Breathing Rate Monitor

Andres F. Lasso, Anna C. Welton, Tyler J. Gazda and Yusef A. Ibrahim

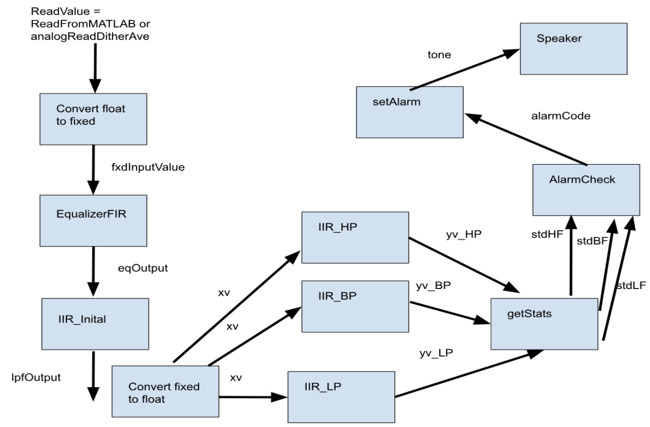
***Abstract*-- In this project, knowledge and procedures from labs completed throughout the semester have been pulled together to create a functioning breathing rate monitor to detect and warn potential acute respiratory infection, also known as pneumonia, in children ages 11 months to 5 years old. The designed breathing dection system uses Butterworth low-pass, high-pass, and band-pass filters to separate signals. Functionality testing was performed with real time breathing data.**

I. Introduction

Young children are most susceptible to a myriad of health problems from all angles. One of the most worrying are breathing issues. An estimate of 120 million episodes of pneumonia occurs in children under 5 years of age across the globe annually. A preemptive measure to detect symptoms of pneumonia early on is via a breathing rate monitor. This measuring system is the least invasive and easiest for the children to undertake since other procedures involve physical testing through a doctor visit. An example such as imaging testing is a more thorough but the most shocking procedure for the children. This testing process includes x-rays or CT scans. Laboratory tests are also used to diagnose pneumonia, these tests include blood tests, mucus tests, or extracting fluid from the pleural space around the lungs to identify bacteria or fungi. Reading through these tests, it is clear that a breathing monitor is the easiest way to preemptively detect pneumonia without causing any mental or physical trauma on young children.

To create a breathing rate monitor, this project was constricted to using only the following materials: a low cost 8-bit microprocessor, a 10 bit A/D converter, and 2k RAM memory, resistors provided in the lab, and a simple buzzer speaker.

This project demonstrates that careful signal conditioning, filter bank design, and statistical energy detection can enable real-time classification of pediatric breathing patterns on low-resource embedded systems despite significant noise and hardware limitations



Code Diagram

Fig. 1

The respiration monitoring is performed via nasal/oral airflow and relies on detecting temperature changes between inhaled and exhaled air. The system is constructed in the following way:The temperature sensor reads data and outputs voltage linearly proportional to temperature to the ADC 10 bit converter that converts this analog voltage to digital counts. This digital signal is then averaged to reduce the noise in the system and finally it is scaled to convert the ADC counts into approximately accurate temperatures. After this the signal is passed and compared through three filters. A low-pass, band-pass, and high-pass filter that serve as our detection for the different frequencies of breathing. These filters were chosen to effectively isolate different breathing rate ranges. This separation allowed for precise classification of breathing states while minimizing the impact of noise and signal overlap. Lastly these frequencies are compared with the decision logic that detects breathing events and triggers alarms for abnormal breathing.

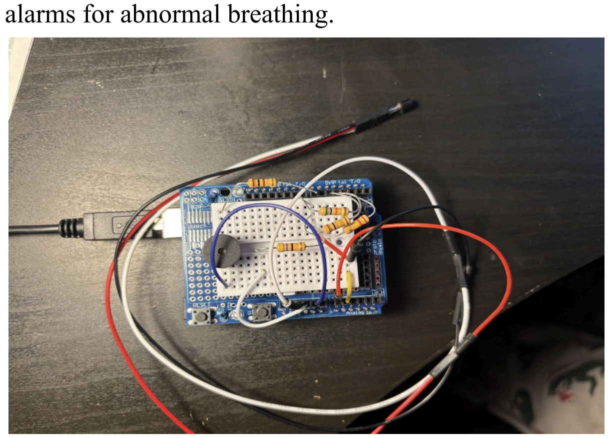


Fig. 2 Complete breathing rate system on arduino uno board

II. Tests

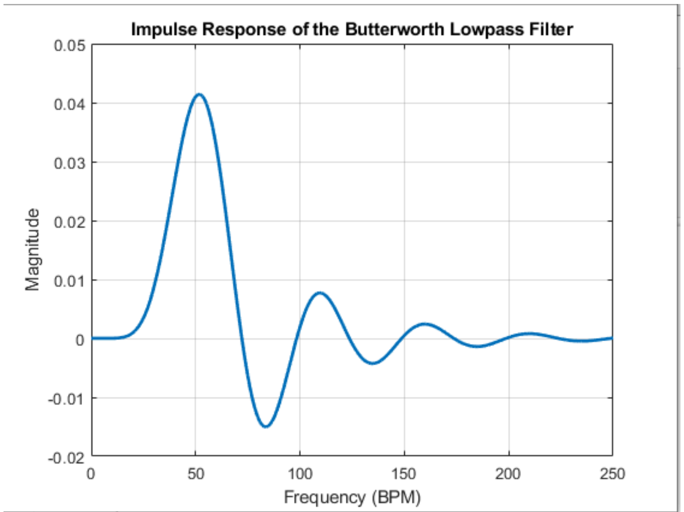


Fig. 3

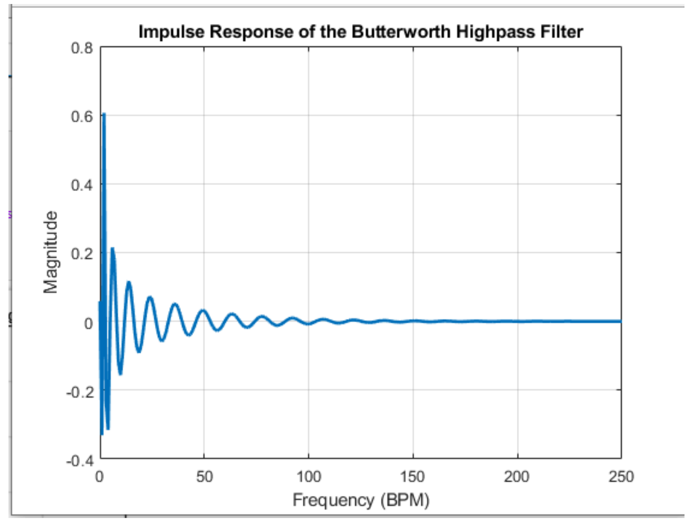


Fig. 5

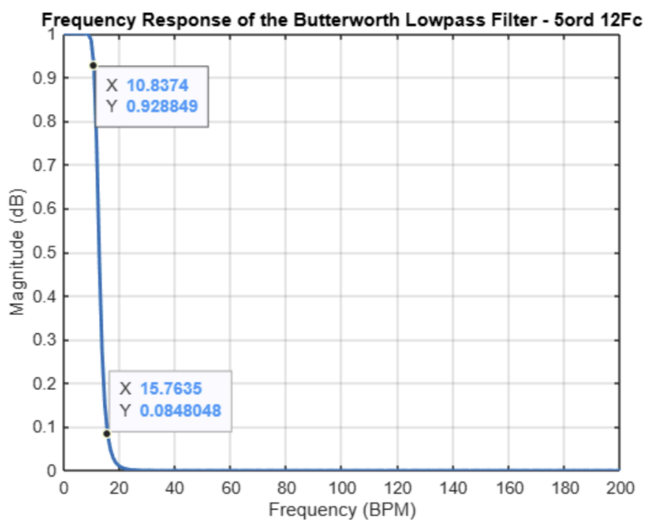


Fig. 4

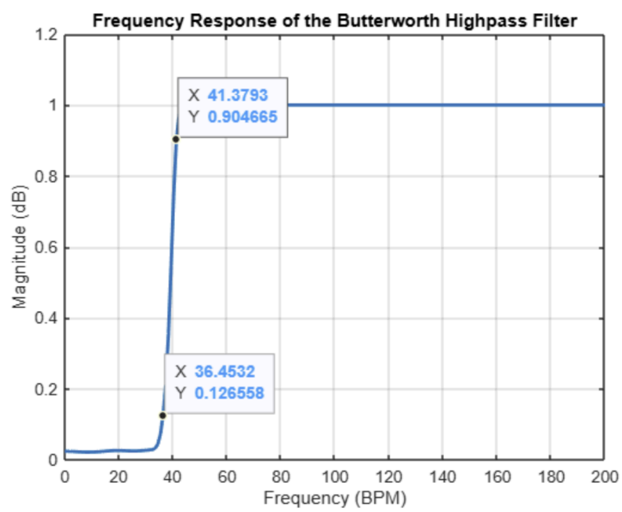


Fig. 6

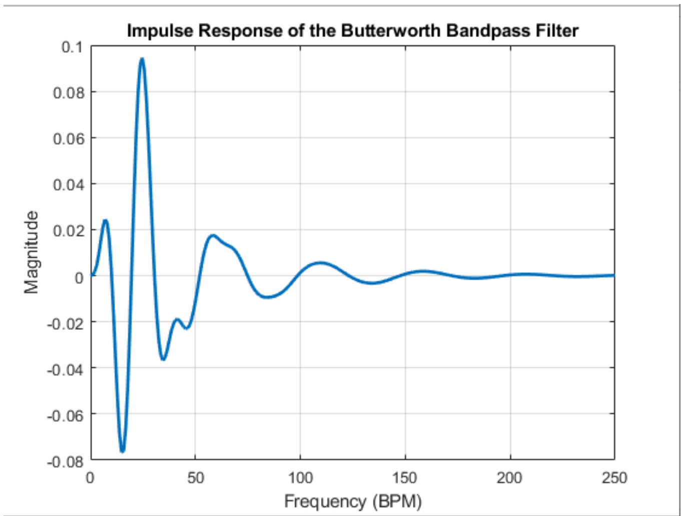


Fig. 7

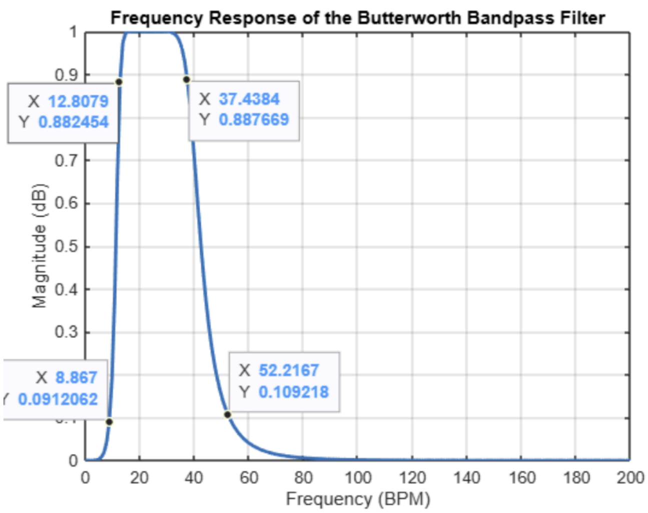


Fig. 8

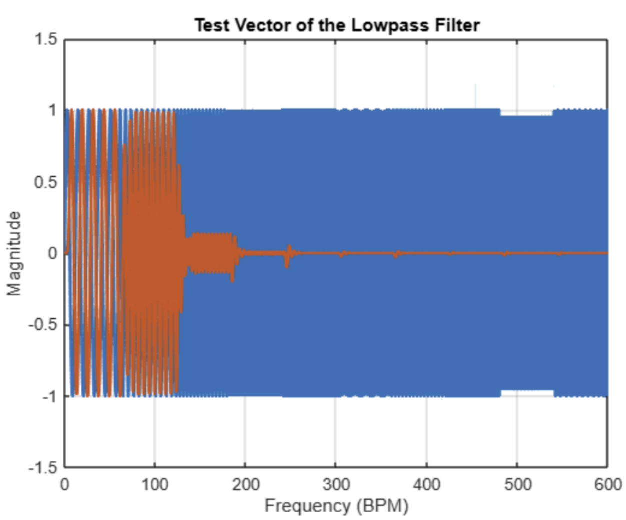


Fig. 9

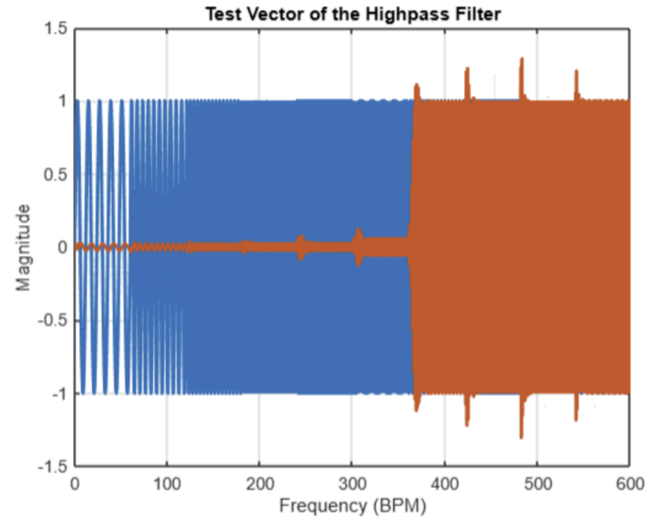


Fig. 10

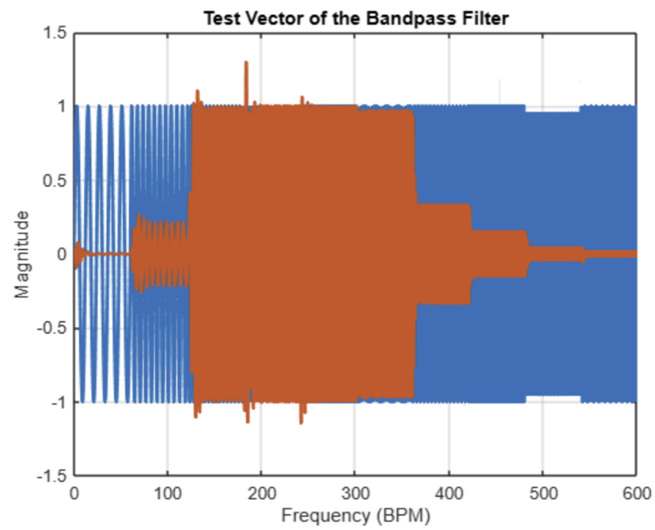


Fig. 11

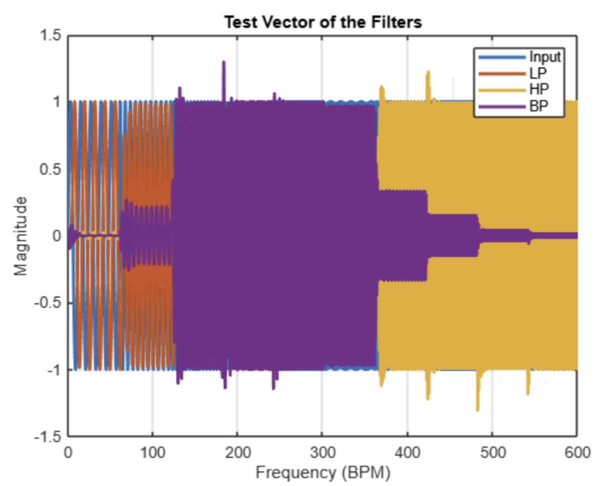


Fig. 12

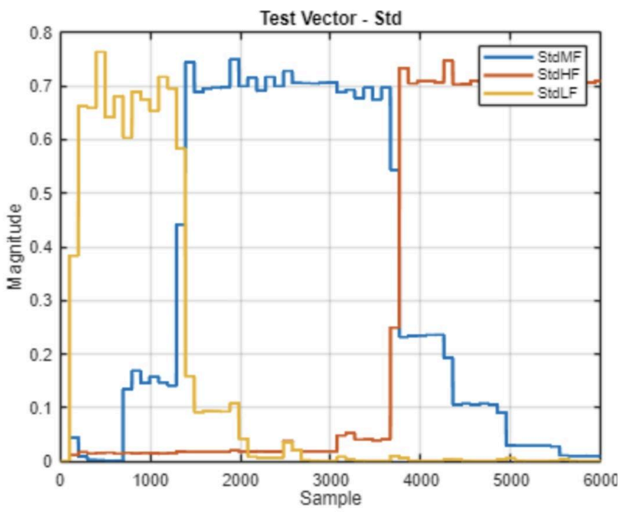


Fig. 13

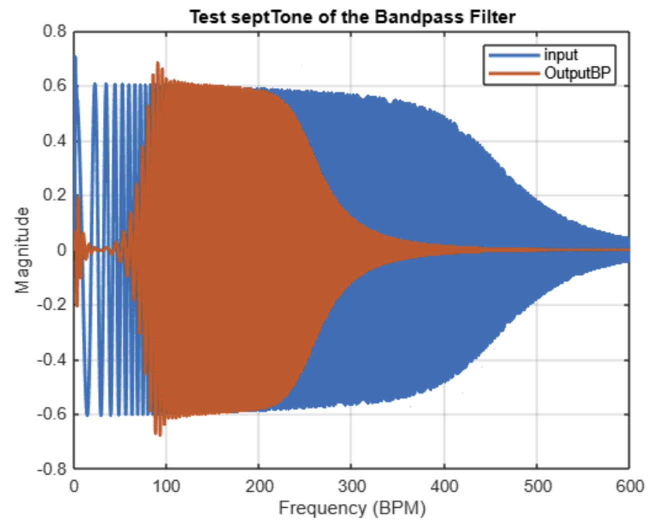


Fig. 16

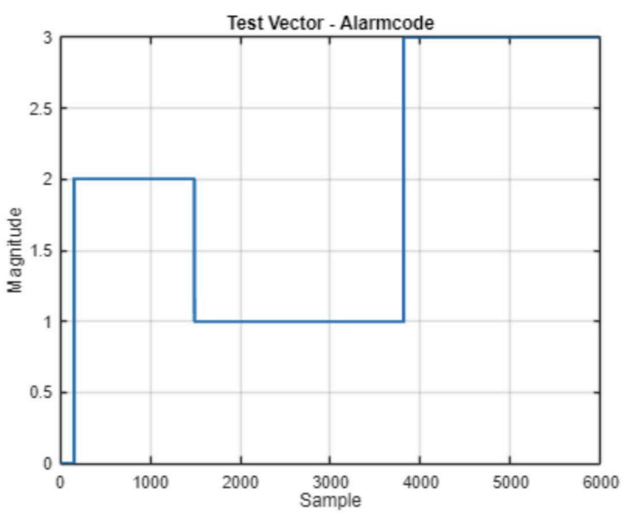


Fig. 14

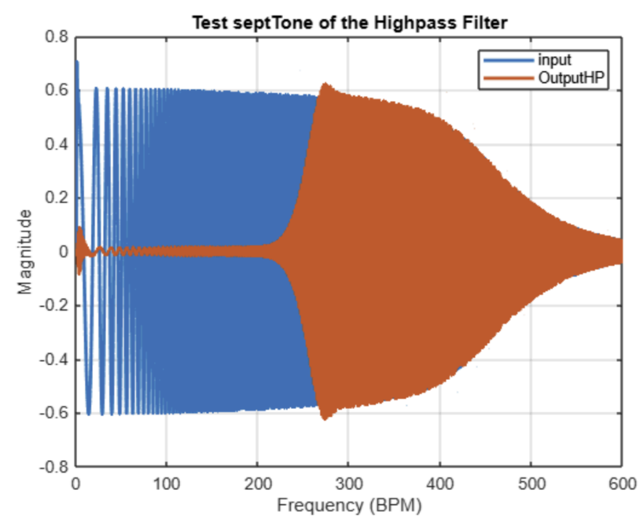


Fig. 17

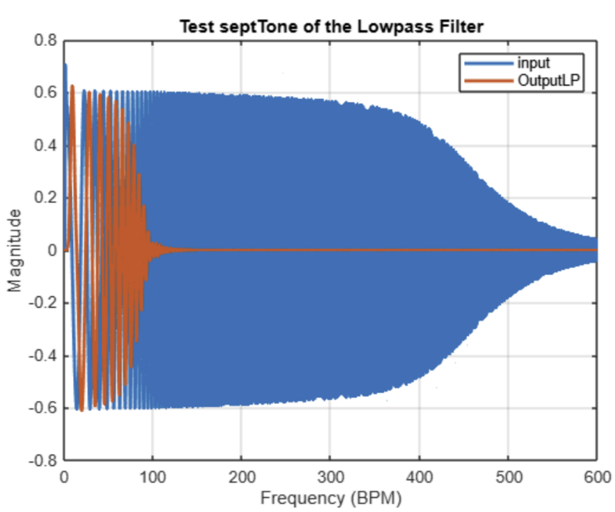


Fig. 15

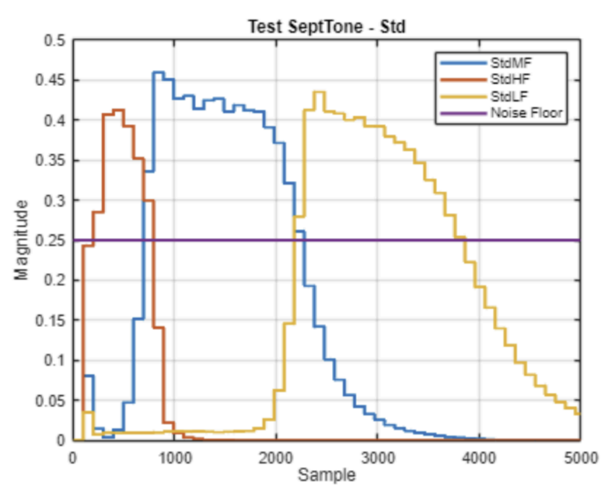


Fig. 18

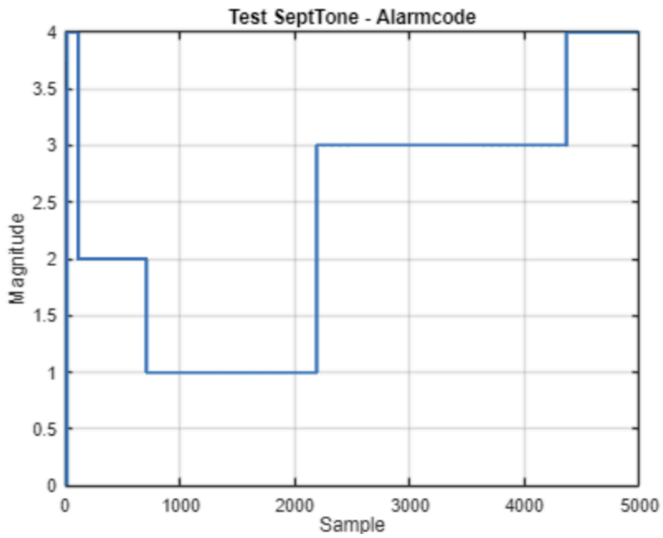


Fig. 19

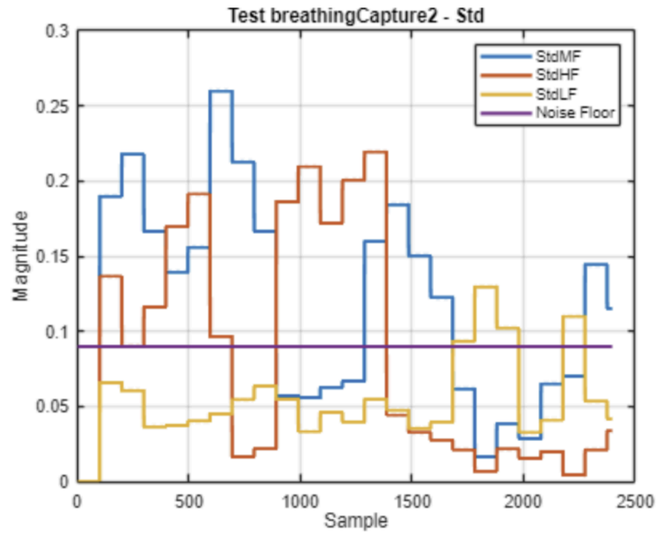


Fig. 22

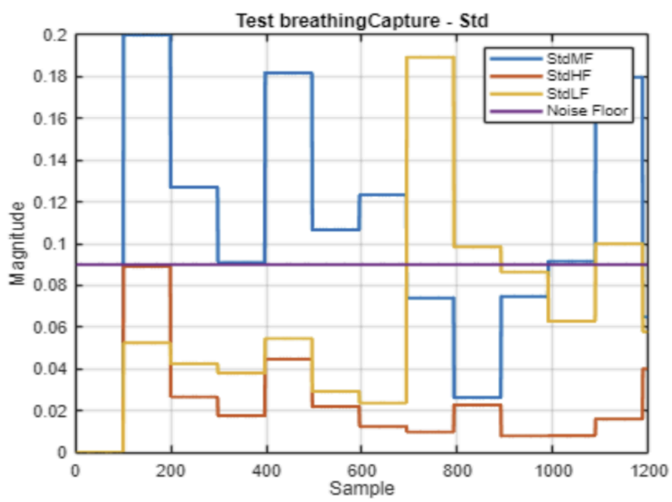


Fig. 20

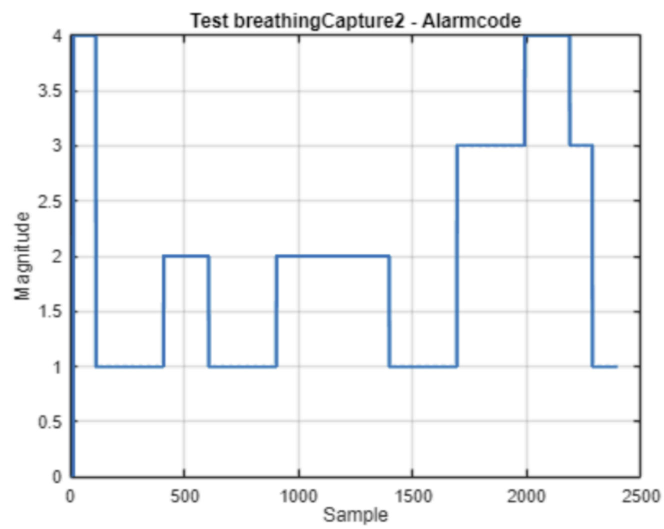


Fig. 23

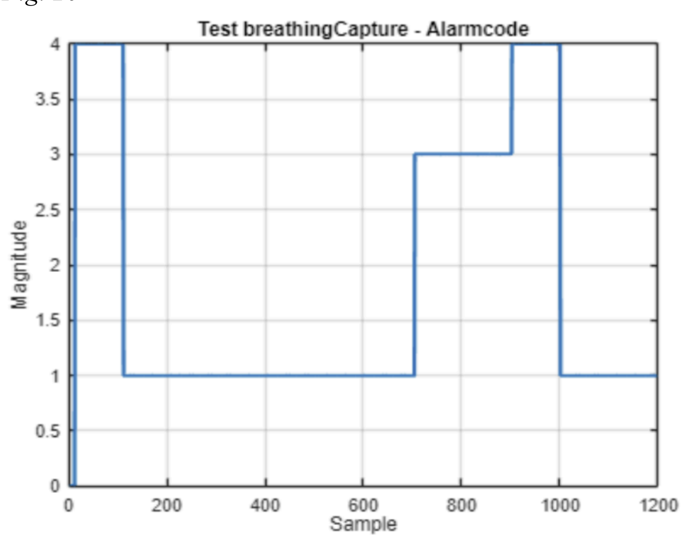


Fig. 21

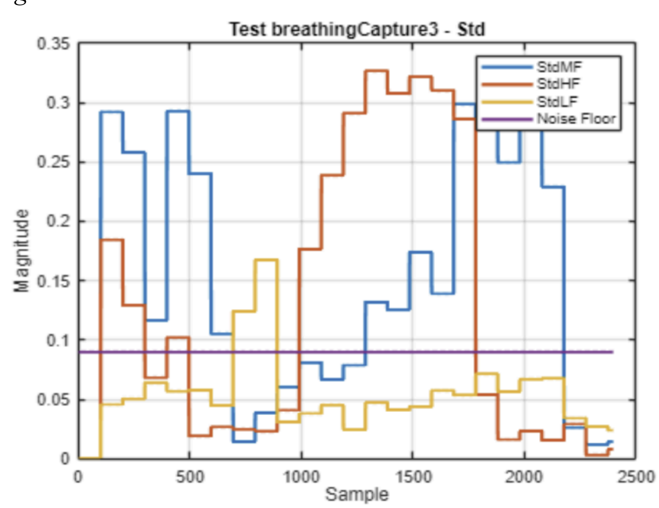


Fig. 24

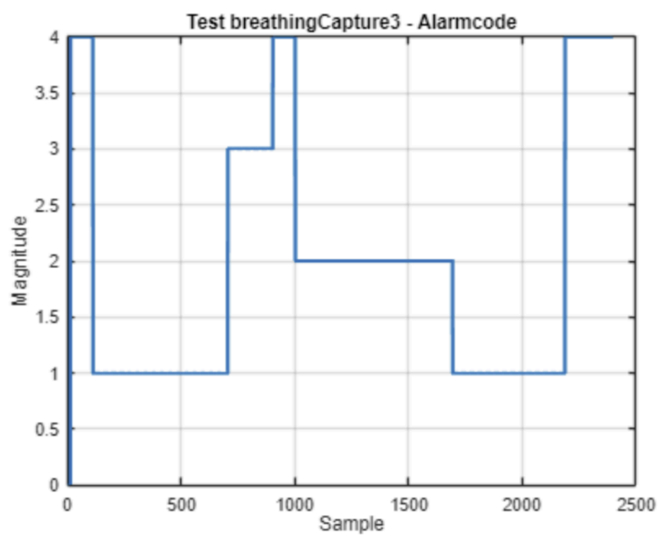


Fig. 25

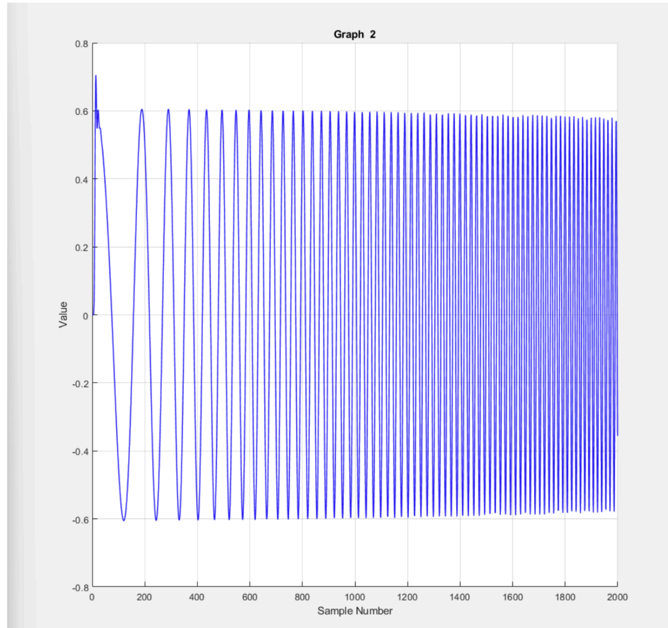


Fig. 27: xv (data after preprocessing)

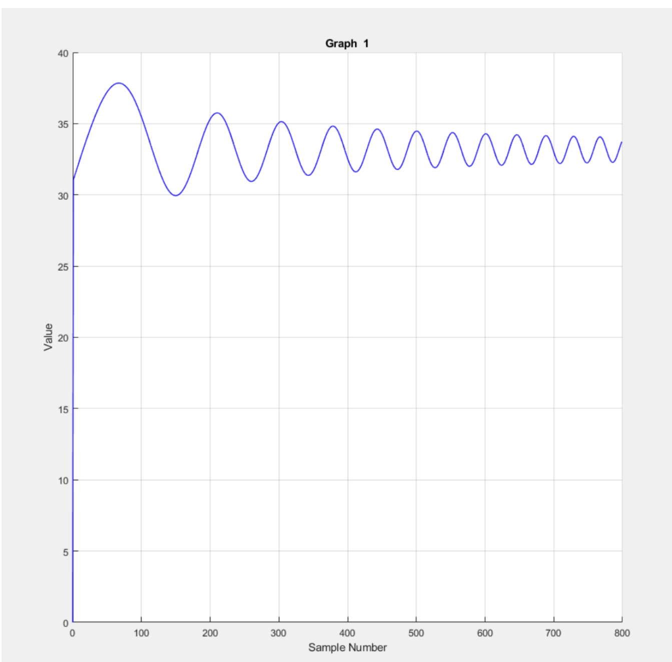


Fig. 26: readvalue(raw data)

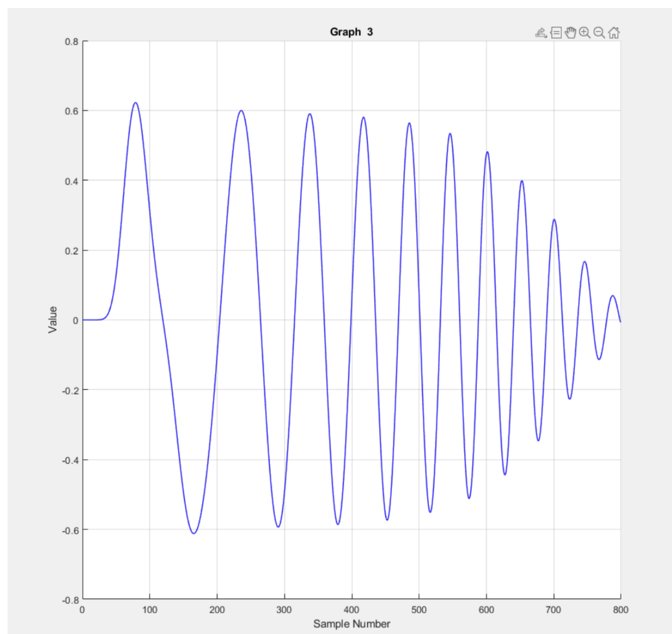


Fig. 28: yv_LP (LF Output)

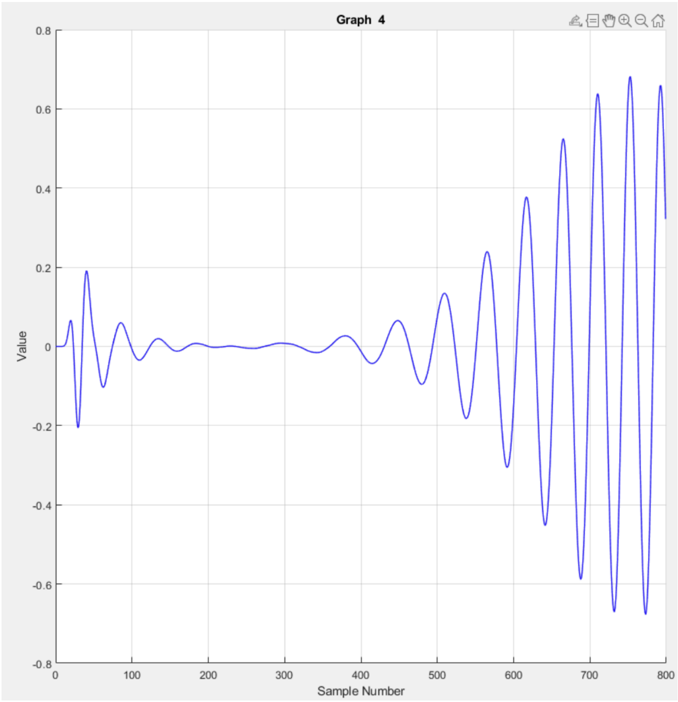


Fig. 29: yv_BP (BP Output)

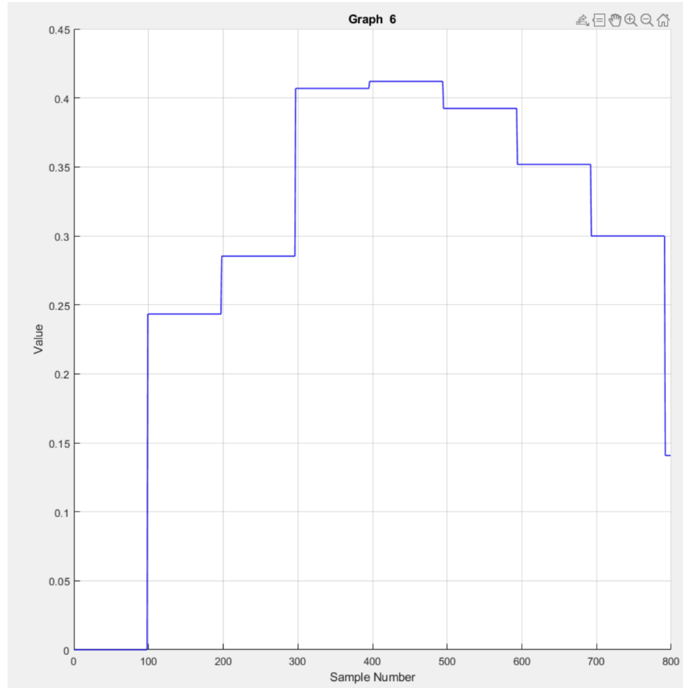


Fig. 31: stdLF (standard deviation of LF)

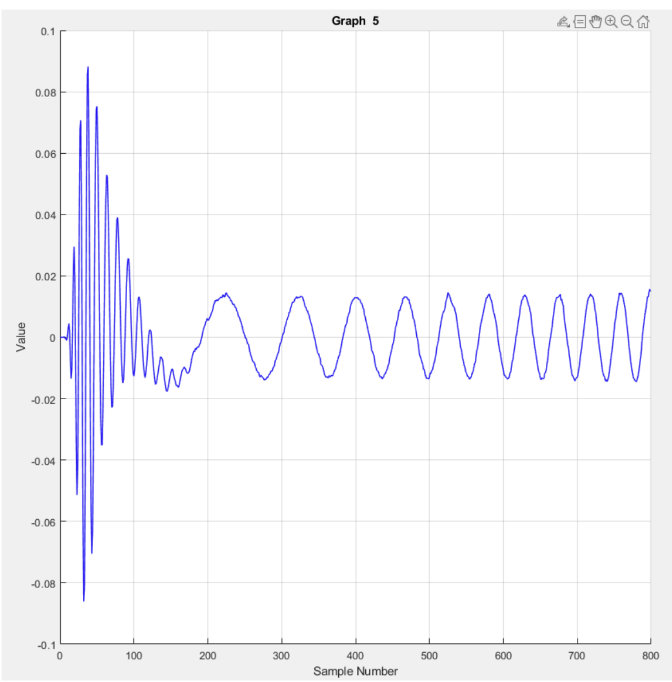


Fig. 30: yv_HP (HF Output)

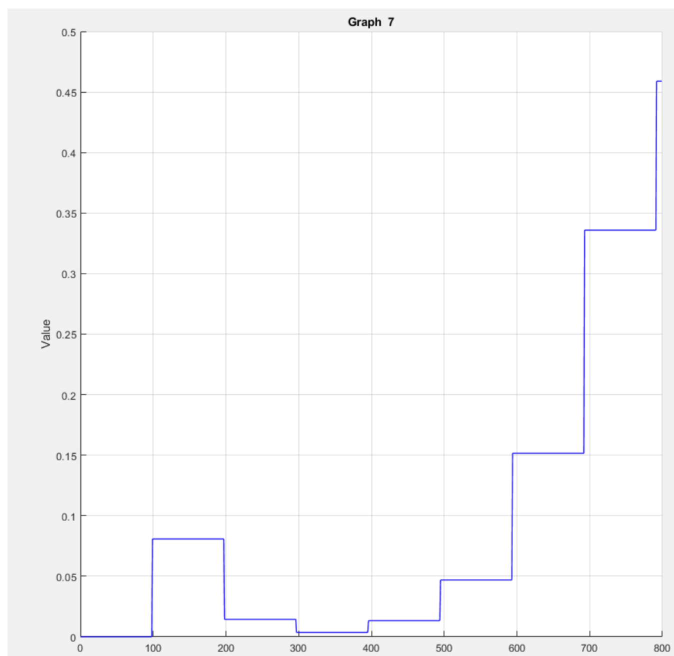


Fig. 32: stdMF (standard deviation of MF)

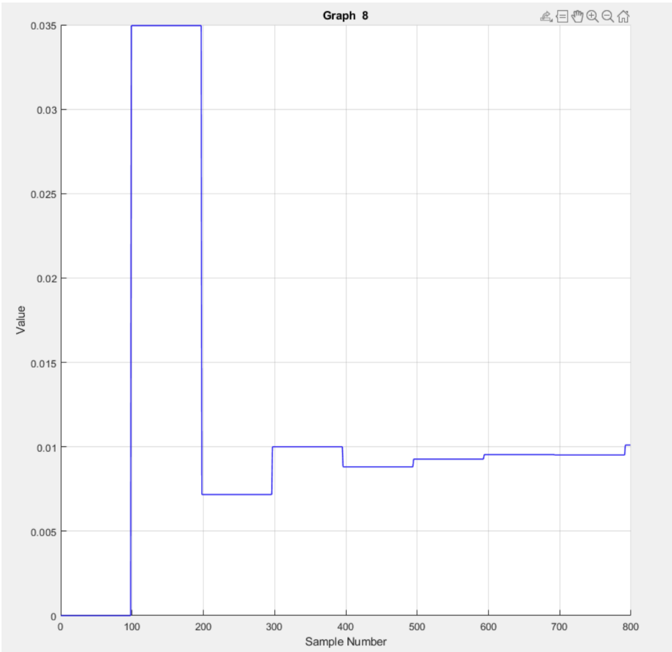


fig. 33: stdHF (standard deviation of HF)

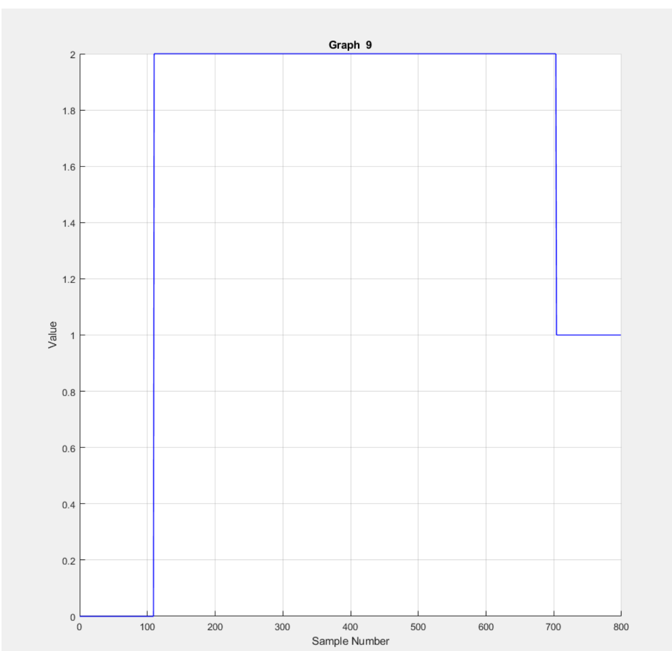


fig. 34: alarmCode (sets the tone)

III. Analysis

The breathing monitor system was evaluated using both synthetic test vectors provided and real world sensor data. From testing the initial plot displays a noisy breathing signal sampled at 50hz over 60 seconds. The breathing frequency becomes hard to distinguish due to the high-frequency noise. Applying an equalizer compensates for the natural low-pass effect of the thermistor-based temperature sensor, stabilizing the baseline while keeping delay or distortion out of the system.

The implemented digital filter bank successfully decomposed the signal into distinct frequency bands associated with different breathing patterns. Low pass filters isolated very slow changes which triggered an alarm if insufficient breathing was detected over a given window, potential respiratory depression. The band pass filter targeted regular breathing between 12-40 Beats per minute. Lastly high pass filters captured fast breathing events like hyperventilation and stress induced breathing.

Each filter's output is monitored continuously by a running standard deviation function. This processes a window of 100 samples to compute the mean, variance, and standard deviation of the signal. This is to relate standard deviation to energy in specific frequency bands serving as the basis for breathing rate classification. The logic successfully compares the energy levels from each band to determine the dominant breathing pattern and if a standard deviation of one band dominates significantly over the others

By designing and cascading these filtering stages, the system was able to cleanly separate signals relevant for downstream breathing detection logic, despite the presence of significant quantization noise and low amplitude temperature fluctuations. Overall, the designed system uses 73% of the memory on the Arduino Uno board, leaving some room for additional calculations if necessary. It takes the system 52,572ms (52 seconds) for the code to detect a signal and flag the alarm to trigger the speaker to notify of a breathing event change.

The alarm device is set to operate in 5 states: waiting, normal breathing which doesn't have an associated tone, low breathing associated with a 400 Hz continuous tone, high breathing associated with a 1 kHz on/off toggle tone, lastly a fail/noise state associated with a 200 Hz fail tone.

Issues found during the analysis section include adding the initial IIR filter since without it an alarm was flagged due to noise. This initial IIR filter was too high which slowed down all code in general. When testing code the warning section was tested with both "&&" and "||" with the latter breaking the code as its values dropped below the noise flow after filter attenuation ending in reading only noise. Mentioning the noise floor, when testing with sensor data it was too large to also be used for breathing data, which initially was the process taken. Presenting a problem due to the size of the noise floor which reads all breathing data as noise. Additionally, the "captureArduino" function was being used in an outdated version which continuously resulted in terminal errors, making sure to always update and revise that all versions of your working code are up to date is extremely important. Lastly the high frequency alarm toggle wasn't set as a static so its value got reset to 0 after an alarm call which made the tone toggle trigger every 100ms rather than the desired 1 second.

IV. Conclusion

The project successfully showcases the design and deployment of a breathing rate monitor capable of classifying pediatric respiratory behavior into low, normal, and high-frequency bands. After signal conditioning, a bank of carefully tuned filters, and real-time statistical monitoring, the system achieved reliable detection of abnormal breathing events, including signs of respiratory depression and distress.

Despite hardware limitations such as low ADC resolution and limited ram the project overcame these constraints through efficient fixed-point arithmetic, careful resource allocation, and modular code design. Using 100 samples for the standard deviation window enabled rapid and robust detection while identifying false positives.

After testing the system consistently showing the appropriate alarms and responses to breathing rate shifts which were validated through both frequency sweep test vectors and real sensor recordings.

Applying the course teachings with the use of several filters averaging and other noise reduction techniques, this group was able to correctly identify and alarm a potential patient of irregularity within their breathing patterns. This approach emphasizes how signal processing, even when basic, can achieve clinically meaningful outcomes when paired with a sound understanding of the system limitations.