A synchronized phonocardiogram for shear wave elastography

G. Hogenhout

A synchronized phonocardiogram for shear wave elastography

by

G. Hogenhout

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday, September 21st, 2018 - 10:15 AM to 12:15 AM.

Student number:	4249216	
Project duration:	November 2016 – July 2018	
Thesis committee:	Prof. dr. P. J. French,	TU Delft, supervisor
	Dr. ir. H. J. Vos,	TU Delft/Erasmus MC, supervisor
	Dr. ir. M. Verweij,	TU Delft/Erasmus MC
	Dr. ir. A. Bossche,	TU Delft

TU Delft

Acknowledgements

G. Hogenhout
Rotterdam, September 2018

AFTER the work done on this thesis, it is time to express my gratitude to the persons that guided me through. This thesis is research conducted for the Delft University of Technology with the actual work done in the Erasmus Medical Center in Rotterdam. Therefore I would like to thank my supervisors prof. dr. P. J. French (TU Delft) and dr. ir. H. J. Vos (EMC Rotterdam) for their great support. I am especially thankful to Rik Vos for bending my enthousiasm to take up a new idea fast into a the more scientific attitude of thinking about the existing knowledge first. I learned a lot from you, also when the report had to be polished and finished. The guiding was great, also when other challenges came along in parallel to the master thesis project. Paddy, thanks for your questions - they were sharp and forced me to think differently about the report and about the process. I'm very grateful to have worked with two talented guides like you were for me.

I would like to thank the people at the Biomedical Engineering department of the Erasmus Medical Center in Rotterdam for supporting, critisizing and enabling my ideas. Without their collaboration, the product development would have taken much more time.

A final word goes to all the people I forgot to mention above: your contribution to this work goes unnamed but has been very helpful and indispensable. A special thanks to the people that knew about the work behind the words, about the sweat on the keys, about the lamp not being off for another night. A special thanks to my girlfriend Leonie Verheul - writing in the evenings gets in the way of going out together. Thank you all!

This work was supported by the TTW – Dutch Heart Foundation partnership program 'Earlier recognition of cardiovascular diseases' with project number 14740, which is (partly) financed by the Netherlands Organization for Scientific Research (NWO).

Contents

1
Introduction

1.1. Project Background

Cardiovascular diseases are a common and growing problem in the ageing society. Worldwide, cardiovascular diseases (e.g. coronary heart disease and stroke) are the leading cause of death[1]. In the Netherlands, the number of patients with heart and vascular diseases is expected to grow with 65 % until 2040, according to research of the Dutch Heart Foundation[2]. Early detection of cardiac failure symptoms and effective treatment can help reducing health care costs[3] and predicting cardiovascular events, which may prevent loss of quality of life, or even loss of life.

Early recognition of imminent heart failure is done by evaluating certain risk factors. It is the general assumption that the myocardial stiffness is correlated to diastolic disfunction[4]. During the heart cycle, shear waves are generated by the different moving substances inside the heart, like flowing blood and moving muscle tissue. The propagation velocity of shear waves depends on the tissue stiffness. Thus, by examining shear waves an indication of the myocardial stiffness can be obtained. Measuring tissue stiffness by recording and quantifying shear waves using ultrasound imaging is called shear wave elastography. To visualize and quantify a shear wave at least a few ultrasound frames are needed to capture the wave. Shear waves last for 10-25 ms in most cases, so for shear wave elastography framerates of over 500 frames per second are required. This measurement method is a non-invasive way of early heart failure detection. Currently, no comparable accurate non-invasive method for obtaining an early indication of imminent heart failure is available on the market[5].

This project is part of EFFECTS(Early Finding of diastolic heart Failure by EchoCardiographic Tissue Stiffness), a project which is supported by the Dutch Organisation for Scientific Research (NWO), the Dutch Heart Foundation(NHS) and Zonare Medical Systems. In this project, the Zonare ZS3 ultrasound imaging machine together with a special programmed research mode are used to record high frame rate images with frame rates up to 1000 frames per second. High frame rates are essential for recording shear waves with sufficient resolution. The obtained ultrasound images are used to characterize and quantify shear waves over the complete cardiac cycle.

A well-known existing method of measuring heart parameters non-invasively is the electrocardiogram or ECG. The ECG provides an overview of the electrical activity in the heart over the heart cycle[6][7]. However, for shear wave imaging it is more important to know at what time the heart valves close, which is more visible in a recording of the heart sounds, called the phonocardiogram or*PCG*. Recording a PCG is generally done using a stethoscope, which is a non-invasive method, like shear wave eleastography. Combining these non-invasive methods to couple heart sounds to shear waves could deliver interesting new insights in the early detection of imminent heart failure. The goal of this work is to record high framerate ultrasound images from the Zonare ZS3 synchronized with a phonocardiogram recording.

The wanted system firstly has to record heart sounds, that is why a section is included that explains the relevant details of heart sounds(paragraph 1.2). Currently, some recorders are on the market, that is whay section 1.3 gives information about currently available electronic stethoscopes. After recording, the signal has to be processed before it can be analysed, as it might contain unwanted information that needs to be filtered out first. paragraph 1.4 describes ways of processing the recorded phonocardiogram signal. For recording high framerate ultrasound images, the Zonare ZS3 is used, this is a boundary condition for the project. At the end of chapter 1, the reader will be informed about the available technologies that can be used to achieve the main goal of this thesis work.

1.2. Background information on heart sounds

1.2.1. Introduction

The solution to the project problem will be a technical solution, but the requirements will build on data and theory originating from medical research. Medical background theory is needed to understand the reasoning behind some technical system decisions. This section supports the reader with the needed background theory about heart sounds. The section will close with a discussion and conclusion about the background literature.

1.2.2. Heart sounds S1 and S2 qualitatively

M1, T1, A2 and P2 The heart sound signal going to be recorded contains several seperate sounds that are audible: the typical 'lub-dub' rhythm in a healthy heart. The 'lub' is the first heart sound, which is called S1. S1 is caused by the closing of the mitral valve(called M1) and the tricuspid valve (called T1) at the beginning of the systole[8]. The 'dub' is the second heart sound (called S2), which comprises of the closing of the aortic valve (called A2) and the pulmonary valve (called P2) after the systolic pause[9]. The delay times between M1-T1 and A2-P2 are called the *split*[10]. The exact starting time of M1 and A2 provides the most useful data, as for the connection with shear wave elastography, the start of S1 and S2 is the wanted information. Therefore, investigating the exact split and the heart sounds T1 and P2 is less important for this work.
The different phases of the ECG are used to indicate at what time S1 and S2 occur, as in figure 1.1. The intensity of S1 depends heavily on the length of the P-R interval, which is the time between the beginning of atrial excitation and the beginning of the ventricular excitation. When the P-R interval is shorter, the pressure difference is higher, so the sound intensity is greater. This result was obtained in a small test consisting of two healthy hearts with different P-R intervals[12]. In a more specific study on the intensity of M1 in the first heart sound, the same effect has been found[13]. The information above gives a short review with information about M1, T1, A2 and P2 considered relevant for this research. The reader is advised to use the referenced and available literature when additional background information is wanted.

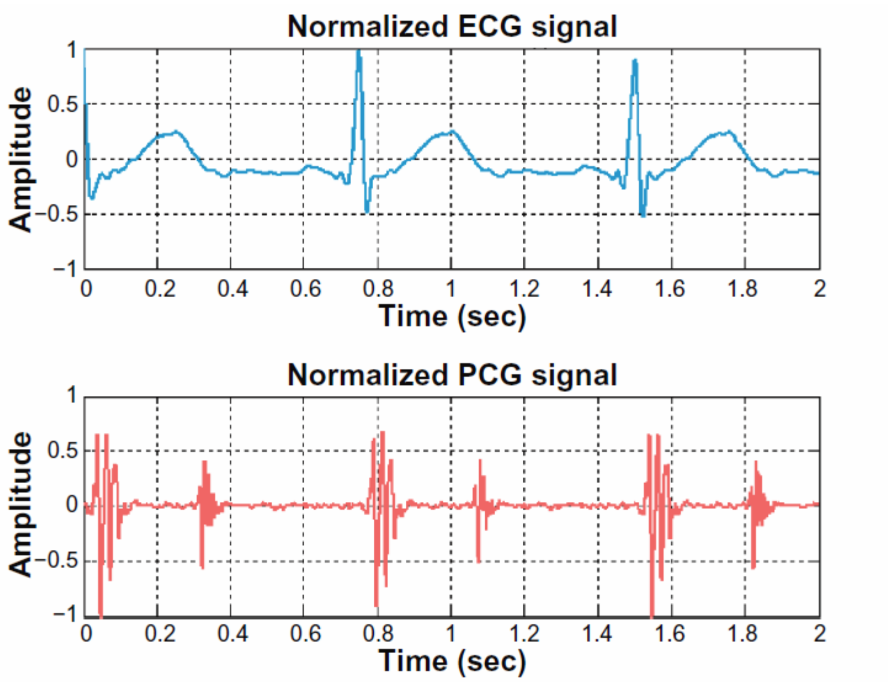


Figure 1.1: An example of normalized ECG and PCG signals[11].

1.2.3. Heart sounds S1 and S2 frequency content

In the previous paragraph, the reader was introduced qualitatively to heart sounds: what heart sounds S1 and S2 consist of. Contrarily, this paragraph will focus on heart sounds in a *quantitative* manner with the help of the signal spectrum. Fourier spectrum analysis of S1 and S2 can deliver useful information about several heart diseases[14][15]. Because the phonocardiogram will contain an analog or digital filter, the expected frequency bands for S1 and S2 have to be known. The frequency content of S1 ranges from 50-270 Hz and for S2, 50-300 Hz according to a 2007 study[8]. In a study concerning 74 normal persons, a dominant frequency range of 24-101Hz was found for S1 and a frequency range of 24-144 Hz for S2. In this dominant range, these frequencies have the highest amplitude. The study found a range from 24-500 Hz for S1 and a range of 24-550 Hz for S2, while the coefficient amplitude drops from 100 Hz relatively fast for both S1 and S2: S1 has a coefficient amplitude of -40 dB at 320 Hz compared to its amplitude at 100 Hz, while S2 is -35 dB at 320 Hz compared to its amplitude at 80 Hz. In this study, it was also found that the dominant frequency is unrelated to sex, blood pressure and body surface area[14]. In another study that measured the sound pressure level, the highest amplitudes for both S1 and S2 were found in the 20-40 Hz range. S1 was found to have a 20 dB rolloff from 70 to 310 Hz, while a 15 dB rolloff from 110 to 310 Hz for the aortic component of S2 was found[16]. In a study comprising 29 healthy males, 28 subjects had one significant amplitude peak in the frequency range of 25-38 Hz[17].

The information frequencies are generally on the same frequencies as many external noise sources like ventilation and medical equipment. Thus, signal filtering might become a crucial part of the new system. Furthermore, these low frequencies mean that the original signal can be recorded at a relatively low sampling rate of 500-1000 Hz (commercial sound recorders go up to 96 kHz, while the healthy human ear can only hear frequencies up to 20 kHz). Although the used literature consists of fairly different studies, a general line can be seen in the data. At frequencies from 20 Hz to 500 Hz, the biggest fraction of the information is present in the analog signal. However, from 100-120 Hz, the amplitude starts dropping rapidly, decreasing the information content in the signal. A recorder needs to be found or constructed that is able to record an analog sound signal from 20 to 500 Hz.

1.2.4. Heart sound S3

Besides S1 and S2, also other heart sounds can occur, as heart valves are not the only cause of heart sounds. Although mainly S1 and S2 are going to be used for working together with the ultrasound imaging machine, additional heart sounds have to be considered, as automatic detection that expect only S1 and S2 could fail on extra sounds in the heart cycle.
S3, also called the 'ventricular gallop' is a heart sound that is normally audible in children and young adults. It occurs 0.12-0.16 seconds after S2 in the early diastole[18] and has a frequency of 25-50 Hz[10]. In young humanss, an S3 can occur in a healthy heart: because the ventricle is not fully grown yet and thus relatively small, it is filling too rapidly. Over time, it will get a larger diameter and thus filling capacity[19]. The sound is caused by a sudden halt to the ventricular filling and it is thought to depend on the ventricular wall compliance and thickness[20]. In a healthy heart, the amplitude of S3 is lower than S2, but an S3 greater than S2 can occur. This was pointed out in a 2011 study concerning athletes with cardiac fatique[21].

1.2.5. Heart sound S4

S4 occurs in late diastole, actually just before S1. The main energy of this heart sound is below 30 Hz[22]. In a study across 100 healthy humans (healthy state confirmed with a full heart catheterization), an S4 was found in 75 persons[23]. The sound is believed to be caused by an impulse at the apex, after a left atrial contraction into a left ventricle that is not compliant or distensible enough[24][25]. Figure 1.2 gives the reader some context on S3 and S4 versus S1 and S2.

1.2.6. Displaying heart sound amplitudes

As can be seen in figure 1.1, figure 1.2, PCG amplitudes are not strictly displayed using the same unit, but merely scaled to a maximum amplitude. This might be caused by the still very human nature of auscultating heart sounds. This report will use the same approach.

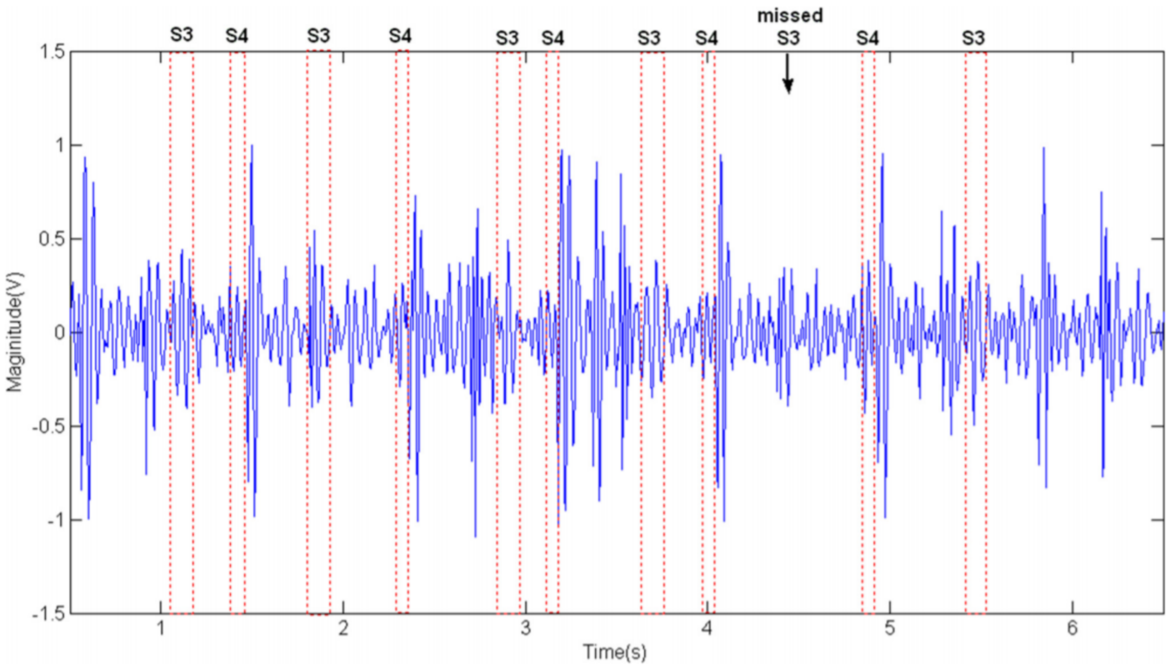


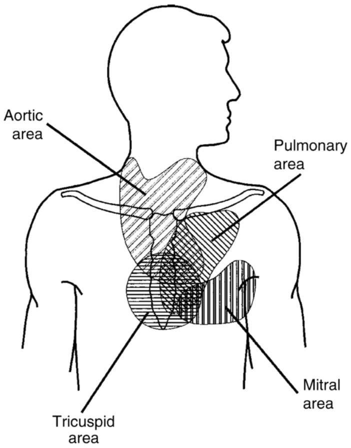
Figure 1.2: Amplitude comparison of heart sounds S3 and S4 versus S1 and S2[26].

1.2.7. Heart murmurs

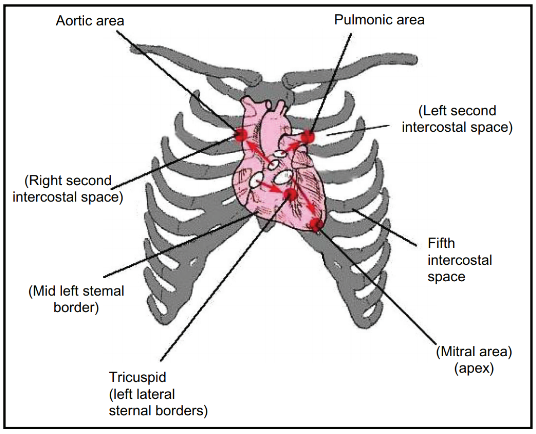
As blood flows through the cardiac vessels and holes, it often changes direction and speed, causing heart sounds called *murmurs*. These so-called physiologic murmurs are normal to occur, especially in children aged 3-6 or 8-12 years[27][28]. Physiologic childhood murmurs can stay and still be innnocent in adulthood[29]. In a research concerning elderly patients aged over 70, audible aortic and basal murmurs could not be linked to clinically significant diseases(excluding aortic stenosis)[30]. When a specific murmur indicates a heart disease, it is called a *pathologic* murmur. Most murmurs occur in the frequency range from 5-800 Hz[31]. A specific murmur does not have a fixed frequency band across humans, as the frequency spectrum of the murmur depends on physical parameters like the pressure gradient and the degree in which the blood flow is altered or blocked[32]. It is therefore hard to characterize murmurs by a frequency and an amplitude. Instead, Freeman and Levine made already in 1933 a six-point scale for heart murmurs, starting at the category *"The murmur is only audible on listening carefully for some time."* ascending in loudness to the sixth category: *"A loud murmur with a palpable thrill. The murmur is audible with the stethoscope not touching the chest but lifted just off it."*[33][34]. This 'Levine approach' is rather subjective as it depends on the quality of physician ears. A new approach that expresses murmur intensity against the intensity of heart sounds was proposed in 2005. It involved a controlled trial of 100 people. With the new system, the detection accuracy and consistency for some specific murmurs were improved, but the system has not been tested with real patients yet[35]. Murmurs are best heard in systole: in diastole, the blood has low pressure, thus the turbulence generated is small, causing a very low sound level[31]. Thus, systolic murmurs are researched in most literature.

1.2.8. Auscultation locations

As this work focuses on auscultation for timing reasons, the stethoscope head has to be in a location where the signal amplitude is the highest. Figure 1.3 gives an overview of where heart sounds (and murmurs) are heard the loudest. In figure 1.3.1, the general area in which auscultation has take place to achieve a good amplitude is displayed, while figure 1.3.2 focuses on the exact spot where the stethoscope records the highest amplitude. As the start of S1 and the start of S2 needs to be found, the mitral and aortic areas are the areas of interest (see section 1.2.2). It can be seen that a compromise already has to be made in recording the PCG, as there will be only one stethoscope head in the final system and the areas have little overlap.



(1.3.1) These four areas indicate where heart sounds and murmurs are the most audible[36].



(1.3.2) This figure explains the source for each heart sound and the advised auscultation location[37].

Figure 1.3: Auscultation locations for the different heart sounds and murmurs.

1.2.9. Discussion and conclusion on the used theory

Although the literature used for this section contains much more information about heart sounds and murmurs, only the information regarded relevant for this work has been treated. For example, in paragraph 1.2.2, the split was only mentioned as a phenomenon and not treated extensively, as measuring the split will not be the first priority of the new system. Furthermore, a thorough explanation about the pathologic causes for heart sounds S3 and S4 and heart murmurs has been skipped, as it does not add any relevant information to this work. What is relevant, is information about amplitude, timing and frequency content as that information will be measured by the new system. Thus, the focus for the preceding literature research is based on those parameters. The reader now has a concise overview of the generally available signal content in a phonocardiogram recording. In the next part of this work, currently available products on the market will be mentioned and compared.

1.3. Options already on the market for recording a phonocardiogram

1.3.1. Introduction

Section 1.2 gives an overview of the signal content that can be expected to be measured with the new system. This section elaborates on the actual device that records the respective frequency content. Phonocardiogram recorders (otherways called electronic stethoscopes) that are already on the market were considered. A search was done for currently available electronic stethoscopes and based on that, three categories were found. This division was based on the way of recording and digitizing:

1. An all-in-one stand-alone recorder with a stethoscope head, earpiece or earphones and a recorder/digitizer
2. A traditional stethoscope with a 'smart' electronic part placed on the tube in between the chest piece and earpiece
3. A 'smart' stethoscope head with a custom digitizer like a laptop or a smartphone

The connection to the outside world of the recorders in these three categories is very important and needs attention, as it will have to fit in and has to be controlled by a bigger system. The new solution should thus have options for external control in order to achieve a good synchronization of the sound signal to ultrasound images. Three specific recorders were chosen ase they fit in the above three categories. These are:

1. 3M Littmann Electronic Stethoscope 3100/3200
2. Eko Core Electronic Stethoscope
3. Thinklabs One Digital Stethoscope

(1.4.1) Thinklabs One[38]

(1.4.2) The Olympus LS12 voice recorder

Figure 1.4: The Olympus LS12(right) can be used to record the analog signals from the Thinklabs One recorder.

The 3M Littmann stethoscope has the general setup of an electronic stethoscope: a stethoscope head with a digitizer and earpieces all-in-one. The Eko Core stethoscope has the same setup, but it has the unique possibility to use it as a conventional stethoscope as the digitizer is in the tube connecting the stethoscope head to the earpiece. The Thinklabs One is a stethoscope head without a digitizer, needing an external digitizer and earphones. In the following paragraphs, their specifications will be explained.

1.3.2. Thinklabs One Digital Stethoscope

This paragraph gives information about the advantages and disadvantages of the Thinklabs One Digital Stethoscope(figure 1.4.1). It consists of a chest piece with some internal digital circuitry to select a range of signal filters. The user can select a frequency range and thus chooses to emphasize a part of the recorded thoracic sounds, like breathing, heart murmurs and heart sounds. An amplification of 100 over normal stethoscope ausculation is claimed[38]. The power used by the internal circuitry and the screen is supplied by an internal rechargeable battery. The analog signal is not digitized, recorded or stored but only amplified, filtered and passed onto a 3.5mm jack socket. A total of 100 patients can be examined on a full charge. An additional device has to digitize and store the signal. This makes Thinklabs One very flexible: it connects to anything with a 3.5mm jack input connector like PC's and smartphones. A splitter is provided with Thinklabs One to connect external headphones: simultaneous recording and listening is possible.

A drawback of this solution is that the recording devices are often not optimized for sound recording at the (relatively low) frequencies needed for recording heart tones. Expensive press voice recorders (like the Olympus LS12 recorder in figure 1.4.2) provide this low-frequency response, while PC's and smartphones are often not optimized for this task. This was tested by making a recording with a laptop and with a smartphone and listening to the signal with headphones. Both tests delivered relatively noisy recordings, the amplitudes of the noise and the heart sounds were in the same magitude range. A dedicated sound card could solve the problem. Another drawback of using the Thinklabs One is that an additional device is needed, which means that in total two devices are needed to record a phonocardiogram. Regarding the movement of the cables involved, this might cause practical issues when the user has to travel or move a lot. In terms of specifications, the final specification will depend on the recording device that is connected to the Thinkslabs output as the Thinklabs One amplifies and filters the signal rather than digitizing it. Therefore specifications like sample rate, dynamic range and storage time all depend on the second device, which gives great flexibility to new user's specifications. The Thinklabs One is priced at $499[39]. To conclude, Thinklabs One must be used together with an additional device that records the analog output signal with a low-frequency response. This secondary device must support the low frequencies present in heart tones and it defines the final specification of the recorder.

(1.5.1) 3M Litmann 3200 digital stethoscope[40]

(1.5.2) The Eko Core electronic stethoscope[41]

Figure 1.5: The Olympus LS12(right) can be used to record the analog signals from the Thinklabs One recorder.

1.3.3. 3M Littmann Electronic Stethoscope 3100 and 3200

The 3M Littmann Electronic Stethoscope 3100 and 3200 are integrated solutions: the microphone is placed in a stethoscope head and the recording is played back into the connected tubing, which is connected to a conventional earpiece(figure 1.5.1)). For both devices, an ambient noise reduction of 65% and a sound amplification of 24 is claimed. The Littmann 3200 can save up to 12 30 second sound tracks, while the Littmann 3100 is a listen-only device[42]. The Littmann 3200 can send its recorded audio signal over Bluetooth to a computer. The signal then can be analysed using the 3M™ Littmann® StethAssist™: with the software, the user can zoom, pan and use a slow playback setting to identify abnormal heart sounds. The predecessor to the 3100 and 3200, the Littmann 3000, was one of the best selling stethoscopes of its time[43]. The battery lasts 50-60 hours. The available Littmann documentation does not reveal any of the other relevant specifications, like dynamic range and response times for the bluetooth connection. These specifications are needed for making a decision on the possible synchronization of the final system. The Littmann 3200 is priced at $400[44]. The important information of the Littmann 3200 is that it can be connected (with Bluetooth) to the Littmann Stethassist software[45]. Thus, for integration with other systems, the Littmann 3200 would be the stethoscope to use over the Littmann 3100. To conclude, the Littmann 3200 is a widely-used stethoscope with excellent noise cancelling functionality and possible a wireless connection with other systems.

1.3.4. Eko Core Electronic Stethoscope

The Eko Core(figure 1.5.2) is also an integrated solution with a conventional earpiece and stethoscope head. The input analog signal is amplified by 40 times. It has an active time of 9 hours on battery power and connects via Bluetooth Low Energy to an Android or IOS App or to Windows PC software[46]. When a recording is made, the recording is remote accessible by other users who have the app and a password. An important selling point for this device is that the physical recording takes place in a tubular module that is connected between a conventional stethoscope head and a conventional earpiece. As a result, any physician can connect the recorder between the earpiece and the head of their own familiar stethoscope, which can help with a faster acceptation of the new technology. The amplifier electronics can also be switched off, in which case the sound waves reach the ears unaltered. The Eko Core is availabe at a price of $299[47] with stethocope head and earpiece. When only the recorder module is bought, the price drops down to $199[48]. Overall, this option might be a good solution for unwilling new users of the new technology, as it provides the old well-known system alongside a new system in one complete package.

1.3.5. Discussion about phonocardiogram recorder options

The phonocardiogram recorders mentioned in the previous paragraphs were selected based on their category and chosen to represent the category due to their current availability and popularity. Besides the audio quality requirements, also the way in which an electronic stethoscope records and shares data is important: it has to be connected to other systems. The required setup follows from the specifications that are listed in chapter 2. It could be necessary for cost reasons to build a custom stethoscope head. In chapter 3, a recorder will be chosen that matches the requirements. When larger numbers of this system are needed, a cheaper stethoscope subsystem could help keeping the cost down, thus influencing the final choice.

1.4. Phonocardiogram signal processing options

1.4.1. Introduction

In paragraph 1.3, three categories and market products of PCG recorders are listed. However, it is still unclear what steps are taken to process the recorded sound. As phonocardiogram signals are relatively weak signals compared to all-available background noise coming from machines often coming from medical machines. This section will inform the reader about processing the audio signal.

1.4.2. Approach for preprocessing a PCG

As human anatomy differs across patients, the ideal placement of the stethoscope and the volume of the signal do slightly differ. Thus, a normalization step can be executed first to get comparable amplitudes. Depending on the application, segmentation of the signal may be necessary to get single heartbeat recordings. After that, signal filtering and/or denoising is important, as many noise sources like air conditioning and medical equipment produces noise frequencies in the 50-100 Hz range, which overlaps with the expected phonocardiogram spectrum(see paragraph 1.2.3). To filter noise with frequencies existing in the signal spectrum, smart noise suppression techniques (also called noise cancelling) have to be applied. The signal will most likely be processed on a research computer, so sufficient computing power is available. Moreover, for reliability reasons a small overall number of devices and analogue circuit boards is advisable. Therefore, the best choice is to do the noise filtering in the digital domain.

1.4.3. Using the PCG: processing the PCG signal

Noise suppresion is generally done in the frequency domain, so a domain transformation is required first. The Fourier transform, wavelet transform and the short-time Fourier transform are generally used[49]. The Fourier transform does not work for converting heart sounds, as it assumes smooth-wave signals that are stationary in time. Heart sounds are non-stationary in time, so no real properties can be deduced from the FT. What can be done, is dividing the time in a lot of separate *windows*, making the frequency analysis time-dependent. However, some physical signals still change too fast for an adequate frequency resolution. Other techniques are using the Wigner Distribution (WD) and the Continuous or Discrete Wavelet Transform[50]. In a study, the wavelet transform performed the best: it distinguished between all four major components of S1 and A2 en P2 of S2 of a normal PCG signal, a performance that FT, STFT and WD couldn't match[8]. Many different wavelets can be used for denoising PCGs with mixed results[51]. In a recent study regarding PCG recording in noisy environments, noise from a TV set that generated 65 dB was removed from the PCG signal using an FIR bandpass filter and a specific wavelet transform[52].

1.4.4. Discussion and conclusion about phonocardiogram signal processing options

A summary of the current available phonocardiogram processing techniques was given, as the subject is well-documented in literature. An extensive amount of literature is available, of which some is referenced to in paragraph 1.4.3. The wavelet transform could be necessary for work executed in a later phase of this research, but an extensive review is left out as the wavelet transform does not belong to the main scope of this work.

1.5. Conclusion and discussion Chapter 1 and structure of report

At the beginning of this chapter, the reader has been informed about the relevance of this research. Three relevant phonocardiogram recorders listed. Of course, an off-the-shelf ultrasound imager that can record PCG and record ultrasound images is a rival to the chosen Zonare ZS3, but the Zonare ZS3 is a boundary condition for this project.

Depending on the specifications for this system(listed in chapter 2), certain subsystems will be chosen over the other and merged into one final system, which is described in chapter 3. After chapter 3, the measurement setup is ready. In chapter 4, the measurement results will be shown. As the recorder has three important subsystems, the results are split into three subresults. A discussion about the construction process and the obtained results is given in chapter 5, while the conclusion follows in chapter 6.

2

System Requirements

2.1. Introduction

Based on the needs of the future product owner, requirements have been made. For this system, there are technical needs and needs from clinicians. The requirements are split in functional and non-functional requirements. Functional requirements describe what the system shall do(the user functions), while non-functional requirements describe how the system works, so these request a specific performance value that has to be achieved[55]. This chapter will first descripe the user requirements in a quantitative manner, concluding with a list of numbered qualitataive requirements.

2.2. Description of the user requirements

In general, the system has to record a phonocardiogram synchronized with ultrasound images that have to be usable for shear wave elastography. The most important data to be obtained from the system are the start times of S1 and S2: the data needed for shear wave elastography. However, when the patient is unhealthy, this should be visible on the phonocardiogram as well. Following this reasoning, the phonocardiogram recorder needs a specified dynamic range to record the heart sounds from paragraph 1.2.2. It has to support the basic medical safety standard IEC 60601 to be usable in hospitals and be able to be cleaned with a wet towel and a desinfectant. When the system is used in a research scenario, the phonocardiogram has to be visualized within a limited period of time. The general process of saving, getting ultrasound images from the ultrasound machine and visualisation takes about a minute. Visualizing the audio should be done in a comparable time. Also, for the researcher or clinician, the heart should be audible during the recording in order to place the recorder in the optimal position for maximum signal amplitude. The recorder has to support the frequencies in which these heart sounds occur. Another requirement is that the system has to be portable and small enough to be mounted onto the ultrasound system that is going to be used. The PCG system is an additional system to get used to for physisians. Thus, to make sure physisians actually will use the system, it should not add too many user steps. This asks for a clear interface with a limited additional number of user actions. Moreover, the cost of the total system should be below €1000, which is a requirement asked for by the product owner.

Most of the described requirements above are described qualitatively, but to check whether the system meets the requirements, a quantitative description is wanted. For a PCG recorder the most relevant requirements are:

- Sample rate
- Bit depth
- Dynamic range
- Signal to Noise ratio (SNR)

2.2.1. Sample rate

As for sample rate, the information bandwidth of a phonocardiogram recording ranges from 20 to 500 Hz(see paragraph 1.2.3). This means that the recorder should sample at a minimum of 1 kHz due to the Nyquist theorem.

2.2.2. Bit depth

The bit depth follows from the dynamic range requirement, as it is just the number of digital levels between the highest and lowest amplitude. A higher bit depth helps to differentiate between noise and signal: the lower the input noise, the earlier the rising amplitude (caused by the heart sound) is distinguished from the noise. The noise requirement thus translates into a timing problem. As the frame rate of the synchronized ultrasound system is about 1000 frames per second, the timing resolution of the total system will be around 1 ms. Thus, having a higher resulution in the sound recording makes no difference for the final synchronization. As a result, the bit depth of the recorder should be such that the timing error is ±0.5 ms.

2.2.3. Dynamic range

For the requirements, also a dynamic range requirement is needed to verify the system quality. The dynamic range of a recorder is the ratio between the highest and lowest sound pressure level that can be recorded by the recorder. It is the main goal of the system to distinguish the heart sounds S1, S2 from the other (heart) sounds in the signal and record when they start. The second goal is to record the frequency content of the specific heart tones. For the first goal, the dynamic range should be high enough to distinguish the heart tones from the surrounding noise. Due to variations in human anatomy, the maximum recorded amplitudes and noise levels will differ across patients. Therefore, the dynamic range differs for every patient. This asks for the dynamic range figure to be relative to the signal amplitude. Therefore, the dynamic range will be measured in decibels. With A_{max} the maximum measurable amplitude and A_{min} the smallest measurable amplitude, the dynamic range equation is:

$$DR = 20\log_{10}\left(\frac{A_{max}}{A_{min}}\right). \tag{2.1}$$

For the lower amplitude, the smallest amplitude at which a repeating pattern in the signal can be seen will be used. In figure 1.2, the smallest repeating amplitude is 0.2V, while the maximum amplitude is 1.5. Therefore the dynamic range of the signal is calculated to be 18 dB. However, the PCG from figure 1.1 seems to have a much larger dynamic range, more close to 25 dB. The reason for this could be that figure 1.1 is an processed signal, while figure 1.2 is an unprocessed sample. Thus, the figure of 18 dB will be used for the requirement.

2.3. Requirements list

A summary of the reasoning in paragraph 2.2 is given below, following the division criterion in paragraph 2.1.

2.3.1. Functional requirements

1. The system has to record a phonocardiogram and save it on non-erasable storage

2. The sound file should be exportable to a PC

3. The recorded sound should be synchronized with high framerate ultrasound images

4. The system will be connected onto the ultrasound system

5. The system should be easy to use for non-technical users

6. The part of the system that is in contact with the patient should be able to be desinfected with a wet towel and a desinfectant.

7. The system has a complete galvanical isolation

8. The system must have an option to make the phonocardiogram live audible

2.3.2. Non-functional requirements

1. The system has to comply to the medical IEC 60601 standard
2. The timing error of the sound recorder cannot exceed ± 0.5 ms
3. The sample rate of the recorder is >1 kHz
4. The dynamic range of the recorder is >18 dB
5. The system has a maximum cost of €1000
6. The process of synchronizing the phonocardiogram audio signals with the ultrasound images consists of a maximum of 5 steps
7. The system should work with recording rates of 900-1000 frames per second.

2.4. Using the requirements

The requirements above are based on the requirements set up by the product owner. Technical and medical requirements have to be adhered to at the same time. Chapter 1 introduces the reader to the available material for the system, this chapter sets the boundary what solution will be used. Chapter 3 will eleborate on construction and validation of the measurement system. The requirements will be revisited in chapter 5, where the system will be reviewed using the same requirements.

3

Materials and Methods: the measurement setup

3.1. Introduction

In chapter 2, requirements were set up for the final system. This chapter gives the argumentation for the chosen components based on the requirements. For adhering to the requirements, three main goals should be achieved:

- Recording a phonocardiogram
- Recording high frame rate ultrasound images with the Zonare ZS3
- Synchronization of the ultrasound images with the phonocardiogram

These goals are dealt with in the same order in this chapter. The chapter finishes with the system integration. At the end of the chapter, the reader has an overview of the main design challenges and knows the the final system setup.

3.2. Recording a phonocardiogram

3.2.1. Introduction

This paragraph summarizes the problems of applying different off-the-shelf electronic stethoscopes to the needs of the new system and describes a way of dealing with cost and form function. The main question is: "What is the best option a *synchronised* phonocardiogram recorder?". The previously mentioned selection of stethoscopes (paragraph 1.3) is fitted to the system requirements to result in a list of working alternatives. The order of the sections below is based on the project decision flow. The first and most important issue of using market solutions will be synchronizing them with the total recorder system (paragraph 3.2.2), so the synchronisation section will come first. The recorder solutions that still adhere to the synchronization specification have the digitizer included, so the digitizer follows next (paragraph 3.2.3). The digitizer should namely fit in a synchronized shell. When the digitizer does not have a recording stethoscope head included, the design goes one step deeper in to the project hierarchy and discusses the best stethoscope head solution (paragraph 3.2.4). The section ends with the final phonocardiogram recorder that will be used.

3.2.2. Synchronisation options for the phonocardiogram recorder

Recording a phonocardiogram is not the only thing that the phonocardiogram recorder has to do: synchronisation is needed to connect the recorded phonocardiogram to recorded ultrasound images. To achieve synchronisation with an ultrasound imager, there has to be communication with the outside world, either to control the recorder by an input or control the outside world with an output. This paragraph will discuss the different synchronisation options for the various devices from (paragraph 1.3. The Littmann 3200 stethoscope does have a Bluetooth connection to the outside world, but it communicates with factory Littmann software only, making automatic synchronization with a external device difficult and time-consuming to implement. The Eko Core allows .wav files to be loaded into custom repositories, so writing an Android app would be a way of getting phonocardiogram recordings into a custom location[56]. However, the Eko Core cannot accept or send external commands, so external device trigger control is impossible. Therefore, the Eko Core is not the device of choice for this application. As said before(paragraph 1.3), the Thinklabs One looks like a stethoscope head and needs an external digitizer. This digitizer then should have a way of communication with other devices. A fourth option is to build a custom stethoscope head using a conventional stethoscope head with an included analog microphone element and use an external digitizer, which should also have a way of communication with other devices. A convential microscope head has the advantage of being known to physicians already. This increases the chance of it being actually used by physicians that might be unwilling to adopt the new technology. The stethoscope head (or chestpiece) can be of the double-sided type, meaning it can be used in diaphragm or bell mode(chosen by using either side). The bell mode is used for hearing lower frequencies and the diaphragm mode for hearing higher frequencies. In the diaphragm mode, external sounds cause a thin chestpiece diaphraghm to vibrate, which causes pressure differences in the hollow air chamber inside the chestpiece. These pressure differences are conveyed to the output of the chestpiece by an internal cilindrical hole. There, a tube can be connected to transport the signal to the physicians' ears, or an electronic recording device can be connected to that same end. The various possible options are known now, based on the system requirements a decision will be made.

3.2.3. Selecting the phonocardiogram digitizer solution for this project

Introduction

The previous paragraph highlighted the synchronisation issues with four selected phonocardiogram recorder options. In this paragraph, the selection of the chosen phonocardiogram recorder is discussed. The Littmann 3200 and the Eko Core have little or no options for synchronisation with other devices. Thus, the final system will have a custom digitizer either with the Thinklabs One or a custom built stethoscope head connected to the analog input. To make synchronisation possible, the digitizer should have a trigger input or output. It was chosen to use a central command microcontroller(as explained later), so the digitizer should have an input trigger.

The audio recorder can be implemented in various ways with different levels on integration and user options. The hardware can be developed in-house or bought as a product. A circuit board can be made that contains an AD converter, sampler and storage. Building a custom circuit for the recorder gives great control to build a device specific for this job. As market solutions are developed often for the more general type of voice recording, features are available that are not going to be used. This causes an inherent larger use of space and power. Having a more general device on the other hand gives greater freedom when new functionality has to be added. To complete the project in time, a balance between buying off-the-shelf and building circuit boards has to be found. Sound can also recorded by a sound card and connected to a mini computer like the Raspberry Pi, achieving the same goals. Summmarizing the options, the following list appears:

- An off-the-shelf recorder
- A (USB) sound card with PC/mini computer
- A dedicated AD converter together with a microcontroller and a storage SD card

These options will get an elaborate review in the next paragraph.

Elaborate review on three possible phonocardiogram digitizer solutions

In the previous paragraph, it was concluded that the final phonocardiogram recorder solution can have various components with different levels of integration. In this paragraph, the three mentioned levels will be explained.

For converting an analog audio signal to a digital file, standard options are available as voice recorders. Many voice recorders have analog inputs, but the frequency range has to be equal to or larger than the bandwidth of phonocardiogram signals. Moreover, the internal amplifier has to support the maximum amplitudes generated by the connected analog microphone. Buying an off-the-shelf voice recorder also complicates the synchronization: the recorder is a finished product, so the device needs some customization to have accurately timed control.

As a second option, a dedicated sound card can also do the AD conversion; this requires a computer system to be included in the system. In terms of maximum sample rate and bit depth, it was found that USB sound cards are as good as off-the-shelf voice recorders[57][58]. A USB sound card normally needs a desktop PC or laptop, which is too large in size to meet the specifications. A solution to that would be to use a minicomputer like the Rasberry Pi, which uses the Linux operation system and fits in small holes with its credit card-sized motherboard. The RaspBerry Pi has a possibility to use the hardware timer to generate an accurate clock signal(for triggering the ultrasound imaging system, see paragraph 3.3), so as a package the Raspberry Pi can do almost all tasks needed from the system: triggering the Zonare, controlling a USB sound card, sending commands over the USB data line and communicating with an optional touchscreen for communication with the physician who controls the system. However, not all sound cards do work together with the Raspberri Pi yet. It is still yet unknown what effect an additional USB sound card will have on the processor performance. For example, if the sound card sends too many interrupts to the main processor, it might affect the triggering accuracy.

Thirdly, the implementation of AD conversion and storage is possible on circuit level too, using an AD converter, a microcontroller, a storage element like an SD card and a custom PCB. As stated before, constructing an entirely new circuit gives great control over all relevant parameters like synchronisation, system cost and power usage. However, building the system up to a reliable specification will include some product iterations and hardware debugging. The scope of the project is not building just the phonocardiogram recorder, but also the synchronisation system. Going deep into electronics for a subpart of the project could draw too much time easily, delaying the project planning. This note should be considered when choosing this option.

To conclude, for this subpart of the project, three approaches were found to build the recorder. They have an increasing construction complexity and an increasing level of control: a custom circuit that serves the specific needs of the project provides much more open ends of control (but requires more time to implement) than a fixed off-the-shelf solution. Using the Raspberry Pi to control a USB sound card could increase timing jitter on the triggering of the Zonare. The safest way to build a prototype in the first place might be to use a dedicated hardware timer and conclude later on using the Raspberri Pi. Based on all considerations in this paragraph, the off-the-shelf recorder was chosen as the digitizer.

Choosing the digitizer

Regarding the digitizer, the Olympus LS12 linear PCM recorder fulfilled the frequency and amplitude requirements, as it supports a frequency range of 20 Hz to 20 kHz[57]. Moreover, it uses only a few large buttons to operate the device, which makes the device simple to use for learning users. It can also be triggered using its remote control input. It works on two AA batteries, which is good for medical application as it prevents ground connections and can be moved around quickly without the need to connect power cables. Testing the device for this research for three months did not drain the batteries, indicating a long battery replacement interval. Its list price of $120 makes it a great value for money option considering all options offered, therefore it was chosen as the standalone audio digitizer for this project.

Conclusion

Based on the synchronisation requirement, the choice for a specific type of digitizers was made. Three digitizer types are evaluated and a conclusion was drawn on the final digitizer. As the digitizer of choice (Olympus LS12) does not have a stethoscope head included, a stethoscope head has to be bought or constructed. Therefore the next part in the project flow will treat the stethoscope head that will be used.

3.2.4. Stethoscope head choice

In the previous paragraph, the Olympus LS12 was chosen as a phonocardiogram digitizer for the project. The stethoscope head is still unknown. Besides converting a standard non-electric stethoscope head to an electric recorder stethoscope head, not many solutions exist that consist of just a stethoscope head with an analog microphone. The Thinklabs One(treated in paragraph 1.3.2) is the only other option that will fit within this project. However, it costs $499, which is half the project budget. A more budget friendly solution for this expensive head would be a custom-built stethoscope head. This would use a standard stethocope head, which costs around $30, together with a built-in microphone that produces an analog output signal. In total, he cost of this custom head is about $50, which is a considerable cost saving compared to the Thinklabs(considering low production numbers). The custom head still can be used together with a synchronized digitizer.

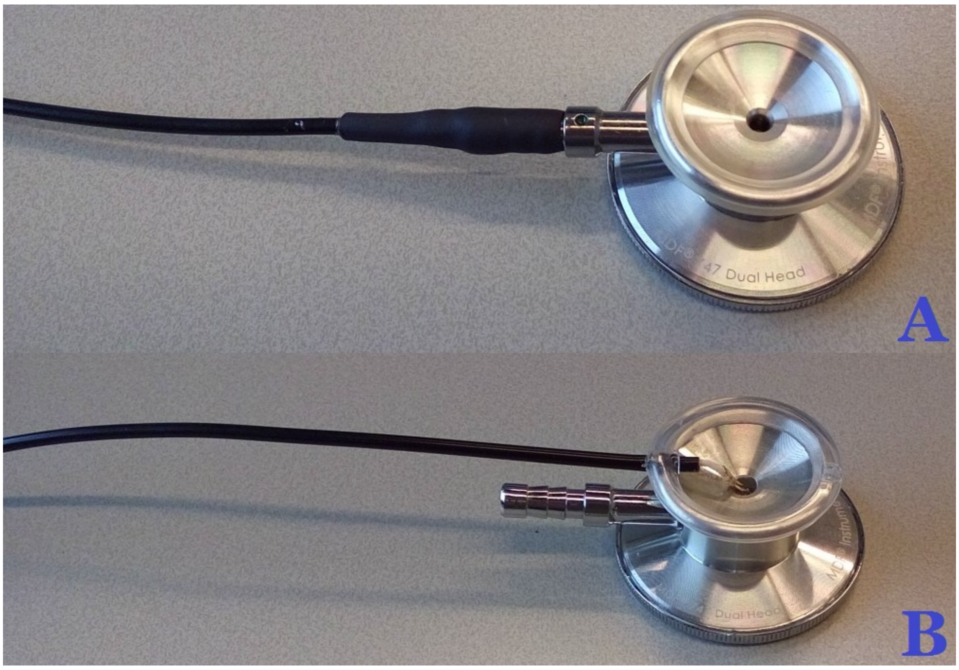


Figure 3.1: A: The TOM-1545P-R microphone is built into the metal tube connector of the analog stethoscope head. B: The TOM-1545P-R microphone is built into the central tube of the analog stethoscope head.

3.2.5. Building the stethoscope head

In the previous paragraph, it was chosen to build a microphone element into a custom stethoscope head. In this paragraph, the microphone choice will be explained. For the available space in the stethoscope head, two technologies were possible: a piezo-electric microphone and an electret microphone. In general, piezo microphones have a higher noise floor than electret condenser microphones. As the incoming signal is already weak(phonocardiogram coming from the body), a low amplifier noise is essential. Furthermore, regarding the electret microphone element types that are going to be used, a FET is included into the electret chip die itself that amplifies the signal. This is not the case for a piezo element. The microphone with an amplifier would consume relatively more space. As a wire of about 1-2 metres will be connected, having the analog signal amplified before the cable is advisable. Based on these findings, the electret microphone is chosen to be used inside the stethoscope head.

In the selection of the electret microphone, size is the major decision factor. It was found using two versions(see 3.1) of the stethoscope head that the best place to put the electret microphone in the analog stethoscope was at the place where the original rubber tube is connected to the metal chest piece. The most limiting specification for this microphone is its size: due to the outer diameter of the metal tube of 5 mm, the inner diameter cannot be larger than 4mm or otherwise the tube will be too weak mechanically. The second limiting parameter is the frequency band in which the frequency response has to be flat. As stated in paragraph 1.2.2, heart tones occur from 10-250 Hz. These relatively low frequencies are a challenge for microphones with an outer diameter of 4-5 mm, as physically larger microphones support lower frequencies: a larger diameter electret condenser microphone means more available area for the internal capacitance that converts an acoustic signal to an electrical signal. A microphone with a larger area capacitance in turn can reproduce lower frequencies. The Projects Unlimited TOM-1545P-R electret microphone was chosen, as it matched these specifications the closest. This microphone has a sensitivity of -45 dB, an SNR of 60 dB, an output impedance of 2.2 kΩ and an active current of 500 μA. The documentation on this microphone is limited but it has a flat frequency response of at least 30 Hz to 20 kHz[59][60]. It is the only electret microphone that combines a 4 mm size with this wide bandwidth. At this point, the choices for stethoscope head and digitizer are made and the stethoscope head is built. The next step in the design will be finding the right ultrasound imager and synchronizing it with the work from the previous paragraphs in this chapter.

3.3. Recording ultrasound images

3.3.1. Introduction

The second goal of the system is recording ultrasound images at a frame rate of at least 900 frames per second using the Zonare ZS3. The following part of this section will introduce the reader to using the Zonare ZS3 in the bigger system. After this section, the Zonare triggering is defined and ready to be implemented in the main system.

The Zonare is normally controlled by its button user interface, but for the research system a USB connection was provided by the manufacturer. This function can be used to activate several custom Zonare research functions. For the EFFECTS project, the Zonare is taken up to its limits, as a number of specific commands sent over USB allow the user to get the maximum frame rate- given the current Zonare settings and physical limitations of the system. The Zonare works up to its maximum achievable speed in this special research mode. When the command

```
ipce set FRAME_RZOH 4000
```

is sent, the Zonare is trying to achieve a frame rate of 4000 Hz, which is too high for the Zonare's processor, as the maximum achievable frame rate (considering its computing power versus the image depth and width) is 900-1000 frames per second in the current setting. The software automatically calculates the highest possible frame rate based on the entered ultrasound parameters, the value will thus continue to drop until it has reached a balance frame rate at which the internal processor is working at the maximum of its capabilities.

Another USB-programmable option is the possiblity to enable an external frame trigger:

```
ipce SetEnableTriggeredFrames 1
```

When this command is given, the internal frame trigger switches off and the system freezes. The system now starts and stops recording only on external frame triggers. The external frame trigger is input via a custom trigger cable provided by Zonare Medical Systems. The required input signal specification can be found in appendix B. This command can be used together with the high speed mode. The external frame trigger has to be set just below this maximum frequency: When the trigger frequency is just too high, the final trigger frequency can be seriously reduced, this is explained in figure 3.2. A solution to this could be to trigger the Zonare with a much higher frequency. In figure 3.2, this option is explained in the fifth sequence. By sending triggers much faster than the available time frame allows, the Zonare will always respond as soon as it becomes available.
However, it was found that by overtriggering the system with frequencies that were at least 3 times the maximum possible trigger frequency, white lines started to appear in the ultrasound images. This started happening from a trigger frequency of 3 kHz. A possible explanation of this is the limited spare computing power in the special research mode: when the new frame trigger arrives too early (caused by the higher frequency), the processor has to spend some memory space on the incoming interrupt. As every full ultrasound frame is made from individual pixel lines, the transition to a new image could appear as a white line. So to conclude, overtriggering the system is not a safe way of finding the optimal maximum frequency automatically. The trigger frequency should be kept carefully under a limit to prevent the trigger frequency from dropping rapidly due to overtriggering.

As the reader now knows how the system is triggered, image saving is the next topic. The Zonare ZS3 does not just save the frames that were triggered by the external trigger. In fact, all frames that are triggered are saved to a circular buffer, the so-called 'cine memory'[61]. When the external triggering is on, data frames are being saved to the cine buffer. When the exernal triggering stops, the Zonare will get in an automatic freeze and the last dataframes in the cine buffer will have the external triggered ultrasound images. The Zonare system starts imaging a new frame on every edge of the trigger input signal or on the first edge of a pulse shorter than 20 microseconds[62]. For example, when 50 Hz pulse is used as an external trigger, the frame trigger frequency is 100 Hz as a 50 Hz pulse has 100 transitions per second. As said befor, care has to be taken not to trigger too often, as it can lead to a large drop in frame rate of about 50 % when trigger pulses are arriving too often. To prevent this dramatich frequency decrease, it is important to know exactly what frequency the frame rate has dropped to (as explained in paragraph 3.3.1). This number is saved by the Zonare and can be retreived using the USB command

```
ipce get ACTUAL_FR
```

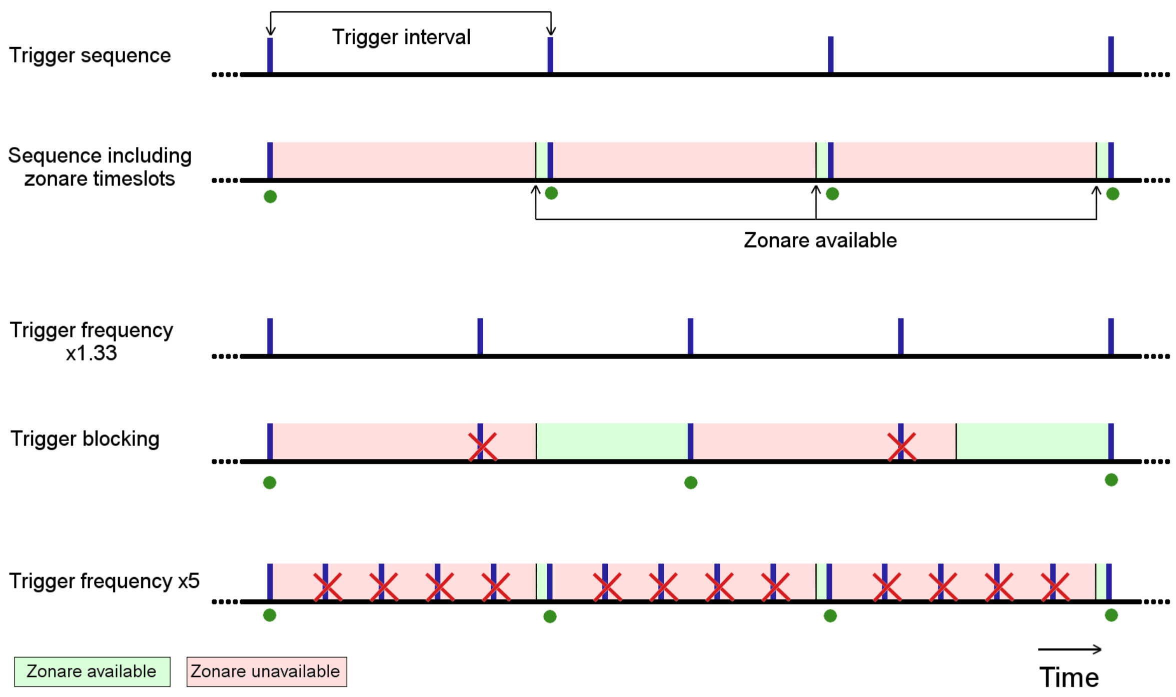


Figure 3.2: This image shows one normal trigger sequence (first sequence) and one trigger sequence that is too fast (third sequence). Each blue bar represents a trigger event(edge change). In sequence 1, each new trigger arrives in an 'available' time frame. In sequence 3 and 4, every second new trigger arrives in an 'unavailable' time frame, thus the Zonare gets no new trigger for some time: the effective trigger frequency decreases. The fifth sequence shows the possible solution of overtriggering.

The obtained value is then used for generating an accurate triggering signal, which is described in appendix C and paragraph 3.4.

3.3.2. Conclusion on ultrasound imaging

In this section it was explained why the Zonare ZS3 ultrasound imager was chosen, what the problems about the triggering system comprise and how the image saving inside the Zonare works, In short, it gives the reader an overview of how to work with the system side of the Zonare ZS3.

3.4. The microcontroller board

3.4.1. Introduction

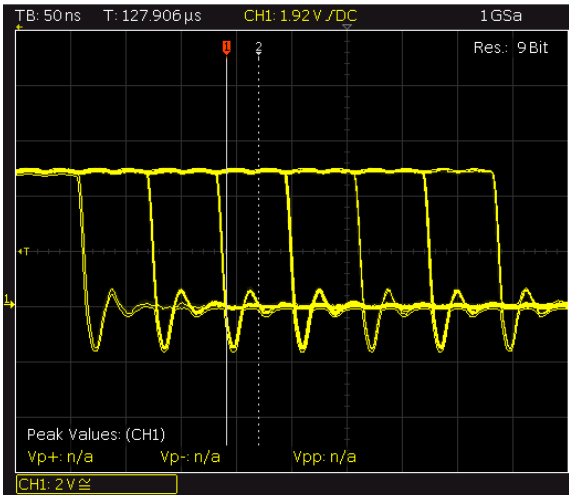
To add sound to the ultrasound images, both ultrasound recording and audio recording need control from the central microprocessor. This section describes how the Zonare ZS3 ultrasound imager is triggered and how the problems regarding the control were solved.

3.4.2. Controlling the Zonare

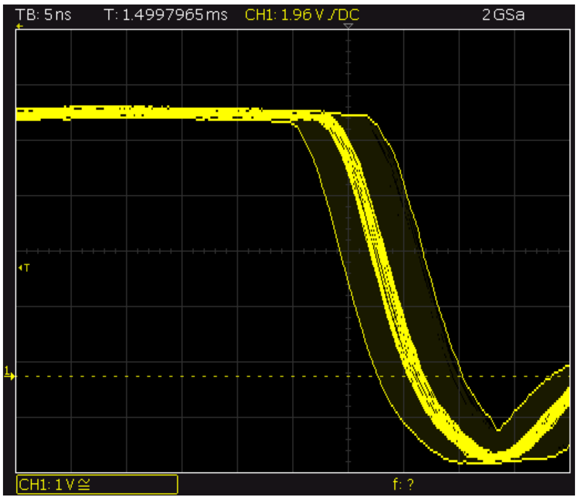
The Zonare has different ways of starting and stopping with images. An external hardware trigger option was provided by Zonare in the shape of a trigger cable. Moreover, the triggering can also start on external commands over the USB cable. As the delay of using the USB option is unknown yet, the external hardware trigger has been chosen as the system to go with. These triggers have to be generated by some digital logic. Also, the audio recorder needs to be controlled. Boards like the Arduino board (with the Atmega328 processor) or the Raspberri Pi are well-documented choices, which is the reason why they are chosen for this project. A Raspberri Pi can be used the USB link to the Zonare. It was found that the externally triggered frames only can be stored to a USB device if the 'freeze' button on the Zonare is pressed or when the command

```
cine freeze
```

is sent. To make the system more automated, using the USB link to execute a freeze is desirable. The Raspberri Pi can take care of this USB link, while at the same time controlling a user interface, for example a



(3.3.1)



(3.3.2)

Figure 3.3: Plot (a) shows all times on which the edge occured using the interrupt routine toggling. It is visible that the signal goes to zero at seven discrete times, separated by 62.5 ns which corresponds to periods of the 16 MHz system clock. Plot (b) is zoomed in on one edge and shows a falling edge using the final code in which not interrupt routine is used but just the interrupt. The deviation is just 4 ns in that case.

touchscreen.
Because a very accurate trigger signal is needed for the Zonare, low level programming is needed. This was pointed out in a test with the atmega328 microprocessor that used an interrupt routine to toggle the trigger output port. The first approach was to use the interrupt routine to toggle the output port (listing 3.1).

Listing 3.1: The interrupt routine code

```
ISR(TIMER1_COMPA_vect) {
if (toggle) {
digitalWrite(TEST, LOW);
toggle = 0;
}
else {
digitalWrite(TEST, HIGH);
toggle = 1;
}
```

but that test pointed out a possible error of 375 ns in the trigger signal(see figure 3.33.3.1) at regular intervals of 62.5 nanoseconds. A cause for this could be that the processor has to write to or copy from a varying number of registers (e.g. because of varying numbers that are used) before the interrupt routine is executed. Because the signal *edges* trigger Zonare ultrasound frames, this results in Zonare images produced 375 ns too early or too late, causing 'image timing jitter'. Regarding the the general scope of the project , this is an unwanted effect. The goal of the high framerate ultrasound imaging project is scanning the velocity of shear waves. Shear wave velocity is derived from the displacement of tissue divided by the time difference (v=dx/dt)[4]. Therefore, when there is timing jitter in the images, there will be noise in the determined shear wave velocity, which is highly undesirable. To counteract the timing jitter, a special way of toggling the output port is used. As previously, an interrupt is used to generate a trigger signal, but the interrupt does not activate an interrupt routine but only toggles the output pin. The corresponding program code can be found in appendix. Using this new method, the deviation of the falling edge from its normal position is about 4 ns, as can be seen in figure 3.33.3.2. The same figures hold for the rising edge cases. The plots in figure 3.3 are made using a 1 khz signal with an amplitude of 5 volts.

3.4.3. Final note on Zonare control

The Zonare is USB controlled with a set of commands. These commands are listed in appendix C. Because it is still unclear whether a sound card can work together with a Raspberri Pi that sends hardware triggers, the safe option now is to choose a relatively simple microcontroller to generate a hardware timer with low timing jitter and use the result of this test to conclude on a system improvement. Indepently generated high quality

Table 3.1: USB commands that control and configure the ATmega328 microcontroller

Command	Action
ptrig setfr [value]	Set the frame trigger frequency
ptrig start	Start the synchronized recording
ptrig stopt	Stop the recording

hardware triggers can reveal system errors that do not have a relation with triggering. Therefore the Arduino microcontroller is choses as a first option with the Raspberry Pi in mind for a further system development.

3.4.4. Controlling the Olympus LS12

The microcontroller board will need some control over the stethoscope digitizer, the Olympus LS12. That device does have a remote jack, but only infrared remote control modules fit to the jack. However, due to the relatively simple protocol, the signals of the remote control receiver can be imitated by a custom circuit. This enables a programmable microcontroller to control the recorder's REC, PAUSE and STOP buttons without changing the hardware in any way. Using a 2.5 mm four-pole plug and a transistor circuit, the LS12 is easily controlled by the central microcontroller by manipulating the voltage on the first ring of the 2.5 mm plug. The protocol is listed in appendix D. To initiate a recorder action, the first ring of the 2.5 mm plug has to be pulled down from 3.0 to 1.5V. This is done by connecting a 100kΩ resistor between the first ring and ground using a switching circuit. In this way, full control over the LS12 recorder is gained by a small external circuit, allowing for a simple microcontroller to control both the Olympus LS12 and to send hardware triggers to the Zonare ZS3.

3.4.5. External control over the central microcontroller

In the previous paragraphs, the control over the Olympus LS12 and the Zonare ZS3 was discussed. They are controlled by the central microcontroller. For testing purposes, a control system is programmed to start and stop a recording using a terminal to the Arduino microcontroller. The present software on the microcontroller then converts the commands in a control sequence that controls the LS12 and the ZS3. The terminal commands can be found in table 3.1. When commands 1-3 of table 1 are sent, the Zonare is configured for using with the synchronizing system. A recording then consists of the following commmands:

1. (To Zonare USB): `cine unfreeze`
2. (To ATmega328 USB): `ptrig start`

The recording can last as long as the user wants, but only the last seconds of the recording can be saved due to the limited cine memory of 90 Mb[61]. In practice, this translates to a maximum of 2000-3000 frames depending on the ultrasound parameters. As frame rates of 900 Hz are possible, this is a maximum of 2-3 seconds. When the user has recorded the wanted ultrasound images, the recording is ended by sending

1. (To ATmega328 USB): `ptrig stopt`
2. (To Zonare USB): `cine freeze`

In this state, no new images are recorded so the cine memory contains the last wanted seconds of ultrasound images. To save these, command 6 of table 1 is used.
The final system setup is ready at this point. The Zonare gets triggered and the LS12 can be switched on and off. What remains is the actual synchronisation of the devices. That topic will be dealt with in the next section.

3.5. Synchronization

3.5.1. Introduction

In the previous section, the system setup was finished. An audio recorder was chosen and the phonocardiogram recorder was constructed. Also the Zonare commands were listed and a microcontroller was chosen and applied. Now, all these components have to start working together as as complete synchronized system. This chapter addresses the issue of the synchronization in further detail.

3.5.2. Audio recorder synchronization considerations

Approach

The Olympus LS12 audio recorder is controlled using the IR remote jack(see paragraph 3.4.4). The initial approach was to trigger the Zonare and the LS12 simultaneously, based on the assumption that sending an ON signal with the ATmega328 would mean the Olympus LS12 immediately start a recording. To check whether this really is the case, the test in figure 3.4 was developed. Figure 3.4 shows in short what materials and method were used to evaluate the debounce delay. It appears that there is a substantial (and varying) delay between the ON signal and the audio file start: the delay times vary from 500 to 550 ms. Because the delay is varying, the LS12 cannot be correctly synchronized with open-loop control, therefore to achieve a synchronized system there has to be a method to compensate for the varying debounce delay.

Test	
Value to obtain	Debounce delay of Olympus LS12 recorder when operated by remote jack input (in milliseconds)
Measurement tools	• Olympus LS12 • ATmega328 • Trigger circuit
Method	Send ON signal to LS12, wait guessed delay time t_{guess}, START pulsed signal on left audio channel, check audio file.
Result is found when:	If the applied pulse (visible in .wav file) starts on a time t_p after the .wav file starts, the debounce delay equals $t_{guess} - t_p$.

Figure 3.4: Test to evaluate the debounce delay of the Olympus LS12 recorder remote jack input.

The delay in the 'REC' action can be determined by activating the recorder during a specific time t_{on} with an accurately timed reference (e.g. a signal generator) and checking the file length afterwards. The file length will then be t_{on} minus the debounce time. However, the STOP command also has a debounce time, which could be at the end of the sound file. Therefore, an additional truncation step is needed. For determining the delay, the triggering signal can be put on one of the audio channels. As the audio recorder is ON before the Zonare starts and OFF when the Zonare stops (due to the long delays inside the LS12), the total Zonare triggering sequence will always be recorded on the audio recorder. In this way, software can afterwards truncate the audio file with high accuracy when the recording speed of the audio is high enough. Assuming the Zonare is recording an image when a trigger is sent, this system exactly synchronizes the audio where the Zonare was recording.

Verification procedure

In the previous paragraph, a way of synchronizing afterwards was presented that promises good results. However, to verify whether audio events really match ultrasonic events, a test is needed in which an audio event and an an ultrasonic event happen at the same time. After compilation of the video (using ultrasound images from the Zonare and the recorded sound from the sound recorder), any deviation in time between the events will indicate a delay in one of the subsystems. The same event for both technologies is generated by tapping the ultrasound probe with the audio recorder chest piece using some ultrasound gel to make the short tap visible on ultrasound images. Figure 3.5 shows in short what materials and method were used to test the quality of the synchronisation. The results for this test will be presented in chapter 4.

3.6. Final note on materials and methods

In this chapter, the final measurement setup became clear to the reader. As the system consists of many different components, many different choices had to be made. The focus of this chapter was to make the choices clear and eleborate on the different options for each choice. At this point in the project flow, the measurement setup is ready for use. A testing and verifying section was added as the system itself might have unwanted behaviour or specifications. The next chapter will contain the project results. Based on the content

Test

Values to obtain	Occurence times in audio data and iq data of a small tap on ultrasound transducer with the stethoscope head
Measurement tools	• Olympus LS12 • ATmega328 • Test code to run program for 500 ms • Trigger circuit • Zonare ZS.3
Method	Start test code, tap ultrasound transducer with stethoscope head (with gel), check audio file and iq data
Result is found when:	In the .wav file, t_{audio} is the time between the synchronisation start (visible on the left channel) and the time on which the tap happens(right channel). t_{us} is the difference in timestamps between the start of file (marked by big difference with previous timestamp) and the time on which the tap happens (visible with `implay`). If t_{audio} equals t_{us} inside the resolution(dictated by framerate), this test is passed.

Figure 3.5: Test to obtain the time difference in the tap test.

in this chapter, a schematic overview of the system has been made. The overview can be found in figure 3.6. The system setup can be found in figure 3.7.

H

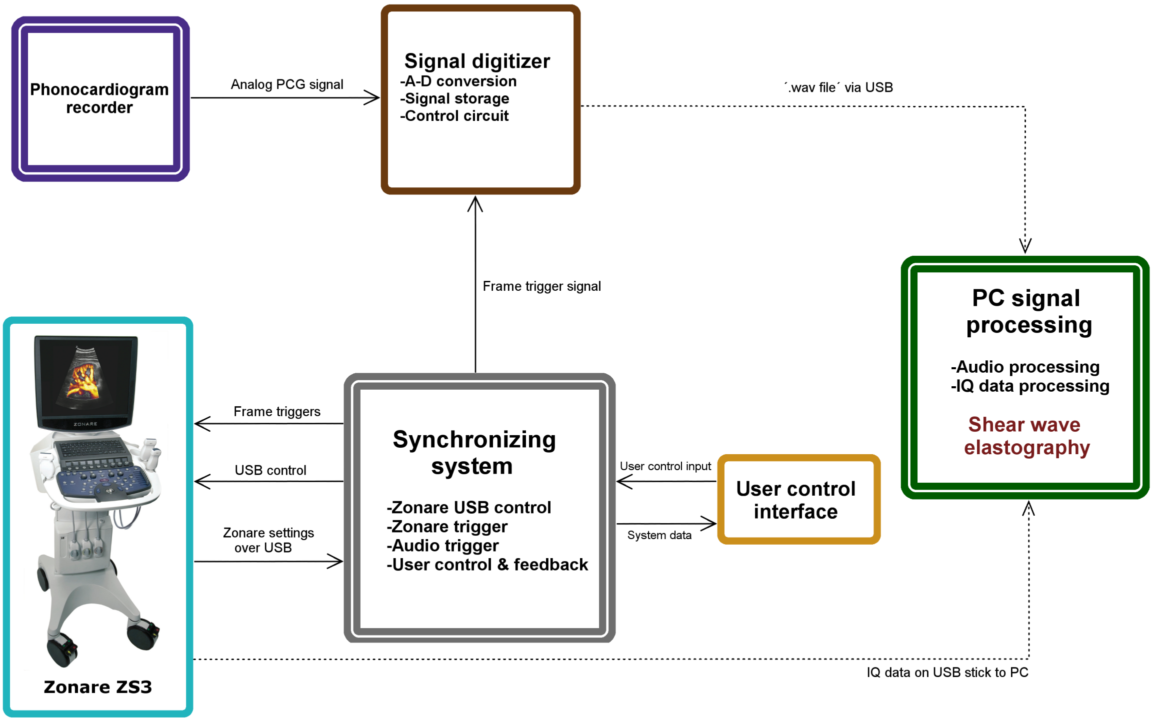


Figure 3.6: This image shows the complete system setup as discussed in chapter 3.

H

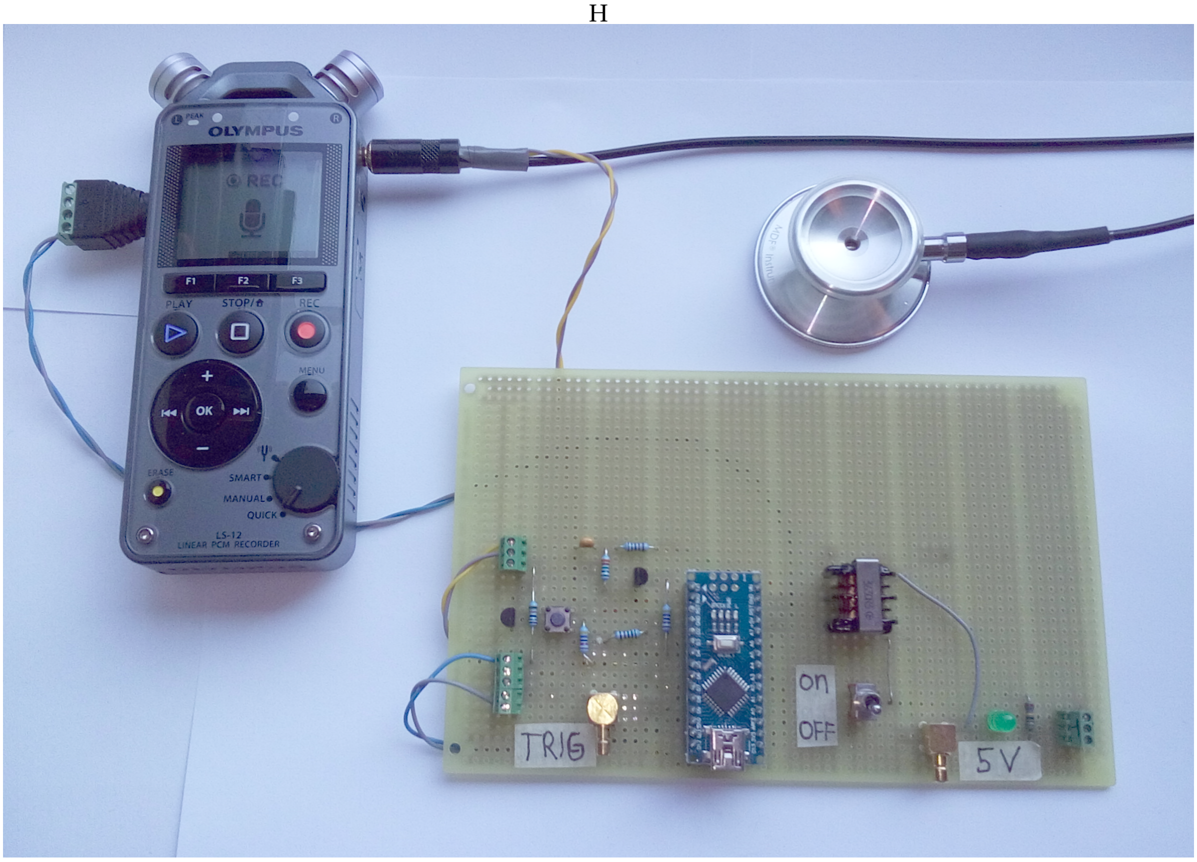


Figure 3.7: This image shows the complete system setup as discussed in chapter 3.

4
Results

4.1. Introduction

In chapter 3, the complete system is finished. In short, the system consists of a phonocardiogram recorder synchronized with the Zonare ZS3 ultrasound imager. This chapter will list the test results divided in subresults. First, the phonocardiogram recorder response is measured(paragraph 4.2). Next, different recorded phonocardiograms with the different used setups will be displayed(paragraph 4.3). The test of the synchronisation also produces results(paragraph 4.4). The results will be discussed in chapter 5.

4.2. Phonocardiogram recorder frequency response

4.2.1. Introduction

A method to quantify a microphone is by measuring its frequency response. As a stethoscope is a contact microphone, the signal source has to touch the stethoscope head physically. This is done using a vibration generator. The system setup can be found in figure 4.1.
The recorder head was put in a stand while just resting on the moving element of a mechanical vibration generator. Frequency sweeps of 20 to 1000 Hz were executed to get the system's response. The Olympus recorder is set at a recording level of 25 %. This percentage followed from several phonocardiograms, setting the level to 25 % was low enough to prevent clipping for the different test cases (the different auscultation locations, paragraph 1.2.8. The voltage amplitude input of the vibration generator was set at 180 mVpp, a higher setting resulted in clipping. A sweep was done from 20 to 1000 Hz. As discussed earlier, most phonocardiogram information will be up to 500 Hz. Therefore frequencies up to 500 Hz are displayed.

4.2.2. Background noise measurement

To be sure about the additional noise from the surrounding area entering the stethoscope, tests were conducted to quantify the noise. For this, the recorder was put on various surfaces and the frequency spectrum was measured. In the initial vibration test with the vibration generator put on the table, interference heavily changes the spectrum of the recording. To verify this, the stethoscope head was put on the table without the vibration generator. As can be seen in figure 4.2, significant interference frequencies are present from up to 200 Hz. To counter this, a 3 cm layer of vibration-absorbing foam was used, resulting in the spectrum in figure 4.3. Not all noise is eliminated by the foam: the most significant amplitude decrease is achieved in the 50-200 Hz region.

4.2.3. Frequency sweeps using vibration generator

Based on the result in the previous section, foam was used under the vibration generator. Sweeps were done from 20-1000 Hz using the waveform generator with an output amplitude of 180 mVpp. The Olympus LS12 supports various limiter modes. The limiter mode should be correctly set to ensure that each frequency is reconstructed without amplitude deformation. Plots of changing the limiter setting to 'voice' or 'no limit' can be found in appendix A. Based on that test, the 'no limiter' option was chosen. Figure 4.4 shows the time domain of the sweep. In a time span of 20 seconds, the frequency increases from 20 Hz to 1000 Hz. The amplitude stays level after the frequency becomes 100 Hz. Figure 4.5 shows the frequency spectrum of the

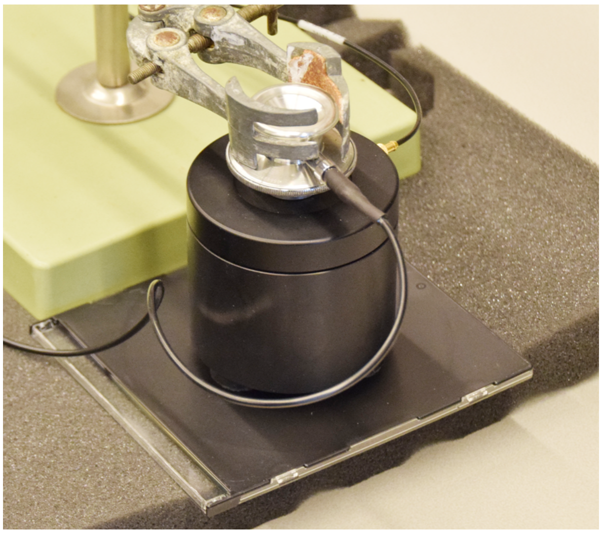


Figure 4.1: This figure shows the vibration generator measurement setup. The vibration generator is right below the stethoscope head membrane, the stethoscope head is placed inside a clamp that is connected to a stand. The foam (used in some test) is also visible.

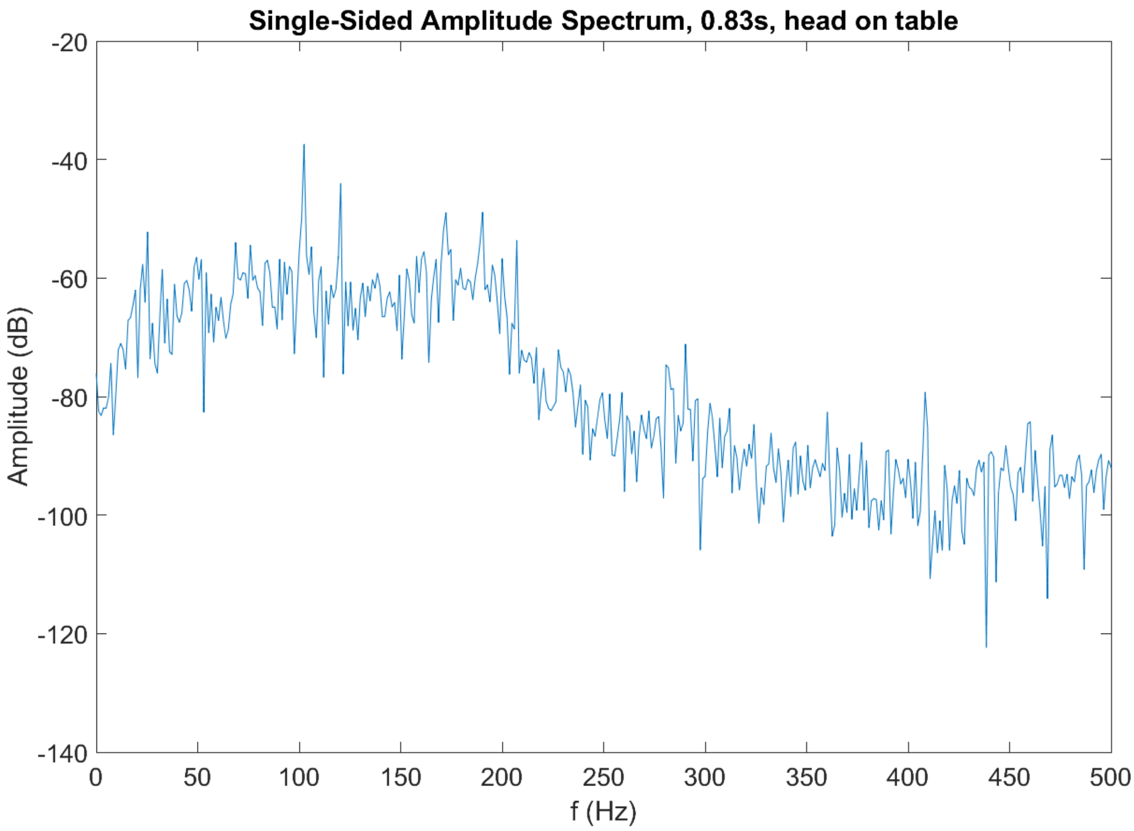


Figure 4.2: FFT of the signal recorded with the stethoscope head placed on the same table on which all measurements were executed. The relevant frequency range of 0-500 Hz is shown.

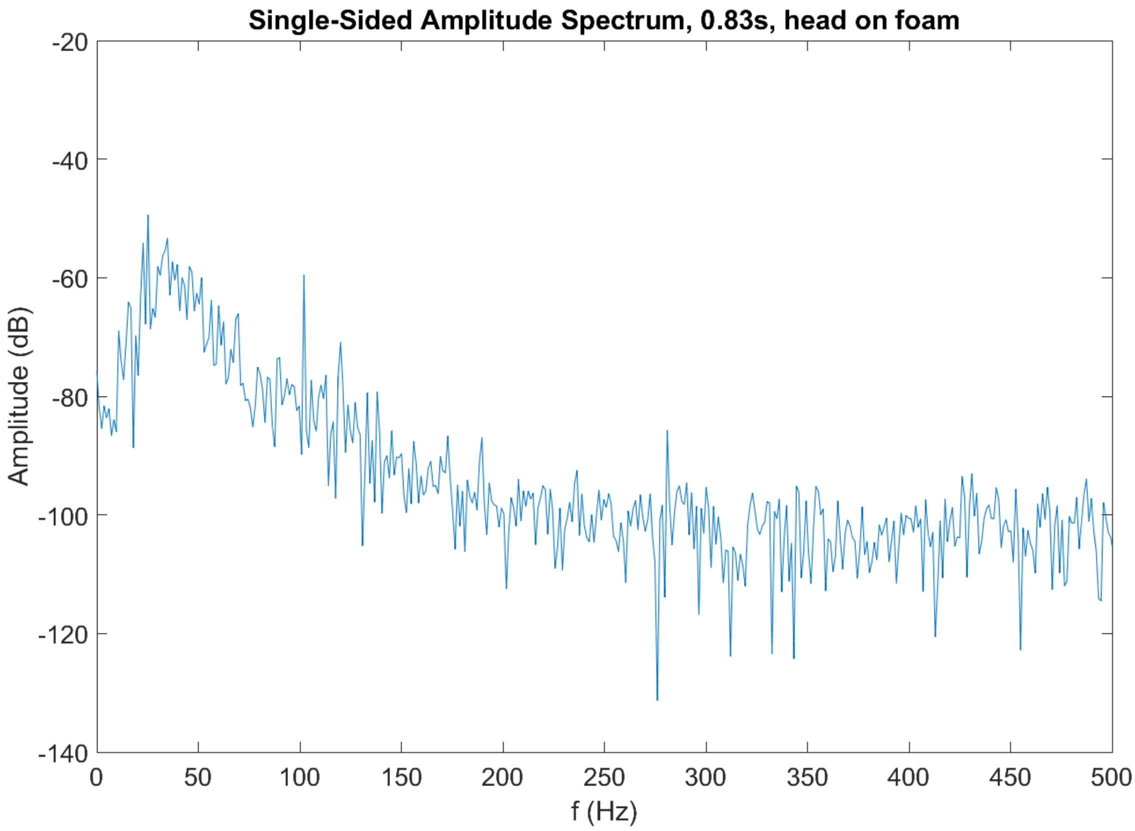


Figure 4.3: FFT of the signal recorded with the stethoscope head placed on foam on the same table on which all measurements were executed. The relevant frequency range of 0-500 Hz is shown.

same sweep. It is seen that the response is not flat, has a peak at 250 Hz and drops with 20 dB at 500 Hz. Assuming the vibration generator has a flat response, the recorder system has a low-pass behaviour.

4.3. Phonocardiogram results

The frequency response of the recorder system is evaluated by the fft. This evaluation is simplified by assuming the signal is a cyclic signal: each following heart cycle is assumed to be nearly identical to the previous heart cycle. Therefore, the fft can be executed over one heart cycle. Both the custom stethoscope head and the Thinklabs One recorder have been used for the measurements. Time domain plots can be found in figure 4.6 for the custom stethoscope head and figure 4.8 for the Thinklabs One, while frequency spectra can be found in figure 4.7 and figure 4.9. The amplitude of the time domains are scaled to the maximum recorder amplitude (referring back to paragraph 1.2.6). One heart beat is used for the fft, as the signal is considered to be repetitive on a short term and using more than one heart beat for the fft causes false Diraq pulses in the spectrum.
In figure 4.5, a frequency spectrum is visible that is made with a vibration generator. Assuming the generator has a flat frequency response, figure 4.5 is a measure for how the recorder system reacts on any sound. So when other sound is recorded, the new recorded sound can be corrected with the now known recorder response. As this work mainly focuses on the system hardware, the sound signal correction has been left outside this thesis. That is why figure 4.7 contains an uncorrected fft of one heart beat.

Using equation (2.1), figure 4.6 and figure 4.8, the dynamic ranges can be determined. For the custom stethoscope head recorder, the dynamic range is 16.5 dB, for the Thinklabs One recorder, the dynamic range is 22.5 dB. This number regards the unprocessed signal and is based on five different phonocardiogram recordings per recorder option.

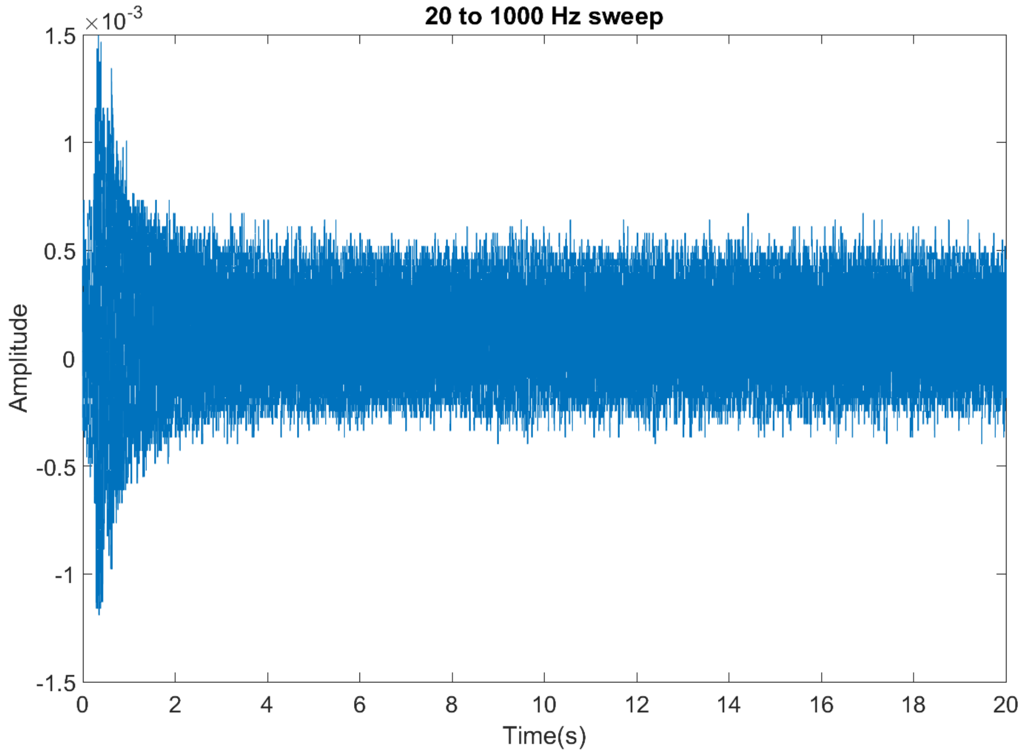


Figure 4.4: Time sweep from 20 to 1000 Hz using the mechanical vibration generator. The plot shows amplitude versus time. The frequency linearly increases from 20 to 1000 Hz in a time span of 20 seconds.

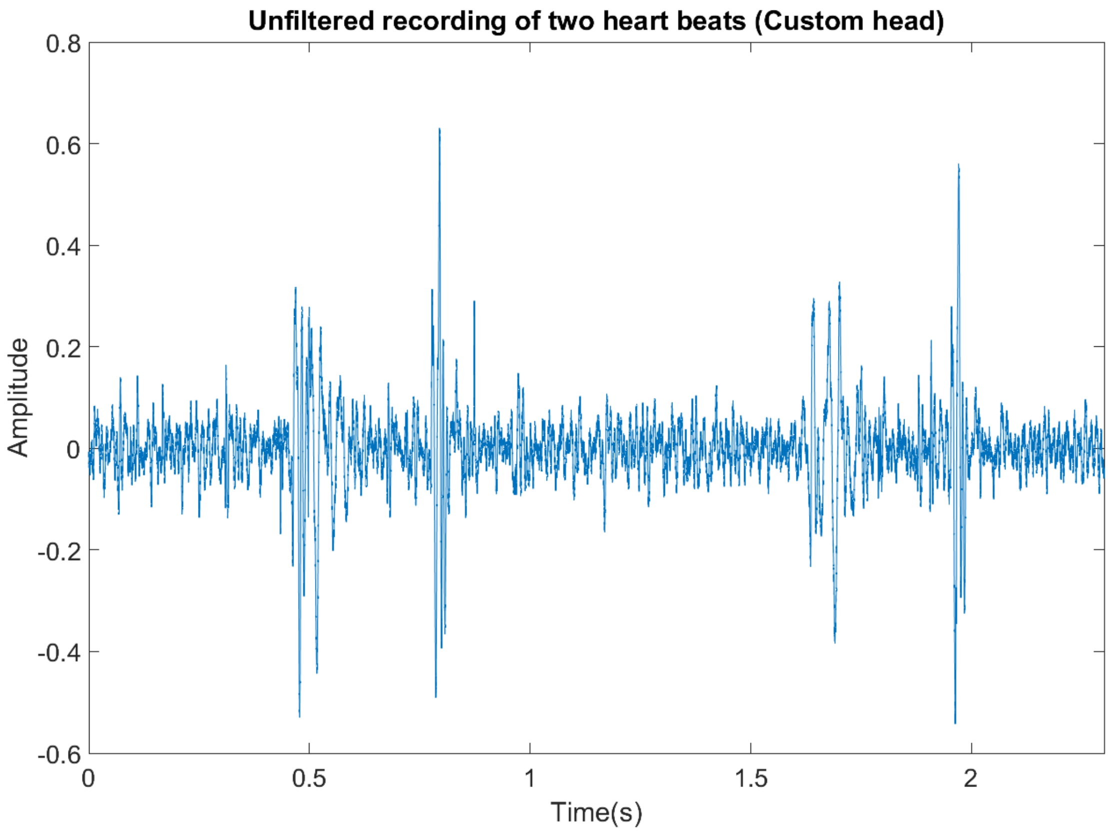


Figure 4.6: This figure shows the time domain of two heart beats recorded with the custom stethoscope head. Based on this data, also the dynamic range is determined. This signal is not yet filtered or corrected for the recorder response.

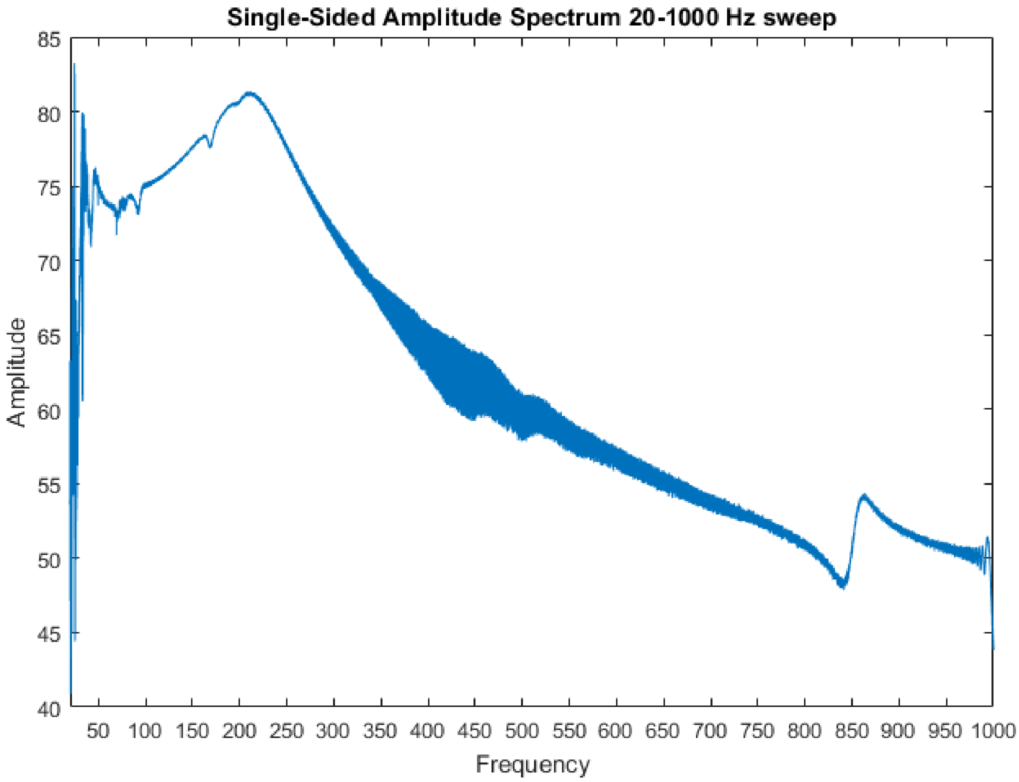


Figure 4.5: Frequency spectrum of the 20-1000 Hz sweep using the setup in figure 4.1. The plot shows amplitude versus frequency. A low-pass filter characteristic is visible.

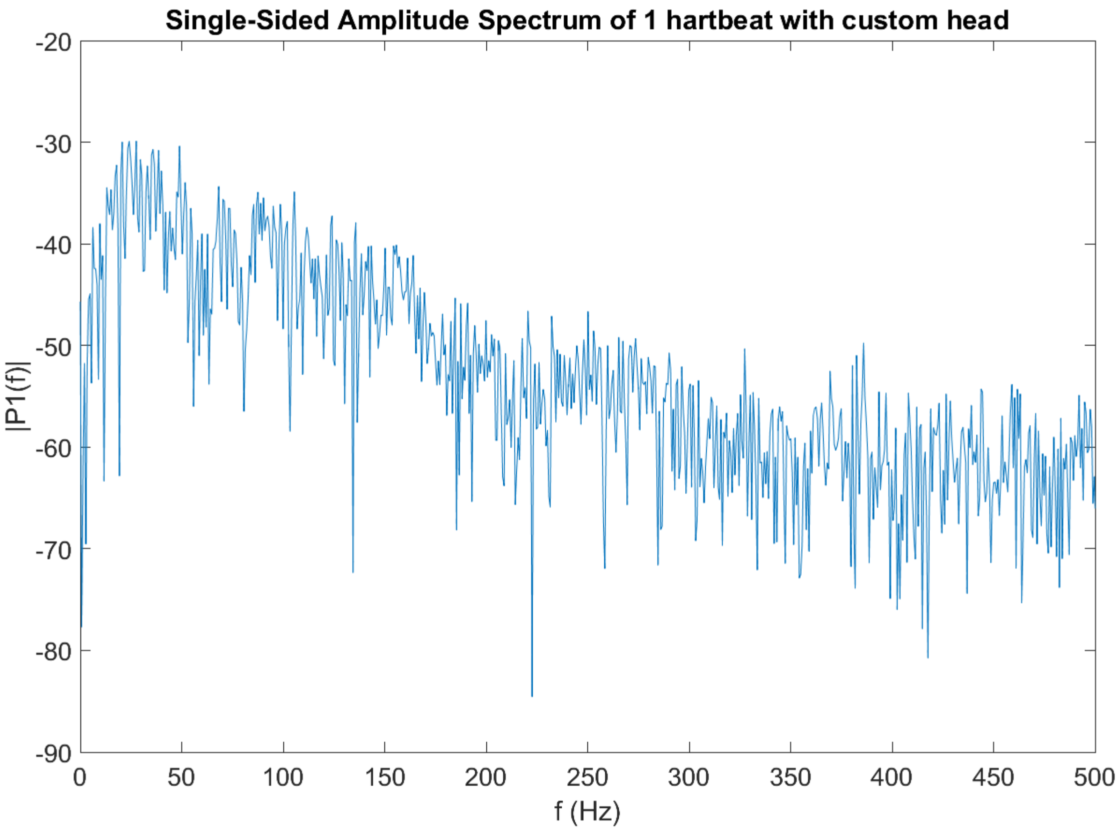


Figure 4.7: The spectrum of 1 heart beat recorded with the custom stethoscope head. The relevant frequency range of 0-500 Hz is shown.

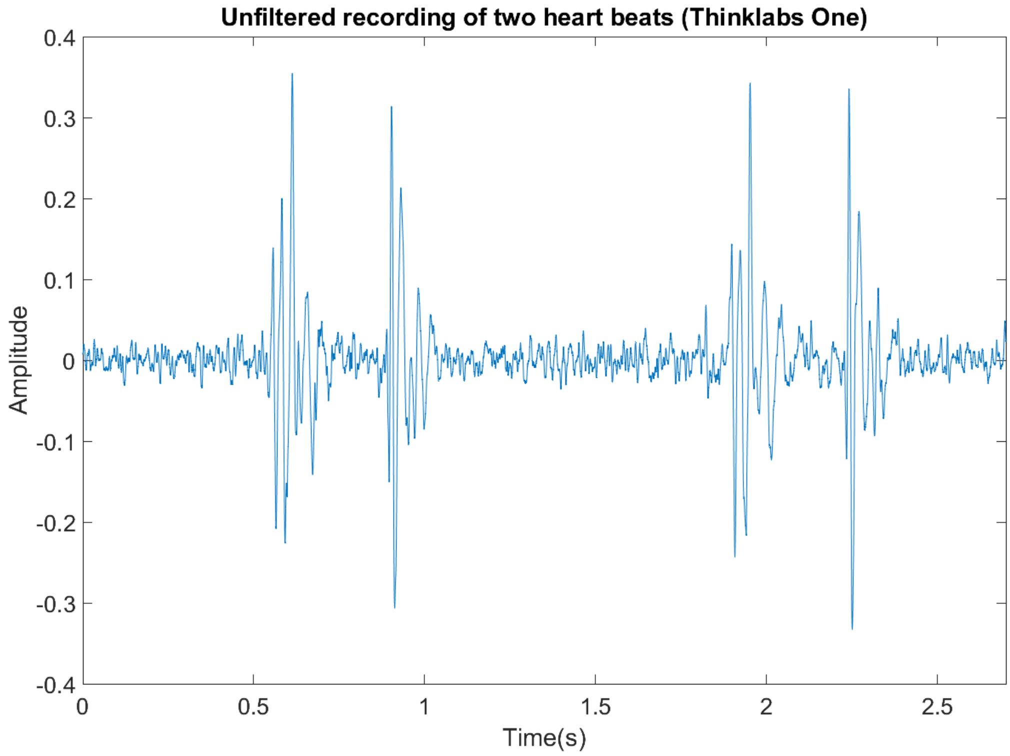


Figure 4.8: Theis figure shows the time domain of two heart beats recorded with the Thinklabs One stethoscope head.

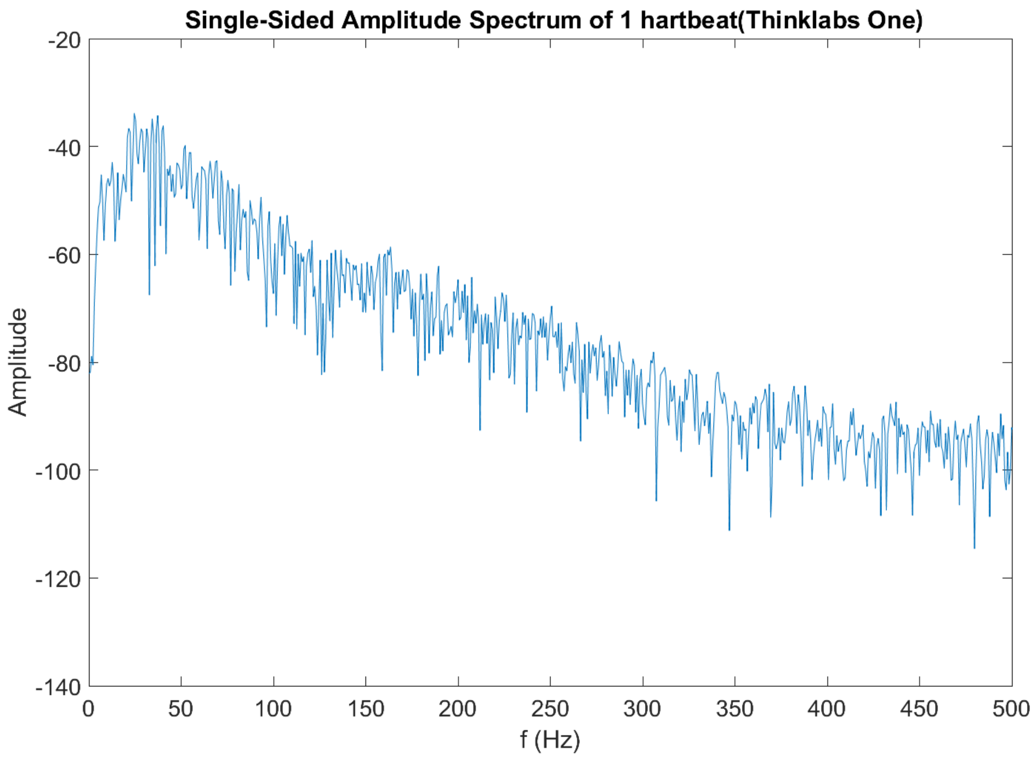


Figure 4.9: The spectrum of one heart beat, recorded with the Thinklabs One recorder. The relevant frequency range of 0-500 Hz is shown. This signal is not yet filtered or corrected for the recorder response.

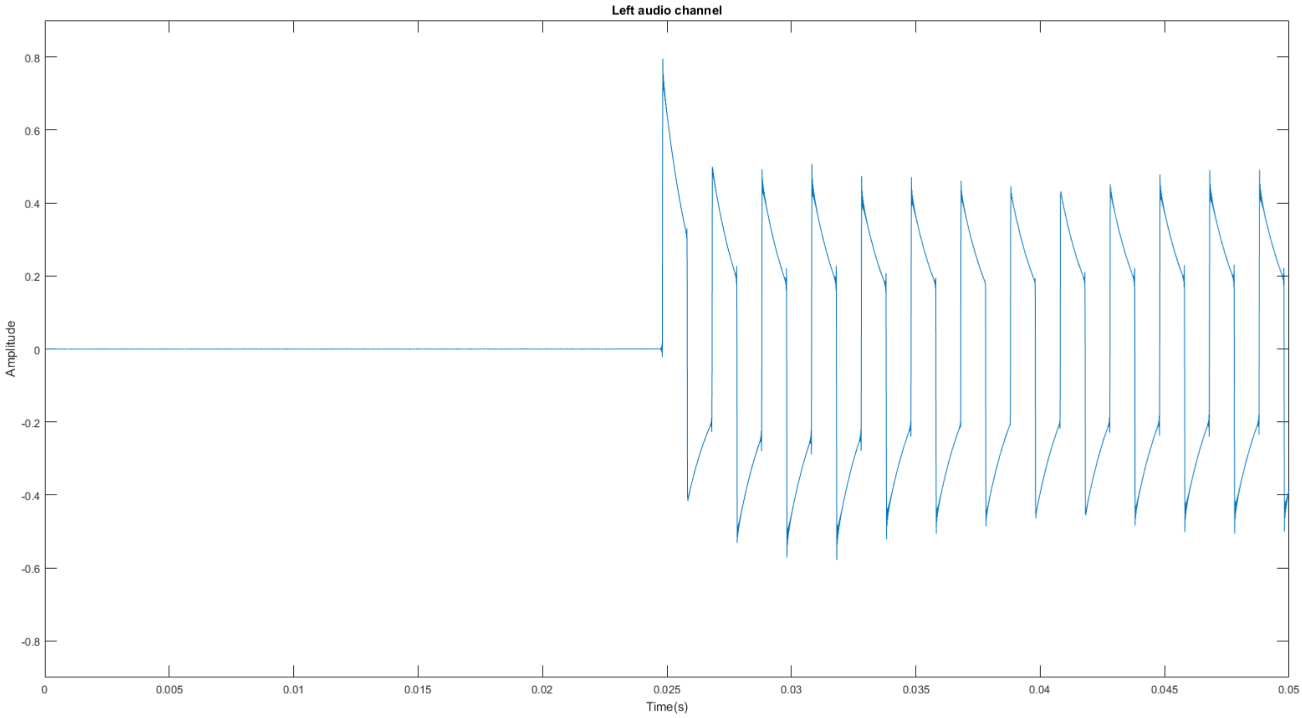


Figure 4.10: This figure shows the safe delay that the synchronization setup uses to start the audio recording synchronous with the ultrasound recording: it makes sure the audio is first running, then triggers the Zonare. The signal in the plot is not the actual signal, but a scaled down and capacitively coupled signal to meet the audio recorder maximum voltage levels.

4.4. Verifying the synchronization

To evaluate the quality of the synchronisation of ultrasound images with the phonocardiogram recorder, an event needs to be generated that can be recorded by both the ultrasound imager and the sound recorder. A clear event is generated in both recorders when the the Zonare probe probe with the stethoscope head(with soms ultrasound gel applied to the stethoscope head). On the ultrasound image, the gel-covered head of the stethoscope looks like a small white spot which appears on the tapping time, while in the audio data, a sharp amplitude peak is seen as the tapping translates into a sound with a large amplitude. The following test procedure is used for the tapping test:

1. Send command `ptrig start` to ATmega328
2. Tap with the stethoscope head on Zonare transducer
3. Press FREEZE button on Zonare
4. Save the Zonare images that were recorded during 1-3.

To check when the triggering has started, the trigger signal is scaled down by a resistor network and capacitively coupled to the left audio channel. The start of each triggering sequence can be seen in figure 4.10. The plot shows that the triggering does not start immediately. As said earlier, this is because the Olympus LS12 has a debounce delay of approximately 550 ms and it should be fully sure that the audio is running on the moment that ultrasound imaging triggering starts. It was found out by checking various sound samples that this debounce delay of 550 ms has some variation, making consistent open loop control impossible. To solve these delay variations, a safe delay between audio trigger start and ultrasound trigger start was built in of 650 ms. The program code can be found in appendix E.

The 'safe delay' makes sure the audio recorder is always running when ultrasound triggering starts. Because no triggers are sent in this period, the corresponding time stamp pause in the IQ data can be detected and used to calculate where the ultrasound file starts. Based on the time of the first pulse, the audio file is cut to

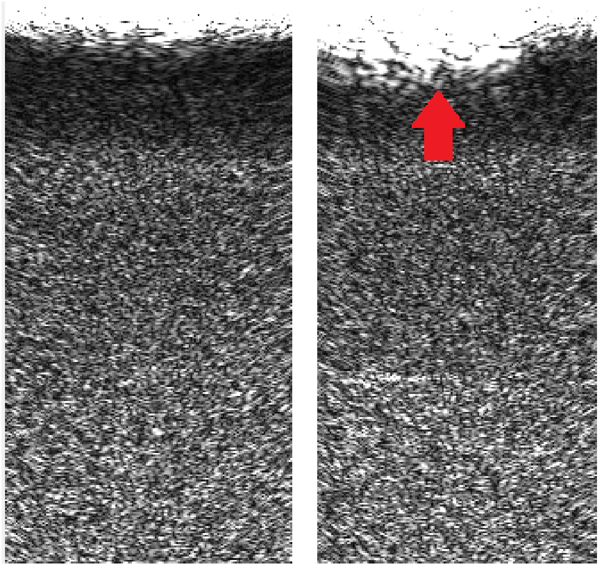


Figure 4.11: This figure shows the ultrasound event of the verification

the exact length of the ultrasound imaging video file using an automatic Matlab script.

In figure 4.10, the audio event is displayed. According to paragraph 3.5.2, also an ultrasound imaging event is required to verify the synchronization. This event is displayed in figure 4.11. The test in figure 3.5 was executed 13 times to get a statistically significant dataset. The Zonare was triggered with a frequency of 1000 Hz, thus the resolution of this test is 1 ms. The audio recorder is configured at a recording frequency of 96000 samples per second. The resulting dataset is shown in figure 4.12.

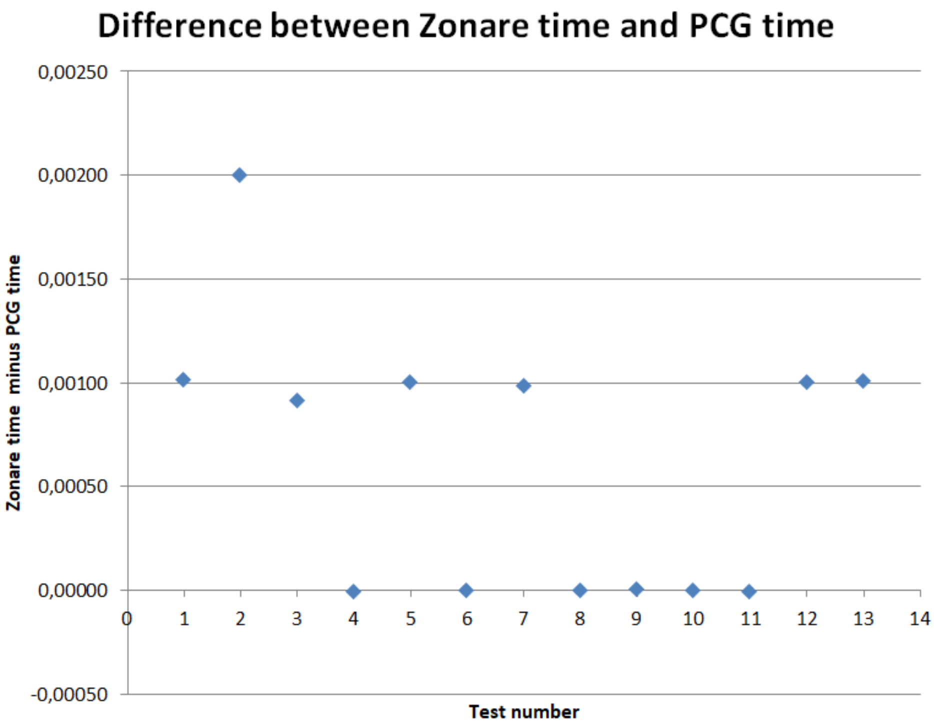


Figure 4.12: The result of 13 tap tests.

4.5. Final note on chapter 4

In this chapter, the relevant results of the system were displayed. The main focus was to get a result on the basis of which can be determined whether system meets the requirements. The next chapter will discuss the results and advise on how to improve the system.

5
Discussion

5.1. Introduction

In chapter 2, the system requirements were discussed. In chapter 3, the system setup is explained, following by chapter 4 that displays the results of the system. The interpretation of the results (with the system requirements in mind) is what forms the discussion on the built system. This discussion of results versus requirements will lead to final system *specifications*. This chapter will start with a discussion of the system requirements one by one. During the build of the system, some possible additional solutions were found for which time was too short but that would improve the system. Therefore a section is added that discusses possible future improvements of the system.

5.2. Comparing the system with the functional requirements

5.2.1. Introduction

The structure of the following sections is based on the requirements list structure in paragraph 2.3. When the system requirements are met, the requirement converts into a system specification. The requirements chapter starts with functional requirements(paragraph 2.3.1): actions that the system has to do. These requirements can be checked with information from the system measurement setup (chapter 3). After this section, the reader has an overview of the final achieved system functionality.

5.2.2. The system has to record a phonocardiogram and save it on non-erasable storage

The phonocardiogram part of the system comprises of a custom built stethoscope head with an in-built microphone and the Olympus LS12 voice recorder to digitize and store the analog signal. Inside the stethoscope head, a 4 mm Projects Unlimited TOM-1545P-R electret microphone is placed. According to the placement of the microphone: two setups were built as visible in figure 3.1. The same heart sound measurement was executed with both versions. It pointed out that version B together with the same digitizer produced heart sounds at an amplitude at least 2 times lower compared to version A. Also, version A of the stethoscope still allows the physician to use the device in the classic analog way, (bell and diaphraghm mode), by rotating the head around the tube axle. This is entirely posible with version A, as the microphone is in the center but impossible with version B. These drawbacks caused the choice for option A. Another choice was to use the Thinklabs One recorder as the stethoscope head. As it does part of the filtering already, data processing capacity is saved. Based on **??**, it is able to deliver very good performance compared to the time domain amplitude characteristics of the custom-built stethoscope head(figure 4.6). However, the Thinklabs One has some drawbacks. As it has its own battery, it has to be recharged after every four listening hours. Charging an extra device is not in line with the ease of use that this work aims to achieve. Furthermore, its $499 price tag uses half the project budget, which might be risky. Paragraph 5.3.4 will elaborate on this. Another problem of the Thinklabs One while using and testing is its build quality. The 3.5 mm jack plug has to be inserted with considerable force to make a proper connection and the jack socket still allows some yaw movement of the plug after insertion. When used by many different people that expectedly do not have the finest feeling with technology, this could quickly end up with a broken socket. Also, the jack socket is located on the Thinklabs One, which is very close to the patient. Potentially, medical wast or pathogenes could end up inside the socket, causing problems for other patients. Instead, the custom stethoscope head is powered by the LS12 digitizer, has no jack socket close to the patient and keeps the original sleek shape, making the device easy to clean. For these reasons, a custom stethoscope head was chosen over the Thinklabs One stethoscope head. However, for occasional research use and to check this design, the Thinklabs One would be a good option as it is compatible with the LS12 digizer, which makes integration into the system easier.

Other ways of recording and storing the signal were evaluated as well. In section paragraph 1.3.1, various options are already mentioned. During the project, another solution was found to the third option(using a custom stethoscope head with external digitizer). The reasoning: if the sound signal has to be digitized, why not use a potent microcomputer as the digitizer and use it to send synchronisation signals to the Zonare ZS3 as well? The Raspberry Pi would be the microcomputer of choice due to its budget-friendly prize and small size. The problem found with this solution is the expected sound quality versus synchronisation speed. Not all USB sound cards work together with the Raspberri Pi and their high frequency recording needs might consume too much processor time to accurately send trigger signals to the Zonare ZS.3. As the outcomings of this option were unsure and the possible additional project time even more unsure, it was decided to build a working prototype with the available components instead while still remembering the idea for possible future system improvements.

Using the reasoning above, the final system comprises of a custom built stethoscope head that produces an analog signal, which is digitized and saved to a .wav file by the Olympus LS12 audio recorder.

5.2.3. The sound file should be exportable to a PC

Currently, the sound file is saved on the internal memory of the Olympus LS12 voice recorder. It connects to a PC over USB and the transfer and Matlab processing is still a manual process. So the requirement is met, but there is still room to improve the setup. The recorder memory is accessible from a computer. A great improvement would be if for example the Raspberri Pi from the previous paragraph could access the LS12 memory and grab the right audio file based on the audio file time stamp. This would depend on the choice of USB sound card, using the LS12 as controlled sound recorder is possible with the Raspberri Pi(using the GPIO pins). In the current setup with the atmega328 chip starting and stopping the audio file, there is no control over the file after it is saved. The Raspberri Pi would be able to read the file and modify/send it to a specific location. Currently, for the prototype version, the file is saved to the memory and sent over USB to the computer for processing.

5.2.4. The recorded sound should be synchronized with high framerate ultrasound images

This section talks qualitatively about the synchronization of the Zonare ZS.3 and the LS12 audio recorder, while paragraph 5.3.2 will use a quantitative approach. The concept of synchronization in this system is remembering when Zonare triggers start and stop. The Zonare trigger signal is converted to a signal with lower amplitude first and then applied onto on the of the recorder audio channels. When the recorder starts recording, it is always visible when Zonare triggers started and stopped. Using FREEZE commands, the ultrasound images are stored on the internal Zonare memory. Using the trigger information on the left audio stereo channel, the phonocardiogram signal on the right stereo channel can be cut to the exact length that the trigger was on. Currently, the FREEZE commands are inserted with Zonare buttons. It is possible to use the USB connector to send the same commands. It was also found that the Zonare can use its internal trigger for high frame rate purposes and can be controlled by the USB FREEZE commands to stop and start a recording. Although there will be the delay of the software going through a sequence of commands first, it is expected by Zonare employees that the delay will never be larger than a few milliseconds. This approach will save the designer a lot of work, as accurate trigger signal generation is not needed any more. Instead, a device with a USB port suffices. On the project, this option was not chosen as the delay was expected to be uncertain. However, if there is a device with a USB connection, like a Raspberri Pi, this option might be feasible. Storing IQ files on the Zonare ZS.3 also happens with terminal commands. Automatic downloads to PC in combination with automatic audio sound file PC uploads could become possible (see paragraph 5.2.3). For now, the system works with manual freezing and storing. The triggering starts with a terminal command to the atmega 328 chip, but can also be activated using a push button on the circuit board.

5.2.5. The system will be connected onto the ultrasound system

The requirement of relatively small component sizes comes from this functional requirement. The currently expected size of the system is 15x15x8 cm, which could be decreased by a smaller custom-made PCB and better packaging of the digitizer. The Zonare ZS.3 provides ample space for systems with these dimensions.

5.2.6. The system must be able to record cardiac ultrasonic images

In short, the chosen Zonare ZS3 medical imager records high frame rate images and stores them on its internal hard drive or on a USB drive. No hardware on the Zonare ZS.3 was changed for this project as it was the system component to stay untouched. Therefore this requirement is achieved.

5.2.7. The system should be easy to use for non-technical users

As possible users of the system are new to it, the system should be quick to use. The current actions needed for to create a synchronized recording are:

1. Using a text command or push button to start the measuring sequence
2. Using a text command or push button to stop the measuring sequence
3. Copying the audio file from the Olympus LS12 recorder to PC
4. Copying the IQ file from the Zonare ZS.3 to PC.
5. Running a Matlab script to generate the video file.
6. Running a Matlab script to generate the audio file.
7. Combining the video and audio files using Matlab or another editor.

In this sequence, step 1 and 2 can be combined with the current system - given that the phsysician agrees on a fixed recording time. Executing step 3 and 4 will be most of the work. A microcomputer like the Raspberri pi could do the job automatically. As for step 5-7, these should acutally be combined into one Matlab script but doing so was a challenge to finish inside the project time. It was decided to leave it as the result already delivered a good proof of concept. Regarding 'ease of use', also battery life is important. According to the product manual, the LS12 recorder should be able to make a recording of 40 minutes on the 96 kHz PCM mode (the most high-quality mode). When the external SD card slot with a 32 GB SD card is used, the recording time jumps to 14 hours and 30 minutes. The battery life in the most economical setting is 50 hours in recording mode and 62 to 86 hours in voice file playback. The recording times from the manual suggest that the recorder will be usable for a full working day, which adds to the ease of use.

5.2.8. The part of the system that is in contact with the patient should be able to be desinfected with a wet towel and a desinfectant

This requirement was easy to match as a standard stethoscope head was used to construct the head. No precautions have to be followed regarding cleaning of this part of the system: the 4 mm microphone is embedded in epoxy.

5.2.9. The system has a complete galvanical isolation

To prevent electric shocks, a first measure is to avoid system connections to grounded wires. The audio recorder is connected to the Olympus LS12 only, which is battery powered. However, the LS12 audio recorder ground is connected to the central microcontroller ground as well, to make triggering possible. The safest option then is to connect the power and ground of the microcontroller to a device that adheres to the IEC 60601, Class I powered equipment standard already. The Zonare ZS.3 adheres to the IEC 60601 standard [61] and has a USB port that is able to provide the current needed by the current system. This means the recorder-microcontroller is safe to use.

5.2.10. The system must have an option to make the phonocardiogram live audible

As soon as the recording starts, the Olympus LS12 sends the recorded sounds to a 3.5 mm jack output port that can be used for headphones. Therefore, the requirement is met. When another system is used that uses for example a Raspberry Pi, this gets harder as the Raspberry Pi then should generate a sound output signal.

5.3. Comparing the system with the non-functional requirements

5.3.1. Introduction

The requirements chapter continues with non-functional requirements(paragraph 2.3.2): requirements that link to a specific system performance item. These requirements can be checked with the data from the system results (chapter 4). After this section, the reader has an overview of the final achieved system performance and knows about possible improvements to the current system.

5.3.2. The timing error of the sound recorder cannot exceed $\pm$ 0.5 ms

The total system records a phonocardiogram, processes IQ data into a video and synchronizes the audio file with the video file. To confirm that the synchronization is indeed working, a test was executed that generated an phonocardiogram event and an Zonare ZS3 event. This was done by hitting the ultrasound transducer with the stethoscope head covered in ultrasound gel. This caused a visible reaction on the ultrasound screen and a clearly visible audio event on the phonocardiogram. The time that the event started was retrieved from both audio and video and the difference listed in figure 4.12. The possible time differences are discrete intervals of about 1 ms, as the Zonare records up to 1000 frames per second. It can be seen that for half of the cases, the error is about 1 millisecond and that the error is zero is the other half of the cases. However, this 1 ms error might not be a system error but a processing error. The detection and testing method was executed manually and the time at which a blip was believed to appear in the screen was rounded to the nearest ultrasound frame start time. The blip was not always visible at the moment of impact due to thin application of ultrasound gel. Because of the above reasons, there could be a start time of 1 ms later due to the visual detection error. Based on this data, the maximum synchronisation timing error to be detected is 1 ms, but the result might be influenced by the fact that the ultrasound imager intervals are no shorter than 1 millisecond. A way of improving this test is thus to use an even higher frame rate. In experimenting with the system using the research mode, it was found that the system for specific settings allows for 1150 frames per second, but was too unstable to enable a measurement like this. However, the Zonare ZS.3 is still a relatively new system. In the future, the frame rate will be higher and the synchronisation error estimate will get a better. For phonocardiograms used in shear wave elastography, the 1 ms uncertainty is still small compared to heart sounds (>100 ms) and thus acceptible.

5.3.3. The sample rate of the recorder is >1 kHz

The LS12 voice recorder has a recording frequency of 96 kHz[57] and is an off-the shelf component. As a device, it is actually way too general for use in a final system. It was chosen for its strong low-frequency performance and intuitive interface. But as said before, the focus of this system was to build a complete working system, not spending all time on meticulously improving system components. The replacement part would be an USB sound card or a dedicated PCB with a low noise amplifier, ADC and a microcontroller for sampling and storing the signal. The USB sound card offers a better control over starting and stopping an audio recorder, while the custom PCB even offers adjustability of parameters like size, power usage, signal quality at the expense of consuming more time. For now, the system has to trigger the LS-12 voice recorder with its debounce delay of about 550 milliseconds, which is a drawback for synchronisation. The above mentioned options could eliminate the debounce delay in newer designs.

5.3.4. The system has a maximum cost of €1000

System cost is a topic of interest, as the concept might be used in various other hospitals in Rotterdam, asking for easy construction and cost-effective system components. Therefore, a system cost of €1000 was specified. This budget is used for the phonocardiogram system and the synchronizing system. The Zonare ZS.3 was already available at the moment of the research and caused no additional project cost. The costs are listed in table 5.1.

The 'various PCB components' item contains the price of the correct interface connectors, a prototyping board and electrical components. The PCB cost is relatively low at the moment, as the PCB is hand made. When the production has to be scaled up, the PCB has to be orderd at a PCB making company. It is expected that at least three PCBs have to be made. The expected cost per manufactured PCB is €30, this value is based on a short search between commonly used PCB manufacturers. The €10 in components then still adds to the €30. However, the decision could be made to eleminate the Arduino Nano and instead built a Raspberri

Table 5.1: System cost

Component	Price (€)
PCB components	10.00
Arduino Nano	16.29
Olympus LS-12	159.00
MDF® Acoustica® Lightweight Dual Head Stethoscope	20.00
TOM-1545P-R electret microphone	4.33
Zonare connection trigger cable	0.00
Total cost	**209.62**

Pi interface, this changes the cost perspective as a Raspberri Pi kit costs €56. The prices above are based on 2018 Farnell component prices. The custom cable to trigger the Zonare with hardware triggers will be expensive, but Zonare sponsored this part of the research. Looking at this figure, there seems to be a lot more budget for development. The $499 Thinklabs One recorder could be added, but the drawback of the Thinklabs is recharging one more device and turning on one more devic every time the ultrasound imager system is started. One of the key requirements of this system is the ease of use, the Thinklabs One does not add to an overall better ease of use so this solution is impossible.

5.3.5. The process of synchronizing the phonocardiogram audio signals with the ultrasound images consists of a maximum of 5 steps

This requirement was already discussed in paragraph 5.2.7, it followed that when the current existing Matlab scripts are combined for the prototype, executing the complete process in 5 steps is possible. This is not the case yet, as the IQ data processing code is updated over time. A possible improvement could be the addition of buttons and the writing of custom PC software that automatically reads the respective files and processes it into a video with audio using Matlab scripts.

5.3.6. The system should work with high frame rates of 900-1000 frames per second

As high frame rate is an important main point for this research, the system has to handle the speed. Currently, the audio sampling speed is 96 kHz, while the frames are recorded at 1000 frames per second, which is a sampling speed of 1 kHz. Because the processing does not happen in realtime but afterwards, the recording and synchronisation can go up to much higher speeds. As the hardware triggers are recorded directly on the audio recorder, the system is always sure when the ultrasound imaging started. The only error when speeding up the system comes from the Zonare reaction time, as the ultrasound imager will need some time to respond to a trigger. In paragraph 5.3.2, the test was discussed that measured the maximum synchronisation error for a hardware triggered system. Currently, the system is hardware triggered, but the Zonare can also be triggered using USB control. According to Zonare employees, the response delay of the the USB controlled Zonare core is in the order of a few milliseconds. When the frame rate is increased using USB control commands, this delay could become a concern. It should be noted here that the Olympus LS12 voice recorder also appeared to have a significant response delay that was succesfully canceled out. Similarly, this might then be needed for the Zonare triggers. To conclude, the system is built and tested with 1000 frames per second, so this requirement is met.

5.3.7. Dynamic range

In paragraph 4.3, it was found that the dynamic range for the custom-built stethoscope head is 16.5 dB while the Thinklabs One delivered a DR of 22.5 dB. The requirement, based on literature, was 18 dB. Taken in mind that the sample size is five, these numbers still show a difference. Choosing the Thinklabs One deliver more signal quality, but might be less practical in a clinical setting, as it has a small battery that needs to be recharged relatively often. Furthermore, its cost consumes nearly half the project budget, which might hinder scaling up the product for bigger sales. However, customizing stethoscope heads is a labour costly process too. For research, where maximum dynamic range is needed to catch all signal details, the Thinklabs One might be the right choice, where the custom head might be the durable long-term choice as both recorder options are able to detect S1 and S2. A further check to enrich the results is to measure deseased hearts with an available S3 and/or S4 to verify the dynamic range for those signals.

5.4. Additional points of improvement for the system

5.4.1. Introduction

In paragraph 5.2 and paragraph 5.3, the functional and non-functional requirements were separately discussed and possible points of improvement regarding the specific requirement were proposed. However, during the design process some new options to the built solutions were found that would improve a future system. This section addresses those details.

5.4.2. Points of improvement

To begin with, the system meets the predefined requirements and works around a subsystem with variable delay behaviour(The random delay of the LS12 voice recorder). However, the central microcontroller does not have USB support when it is generating hardware triggers, as all external ports are shut off to avoid system jitter. Moreover, the audio recorder is used currently as a stand-alone device with slow control. The current solution is only a workaround and based on post-processing of the signal. A much 'cleaner' solution would be the audio recorder and the Zonare starting simultaneously, but it has been found that this is hard to achieve due to different response times of completely different systems. When a faster recorder is used, like a USB sound recorder, the sytem setup has to change to a mini computer like the Raspberri Pi, then again the Zonare triggering system has to be reworked. Actually, a solution with the Raspberri Pi might be more flexible regarding sytem compatibility to other ultrasound imagers. The current system, while relatively simple, is made with one specific hardware output for which other ultrasound imagers might have no input. However, the current system has simplicity as its strong point. Debugging time for this project was minimal, writing the microcontroller software is easily understood. This is important for a new system in an environment with few trained computer programmers available. The last question to discuss would be how the current system could be improved: the answers lies in further developing and finetuning of the used software. Scripts that were separate can be merged, file transfer can be more automated, placing the sound files in the ultrasound video has to be automated. Another point of improvement regards the used microphone. In this case, the smallest microphone on sale was chosen, to fit the stethoscope shaft. The microphone was fit for recording from 20 Hz, but the sensitivity was not a parameter that could be changed. A larger design of microphone would mean a different choice of stethoscope head, but would also give more options in choosing a microphone, thus giving the designer more control over the final phonocardiogram signal. Furthermore, regarding the sweeps with the frequency vibration generator, the response of the vibrator in the frequency range was assumed to be flat. To verify this, the vibration amplitude over the frequency range needs a closer look. To conclude, the current system adheres to the requirements. However, in the design process some points of improvement were found that are expected to improve a possible future system.

5.5. Final note on Discussion

In this discussion, the system requirements were evaluated step by step, followed by advice for possible system improvements. As the designer freedom of the system is relatively large, many different choices on very diverse topics had to be made in the process. The discussion evaluates the system results and evaluates the design choices in retrospect. Because this system has relatively many open ends, the list of possible improvements and advice is extensive.

6 Conclusion

In this project, a phonocardiogram recorder was built together with a synchronisation system to synchronize the audio data with high frame rate ultrasound cardiac image frames. As many different subsystems had to be built, the expertise fields were very diverse. Custom electronic hardware works together with embedded devices and software to communicate with off-the shelf devices. Both data streams (audio and ultrasound video) are synchronized according to the predefined requirements. The sound recorder has sufficient quality to record a phonocardiogram and to distinguish between different heart sounds. The Zonare ZS.3 ultrasound imager is under development itself at the moment. This was seen sometimes: in some cased, the frame rate increased to 1150 frames per seconds, at the cost of unstable system performance. In chapter 1, it was found that there are many ways to solve the research problem of this work. The main challenge was to narrow down the number of possible solutions as early as possible in the design process. During the build, some very achievable solutions were found that also would have been possible as project parts. These have been listed in the discussion(chapter 5).

The end result is a phonocardiogram recorder system that works together with the Zonare ZS.3 at frame rates from 900-1000 Hz, synchronizes the audio file with the ultrasound images with a maximum error of 1 frame, works over a day on its own batteries and has a cost lower than €1000. It has a dynamic range of 22.5 dB (or 16.5 dB if the system has to be cheap to make) and has a selectible sample rate of up to 96000 Hz with a bit depth of 24 bits. The current solution is not ready for a clinical setting but more for a research setting: packaging the circuit board and recorder inside a waterproof enclosure still needs to be done. The complete system has also not been tested yet on real patients, but the phonocardiogram recorder as a part has been tested on a small number of humans inside the research team. Although various tests were done for this work, more tests need to be done on real patients to verify whether the system works with diseased hearts, which it is currently assumed to be able to based on the amplitudes of S3 and S3 compared to S1 and S2. The size of the microphone in the custom stethoscope head could be increased with some metal work, giving a better low frequency response. The system control could be changed to a Raspberry Pi, which makes recording and saving easier and faster to understand for new configurators and users of the system. Finally, the reliability of the system should be evaluated in the long-term, which is not within the scope of this project.

An important challenge to the build was the medical environment of the system. For example, every phonocardiogram in every human is different. Another example is the use of battery-operated devices. Therefore, to help with the design, the sytem requirements were aimed at the project goal with this question in mind: what will be the benefit of the concerned patiënt? Recording heart sounds is not the newest technology on the market, but combining it with ultrasound imaging is a new connection. Indeed, connecting the different subsystems was the hardest part of this project. Ultimately, this work will improve the knowledge about shear wave elastography, which in turn aims at early detection of imminent heart failure. In this way, this work will contribute to a better (quality of) life for us mortals.

Appendices

A. The effect of different Olympus LS12 settings on system output

This appendix is added to point out the difference between the Olympus LS12 'voice limiter' and 'no limiter' setting. The measurements are done using the vibration generator setup from figure 4.1. A waveform generator is used to generate a sinus wave with an Vpp of 250 mV and with frequencies increasing from 20-1000 Hz. As can be seen from the time domain plots in figure 1 and figure 2, the voice limiter limits frequencies of 250 Hz.

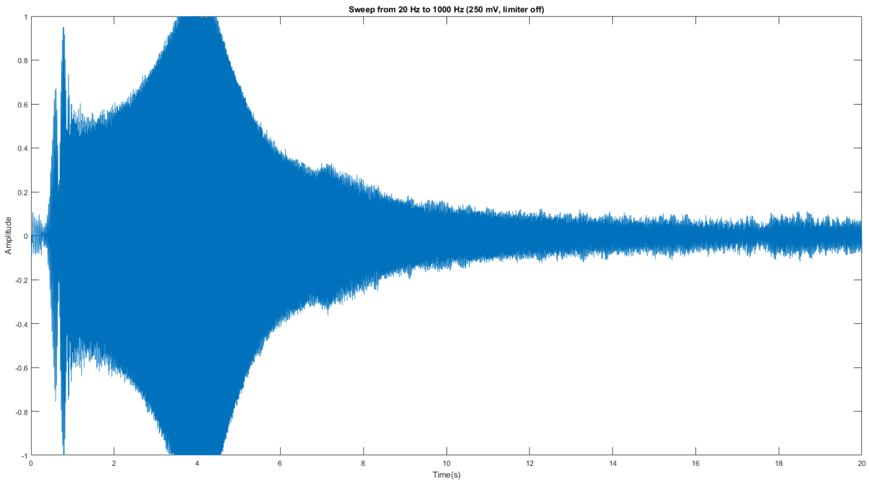


Figure 1: Sweep from 20-1000 Hz with the Olympus LS12 recorder using an amplitude of 250 mV for the vibration generator and the 'no limiter setting' on the LS12

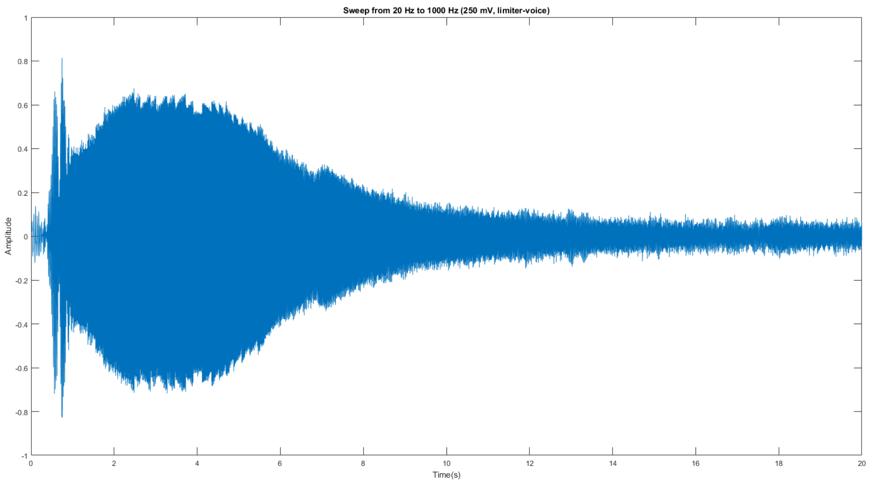


Figure 2: Sweep from 20-1000 Hz with the Olympus LS12 recorder using an amplitude of 250 mV for the vibration generator and the 'voice limiter setting' on the LS12

B. Zonare Trigger Cable

Figure 3: The cable that connects the Zonare to the custom trigger circuit. The blue and red cable are connected to the circuit, while the connector on the right (the Transducer Port Connector) is connected to the Zonare.

The Zonare document "Connecting the probe" recommends the following settings regarding the cable in figure 3[62]:

- To prevent double triggering, the pulse width of the trigger pulses should not exceed 20 μs.
- The system triggers on falling and rising edges (after the system has finished its frame recording). This means that a 1 Hz signal triggers at a rate of 2 Hz.
- The external trigger circuit is electronically isolated from the Zonare circuit by the trigger cable.
- The cable needs a regulated 5V 50mA power supply.
- The trigger signal is DC coupled and should have a voltage of 2.5V-5V.
- The trigger input does not have a 50 ohm termination.
- The cable uses around 10 mA.
- Two BNC plugs are used: the red BNC plug is the power supply plug while the blue BNC plug is the trigger input plug. The other BNC plugs are unused and contain no meaningful signals.

C. Zonare control commands and recording procedure

The commands to configure the Zonare are listed in table 1.

D. Olympus LS12 remote control input protocol

The Olympus LS12 has a 2.5 mm jack plug. This jack can be used for infrared remote control using the RS30W Olympus Remote(see figure 4) and can be wire controlled with the protocol listed in table 2[63]. Voltage levels of the user hardware do not matter in this case, as can be seen in the table: the switching actions only happen by shorting certain connections.

Table 1: USB commands that control and configure the Zonare

Nr	Command	Action
1	ipce setszi L 17200	Enables the special research mode
2	ipce set FRAME_RZOH 4000	Forces the system to achieve the highest framerate possible
3	ipce SetEnableTriggeredFrames 1	Enable the external trigger mode
4	cine unfreeze	Start recording ultrasound images
5	cine freeze	Stop recording new ultrasound images and update cine memory
6	cine store y:[filename].iq [Start] [End] 2 1	Store frames [Start]-[End] with name [filename] to USB

Table 2: The Olympus LS12 control protocol

Electric action	Button action on recorder
Connect first ring with 100k resistor to ground	START
Disconnect first ring with 100k resistor from ground	PAUSE
Connect first ring to ground briefly	STOP

E. Olympus LS12 remote control input protocol

The program code that is used on the central microcontroller is placed below. First, the interrupt timers are set. The code waits for a serial command and processes the incoming byte. The hardware timers are set by writing directly to the respective registers. As the interrupt routine call time had a length that varied randomly with one clock pulse, it was impossible to use any interrupt routine. Therefore, the microcontroller code does not enter an interrupt routine, but instead the output data register is toggled directly in hardware. The other part of the code triggers the Olympus LS12, starts the Zonare ZS3 triggering and resets those actions.

Listing 1: The source code used for the project.

```
// pin definitions
#define BUTTON 3  // the BTNPIN we are interested in
#define REC_PIN 4
#define STOP_PIN 5
#define TEST 6
#define TRIGPIN 9 // The trigger pin
#define LED 13

boolean system_on = 0;
boolean button_pressed = 0;
boolean toggle = 0;

uint32_t trigger_frequency = 200;

const unsigned int MAX_INPUT = 50;

void setup() {
pinMode(9, OUTPUT); // OC1A
pinMode(TEST, INPUT);

Serial.begin(9600);

//set timer1 interrupt
TCCR1A = 0;// set entire TCCR1A register to 0
TCCR1B = 0;// same for TCCR1B
TCNT1  = 0;//initialize counter value to 0

// Set CS bits for prescaler at 8
TCCR1B |= (1 << CS11);  // TCCR1B |= (1 << CS12) | (1 << CS10);
// turn on CTC mode (clear timer compare match)
TCCR1B |= (1 << WGM12);
// set compare match register

OCR1A = 2222;

TCCR1A |= (1 << COM1A0);   // Toggle OC1A on Compare Match and override port 9
TCCR1A &= ~(1 << COM1A0);  // Port 9 OFF*/
```

Figure 4: The IR remote that can be used to control the Olympus LS12 recorder remotely[64]

```
//TIMSK1 |= (1 << OCIE1A); // timer interrupt ON

Serial.setTimeout(200); // Serial timeout

/* delay(5000);
TCCR1A &= ~(1 << COM1A0);  // Port 9 OFF*/
}

void loop()
{
while (Serial.available () > 0)
processIncomingByte(Serial.read());
}

void process_data (const char * data)
{
uint32_t new_frequency = 0;

String datastring = String(data);
if (datastring.substring(0, 5) != "ptrig") {
Serial.print("ptrig: Invalid input. Your input '");
Serial.print(data);
Serial.println("' is not recognized.");
}

// second part of input string identifies function
String function = datastring.substring(6, 11);
if (function == "setfr") {
new_frequency = datastring.substring(12, 18).toInt();
if(new_frequency < 61){
Serial.println("Enter a frequency that is higher than 61 Hz");  // 61 hz was derived from the
     maximum value the 16 bits counter can count up to
}
else if(new_frequency > 40000){
Serial.println("Enter a frequency that is lower than 40 kHz");   // 40 khz was derived from
    maximum error in frequency due to quantized possibilities lower than 1%
}
else{
trigger_frequency = new_frequency;

double new_ocr1a = 2000000.0/trigger_frequency; // divide prescaled clock frequency by
    trigger frequency to get the new OCR1A
```

```
int ocr1a = (int)(new_ocr1a + 0.5);
Serial.print("ptrig: New trigger frequency set: ");
Serial.print(round(2000000.0/ocr1a));
Serial.println(" Hz.");

OCR1A = ocr1a;
}
}
else if (function == "start") {
Serial.println("ptrig: Trigger started, frequency: " + String(trigger_frequency) + " Hz");
start_rec();
}
else if (function == "stopt") {
Serial.println("ptrig: Trigger stopped");
stop_rec();
}
else if (function == "ttest")  {
Serial.println("ptrig: Executing special time delay tester");
start_rec();
sdelay(500);
stop_rec();
}
else {
Serial.println("ptrig: Invalid input. Your input '" + String(data) + "' is not recognized." )
    ;
}
}

void processIncomingByte (const byte inByte)
{
static char input_line [MAX_INPUT];
static unsigned int input_pos = 0;

switch (inByte)
{
case '\n':   // end of text
input_line [input_pos] = 0;  // terminating null byte

// terminator reached! process input_line here ...
process_data (input_line);

// reset buffer for next time
input_pos = 0;
break;

/* case '\r':   // discard carriage return
break;*/

default:
// keep adding if not full ... allow for terminating null byte
if (input_pos < (MAX_INPUT - 1))
input_line [input_pos++] = inByte;
break;
}
}


void led_on()
{
digitalWrite(LED, HIGH);
}

void led_off()
{
digitalWrite(LED, LOW);
}

void ledflash(int ftime)
{
led_on();
sdelay(ftime);
```

```
led_off();
}

// ### safe delay because the delay() function of arduino is blocking ###
void sdelay(int duration)
{
unsigned long start_time = millis();
while ((millis() - start_time) < duration) {}
}

void start_rec()
{
digitalWrite(STOP_PIN, LOW);
digitalWrite(REC_PIN, HIGH);
sdelay(450);
TCCR1A |= (1 << COM1A0);   // Toggle OC1A on Compare Match and override port 9
led_on();
}

void stop_rec()
{

//pulse(1000, 40);
TCCR1A &= ~(1 << COM1A0);  // Port 9 OFF*/
digitalWrite(STOP_PIN, HIGH);
digitalWrite(REC_PIN, LOW);
led_off();
}

// pulse on TEST pin
void pulse(int frequency, byte number)
{
uint32_t wait_delay = 1000000 / frequency / 2;
for (byte i = 0; i < number; i++) {
digitalWrite(TEST, HIGH);
delayMicroseconds(wait_delay);
digitalWrite(TEST, LOW);
delayMicroseconds(wait_delay);
}
}
```

Bibliography

[1] S. Bansilal, J. M. Castellano, V. Fuster, C. Nacional, D. I. Cardiovasculares, and C. Carlos, "Global burden of CVD : focus on secondary prevention of cardiovascular disease," vol. 201, no. 1, pp. 1–7, 2015.

[2] I. Van Dis, J. Buddeke, I. Vaartjes, F. Visseren, and M. L. Bots, "Hart- en vaatziekten in Nederland 2015, cijfers over heden, verleden en toekomst." Den Haag, 2015.

[3] O. H. Franco, A. J. Der Kinderen, C. De Laet, A. Peeters, and L. Bonneux, "Primary prevention of cardiovascular disease : Cost-effectiveness comparison," vol. 23, no. 1, pp. 71–79, 2007.

[4] H. J. Vos, B. M. van Dalen, I. Heinonen, J. G. Bosch, O. Sorop, D. J. Duncker, A. F. van der Steen, and N. de Jong, "Cardiac Shear Wave Velocity Detection in the Porcine Heart," *Ultrasound in Medicine & Biology*, pp. 1–12, 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0301562916304124

[5] N. de Jong et al, "EFFECTS proposal," pp. 1–16, 2017.

[6] L. Biel, O. Pettersson, L. Philipson, and P. Wide, "ECG Analysis : A New Approach in Human Identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808–812, 2001.

[7] F. Kusomoto, *ECG Interpretation.* Springer Science+Business Media, New York, 2009.

[8] S. M. Debbal and F. Bereksi-Reguig, "Time-frequency analysis of the first and the second heartbeat sounds," *Applied Mathematics and Computation*, vol. 184, pp. 1041–1052, 2007.

[9] P. Sudhakar Reddy, R. Salerni, and J. A. Shaver, "Normal and abnormal heart sounds in cardiac diagnosis: Part II. Diastolic sounds," *Current Problems in Cardiology*, vol. 10, no. 4, 1985.

[10] H. K. Walker, W. D. Hall, and H. J. W., *Clinical Methods: The History, Physical, and Laboratory Examinations 3rd edition*, 1990.

[11] W. Phanphaisarn, A. Roeksabutr, P. Wardkein, J. Koseeyaporn, and P. Yupapin, "Heart detection and diagnosis based on ECG and EPCG relationships," *Medical Devices: Evidence and Research*, vol. 4, no. 1, pp. 133–144, 2011. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-83755188590{&}partnerID=40{&}md5=9a5534b916e001757e722aab8b9ceea9

[12] J. D. Richard and C. Gables, "Frequency spectra of some normal heart sounds," *American Heart Journal*, vol. 53, no. 2, pp. 183–192, 1956.

[13] M. E. Stept, C. E. Heid, J. A. Shaver, D. F. Leon, and J. J. Leonard, "Effect of Altering P-R Interval on the Amplitude of the First Heart Sound in the Anesthetized Dog," *Circulation*, vol. 25, no. 3, pp. 255–263, 1969.

[14] P. J. Arnott, G. W. Pfeiffer, and M. E. Tavei, "Spectral analysis of heart sounds: relationships between some physical characteristics and frequency spectra of first and second heart sounds in normals and hypertensives," *Journal of Biomedical Engineering*, vol. 6, pp. 121–128, 1984.

[15] S. H. Kim, H. J. Lee, J. M. Huh, and B. C. Chang, "Spectral Analysis of Heart Valve Sound for Detection of Prosthetic Heart Valve Diseases," *Yonsei Medical Journal*, vol. 39, no. 4, pp. 302–308, 1998.

[16] A. Sakai, L. P. Feigen, and A. A. Luisada, "Frequency distribution of the heart sounds in normal man," *Cardiovascular Research*, vol. 5, pp. 358–363, 1971.

[17] A. P. Yoganathan, R. Gupta, F. E. Udwadia, J. W. Miller, W. H. Corcoran, J. L. Johnson, and R. J. Bing, "Use of the fast Fourier transform for frequency analysis of the first heart sound in normal man*," *Medical and biological engineering*, vol. 14, no. 1, pp. 69–73, 1976.

[18] M. Johnston, S. P. Collins, and A. B. Storrow, "The Third Heart Sound for Diagnosis of Acute Heart Failure," *Current Heart Failure Reports*, vol. 4, no. 3, pp. 164–169, 2007.

[19] "Mechanism of the third heart sound," *The Lancet*, p. 1137, 1986.

[20] Y. Ozawa, D. Smitch, and E. Craige, "Origin of the Third Heart Sound Studies in Human Subjects," *Circulation*, vol. 67, no. 2, pp. 399–404, 1983.

[21] L.-s. Zhong, X.-m. Guo, S.-z. Xiao, D. Wang, and W.-z. Wu, "The Third Heart Sound after Exercise in Athletes : An Exploratory Study," vol. 54, no. 4, pp. 219–224, 2011.

[22] D. H. Spodick and N. O. Fowler, "Fourth Sound Gallop or Split First Sound?" *The American Journal of Cardiology*, vol. 31, no. 4, pp. 530–531, 1973.

[23] A. Benchimol and K. B. Desser, "The fourth heart sound in patients without demonstrable heart disease," *American Heart Journal*, vol. 93, no. 3, pp. 298–301, 1977.

[24] J. A. Shaver, R. Salerni, and P. S. Reddy, "NORMAL SOUNDS AND ABNORMAL IN CARDIAC DIAGNOSIS PART I : SYSTOLIC SOUNDS," *Current Problems in Cardiology*, vol. 10, no. 3, pp. 1–68, 1985.

[25] Y. Sato, T. Kawasaki, S. Honda, K. Harimoto, S. Miki, T. Kamitani, H. Shiraishi, and S. Matoba, "Third and Fourth Heart Sounds and Myocardial Fibrosis in Hypertrophic Cardiomyopathy," *Circulation Journal*, vol. 82, no. 2, pp. 509–516, 2018.

[26] Y.-L. Tseng, "Detection of the third and fourth heart sounds using Hilbert-Huang transform," *BioMedical Engineering OnLine*, vol. 11, no. 8, pp. 1–13, 2012.

[27] R. Johnson and R. Holzer, "Evaluation of asymptomatic heart murmurs," *Current Paediatrics*, vol. 15, no. 7, pp. 532–538, 2005.

[28] E. Begic and Z. Begic, "Accidental Heart Murmurs," *Medical Archives*, vol. 71, no. 4, pp. 284–287, 2017.

[29] W. F. Weaver and C. H. M. Walker, "Innocent Cardiovascular Murmurs in the Adult," *Circulation*, vol. 29, no. 5, pp. 702–707, 1964.

[30] A. Pomerance, "Cardiac Pathology and Systolic Murmurs in the Elderly," *British Heart Journal*, vol. 30, no. 5, pp. 687–689, 1968.

[31] A. N. Pelech, "The physiology of cardiac auscultation," *Pediatric Clinics of North America*, vol. 51, no. 6, pp. 1515–1535, 2004.

[32] M. R. Rangarai and I. S. N. Murthy, "QUANTlTATlVE ANALYSIS OF THE PHONOCARDIOGRAM FOR DETECTION OF MURMURS," *Journal of Biomedical Engineering*, vol. 1, no. 4, pp. 247–252, 1979.

[33] A. R. Freeman and S. A. Levine, "Clinical significance of systolic murmurs: Study of 1000 consecutive "noncardiac" cases," *Annals Of Internal <edicine*, vol. 6, no. 11, pp. 1371–1385, 1933.

[34] S. A. Levine, "THE SYSTOLIC MURMUR ITS CLINICAL SIGNIFICANCE," *Journal of the American Medical Association*, vol. 101, no. 6, pp. 436–438, 1933.

[35] R. Keren, M. Tereschuk, and X. Luan, "Evaluation of a Novel Method for Grading Heart Murmur Intensity," *Archives of Pediatrics and Adolescent Medicine*, vol. 159, no. 4, pp. 329–334, 2005.

[36] Pelech and A. N., "THE CARDIAC MURMUR: When to Refer?" *Pediatric Clinics of North America*, vol. 45, no. 1, pp. 107–122, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003139550570585X?via{%}3Dihub

[37] S. S. Hussain, H. S. Hussain, T. T. Swee, C.-M. Ting, A. M. Noor, S. Pipatsart, J. Ali, and P. P. Yupapin, "Acoustic cardiac signals analysis : a Kalman filter – based approach," *International Journal of Nanomedicine*, vol. 7, pp. 2873–2881, 2012.

[38] "Thinklabs One User's Manual," 2017. [Online]. Available: http://docs.wixstatic.com/ugd/51509e{_}3ec7a53d90c54c9aab120c2fc93178cd.pdf

[39] Thinklabs, "Thinklabs One - Digital Stethoscope: The smallest, most powerful stethoscope in the world," 2017. [Online]. Available: https://store.thinklabs.com/products/thinklabs-one-digital-stethoscope

[40] 3M Health Care, "3M™ Littmann® Electronic Stethoscope Model 3200," 2017. [Online]. Available: https://www.littmann.com/3M/en{_}US/littmann-stethoscopes/products/{~}/3M-Littmann-Electronic-Stethoscope-Model-3200?N=5932256+8711017+3293188392{&}rt=rud

[41] Eko, "Core." [Online]. Available: https://ekodevices.com/core/

[42] 3M Health Care, "Hear it. With Confidence, 3M™ Littmann® Electronic Stethoscope Model 3100," 2017.

[43] ——, "Sound Strategy. The story behind the development of 3M™ Littmann® Electronic Stethoscopes," 2017.

[44] Allheart.com, "3200 Electronic Ambient Noise Reduction 27" Stethoscope," 2017. [Online]. Available: https://www.allheart.com/3m-littmann-3200-electronic-ambient-noise-reduction-27-inch-stethoscope/p/litt3200/

[45] 3M Health Care, "3M™ Littmann® StethAssist™ Heart and Lung Sound Visualization Software User Manual," 2017.

[46] Eko Devices, "Eko Electronic Stethoscope System," Tech. Rep., 2015.

[47] Eko, "Eko CORE Digital Stethoscope," 2017. [Online]. Available: https://ekodevices.com/shop/?bundle

[48] Allheart.com, "Eko Digital Stethoscope Attachment Only," 2017. [Online]. Available: https://www.allheart.com/eko-digital-stethoscope-attachment-only/p/ekcore/

[49] S. Chauhan, P. Wang, C. Sing Lim, and V. Anantharaman, "A computer-aided MFCC-based HMM system for automatic auscultation," *Computers in Biology and Medicine*, vol. 38, no. 2, pp. 221–233, 2008.

[50] M. S. Obaidat, "Phonocardiogram signal analysis: Techniques and performance comparison," *Journal of Medical Engineering and Technology*, vol. 17, no. 6, pp. 221–227, 1993.

[51] S. R. Messer, J. Agzarian, and D. Abbott, "Optimal wavelet denoising for phonocardiograms," *Microelectronics Journal*, vol. 32, no. 12, pp. 931–941, 2001.

[52] D. Gradolewski and G. Redlarski, "Wavelet-based denoising method for real phonocardiography signal recorded by mobile devices in noisy environment," *Computers in Biology and Medicine*, vol. 52, pp. 119–129, 2014.

[53] "iE33 Ultrasound System User Manual," 2012.

[54] Verasonics Inc., "Vantage External Synchronization Triggers," 2017.

[55] "SYSTEMS ENGINEERING HANDBOOK, A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES," 2011.

[56] Eko Devices, "Integrate with Eko," 2018. [Online]. Available: https://ekodevices.com/software-integrations/

[57] Olympus Imaging America Inc., "Linear PCM Recorder LS-14/LS-12," 2013.

[58] "Sound Blaster Omni Surround 5.1 Users's Guide," 2013. [Online]. Available: https://images-na.ssl-images-amazon.com/images/I/C1EI0KIWvFS.pdf

[59] "TOM-1545P-R Omni-Directional Microphone," Projects Unlimited, Tech. Rep., 2015.

[60] "TOM-1545P-R," Projects Unlimited, Tech. Rep., 2006.

[61] D. DeBusschere, "z.one Research Package User's Guide," 2016.

[62] "External Trigger Cable For Z.One Ultrasound Research Applications Quick Reference Guide," pp. 3–5, 2017.

[63] R. Mudhar, “Olympus LS10 and LS14 DIY wired remote control,” 2018. [Online]. Available: http://www.richardmudhar.com/blog/2016/10/olympus-ls10-diy-wired-remote/

[64] Olympus, “RS30W Remote Control,” 2018. [Online]. Available: https://www.olympus.co.uk/site/en/a/audio{_}accessories/accessories{_}audio{_}recording/adapters{_}cables{_}remote{_}controls{_}connectors{_}4/rs{_}30w/index.html