P-45
Relative Dielectric Constant		
K^T	1250	1350
Dielectric Dissipation Factor (Dielectric Loss (%)*		
Tan δ	0.4	0.35
Curie Point (°C)**		
T_C	325	320
Electromechanical Coupling Factor (%)		
K_P	0.59	0.60
K_{33}	0.70	0.68
K_{31}	0.35	0.33
K_{15}	0.70	0.67
Piezoelectric Charge Constant (10^{-12} C/N or 10^{-12} m/V)		
d_{33}	300	300
$-d_{31}$	125	109
d_{15}	480	450
Piezoelectric Voltage Constant (10^{-3} Vm/N or 10^{-3} m²/C)		
g_{33}	26.5	25.5
$-g_{31}$	11	10.5
g_{15}	38	35
Young's Modulus (10^{10} N/m²)		
Y^E_{11}	8	7.6
Y^E_{33}	6.8	6.3
Frequency Constants (Hz.m or m/s)		
N_L (longitu	1524	1700
N_T (thickne	2100	2005
N_P (planar)	2130	2055
Density (g/cm³)		
ρ	7.6	7.6
Mechanical Quality Factor		
Q_m	600	1400

All values nominal; measurements made 24 hours after polarization.
Maximum voltage:
5-7 VAC /mil for T&P-5 VDC ~2X.
9-11 VAC /mil for T&P-4, T&P-8 VDC ~2X.
*At 1 kHz, low field.
**Maximum operating temperature = Curie point/2.

Figure 8 Datasheet of PZT-4

3.3.2 Dimensions of the Piezoelectric Disc (Diameter and thickness)

Following are the dimensions for the Piezoelectric Disc:

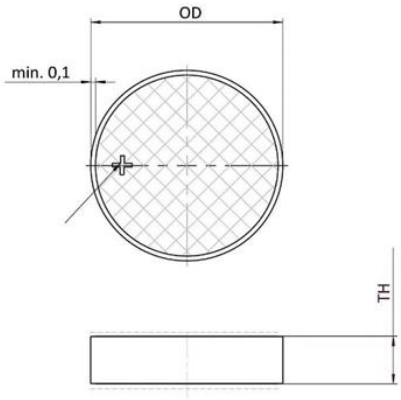


Figure 9 Piezo disc dimensions

TH=Thickness=1 mm, OD= Outer Diameter=30mm

3.3.3 Resonance Frequency in Radial Disc Mode

The resonance frequency of a piezoelectric disc in the radial disc mode can be calculated using the following formula:

$N_1=F_r$ D (Hz.m) Radial Mode Disc

N_1=1524, D=30mm= 0.03

F_r =1524/0.03=50800 Hz = 0.0508 MHz

4.3.2 Resonance Frequency in Thickness Disc Mode

The resonance frequency of a piezoelectric disc in the thickness disc mode can be calculated using the following formula:

$N_4=F_r$ h (Hz.m) Thickness Mode Disc, Plate

N_4=2100, h= 1mm = 0.001m

F_r = 2100/0.001 = 2100000 Hz = 2.1 MHz

3.4 COMSOL Analysis

3.4.1 Meshing

Meshing is a crucial step in the finite element analysis (FEA) process, including when using COMSOL Multiphysics software. Meshing involves dividing the computational domain into a finite number of smaller subdomains, called elements or cells, to approximate the geometry and physics of the problem being solved. COMSOL provides various meshing techniques to ensure accurate and efficient simulations.

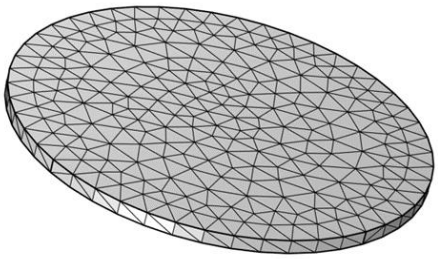


Figure 10 Meshing of PZT-4

3.4.2 Eigen Frequency Analysis

Eigenfrequency analysis, also known as modal analysis or natural frequency analysis, is a numerical technique used to determine the natural frequencies and corresponding mode shapes of a structure or system. The eigenvalues represent the natural frequencies at which the system can vibrate, while the eigenvectors describe the corresponding patterns of motion or deformation. The analysis of the disc is shown as follow:

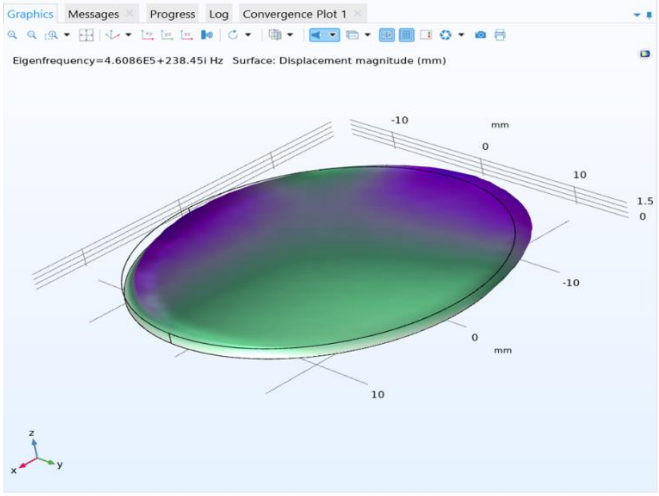
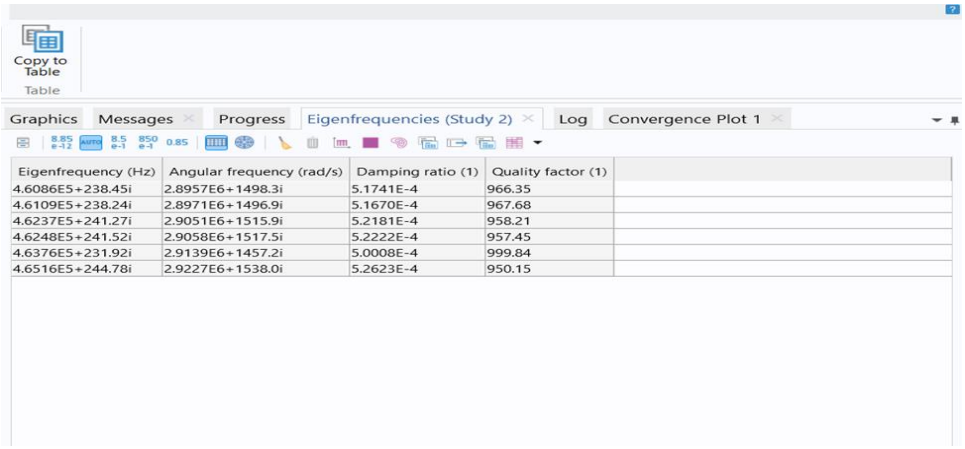


Figure 11 Surface displacement



Eigenfrequency (Hz)	Angular frequency (rad/s)	Damping ratio (1)	Quality factor (1)
4.6086E5+238.45i	2.8957E6+1498.3i	5.1741E-4	966.35
4.6109E5+238.24i	2.8971E6+1496.9i	5.1670E-4	967.68
4.6237E5+241.27i	2.9051E6+1515.9i	5.2181E-4	958.21
4.6248E5+241.52i	2.9058E6+1517.5i	5.2222E-4	957.45
4.6376E5+231.92i	2.9139E6+1457.2i	5.0008E-4	999.84
4.6516E5+244.78i	2.9227E6+1538.0i	5.2623E-4	950.15

Figure 12 Eigen Frequency(Hz)

3.4.3 Electric Potential Analysis

Electric potential analysis, also known as electrostatic analysis, is a numerical technique used to study the distribution of electric potential in a system or structure. The electric potential represents the work required to move a unit positive charge from a reference point to a specific location in an electric field. Following is the analysis for the disc:

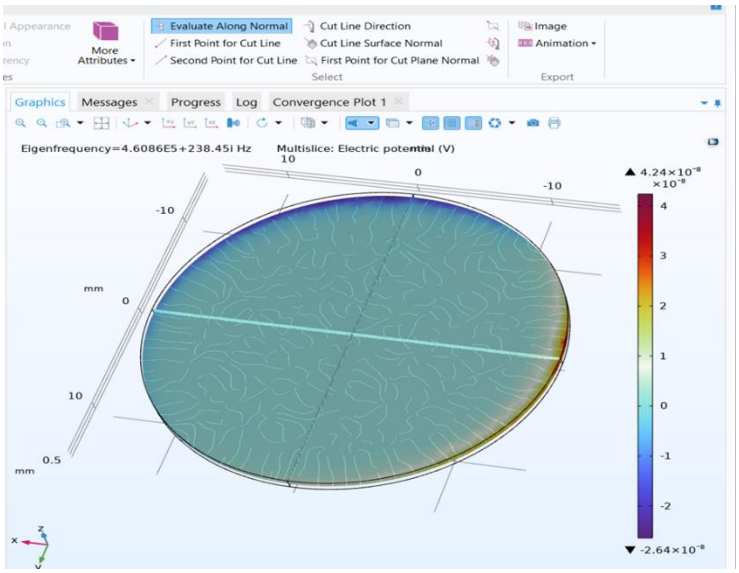


Figure 13 Electric Potential

3.5 Printed Circuit Board (PCB)

The PCB is made up of 3 sections:

1. Microcontroller Section
2. Analog Filtering Section
3. LCD Display Section

All these 3 sections are inter-related with each other. This is shown in figure below:

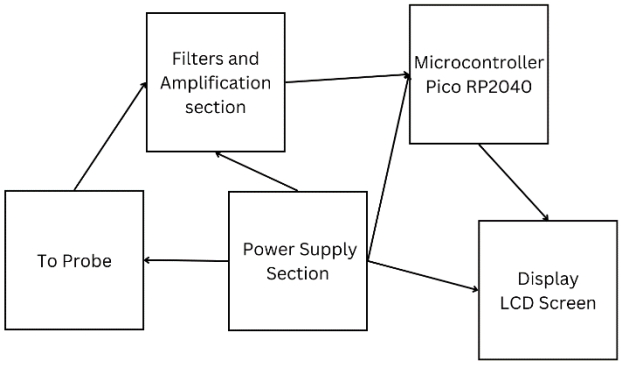


Figure 14 Flow Diagram of PCB

3.5.1 Power Supply Section

Our filters, microcontroller and lcd require 5v constant input. For this purpose, we have used LM7805 linear voltage regulator IC. For supply input, we have used 2x 3.7v Ni-Cad Cells in series which becomes 7.4v in total. The output of 5v from voltage regulator Ic is then further used in the circuit and is labeled as "**Vcc**".

This Vcc is used to power up the op-amps used in filtering section, microcontroller, lcd screen and the 3Mhz Probe. The schematic of Lm7805 Ic is as follows:

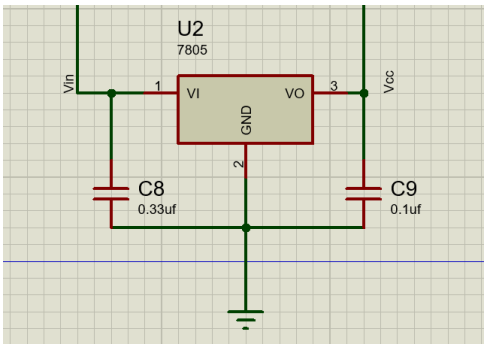


Figure 15 Schematic of LM7805

3.5.2 Design of Digital Filters and Amplifier

The scope of the filter design is to detect the fetus's heartbeat pks from the noisy Doppler signal. The 3Mhz provides the demodulated Doppler shifted signal. This signal varies from 0-300hz including noise. However according to [16]–[18] frequency of fetus heart lies in the range from 10hz-200hz. Similarly, the amplitude of this Doppler signal provided by the probe varies from 100mv to 600mv. Therefore, a bandpass filter was required to remove the noise and unwanted signals and to amplify the signal that can be sent to the ADC of the microcontroller. We designed and tested all the filters and amplifiers in Lt. spice software and made the schematic and PCB layout in Proteus software.

First, we created a 2nd order Sallen-Key High-pass filter with a corner frequency of 7hz and a gain of 4db. This filter removes the DC noise and other low frequency noise. Our calculation or as follow:

$$cutoff\ frequency = fc = \mathbf{7}hz$$

$$So\ having\ 1uf\ capacitor\ , required\ resistor\ value\ will\ be\ R = \frac{1}{2pi * fc * C}$$

$$= \frac{1}{2 * pi * \mathbf{7} * 1x10^{-6}} = \ 22.7\ \text{K} = 22\text{k}$$

$$R = 22\text{k}\ ohms$$

$$and\ Gain = A = \frac{Vout}{Vin} = 1 + \frac{R3}{(R4^2 + Xc^2)^{\frac{1}{2}}} = 1.7$$

So, our high pass filter looks like this:

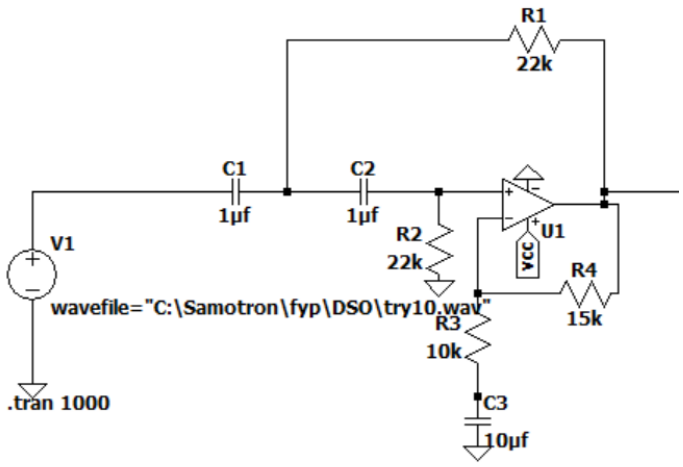


Figure 16 High pass filter

Then we fed the output of this filter into a 2nd order Sallen-Key Low-pass filter with a corner frequency of 160hz and a gain of 4db. This filter removes maternal heartbeat noise and other noises.

Our calculation or as follow:

$$cutoff\ frequency = fc = 160hz$$

$$So\ having\ 1uf\ capacitor\ , required\ resistor\ value\ will\ be\ R = \frac{1}{2pi * fc * C}$$

$$= \frac{1}{2 * pi * 160 * 1x10^{-6}} = \ 1\text{k}$$

$$R = 1\text{k}\ ohms$$

$$and\ Gain = A = \frac{Vout}{Vin} = 1 + \frac{R3}{(R4^2 + Xc^2)^{\frac{1}{2}}} = 1.7$$

So, our Low pass filter looks like this:

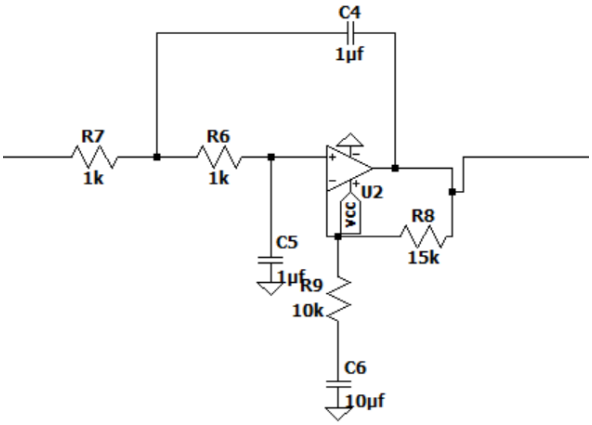


Figure 17 Low Pass filter

Thus, we made a 4th order bandpass filter with a total gain of 9.2db.

However, the amplitude of resulting signal from the bandpass filter Was not enough to be send to the ADC of microcontroller.

Thus, a final amplification was required. So, we designed a linear noninverting amplifier with the gain of 4.6. We fed the output of bandpass filter into this amplifier. Now our signal was in the range of 2 to 5 volts, and we were able to pass the signal to the ADC. Calculation for the linear amplifier are:

$$Gain = A = \frac{Vout}{Vin} = 1 + \frac{R10}{(R11^2 + Xc^2)^{\frac{1}{2}}} = 1.7$$

So our amplifier looks like this:

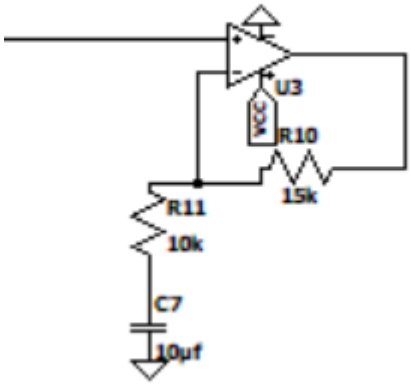


Figure 18 Final amplifier

LM324 Ic is used for filtering and amplification. It consists of 4 op-amps thus suitable for our case.

The schematic diagram of circuit is as follow:

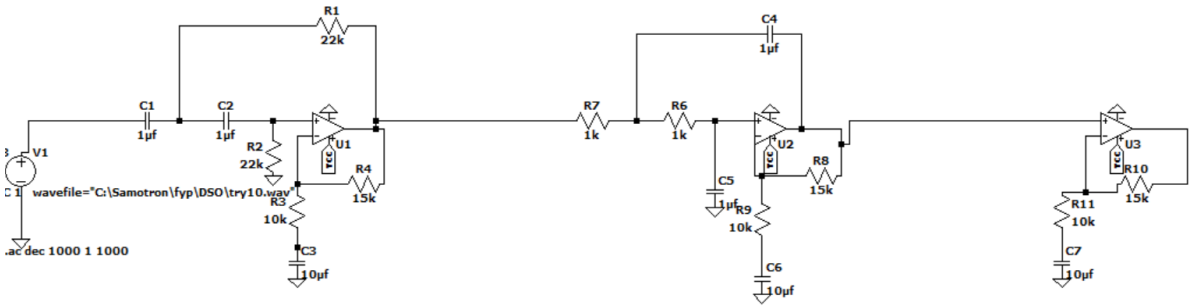


Figure 19 Schematic diagram of Filter and Amplifier

This circuit successfully removes the noise and unwanted signals and detects the fetus heartbeat peaks in the signal.

3.5.3 Microcontroller and Lcd Display

The microcontroller we have used is Raspberry Pi RP2040 that comes with a Pico Development Kit. We have used this controller since it is cheaper yet has microprocessor ability to run ML programs. This controller is also powered up by the Vcc coming from the supply section.

Also, a 16x2 Lcd is used to display the final beats per minute of the fetus and Normal or Abnormal Fetus heart beats.

3.6 Fetal Heart Rate Calculation and Fetal Signal Analysis:

Fetal heart monitoring is important for health monitoring of fetal inside mother womb. Thousands of doctors around the world are concerned to supply best health services to pregnant women for the better development of fetus. Many lives can be saved if irregularities in fetus heart rate are detected promptly. An extremely critical aspect of medical aspect is the automatic classification of fetal heart disease.[19]

i. Basic flow of fetal heart rate monitoring

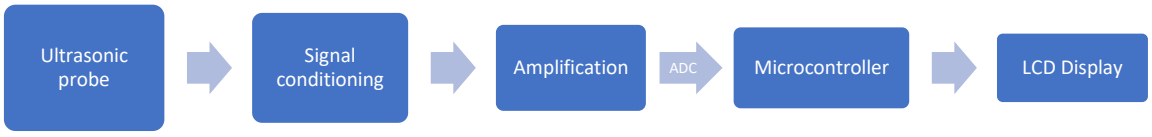


Figure 20 Basic flow of fetal heart rate monitoring

These are the 5 steps in which we categorized the project for the development of Fetal heart rate monitor with AI interface. In the coming sections the discussion will be about the microcontroller part and the displaying of the results on the LCD display. The main two things which are done for the microcontroller part are as follows:

a. Algorithm for calculating beats per minute.

b. Deployment of Machine learning model on RP2040 microcontroller

3.6.1 Beats per Minute Calculations:

The beats per minute (BPM) are the number of beats in 60 seconds. This project devised an algorithm to calculate beats per minute from the fetus heart signals.

The mathematical formula to calculate the BPM is as follows:

$$BPM = \frac{1}{peak\ to\ peak\ time} \times 60{,}000$$

The reason to multiply the 60,000 is because the peak-to-peak time is in milli seconds.

Here, we estimate the heart rate of fetal signal by using doppler shift. The doppler shift from the fetal signal usually varies from 50-200 Hz [[20].

Now in this project we have used already build probe which gives the data of doppler shift. This doppler shift signal is then filtered out using band pass filters as described in previous chapters.

After the process of filtering, the signal is fed into the 12-bit ADC of Raspberry Pi Pico.

3.6.2 About raspberry pi Pico and its ADC:

The Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation. It is based on the RP2040 microcontroller chip, which is designed by Raspberry Pi and features a dual-core Arm Cortex-M0+ processor.

The Pico microcontroller board is compact and affordable, making it suitable for a wide range of embedded systems and IoT projects. It offers a variety of features, including

GPIO pins, PWM (Pulse Width Modulation) channels, UART, SPI, I2C interfaces, and more.

One important feature of the Raspberry Pi Pico is its built-in Analog-to-Digital Converter (ADC). The ADC allows the Pico to measure analog voltages and convert them into digital values that can be processed by the microcontroller. This capability is useful for applications that require reading sensors or buying data from the physical world.

Pico's ADC has a resolution of 12 bits, which means it can represent analog voltages with high precision. It supports input voltages in the range of 0 to 3.3 volts. The ADC on the Pico can be accessed through the GPIO pins, allowing you to connect analog sensors or other devices to measure analog signals.

The pinout of Raspberry Pi Pico is:

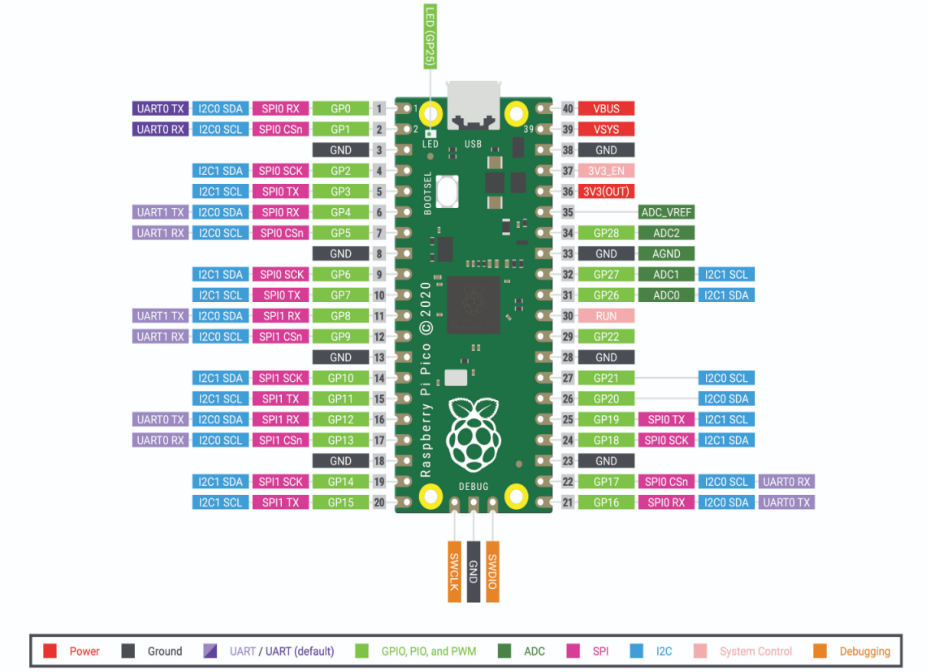


Figure 21 Pinout of Raspberry Pi Pico by official website of Raspberry Pi

The data sheet of Raspberry Pi Pico and its C/C++ SDK can be found on the official website of Raspberry PI.

To use the ADC on the Raspberry Pi Pico, you would typically write code that configures the ADC, selects the desired input pin, and reads the digital value from the ADC. Pico's software development tools, such as the Micro Python programming language or the C/C++ SDK, provide libraries and APIs to access and utilize the ADC functionality.

In summary, the Raspberry Pi Pico microcontroller board is a versatile and affordable embedded system that offers various features for building IoT projects. Its built-in Analog-to-Digital Converter (ADC) allows for the measurement of analog voltages, making it suitable for applications that involve sensor readings or analog data acquisition.

Now the filtered signal is fed into the ADC to which were recorded and lodged in the laptop by using the following C program:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"
#include "hardware/adc.h"
int main() {
stdio_init_all();
    adc_init();
    // Make sure GPIO is high-impedance, no pullups etc
    adc_gpio_init(26);
    // Select ADC input 0 (GPIO26)
    adc_select_input(0);
```

```
    while (1) {
        // 12-bit conversion, assume max value == ADC_VREF == 3.3 V
        const float conversion_factor = 3.3f / (1 << 12);
        uint16_t result = adc_read();
        printf("Raw value: 0x%03x, voltage: %f V\n", result, result * conversion_factor);
        sleep_ms(1);
    }
}
```

The data logging is done by using the default feature of Serial monitor of VS code.

Here each sample is recorded after every 1ms. Hence, the data is sampled at the rate of 1kHz.

3.6.3 Data acquisition

For the data acquisition we went to Benazir Bhutto Hospital, Murree Road, Rawalpindi. Here we recorded the data on our laptop by using the probe, filter circuit, and above code.

Now the next task was applying the algorithm to find the peaks in the data acquired from the hospital.

3.6.4 Development of Algorithm to find Beats per minute:

A threshold was set to detect the 1st peak and then the time is recorded until the 2nd peak is recorded. Normally the time between the peaks was around .4 seconds. Which gives the value of 150 Beats per minute.

3.6.5 Flow chart for calculating peak to peak time:

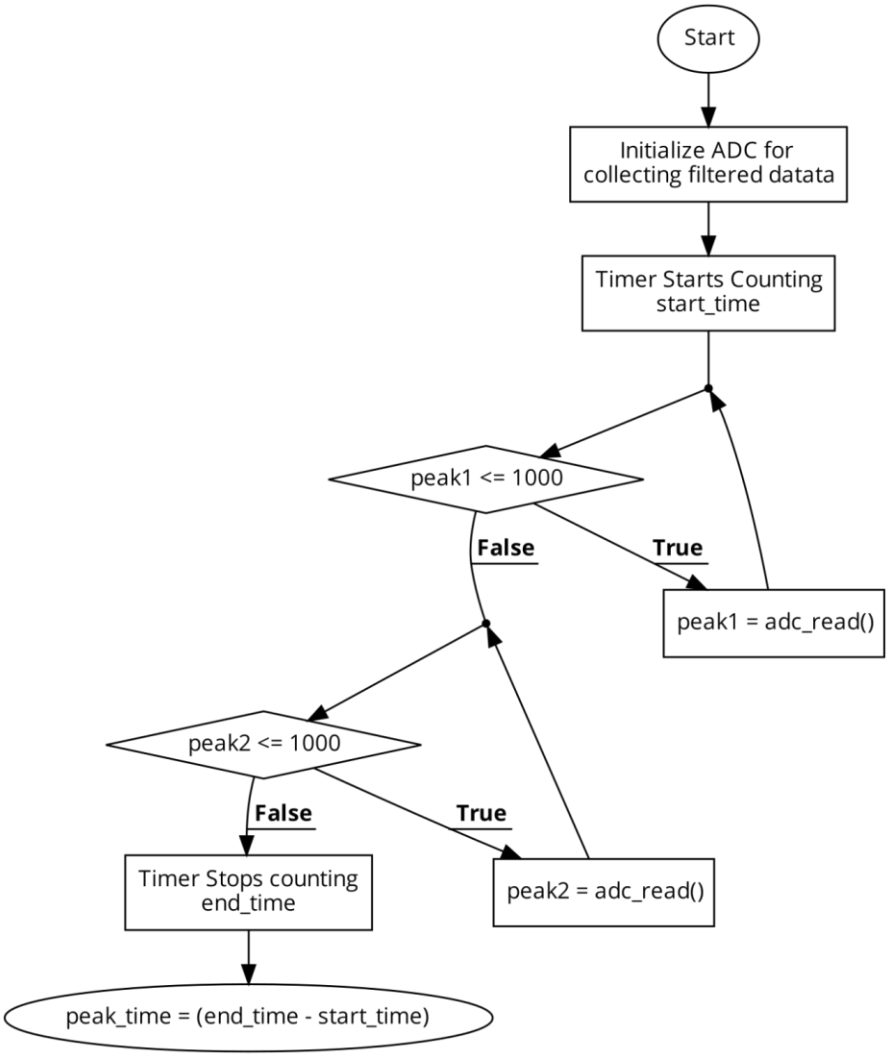


Figure 22 Flow chart for calculation of peak to peak time

The C program is as follows:

```
uint32_t start_time, end_time,peak_time;

float bpm;

start_time = millis();

 while (peak1 <= 1000)
```

```
{  peak1 = adc_read();
}
 while (peak2 <= 1000)
    {
    peak2 = adc_read();
    }
 end_time = millis();
    peak_time = (end_time - start_time);
    printf("Peak time %.2f\n", peak_time);
     bpm = 60.00 / peak_time;
 while (peak2 <= 1000)
    {
    peak2 = adc_read();
    }
 end_time = millis();
    peak_time = (end_time - start_time);
    printf("Peak time %.2f\n", peak_time)
     bpm = 60000.00
```

.

According to the research, the fetal heart rate in one minute is around 110 to 160 bpm, which classifies our algorithm as true.

3.7 Tiny Machine learning Model using Edge Impulse Studio:

To do the analysis of fetal heart signal the Embedded Machine learning is used. Embedded machine learning is also known as Tiny Machine Learning (TinyML).

Embedded Machine Learning or Tiny ML is On-device machine learning applications in the single mW and below.

The method for the classification of fetus heart signal used is ML classification. The approach used is to collect the data using the microcontroller and deploy the ML model on the microcontroller.

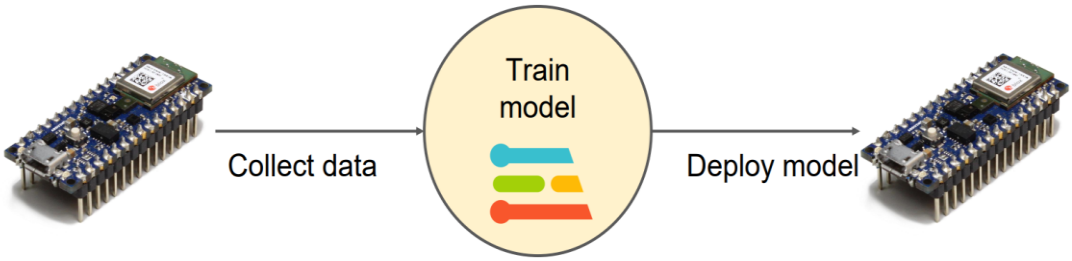


Figure 23 Basic Approach to train and deploy model [21]

The said model was trained and tested using holdout method. In which the dataset is divided into 80% training and 20% validation set. And when the model is trained then the model is tested using 20% test set.

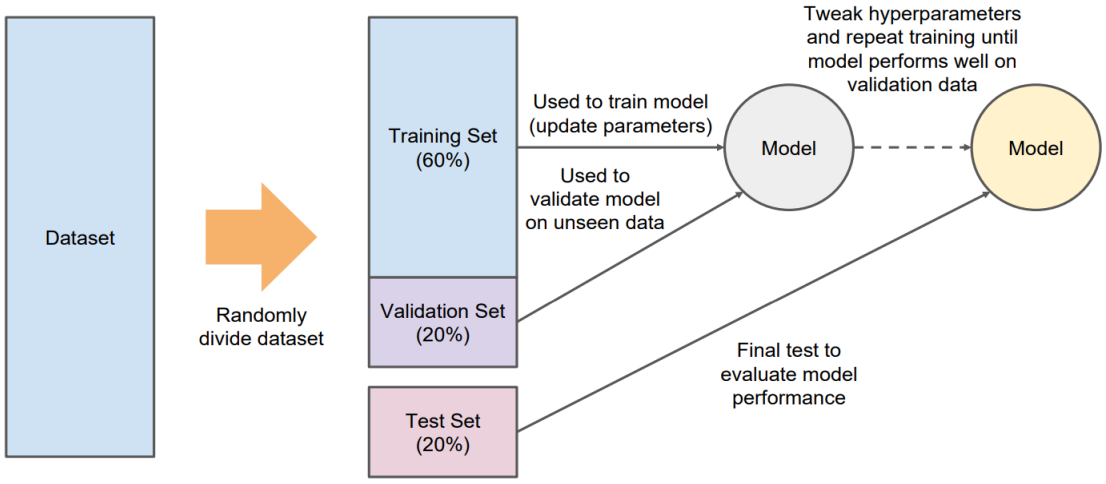


Figure 24 Basics of Model Training in Embedded Machine Learning [21]

The dataset can be fed and deployed using edge impulse using the following ways as described by the edge impulse:

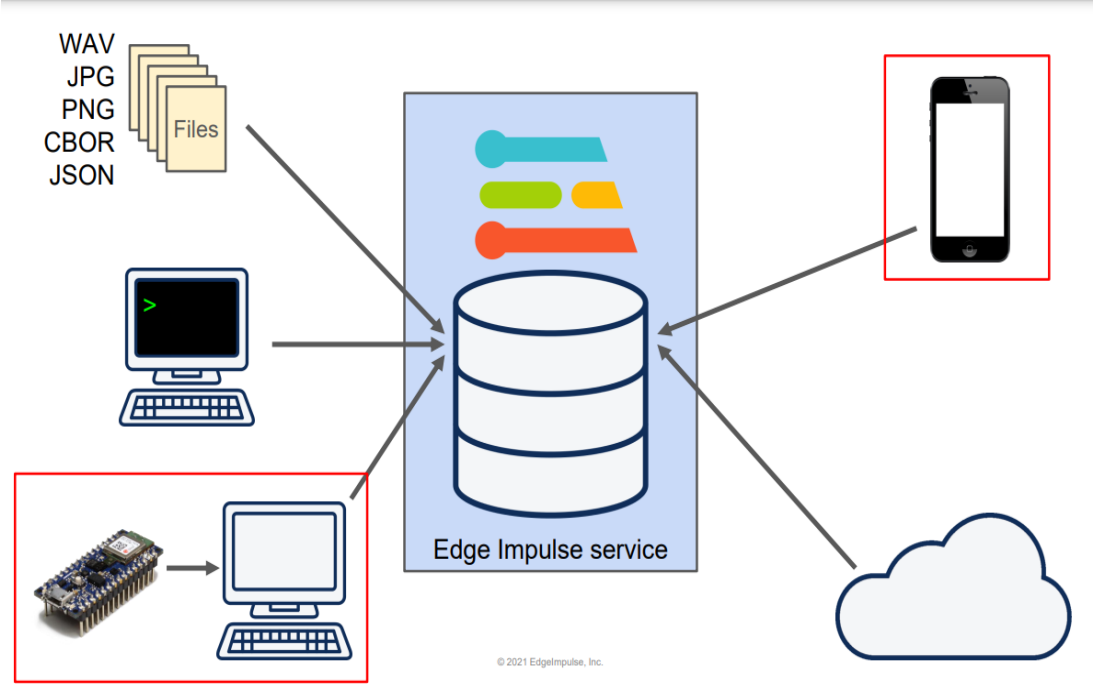


Figure 25 Ways to feed, train and deploy data in edge impulse [21]

In our case we have stored the data of fetal heart signal using .csv file.

3.7.1 What is classification in Machine learning:

Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

In the case of fetal heart signal classification, I have used neural network classification. The basic idea is that a neural network classifier will take some input data and output a probability score that indicates how likely it is that the input data belongs to a particular class.

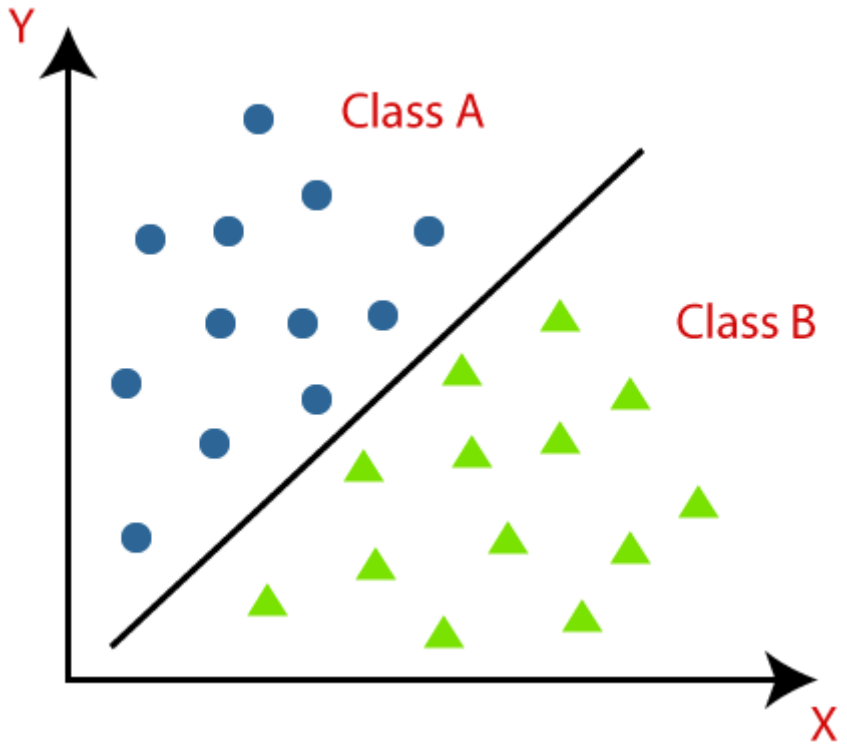


Figure 26 Classification Algorithm in ML [21]

3.7.2 Dataset and its preparation for Fetal Heart Signal Classification:

An abdominal fetal database was taken from Physionet Bant ATM. Heartbeat signals were captured at a sample rate of 1 kHz and were digitized in 16-bit resolution. Several

four different abdominal readings were recorded using ECG electrodes placed on five different women in labor who were between 38 to 41 weeks (about 9 and a half months) of gestation. In we chose only the recording from the abdomen electrodes. Every recording consisted of five signals, four from the abdomen and one from the head of the fetus, the duration of each signal is about five minutes. In which we have taken only one recording of 1 minute. The abdominal electrodes contain four electrodes which were placed around the navel, a reference electrode, and a common mode reference electrode (with active ground signal) placed above the pubic symphysis, and on the left leg, respectively. The data was separated into two different files with fetal heart recordings. One file contains the label "Normal Fetal Heart Signal" and the other file has the label of "Abnormal Fetal Heart Signal."

Each 1-minute recorded signal is sub-divided into sixty signals of one second duration. The heart rate was recorded using the algorithm as described in earlier chapter.

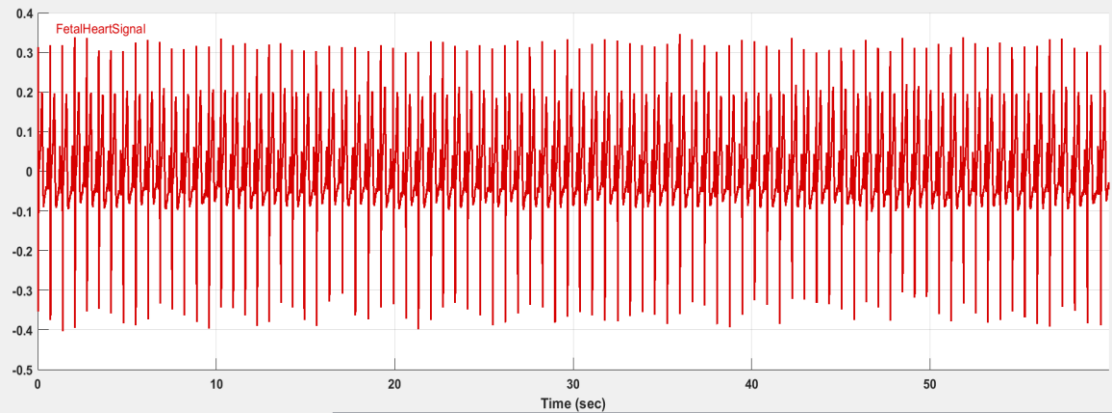


Figure 27 Data of normal fetus heart beats

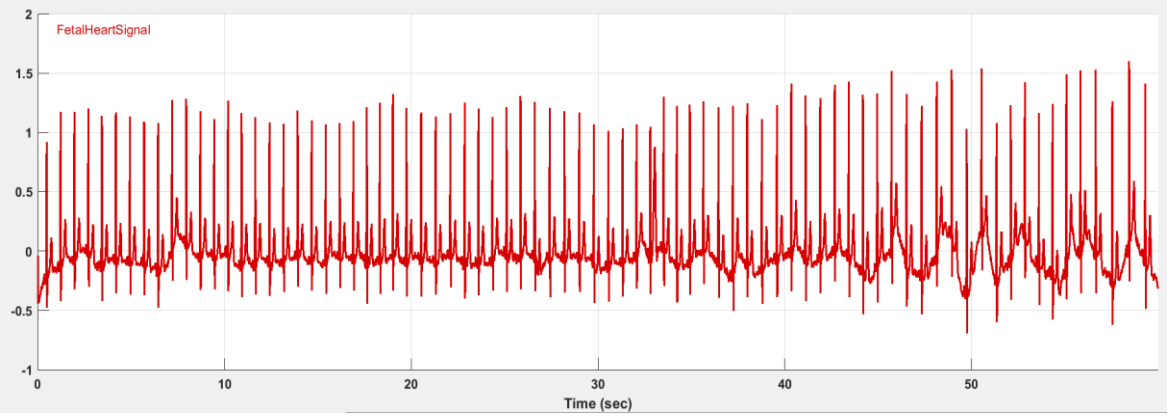


Figure 28 Data of abnormal fetus heart beats

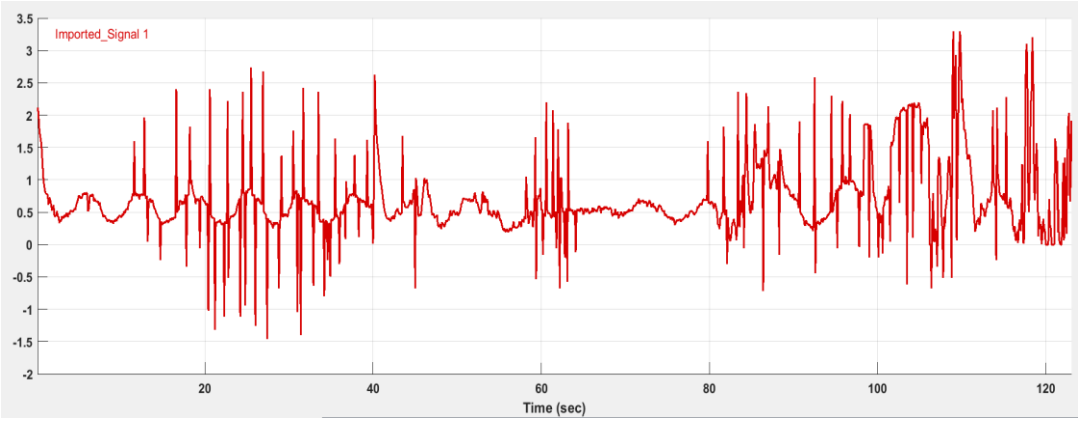


Figure 29 Data from proposed device

3.7.3 TinyML Framework and Modeling:

Edge Impulse Studio (EIS) is a cloud-embedded Machine Learning platform specifically designed for developing, training, evaluating, and deploying models. When executing ML models on embedded devices, such as those based on the ARMTM architecture, TinyML is employed. This involves using TensorFlow Lite (tflite) for microcontrollers, which quantizes the originally trained float32 precision models into int8 precision tflite models. The modeling process begins with training the model on the cloud using high-performance computing. Once training is complete, the model is converted into a flat

buffer format and then transformed into a binary C program. This program is integrated into the firmware of the edge device to enable edge inference.

In the EIS modeling process, 80% of the dataset was utilized for training, while 20% was reserved for testing purposes. Within the training dataset, 20% of the data was further set aside for validation. To segment the data for analysis, a window size of 1000 ms (approximately 1 second) was chosen.

The TFlite code for training the model is:

```
import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten, ReLU, Softmax

from tensorflow.keras.optimizers import Adam

EPOCHS = args.epochs or 30

LEARNING_RATE = args.learning_rate or 0.005

# this controls the batch size, or you can manipulate the tf.data.Dataset objects yourself

BATCH_SIZE = 32

train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)

validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)


# model architecture

model = Sequential()

model.add(Dense(20, activation='relu',

    activity_regularizer=tf.keras.regularizers.l1(0.00001)))
```

```
model.add(Dense(10, activation='relu',
    activity_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(Dense(classes, name='y_pred', activation='softmax'))

# this controls the learning rate
opt = Adam(learning_rate=LEARNING_RATE, beta_1=0.9, beta_2=0.999)
callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count, epochs=EPOCHS))

# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
model.fit(train_dataset, epochs=EPOCHS, validation_data=validation_dataset, verbose=2, callbacks=callbacks)
# Use this flag to disable per-channel quantization for a model.
# This can reduce RAM usage for convolutional models, but may have
# an impact on accuracy.
disable_per_channel_quantization = False
```

3.7.4 Performance of TinyML Model for Fetal Heart Signal Classification:

We selected the RNN classifier as our learning block for training because our models are intended for deployment on edge devices where memory size is a cost consideration. Compared to other neural networks, RNN requires less memory, making it a suitable choice. The fundamental concept behind the RNN classifier is to take input data and generate a probability score indicating the likelihood of that data belonging to a specific class. Throughout the training process, we adjusted and utilized model training

hyperparameters, such as the number of training cycles and learning rate, to attain maximum accuracy from our TinyML model [22]. The achieved accuracy was determined through the classification of the fetus's heart signal. To optimize the model for deployment on microcontrollers, the weights and biases of the originally trained model, which were in float32 precision, were quantized into an int8 precision model using the tflite for micro-controller framework. To evaluate the performance of the model on the test and validation sets, a confusion metric was computed and analyzed.

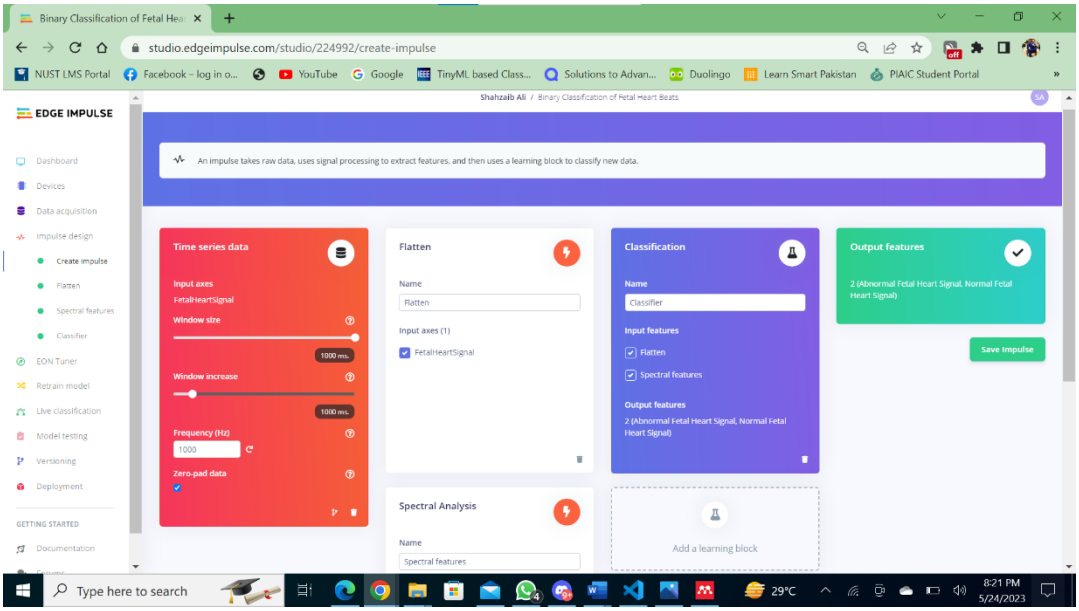


Figure 30 Impulse Design on Edge Impulse studio

S.No.	Training cycles	Learning Rate	Accuracy
1.	100	0.5	69.2%
2.	100	0.05	89.7%
3.	100	0.005	92.3%
4.	40	0.005	92.3%
5.	30	0.005	92.3%

Table 1 Variation in Parameters of RNN Model

	ABNORMAL FETAL HEART SIGNAL	NORMAL FETAL HEART SIGNAL
ABNORMAL FETAL HEART SIGNAL	85%	15%
NORMAL FETAL HEART SIGNAL	0%	100%
F1 SCORE	0.92	0.93

Figure 31 : Confusion Matrix for binary classification of Fetal Heart Signal

3.7.4.1 Accuracy (Acc)

Ratio of correct prediction and the total number of predictions.

$$Acc = \frac{T_p + T_n}{T_p + F_p + T_n + F_n}$$

3.7.4.2 Sensitivity (Sen)

The ability of a model to predict true positives in each of the available categories.

$$Sen = \frac{T_p}{T_p + F_n}$$

3.7.4.3 Specificity (Spec)

The ratio of true negatives to all negatives.

$$Spec = \frac{T_n}{T_n + F_p}$$

Where,

Tp: Correct values which are detected as correct.

Tn: Incorrect values which are detected as correct.

Fp: Correct value which is detected as incorrect.

Fn: Incorrect values which are detected as incorrect.

S.No.	Metrics	Validation set	Test set
1	Acc	92.3%	91.67%
2	Spec	89.2%	88.3+
3	Sen	0.085	0.08

Table 2 Model Evaluation using Test and Validation set

3.7.4.4 F1 Score (F1)

F1 score measures the percentage of correct predictions that a machine learning model has made.

$$F_1 = \frac{2 * P * R}{P + R}$$

Where, P is the precision … R is the recall.

3.8 Deployment of Machine Learning Model on RP2040 Microcontroller

The most difficult part was to deploy the trained machine learning model from edge impulse on the microcontroller we chose was RP2040 microcontroller.

In order to deploy machine learning model on any microcontroller edge impulse offers the following ways which are in the scope of our microcontroller:

- Arduino Library
- C++ Library
- STM cube ide library

Edge impulse can also give the .uf2 binary file to deploy to microcontroller to test the inference using edge impulse data forwarder.

Here I have chosen to deploy the C++ library to the RP2040 microcontroller as the library is customizable and can be used to show only the desired results to users on

UI/UX or LCD screen. This is also useful for utilizing less memory rather than the binary file directly from the edge impulse, which is also not customizable.

As already discussed in the earlier chapter, the about the binary classification of the fetus heart signal and the trained model. I have downloaded the C++ library by the name of "Binary Classification of Fetal Heart Signal" to deploy onto the RP2040.

Edge impulse library give us the a .zip file which has the follow folders and files:

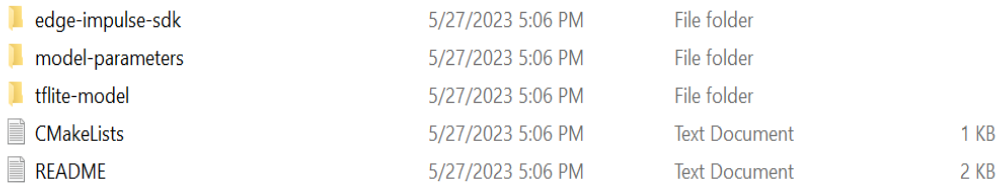


Figure 32 SDK files from Edge impulse

These files have all your signal processing blocks, configuration and learning blocks up into a single package that were shown in the impulse design in previous chapter.

To compile this library for RP2040 you will need the pico-sdk, CMake, a cross-platform tool used to build the software, and the GNU Embedded Toolchain for Arm which are given in the references section.

In the files mentioned above in the figure there is the following C++ header file "edge-impulse/classifier/ei_run_classifier.h" which is the use full purposes and is used in the mai.cpp program.

Figure 33 Important Header File in EI SDK

Add this header file to your main.cpp file using preprocessor directive, to access all the functions and results provided by the edge impulse.

3.8.1 Printing inference and prediction results on serial monitor:

In order to get the idea how the model is progessing on he edge device I have designed an algorithm that how we can get the desird classification results on the VS code.

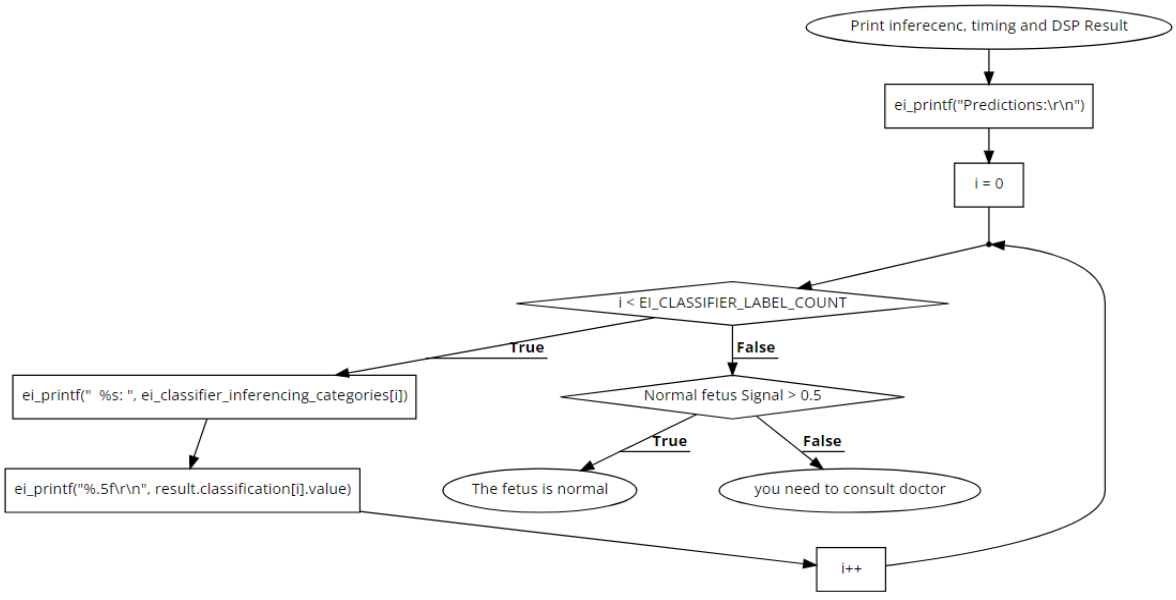


Figure 34 Flow chart for printing inference and classification results

Following code which I used to print the inference results and prediction results on the serial monitor of the VS code.

```
const uint LED_PIN = 25;
float features[]={
    //paste raw features from edge impulse here
};
int raw_feature_get_data(size_t offset, size_t length, float *out_ptr) {
  memcpy(out_ptr, features + offset, length * sizeof(float));
  return 0;
void print_inference_result(ei_impulse_result_t result) {
    // Print how long it took to perform inference
    ei_printf("Timing: DSP %d ms, inference %d ms \r\n",
            result.timing.dsp,
            result.timing.classification);
    // Print the prediction results (classification)
    ei_printf("Predictions:\r\n");
    for (uint16_t i = 0; i < EI_CLASSIFIER_LABEL_COUNT; i++) {
        ei_printf("  %s: ", ei_classifier_inferencing_categories[i]);
        ei_printf("%.5f\r\n", result.classification[i].value);
    }
}
```

In the above code I have used the functions from ei_classifier header file such as print_inference_result() and print_classification_result(). These functions print the the inference and classification results of Fetus heart signal as per the model deployed on the microcontroller.

The flow chart for the main function code is shown below:

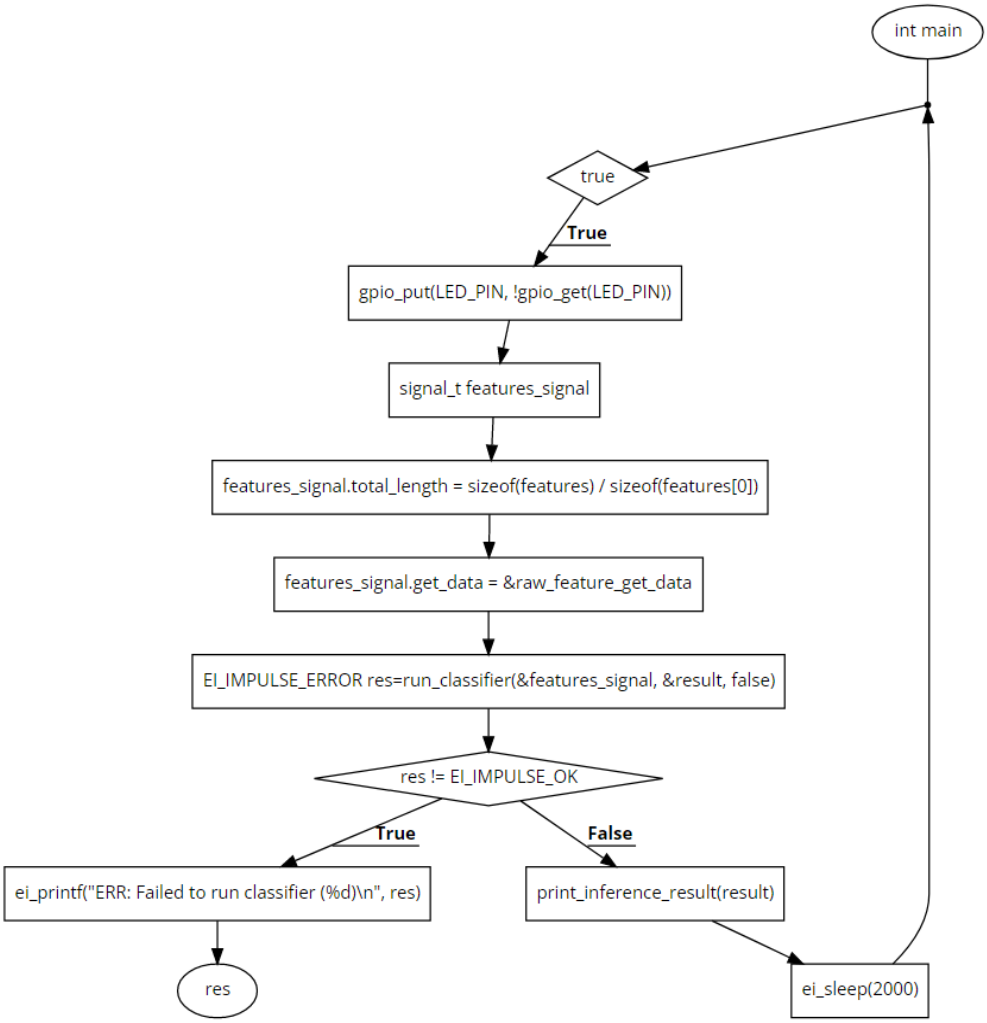


Figure 35 Flow chart of main function

Following is the main function code for getting the classification results for fetus heart signal:

```
int main() {
    stdio_usb_init();
    stdio_init_all();
    printf("ADC Example, measuring GPIO26\n");
    adc_init();
    // Make sure GPIO is high-impedance, no pullups etc
 adc_gpio_init(26);
    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);
    ei_impulse_result_t result = {nullptr};
    ei_printf("Edge Impulse standalone inferencing (Raspberry Pi Pico)\n");
   if (sizeof(features) / sizeof(float) != EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE)
{
        ei_printf("The size of your 'features' array is not correct. Expected %d items, but
had %u\n",
       EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE, sizeof(features) / sizeof(float));
       return 1;}
    while (true) {
       gpio_put(LED_PIN, !gpio_get(LED_PIN));
       signal_t features_signal;
       features_signal.total_length = sizeof(features) / sizeof(features[0]);
```

```
        features_signal.get_data = &raw_feature_get_data;
        EI_IMPULSE_ERROR res=run_classifier(&features_signal, &result, false);
        if (res != EI_IMPULSE_OK) {
            ei_printf("ERR: Failed to run classifier (%d)\n", res);
            return res;  }
        print_inference_result(result);
        ei_sleep(2000);
    }
}
```

This code helped to get the inference, timing and classification results from the trained neural network.

By this the trained neural network model “Binary Classification of Fetus Heart Signal” is ready to be deployed on the RP2040 micro controller.

Now, after configuring and building the whole project the build directory is read and a standalone_infernce.uf2 is ready to be deployed on to the microcontroller.

To burn the file, press the bootsel button of the RP2040 microcontroller and drag the file into the disc of RP2040 microcontroller.

Now in the features array we are supposed to give 1000 samples of raw features of Fetal heart signal which we can copy and paste from the edge impulse model testing tab uder the head of the raw features as shown in the figure below of raw fetus signal on edge impulse:

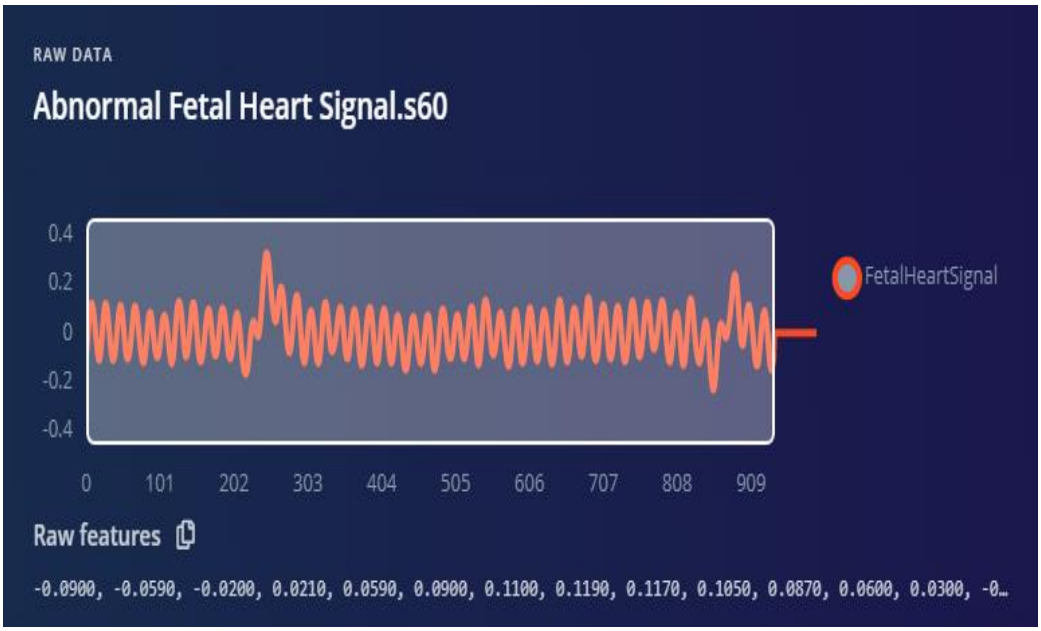


Figure 36 Raw Fetus Signal on Edge Impulse Studio

The results on the edge impulse model testing tab are shown in the figure below:

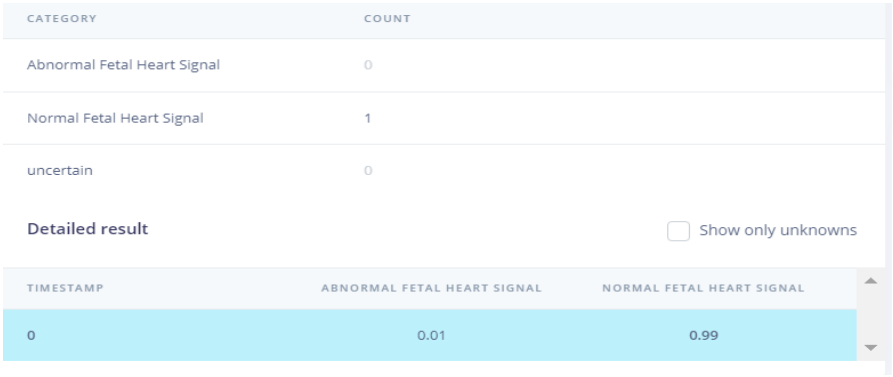


Figure 37 Classification Results on Edge impulse

These raw features are those features which are gathered from the ADC of microcontroller in the coming section I will explain how I automated the result to copy the raw feature into the features array.

Meanwhile for this the inference was showed the following results on the serial monitor for VS code:

```
  Normal Fetal Heart Signal: 0.21094
You need to Consult Your Doctor...Timing: DSP 151 ms, inference 1 ms
Predictions:
  Abnormal Fetal Heart Signal: 0.78906
  Normal Fetal Heart Signal: 0.21094
You need to Consult Your Doctor...
```

Figure 38 Classification Results from Algorithm Deployed on RP2040

Now do the same thing with the normal fetal heart signal. The following raw feature is given to the features array in main.cpp:

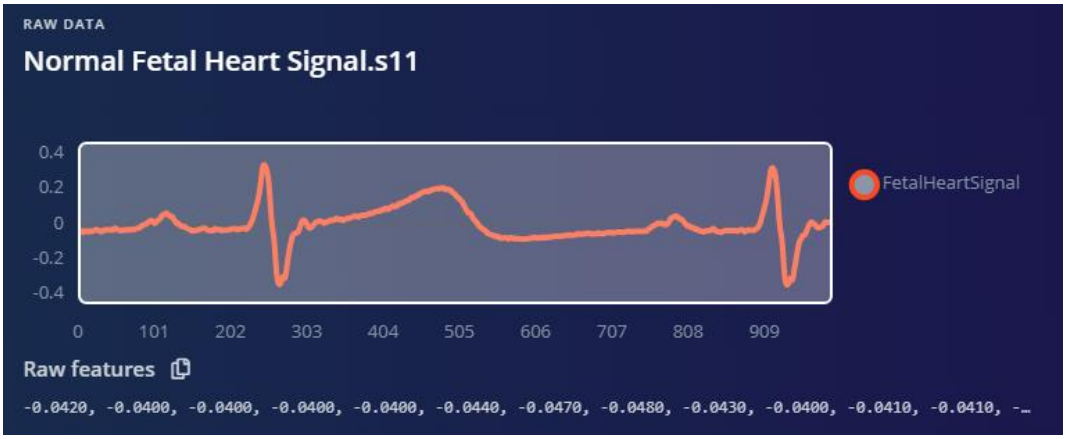


Figure 39 Normal Fetus signal on Edge impulse studio

These raw features are those features which are gathered from the ADC of microcontroller in the coming section I will explain how I automated the result to copy the raw feature into the features array.

Meanwhile for this the inference was showed the following results on the serial monitor for VS code:

```
Predictions:
  Abnormal Fetal Heart Signal: 0.32812
  Normal Fetal Heart Signal: 0.67188
Everything is fine.....No worriesTiming: DSP 150 ms, inference 1 ms
Predictions:
  Abnormal Fetal Heart Signal: 0.32812
  Normal Fetal Heart Signal: 0.67188
```

Figure 40 Classification Results of ML Model deployed on RP2040

These results verified that the raw data in the edge impulse and the same raw data given to the features array in the C++ code resulted in the same classification results. This means that the method of deploying the Edge impulse SDK is correct and ready to be automated.

3.8.2 Automation of features Array in C++ code:

To automate the features array in C++ code I have written the following code:

```
adc_select_input(0);
for (int i = 0; i < 1000; i++) {
const float conversion_factor = 3.3f / (1 << 12);
    uint16_t result = adc_read();
    const float Voltage = result * conversion_factor;
    printf("%d\n", result);
    features[i]=Voltage;
    sleep_ms(1);
  }
```

As in the earlier section you have seen that the features array is defined globally. Now in the above code the for loop runs for 1000 times and stores the ADC values in result and a conversion factor is applied to convert 12-bit ADC values into decimal values which are stored in “voltage” variable. Which are eventually stored in a features array.

Conversion Factor:

The conversion factor mentioned in the code snippet is used to convert the ADC (Analog-to-Digital Converter) readings into voltage values.

Let's break down how the conversion factor is calculated:

The maximum value of the ADC is decided by the number of bits it uses. In this case, the code snippet assumes a 12-bit ADC, which means it can be 2^12 (4096) discrete levels.

The voltage range the ADC covers is specified as 3.3V, the reference voltage used. To convert the ADC readings to voltage, the conversion factor is computed by dividing the voltage range (3.3V) by the maximum ADC value (4096). This division results in a scaling factor that maps each ADC level to its corresponding voltage value. The conversion factor is then stored as a constant float variable called conversion_factor for later use in converting the ADC readings to voltage.

By multiplying the ADC readings by the conversion factor, the code snippet calculates the corresponding voltage value for each reading.

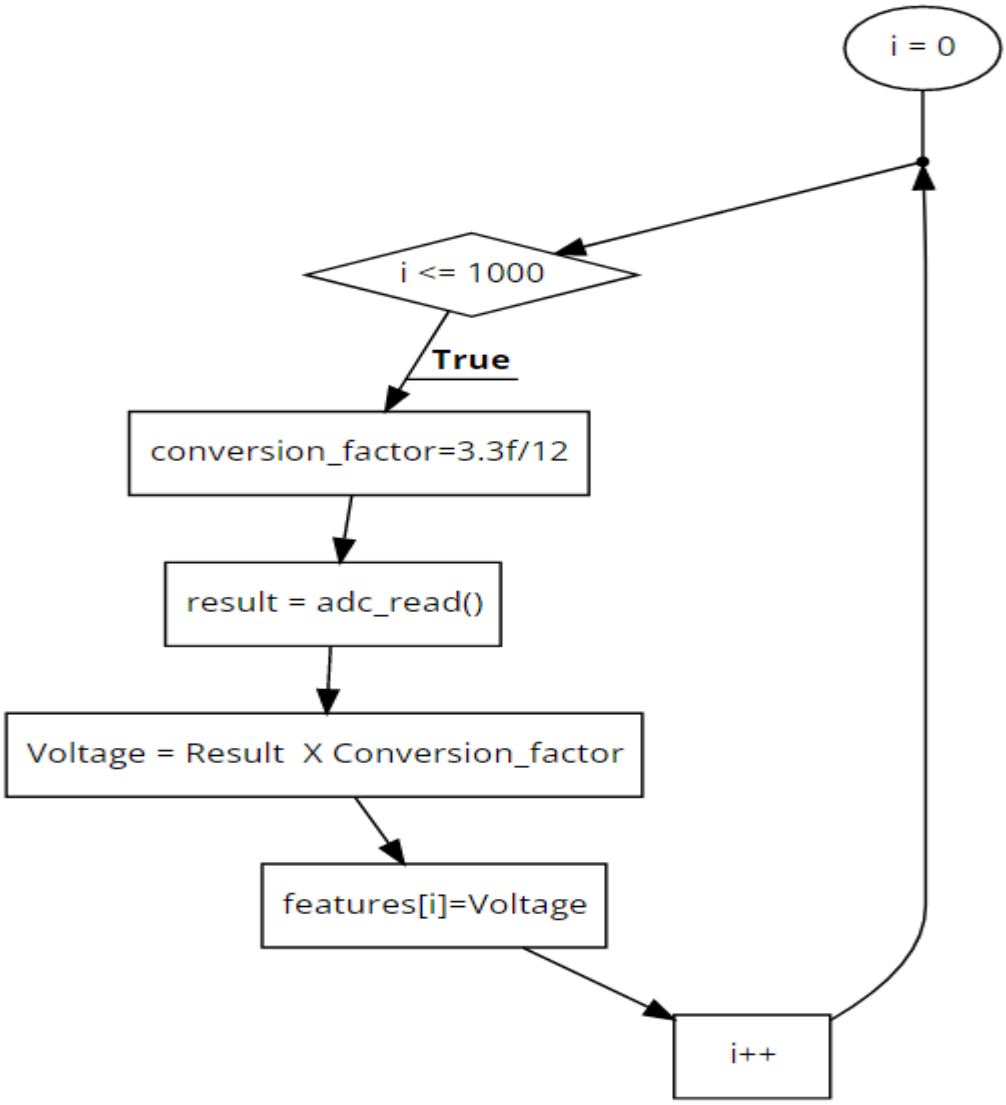


Figure 41 Flow chart for Automation Features Array

Also define the features array globally as:

```
#define NUM_SAMPLE 1000

float features[NUM_SAMPLE];
```

Hence the following modification in the code will automate the features and will be updated every time when the probe is placed on the womb of mother.

3.9 Summary of Fetal Heart Rate Calculation and Fetal Signal Analysis:

The flow of fetal heart rate monitoring is divided into five steps, with a focus on the microcontroller part and displaying the results on an LCD display. The algorithm for calculating beats per minute (BPM) is implemented using two approaches: FHR (Fetal Heart Rate) estimation by peak-to-peak time between Doppler shifts and FHR estimation using the peak-to-peak time between Doppler shifts. The fetal signal is obtained using a probe that measures Doppler shift, and the signal is filtered using bandpass filters before being fed into the Raspberry Pi Pico's 12-bit ADC.

The filtered fetal signal is recorded and logged on to a laptop using a C program. The data acquisition is performed by visiting a hospital and using a probe, filter circuit, and the mentioned code. The acquired data is then used to develop an algorithm for finding peaks in the signal, which is used to calculate the beats per minute (BPM) of the fetal heart rate.

For classification of fetal heart signal non-invasive data of fetal heart taken is taken. The dataset consists of abdominal fetal recordings captured at a sample rate of 1 kHz and digitized in 16-bit resolution. The signals are divided into normal and abnormal fetal heart signals, and an algorithm is applied to classify them.

The classification is done using the TinyML framework and modeling in Edge Impulse Studio. An RNN classifier is trained on 80% of the dataset, and the performance is evaluated using the remaining 20%. The accuracy, sensitivity, specificity, and F1 score are computed to analyze the model's performance. During training the accuracy of 91.5% was achieved. The beauty of the model is that it is deployed on the embedded system.

Once the model is trained, it is deployed on the RP2040 microcontroller using the Edge Impulse C++ library. The deployment involves incorporating the trained model into the microcontroller firmware. The main.cpp program is modified to include the necessary headers and functions to access the results provided by Edge Impulse.

The inference results and prediction results from the trained neural network are printed on the serial monitor of VS code. The features array in the C++ code is automated by reading the ADC values and applying a conversion factor to convert them into voltage values. This automation ensures that the features array is updated every the probe is placed on the mother's womb.

Concluding all testing of the model in Benazir Bhutto Hospital, Murree Road Rawalpindi, and the need for further evaluation of the system's performance and accuracy.

In summary, the described project focuses on the development of a fetal heart rate monitoring system using an AI interface. It involves calculating the beats per minute from fetal heart signals, deploying a machine learning model on the Raspberry Pi Pico microcontroller, and analyzing the fetal heart signals for classification. The system shows promising results and can improve the detection and monitoring of fetal heart diseases.

Chapter 4 - Results

This section includes the results that we achieved In our hardware and software section. We were able to acquire the desired simulation of piezoelectric disc, analoge filter response, algorithm efficiency, beats per minute of fetus and their classification as normal and abnormal heart beats.

4.1 Device Hardware:

We successfully fabricated the circuit of our device. We made a single layer PCB from the schematic describing chapter 3. Our hardware device include:

- LCD display
- Pico microcontroller development kit
- Analog front end
- Batteries
- 3-megahertz probe
- On off switch
- Acrylic base

We were able to power-up and operate the 3 MHz probe directly from our circuit. Thus we were able to get the doppler signal from probe and were able to calculate beats per minute and displayed them on the LCD screen. The images of ours hardware device are shown in figures on next pages. This is now a complete handheld device that is portable, cost effective, and intelligent. The device is mounted on acrylic 2mm sheet to protect the soldering and to make it a one complete unit.

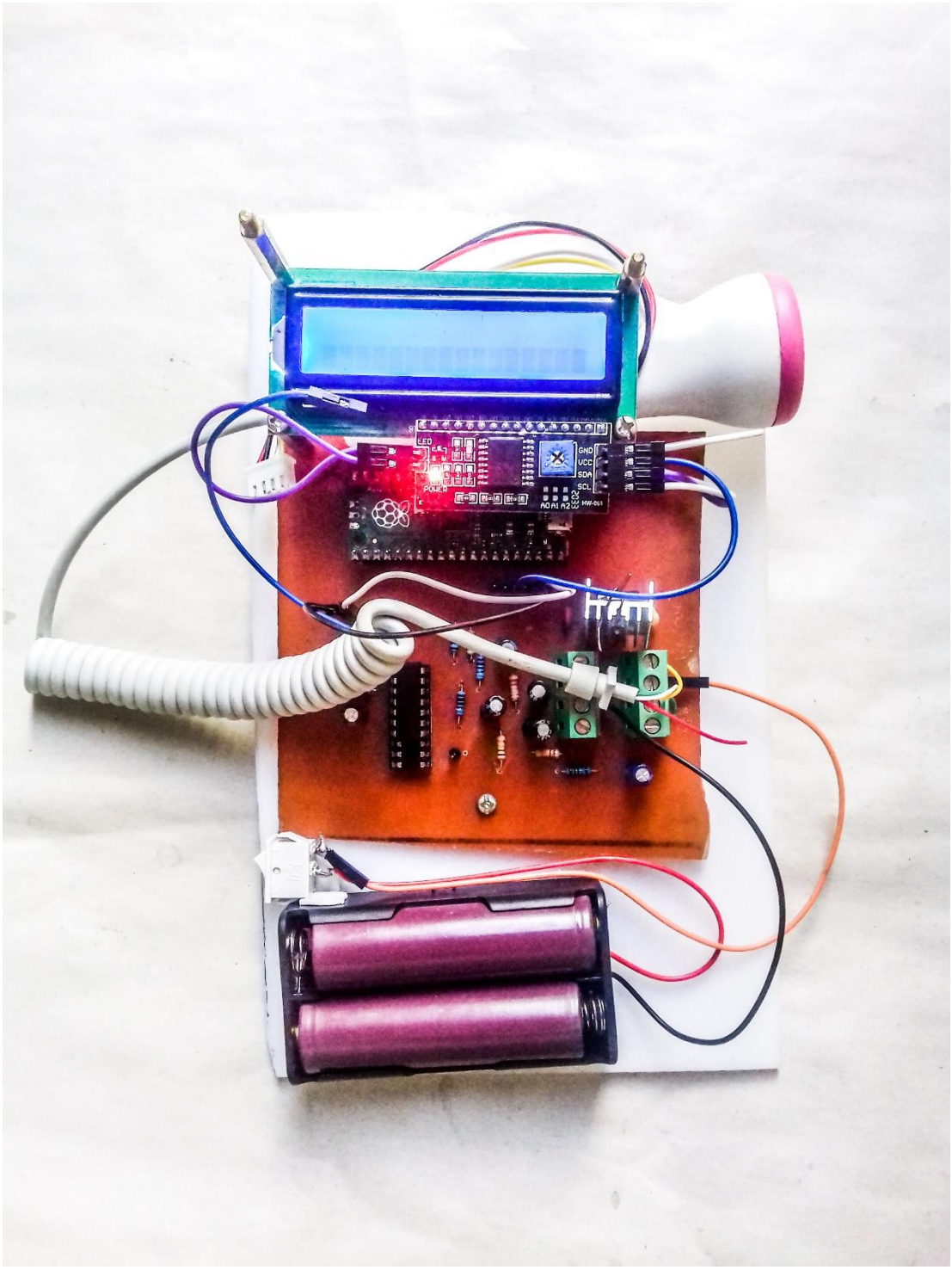


Figure 42 Final Device that we were able to operate and calculate BPM of fetus

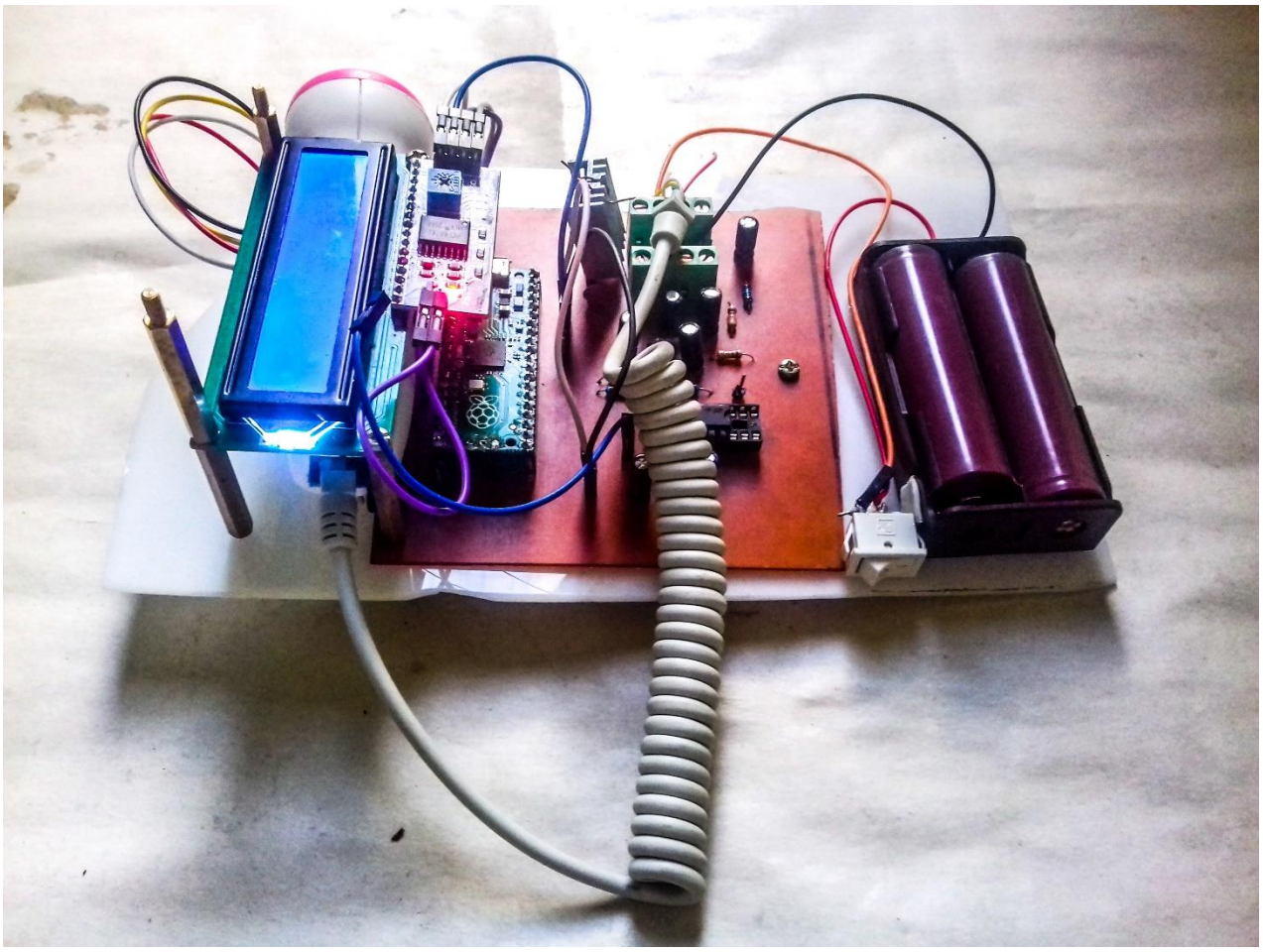


Figure 43 Side view of the device

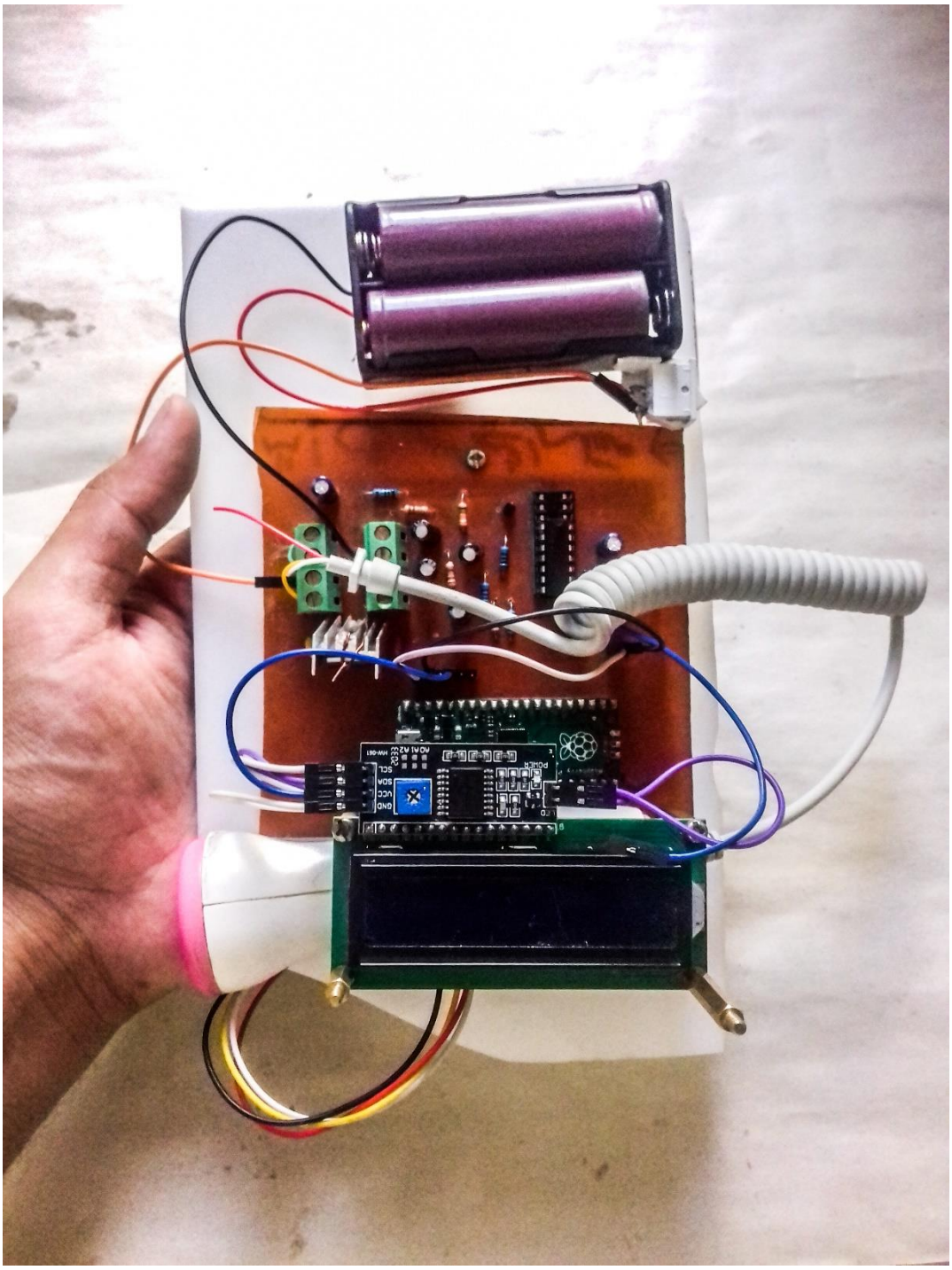


Figure 44 Device placed in hand

4.2 Analog Filter Response:

Our device includes one bandpass filter that has a pass band frequency bandwidth of 7hz-160hz. We simulated the schematic of our circuit in Lt. spice to get the bode plot. This is how a bode plot resulted:

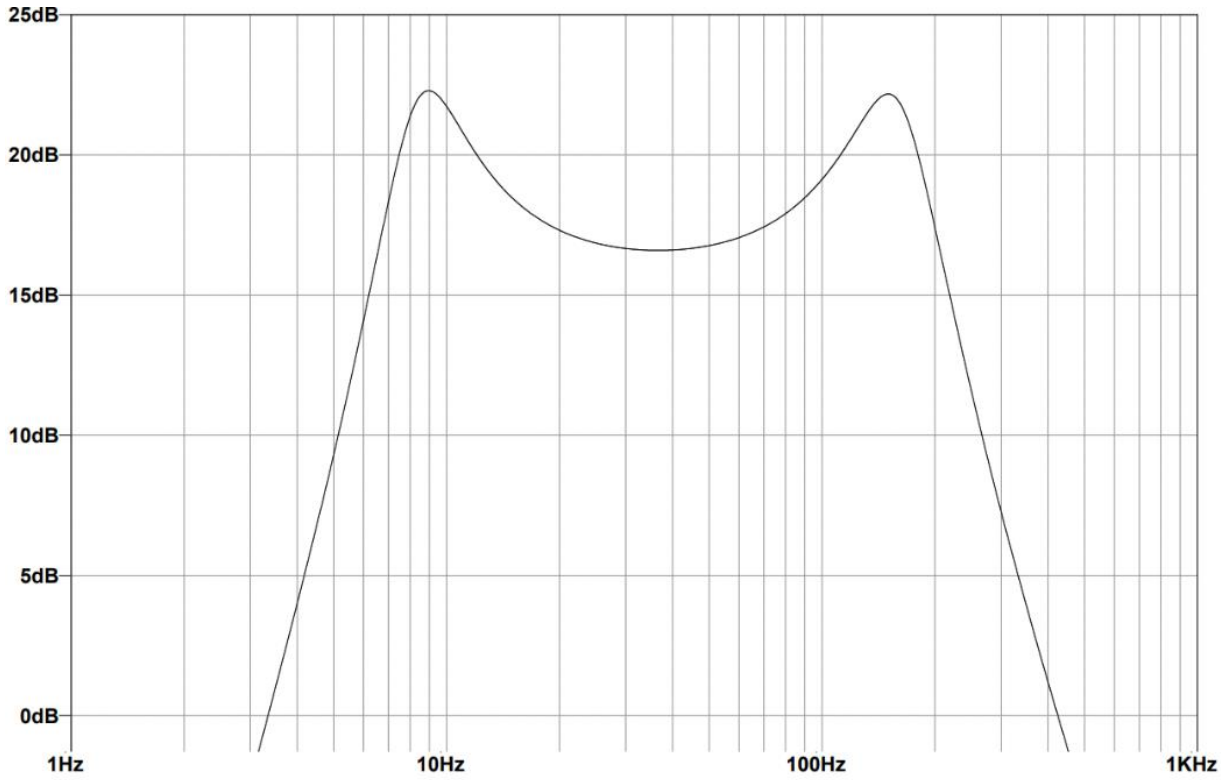


Figure 45 Bode plot of our filter

This body plot is of our complete circuit that includes a bandpass filter and an amplifier. There are ripples at the start and at the end of border plot other than that the response is stable. These ripples or actually our corner frequencies. Through this body plot we can easily observe that the frequencies other than 7 Hertz-160Hz attenuated by our filter.

Similarly we used Lt. spice to get the transient response of our filter. We got a good transient response that is shown on next page.

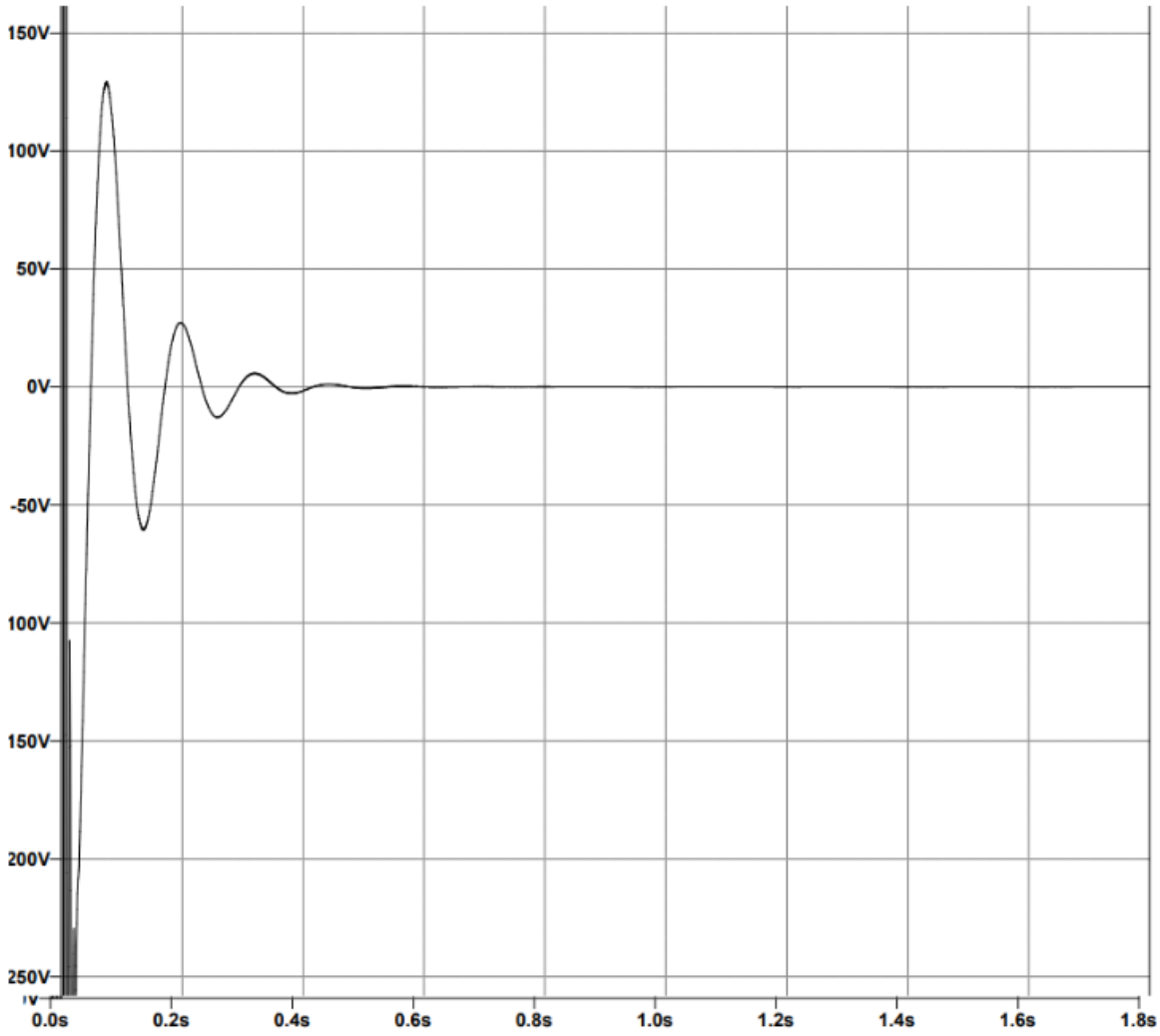


Figure 46 Transient response of our filter

This transient response has following characteristics:

- Delay time = 0.03s
- Steady state error = Zero
- Peak time = 0.05s
- Settling time = 0.5s

Therefore, a transient response this stable

4.3 Output of analog front end:

We are testing our circuit by rubbing the finger of our hand adt the head of our probe and observed the inputs and outputs of our circuit on oscilloscope. Figure 47 shows the response of our complete circuit. Our input signal coming from the probe as noise and amplitude in millivolts. Whereas our output signal only contain the peaks that resembles to the heartbeat of fetus. You can also see the change in amplitude. Our output signal is greater than 1Volt. You can see that noise is efficiently removed and only the required output signal is generated by our circuit.

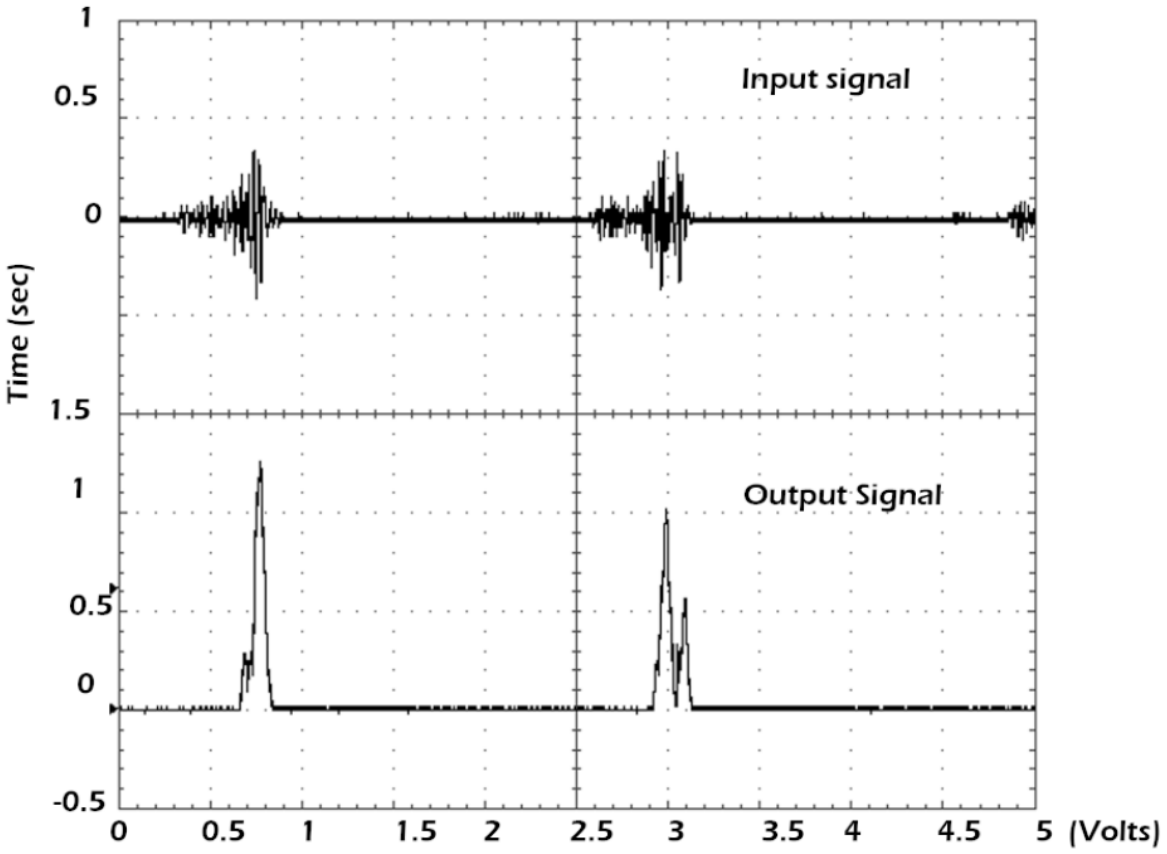


Figure 47 Response of our complete circuit

Similarly, here is the response of our bandpass filter. The peaks are still over there but they are not amplified. And has the same amplitude as of input signal. However there is almost no noise in the output signal.

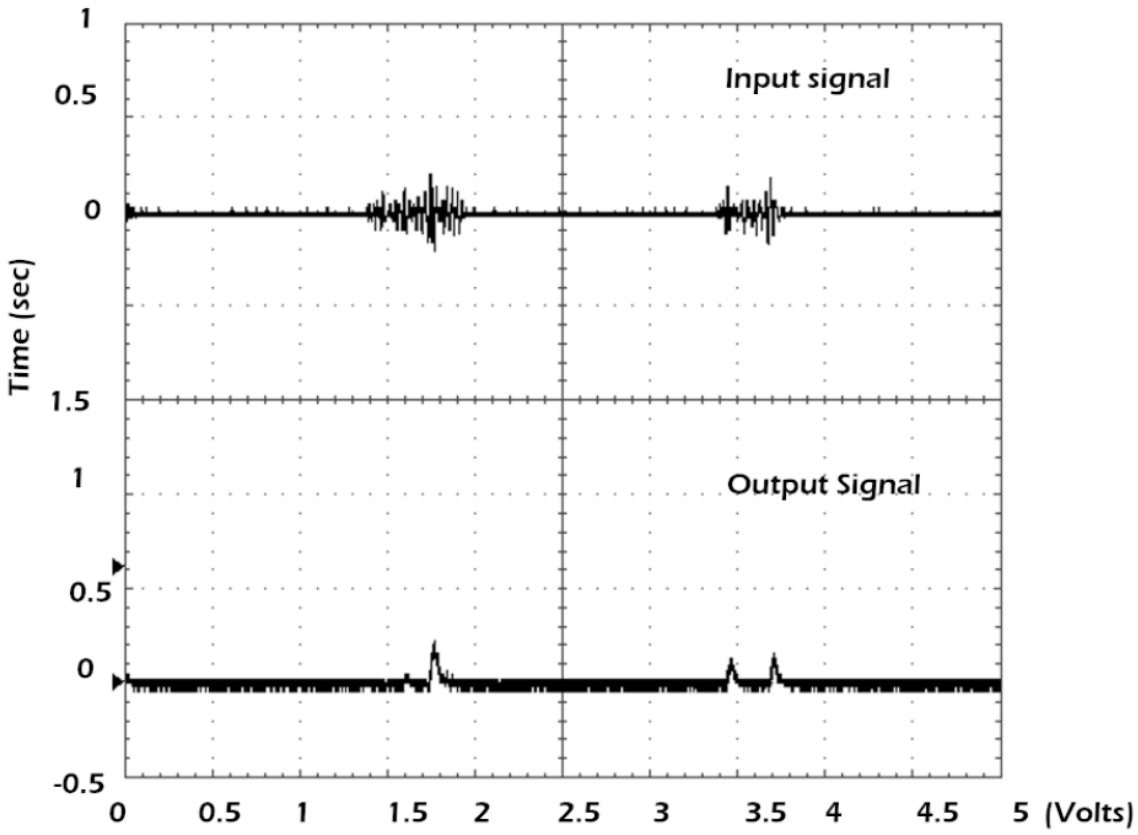


Figure 48 Output of bandpass filter only.

Similarly, you can also see the output response of our high pass filter only. The high pass filter removes the noise but is not able to remove the noise presenting heartbeat signal. This noise may include maternal heartbeat noise and other noises related to the flow of blood. However, this filter is not able to be implied that signal. Therefore, the result of output signal is also in millivolts.

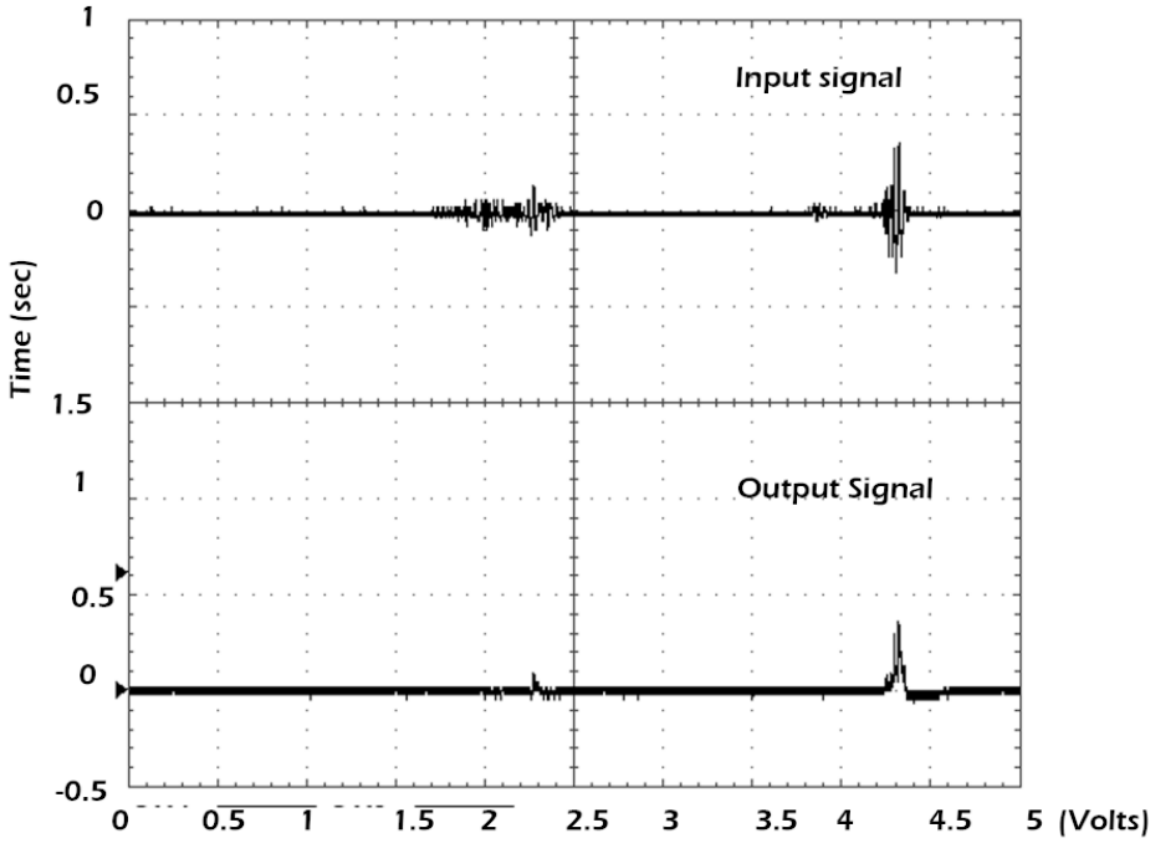


Figure 49 Result of High pass signal only

4.4 Beats per Minute of Fetus:

To get the beats per minute the peak time is an important factor to calculate as discussed in chapter 3. So, after the immense testing of the proposed device the average peak time came out to be around 250 ms to 450 ms for the case of fetus which give us accurate results of 110 to 150 BPM with an accuracy of almost 90%.

4.5 TinyML Model Testing Results:

For the Embedded machine learning model testing we have given 20 samples to the trained model which gave the model accuracy of 91.67%. Which shows that our model is best fitted for the classification as shown in the figure 40.

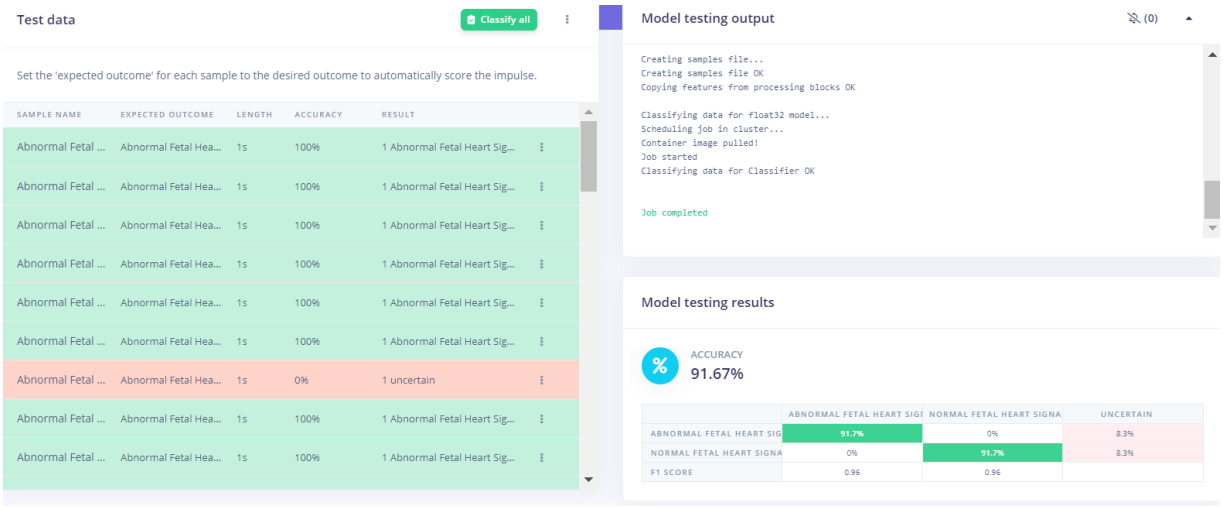


Figure 50 Classification Model testing results

In the above figure, 20 different unknow samples are given to the trained model of classification on edge impulse studio. The trained model of classification classifies the signal as normal and or abnormal fetus heart signal. Only 3 samples were not classified by the model and all the remaining samples were classified into normal and abnormal fetus heart signal.

Upon the deployment of the classification model on the RP2040 microcontroller the latency, RAM, flash, Accuracy in real time were detected. Which was quite good for the RP2040 which has only 2 Mega bytes of flash memory.

	FLATTEN	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	17 ms.	234 ms.	20 ms.	**252 ms.**
RAM	7.8K	8.3K	1.3K	**8.3K**
FLASH	-	-	13.1K	**-**
ACCURACY				**91.67%**

Table 3 Results of model optimization using EON compiler.

4.6 Piezo Disc Simulation Results

The dimensions of the piezoelectric disc used in the analysis are a thickness (TH) of 1 mm and an outer diameter (OD) of 30 mm. The resonance frequencies of the piezoelectric disc in the radial disc mode and thickness disc mode were calculated:

a. The resonance frequency in the radial disc mode is approximately 50.8 kHz (0.0508 MHz).

b. The resonance frequency in the thickness disc mode is approximately 2.1 MHz

Using the COMSOL Analysis of the disc, the following the outcomes for Eigenfrequencies, Angular frequencies, Damping ratios and Quality Factors. The imaginary part in the eigenfrequencies shows the losses factors.

Eigenfrequency (Hz)	Angular frequency (rad/s)	Damping ratio	Quality Factor
4.6086E5+238.45i	2.8957E6+1498.3i	5.1741E-4	966.35
4.6109E5+238.24i	2.8971E6+1496.9i	5.1670E-4	967.68
4.6237E5+241.27i	2.9051E6+1515.9i	5.2181E-4	958.21
4.6248E5+241.52i	2.9058E6+1517.5i	5.2222E-4	957.45
4.6376E5+231.92i	2.9139E6+1457.2i	5.0008E-4	999.84
4.6516E5+244.78i	2.9227E6+1538.0i	5.2623E-4	950.15

Table 4 Eigen Frequencies

Chapter 5 - Conclusions and Future Work

In this methodology, we discussed the process of data collection, hardware design, and the development of a machine learning model for fetal heart signal analysis. The data collection involved obtaining fetal Doppler data from an online dataset and using a 3MHz probe connected to an oscilloscope. The hardware design included the use of a piezoelectric disk, a PCB for filtering and data transmission, and a microcontroller with an LCD display.

For the digital filter design, we implemented a 4th order bandpass filter with a gain of 9.2dB to remove noise and unwanted signals, followed by a linear non-inverting amplifier with a gain of 4.6dB to amplify the filtered signal. This resulted in a clear and detectable fetal heartbeat signal.

The microcontroller used in the system was the Raspberry Pi Pico, which featured a built-in Analog-to-Digital Converter (ADC) to measure the analog voltages from the filtered signal. We developed an algorithm to calculate the beats per minute (BPM) based on the recorded signal peaks.

Furthermore, we explored the application of machine learning for the analysis of fetal heart signals. A dataset containing normal and abnormal fetal heart signals was prepared and used to train a classification model using the Edge Impulse Studio platform. The model was deployed on the microcontroller for real-time classification of fetal heart signals as normal or abnormal.

COMSOL analysis was conducted on the disc, providing outcomes such as eigenfrequencies, angular frequencies, damping ratios, and quality factors. The imaginary part in the eigenfrequencies indicates the losses factors.

Overall, this methodology provides a comprehensive approach to the development of a fetal heart rate monitor, including data collection, hardware design, signal processing, beats per minute calculation, and machine learning-based signal analysis. The designed

system can help healthcare professionals monitor fetal health and detect abnormalities in a timely manner.

While the presented methodology provides a solid foundation for a fetal heart rate monitor, there are several avenues for future work and improvements that can be explored:

1. Enhanced Signal Processing Techniques: The current methodology utilizes basic digital filters to remove noise and extract the fetal heart rate. Further research can be conducted to develop advanced signal processing techniques, such as adaptive filtering or wavelet transform, to improve the accuracy and robustness of fetal heart rate extraction. These techniques can help in handling noisy and complex fetal heart signals more effectively.
2. Wireless Monitoring: The developed hardware system relies on a wired connection between the probe and the monitoring unit. Future work can focus on incorporating wireless communication capabilities, such as Bluetooth or Wi-Fi, to enable remote monitoring of fetal heart signals. This would provide greater mobility and convenience for both healthcare professionals and expectant mothers.
3. Real-Time Visualization and Alerts: Currently, the hardware system displays the fetal heart rate on an LCD screen. Expanding the system to include real-time visualization of the fetal heart waveform can offer additional insights and assist healthcare professionals in better understanding the fetal heart patterns. Furthermore, implementing an alert system that notifies medical staff in case of abnormal heart rates or patterns can enhance the timely detection of potential fetal distress.
4. Long-Term Monitoring and Data Analysis: Extending the monitoring capabilities beyond real-time tracking can be valuable for longitudinal studies and monitoring high-risk pregnancies. Developing a mechanism to store and analyze long-term data can help identify patterns, trends, and correlations that contribute to a deeper understanding of fetal health.

5. Clinical Validation and Regulatory Approval: To ensure the reliability and safety of the fetal heart rate monitor, future work should involve clinical validation studies involving a larger sample size of pregnant women. This would provide valuable data for assessing the accuracy, sensitivity, and specificity of the developed system. Also, seeking regulatory approval and compliance with relevant medical device standards is essential for the technology's widespread adoption.
6. Integration with Electronic Health Records (EHR): Integrating the fetal heart rate monitor with existing electronic health record systems can streamline the documentation process and enable seamless sharing of fetal health data between healthcare providers. This integration can enhance the continuity of care and facilitate collaboration among different healthcare professionals involved in the management of pregnancies.

In conclusion, future work in the field of fetal heart rate monitoring can focus on advancing signal processing techniques, incorporating wireless monitoring capabilities, enabling real-time visualization and alerts, facilitating long-term monitoring and data analysis, conducting clinical validation studies, seeking regulatory approval, and integrating with electronic health record systems. These efforts will contribute to the development of more accurate, reliable, and accessible tools for monitoring fetal health and ensuring the well-being of expectant mothers.

References:

[1] *E. M. Mcclure Et Al., "Stillbirth Rates In Low-Middle Income Countries 2010 - 2013: A Population-Based, Multi-Country Study From The Global Network," Reprod Health, Vol. 12, No. S2, P. S7, Dec. 2015, Doi: 10.1186/1742-4755-12-S2-S7.*

[2] *I. Jehan Et Al., "Stillbirths In An Urban Community In Pakistan," Am J Obstet Gynecol, Vol. 197, No. 3, Pp. 257.E1-257.E8, Sep. 2007, Doi: 10.1016/J.Ajog.2007.07.012.*

[3] *T. R. Nelson1 And D. H. Pretorius, "The Doppler Signal: Where Does It Come From And What Does It Mean?," 1988. [Online]. Available: Www.Ajronline.Org*

[4] *S. A. Alnuaimi, S. Jimaa, And A. H. Khandoker, "Fetal Cardiac Doppler Signal Processing Techniques: Challenges And Future Research Directions," Frontiers In Bioengineering And Biotechnology, Vol. 5. Frontiers Media S.A., Dec. 01, 2017. Doi: 10.3389/Fbioe.2017.00082.*

[5] *P. Hamelmann Et Al., "Doppler Ultrasound Technology For Fetal Heart Rate Monitoring: A Review," Ieee Transactions On Ultrasonics, Ferroelectrics, And Frequency Control, Vol. 67, No. 2. Institute Of Electrical And Electronics Engineers Inc., Pp. 226–238, Feb. 01, 2020. Doi: 10.1109/Tuffc.2019.2943626.*

[6] *A. M. Vintzileos And J. C. Smulian, "Decelerations, Tachycardia, And Decreased Variability: Have We Overlooked The Significance Of Longitudinal Fetal Heart Rate Changes For Detecting Intrapartum Fetal Hypoxia?," Am J Obstet Gynecol, Vol. 215, No. 3, Pp. 261–264, Sep. 2016, Doi: 10.1016/J.Ajog.2016.05.046.*

[7] *J. C. Grimwade, D. W. Walker, M. Bartlett, S. Gordon, And C. Wood, "Human Fetal Heart Rate Change And Movement In Response To Sound And Vibration,"*

Am J Obstet Gynecol, Vol. 109, No. 1, Pp. 86–90, Jan. 1971, Doi: 10.1016/0002-9378(71)90839-8.

[8] A. Fina, A. L. Year De-Sub, C. T. Repo, A. K. Bbas, And A. L. Engin, "De Esign N Ba N E Mecha N And Hear D Dev Rt Ra Colleg Rical A L Engi Velop Ate M San Mum Mmad A Mmar Ab Ge Of And Ineeri Pment Onit Fetal L."

[9] B. A. Kamala Et Al., "Intrapartum Fetal Heart Rate Monitoring Using A Handheld Doppler Versus Pinard Stethoscope: A Randomized Controlled Study In Dar Es Salaam," Int J Womens Health, Vol. Volume 10, Pp. 341–348, Jul. 2018, Doi: 10.2147/Ijwh.S160675.

[10] R. Shimabukuro, K. Takase, S. Ohde, And I. Kusakawa, "Handheld Fetal Doppler Device For Assessing Heart Rate In Neonatal Resuscitation," Pediatrics International, Vol. 59, No. 10, Pp. 1069–1073, Oct. 2017, Doi: 10.1111/Ped.13374.

[11] L. Omo-Aghoja, "Maternal And Fetal Acid-Base Chemistry: A Major Determinant Of Perinatal Outcome," Ann Med Health Sci Res, Vol. 4, No. 1, P. 8, 2014, Doi: 10.4103/2141-9248.126602.

[12] K. K. Shung, J. M. Cannata, And Q. F. Zhou, "Piezoelectric Materials For High Frequency Medical Imaging Applications: A Review," J Electroceram, Vol. 19, No. 1, Pp. 141–147, Oct. 2007, Doi: 10.1007/S10832-007-9044-3.

[13] Advanced Piezoelectric Materials. Elsevier, 2017. Doi: 10.1016/C2015-0-01989-X.

[14] W. J. Merz, "Piezoelectric Ceramics," Nature, Vol. 236, No. 5344, Pp. 245–245, Mar. 1972, Doi: 10.1038/236245a0.

[15] "Piezoelectric_Materials_And_Devices" S.C. Abrahams, K. Nassau, In Concise Encyclopedia Of Advanced Ceramic Materials, 1991.

[16] *J. Kolarik, M. Golembiovsky, T. Docekal, R. Kahankova, R. Martinek, And M. Prauzek, "A Low-Cost Device For Fetal Heart Rate Measurement," Elsevier B.V., Jan. 2018, Pp. 426–431. Doi: 10.1016/J.Ifacol.2018.07.116.*

[17] *M. Dai, X. Chen, K. Zhan, H. Lin, S. Li, And S. Chen, "Design Of A Novel Portable Fetal Cardiac Detection System," 2016.*

[18] *N. Amira, N. Binti, M. Shabry, O. P. Singh, And M. B. Malarvili, "Home Based Fetal Heart Rate Monitor," 2017. [Online]. Available: Http://Www.Ripublication.Com*

[19] *C. Liu, J. Yao, C. Ji, H. Huang, And X. Lin, "Study And Implementation Of A Signal Processing Algorithm For Doppler Ultrasound Fetal Heart Signals".*

[20] *C. Liu, J. Yao, C. Ji, H. Huang, And X. Lin, "Study And Implementation Of A Signal Processing Algorithm For Doppler Ultrasound Fetal Heart Signals".*

[21] *"Getting Started - Edge Impulse Documentation." Https://Docs.Edgeimpulse.Com/Docs/ (Accessed Jun. 03, 2023).*

[22] *L. Pullagura, M. Rao Dontha, And S. Kakumanu, "Recognition Of Fetal Heart Diseases Through Machine Learning Techniques," Ann Rom Soc Cell Biol, Vol. 25, No. 6, Pp. 2601–2615, May 2021, Accessed: May 27, 2023. [Online]. Available: Https://Www.Annalsofrscb.Ro/Index.Php/Journal/Article/View/5873*