Hear the Rhythm Digital Stethoscope

EE 403W Senior Design Project: Embedded Audio Engineering
College of Electrical Engineering and Computer Science
The Pennsylvania State University
University Park
Spring 2017

Donald Hall, Matthew McTaggart, and Matthew Mox

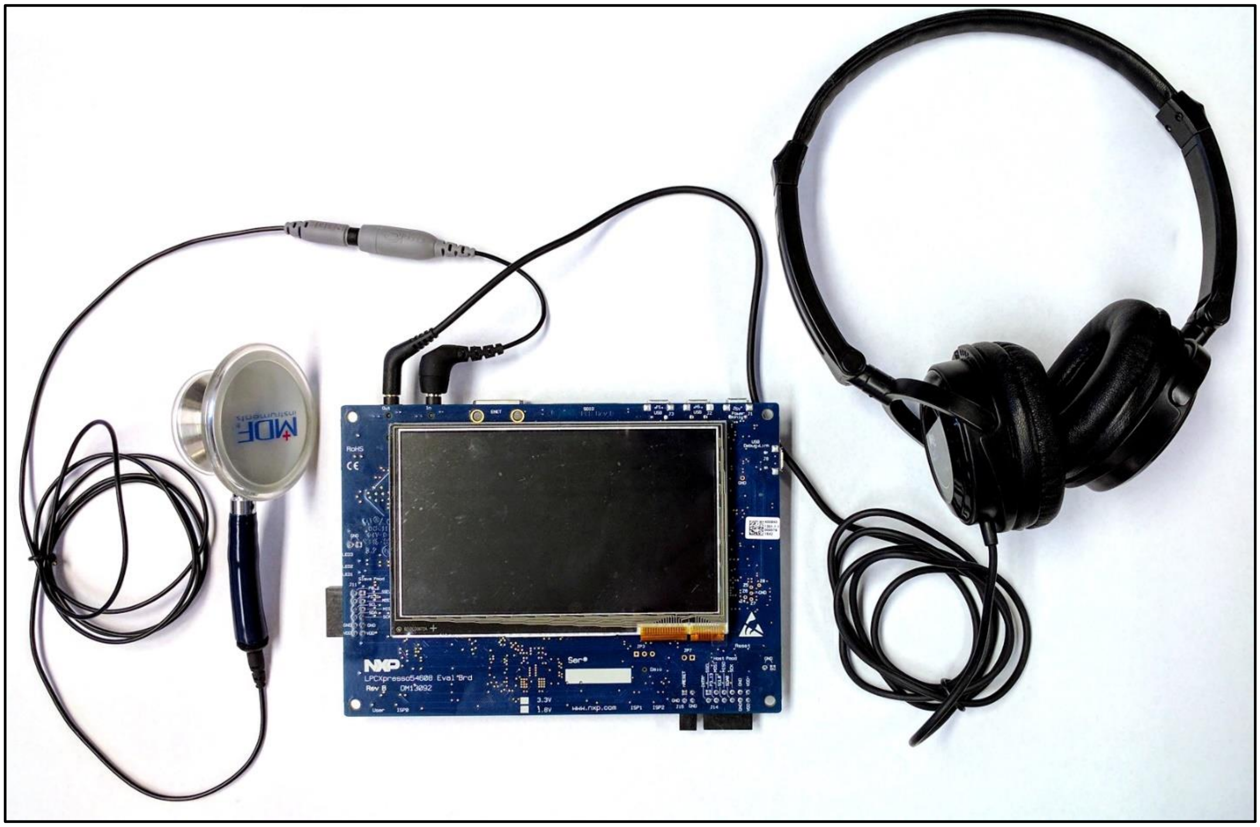

Table of Contents

List of Figures

Executive Summary

Digital stethoscopes are an emerging technology in the medical industry. The desire for an electronic device to capture the critical components of a patient's heart motivated the purpose of this project. According to [1] and [2], common problems that are faced by traditional acoustic stethoscopes are difficulties diagnosing common heart problems, low-signal level, signal integrity affected by noise, and unavoidable attenuation from stethoscope tubing.

With noise and the aforementioned characteristics of analog stethoscopes being an issue, the implementation of a digital solution leads to more accurate results that are critical in a field reliant on precision. With this motivation, simulated MATLAB trials have been completed and a robust digital stethoscope has been developed for calculating the heart rate of any user in any environment. Upon completion of the simulated signal processing algorithms, the LPCXpresso54608 development board by NXP products was utilized and programmed for generation of a real-time digital stethoscope for analysis of a real-time user heartbeat input. The potential of the device is astronomical in the sense that the creation can simultaneously output a real-time heart rate, display the rhythmic components of the heartbeat, amplify low-level signals, and marginally reduce ambient noise levels at a much cheaper cost than the digital stethoscopes currently on the market. The current price of a digital stethoscope can vary from $400 to upwards of $1000. With this being the current value of the market, the price of the Hear the Rhythm Digital Stethoscope is competitive at $245.15.

Problem Statement

The medical industry requires devices designed for precision, speed, and efficiency. Even though this is the case, acoustic stethoscopes are still utilized for collection of vital heart sounds. Common problems being faced by the traditional stethoscope are difficulties diagnosing heart problems, low-signal level, signal integrity affected by noise, and unavoidable attenuation from stethoscope tubing. With the upward trend of technology in the 21st century, the development of a digital stethoscope can replace the traditional stethoscope effectively.

Proposed Solution

The implementation of the digital stethoscope started by the development of various signal processing algorithms. Upon derivation of the algorithms, simulations were generated in MATLAB and Simulink software to prove that the realization of a digital stethoscope was feasible. After completing the simulations of the signal processing algorithms, the same algorithms were then evaluated and programmed in Keil µVision 5 (C language). The Hear the Rhythm Digital Stethoscope can simultaneously output a real-time heart rate, display the rhythmic components of the heartbeat, amplify low-level signals, and marginally reduce ambient noise levels at a much cheaper price than the digital stethoscopes currently on the market. The thorough details of the simulations and development of the digital stethoscope device can be evaluated throughout the implementation details section. The following figure shows the block diagram of the Hear the Rhythm Digital Stethoscope. The microphone is placed at the resonator which amplifies the vibration from the diaphragm that is in contact on the user.

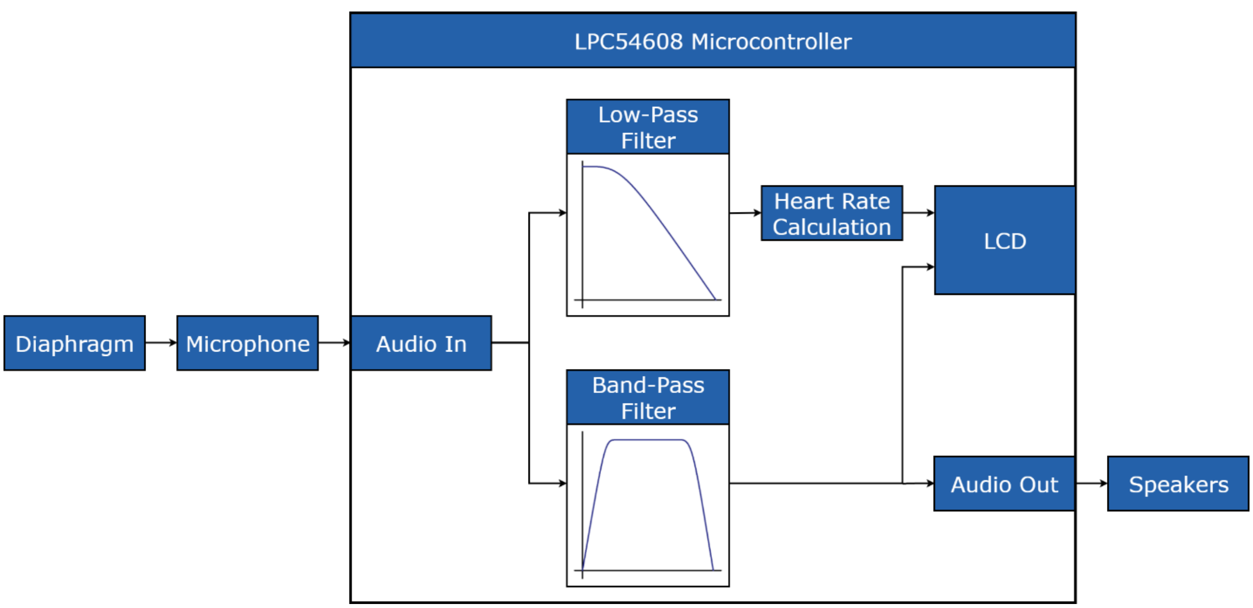

Figure 1: Block Diagram of Digital Stethoscope Hardware/Software Implementation

Implementation Details

The implementation of the digital stethoscope started by the development of various signal processing algorithms. Upon derivation of the algorithms, simulations were generated in MATLAB and Simulink software to prove that the realization of a digital stethoscope was feasible, prior to the implementation of the algorithms on a LPC54608 microcontroller. After successfully completing the simulations of the signal processing algorithms, the same algorithms were then evaluated and programmed in Keil µVision 5 (C language). This section evaluates the details of the simulations and development of the Hear the Rhythm Digital Stethoscope.

Analysis of MATLAB and Simulink Simulations

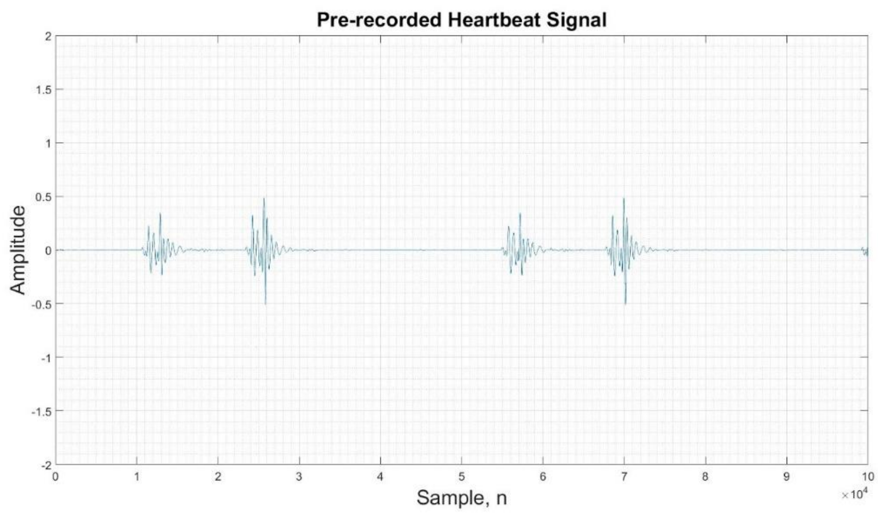

Figure 2: Pre-recorded User Heartbeat Signal

Figure 2 represents the contraction and expansion (neighboring signals) of an ideal heartbeat signal as an input to the digital stethoscope. In an ideal world, the device would receive and output a signal that resembles the qualities of the above heartbeat signal. In reality, the input will have noise with the heartbeat signal buried within. To simulate this, white Gaussian noise was added as shown in **figure 3**. This allowed for the development of a filtering algorithm to attenuate any noise recorded by the device.

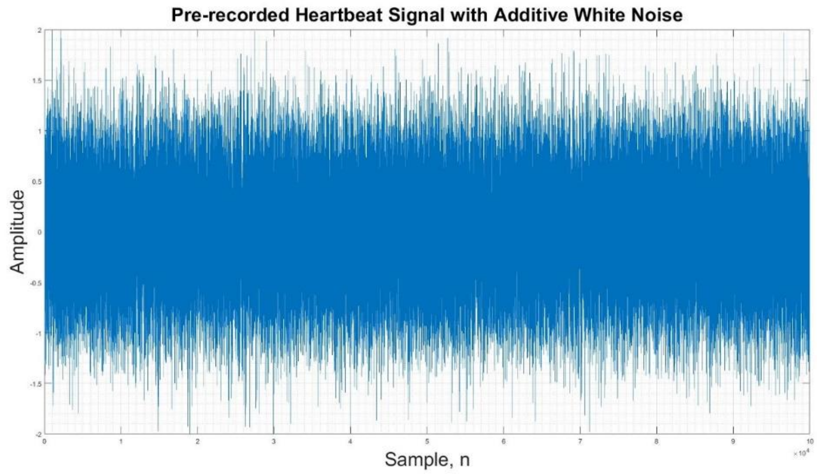

Figure 3: Pre-recorded User Heartbeat Signal with Additive White Gaussian Noise

Through knowledge of the characteristics of a heartbeat signal, the higher frequency pseudo white noise can be attenuated by a low-pass filter. Therefore, a low-pass filter can be used to help extract the desired characteristics from the white Gaussian noise as found in **figure 4** and for peak detection for the heart rate calculation.

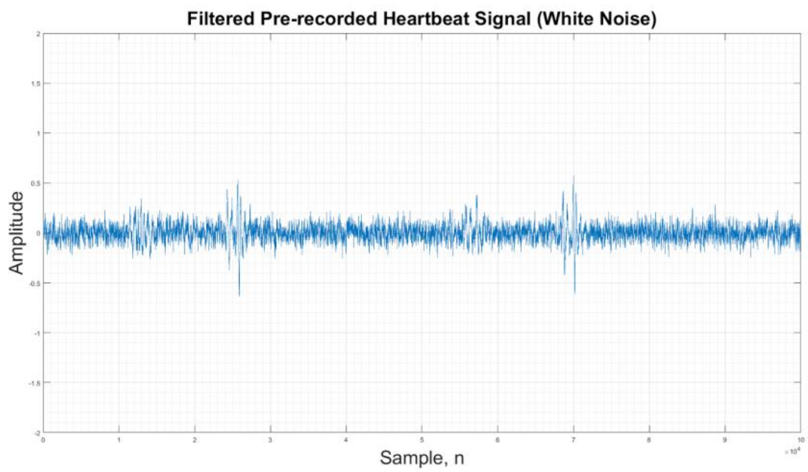

Figure 4: Filtered White Gaussian Noise in Pre-recorded User Heartbeat Signal

The filter output shown in **figure 4** resembles the similar characteristics of the true heartbeat signal. The incorporation of the filter has greatly improved the signal to noise ratio, however, remnants of the white Gaussian noise are still present. The increased signal-to-noise ratio leads to an optimization of peak detection for the digital stethoscope. **Figure 5** and **6** elaborate the effectiveness of a low-pass filter for peak detection. The microphone is placed inside the stethoscope tubing as closely to the resonator and diaphragm. **Figure 5** shows the results collected from the microphone prior to being sent through the filter, using Simulink. The input to the simulation was generated by the diaphragm, that is placed on the user, vibrating within a resonator of a traditional analog stethoscope. A microphone had been placed as close as possible to the resonator to eliminate signal attenuation from the tubing of the traditional analog stethoscope.

Through observation of **figure 5**, the hardware setup of the digital stethoscope realizes the low signal level of the heartbeat. The figure expresses max amplitudes/peaks of the heartbeat signal with the corresponding harmonic components of the heart in the time-domain. After incorporation of an 8^{th} order infinite impulse response low-pass filter of four 2^{nd} order cascaded sections at 175Hz to the user heartbeat signal, the results should reveal a similar observation with a lower noise accompaniment as shown in **figure 6**.

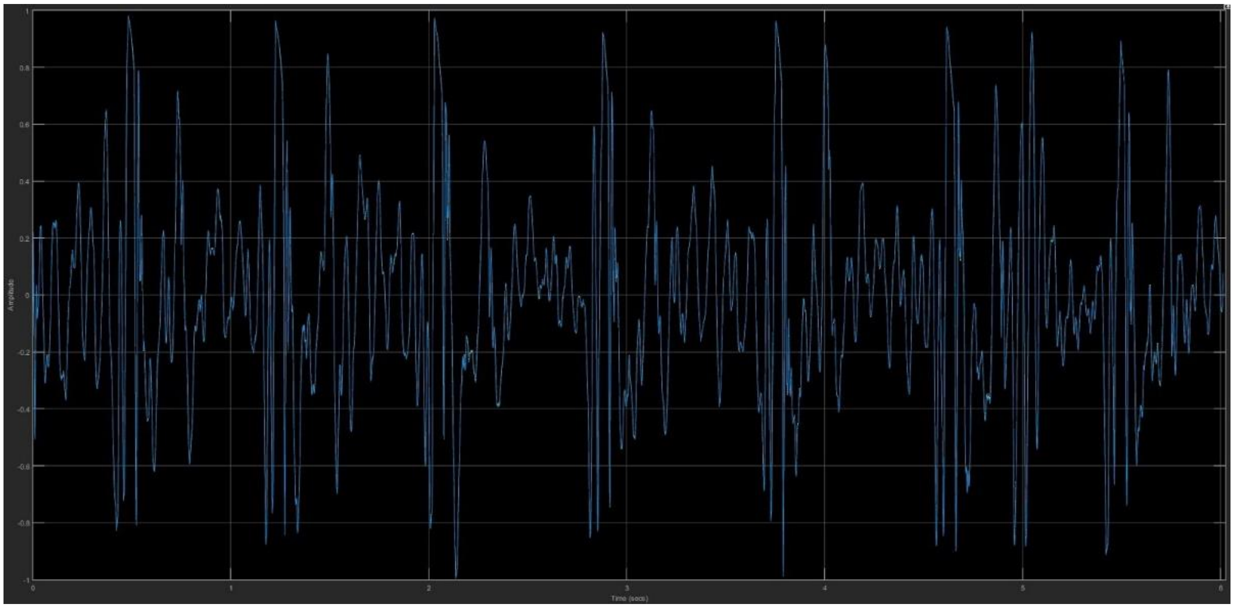

Figure 5: User Heartbeat Input Signal for Low-pass Filter

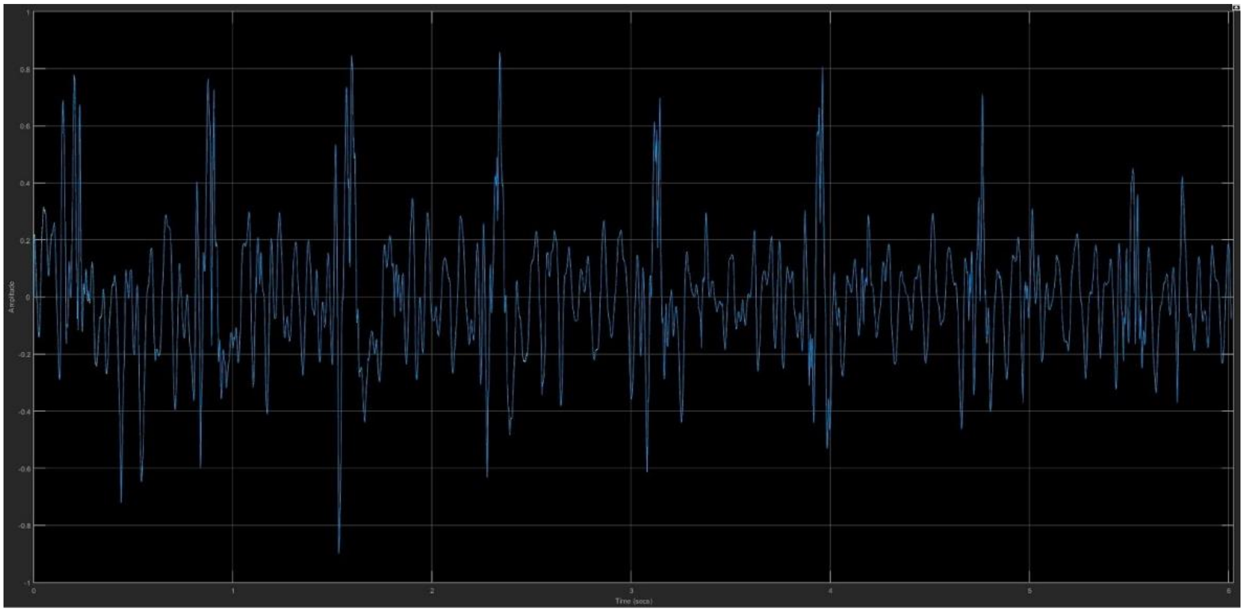

Figure 6: Low-pass Filtered User Heartbeat Output Signal

The results depicted in **figure 6** prove that the filtering algorithm improves peak detection for the heart rate calculation from the user's heartbeat signal. The filter decreases the noisiness of the input heartbeat signal and filters the contraction pulse (larger pulse immediately following the max peaks in **figure 5**). Also, the low-pass filter only attenuates the amplitude of the peaks by 20%. In doing so, the filter can optimize peak in any environment.

The following figures shows the simulation of the band-pass filter for the display of the real-time signal on the LCD and for audio playback. The band-pass filter extracts the heart beat signal but attenuates the signal about 40%. **Figure 7** shows the input signal and **figure 8** shows the band-pass filtered output. The band-pass filter is an 8th order infinite impulse response filter of two cascaded 4th order sections. The center frequency is at 175Hz with a bandwidth of 150Hz. This frequency range is designed to capture all frequency components of the heartbeat.

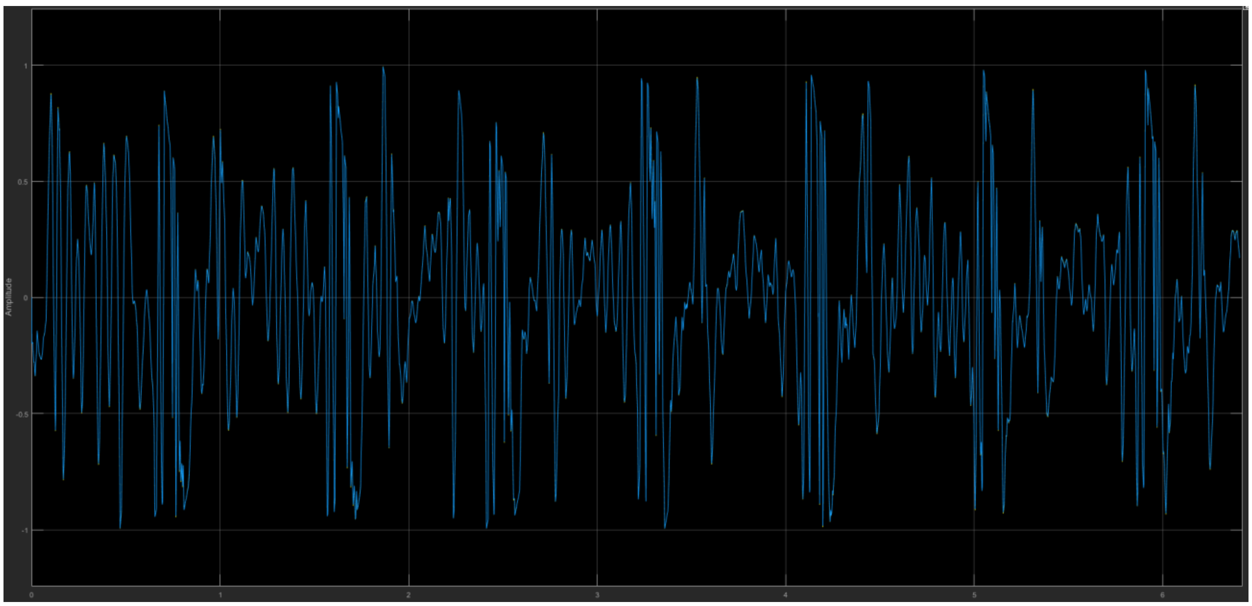

Figure 7: User Heartbeat Input Signal for Band-pass Filter

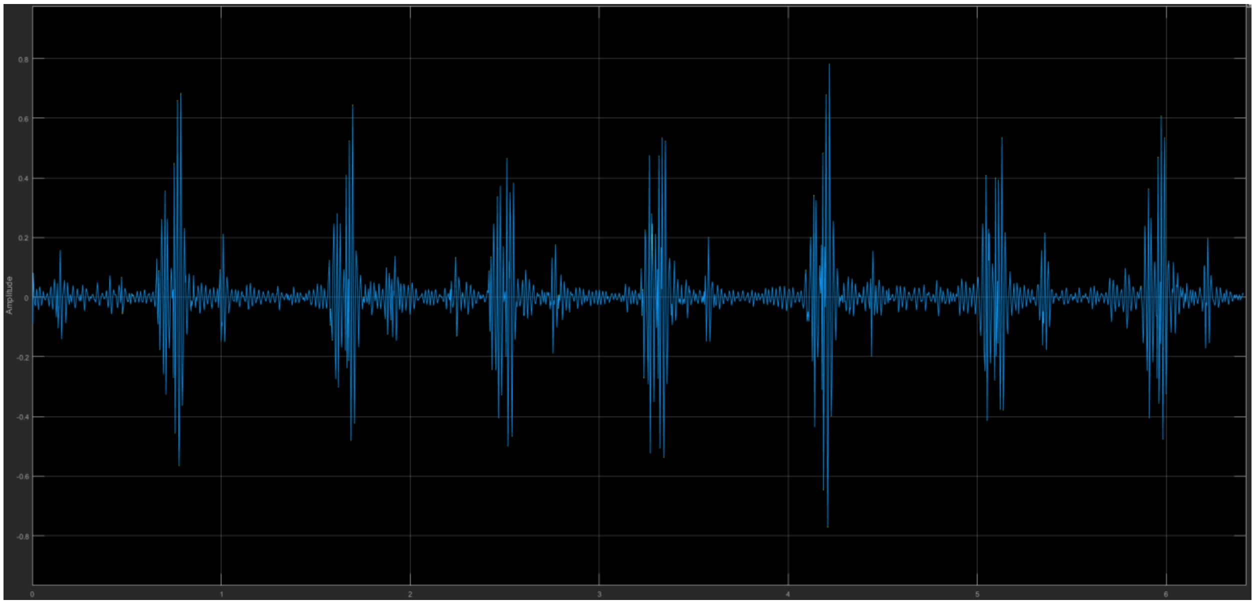

Figure 8: Band-pass Filtered User Heartbeat Output Signal

Implementation on the LPCXpresso54608

The Feel the Rhythm Digital Stethoscope was developed on the LPCXpresso54608 using Keil µVision 5. The following figures shows the code and the development of the digital stethoscope.

```
//LP - Peak Detection
//Cascade 1
InputDataBufferLP1[0]    = (float)(FIFO_Data.Channel[1]);
OutputDataBufferLP1[0]   = (float)(lp1num1*InputDataBufferLP1[0] + lp1num2*InputDataBufferLP1[1] + lp1num3*InputDataBufferLP1[2]
                            - lp1den2*OutputDataBufferLP1[1] - lp1den3*OutputDataBufferLP1[2]);
//Cascade 2
InputDataBufferLP2[0]    = (float)OutputDataBufferLP1[0];
OutputDataBufferLP2[0]   = (float)(lp2num1*InputDataBufferLP2[0] + lp2num2*InputDataBufferLP2[1] + lp2num3*InputDataBufferLP2[2]
                            - lp2den2*OutputDataBufferLP2[1] - lp2den3*OutputDataBufferLP2[2]);
//Cascade 3
InputDataBufferLP3[0]    = (float)OutputDataBufferLP2[0];
OutputDataBufferLP3[0]   = (float)(lp3num1*InputDataBufferLP3[0] + lp3num2*InputDataBufferLP3[1] + lp3num3*InputDataBufferLP3[2]
                            - lp3den2*OutputDataBufferLP3[1] - lp3den3*OutputDataBufferLP3[2]);
//Cascade 4
InputDataBufferLP4[0]    = (float)OutputDataBufferLP3[0];
OutputDataBufferLP4[0]   = (float)(lp4num1*InputDataBufferLP4[0] + lp4num2*InputDataBufferLP4[1] + lp4num3*InputDataBufferLP4[2]
                            - lp4den2*OutputDataBufferLP4[1] - lp4den3*OutputDataBufferLP4[2]);

//HeartRate Signal to use for peak detection
HeartRateLP = (float)OutputDataBufferLP4[0];
```

Figure 9: Low-pass Filtering Algorithm for Optimizing Peak Detection

The low-pass filter shown in **figure 9** is used for optimizing the peak detection threshold implemented in **figure 10** as well as being the sample set for used for the heart rate calculation. The 8^{th} order infinite impulse response incorporates four cascaded 2^{nd} order sections for filtering data.

```
//Threshold Adaptation
if (SampleMax > Threshold)
{
    Threshold += 2000.0f;
}
if (SampleMax <= Threshold){
        Threshold -= 2000.0f;
}
if (Threshold < 2000.0f) {
    Threshold = 2000.0f;
}
```

Figure 10: Thresholding Algorithm for Peak Detection of Heartbeat Waveform

The threshold adaptation algorithm was implemented in order to detect a heartbeat from any user. Given that every person has a different heart, the threshold adapts to each user. The threshold adapts every 4,320 input samples at a sampling rate of 8kHz.

The heart rate is calculated by recording the samples between two peaks above the adaptive threshold. The following **figure 11** shows how the heart rate is calculated. The "waitwidth" variable is used to ignore samples once the first peak is occurring. This is important because with a sample rate of 8kHz, multiple successive samples will be above the threshold.

```
//Peak Detection for Heart Rate Calculation
//Each Heart Rate Calculation is based on the knowledge of the sample rate and the amount of samples between each peak
//Flag asserts "waitwidth" samples after peak has been detected to avoid multiple peak detects on the same recorded heart beat
if ((HeartRateLP > Threshold)&&(flag==0))
{
    if(samplecounter > (waitsamples + waitwidth))
    {
        //Amount of samples between each heart beat peak
        sample1 = sample2;
        sample2 = samplecounter;
        time = sample2-sample1;
        //Shift the data once along the buffer for the next current heart rate calculation can be indexed at 0
        for (int k = DATABUFFER_SIZE-1; k > 0; k--){
            HRDataBuffer[k]=HRDataBuffer[k-1];
        }
        HRDataBuffer[0] = (60*SAMPLERATE/time)+1;
        //Median filter for outlier heart rate calculations
        heartrate = medsort();
        flag = 1;
        waitsamples = samplecounter;
    }
}
//Reset flag for the following heart rate calculation
if((flag==1)&&(HeartRateLP<threshold)){
flag = 0;
}
```

Figure 11: Heart Rate Calculation Algorithm

The median filter shown in **figure 12** is used to generate an accurate heart rate calculation. The median filter sorts the heart rate calculations stored in the "HRDataBuffer" and outputs the middle value. The filter length is set at 11 to remove any outlier calculations without affecting the heart rate output.

```c
//Sorting for Median Filter
int medsort()
{
    int i,j,temp;
    int sortHRData[DATABUFFER_SIZE];
    for(i=0;i<DATABUFFER_SIZE;i++){
        sortHRData[i] = HRDataBuffer[i];
    }

    for(i=0;i<DATABUFFER_SIZE-1;i++) {
        for(j=0;j<DATABUFFER_SIZE-i-1;j++) {
            if(sortHRData[j]>sortHRData[j+1]){
                temp = sortHRData[j];
                sortHRData[j] = sortHRData[j+1];
                sortHRData[j+1] = temp;
            }
        }
    }
    return sortHRData[(DATABUFFER_SIZE-1)/2];
}
```

Figure 12: Median Filtering Algorithm for Heart Rate Calculation

```c
//BP - For LCD and Audio Playback
//BP - Cascade 1
InputDataBufferC1[0]    = (float)(FIFO_Data.Channel[1]);
OutputDataBufferC1[0]   = (float)(c1num1*InputDataBufferC1[0] + c1num2*InputDataBufferC1[1] + c1num3*InputDataBufferC1[2]
                          + c1num4*InputDataBufferC1[3] + c1num5*InputDataBufferC1[4] - c1den2*OutputDataBufferC1[1]
                          - c1den3*OutputDataBufferC1[2] - c1den4*OutputDataBufferC1[3] - c1den5*OutputDataBufferC1[4]);
//BP - Cascade 2
InputDataBufferC2[0]    = (float)OutputDataBufferC1[0];
OutputDataBufferC2[0]   = (float)(c2num1*InputDataBufferC2[0] + c2num2*InputDataBufferC2[1] + c2num3*InputDataBufferC2[2]
                          + c2num4*InputDataBufferC2[3] + c2num5*InputDataBufferC2[4] - c2den2*OutputDataBufferC2[1]
                          - c2den3*OutputDataBufferC2[2] - c2den4*OutputDataBufferC2[3] - c2den5*OutputDataBufferC2[4]);

//RightChannel is used for the display on the LCD and for audio playback
HeartRateDispPlay = (float)OutputDataBufferC2[0];

//NextSampleOut constructs the processed signal on both Left and Right Channels for audio playback.
NextSampleOut = (uint16_t)OutputDataBufferC2[0];
NextSampleOut = ((uint32_t)(NextSampleOut)) | ((uint32_t)(NextSampleOut<<16));
```

Figure 13: Band-pass Filtering Algorithm for Heartbeat Waveform Display and Audio

The band-pass filter was designed to capture the heartbeat waveform from any user. The band runs from 150 to 300 hertz, attenuating other frequencies outside the band. The design was based off of the frequencies of a heartbeat and heart valves opening and closing. Capturing these components increases the performance of the visualization aspect of the waveform display as well as the sound output for the medical examiner.

Results

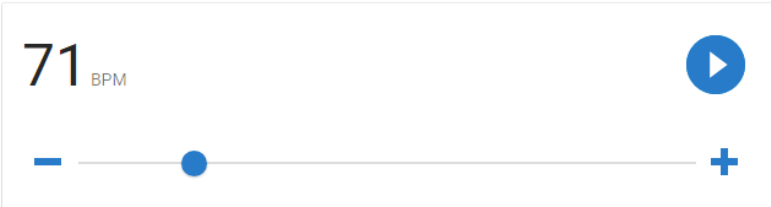

Figure 14: Ideal Signal as Input for Digital Stethoscope

The figure shown above resembles the input into the digital stethoscope. The metronome signal is a simulated test of an input that has periodicity and a short duration peak. Utilizing the low-pass and median filter algorithms in **figures 9** and **12**, the accuracy of the digital stethoscope is shown in **figures 14** and **15**.

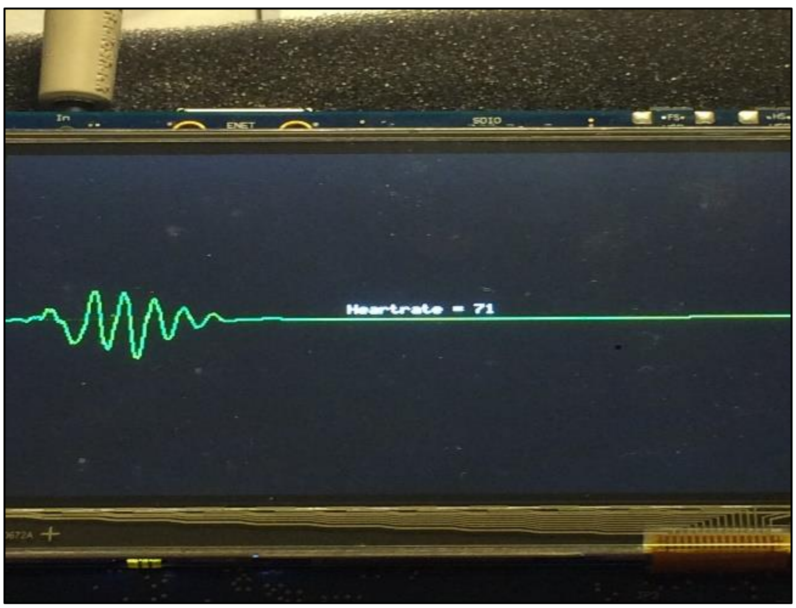

Figure 15: Display of Ideal Heartbeat Signal on Digital Stethoscope Device

The following figures 16 and 17 shows that the digital stethoscope works on periodic heart sounds using the LPCXpresso54608. It was tested using a pre-recorded heart beat sound in figure 16 and then tested on a user's heart beat sound in figure 17.

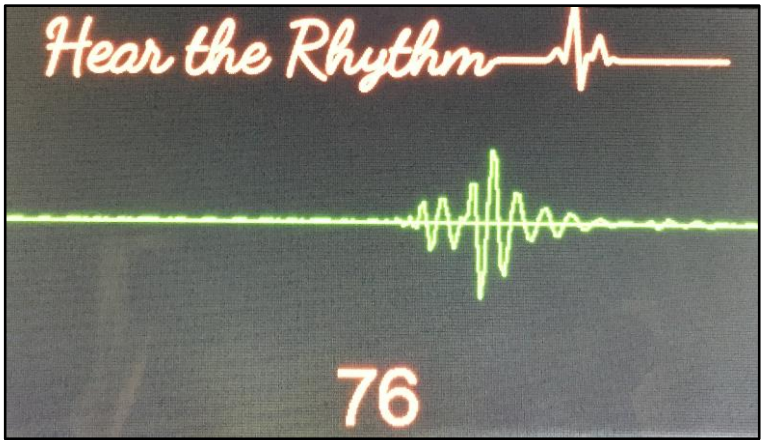

Figure 16: Display of Pre-recorded Patient's Heartbeat Signal on Digital Stethoscope

The simplistic programmability of the LPCXpresso54608 board reveals that the band-pass filter shown in **figure 13** successfully records and displays an ideal heartbeat signal. The same interval plotted in **figure 2** is also displayed now on the digital stethoscope, as shown in **figure 16**. This reveals that the device has been properly created to not only display a heartbeat signal, but also to simultaneously calculate a user's heart rate. The next simulation in **figure 17** shows similar results, however, now in real-time with a user. The median filter is designed to filter the heart rate signal such that outlier heartbeats do not affect the heart rate calculation. Outlier heartbeats refer to ones that beat faster or slower than the norm.

Figure 17: Display of Real Time User Heartbeat Signal on Digital Stethoscope

Observing the figure, the corresponding waveform is the realization of one single heartbeat captured and displayed by the digital stethoscope. The output of the system is the same analog signal that was sent into the system, however, it has now been filtered and re-constructed for the purpose of an accurate diagnosis to be made.

Conclusion

The conclusion of the digital stethoscope simulations lay the foundation towards future growth of heart rate calculations on an LPCXpresso54608 board. The incorporation of the developed algorithms on the device shall lead to an accurate digital stethoscope for the medical industry, at a more reasonable price point at $245.15. In turn, the device will alleviate noisiness accompanied by the traditional stethoscope and lead towards more precise readings of the components associated with a user’s heart, generated in real-time. The implementation of the developed algorithm for adaptive noise cancellations within the digital stethoscope will be a groundbreaking solution to a problem that industry has yet to solve. Various digital stethoscopes currently on the market cannot account for the random fluctuations of noise in an environment, especially as the gain of the system is increased for better signal resolution at the audio output. Therefore, the developed device can revolutionize the medical industry in heart rate calculation, acoustic signal display, and significantly increase the signal-to-noise ratio of the heartbeat signal heard by any medical examiner for more precise diagnostics.

References

[1] "Toolkits," *Electronic Stethoscopes - Technology Overview | National Telehealth Technology Assessment Resource Center*. [Online]. Available: http://telehealthtechnology.org/toolkits/electronic-stethoscopes/about-electronic-stethoscopes/technology-overview. [Accessed: 04-Apr-2017].

[2] "Devices: Smarter stethoscopes," *Berkeley Engineering*, 08-Jan-2015. [Online]. Available: http://engineering.berkeley.edu/magazine/fall-2014/devices-smarter-stethoscopes. [Accessed: 04-Apr-2017].

Appendices

Appendix A: GitHub Repository

The GitHub Repository can be found below.

github.com/mlm6016/Hear_the_Rhythm.git

Appendix B: Cost Breakdown

Description	Quantity	Unit Price	Cost
LPCXpresso 54608	1	$68.75	$68.75
Microphone	1	$74.95	$74.95
TRRS to TRS Adapter	1	$14.98	$14.98
Stethoscope Diaphragm	1	$25.48	$25.48
Case	1	$48.00	$48.00
Battery	1	$12.99	$12.99
Manhours	120	$80.00	$9,600.00
Overhead	120	$80.00	$9,600.00
		Product Cost =	$245.15
		Total Cost =	$19,445.15

Figure 18: Cost Breakdown of the Hear the Rhythm Digital Stethoscope

Appendix C: Specifications

Accuracy	±2 BPM
Weight	<= 3 lbs
Processing Delay	<= 0.5 sec
Battery Life	>= 2 hours
Signal Noise Difference	>= 20 dB
Peak Detection Accuracy	>= 95%

Figure 19: Specifications of the Hear the Rhythm Digital Stethoscope

Appendix D: Demonstration Video

A video demonstration for the Hear the Rhythm Digital Stethoscope can be found below.

www.youtube.com/watch?v=CHKEfmI9dlM