Development of Detachable Audio Accessory for Surveillance Cameras

Eric Hammar

Pontus Sjöstrand

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6201
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2023

Abstract

Surveillance cameras are used for video monitoring. To complement the video stream, different external accessories can be attached to a camera. The addition of external accessories brings certain limitations in terms of aesthetics and functionality. To overcome these limitations, an internal platform for connecting accessories would enable the creation of better and more future-proof products.

This Master's Thesis investigates the development of a detachable two-way audio accessory and an internal platform for a panoramic surveillance camera. A structured product development method was followed, which resulted in a functional prototype. The accessory consists of a speaker, an amplifier, and two microphones for audio output and input. The components are enclosed by 3D printed parts with a geometry adapted to the camera's internal platform. The mechanical fastening of the accessory is made in two steps utilizing magnets and screws, and the electrical connection between the accessory and the camera is made using pogo pins. The playback and recording are interacted with through a graphical user interface. Pros and cons of the resulting prototype are discussed, and possible future improvements are suggested.

Acknowledgements

We would like to express our gratitude to some people who have contributed greatly to our Master's Thesis:

Thanks to Linus Wannebro and Magnus Sjöberg, our industrial supervisors. Thank you for always being so encouraging, showing genuine interest and always believing in our abilities.

Thanks to our colleagues at MCP and to all the other talented people at the company who have shared their knowledge with us when we needed it the most.

Thanks to Björn Olofsson and Bo Bernhardsson, our academic supervisor and examiner at LTH, for supporting us. Your guidance has been truly valuable and has made us feel confident throughout the Master's Thesis.

Finally, thanks to *Maskinsektionen inom TLTH* for making our previous five years at LTH a blast and to our friends and families for supporting and believing in us during our time in Lund.

Contents

1

Introduction

1.1 Background

Surveillance cameras are today widely used to create a safe environment for humans. There are many different types of surveillance cameras for different purposes and environments. Some camera sensors are fixed in their position and zoom while others can be adjusted in different directions. Some cameras consist of a single camera sensor, while others are made up of several sensors. One type of multi-sensor camera is a panoramic camera. These consist of multiple video streams that together provide a 360-degree overview of an area when mounted on the ceiling of a room or suspended outdoors.

To complete the video stream, other functions than video can be built into the camera. For example, audio can be used in the video stream by adding microphones and speakers to the camera housing. IR sensors and LED lights are other examples of parts that can be added. The camera will then be delivered with these functions integrated permanently, and if the customer wants different functions, a new camera needs to be purchased. An alternative to this is to use external accessories that can be attached to the camera. However, these are often not designed specifically for a particular camera. This has its limitations, both aesthetically and functionally, since the accessories' design and connection interface depend on what type of accessory is going to be attached. A better solution, that today does not exist on the market, would be to design both modular and detachable accessories that can be connected to a specific camera internally with a generic connection interface. The customer will then be able to buy one camera with exchangeable functions through the modular accessories designed for that particular camera. A modular and detachable accessory with a generic connection interface will expand the possible use cases for the customer buying a surveillance camera with these accessories available. This will not only be user-friendly but also cost-efficient and create better and more future-proof products.

For panoramic cameras, different kinds of accessories could potentially be developed. Such accessories could be infrared lights (IR) to be able to use the camera sensors in darkness, a pan-tilt-zoom camera (PTZ camera) to get a high-resolution image of a specific area, or a strobe to light up dark environments. Another accessory that could be combined with a panoramic camera is an accessory for audio input and output. By using this two-way communication, the camera can detect sounds of interest and communicate with people close to the camera. The number of possible use cases increases.

There are multiple advantages to combining detachable audio accessories with surveillance cameras. The user of the camera gets more freedom because accessories can be chosen and swapped out based on the use case and purpose of the camera. Furthermore, more cameras that are sold to the customer can potentially use the same base before adding a specific accessory, which means more cameras can be mass-produced and in turn reduce costs.

1.2 Problem Formulation

A manufacturer of network cameras today manufactures a panoramic camera, from here on referred to as "the camera". The camera consists of four camera sensors positioned in a circle to create a 360-degree overview. In the center of the circle, there is room for other parts to be included in the camera. An illustration of the camera consisting of four camera sensors and a void is shown in Figure 1.1.

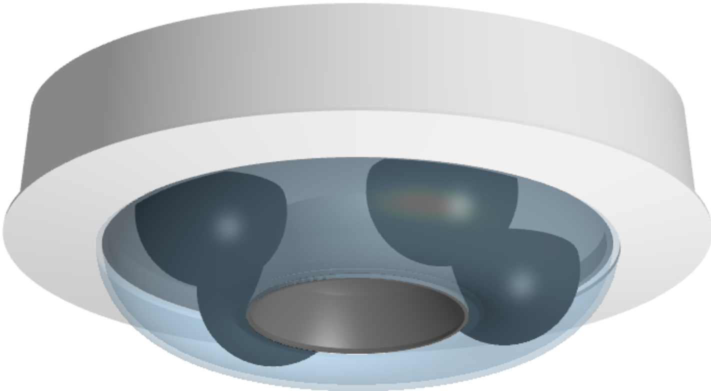

Figure 1.1 An illustration of the panoramic camera.

The goal of this Master's Thesis is to utilize the free space in the center of this panoramic camera and develop a detachable accessory for two-way audio communication. The accessory should:

- Be able to play sound, especially speech, to be heard clearly at a reasonable distance
- Detect sound and make a simple analysis of the audio input
- Process the audio input and output signals
- Connect electronically to the existing camera
- Fit geometrically into the void inside the circle with the camera sensors
- Be detachable

The development of the accessory includes choosing appropriate components, integrating them neatly into a platform, and connecting the components to the printed circuit board (PCB) of the existing camera.

The accessory is supposed to be used in public spaces, e.g., airports, train stations, and fuel stations where the camera would sit in the ceiling. If unusual behavior is detected via the video stream or if a sound of interest is detected, the speaker should be able to send a pre-recorded sound message to people on site.

1.3 Assumptions and Limitations

The goal of the thesis is not to develop a complete product ready for mass production, but rather a functional prototype as a proof of concept. A limitation set is that a PCB in the accessory does not necessarily have to fit internally in the prototype of the accessory since evaluation boards used during development can be hard to fit internally. Neither does the accessory have to use the current system-on-chip for computing when the accessory is connected to the camera. To tailor an own PCB or use the company's available chips in the prototype is perceived as unnecessarily complex work that would take time from other important parts of the project. Instead, the goal is to use existing, easy-to-use platforms as replacements for both the camera's chip and the PCB of the accessory.

A Raspberry Pi [*What is a Raspberry Pi?* 2023] will be used instead of the camera's system-on-chip. Necessary electronic components will in this Master's Thesis be used instead of the accessory's PCB. The idea is that the Raspberry Pi and the electronic components should be connected to the accessory's internal components to conceptually show the function of the prototype. A Raspberry Pi is suitable to use as a microprocessor for this project because it is easy to use and has many input and output pins, including pins for digital sound. See Section 2.3 for details about the Raspberry Pi.

In a real-world product, the camera and the accessory should act as one unit once the accessory is installed, and processing of audio input and output should take place on the camera's system-on chip. The system-on chip should be connected to an Ethernet cable which would make it make it possible for an operator to communicate with the accessory remotely. In this thesis, however, a remote connection is not looked into. Sending and receiving audio signals will in this thesis instead be performed on the Raspberry Pi.

A detailed cost estimation will be outside the scope of this thesis. However, the parts will be selected with their price in mind but it will not be a deciding factor when choosing between two different parts.

1.4 Disposition of the Report

Important concepts and vocabulary are described in a theory part in Chapter 2. The product development method used is described thoroughly and audio and electronic concepts are introduced. These are relevant in order to understand the remaining part of the report. The theory chapter is followed by Chapter 3 where it is described how the product development process will be applied to the development of the specific audio accessory.

In Chapter 4, details of the development process are presented and it is described which concepts that have been chosen and why. The final prototype is presented at the end of Chapter 4. The results are then discussed in Chapter 5. Finally, suggestions for future work are presented and a conclusion is drawn in Chapter 6.

2

Theory

2.1 Product Development Process

Developing a product can often be a complex process. Therefore, it is good to follow a method where the process is divided into a few steps to simplify and clarify the work process. The method chosen in this project is the one described by Karl T. Ulrich and Steven D. Eppinger in the book "Product Design and Development" [Ulrich and Eppinger, 2012]. There are several benefits to following this model, including that the quality of the resulting product is guaranteed, a well-planned work with milestones at each phase, as well as great opportunities for evaluation and problem-solving during the ongoing process.

Planning

The first phase in the product development model presented by [Ulrich and Eppinger, 2012] is the planning phase. This phase presents which goals that exists with the development and the resources to achieve these goals. It is important that this phase is done carefully because the product being developed must be in line with the company's business strategy and existing product portfolio. It is also in this phase that ideas are developed, and a timetable is created. The result of this phase is a "Mission Statement" where the purpose of the project is described and identifies relevant stakeholders, customers and business goals.

Concept Development

In the next phase of the process according to [Ulrich and Eppinger, 2012], the goal is to generate several concepts, test these and choose one that will be developed in subsequent phases. The basis of this step is to identify the customer needs that are concretized into product specifications. Once it has been clarified what characteristics the product should have, concepts can be generated. This is done by searching for information internally and externally and generating ideas and solutions together in teams. The different concepts are then analyzed and compared against each other in order to systematically select one or more that go on to testing. In the final step, the concept or concepts are then tested to ensure that customer needs are met and to identify any flaws that may prove costly later in the process.

System-Level Design

The purpose of this phase in the approach by [Ulrich and Eppinger, 2012] is to develop the architecture of the product. This means that the product's physical subsystems and elements are clarified based on which functions these should have and how these are connected to each other. This step is based on the product's functions together with the individual parts and components. This is usually illustrated schematically and is then called the product's architecture.

Detail Design

Detail Design in the model presented by [Ulrich and Eppinger, 2012] is the phase when the individual components and parts are completed. This includes part geometry, material selection and tolerances. It is common to use the "Design for X" method, where X can stand for different criteria. It is common to use "Design for manufacturing", where the parts are designed and constructed in order to be manufactured as efficiently as possible and still maintaining the desired level of quality and safety.

Testing and Refinement

In the penultimate phase, according to [Ulrich and Eppinger, 2012], the product is tested, both as a whole and at the component level. During the testing, it may occur that errors are found or that something needs to be developed further. Therefore, this becomes an iterative process.

Production Ramp-Up

In the final stage of the product development process in the approach by [Ulrich and Eppinger, 2012], the product is made using the intended production system. During the ramp-up, the workforce is trained, and the remaining errors in the production are identified.

2.2 Audio Concepts

Sound Pressure Level

The sound volume from a speaker can be experienced differently depending on the person listening and the environment where the speaker is operating. In order to more easily compare the capacity of speakers, it is therefore important to have an objective unit of measurement. An example of this is "sound pressure level" (SPL). Sound pressure is the average variation in atmospheric pressure caused by the sound. The sound pressure level is what level this pressure constitutes and is measured in decibels (dB) [*What is Sound Pressure Level (SPL) and how is it measured?* 2023].

Described by [Long, 2014], "sound pressure level" (L_p) is defined as follows:

$$L_p = 10 \log \left(\frac{p^2}{p_{\text{ref}}^2} \right)$$

where

$$p = \text{rms sound pressure (rms)}$$
$$p_{\text{ref}} = 2 \cdot 10^{-5} \text{ Pa (reference pressure)}$$

Speaker Sensitivity

In order to be able to compare speakers with each other in a fair way, it is important that they are compared under the same conditions. Speaker sensitivity is a measurement based on sound pressure level but with a specific given power and at a specific distance, explained in [*Speaker Sensitivity* 2023]. When the value of this measurement is reported in the speaker's specifications, a power and distance are often stated. At the given power level and at the given distance, the sound level is then equal to SPL. The measure can thus be interpreted as the speaker's efficiency.

Loudspeaker Enclosure

A loudspeaker is primarily based on one simple concept, to move air to get sound. This is done by a diaphragm or cone that is put in motion by the mechanical movement created by an electromagnetic field, according to [Dickason, 2006]. The electromagnetic field is generated by applying a current to a coil.

When the air in front of the loudspeaker is pushed forward by the diaphragm or cone, the air on the other side of the speaker is diluted or rarefied, described by [Fantel, 1986]. This means that the air is behaving in an opposite manner on one side of the speaker compared to the other. The rarefied air on the back will then nullify the compressed air in the front resulting in a cancellation of sound. To avoid this, a loudspeaker enclosure is used, often consisting of a sealed box trapping the rear of the speaker and delimiting the air behind the speaker from reaching the air in front of it. Ideally, this enclosure box is made airtight and the inside walls are lined with absorbing material.

Speaker Frequency Response

To obtain a high-quality sound played by the speaker, it is important that the speaker can generate sound in a wide range of frequencies. The range of frequencies that a speaker can reproduce is measured by the frequency response [*Understanding Speaker Frequency Response* 2023]. The human ear can perceive and hear sounds between 20 and 20 000 Hz. In order to use the speaker's frequency response as a parameter when speakers are compared to each other, it is not only interesting to look at which frequencies the speaker can play but also any deviations across the range at certain frequencies. This is often illustrated in a graph in the datasheet for the speaker in question. In such a graph, the Sound Pressure Level that the speaker is able to reproduce at specific frequencies is illustrated. An example is shown in Figure 2.1. Different frequency responses can be preferable in different cases and this depends on the case where the speaker is going to be used.

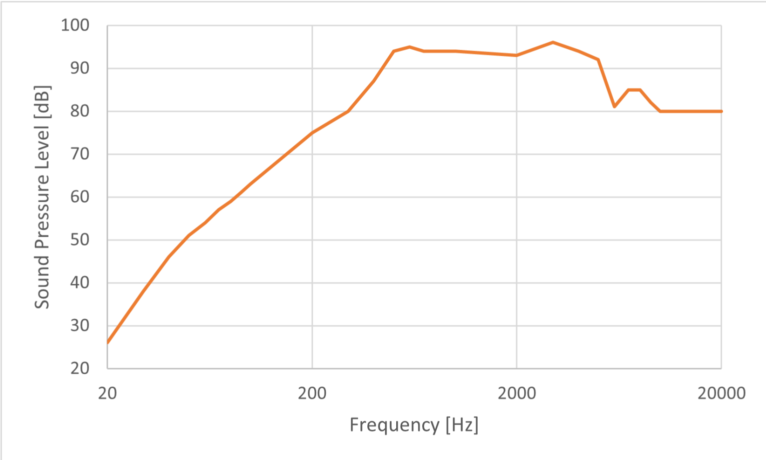

Figure 2.1 Example of how a speaker frequency response can look like.

Microphone Sensitivity

The sensitivity of a microphone is the ratio between the electrical response and a given acoustic input signal [*Microphone Specifications Explained* 2023]. The acoustic input signal is often standardized and represented by a sine wave with a frequency of 1 kHz at 94 dB SPL. A higher sensitivity of a microphone gives a higher output for the electrical response for the given acoustic input. The value is often specified as negative which means that a higher sensitivity is represented with a smaller absolute value.

The sound quality of a microphone cannot only be testified by the sensitivity itself, but also depends on the context in which the microphone is to be used [*Microphone Specifications Explained* 2023]. If the microphone is to be used in situations where the sound source is close, a microphone with higher sensitivity will cause more distortion and poorer sound quality. A microphone with higher sensitivity should therefore be used in situations where the sounds are not particularly isolated and directed. For such a situation, a microphone with lower sensitivity is better.

Microphone Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) describes the ratio between a reference signal and noise that degrades the signal [*Microphone Specifications Explained* 2023]. When this value is specified in specifications, the reference signal is usually standardized (1 kHz, 94 dB SPL). Since the reference signal can be seen as the signal and the sound you want to hear, SNR can be seen as a measure of how much noise interferes with the recording of a specific microphone. A higher value of SNR thus gives less interference from background noise.

Microphone Frequency Response

The frequency response of a microphone indicates the level of the output signal at a certain frequency and which frequencies the microphone can reproduce [*Mic Basics: What is Frequency Response?* 2023]. Like the frequency response of speakers, the frequency response of microphones is also illustrated with a graph. A frequency response of a microphone can be more or less even for a certain interval of frequencies. An even response in a range of frequencies means that the microphone is equally sensitive to those frequencies, which will mean that the sound that the microphone reproduces does not differ that much from the original sound. This is beneficial when recording musical instruments or sound effects, but works worse when recording voices.

A more uneven frequency response makes the microphone more sensitive to some frequencies than others [*Mic Basics: What is Frequency Response?* 2023]. The response then has peaks in gain at certain frequencies and troughs at others. Microphones with an uneven frequency response are usually less sensitive to sounds with low frequencies, which makes them less sensitive to noise and other sounds in the environment. Instead, it is common for frequencies in the range of speech and instruments to be amplified and these sounds therefore become clearer and can be reproduced with a higher quality. As a rule, the frequency response should be relatively smooth and not have overly large peaks and valleys. This can cause the sound to be perceived as unnatural.

I^{2}S Sound Protocol

I^{2}S (Inter-IC sound) is a digital sound protocol developed by Philips Semiconductors (now NXP Semiconductors), described in [*I^{2}S bus specification* 2022]. Although the technology was developed already in the 1980s, it is still widely used for sending digital audio between chips. The I^{2}S standard uses only three lines to keep wiring simple: Serial data (SD), Word select or Left-right clock (LRCLK) and Continuous serial clock or Bit clock (BCLK). An illustration of the three lines can be seen in Figure 2.2.

As described in [*I^{2}S bus specification* 2022], the Serial data line is the line used to represent an actual sound tone. For this line, a word length is chosen to represent an audio frequency. A longer word length (more bits) allows for higher precision but requires a higher clock frequency or lower sampling rate. A bit can be represented with either a 0 or a 1, and the number of different audio frequencies that can be represented is therefore two to the power of the number of bits. The left-right clock line simply determines whether the current information belongs to the right or left channel. This line can be represented with a 0 or a 1.

Also explained in [*I^{2}S bus specification* 2022], the bit clock line counts the number of bits that passes and the number of bits per second depends on the bit depth of the application and the sampling rate. Every second, the number of counts is the sampling rate multiplied by the bit depth. Additionally, since there are two channels (left and right), the number of bits needs to be doubled. For instance, if an application would use a bit depth of 16 and a sample rate of 48 kHz, the clock frequency would be $16 \cdot 48000 \cdot 2 = 1.536$ MHz.

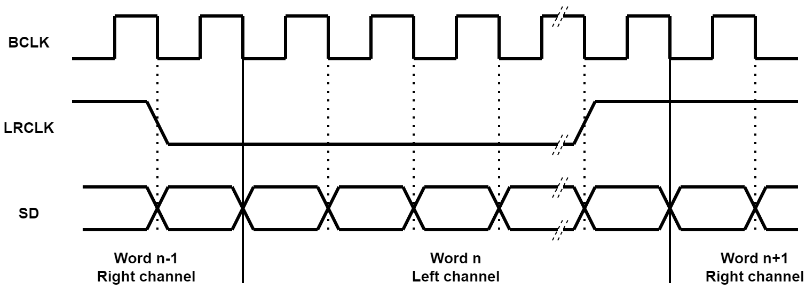

Figure 2.2 Illustration of the three lines of the I^2S sound protocol. Reproduced from [*I^2S bus specification* 2022].

PDM Sound

Another common digital sound protocol used is PDM (Pulse density modulation), which is another digital representation of analog sound [*PDM vs. I^2S: Comparing Digital Interfaces in MEMS Microphones* 2023]. PDM only consists of one line and is made up of a series of continuous bits which can have a value of either 0 or 1. The density of the high and low bits represents an audio frequency. The higher the amplitude of the audio signal, the higher the density of the high bits. An illustration of how the PDM sound protocol works is shown in Figure 2.3. Since many bits are required to describe only a single oscillation of the analog sound the bits represent, a very high sampling rate is required to accurately represent the sound wave.

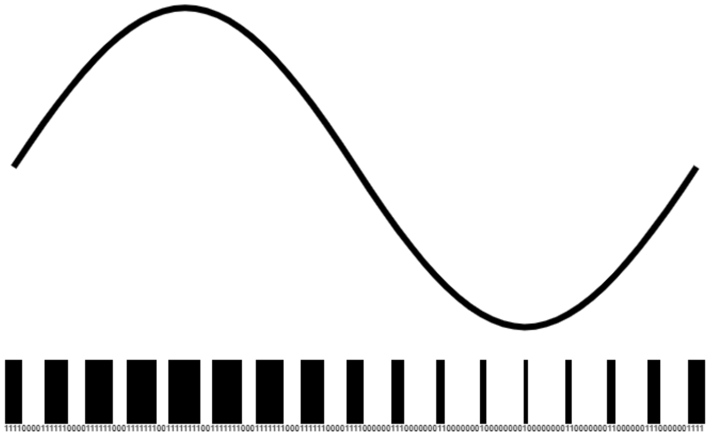

Figure 2.3 Illustration of how an analog tone is represented in binary format using the PDM sound protocol. Reproduced from [*PDM vs. I^2S: Comparing Digital Interfaces in MEMS Microphones* 2023].

2.3 Electronics

Amplifiers

The purpose of using amplifiers in audio circuits is to reproduce and enhance input signals, delivering them as audio output signals at a desired volume and power level [*Analog Dialogue* 2023]. Amplifiers can be implemented differently by using transistors in various ways. The transistors operate in linear mode to scale the input voltage to a given output voltage.

The transistors in an amplifier can also operate in a switching state where they switch from being on to off. These are the basics for so-called Class D amplifiers [*Analog Devices* 2023]. The switching happens at a frequency that is higher than the highest audio signal that needs to be reproduced. Since the transistors are either on or off, the efficiency of these amplifiers is high.

Raspberry Pi

A Raspberry Pi is a small computer that comes at a low cost and can be used to realize projects where programming and computation are necessary [*Raspberry Pi 4 Product Brief* 2021]. A common language to use when programming with a Raspberry Pi is Python [*What is a Raspberry Pi?* 2023].

A Raspberry Pi has so-called GPIO pins which stand for General-Purpose Input/Output pins [*Raspberry Pi GPIO pins* 2023]. These are used to connect other devices to make it possible for communication between these and the Raspberry Pi. The latest models have 40 GPIO pins providing different functions. One of these functions is communication via the I^2S sound protocol [*Raspberry Pi Audio* 2023]. These pins make it possible to send and receive digital sound data to and from other devices, for example, microphones and amplifiers.

Digital Signal Processor

A digital signal processor (DSP) is one specific type of microprocessor [W. Smith, 1997]. It is designed to process digital signals in real-time and can be used in many contexts, for instance when processing audio or images. Any microprocessor can generally function as a digital signal processor but will not perform as well as a designated DSP, and the energy efficiency will be worse and the cost will be higher.

In terms of audio signal processing, which in this Master's Thesis is the relevant area, a DSP offers multiple features according to [W. Smith, 1997]. Its use cases include filtering of audio input and output, volume control, limiting, mixing, and speech recognition. Some audio DSPs also have built-in analog-to-digital converters and digital-to-analog converters, also known as codecs. This eliminates the need for external codecs.

MEMS Microphone

MEMS is short for microelectromechanical systems, explained by [Lindroos, 2020]. In MEMS technology, mechanical and electrical parts are combined creating structures on the micrometer scale, hence the name microelectromechanical. In recent times, the usage of MEMS microphones has increased drastically and this is mainly due to their high stability against environmental conditions such as temperature humidity and vibrations. The MEMS microphones can also deliver better electro-acoustical performance at a smaller size compared to the available alternatives.

As there are different types of microphones, there are also different types of MEMS microphones that use different technologies. The most common technique is based on a capacitive sensing principle, according to [Lindroos, 2020]. The basics of capacitive sensing are that an electrical field is changing when something is approaching or touching the sensor. A rigid backplate and a movable membrane are formed by standard silicon material and MEMS manufacturing processes. The capacitor is charged by a so-called ASIC, which stands for application-specific integrated circuit [*Capacitance Sensors for Human Interfaces to Electronic Equipment* 2023]. In this case, the sound port is located on the same side as the solder pads. This is the most common way of design and gives high acoustical performance at the same time as the design around the microphone does not get very complex compared to other designs. The alternative is to have the sound port in the metal lid. This will be favorable in some applications, for example when the mechanical integration of the microphone is of high importance.

MEMS microphones can be either analog or digital [*Analog and Digital MEMS Microphone Design Considerations* 2023]. In the latter case, the microphone has a built-in analog-to-digital conversion function. Analog microphones are generally more prone to electromagnetic interference that can cause distortion in audio signals.

I^2C Communication

The Inter-Integrated Circuit (I^2C) serial bus (not to be confused with I^2S) is a simple bidirectional 2-wire bus for communication between devices [*I2C-bus specification and user manual* 2021]. A big benefit of this protocol is that only two pins are required for two-way communication: a serial data line (SDA) and a serial clock line (SCL). I^2C is the world standard and widely used in industry.

3

Method

The problem in this thesis is approached by following the steps of a concept development process, to a reasonable extent. The Ulrich and Eppinger product development process [Ulrich and Eppinger, 2012] is a great support in the development of the accessory, but should not be a limitation when the method is not perfectly applicable to the specific product. In this project, the product development process by Ulrich and Eppinger has been adapted in a way to suit the specific development in the most efficient way. How this has been done, will be explained in this chapter.

3.1 Concept Development

The accessory that has been developed during this project has many different potential areas of usage and therefore it has been critical to identify key customer needs before starting to develop concepts. The key customer needs have then developed into product specifications, which in turn have been the foundation for how the concepts have been generated and analyzed. Since this product includes functions covering different functional areas, many sub-concepts have been generated, one that will solve each sub-problem. When these sub-concepts have been analyzed, they have not only been analyzed in isolation but also in a more holistic perspective taking into account how they affect the whole product structure.

Customer Needs and Metrics

By interviewing stakeholders with interest in the developed accessory, customer needs have been identified. The stakeholders have been product owners, managers, and supervisors. Although these people are not the end customers of a real-world product, they can in this case be seen as the customers since the purpose has been to develop an accessory that satisfies their needs and wishes. After the interviews, statements were interpreted into needs and then ranked by importance resulting in a group of key customer needs. The importance was given through intuition after interpreting what features the stakeholders thought were of importance. The needs are shown in Table 3.1.

Table 3.1 List of customer needs with corresponding priority.

No.	Need	Comment	Priority
1	The communication between camera and accessory works neatly	When the accessory is connected, the camera and this should work as one unit	***
2	The accessory fits inside the existing camera		***
3	The accessory is firmly attached to the camera	The accessory should not fall off	***
4	The accessory starts to record audio when a loud sound occurs close to the camera		**
5	The contact between the accessory and the camera is universal and suits a wide range of accessories		**
6	It is easy to install and uninstall the accessory		**
7	The microphone can detect sound in a wide range of volumes	Loud sounds, e.g., explosions, as well as quieter sounds, conversations	**
8	The speaker can play sound loudly		**
9	The speaker can play sound in a wide range of frequencies	Conversations: approx. 100-8000 Hz	**
10	The microphone can detect sound in a wide range of frequencies	Conversations: approx. 100-8000 Hz	**
11	The camera is weatherproof when the accessory is installed	Ideally, the camera should sustain moisture, wind and dust	*
12	The accessory is cheap to produce		*
13	The accessory has a aesthetically pleasing design		*

These needs can then be matched against corresponding metrics. These metrics are introduced in Table 3.2.

Table 3.2 Needs-Metrics Matrix.

		1	2	3	4	5	6	7	8	9	10	11
	Metric / **Need**	Compatibility between camera and accessory	Size and design of accessory	Impulse needed to cause the accessory to fall off the camera	Quality of software	Microphone sensitivity	Speaker sensitivity	Speaker frequency range	Microphone frequency range	Resistance against weather	Cost of materials and manufacturing	Aesthetics
1	The communication between camera and accessory works neatly	•										
2	The accessory fits inside the existing camera		•									
3	The accessory is firmly attached to the camera			•								
4	The accessory starts to record audio when sound is detected				•	•						
5	The contact between accessory and camera is universal and suits a wide range of accessories	•										
6	It is easy to install and uninstall the accessory	•										
7	The microphone can detect sound in a wide range of volumes					•						
8	The speaker can play sound loudly						•					
9	The speaker can play sound in a wide range of frequencies							•				
10	The microphone can detect sound in a wide range of frequencies								•			
11	The camera is weatherproof when the accessory is installed									•		
12	The accessory is cheap to produce										•	
13	The accessory has an aesthetically pleasing design											•

Target Specifications

When the customer needs had been identified, the next step was to develop these into product specifications and in more detail describe how well the product should perform. The performance is measured by establishing metrics out of the needs. Due to the big scope of this project, specific values and ranges for the different metrics have not been set. The metrics will instead be helpful during the concept development phase when comparing different concepts.

Concept Generation

Since the thesis involved multiple problems in terms of mechanics, electronics and software, concepts for different sub-problems were generated. The concepts were generated by brainstorming ideas, searching for solutions externally, and also looking internally at similar problems. All the ideas from the different sources were then analyzed and customized to fit into this project and concretized by sketches, schematics and CAD models.

Concept Selection

After the concepts for the sub-problems had been generated, the next step was to select which concept to develop further. The selection process included a comparison of the concepts through scoring and testing. Selecting a winner through both scoring and testing was valuable since the scoring gave a more isolated view of each concept whereas the testing gave a good view of how the concept function with other parts of the product and also how realistic it is.

Concept Scoring. The concept scoring was made by scoring the concepts on each sub-problem based on different criteria. The criteria were ranked with the most important given the largest weight points. A score was given to each concept between 1-5 where 5 were seen as the best. The final ranking then gave a winner based on a weighted score. When evaluating the different concepts, not all have been evaluated through concept scoring. This method has been used where it has been relatively easy to get an intuitive feeling of how the concept will work. Since the scoring has been done before any testing, the concepts that have been harder to get an intuitive feeling of their function, have been left out from the scoring and instead evaluated through careful testing.

Concept Testing. The testing phase was carried out by realizing the concepts further. As mentioned before, this gave a more realistic view of how well the concept will function and if it is feasible with the rest of the parts. Some concepts may have scored high but were harder to implement and vice versa which made this way of working valuable. Depending on which concept was going to be tested, different methods on how to realize them were used. For example, 3D printing has been used for mechanical concepts, and for software concepts, the writing and debugging of code have been a big part of the testing process.

3.2 System Level Design

When all concepts had been generated and selected, the next step was to combine these with each other to obtain the architecture of the product. The concepts that were chosen as winners were tested with each other to secure that it was possible for them to work together.

3.3 Detail Level Design

During Detail Level Design, the design of each individual part was finalized. After establishing the architecture of the product and that the concepts generated were feasible it was easier to develop them further and to customize them to fit into this product.

3.4 Testing and Refinement

In this project, Testing and Refinement was the last step that was used in the product development process by Ulrich and Eppinger. This was the last step because the thesis goal was to develop only one prototype and therefore production ramp-up was not done. The testing in this step was done for the product as a whole and not for the individual parts by themselves. However, the methods used were similar to the ones used for the more individual testing.

A lot of refinement has been made through changes and updates in the code used and also smaller design updates for the hardware.

4

Results

4.1 Mechanical Outline of the Accessory

The starting point for the concept generation process in terms of the mechanical design of the accessory will be to make the design fit the existing camera. The goal is to avoid changing any other parts of the camera and for the accessory to be adapted accordingly. With this in mind, the geometry of the inside of the camera ring will be used as inspiration and it will be assumed that the measurement of this part will be the form factor.

Top and Bottom Parts

This first concept consisting of one top and one bottom part is based on having a chassis that maximizes the available space in the middle of the camera and splitting the chassis into two parts. The accessory will then consist of a larger top part that must be attached to a bottom part that is attached permanently to the camera. If no accessory is mounted in the camera, there will be a relatively small part that remains in the camera if no extra placeholder is added. This concept requires a stable attachment with the bottom part attached to the camera. As the attachment between the bottom and top part will be relatively hard to reach, the rest of the accessory has to be designed to allow access to tools during assembly.

The ambition with the accessory is to be able to mount it without having to remove any other part of the camera. It is therefore important to make sure that no part of the upper part of the accessory is wider than the hole it has to pass through when it is attached to the camera. An illustration of the concept described in this section can be seen in Figure 4.1.

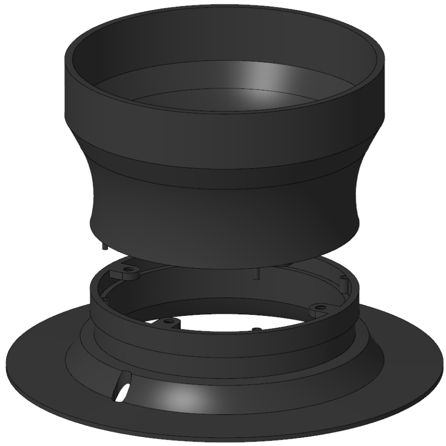

Figure 4.1 Illustration of the concept consisting of one top and one bottom part.

Inside and Outside Parts

The second concept for the design of the accessory is to create something that can be likened to an inner part and an outer part where the inner part is the accessory that is placed in a shell attached to the camera. The shell, or the outer part, is designed to fit in the camera with other parts, and the inner part is designed accordingly to be able to fit effectively in it. This concept means that a larger part remains in the camera when the accessory is not mounted. In this way, what acts as a placeholder does not need to be as large as the other concept described previously. Since this concept is more about filling a space than adding a new part, the camera can have a more uniform look, regardless of whether the accessory is installed or not. An illustration of the concept described in this section can be seen in Figure 4.2.

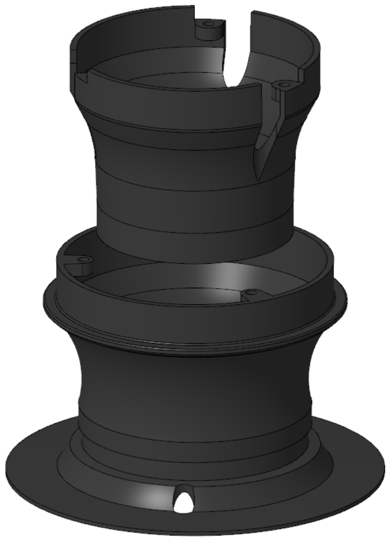

Figure 4.2 Illustration of the concept consisting of one inside and one outside part.

Scoring of Mechanical Outline

In Table 4.1 the two concepts for the mechanical outline of the accessory have been ranked out of a score based on four selection criteria. These selection criteria were chosen as the most important characteristics of the mechanical outline and combined they gave a broad picture of how well the accessory would function depending on which concept was chosen. As can be seen in Table 4.1, the most important criterion was that the other parts of the camera housing would not be exposed. However, the weights were spread relatively even.

Table 4.1 Concept scoring for the mechanical outline of the accessory.

Selection Criteria	**Weight**	**Top/Bottom**	**Inside/Outside**
Space available	0.25	5	3
Ease of installation	0.2	2	4
Stability	0.25	2	4
Other parts not being exposed	0.3	2	4
Weighted score		**2.75**	**3**
Rank		**2**	**1**

The scoring gave that the inside/outside concept was the winner based on these criteria. This concept scored higher in all of the selection criteria except "Space available" where the top/bottom concept was seen as the best alternative.

Testing of Mechanical Outline

To test the two concepts for the mechanical outline, the CAD models shown in Figure 4.1 and 4.2 were 3D-printed. By doing this, the stability and ease of installation were evaluated, which would not be possible if just having the CAD model. For these concepts, it was also necessary to test them together with the camera housing. This gave a good view of how the two concepts work with the other parts of the camera housing.

Top and Bottom. The concept consisting of one top and one bottom part made the other parts in the camera housing exposed during installation. When the accessory is not installed, a placeholder for the accessory is therefore needed to prevent sensitive parts from being exposed.

Since the fastening of the top and bottom parts happens at a relatively low point, there will be space needed to reach with the tool required for fastening. This will put requirements on the design of the inside of the accessory, for example where to put the speaker and the microphones.

Inside and Outside. Since this concept consists of one extra layer of material, the available volume in the accessory will be smaller than the one for the top and bottom concepts. However, the testing showed that the extra layer of material increases the ease of installation. It functioned as a guiding wall which made it relatively easy to get the accessory in the correct place. The extra layer will also act as a physical boundary between the accessory and the other parts of the camera housing.

Choice of Mechanical Outline

The concept that was chosen for the mechanical outline of the accessory was the inside and outside concept. During the testing, it was made clear that it is valuable to be able to mount the accessory easily and have it mounted stable. As the scoring showed, the only downside with the inside/outside concept compared to the top/bottom was the reduced available design space because of the extra layer of material. However, the testing showed that this was not as big an issue as the scoring suggested which made the advantages for the inside/outside concept outweigh the advantages for the top/bottom concept. Therefore, the inside/outside concept was a clear winner.

The scoring resulted in the inside/outside concept as a winner and the testing confirmed this ranking rather than speaking against it. An example of that can be seen by looking at the selection criteria with the largest weight, that other parts were not going to be exposed. The testing done after scoring made it clear that a lot of parts in the camera housing were exposed when using the top/bottom concept. Since this accessory is going to be detachable, it is not preferable to have this exposure to sensitive parts such as electrical components and camera modules.

4.2 Mechanical Fastening During Installation

To make the installation process easier, the ambition is to have one fastening before the accessory is secured with screws using a special tool. Since the accessory is going to be placed in a camera mounted on the ceiling, it is beneficial to fasten it in some way to make it easier to fasten the screws.

Snap-Fits

To fasten the accessory in the camera by using snap fits, the material that is being used must be flexible. In this concept, the snap fits used requires some elastic deforming. It is important that it is possible to detach the snap-fit so it is not permanent. The snap fits in this concept are therefore created in a way so the user can press these together when loosening the snaps and detaching the accessory. This concept of fastening mechanism does not require any other type of material added and the production will therefore be easier and cheaper. The CAD model is shown in Figure 4.3.

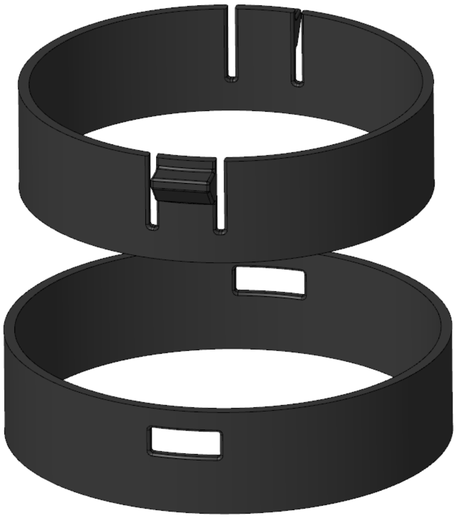

Figure 4.3 CAD-model showing the fastening concept using snap fits.

Magnets

The second concept for fastening the accessory is using magnets. This is done by creating pockets in both the accessory and the holder sitting in the camera. The magnets can then be placed in the pockets by using glue or press fitting. With the coming development of the electronic connection in mind, it is preferable to only have one possible way for the magnets to attract each other. In this concept, this is done by alternating polarity directions for the different magnets. An illustration is shown in Figure 4.4.

Figure 4.4 CAD-model showing the fastening concept using magnets.

Bayonet Mount

The bayonet mount concept is inspired by for example the mounting of a smoke detector in the ceiling. The bayonet mount requires not as much twisting as a normal thread but provides an easy and secure fastening. In this concept, the slot narrows down gradually after the bigger hole, which causes a grip fit that prevents the accessory from sitting loose when attached. An illustration is shown in Figure 4.5.

Figure 4.5 CAD-model showing the fastening concept using bayonet mount.

Scoring of Fastening During Installation

The scoring of the fastening is presented in Table 4.2.

Table 4.2 Concept scoring for the mechanical fastening during installation.

Selection Criteria	Weight	Snap fits	Bayonet	Magnets
Stability	0.25	2	4	2
Ease of installation	0.3	4	3	5
Durability	0.25	2	4	3
Movement simplicity	0.2	1	2	1
Weighted score		**2.4**	**3.3**	**2.95**
Rank		**3**	**1**	**2**

Testing of Fastening During Installation

The testing of the developed concepts has primarily been made by 3D printing the models and evaluating after performance and feel. 3D printing has been a necessary tool in the testing process since the printed models give a more realistic experience of how the concepts work compared to only evaluating through CAD models. The tests during the testing process gave rise to insights, which resulted in the different concepts being updated and improved with small changes between versions.

Snap fits. For the concept of using snap fits, 3D printing was key in evaluating how well-functioning this fastening mechanism was. The two parts attached together can be seen in Figure 4.6. The performance is very much dependent on the material used, and this design was customized for the plastic material used when 3D printing. One of the advantages of this concept is that it does not have any twisting motion which, will probably be beneficial when selecting which electronic interface to use. In other words, a twisting motion will probably limit the possible choices of the electronic interface. However, a disadvantage when detaching the accessory with snap fits is that this can be difficult if the applied force is not large enough. This will cause a trade-off in the design where weaker and softer snaps make it easier to detach but at the same time makes the accessory more fragile. The experience and value of this concept will therefore vary from person to person who is detaching and installing the accessory.

Figure 4.6 3D-printed model for testing snap fits.

Magnets. To test the concept that includes magnets, a model was 3D-printed with pockets where the magnets could fit. The two parts attached together can be seen in Figure 4.7. The pockets had so-called crush ribs which created a press fit between the magnet and the pocket. In the tests, neodymium magnets were used but these could be replaced with different strengths and sizes if this concept is chosen.

One of the advantages of this concept is the ease of installation. Another advantage is the feel it gives the user when the accessory is placed correctly. It is very clear when the accessory is placed in the right way and this is favorable. A possible disadvantage of this concept is possible instability because of vibrations. An important addition to this disadvantage is, however, that this fastening is only supposed to ease the installation and should not work as the only fastening mechanism for the accessory. Another disadvantage could be a possible increase in production costs for the final product since parts made of different materials are assembled together.

Figure 4.7 3D-printed model for testing magnets.

Bayonet Mount. The testing for the bayonet mount was done by 3D printing two parts that should fit together, one with slots and one with plastic pins that should fit into the slots. The two parts attached together can be seen in Figure 4.8. An attribute that is important when choosing a concept is that the user is made aware when the accessory is mounted correctly. In this design, this is done by having a grip fit between the two parts by making the slot gradually narrower. The bayonet mount does not require as large of a twisting motion as a normal thread would. This will not limit the choices of electrical interfaces as much as if using normal threads. A design and choice of the electrical interface that takes this twisting motion into account are, however, necessary in this case in contrast to the other concepts described previously. The bayonet mount is nevertheless a reliable and durable mechanism to use when fastening the accessory in the camera.

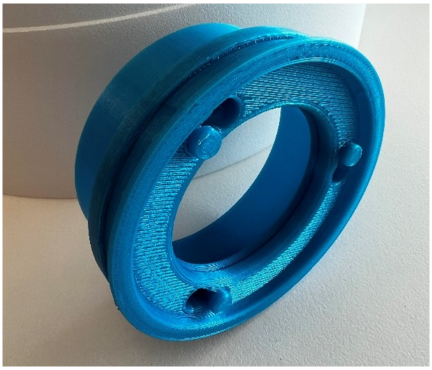

Figure 4.8 3D-printed model for testing bayonet mount.

Choice of Fastening During Installation

As the way of fastening the accessory during installation, the choice was magnets. This may not be obvious when looking at the scoring for these concepts since the bayonet mount was the one that had the highest score. But the choice has been made with both the scoring and testing taken into account and the magnet concept performed very well during the testing phase which made it the best choice. The ease of installation using the magnets solution was something that was better than expected when testing the concepts. During the testing, it got also clear that the ease of installation perhaps was underrated in the scoring phase in a way where it should have been given a larger weight.

Stability and durability were two selection criteria that other concepts scored higher than magnets. Worth mentioning is that the scoring has only looked at the concepts in isolation, whereas the testing has been done with other parts of the camera in mind. Since the fastening of the accessory is going to be supplemented with screws to prevent theft, the stability and durability gotten from the magnets were good enough when putting the accessory in place in the camera house.

4.3 Audio Signal Processing

To enable two-way communication, at least one speaker and at least one microphone is needed. With the size and round shape of the accessory in mind, it is a natural choice to use only one speaker. Multiple microphones allow for potentially better sound quality and more possibilities to analyze sound input, but also come with an increase in complexity. To keep complexity at a reasonable level for the project, the ambition was to use two microphones in the accessory. However, due to limited time, one of these was only used. The hardware in the accessory was designed to be able to use two microphones even though only one was installed.

It is necessary to convert between analog and digital signals since the Raspberry Pi will only be able to interpret a digital sound format. In addition to that, noise canceling, filtering and other operations can be made on the digital signals to achieve the best possible sound quality for output as well as input.

These signal processing tasks can be solved in multiple ways, and two possible generated solutions to the problem are presented in the following sections. One solution is more complex and potentially more suited for a real-world product as it offers more functions, while the other solution is more simple.

Digital Signal Processor (DSP)

One option for signal processing is to use a digital signal processor as described in Section 2.3. Analog Devices manufacture DSPs and one of their modern processors is called ADAU1787 [*ADAU1787 Data Sheet* 2020]. This chip makes it possible to control multiple digital microphones as well as analog and digital outputs for the speaker which is desirable. Audio input and output via I^2S are possible which is desirable since one of the most important criteria when choosing a signal processing device is the possibility to communicate via I^2S.

In the concept design for the accessory and in the scope of this project, an evaluation board for the ADAU1787 is used to evaluate and determine what functions of the processor are needed for the accessory's PCB. The evaluation board gives access to the different interfaces on the chip. The processor itself is programmable via a software called SigmaStudio. Among other functions, the ADAU1787 can be used for EQ filtering which means boosting or cutting certain frequency ranges of the sound output to suit the speaker and to reduce potential background noise from the microphones [*ADAU1787 Data Sheet* 2020]. A schematic diagram of the audio signals is shown in Figure 4.9, assuming that PDM microphones are used.

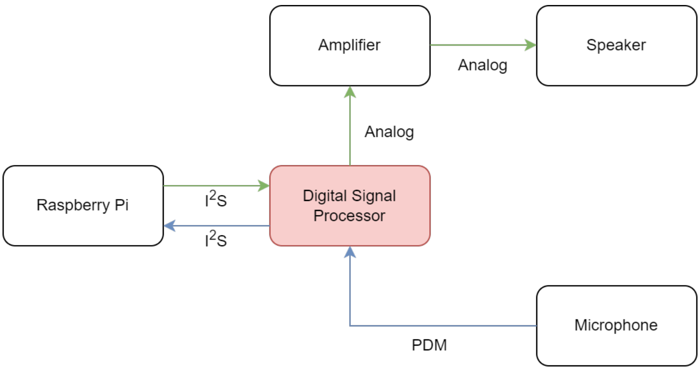

Figure 4.9 Schematic of components and the audio signals when using a digital signal processor for audio processing, red indicating where the processing is made. The Raspberry Pi is used for sending and receiving audio signals.

Direct Connection Between Microphone, Amplifier and Raspberry Pi

An alternative to using a DSP for audio processing is to do the processing on the Raspberry Pi and use other components than a DSP for audio digital-to-analog conversion. The open-source hardware company Adafruit provides a microphone breakout board with a built-in analog-to-digital converter that outputs the I^2S sound format [*Adafruit I2S MEMS Microphone Breakout* 2022]. They also offer an I^2S to analog amplifier breakout board. Both of these products are both easy to use and by using these two products, there is no need for separate components and to convert between analog and digital signals. A schematic diagram of the audio signals is shown in Figure 4.10.

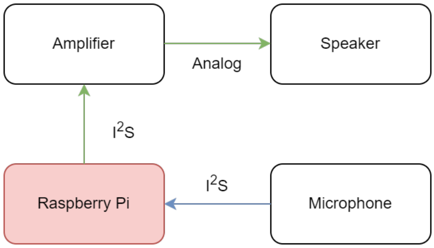

Figure 4.10 Schematic of components and the audio signals when using the system-on chip for audio processing, indicated by red.

Testing of Signal Processing

During the concept generation phase, the initial first-hand choice was to use the digital signal processor for audio processing. On paper, this solution is the better of the two in most ways. It offers the best audio processing opportunities since the DSP is designed to do exactly that. Having the processing on a chip built into the accessory, in contrast to having the processing on the system-on chip (the Raspberry Pi), also makes it easier to swap between different accessories because the software on the system-on chip does not have to be adapted to any specific accessory. Furthermore, using a DSP would also likely make the proof of concept more similar to a future real-world product.

Testing of DSP. Since there are a total of four input/output signals to and from the DSP, the testing was divided into four steps:

- Digital audio from digital microphone
- Digital audio from Raspberry Pi
- Digital audio to Raspberry Pi
- Digital audio to codec

The testing of the DSP and its associated graphical development tool SigmaStudio was a time-consuming process. Multiple weeks were spent trying to get sound in different formats to pass through the DSP. It has been very challenging to make any major progress and the work repeatedly did not proceed in the pace expected, and although there is documentation available, the area is very niched and it is hard to find information about exactly how to set up the hardware in form of inputs on the evaluation board and software in form of SigmaStudio. The connection between the digital microphone and the DSP worked, but establishing sound connections on the other three steps was never really managed. Hence, the opportunity to start with signal processing using this option was never implemented.

Testing of Direct Connection Between Microphone, Amplifier and Raspberry Pi. Some software installations and configurations on the Raspberry Pi were required to get started with the I^2S amplifier and microphone. After the installations were complete and the wiring between the appropriate pins on the Raspberry Pi, amplifier and microphone had been made, the sound input and output worked as they should. The wiring is shown in Figure 4.11.

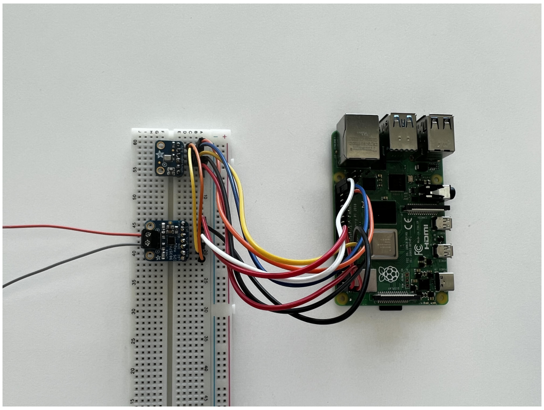

Figure 4.11 Wiring between Raspberry Pi, microphone (top left) and amplifier (bottom left).

There are opportunities for audio signal processing in the Raspberry Pi, but less straightforward compared to the DSP. The CPU on the Raspberry Pi can be used for a wide range of tasks, so writing a program for audio signal filtering is possible.

Choice of Audio Signal Processing

In terms of time, the testing phase of the project was dominated by getting the digital signal processor to work. Since there were many important parts of the project left to complete after this phase and time was running out, the easier way to go for the option to do the processing on the Raspberry Pi was chosen. By using Adafruit's microphone and amplifier instead of the DSP, the complexity was reduced but the possibilities regarding signal processing that the DSP offers were lost.

4.4 Electronic Interface and Connector

A working electronic connection between the accessory and the camera is required for the accessory's functionality. The amplifier and the microphone both need to be connected to the Raspberry Pi by a number of pins to work correctly. With the theory from Section 2.2 in mind, the following pins are needed for one I^2S microphone and one I^2S amplifier:

Microphone:

- 3 V
- Ground
- I^2S data in
- Bit clock (BCLK)
- Left-right clock (LRCLK)

Amplifier:

- 5 V
- Ground
- I^2S data out
- Bit clock (BCLK)
- Left-right clock (LRCLK)

There are five pins each needed for the microphone and the amplifier, but three of these pins (ground, BCLK and LRCLK) can be shared by the two, which reduces the required number of pins for the accessory from ten to seven. A schematic of these pins is shown in Figure 4.12. Furthermore, since it should be possible to attach different types of accessories to the camera, additional pins should be available in the accessory holder permanently attached to the camera. It is reasonable to think that other accessories would use

I^2C for communication since this protocol is world standard. As described in Section 2.3, I^2C only needs two pins for two-way communication. By adding these two pins to the previously described seven, a total of nine pins are required.

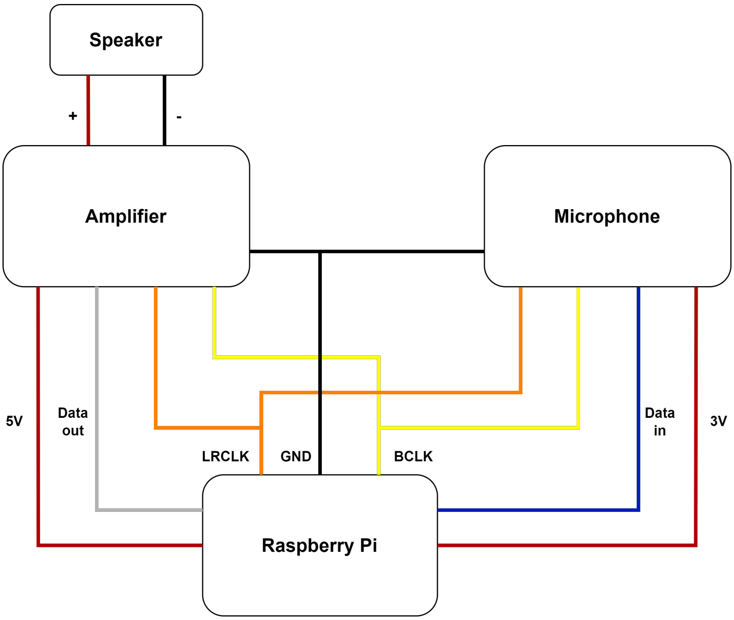

Figure 4.12 Electric circuit between camera (Raspberry Pi) and accessory (amplifier, speaker and microphone).

There are different options available when it comes to choosing a suitable electrical connector. Two common kinds of connectors are board-to-board connectors and pogo pins.

Board-to-Board Connector

Board-to-board connectors come in different shapes and sizes. They typically consist of two mating halves: a male connector and a female connector. The male connector has a series of pins that align with the corresponding pins on the female connector. When the connectors are mated, the pins from one board make contact with the pins on the other board, establishing electrical connections. The plastic cover around the pins creates a grip between the two parts to establish a secure connection. The grip makes it difficult to detach the connectors from each other, which can be an advantage or disadvantage depending on the situation. Another function of the plastic is that it guides the pins to their correct position during installation. A pair of board-to-board connectors are shown in Figure 4.13.

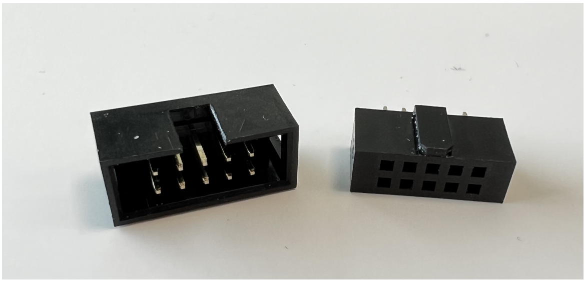

Figure 4.13 Male and female board-to-board connectors.

Pogo Pins

Pogo pins, also known as spring-loaded pins, are similar to board-to-board connectors in that they provide a number of pins for electronic connection between two units. The springs inside the pins provide the necessary force to establish a reliable electrical connection when the pins are pressed against a surface. Like board-to-board connectors, they come in various shapes and differ in pin length and pin quantity. Pogo pins are more simple to connect and disconnect compared to board-to-board connectors but require an external solution in order to align the male and female connectors to prevent unintentional disconnections. A pair of pogo pins is shown in Figure 4.14.

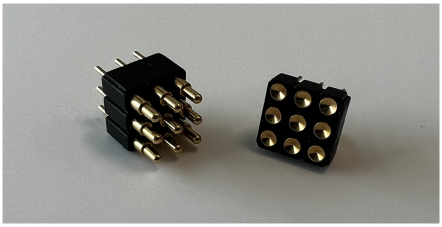

Figure 4.14 Male and female pogo pins.

Testing of Electronic Interface and Connector

Basic testing of the board-to-board connectors and pogo pins was conducted without actually installing connectors in the accessory. The required force to connect and disconnect a board-to-board connector was tested. Pogo pins of different lengths were tested and analyzed, and it was concluded that shorter pins would be preferable since that would allow the bottom surface of the accessory and the surface of the accessory holder to lean against each other when connected.

Choice of Electronic Interface and Connector

Applied to the accessory developed in this thesis, there are two main differences between the two options:

- Board-to-board connectors require considerable force for connection and disconnection while pogo pins do not.
- The two parts of a board-to-board connector must be accurately aligned before installation while pogo pins can be rotated in place.

With the results of the mechanical fastening during installation in hand, it is natural to choose pogo pins for the electronic interface. The magnets already provide enough grip to keep the accessory from accidentally falling down during installation and removal, which means that the additional grip from a board-to-board connector would be superfluous. Aligning pogo pins also works great with the magnets because the accessory will always rotate in place. Since the two magnets in the accessory have opposite polarity, the accessory will always be correctly rotated if the magnets are snapped in place. This ensures that the pins will always be correctly mated. Using pogo pins for the electronic interface, the installation and removal of the accessory is very simple. Photos of the pogo pins during and after installation in the accessory are shown in Figures 4.15, 4.16 and 4.17. Details of each pin are explained in Figure 4.18.

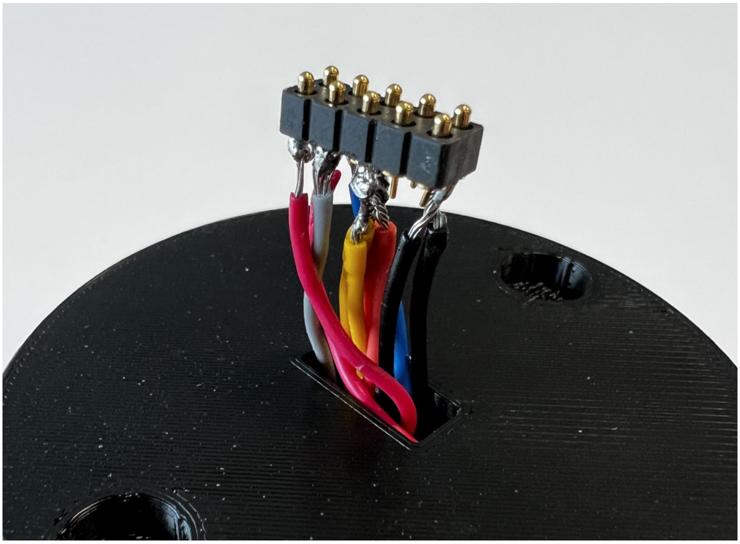

Figure 4.15 Pogo pins soldered to wires before being inserted in the accessory.

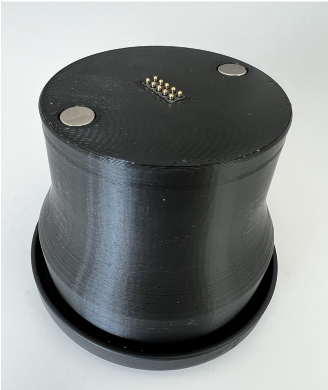

Figure 4.16 Pogo pins inserted in the accessory.

Figure 4.17 Pogo pins inserted in the bottom of the holder.

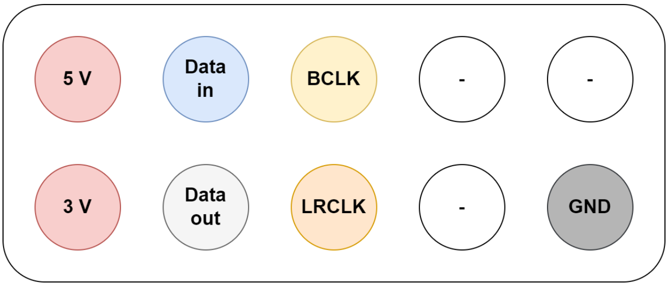

Figure 4.18 Details of the seven pins used (while looking from the accessory lid and having the installed microphone to the bottom-right).

4.5 Speaker

The speakers that have been considered for use in the accessory have primarily been selected to fit into the accessory, but also based on their sound quality and performance. Since the accessory is supposed to be an audio device with a speaker and microphones to communicate via voice, the objective was to find a speaker that could deliver a relatively high sound pressure level and have a good frequency response for conversations. Photos of the two speakers are shown in Figures 4.19 and 4.20.

Figure 4.19 The speaker in this thesis is called speaker 1.

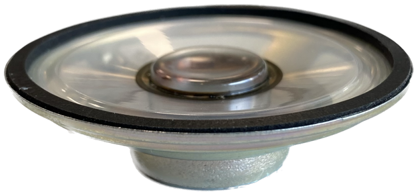

Figure 4.20 The speaker in this thesis is called speaker 2.

Table 4.3 Specifications for the two speakers considered. Data received from: [*Micro Dynamic Speaker Specification* 2021], [*W283R Speaker Specification* 2018]
Note: Speaker Sensitivity converted with [*Distance Attenuation Calculator* 2023]

	Speaker 1	**Speaker 2**	**Unit**
Speaker Sensitivity	88 ± 3	79 ± 3	dB SPL @1m/1W
Impedance	$4 \pm 15\%$	$8 \pm 20\%$	Ω

Table 4.3 shows that speaker 1 is having significantly higher sound pressure level which is preferred in this case. Only looking at this characteristic, this speaker would be the better choice.

As mentioned in Section 2.2, it is important that the evaluation of a frequency response takes the use case into consideration. In general, a speaker with a relatively even response in the range of interest is going to reproduce a sound very true to life and this is something that is preferable in this case. This combined with good response in the span of voice frequencies has been the foundation when choosing which speaker that has the best frequency response.

Looking at the frequency responses for the two speakers in Figures 4.21 and 4.22, it can be seen that they are relatively similar in the case that the lower frequencies are attenuated. In the range of 100-4000 Hz, which is a large part the conversations frequency span, speaker 2 has a relatively even frequency response, slightly better than speaker 1. At frequencies above 4000 Hz, speaker 2 has a bit more uneven response than speaker 1. However, advantages for speaker 1 over 4000 Hz does not weigh up for the fact that speaker 2 has a higher maximum sound pressure level which makes speaker 2 the best choice.

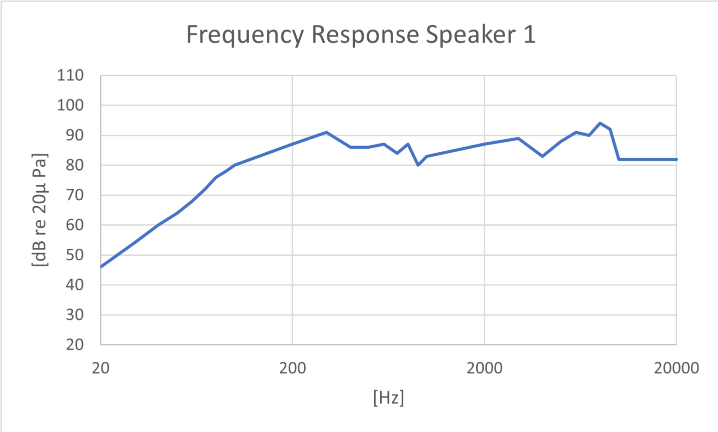

Figure 4.21 Frequency response of speaker 1. Reproduced from: [*Micro Dynamic Speaker Specification* 2021]

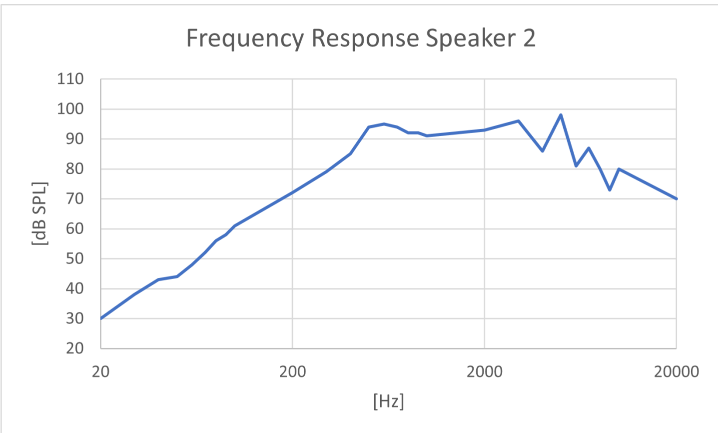

Figure 4.22 Frequency response of speaker 2. Reproduced from: [*W283R Speaker Specification* 2018]

4.6 Microphone

There is a variety of microphones that output different sound protocols, e.g., I^{2}S microphones, PDM microphones and analog microphones as described in Section 2.2. Since analog microphones are more prone to electromagnetic interference that can cause distortion in audio signals [*Analog and Digital MEMS Microphone Design Considerations* 2023] and the accessory will include many electric components in a small space, digital microphones are preferable. As mentioned in Section 2.3, Raspberry Pi has support for I^{2}S sound, and therefore a digital I^{2}S microphone is suitable for this project.

The accessory to be built also has limited space available, which puts requirements on the size of the components used. A technology that can deliver sound quality that meets this project's requirements and at a smaller size than other technologies is MEMS microphones. This has been described in Section 2.3 where the MEMS microphones are also said to have high stability against external impact.

The microphone that has been chosen is the Adafruit MEMS Microphone breakout board [*Adafruit I2S MEMS Microphone Breakout* 2022]. This microphone was chosen because it supports the I^{2}S sound protocol and the breakout board makes it easier to mount it in the accessory as well as soldering wires to it. A selection of technical details for the microphone is presented in Table 4.4 and photos of the microphone can be seen in Figures 4.23 and 4.24.

Table 4.4 Specifications for the microphone considered. Data received from: [*Datasheet SPH0645LM4H-B* 2015]

	Conditions	**Min**	**Typ.**	**Max**	**Unit**
Sensitivity	94 dB SPL @ 1kHz	-29	-26	-23	dBFS
Signal to Noise Ratio	94 dB SPL @ 1kHz, A-weighted		65	-	dB(A)

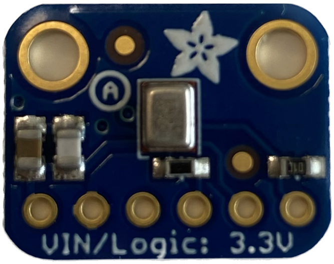

Figure 4.23 I²S MEMS microphone breakout board from Adafruit, top side.

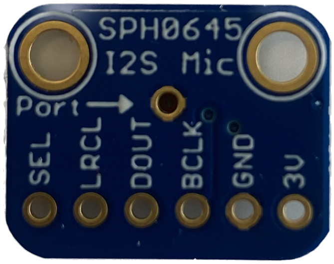

Figure 4.24 I²S MEMS microphone breakout board from Adafruit, bottom side.

4.7 Amplifier

To get the most out of the speaker in the accessory it is necessary to have an amplifier in the circuit between the Raspberry Pi and the speaker. Since the Raspberry Pi supports the I^2S sound protocol and the ambition has been to use this, the goal was to find an amplifier communicating via this protocol. A suitable amplifier is Adafruit I^2S 3W Class D Amplifier Breakout MAX98357A [*Adafruit MAX98357 I2S Class-D Mono Amp* 2023]. The amplifier is, as for the microphone, assembled on a breakout board which eases the installation. A photo of the amplifier can be seen in Figure 4.25 and its specifications are presented in Table 4.5.

Table 4.5 Specifications for the amplifier considered. Data received from: [*Datasheet MAX98357A/MAX98357B* 2016]

Voltage supplied	Impedance	Output Power
5 V	4 Ω 8 Ω	3.2 W 1.8 W

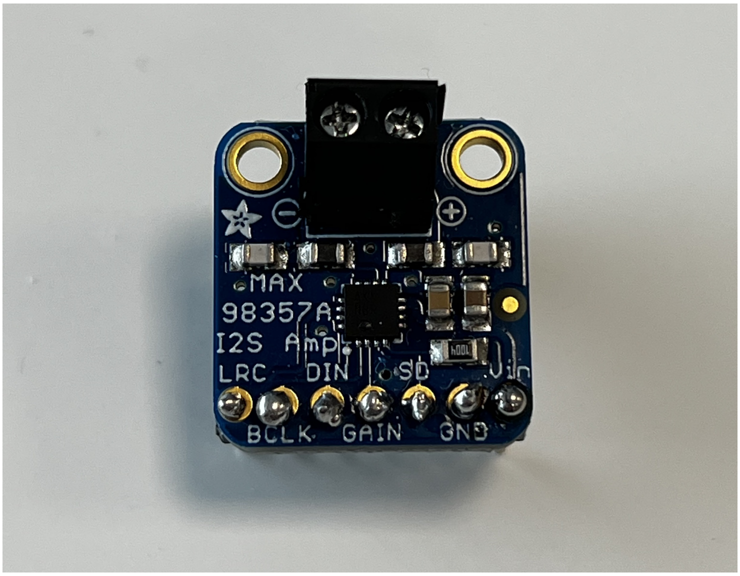

Figure 4.25 I2S Amplifier breakout board from Adafruit.

4.8 Software

In order to use the accessory, a working software program is needed. The program will run on the Raspberry Pi OS Unix-like operating system. The program must be able to control the different functions of the accessory described in the Problem Formulation, Section 1.2. To do this, three classes have been established for playing audio files, recording audio files and detecting audio input. In addition to these classes, a graphical user interface has been developed for convenient control of the features.

Python [*Applications for Python* 2023] has been chosen as the programming language for all classes. Python is an easy-to-use language and there is a wide range of code libraries available to use for various tasks, including various libraries for audio handling. The file format used both for recording and playback is the WAV file format, an uncompressed file format used to store digital sound [Whibley, 2016].

The code for each of the four Python files can be found in Appendix A.

Playing Audio Files

As described in the Problem Formulation in Chapter 1.2, the camera could be mounted in public areas. In those kinds of environments, it is reasonable for the user of the camera and accessory to be able to play prerecorded audio files, for instance when a trespasser is detected on the video stream.

The `audio_player` class defines an audio player. It uses the open source code library `pygame` and their module `pygame.mixer` [Pete Shinners, 2023] for playing WAV audio files. The program consists of methods for opening a WAV file, for playing and pausing the open clip and a method for volume adjustment.

Recording Audio Files

As mentioned in Section 1.3, the accessory should be able to be controlled remotely. When the operator of the camera remotely detects an interesting event, either from the video stream or from the sound input, it should be possible to record audio and save the recorded data as a WAV file. The class `audio_recorder` handles this task. It uses the `wave` module and the module `pyaudio` [Hubert Pham, 2022] from the Python Standard Library for audio recording. The class `audio_recorder` simply contains two methods in addition to the constructor: A `start_record` and `stop_record` method. These let the user start an audio recording at any given point in time, record for an arbitrary time and stop the recording and save it as a WAV file as soon as the `stop_record` method is called. The file is saved as "recording_" plus the time of the day in a designated folder.

Audio Detection

Another feature that has been implemented in the program is the real-time plotting of the amplitude of the sound input. The code for this class is almost entirely adopted from example `plot_input` from the `python-sounddevice` module on GitHub [Geier, 2023]. This feature plots the amplitude of the incoming sound in real-time, which is useful for detecting unusually loud sounds. The audio detection feature also flags when the amplitude exceeds a threshold by printing a warning message in the terminal. This threshold is continuously calculated and updated by storing the average amplitude in a variable.

Graphical User Interface

The graphical user interface (GUI) gives the user of the camera the opportunity to control the functions of the accessory. The GUI for the playback and recording features has been created in Python using the `tkinter` package [*tkinter — Python interface to Tcl/Tk* 2023]. The library `tkinter` is a standard GUI library for Python and is easy to use. This graphical user interface consists of a window with two sections: One section for audio playback and one section for audio recording.

In the audio playback section, buttons can be pressed to play stored audio clips from the hard drive in the speaker. There is also a button for pausing an active clip or resuming a clip that has been paused. Furthermore, there is a slider for adjusting the playback volume. In the audio recording section, there is one button for starting an audio recording and one button for stopping an ongoing recording. The GUI can be seen in Figure 4.26.

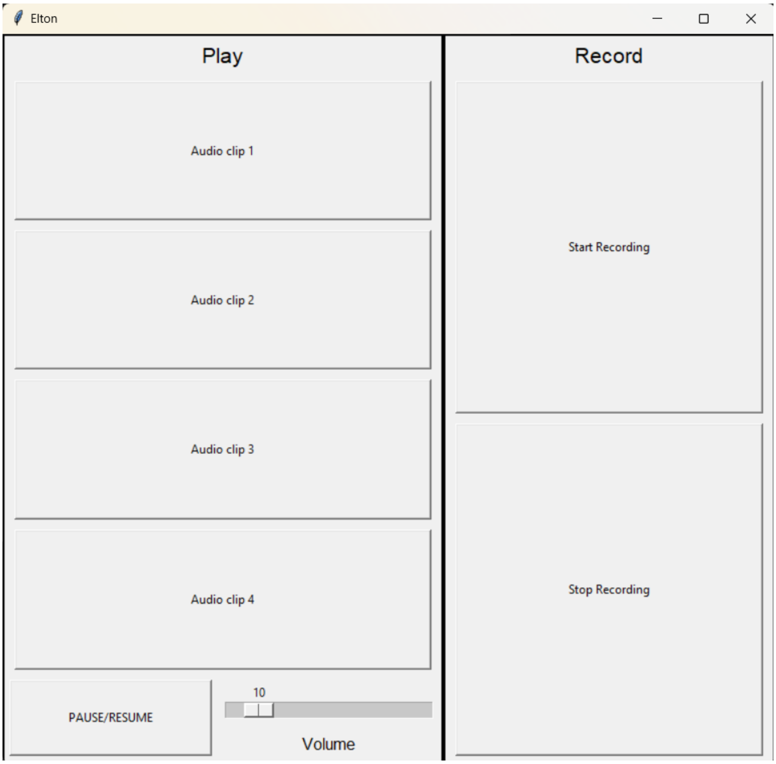

Figure 4.26 The graphical user interface.

The audio detection graph is plotted in a separate window. This graph is created using `matplotlib` for plotting. The graph in the window shows the amplitude of the audio input for the past 10 seconds. A snapshot of an example of the graph is shown in Figure 4.27.

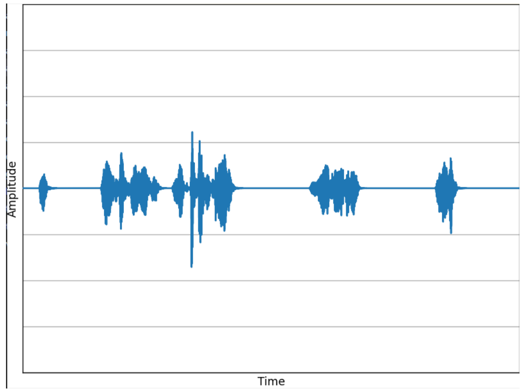

Figure 4.27 Plot of the audio input amplitude in real-time.

There are limitations with the current software as the audio detection feature can not run simultaneously with the other GUI window. Another major flaw with the audio detection feature is that it does not show any unit of the volume on the y-axis or of the time on the x-axis.

In a real-world product, the graphical user interface would not run on the system-on chip but for the sake of showing the concept and reducing complexity, this is where the GUI of the program is running. By doing so, there is no need to involve a connection between the user's computer and the Raspberry Pi.

4.9 Final Prototype

Remaining Parts

In this section, the remaining parts of the prototype that have not been mentioned before will be presented. The design of each part will be described and also how it will function together with the other parts in the prototype. At the end of the section, the prototype as a whole will be presented.

Accessory Holder permanently attached to Camera. As mentioned in Section 4.1, the accessory holder permanently attached to the camera consists of what can be seen as a shell for the accessory itself. The holder is designed to fit into the existing camera house and also to not interfere with any other parts mounted in the same housing.

Since the magnets were the concept chosen for the fastening during installation, magnets have been inserted into the bottom plate of the holder. On the same plate, there is also an opening for fastening the electronic connector with a press fitting. On the upper edge, there are holes aligned with equal-sized holes on the accessory to enable additional fastening with the ambition to prevent theft. CAD models of the holder are shown in Figures 4.28 and 4.29.

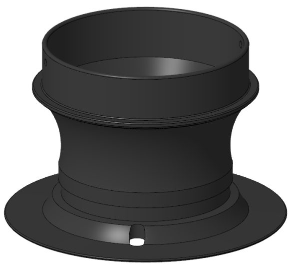

Figure 4.28 Illustration of the accessory holder attached to the camera.

Figure 4.29 Illustration of the bottom of the accessory holder attached to the camera.

Accessory Chassis. The chassis of the accessory is supposed to be mounted and fit into the holder and has therefore been designed for this purpose. The shape has been optimized to minimize the volume between the accessory and the holder when it is installed.

In the same way as for the holder, the accessory chassis has magnets inserted in the bottom lining up with the ones inserted in the holder. When the accessory is installed, there is some material between the two pairs of magnets which allows for adjusting the strength and also minimizing the risk of the magnets loosening from one of the parts. The CAD model can be seen in Figures 4.30 and 4.31.

Figure 4.30 Illustration of the top of the accessory chassis.

Figure 4.31 Illustration of the bottom of the accessory chassis.

Speaker Enclosure Box. The speaker enclosure has been done by making a closed box for the speaker. This has been designed to fit into the accessory chassis and to be as big as possible. The speaker is placed on the edge of the enclosure box and the wires are led through the holes in the bottom, which in turn is sealed with adhesive material. An illustration of the box can be seen in Figure 4.32.

Figure 4.32 Illustration of the speaker enclosure box.

Speaker Clamp. To lock the speaker in its place and secure that the enclosure box is sealed, a speaker clamp has been designed. This clamp is mounted on top of the speaker and holds the speaker in place between the enclosure box and the clamp. The microphones are also supposed to be mounted on the clamp which has been designed so the microphone ports are angled to capture sounds from different angles. An illustration of the clamp can be seen in Figure 4.33.

Figure 4.33 Illustration of the speaker clamp.

Accessory Lid. To cover the speaker and microphones, a lid has been designed. The lid's cover has a mesh designed to allow sound to go in and out from the accessory and at the same time cover the electronic parts.

Screw holes that align with the ones in the accessory chassis, speaker enclosure box and speaker clamp have been added so that all those four parts can be put together with the same screws.

As for the accessory chassis and the accessory holder, screw holes have also been added on the side of the lid to enable additional fastening in addition to magnets. The lid is shown in Figure 4.34.

Figure 4.34 Illustration of the accessory lid.

Complete Prototype

The detachable audio accessory consists of the accessory chassis, speaker enclosure box, speaker clamp and accessory lid. In addition to this, the speaker chosen in Section 4.5, two of the microphones presented in Section 4.6 and the amplifier presented in Section 4.7 has been mounted onto the accessory which is shown in Figures 4.35 and 4.36. The wires from these electrical components have been soldered to a pogo-connector, which functions as the interface between the accessory and the camera. An illustration of the final prototype and the final 3D printed prototypes are shown in Figures 4.37 and 4.38.

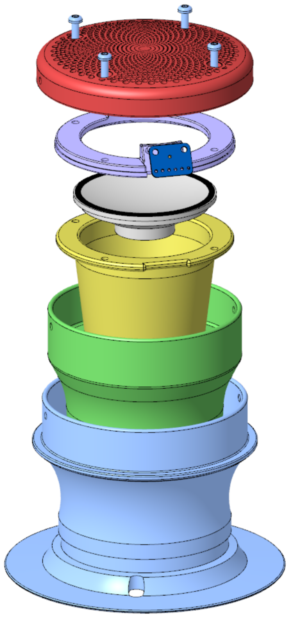

Figure 4.35 Illustration of the parts of the final prototype.

Figure 4.36 The speaker box together with the speaker and the speaker clamp with microphones installed on the clamp. The amplifier is mounted on the speaker box.

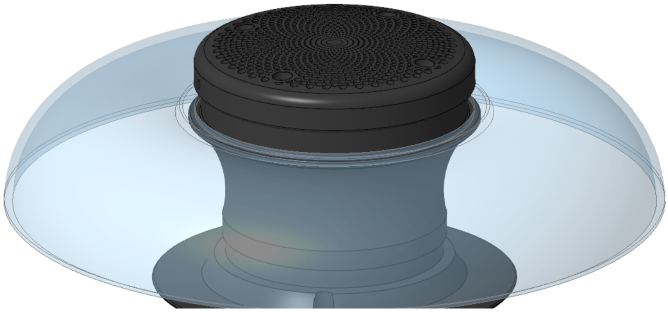

Figure 4.37 Illustration of the final prototype with dome.

Figure 4.38 The final prototype of the accessory and its holder.

5

Discussion

One of the main focuses for this thesis as described in Section 1.2 has been to be able to play sound from the accessory. The sound that was going to be played should be of good quality for speech and be heard clearly at a reasonable distance. The use of a Raspberry Pi with its GPIO pins, making it possible to communicate via I^2S, together with the chosen speaker made it possible to play sound with good quality. As has been described in Section 4.5, the speaker that was chosen reproduces sound in the voice frequency range well. The sound pressure level that is given with this speaker is also at a good level, which makes it possible to hear the sound from a reasonable distance. Of course, there are speakers with both better frequency response and higher sound pressure levels, but in this case, the size has been an important factor since the available space has been limited. It is worth keeping in mind that with a smaller accessory, the sound pressure level would probably be lower, so this can be seen as a trade-off in sound quality and size. When similar accessories are going to be developed in the future, this is therefore important to consider and keep in mind when the accessory is being designed.

Another goal listed in the Problem Formulation was the ability to detect and record sound input. In the final prototype, two microphones are mounted in the accessory but only one of them is connected to the Raspberry Pi, which was a trade-off made because of lack of time. Multiple working microphones would be preferable, since this, among other features, would allow for the identification of the direction of sound input, if the microphones were directed in different directions. The microphone was chosen because it was a good replacement when the idea of using a DSP was dismissed and works well to prove the concept, but more effort can be put into choosing a suitable microphone for a real-world product.

The Python program is able to record the audio input for an arbitrary time which is a feature that works well. The audio detection feature has potential but needs improvement. The code example used for this feature is rather advanced and hard to modify to fit the needs perfectly. Nevertheless, the plotting shows that there are possibilities to implement a similar feature in the future.

The signal processing of the audio input and output signals was a big challenge. The option to use a digital signal processor as described in Section 4.3 would have been a great addition to the product, and it is unfortunate that the DSP did not work as intended despite all the time and effort that was put into it during the product development process. Among other features, a DSP could have been used for equalization to fit the chosen speaker to achieve a more balanced sound. Luckily, the prototype still works well when it comes to audio input and output thanks to alternative components that were used.

Another limitation with the prototype that could have been solved with signal processing is that it is currently not possible to record and play sound simultaneously without recording the sound played by the speaker. By canceling out this sound with signal processing, it would have been possible to continue recording even when the speaker is playing.

An important part of the project was the electronic interface between the accessory and the camera, a problem that was solved using pogo pins. The solution works neatly for the audio accessory described in this thesis as well as for other potential future accessories. The solution reduces complexity during installation substantially compared to other potential methods. In Section 4.4, it is described that nine pins are required, but the number of pins can easily be increased or reduced by using other pogo pin connectors.

This accessory has been designed to fit geometrically into the void inside the four camera sensors and the size and shape have been optimized to fit into this space. In addition to this, the fastening and the electronic interface have been chosen and designed to increase the modularity of future accessories having other functions. Modularity is something that has been kept in mind during the whole development process. The shape and electronic interface of the prototype show that the accessory is exchangeable with other types of accessories, and at the same time optimized for this particular camera. However, since the design is customized for the camera used in this project, some design changes will have to be made if the accessory is going to fit with other cameras. These design changes are probably not only going to be mechanical, but also some electronics may need to be replaced. For example, the speaker may not fit into a similar accessory in a different camera. Even if the accessory that has been made, does not fit as well in other camera models, its mechanical, electrical and software design allows for modularity with other types of accessories in this particular camera. From the start, this has been the goal and the result is a modular accessory designed to fit geometrically in the void inside the camera sensors.

One of the other main focuses has been to design an accessory that is detachable. This has been realized by using magnets together with pogo pins. This makes it very easy to detach and install the accessory, which is in line with the desires from the beginning. The fact that magnets and pogo pins make the accessory very easy to detach also comes with an increased risk of thieving. This is the reason why additional fastening has been added and this is thought to be made with some kind of special tool to secure it even more. Even though this fastening with a special tool has not been fully realized, this has been added to show the idea that additional fastening is required when magnets are used. An alternative would be to use a different fastening method than magnets during installation to decrease the risk of theft already at this step. The optimal solution would be to have only one type of fastening that is easy to detach and makes the accessory difficult to steal. In Section 4.2, alternatives to magnets have been presented and explored but they have been assessed not to be enough to use as the only fastening method because of the risk of thieving. Magnets were chosen because of the ease of installation and also because the pogo connector aligns correctly when the magnets are used. The electrical interface is then connected with high accuracy which is a big advantage.

6

Future Work and Conclusion

6.1 Future Work

In this kind of product development project, there will always be room for improvement and further development. During this project, choices have been made to focus on several areas and aspects to build a functional prototype which has left others for the future. Some of these areas of improvement and additional work to develop the product, even more, will therefore be presented in this section.

Making a real PCB. A product like the accessory built in this project consists of several different electronic components. In complete products, a real PCB will therefore be included. This should be designed to fit into the accessory with all the necessary components. The components that are necessary on a complete PCB will for example be an amplifier and a DSP making signal processing possible. When the accessory is installed in the camera housing, communication between the accessory and the camera housing's PCB also needs to function. On the accessory's PCB, either the pogo connector itself or connecting wires to it, therefore needs to be included.

Protection against external environment. One important feature in complete products, especially if they are going to be used both indoors and outdoors, is that they can sustain an impact from the surrounding environment.

For example, it is of high importance that the electrical components do not get in contact with any moisture or dust. The protection can be done by using gaskets and sealing in the joints between the different parts.

Not only should a product sustain dust and moisture, but also for example vibrations and other types of mechanical impact. This has been thought of when designing the fastening but more work has to be done to reach a certain classification which is preferable in protection against the external environment. This work has been hard to implement in this project because of the lack of time, but the importance of it has been well understood and kept in mind during the design process.

Signal processing. More advanced signal processing is necessary to make the product work in environments with a lot of noise. In such environments, the signal processing can filter out certain frequencies and amplify the desired signal. In the future, it is recommended to implement some kind of equalizer and filtering for this purpose. It would also be good to be able to record and play sounds simultaneously, but this will also require more advanced signal processing.

Power consumption. The prototype is powered by the Raspberry Pi, which is not going to be the case in the final product. In a final product, the accessory and the camera will most likely be powered from the same source. To make a detailed calculation on the power consumption for the accessory is therefore important so the power supplied will be enough for the camera and the accessory together.

Investigate parts further. Some of the parts in the prototype have been evaluated, analyzed and designed more accurately than others. For example, further development could be made on the loudspeaker enclosure box where the box currently is designed to fit in the accessory chassis and contributes to a sound quality seen to be good enough. A speaker box can be designed more carefully by running simulations with different input parameters resulting in different-looking frequency response diagrams as output. In the speaker box used in this prototype, the inside walls do not have any lining with absorbing material which would increase the quality even more.

Another feature of the parts that could be developed further in the future is the additional fastening created to prevent thieving. In Chapter 5, it was discussed that an additional fastening requiring a special tool has not been realized but this is something that would have decreased the risk of thieving and is therefore recommended to be further investigated.

Cost Estimation. During this thesis, any comparisons on cost between different parts have not been made. The parts have therefore been chosen from other selection criteria. For a future product, a cost estimation of the whole product will be necessary. This could lead to different choices where some parts may be too expensive.

Choice of material and manufacturability. Since the parts in the prototype have been produced mainly by 3D printing, there have been limitations in choosing material and the parts have been printed in plastic material. The components in the future will probably differ in the material used and this will depend on functionality, cost and manufacturability.

The design of the parts can also be developed to improve the manufacturability of the whole prototype. Depending on which manufacturing method is going to be used, there will be different important factors to take into consideration so the product is designed for manufacturing.

Stereo microphones. In the current prototype, there is only one microphone used. This limits the opportunities for certain signal processing so for future products, there will be a need for at least two microphones. With multiple microphones, it will be possible to analyze the sound in a more sophisticated way, for example in which direction the sound source is located.

More advanced software features. The software used in this project can be developed further by introducing more advanced functions. An example can be to start recording when a loud sound is detected. When a loud sound is detected, there could also be a certain voice message going out from the speaker. Another function that would be good to implement is to be able to have two-way communication in real-time and not just only play pre-recorded sound. For that to be possible, the software would need to be updated to allow the speaker to playback sound input from a remote microphone in real-time.

6.2 Conclusion

The prototype that has been made in this Master's Thesis shows that it is possible to build a similar accessory as a real product. Since it is detachable and modular, it can be exchangeable to get different functions for different matters. Audio signal processing can be developed further to get better sound quality, for example by using a DSP. The mechanical and electrical interfaces are set for an accessory made for the camera used in this project and the software makes it possible to play, detect and record sounds that are in line with the goals of the thesis.

Bibliography

Adafruit I2S MEMS Microphone Breakout (2022). Adafruit Industries. URL: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-i2s-mems-microphone-breakout.pdf.

Adafruit MAX98357 I2S Class-D Mono Amp (2023). Adafruit Industries. URL: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-max98357-i2s-class-d-mono-amp.pdf.

ADAU1787 Data Sheet (2020). Rev. A. Analog Devices.

Analog and Digital MEMS Microphone Design Considerations (2023). URL: https://invensense.tdk.com/wp-content/uploads/2015/02/Analog-and-Digital-MEMS-Microphone-Design-Considerations9.pdf (visited on 2023-05-08).

Analog Devices (2023). URL: https://www.analog.com/en/design-center/glossary/class-d.html (visited on 2023-05-19).

Analog Dialogue (2023). URL: https://www.analog.com/en/analog-dialogue/articles/class-d-audio-amplifiers.html (visited on 2023-05-19).

Applications for Python (2023). URL: https://www.python.org/about/apps/ (visited on 2023-05-29).

Capacitance Sensors for Human Interfaces to Electronic Equipment (2023). URL: https://www.analog.com/media/en/analog-dialogue/volume-40/number-4/articles/capacitance-sensors-for-human-interfaces-to-electronics.pdf (visited on 2023-05-08).

Datasheet MAX98357A/MAX98357B (2016). Rev. 7. Maxim Integrated.

Datasheet SPH0645LM4H-B (2015). Rev. B. Knowles.

Dickason, V. (2006). *Loudspeaker Design Cookbook 7th edition*. KCK Media Corp. Chap. 0.

Distance Attenuation Calculator (2023). URL: `https://www.omnicalculator.com/physics/distance-attenuation` (visited on 2023-05-02).

Fantel, H. (1986). "Sound; the importance of the loudspeaker box". *New York Times*. Available at: https://www.nytimes.com/1986/10/19/arts/sound-the-importance-of-the-loudspeaker-box.html (visited on 2023-05-22), p. 25.

Geier, M. (2023). *Python-sounddevice*. `https://github.com/spatialaudio/python-sounddevice`.

Hubert Pham (2022). *Pyaudio*. Version 0.2.13. URL: `https://people.csail.mit.edu/hubert/pyaudio/`.

I^2S bus specification (2022). UM11732. Rev. 3.0. NXP Semiconductors.

I2C-bus specification and user manual (2021). Rev. 7.0. NXP Semiconductors.

Lindroos, V. (2020). *Handbook of Silicon Based MEMS Materials and Technologies (Third Edition)*. Elsevier. Chap. 48.

Long, M. (2014). *Architectural Acoustics Second Edition*. Elsevier/Academic Press. Chap. 2.

Mic Basics: What is Frequency Response? (2023). URL: `https://invensense.tdk.com/wp-content/uploads/2015/02/AN-1112-v1.1.pdf` (visited on 2023-02-20).

Micro Dynamic Speaker Specification (2021). TRAM66004S-100F. Edition 1.2. Kingstate Electronics Corp.

Microphone Specifications Explained (2023). URL: `https://invensense.tdk.com/wp-content/uploads/2015/02/AN-1112-v1.1.pdf` (visited on 2023-02-17).

PDM vs. I^2S: Comparing Digital Interfaces in MEMS Microphones (2023). URL: `https://www.cuidevices.com/blog/pdm-vs-i2s-comparing-digital-interfaces-in-mems-microphones` (visited on 2023-02-17).

Pete Shinners (2023). *Pygame*. Version 2.3.0. URL: https://www.pygame.org/docs/ref/mixer.html.

Raspberry Pi 4 Product Brief (2021). Raspberry Pi Trading Ltd. URL: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

Raspberry Pi Audio (2023). URL: https://www.raspberrypi.com/documentation/accessories/audio.html (visited on 2023-05-08).

Raspberry Pi GPIO pins (2023). URL: https://projects.raspberrypi.org/en/projects/physical-computing/1 (visited on 2023-05-19).

Speaker Sensitivity (2023). URL: https://support.klipsch.com/hc/en-us/articles/360025619391-Speaker-Sensitivity (visited on 2023-02-17).

tkinter — Python interface to Tcl/Tk (2023). URL: https://docs.python.org/3/library/tkinter.html (visited on 2023-05-29).

Ulrich, K. T. and S. D. Eppinger (2012). *Product Design and Development Fifth Edition*. McGraw-Hill.

Understanding Speaker Frequency Response (2023). URL: https://www.audioreputation.com/understanding-speaker-frequency-response/ (visited on 2023-02-22).

W. Smith, S. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.

W283R Speaker Specification (2018). Version A/0. Acoustic Professor.

What is a Raspberry Pi? (2023). URL: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/ (visited on 2023-05-19).

What is Sound Pressure Level (SPL) and how is it measured? (2023). URL: https://pulsarinstruments.com/news/sound-pressure-level-and-spl-meters/ (visited on 2023-02-20).

Whibley, S. (2016). *Wav format preservation assessment*. https://wiki.dpconline.org/images/archive/4/46/20160217163258!WAV_Assessment_v1.0.pdf.

A

Python Program

A.1 Main Script

```python
from gui import gui
import audio_detector as ad # start audiodetector

def main():

    try:
        pass
        gui(10) # set initial volume to 10 percent
    except Exception as e:
        print("Error!")
        print(e)

if __name__ == "__main__":
    main()
```

A.2 Graphical User Interface

```
import tkinter as tk
from audio_player import audio_player
from audio_recorder import audio_recorder

class gui:

    def __init__(self, start_vol):

        # Create the main window
        self.root = tk.Tk()
        self.root.title("Elton")
        self.root.geometry("500x500")

        # Create recorder and player
        self.ar = audio_recorder(root=self.root)
        self.ap = audio_player(start_vol)

        self.start_vol = start_vol

        # Create a frame to hold the left section (Play)
        play_frame = tk.Frame(self.root, width=250, height=500, bd=2, relief=tk.SOLID)
        play_frame.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

        # Create a label for the left section title (Play)
        play_label = tk.Label(play_frame, text="Play", font=("Helvetica", 16))
        play_label.pack(side=tk.TOP, padx=5, pady=5)

        # Create the play buttons in the play section
        clip_button_1 = tk.Button(play_frame, text="Trespassing", command=lambda: self.ap.new_clip("announcements/trespassing.wav"))
        clip_button_1.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)
        clip_button_2 = tk.Button(play_frame, text="Barking", command=lambda: self.ap.new_clip("announcements/barking.wav"))
        clip_button_2.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)
        clip_button_3 = tk.Button(play_frame, text="Hallau", command=lambda: self.ap.new_clip("announcements/hallau.wav"))
        clip_button_3.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)
        clip_button_4 = tk.Button(play_frame, text="Six languages", command=lambda: self.ap.new_clip("announcements/haspeech.wav"))
        clip_button_4.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)

        # Create a normal button in the left section with the same height as the left column buttons
        pause_button = tk.Button(play_frame, text="PAUSE/RESUME", command=lambda:self.ap.play_pause())
        pause_button.pack(side=tk.LEFT, padx=5, pady=5, fill=tk.BOTH, expand=True)
```

```python
# Create a label for the volume slider
volume_label = tk.Label(play_frame, text="Volume", font=("Helvetica", 12))
volume_label.pack(side=tk.BOTTOM, padx=5, pady=5)

# Create a scale widget (slider) for volume control
volume_slider = tk.Scale(play_frame, from_=0, to=100, orient=tk.HORIZONTAL, command=lambda v: self.ap.set_volume(v))
volume_slider.set(self.start_vol)
volume_slider.pack(side=tk.RIGHT, padx=5, pady=5, fill=tk.BOTH, expand=True)

# Create a frame to hold the right section (Record)
record_frame = tk.Frame(self.root, width=250, height=500, bd=2, relief=tk.SOLID)
record_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

# Create a label for the right section title (Record)
record_label = tk.Label(record_frame, text="Record", font=("Helvetica", 16))
record_label.pack(side=tk.TOP, padx=5, pady=5)

# Create the two buttons in the right section with the same height as the left column buttons
#button6 = tk.Button(record_frame, text='Start Recording', command=lambda: self.ar.start_record())
button6 = tk.Button(record_frame, text='Start recording', command=lambda: self.ar.start_record())
button6.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)
#button7 = tk.Button(record_frame, text='Stop Recording', command=lambda: self.ar.stop_record())
button7 = tk.Button(record_frame, text='Stop recording', command=lambda: self.ar.stop_record())
button7.pack(pady=5, padx=10, fill=tk.BOTH, expand=True)

# Bind the left section's width to the root window's width
def on_configure(event):
    width = event.width // 2
    play_frame.configure(width=width)

    # Make the left column buttons height change dynamically as window changes
    play_frame.configure(height=play_frame.winfo_height()-40)

self.root.bind('<Configure>', on_configure)

# Start the main event loop
self.root.mainloop()
```

A.3 Audio Player

```python
import pygame
from pygame.locals import *

class audio_player:
    def __init__(self, start_vol):
        pygame.mixer.init()
        self.playing = False
        self.open = False
        self.clip = None
        self.vol = start_vol

    # Starts to play a WAV file
    def new_clip(self, new_path):
        # stop current clip if one is already open
        if self.open:
            self.clip.stop()

        self.clip = pygame.mixer.Sound(new_path)
        self.open = True
        self.set_volume(self.vol)

        self.__play()

    # private method
    def __play(self):
        self.clip.play(-1)
        self.playing = True

    # starts playing if clip is paused, or pauses if clip is playing
    def play_pause(self):
        # do nothing if there is no open clip
        if not self.open:
            return
        # pause if clip is playing
        if self.playing:
            self.playing = False
            self.clip.stop()
        # play if clip is paused
        else:
            self.__play()
```

```
    # sets volume to value (number between 0 and 100)
    def set_volume(self, value):
        self.vol = value

        # set volume only if clip exists
        if self.open:
            self.clip.set_volume(float(int(value) / 100))
```

A.4 Audio Recorder

```
import tkinter as tk
import pyaudio
import wave
from datetime import datetime
import time

class audio_recorder:
    def __init__(self, root):

        # initialize attributes

        self.root = root # set GUI
        self.chunk = 1024 # 1024 bits per chunk
        self.format = pyaudio.paInt32 # 32 bits
        self.channels = 1 # mono sound
        self.rate = 44100 #sample rate
        self.p = pyaudio.PyAudio()
        self.frames = [] # array to hold audio data
        self.stop = False

    # starts recording and saves recording after user has pressed stop button
    def start_record(self):
        self.stop = False
        self.frames = []
        stream = self.p.open(format=self.format, channels=self.channels, rate=self.rate, \
                             input=True, frames_per_buffer=self.chunk, input_device_index=0)
        while self.stop == False:
            data = stream.read(self.chunk)
            self.frames.append(data)
            print("recording...")
```

```
            self.root.update() # check if stop button has been pressed

        stream.close()
        file_name = 'recording_' + datetime.now().strftime("%H:%M:%S") + '.wav'

        wf = wave.open('recorded/' + file_name, 'wb')
        wf.setnchannels(self.channels)
        wf.setsampwidth(self.p.get_sample_size(self.format))
        wf.setframerate(self.rate)
        wf.writeframes(b''.join(self.frames))
        wf.close()

        print(f"Recording stopped and saved as \"{file_name}\".")

    # sets stop variable to True to end recording
    def stop_record(self):
        self.stop = True
```

A.5 Audio Detector

This code was copied from example `plot_input` from the `python-sounddevice` module on GitHub [Geier, 2023] and then marginally modified to fit the needs of the prototype.

```
import argparse
from datetime import datetime
import queue
import sys
from matplotlib.animation import FuncAnimation
import matplotlib.pyplot as plt
import numpy as np
import sounddevice as sd

def int_or_str(text):
    """Helper function for argument parsing."""
    try:
        return int(text)
    except ValueError:
        return text

# Choose a microphone by its ID
```

```python
sd.default.device = 0

parser = argparse.ArgumentParser(add_help=False)
parser.add_argument(
    '-l', '--list-devices', action='store_true',
    help='show list of audio devices and exit')
args, remaining = parser.parse_known_args()
if args.list_devices:
    print(sd.query_devices())
    parser.exit(0)
parser = argparse.ArgumentParser(
    description=__doc__,
    formatter_class=argparse.RawDescriptionHelpFormatter,
    parents=[parser])
parser.add_argument(
    'channels', type=int, default=[1], nargs='*', metavar='CHANNEL',
    help='input channels to plot (default: the first)')
parser.add_argument(
    '-d', '--device', type=int_or_str,
    help='input device (numeric ID or substring)')
parser.add_argument(
    '-w', '--window', type=float, default=200, metavar='DURATION',
    help='visible time slot (default: %(default)s ms)')
parser.add_argument(
    '-i', '--interval', type=float, default=30,
    help='minimum time between plot updates (default: %(default)s ms)')
parser.add_argument(
    '-b', '--blocksize', type=int, help='block size (in samples)')
parser.add_argument(
    '-r', '--samplerate', type=float, help='sampling rate of audio device')
parser.add_argument(
    '-n', '--downsample', type=int, default=10, metavar='N',
    help='display every Nth sample (default: %(default)s)')
args = parser.parse_args(remaining)
if any(c < 1 for c in args.channels):
    parser.error('argument CHANNEL: must be >= 1')
mapping = [c - 1 for c in args.channels]  # Channel numbers start with 1
q = queue.Queue()


def audio_callback(indata, frames, time, status):
    """This is called (from a separate thread) for each audio block."""
    if status:
```

```
        print(status, file=sys.stderr)
    # Fancy indexing with mapping creates a (necessary!) copy:
    q.put(indata[::args.downsample, mapping])

# collect avarage amplitude of all samples
allAvgs = []

def update_plot(frame):
    """This is called by matplotlib for each plot update.

    Typically, audio callbacks happen more frequently than plot updates,
    therefore the queue tends to contain multiple blocks of audio data.

    """
    global plotdata
    while True:
        try:
            data = q.get_nowait()
        except queue.Empty:
            break
        shift = len(data)
        plotdata = np.roll(plotdata, -shift, axis=0)
        plotdata[-shift:, :] = data

        # add avarage of current amplitudes to array
        allAvgs.append(np.mean(data))

        # calculate new ambient amplitude
        ambient = np.mean(allAvgs)

        # check if current amplitude is greater than 0.5 bigger than ambient, and print warning message if so
        if np.max(data) > ambient + 0.5 or np.min(data) < ambient - 0.5:
            now = datetime.now()
            print('Loud sound detected: ' + str(round(np.max(data), 2)) + ' at ' + now.strftime("%H:%M:%S"))

    for column, line in enumerate(lines):
        line.set_ydata(plotdata[:, column])
    return lines

try:
    if args.samplerate is None:
        device_info = sd.query_devices(args.device, 'input')
        args.samplerate = device_info['default_samplerate']
```

```
    length = int(args.window * args.samplerate / (1000 * args.downsample)) * 50
    plotdata = np.zeros((length, len(args.channels)))

    fig, ax = plt.subplots()
    lines = ax.plot(plotdata)
    if len(args.channels) > 1:
        ax.legend([f'channel {c}' for c in args.channels],
                  loc='lower left', ncol=len(args.channels))
    ax.axis((0, len(plotdata), -1, 1))
    ax.set_xlabel('Time')
    ax.set_ylabel('Amplitude')
    #ax.set_yticks([0])
    ax.yaxis.grid(True)
    ax.tick_params(bottom=False, top=False, labelbottom=False,
                   right=False, left=False, labelleft=False)
    fig.tight_layout(pad=0)

    stream = sd.InputStream(
        device=args.device, channels=max(args.channels),
        samplerate=args.samplerate, callback=audio_callback)
    ani = FuncAnimation(fig, update_plot, interval=args.interval, blit=True)
    with stream:
        plt.show()
except Exception as e:
    parser.exit(type(e).__name__ + ': ' + str(e))
```

Lund University **Department of Automatic Control** **Box 118** **SE-221 00 Lund Sweden**	*Document name* MASTER'S THESIS *Date of issue* June 2023 *Document Number* TFRT-6201
Author(s) Eric Hammar Pontus Sjöstrand	*Supervisor* Linus Wannebro, Axis Communications AB, Sweden Magnus Sjöberg, Axis Communications AB, Sweden Björn Olofsson, Dept. of Automatic Control, Lund University, Sweden Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle

Development of Detachable Audio Accessory for Surveillance Cameras

Abstract

Surveillance cameras are used for video monitoring. To complement the video stream, different external accessories can be attached to a camera. The addition of external accessories brings certain limitations in terms of aesthetics and functionality. To overcome these limitations, an internal platform for connecting accessories would enable the creation of better and more future-proof products.

This Master's Thesis investigates the development of a detachable twoway audio accessory and an internal platform for a panoramic surveillance camera. A structured product development method was followed, which resulted in a functional prototype. The accessory consists of a speaker, an amplifier, and two microphones for audio output and input. The components are enclosed by 3D printed parts with a geometry adapted to the camera's internal platform. The mechanical fastening of the accessory is made in two steps utilizing magnets and screws, and the electrical connection between the accessory and the camera is made using pogo pins. The playback and recording are interacted with through a graphical user interface. Pros and cons of the resulting prototype are discussed, and possible future improvements are suggested.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title		*ISBN*
0280-5316		
Language English	*Number of pages* 1-89	*Recipient's notes*
Security classification		

http://www.control.lth.se/publications/