2014

Sound Level Controlling

Analog instrumentation

B312a:
Aleksandrs Levi
Andrei-Viorel Radulescu
Mike Castro Lundin
Rolan Abdulrazzak Ossi
Minik Nathanielsen Olsen

AALBORG UNIVERSITY ESBJERG

Title: Sound Level Controlling

Printing Year: 2014

Group ID: B312a

University: Aalborg University Esbjerg

Authors:

Aleksandrs Levi

Andrei-Viorel Radulescu

Mike Castro Lundin

Rolan Abdulrazzak Ossi

Minik Nathanielsen Olsen

Abstract

Music industry is very popular nowadays, which may cause several problems related to noise pollution. In this project our group aimed at creating a system that would allow diminishing the noise pollution level, namely the one caused by audio.

Both theoretical and practical ways were approached in order to solve the problem. We created a working prototype to represent the way our system should work. We also conducted a research in order to better understand the relevance of the problem we sought to solve.

The work concludes that implementation is definitely possible, despite relatively high prices of the components. We do recommend the implementation, with hope that the problem of noise pollution will be at least partly solved by doing so.

Preface

The project “Sound level controlling” has been made by 5 students of the Electronics and Computer Engineering program in the second semester, part of the Project 2 or P2. It has been developed in the time frame of approximately 4 months (from 01/02 until 20/05), with the supervision of professors Akbar Hussain and Torben Rosenørn. We would also like to express gratitude to the people and companies that answered some of our questions regarding our idea (namely Poul Erik Lykke, technical manager of Tobakken, and Martin Poulsen, owner of Lydkonsulenten).

The solution to the problem has been found in a variety of sources, such as scientific literature, the Internet, knowledge gained by courses and qualified people, as well as the survey among people of different age groups.

The report is targeted at music industry as well as at ordinary people concerned with sound pollution and wanting to solve this issue.

For this report the Vancouver citation style has been used, using numbers to organize references. All references are contained in the last section named “References”. The two programs developed over the course of this project are located in Appendices A and B, as well as the results of the survey in Appendix C.

Table of Contents

1. Introduction

Over the course of history, people were able to achieve progress and development in all areas of our life. One of the challenges we are facing is the spread of noise and attempts to reduce this phenomenon, which poses a real danger to our health.

Exposure to noise caused by the cars and factories make us require periods of rest at home. But, unfortunately, we cannot feel comfortable at homes, because of other sources of noise.

We define noise as unwanted sound, whether it is music or sounds from nearby sources, such as a construction site, or a concert venue.

Noise caused by music is a source of nuisances, if you are in close proximity of the source. Exposure to noise for a long period of time may lead to many health issues which we will list later, and the problem is that people listen to music louder than they should for the safety of their own health. So, we decided to create a product that would be able to solve these issues and help to reduce the music noise.

The purpose of the project is to control the audio output through a limiting system inside the sound device, in order to ensure compliance with the legal limits that some countries have, as described later in the report. The project is not only for private use, but can also be extended for public use such as concerts.

The system works by receiving sound into a group of microphones, and processing that input in order to limit the output. This way there will be no possibility of exceeding the limit.

The university has allocated a sum of money to buy some of the materials needed in our project. The materials for the project are a wave shield, two Arduino kits and a package of sensors (microphones).

As a result of Introduction to Electrical Engineering course and Digital Design & Sensors courses, we have got a good experience in working on hardware in this semester, in addition to our experience in the software in the first semester.

Our goal in this project is to introduce a new system that is able to control the sound, which may be a source of disturbance.

2. Problem Analysis

2.1 Decibels (dB):

dB is a unit to measure the level of sound intensity, and it is equal to one-tenth of a Bel. A Bel is used in physics to measure the difference in the intensity level of sounds to normal human ears, equal to ten decibels. [1]

It is estimated that 0 dB corresponds roughly to the quietest sound that can be heard by a healthy young adult. Normal conversation has a level of 60-70 dB, while sounds above about 100 dB tend to be uncomfortably loud and can cause damage our ears if heard for a long time. Sounds with a level above 120 dB can damage peoples' ears permanently, even when exposed for only a few minutes. [2]

A Bel (symbol B) is a unit of measure of ratios, such as power levels and voltage levels. It is mostly used in telecommunication, electronics and acoustics. Invented by engineers of the Bell Telephone Laboratory, it was originally called the transmission unit or TU, but was renamed in 1923 or 1924 in honor of the laboratory's founder and telecommunications pioneer Alexander Graham Bell. [3]

The decibel unit is often used in acoustics to quantify sound levels relative to 0 dB. The reference may be defined as a sound pressure level (SPL). The normal range of human hearing extends from about 0 dB to about 140 dB. [4]

0 dB is the threshold of hearing in healthy, undamaged human ears; 0 dB is not an absence of sound, and it is possible for people with exceptionally good hearing to hear sounds at -10 dB. [5]

A 3 dB increase in the level of continuous noise doubles the sound power, however experimentation has determined that the frequency response of a human ear results in a perceived doubling of loudness with every 10 dB increase; a 5 dB increase is a readily noticeable change, while a 3 dB increase is barely noticeable to most people. [6]

The decibel is used rather than arithmetic ratios or percentages, because when certain types of circuits, such as amplifiers and attenuators, are connected in series, expressions of power level in decibels may be arithmetically added and subtracted. It is also common in disciplines such as audio, in which the properties of the signal are best expressed in logarithms due to the response of the ear. [7]

In radio electronics, the decibel is used to describe the ratio between two measurements of electrical power. It can also be combined to create an absolute unit of electrical power. For example, it can be combined with "m" for "milliwatt" to produce the "dBm". [8]

2.1.1 Absolute measurements [9]

- Electric power: dBm or dBmW. Db =1 mW, in analog audio, power measurement relative to 1 milliwatt.
- Electric voltage: dBv. dB = 1 V, voltage amplitude of an audio signal in a wire, relative to 1 volt, not related to any impedance.
- Acoustics: dB (SPL), dB (Sound Pressure Level), relative to 20 micropascals, (μPa) = 2×10^{-5} Pa, the quietest sound a human can hear. This is roughly the sound of a mosquito flying 3 meters away. This is often abbreviated to just "dB", which gives some the erroneous notion that a dB is an absolute unit by itself.

2.2 Research on decibel levels in loud places

Our group had a visit to 4 concerts in order to measure the intensity of dB. The results of one of them were as follows:

Time	Position	Decibel	Context	General Notes
21:05	*Back of the room*	98	Slow Music	*Band*: 6 *Piece*: *Drums*, *Bass*, 2 *Electric guitars*, 1 *lead singer*, 1 *keyboard*
21:05	*Back of the room*	82	Peak Music	
21:25	*Back of the room*	Average: 98-101 Max/Min: 92-103	Fast Music	*Reading types*: *Slow or peak reading*, People wearing ear protection, overall satisfaction over sound level.
21:27	–	–		*Pause*
21:55		100.6	Fast Music Rock Supercharger	Software used: 10EaZy RT version
22:00	*Back of the room*	104	Fast Music Rock	
22:05	*Back of the room*	106	Fast Music Rock	Roof and 1/2 of side walls relatively sound proofed
22:15	*Back of the room*	$90 - 94$	Fast Music Rock	
22:25	*Back of the room*	103	Fast Music Rock	
22:35	*Back of the room*	$100 - 103$	Fast Music Rock	
22:45	*Back of the room*	$75 - 80$	Fast Music Rock	
22:55	*Back of the room*	$85 - 90$	Between Songs	

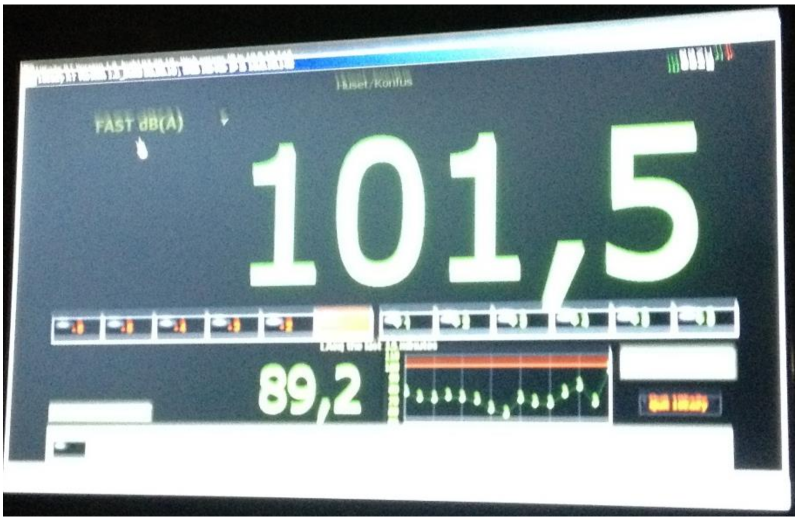


Figure 1: 10Eazy reading from Huset during the concert (05-04-2014)

From our readings we have learned the following:

- Loud sound is seen as an issue, as some people used earplugs to decrease the sound level. However, there were no complaints about loudness, as a high sound level is expected at events.
- Having accurate readers makes a huge difference, and the results can be recorded in several ways (slow or fast readings, peak or average values).
- Distance is a huge factor in the reading of how loud music will be. The distance to the source can reduce the sound levels greatly, even though the distance is not that big (~6dB for doubling distance meters which will be discussed later in thereport). Equally, a wall can be a very good sound insulator, or a very bad one, depending on many factors.
- As a general conclusion, we saw that measurements are highly dependent of the context, such as a different room size, different wall thickness as well as any sound insulation, can cause significantly different results.

From this we can conclude that our system must take this into account, providing a flexible and automatic or manual calibration function, in order for the system to be relevant to the context.

2.3 Management

Noise pollution is one of the reasons that cause harm to human health. Noise pollution problems are increasing day after day, especially in urban areas, densely populated areas, highways, airports, industrial zones and others. Noise is a type of vibratory issued air pollution, such as waves. [10]

2.3.1 Health

The effect of noise on human health is a source of concern, because of spread of the noise sources everywhere. Noise pollution can affect us in several ways, some of which are listed below:

- **Hearing Problems:** exposure to noise can damage the ear. Hearing impairment due to noise pollution can either be temporary or permanent. When the sound level is 70 dB, it becomes loud. Sound levels above 80 decibels produce damaging effects to the ear. When ear is exposed to extremely loud noise (above 100 decibels) for a long period of time, it can lead to irreparable damage and can cause permanent hearing loss. [11]

- **Sleep Disturbances:** exposure to high level noise can cause difficulty sleeping. Noise can interrupt a good night's sleep. People deprived of uninterrupted sleep show a sharp dip in their energy levels which often results into extreme fatigue. This can considerably decrease a person's ability to work efficiently. [12]
- **Poor Cognitive Function**: exposure to loud noise reduces the ability to read, learn and understand over time. Research has proven that children studying in noisy environments tend to show relatively low cognitive function. For instance, the cognitive status of children sent to schools that are in the close proximity of highways is worse in comparison to those learning in quieter surroundings. [13]
- **Heart disease**: Noise pollution can also lead to an increased chance of heart disease and high blood pressure, which, in some cases can lead to atherosclerotic that can cause death. Studies have shown that high intensity sound causes a dramatic rise in blood pressure as noise levels constrict the arteries, disrupting the blood flow. The heart rate (the number of heart beats per minute) also increases. This was evident in one study wherein the pulse of children staying in noisy surroundings was measured. It was found to be more than the heart rate of children living in less noisy environments. [14]

2.3.2 Environment

Human's noise pollution has made the Earth an uncomfortable place to stay for animals as well. Hearing loss and rapid increase in heart rate are some of the ill-effects of noise pollution on animals. High intensity sounds cause animals fear, forcing them to escape their habitat.

- **Animal's behavior:** when animals are expose to high levels of decibels, it is observed that animals are trembling. Being exposed to high level sounds reduces a cow's capacity of milk production.

- **Underwater animals:** whales and dolphins too experience discomfort due to noise pollution caused by submarines, shipping companies, and sonars.

- **Birds in urban areas:** Some kinds of birds that depend on hearing abilities in order to obtain food are facing a big problem because of noise. This is one of the reasons why certain species have become extinct. [15]

Recent studies on impact of low and high intensity sound on marine life showed that aquatic animals like cuttlefish and octopuses suffer serious damage from noise pollution. Oil-drilling and other activities like commercial shipping lead to the death of these marine organisms, and thus cause an imbalance in the ecological balance. [16]

2.3.3 Ethics

We cannot foresee any major ethical problems within the scope of our project. However, it could be considered an ethical problem for people who are unwilling to be limited by the bounds defined by the system.

2.3.4 Economy

The materials for the project are a wave shield, 2 packages of Arduino boards, and a group of sensors (microphones).

The current prices of Arduino boards are around 450 DKK [17]. The price is not expensive compared with the prices of other Arduino. So price of Arduino that used in our project has a normal price. This is very important for production factories, because the price of raw materials is important for profitable production.

Second part is Wave Shield. Wave Shield can play any uncompressed 22 050 Hz, 16 bit, mono Wave (.wav) files of any size. While it isn't CD quality, it is certainly good enough to play music, have spoken word, or audio effects.

- Output is mono, into L and R channels, standard 3.5mm headphone jack and a connection for a speaker that is switched on when the headphones are unplugged.
- Files are read off of FAT16 formatted SD/MMC card.

- Included library makes playing audio easy.

The price of Wave Shield is about 200 DKK [18]. Price can be a major problem in our project, because the addition of our prototype will cause a major increase in price of any stereo system.

The sensor used in our project is an electret microphone. There are three kinds of electret microphone, foil-type or diaphragm-type, back electret, and front electret.

We use a front electret microphone in the project, because it is the newer type of electret microphone and electret types require no polarizing voltage. The price of Front electret microphone is a little bit high compare to other kind. Front electret microphone cost 50-55 DKK. But if we look to other kinds, the price is 10-27 DKK. [19]

The problem facing us in the project is the cost of individual parts, which are a little expensive. When we look at the material used in the project, we can note that the cheapest part is at least 50 DKK. In general, companies have strict regulations regarding production, because they are always looking for profits. [20]

This is all taken into consideration when making such a project in order to apply it in practice.

2.4 Legislation

Many cities prohibit sound above threshold intensity from transmitting over property at night, typically between 10 p.m. and 6 a.m., and agree to an increased limit during the day.

Many municipalities do not follow up on complaints, even in instances where a municipality has a special office for complaints. But there are some cities that have initiated strict laws to limit high sounds level, such as Portland, and Oregon City in United States, which has instituted fines reaching as high at $5000 per infraction.

A lot of people demanded since 1960s, that citizens might be entitled to be protected from high sound level exposure. The legal circumstances in the United States changed rapidly with passage of the National Environmental Policy Act (NEPA) in 1969 and the Noise Pollution and Abatement Act, more commonly called the Noise Control Act (NCA).

Several European countries emulated the U.S. national noise control law: Netherlands (1979), France (1985), Spain (1993), and Denmark (1994). [21]

German and Swiss Transport Ministers have signed a new law to limit noise pollution in capital Bern, because of noise pollutions affects the population of Zurich and southern Germany. Meanwhile, the European Parliament approved in February 2013 a draft law which demands to reduce the noise of cars in Europe for the protection of public health.

According to the project, which must be negotiated with the governments of the EU countries before final approval, the European Parliament will put the new limits of all fields that produce noise. That will be implemented gradually over six to eight years, after its entry into the application.

According to the European Parliament's statement,'' continuous exposure to high levels of noise can lead to health damage and disorder in the performance of the physiology of the human body''.

Experts say the noise that affects the human sleep is from 30 decibels, and more than that may cause anxiety, headaches, and sleep deprivation. The noise is one of the main reasons for the reduction in performance, concentration and distraction. This causes feeling unwell, and reduces workers' production and ability to work. [22]

One of the studies in 1962 indicates the importance of this problem in urban areas, especially when conducting comparisons with quiet areas. In the quiet of civilization, a 70 year old man, has good hearing capabilities compared to American people in their twenties.

In general it would be a rule for a workplace to provide ear protection when exposed to above 85 dB. In practice, workplaces implement the general rule that you are allowed to work 8 hours under 80 dB, 4 hours under 85 dB, 2 hours for 90 dB and half an hour for 110 dB.

2.5 Stakeholders

The stakeholders that are relevant for our project are the following:

1. Hi-fi system companies: The companies that produce and sell the equipment will be very important to guarantee compatibility and encourage the correct use of the system.
2. Clients: It is important for the client to wish to respect the rules and accept the fact that they will not be able to surpass the defined threshold.
3. Arduino board manufacturers: Since they will be providing the microcontroller, it is vital that they keep the price at a reasonable level, and still make profit enough to sustain production levels.

3 Definition of the Problem

3.1 Conclusion of Problem Analysis

From the problem analysis it can be seen that the system that we have designed definitely can be implemented on practice.

We have studied sound values in order to get necessary understanding of sound measuring techniques, required for our project. We defined dB and its relation to the sound level, as well as some absolute measurements. Furthermore, we made a research on health and environment to ensure that our system does not violate any existing rules.

We also added a table of values that we created using data gained within our research on decibel levels in loud places. It also helped us to get more understanding of the high dB levels: 80 and more. The research also showed that many people are concerned with possible hearing impairments because of the loudness of the concerts, since some of them were wearing ear protection over the course of the concert.

A lot of information was gained about the health issues connected to excessive noise. Namely, various conditions, such as heart diseases, different hearing problems, sleep disturbances and poor cognitive function, were discovered to be not rarely found among people often exposed to loud sound.

Environmental issues have also been inspected. It was discovered that sound pollution may be dangerous for a variety of animal species, sometimes even leading them to extinction.

We also made an investigation in the economical aspect of the project and found out that, although the price of implementation cannot be called low, it is still a viable option which may be considered by the sound hardware developers.

When we looked into legislations of different states we found out that depending on the country there can be different maximum allowed sound level values. However, mostly governments do not set a specific limit on how much loudness it is allowed to play to audio – norms are usually only present for the work conditions. People just have to respect their neighbours and decrease music volume when asked to.

It has also been understood that, although some people might find it offensive to reduce the volume on their audio, it is very important to make sure that people do not suffer from sound pollution caused by others, therefore our system is quite useful in that respect.

In the survey that we have conducted quite a large amount of people (namely 42.19% - see appendix C) have responded that they had at some time had problems sleeping because of the loud parties close by. Therefore, we conclude that the prototype that we have developed will definitely be of help to these people.

To sum it up, we found out that our product should be able to get into development process despite some difficulties, and that the idea as a whole has potential to become popular and useful. Even though it is unlikely that it will happen soon, it is nevertheless possible to find people who would gladly use such system as soon as it is available.

3.2 Description of Problem on a larger scale

The problem we are trying to solve is noise pollution caused by people listening to loud music. We are sure that this problem is getting more and more relevant over the years, since the amount of cities and their populations grow – and there is not much that is done to stop it. We believe that our solution is a good way to make people's lives better by decreasing the pollution – especially in the cities.

First of all, the technology that we are developing would be able to decrease sound pollution caused by music significantly. It, in turn, would make much less people suffer from diseases and conditions related to exposure to loud sound. Thus, people would understand how much the pollution affected their lives and become more eager to get rid of it, causing a chain reaction.

Secondly, sound engineers all over the world would not have to worry about maintaining safe music level at the concerts. The system is able to automatically limit the decibel level, depending on the calibrations. That means that it is not only better for the people, but also for the staff at different events that have loud music.

Eventually, we hope that the system will be implemented in sound systems at concerts and other loud events. It would ensure that the sound level norms are being respected, which would benefit to ordinary people as well as to the loud music lovers – they would not have to use ear protection at these events anymore, as the risk of hearing impairment decreases.

3.3 Delimitations of Problem

What we were planning to create within this project was a sound sensor system working with different sound hardware that would be able to perform the following actions:

- Determine the current sound level in decibels.
- Limit the maximum sound level when it reaches specific values set by the user.
- Change the maximum decibel level available depending on the time of a day.

Nevertheless, we do not necessarily need our system to be able to show the sound level to the user, since the main point of it is to limit the sound level when it reaches certain values. We concentrated our work on creating both hardware and software for our system, having chosen Arduino boards as our main platforms. It was chosen due to the fact that it is considered rather convenient and is well accepted by many developers all over the world, including not only students, but also qualified specialists. [23]

It is also worth mentioning that we are not using the official Arduino boards because of their relatively high price. However, the boards that we are using are fully compatible with almost all Arduino software and hardware, and we have not experienced any major hiccups.

3.4 Problem Definition

What is the problem?

The main problem we were trying to find a solution to is noise pollution originated from music sources. There are many sources that cause that, but we were aiming at the people who tend to listen to music too loud and to loud music in general – in such places as night clubs or open-air concerts (especially the ones close to residential areas).

Where is it a problem?

The noise pollution problem is relevant across the globe – people often do not seem to think about whether or not their sound level preferences apply to their neighbours as well. That said, it is important to note that mostly the problem is relevant in urban areas, with large amount of places where loud music is a norm – such as the aforementioned night clubs and concerts.

Why should we solve this problem?

The reason why tackling this problem is important is that there is evidence of how noise pollution affects people lives in a negative way. These include, but are not limited to, stress related illnesses, high blood pressure, speech interference, hearing loss, sleep disruption, and decreased productivity. [24] Needless to say, none of these consequences of being exposed to noise pollution are harmless and, therefore, the problem definitely needs to be dealt with.

When does the problem occur?

Although the noise pollution problem is happening 24 hours a day, it arguably does most harm in the night hours. People’s inability to sleep because of noise might make them stressed and anxious, thus causing other problems in their lives. Nevertheless, there are some sound level norms that should be abided regardless the time of the day.

Who is the problem?

The people who tend to listen to loud music definitely pose a threat to those who are not used to it but are exposed to it due to living nearby. They might simply not know that their music is too loud, but they are still an important part of the problem.

How is it a problem?

By having a negative impact on people's lives noise pollution is causing multiple illnesses, hence being dangerous to their health. This reason alone is enough to try to find a solution to at least some sources of this problem, thus making people's lives better.

3.5 Solution Definition

In this section we will try to do the solution analysis using the concept similar to the one used in problem definition.

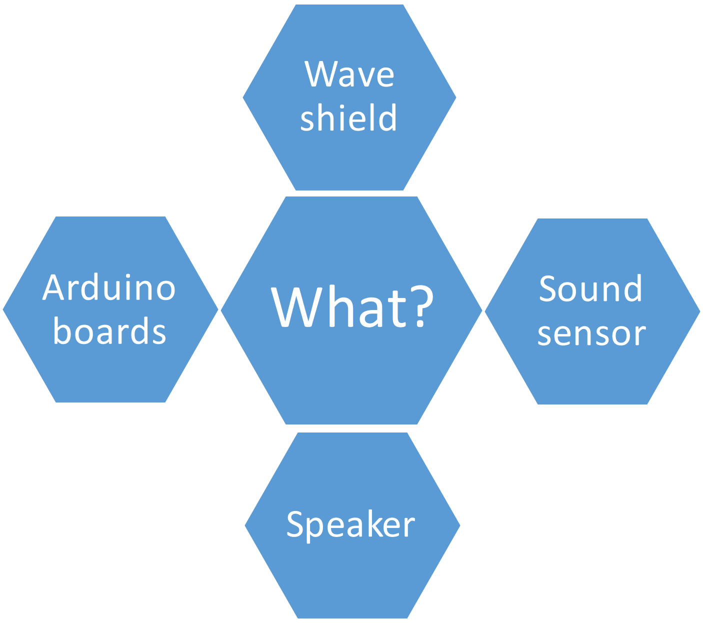
What – What is the way to get the solution?
Why – Why is the solution required?
Where – Where did we get the solution?
Who – Who is going to use our solution?
When – When can our solution be used?
How – How did we find the way to implement solution?

3.5.1 What

In this case we decided on what technology we were going to use at the very beginning of the project. We knew that in order to create the sound limiting system we needed a microcontroller, as well as the software to make it work.

We applied for the funds in order to get the Arduino board, which seemed to be an efficient tool to create our system. It was selected because it is widely used by many beginning developers and is considered to be a satisfactory system overall. [25]

We also ordered a wave shield, which we use to play .wav audio files with the Arduino. Aside from that we got a Grove sound sensor, used to get an analogue audio input (to subsequently limit it in case of reaching a certain value, set by the user). And also one of our group members brought a speaker in order to play the music file.



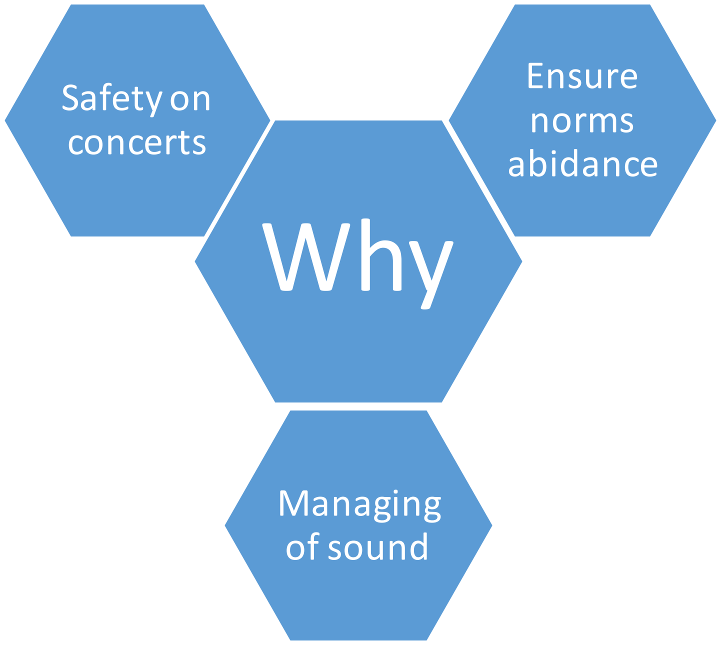
3.5.2 Why

The solution that we have come up with can be utilized in a variety of ways, by both sound device manufacturers and their customers.

First of all, it can be used as a sound controlling device on different events that amplify audio, such as concerts and discos. It is highly recommended to ensure safety of the audience. The sound level at these events can reach 100 dB and more, which may cause hearing impairment if one is exposed to that sound for an extensive amount of time. The technical staff at concerts keeps track of the overall dB level, but with our technology the limitation of the sound can be done automatically.

It can also be used by regular people in their apartments, in order to ensure that they do not disturb other people. It could help to abide the norms in the night hours, for example.

The system, technically, can also be used as a sound managing system, meaning that one can maintain not only low sound level, but high level as well. A lot of possible applications may be discovered by the users.



3.5.3 Where

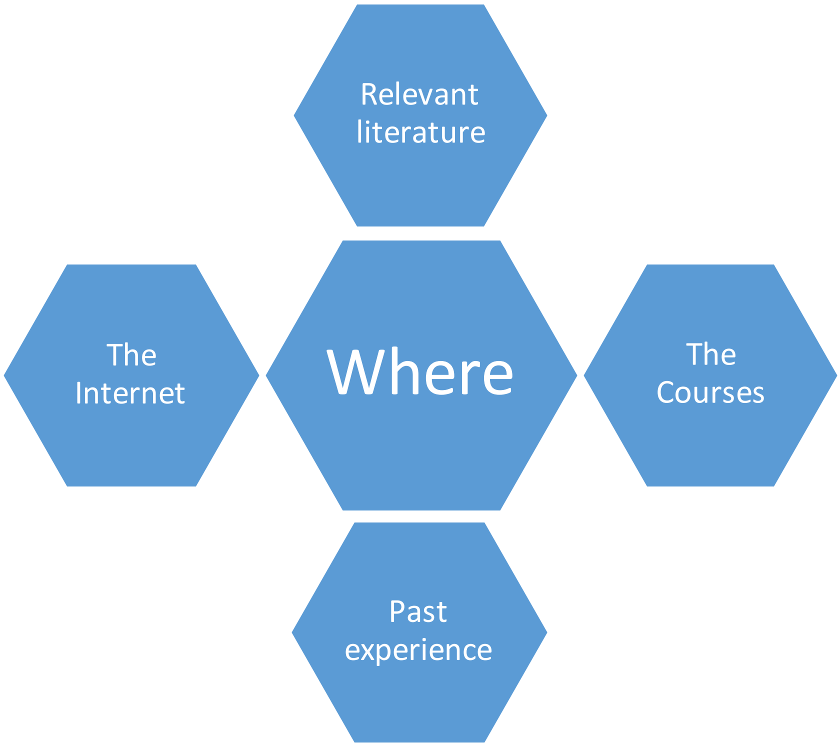
For finding information we needed for creation of the system we utilized multiple reliable sources, as well as our own knowledge.

We used a few useful library books to gain more knowledge about sound theory and Arduino boards and C programming.

We also searched the Internet for some examples in order to better understand the basics of Arduino programming. The official forums appeared to be of great help, especially when we encountered initial problems while working with the board.

The courses that we had during this semester were very useful to us as well. We utilized 3dCAD course to make a 3d model of a case for our prototype. We used Calculus in some formulas used in sound theory. Digital design was used when designing the logic of our system. Electrical engineering was helpful in modeling the circuit itself. We also used some of the programming skills gained from last semester's Introduction to programming course.

Among other things, some of our group members were already familiar with soldering techniques, which appeared to be necessary to assemble the wave shield.

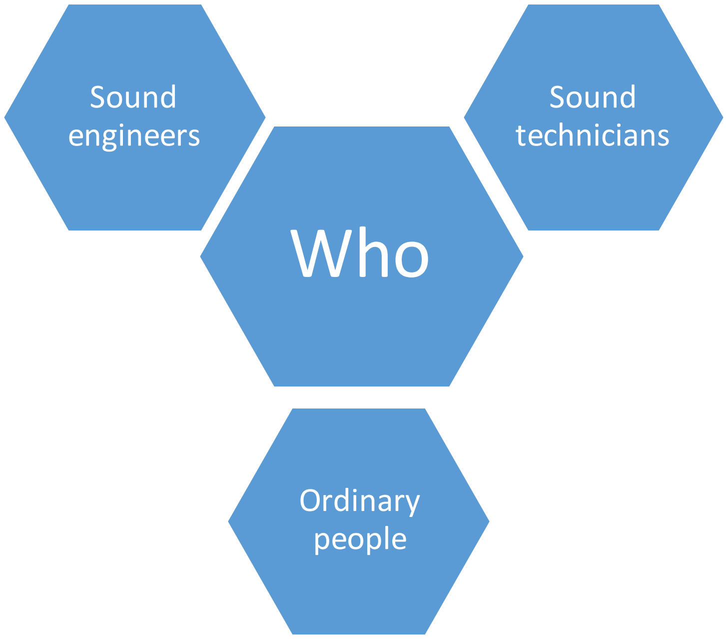


3.5.4 Who

The users that we aim most at are sound engineers, as well as other technicians working at events, such as concerts or big shows.

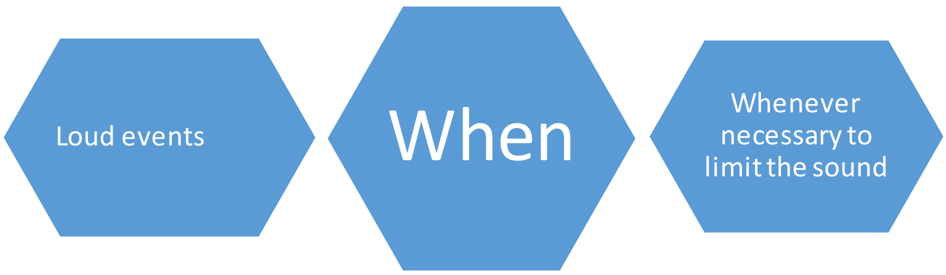
Of course, it could also be used in many other places, such as, for instance, libraries or hospitals, in order to manipulate the sound level.

We also suggested that the system could be used by the ordinary people to control the sound level within their household. It would be helpful to make sure that they are not too loud when playing audio at night hours, for example.



3.5.5 When

The system should be used during any type of events that amplify music or speech, especially the ones that are happening in the night hours. It should also be used whenever a person deems it necessary to limit the sound output, whether in his own house or apartment or in some public place where only quiet music is acceptable, such as libraries and hospitals. [26]



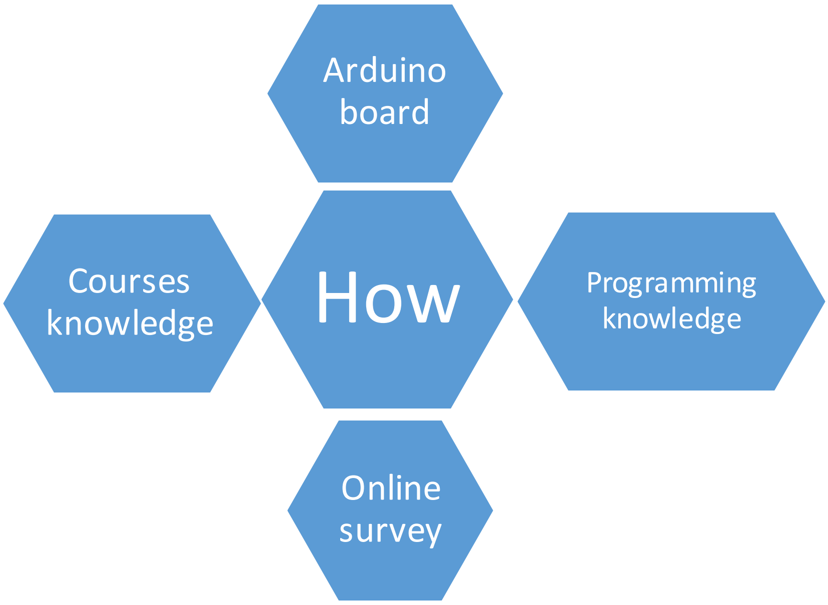
3.5.6 How

In order to implement our idea in practice we decided to use Arduino platform. As it has been stated before, it was selected because of being widely accepted by both the new and experienced developers all over the world. However, neither of us had any prior experience with the platform before. Therefore, we read useful books from the library and started researching. We also looked in the Internet, where we found a lot of helpful material that we used throughout the project.

Besides that, we used the knowledge gained from Introduction to Electrical Engineering and Digital Design courses. It appeared to be of much help when we started working with the hardware. The laboratory exercises that we had gave us some useful experience in terms of working with circuits.

The skills that we gained last semester thanks to Introduction to Programming course also helped a lot when writing the software for our system.

We also conducted an online survey, in order to get opinions of different people regarding the problem we were trying to solve. It contained multiple questions regarding how loud people tend to listen to music and if they ever encountered some troubles related to someone else doing so. The survey can be found in the appendix C at the end of the report.



3.6 Success conditions

At the beginning of our project we decided on a set of goals we wanted our system to be able to perform in order to solve our problem. Over the course of the project we came up with more goals, and those that were considered of most importance and relevance were also included in the success conditions for the system.

Firstly, we wanted our system to be able to detect the current sound level. It is simply required for almost all other functions of the system. However, the ability to display this value is not required, although appreciated.

Secondly, it is important that the system is able to limit the sound output when it reaches a specific value. That is basically the main function of the system, required for it to be considered successful.

Of course, it goes without saying that it should also be customizable by users. People might want to set different sound level limits, so it should be definitely available to them as an option.

The system should also be economically viable for the sound hardware manufacturers. Otherwise, the likelihood of it becoming popular would not be very high.

Lastly, we want the system to be universally embeddable in as many types of existing sound systems as possible. It would definitely be a key to making it popular in all world regions.

3.7 Relevance of the problem

First of all, it is important to understand that not all people might find the problem we are solving relevant. Some do not have any problems with loud music because they do not live close to noisy neighbors, or for some other reason. Others may enjoy what other would consider loud music, thus not being able to understand the problem.

However, the noise pollution problem itself is not something that people can deny. Music does not play the last role in that phenomenon as well. Meaning that, although some people might not be concerned about the problem, we need to approach it somehow, simply because otherwise it will become more complex to solve, because of less and less people understand the importance of it.

The health issues described in the Problem Analysis part of the report also show that the problem is serious and needs to be solved.

To sum it up, from all the information above we can definitely see that the problem is relevant and requires a solution.

3.8 Existing solutions

The most common solution employed in Denmark is 10EaZy. Venues like Train, Paletten, Tobakken, VEGA, AmagerBio etc. and festivals such as Roskilde Festival, Start! Festival etc. utilize the system at their events. [27]

We conducted our tests locally in Esbjerg. Our tests were conducted at Tobakken (Tobakken.dk), Huset Esbjerg (husetesbjerg.dk) and Konfus (konfus6700.dk). As Huset and Konfus are part of Tobakken's administrative system, the same measurement system mentioned earlier is applied: 10EaZy. Tobakken has its own system; Huset and Konfus share another.

Our data was collected at various concerts in a wide range of musical styles: Loud rock music, blues music and quiet singer-songwriter types of music, to obtain varied results in order to have an idea of the typical decibel output for varied types of music. Measurements were done in different room sizes as well: From 1350 capacity and down to 150. The 3 concerts were measured using data from 10EaZy whose measurement microphone was placed at what is known as Front of House (FOH), which is the mixing desk position in the house.

We also conducted interviews with sound professionals to learn how they managed to abide the rules during the events. Most sound engineers we talked to use 10EaZy as a reference from which they control their mix. They monitor the levels throughout the event in order not to exceed the maximum level. The limit varies from place to place. Maximum is 103 dB, but some venues prefer 100 dB or even down to 97 dB in some cases. The average level is measured over a 15 minute period and 10EaZy calculates the current average level (dBA) in real time based on the past 15 minutes. As most systems does not have a limiter implemented in the system, the sound engineer in charge is responsible to not exceed the level. It *is* possible to implement 10EaZy as a limiter that blocks the mix signal to the amplifiers when the level is exceeded, but it is not a common practice. Furthermore in our interviews, the sound professionals explained that they have not experienced 10EaZy being used as a limiter, but they talked about another system that functioned as a limiter between the mixing desk and the amplifiers: SoundEar and The Guvnor SPL2 Limiter [28]. While the system is useful for work places with Health and Safety regulations like hospitals factories etc., it does not suit well for concert venues, clubs, bars and likewise public areas. No further viable explanation was given except that, the system don't take the momentary peaks into its calculations, so the sound cuts off each time i.e. the drummer hit a cymbal or the lead singer hit a particular loud peak. With that in mind it makes sense that 10EaZy is a suitable system.

In short, solutions vary depending on how venues handle the issue. And even though our data confirms that the rule is abided, there is the chance of human error which we try to eliminate by introducing our solution.

4 Solution

4.1 Sound theory

4.1.1 Basic sound theory

So what *is* sound exactly? Sound is elastic vibration through air, liquids and solid materials; sound is transmitted in sine waves [29]. Its most basic form as a function of time (t) is:

$$y(t) = A\sin(2\,\pi f t + \phi) = A\sin(\omega t + \varphi)$$

where:

- A, the *amplitude*, is the peak deviation of the function from zero.
- f, the *ordinary frequency*, is the *number* of oscillations (cycles) that occur each second of time.
- $\omega = 2\pi f$, the *angular frequency*, is the rate of change of the function argument in units of radians per second
- φ, the *phase*, specifies (in radians) where in its cycle the oscillation is at $t = 0$.
    - When φ is non-zero, the entire waveform appears to be shifted in time by the amount φ/ω seconds. A negative value represents a delay, and a positive value represents an advance.

Sound waves cannot travel through vacuum unlike electromagnetic waves such as radio waves and light. When we are talking about wavelengths of sound we actually mean that it is the distance a sound travels during one cycle of vibration or wavelength. In principle, sound vibrations are alternating current signals, which is why we handle sound work as an alternating current-technique.

The *frequency* of sound is the number of vibrations per second and its measuring unit is hertz (Hz). As stated earlier the human hearing goes from 20 Hz to about 18-20 kilohertz (kHz). Thus the frequency-range between 20 Hz-20 kHz is the so called audio-range for a healthy human being. Please note that one's individual hearing depends on one's eardrum, intensity of sound, type of sound and tone, distance between sound source and the listener and even climatic conditions. Even the sensitivity of human ear is not constant: Sensitivity is lower between 20 Hz and up to 3-400 Hz depending on the individual's state of hearing, while the sensitivity is higher in the area of 500 Hz to 5 kHz, which is the average frequency range of human speech. Infrasound is below 20 Hz and ultrasound-areas are from 20 kHz and above.

The velocity of sound is the speed in which sound is transmitted through the relevant medium be it air, liquid or solid materials. The speed varies depending on the medium, pressure and temperature. Sound travels at approximately 340 meters per second when moving through air in a temperature of 20 °C. [30]

The wavelength of sound it the length of one cycle of vibration. Its widely used unit is lambda (λ). The lower the frequency, the bigger the wave length is. The higher the frequency, the shorter the wave length is.

The speed of sound c, wavelength λ and the frequency f can be calculated:

$$c = \lambda * f \ or \ \lambda = \frac{c}{f} \ or \ f = \frac{c}{\lambda}$$

For example:

$$20\ Hz: \frac{340\ meters/sec}{20} = 17\ meters$$

$$18\ kHz: \frac{340\ meter/s}{18.000} = 1{,}9\ cm$$

Name of the medium	Velocity of sound at 20°C in m/s
Atmospheric air	343
Hydrogen	1305
Nitrogen	338
Pure water	1450
Brick	4300
Concrete	4000
Granite	6400
Glass	5000-6000
Aluminum	5100
Iron	4700-5100
Copper	3900
Brass	3500
Silver	2600
Cork	450-530
Rubber	40-150

Fig. 1: Velocity of sound [31]

Transmission through air causes small variations in pressure. Sound pressure is proportional to air pressure. When pressure increases, sound is perceived louder to human ears and vice versa. Its measure unit is Pascal (Pa).

The decibel is a tool to express “how much” sound is perceived by human ears. We would like to introduce in short how the decibel works for general audio work as we need a little understanding in basic sound theory to comprehend how we have created our solution and prototype. Most physical variables are considered in linear terms: “Twice as much” of a given quantity produces twice the end result: If you order a cup of coffee twice in a café, you get two cups of coffee. This linear relationship does not hold true for the human hearing. Following that logic, twice the amplifier power will not produce sound twice as loud. Subjective testing has shown that power applied to a loudspeaker must be increased by 26% to be audible: A ratio of 1.26:1 produces the minimum change in audio, regardless of initial power quantity. If initial quantity is 100 Watt, then 126 Watt will produce an audible change. A scale that is calibrated proportionally is called a logarithmic scale, logarithmic meaning "proportional numbers". For the sake of simplicity, base 10 logarithms are used when working with audio. Changes in level are determined by finding the ratio of change in the parameter of interest and taking the base 10 logarithm. The resultant number is the level change between two quantities is expressed in Bels. The common logarithm is determined using a look-up table or scientific calculator. That log conversion helps us understand:

1. It puts the ratio on a proportional number scale that correlates better with human hearing than a linear relationship would.
2. It allows very large numbers to be expressed in a more compact form.

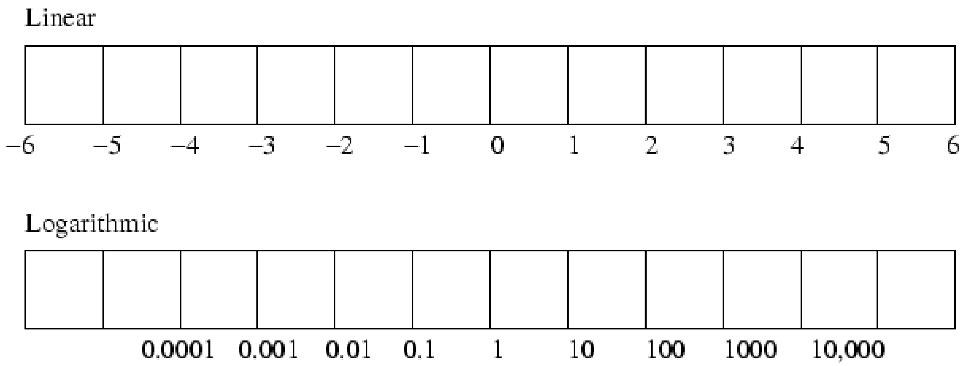


Figure: Comparison between linear and logarithmic scales

The last step of the decibel conversion is scaling the Bel quantity by a factor of 10 thus converting "Bels" to "decibels". The decibel is always a power ratio just like electrical and acoustical changes can be converted in the aforementioned method.

Remember, a decibel conversion requires two quantities that are in the same unit: Watts, volts, meters, feet etc. The unit cancels when dividing, leaving only the ratio between the two quantities. Meaning: The decibel is dimensionless and cannot be called a proper unit. If two arbitrary quantities are compared, the result is a relative level change. If a standard reference quantity is used, the result will be an absolute level. Relative levels are mostly applied to live concerts. Absolute levels are very useful for equipment specifications and calibration. [32]

Sound pressure level (SPL) or sound level L_p is a logarithmic measure of the effective sound pressure of a sound relative to a reference value. It is measured in decibels (dB) above a standard reference level as mentioned earlier. The equation is:

$$L_p = 10log_{10}\left(\frac{(p_{rms})^2}{(p_{ref})^2}\right) = 20log_{10}\left(\frac{p_{rms}}{p_{ref}}\right)\ dB$$

where p_{ref} is the reference sound pressure and p_{rms} is the rms sound pressure being measured. As the frequency response of human hearing changes with amplitude, three weightings have been established for measuring sound pressure: A, B and C. A-weighting applies to sound pressures levels up to 55 dB, B-weighting applies to sound pressures levels between 55 and 85 dB, and C-weighting is for measuring sound pressure levels above 85 dB.

4.1.2 Combining and Averaging Sound Levels

Decibels cannot be added arithmetically. For example, if two noise sources are each producing 90 dB right next to each other, the combined noise sound level will be 93 dB, as opposed to 180 dB. The following equation should be used to calculate the sum of sound pressure levels, sound intensity levels, or sound power levels:

$$Total\ L = 10 \times log10(\sum 10^{LN/10})$$

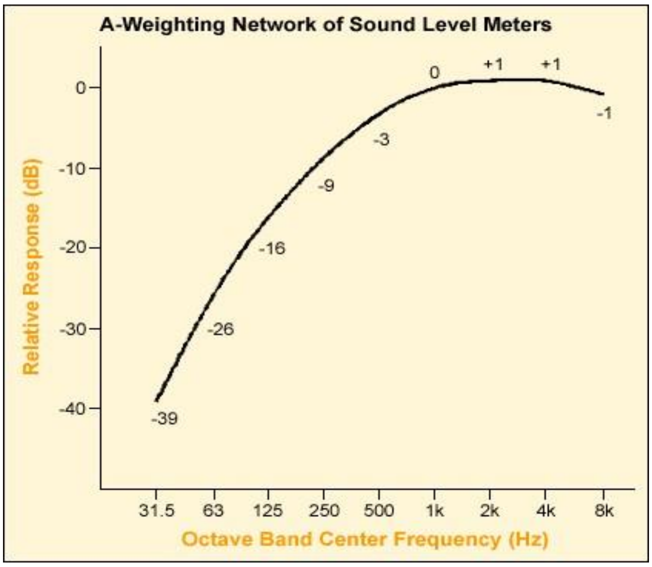
Often, using this equation to quickly sum sound levels when there is no calculator or computer available is difficult. The following table can be used to estimate a sum of various sound levels:

Difference Between Two Levels to Be Added	**Amount to Add to Higher Level to Find the Sum**
0–1 dB	3 dB
2–4 dB	2 dB
5–9 dB	1 dB
10 dB	0 dB

Example: There are three noise sources immediately adjacent to one another, each producing a sound level of 95 dB. The combined sound level can be found using the table above. The difference between the first two noise sources is 0 dB, which means the sum will be 95 + 3 = 98 dB. The difference between 98 dB and the remaining noise source (95 dB) is 3, which means the sum will be 98 + 2 = 100 dB.

4.1.3 Calculating the Equivalent A-Weighted Sound Level (L_A)

Occasionally, it is necessary to convert a set of octave band sound pressure levels into an equivalent A-weighted sound level. This is easily done by applying the A-scale correction factors for the nine standard octave centre frequencies and combining the corrected values by decibel addition. The A-scale correction factors are the values of the A-weighting network at the centre of each particular octave band. The value derived by combining the corrected values for each octave band is designated the A-weighted sound level (dBA).



Octave Band Center Frequency (Hz)	Example L_p (dB)	A-Scale Correction Factor (dB) *	Corrected Values (dB)**
31.5	94	-39	55
63	95	-26	69
125	92	-16	76
250	95	-9	86
500	97	-3	94
1,000	97	0	97
2,000	102	+1	103
4,000	97	+1	98
8,000	92	-1	91

* Look up on A-weighted network chart for each value L_p.
** L_p corrected to the A-scale = Li.

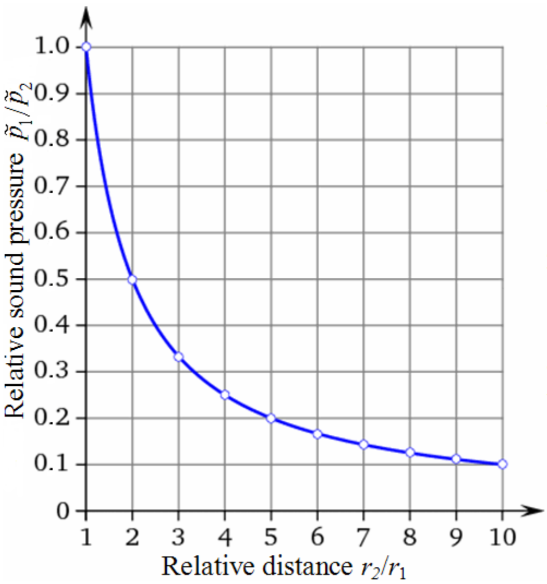
The A-weighted sound level is calculated by combining the corrected band levels:

$$L_A = 10 * log\left(\sum 10^{\frac{Li}{10}}\right)$$

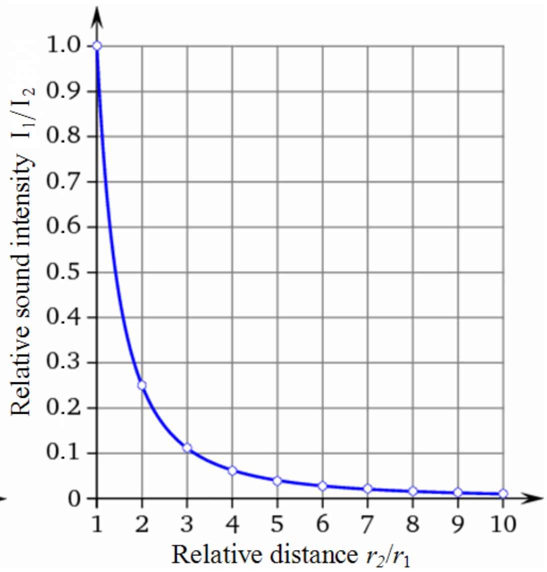
Where L_A is the A-weighted sound level and Li is the corrected decibel level value for each individual octave band.

4.1.4 Calculating Sound Pressure Level at a Distance

If a sound is generated at a point source in an open field, meaning there are no walls or other obstructions, the sound pressure level, L_p, will be reduced by 6 dB each time the distance from the noise source is doubled. Alternatively, L_p will increase by 6 dB in a free field each time the distance to the noise source is halved. Consider the following example:



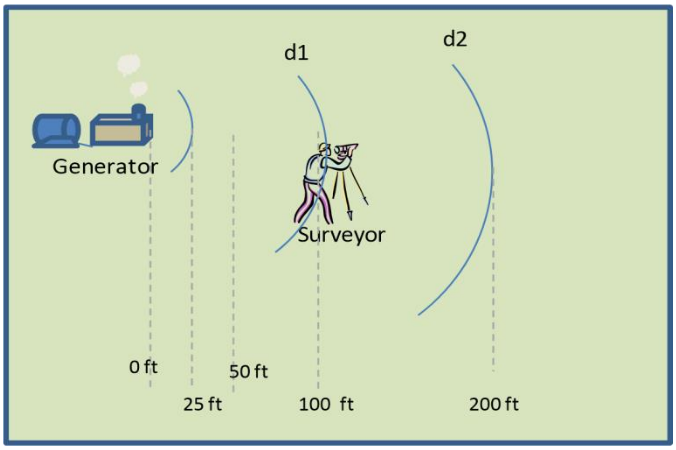
Inverse square law J Tilde 1/r² for acoustic intensity



Inverse distance law p Tilde 1/r for sound pressure

Example: A worker is surveying an open field, which has a diesel generator running in the middle of it.

The worker is 100 m from the generator and is exposed to a noise level of 85 dBA. When the worker is 25 m from the generator, the noise level will be 97 dBA. At 200 m from the generator the worker will be exposed to a noise level of 79 dBA.



Calculating the sound pressure level at a specific distance from a noise source is often useful. The following equation allows one to calculate the sound pressure level at any distance from a noise source in a free field:

$$L_{pd2} = L_{pd1} + 20 \times log(d_1/d_2)$$

Where L_{pd2} is the sound pressure level at the new distance from the noise source, L_{pd1} is the sound pressure level at the original distance, d_1 is the original distance, and d_2 is the new distance.

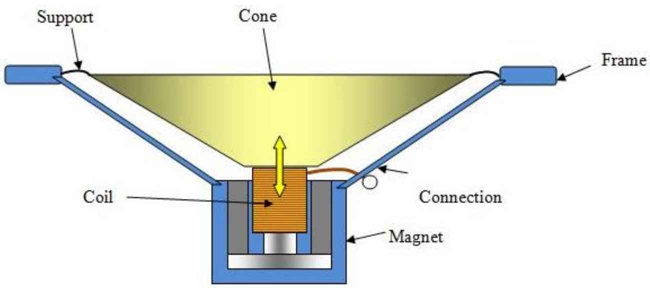
Example: The sound pressure level of an aircraft engine in the middle of an open runway is 120 dBA at a distance of 50 m from the receiver. The sound pressure level at a distance of 80 m is calculated using the equation above. L_{pd1} is 120 dBA, *d1* is 50 m, and *d2* is 80 m. Therefore, L_{pd2} is 120 + 20 × log(50/80), which is 116 dBA.

4.1.5 How do speakers work?

A loudspeaker (or electro-acoustical transducer) is a device that converts electrical energy into acoustical energy [33]. They are present in our daily lives from the alarm clock in the morning to the news on the radio, television. They are also used to reinforce audio at concert events and sounds at public addresses to playback music at cafés, etc. Most common purposes of loudspeakers can be:

1. Communication
2. Sound reinforcement
3. Sound production
4. Sound reproduction

Generally, the components of a loudspeaker are: Transducer, enclosure and crossover. A loudspeaker driver element (driver for simplicity's sake) is an electro-mechano-acoustic transducer. Electrodynamic (dynamic) drivers are the familiar fixed-magnet, moving coil that incorporate cones and domes as their radiating elements. Commonly called woofers reproduce low- to mid-range frequencies: 20 Hz-6 kHz, while tweeters reproduce 1-20 kHz and they usually employ dome-shaped diaphragms. An enclosure is the cabinet in which a driver is mounted. A crossover is the electronic dividing network that filters and distributes sound to the system's drive elements. A loudspeaker system is the sum total of its drive element(s), radiation aid(s) and crossover. [34]The coil is centred with the aid of an electric handle that allows the coil to move one direction only like shown in the following figure.



The most popular speaker used today is the dynamic speaker. The dynamic speaker operates on the same basic principle as a dynamic microphone. When an alternating current (i.e., electrical audio signal input) is applied through the voice coil that surrounds a magnet (or that is surrounded by a permanent magnet), the coil is forced back and forth as described by Faraday's law of induction [35], which causes the paper cone attached to the coil to respond with a rapid back-and-forth motion that creates sound waves. The coil and the driver's magnetic system interact, generating a mechanical force that causes the coil (and thus, the attached cone) to move back and forth, thereby reproducing sound under the control of the applied electrical signal coming from the amplifier.

The diaphragm is usually manufactured with a cone- or dome-shaped profile. Most common are paper, plastic and metal. The ideal material should:

1. Be rigid to prevent uncontrolled cone motions.
2. Have low mass to minimize starting force requirements and energy storage issues.
3. Be well damped to reduce vibrations that continue to reverberate after the signal has stopped with little or no audible ringing due to its resonance frequency as determined by its usage.

In practice, all three of these criteria cannot be met simultaneously using existing materials; thus, driver design involves trade-offs. I.e., paper is light and typically well damped, but is not stiff; metal may be stiff and light, but it usually has poor damping; plastic can be light, but typically, the stiffer it is made, the poorer the damping. As a result, many cones are made of some sort of composite material. For example, a cone might be made of cellulose paper, into which other materials as some carbon fibre, Kevlar, glass, hemp or bamboo fibres have been added; or it might use a honeycomb sandwich construction; or a coating might be applied to it so as to provide additional stiffening or damping.

The production of a high-fidelity (HiFi) speaker requires the speaker to be enclosed in a closed box to improve sound quality. Most modern speaker enclosures typically carry several loudspeakers with a crossover network that manage the incoming signal.

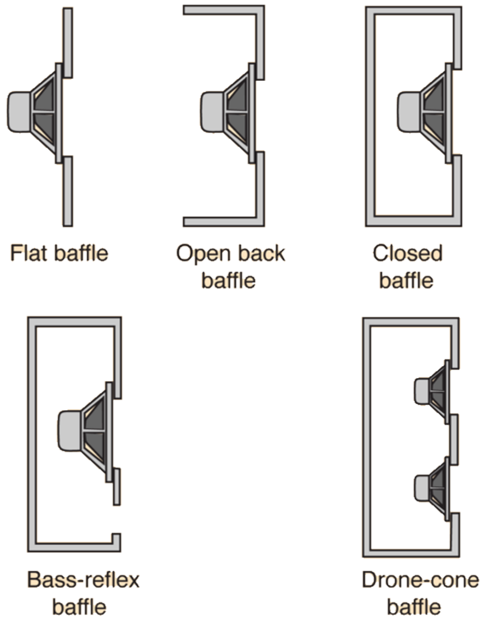


Fig. 3: Different types of enclosures [36]

4.1.6 Measurement of sound

When measuring sound pressure levels, sound level meters (SLM) are generally used. They are designed to measure sound over a range of frequencies and levels compared to human hearing. As stated earlier, human hearing corresponds to 20 - 20 000 Hz and pressure changes (sound levels) in excess of $1 - 1x10^7$. SLMs measure and display changes in acoustic pressure in a systematic and presentable fashion. Sound pressure is compressed logarithmically making us able to express $1 - 1x10^7$as 0-140dB. The display (be it a ballistic movement or digital display) does not display acoustic pressure changes instantaneously. However, it averages the changes and produces a readable number. Optional features can include peak level, maximum level, minimum level and sound exposure. They may even be able to log the data for statistical purposes and time histories.

Exponential-averaging meters measure sound pressure from a microphone (often built in in the device itself). The amplifier and weighting circuit limit the frequencies to a prescribed range. As stated earlier, microphones are transducers that convert changes in acoustic pressure to an electric signal. The microphone built in the SLM is highly likely to be the most fragile component of the device and its response is likely to change with frequency, direction, temperature and several other factors. [37] Three most commonly used microphone-types are: Piezoelectronic (ceramic), and two types of air condensers - Polarized and electret. Polarized air condensers require an external voltage (often in the range of 200 V). It must be well regulated as sensitivity is a function of the voltage. Electret (or permanently charged) microphones have permanent charge built into a plastic membrane, and response is similar to the polarized. We are using an electret microphone for our prototype.

Fig. 4: A typical digital sound level meter

4.2 Description of solution

As we have established that most venues do their best in order to protect the audience’s hearing, we still believe in implementing our solution as a “just in case”-scenario. We might also be able to introduce our solution to other public areas like night clubs and bars as they play loudly on weekend nights in order to generate more income. If the venues/clubs implement our system, we could make sure that the places protect the visitors’ hearing. The effect of the solution could be beneficial to the general public health as hearing loss (tinnitus for instance) can cause stress, depression etc. and even suicidal thoughts in some extreme cases. Ability to work decreases when stress and depression hits, which in turn could cause a negative effect in public economy/on a larger scale which can be prevented by implementing a relatively simple solution like ours.

In order to be able to modify the output volume of a sound system without blocking the volume/tone button on it we have decided to work in the preamplifier area of it as you will see in the following schematic.

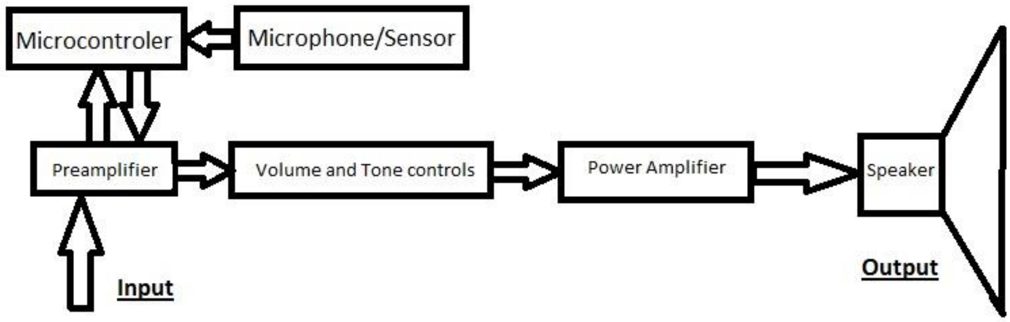
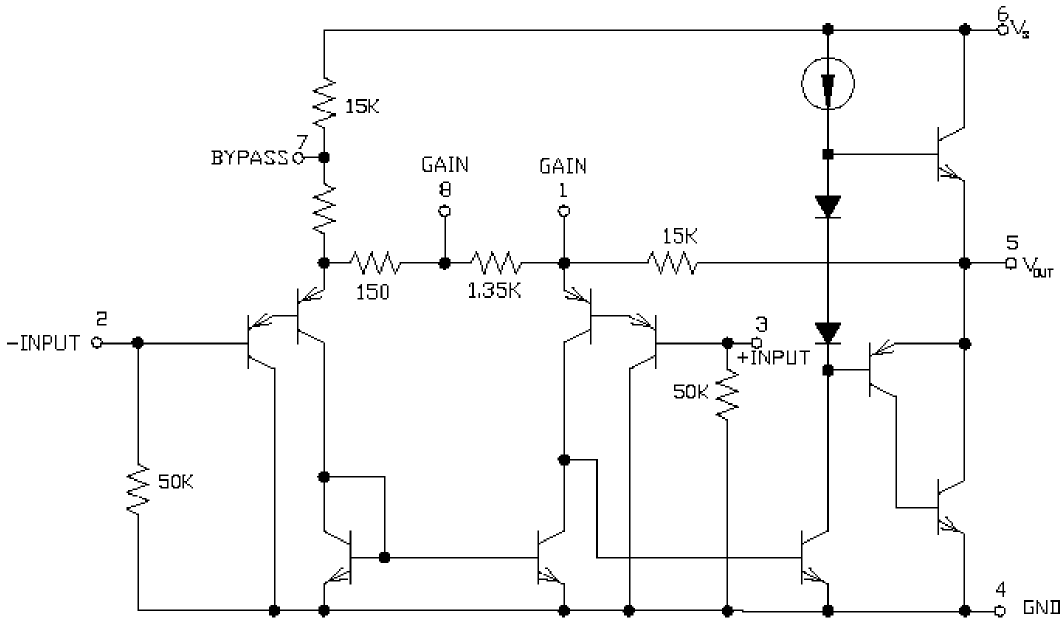


Fig.: Basic Sound system schematic with sound sensor

As we can see in the above schematic the direction of the signal from input to output is unidirectional, and can be best influenced in the preamplifier, by creating a gain control.

One of the problems that the gain control will face is that it can create an unwanted signal noise disturbing the sound quality at the output stage of the signal chain. This can be rectified with an additional filter to clear out the noise and calculating the maximum possible gain of the preamplifier. Controlling the preamplifiers gain is done by simply bypassing the input resistance of the OP-AMP from the preamplifier.

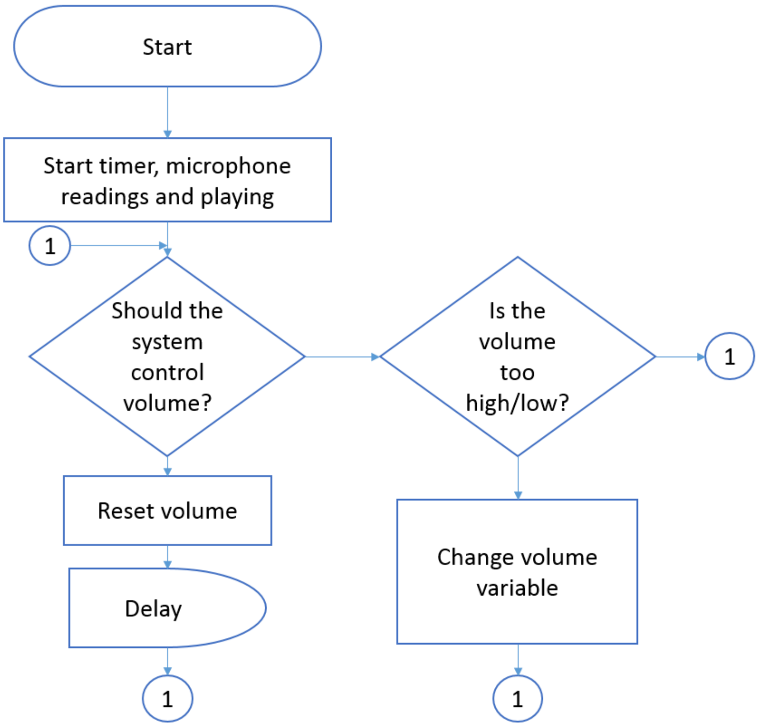


LM386 internal circuit

Usually the OP-AMP's can offer a certain gain by simply bypassing the input impedance. As can be seen in the above schematic, if we connect a capacitor to the 2 gain pins of the amplifier, than the 1.35kΩ resistance will be bypassed and the total gain factor will grow from x20 to a maximum of x200. In order to have a gain factor value between x20 and x200 we will have to put another resistor in series with the capacitor.

In order to have a variable gain we can use a potentiometer or a digitally controlled potentiometer in order to control the resistance of the bypass.

Software-wise, our solution follows the following flow diagram:



4.3 Arduino boards

Arduino is a single-board microcontroller, which aims to make Electronic applications more accessible. In 2005, Arduino was created for students as an inexpensive and easy program using interactive objects, and it is used by students from many technical disciplines at university level. The price of Arduino now is from 50 to 150 DKK. Arduino boards can be purchased pre-assembled or as do-it-yourself kits. In 2011 there were produced more than 300 000 Arduino boards, and in 2013 more than 700 000 official boards were in users' hand.

The Hardware of an Arduino UNO consists of open source board designed around an 8-bit or a 32-bit microcontroller. Current models feature a USB interface, 6 analog input pins and 14 digital I/O pins which allow the user to attach various extension boards. [38]

The software of an Arduino board is written in a language based on C / C++, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed for students to work on software development.

An Arduino consists of some main parts:

- Digital Pins
- Power LED
- Reset Button
- ICSP Header
- Microcontroller
- Analog Input Pins
- Power Pins
- Power Jack
- Voltage Regulator
- Power Selection Jumper
- USB Jack
- FIDI USB Chip
- RX + TX LEDs
- Pin 13 (L) LED

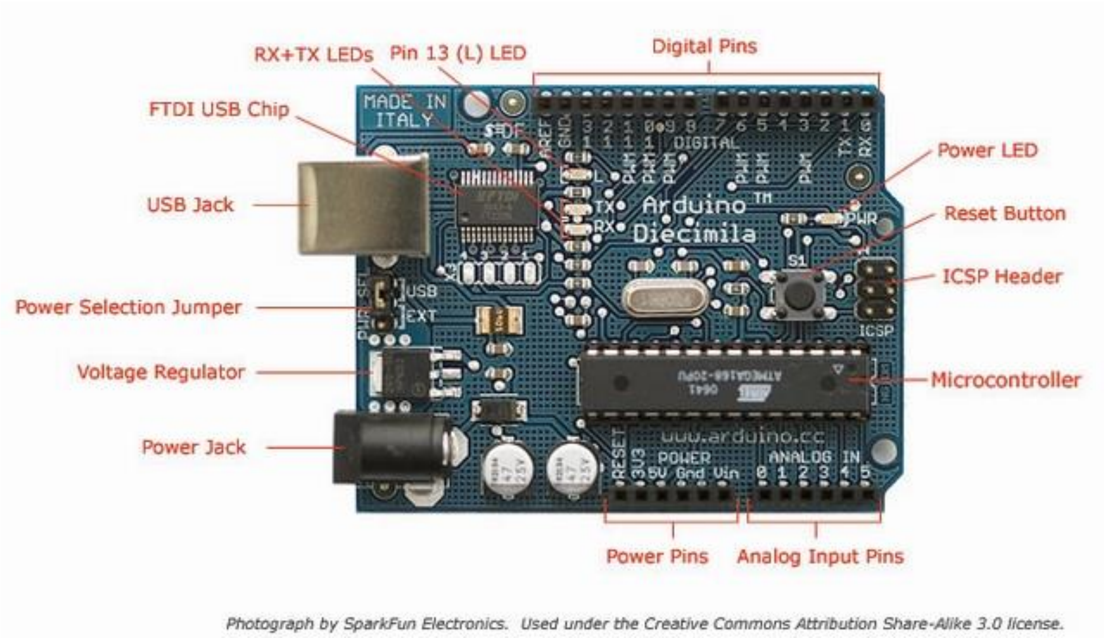
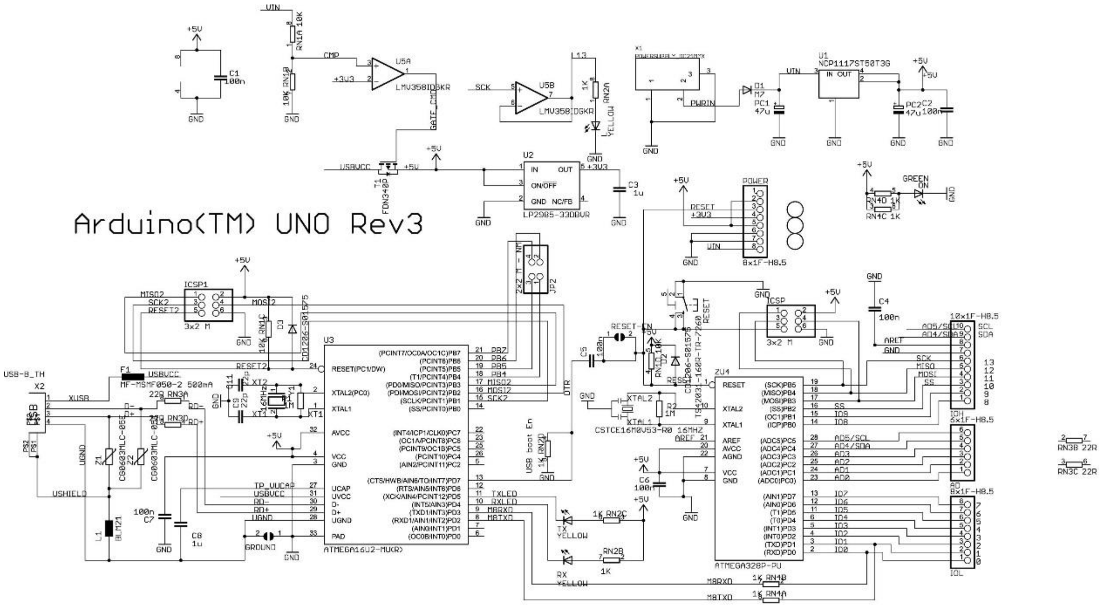


Figure: Arduino Diecimila board



Arduino UNO schematic

In our project we used 2 types of Arduino, Arduino Uno, and Arduino Due. [39] [40]

Name	Processor	Frequency	Dimensions	Voltage	Memory	EEP ROM	SPAM	Digital pins	Analog pins	Release Date
Arduino Uno	ATmega328P	16 MHz	2.7 in × 2.1 in [68.6 mm × 53.3 mm]	5v	32 KB	1 KB	2 KB	14	6	Sept. 24, 2010
Arduino Due	AT91SAM3X8E [17] (ARM Cortex-M3)	84 MHz	4 in × 2.1 in [101.6 mm × 53.3 mm]	3.3v	512 KB	0 KB	96 KB	54	12	Oct. 22, 2012

4.4 Arduino program Analysis

4.4.1 In-built Clock System:

Specifications and circuit:

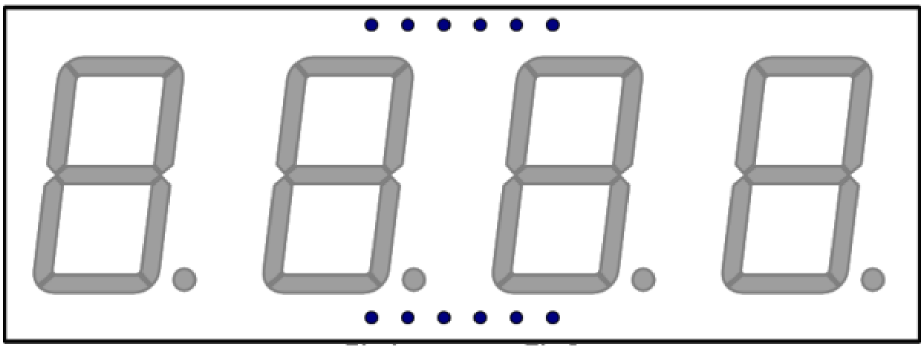


Figure 1: Eight-segment 4 digit display with pins

Our system depends on the current time, in order to judge whether the system should be ON or OFF. For this we have added a visible clock to the system using a 4 digit display (Figure 1) CC-5461AS. The way 4 digit displays work is usually by 12 digital pins, 6 on top and 6 on the bottom, where pin 1 is located in the bottom left, and the pin number increases anti-clockwise. Each digit is split up in eight segments (Figure 2), and each of these segments has its own digital pin into the microcontroller, and works as an anode. In this case we are using pins 28 to 35 as listed in the first section of the program. So, in practice, all 4 anodes for segment A (one for each digit) will be connected together, and sometimes to a driver circuit. Cathodes, on the other hand, are put together by digits, so cathodes A, B... DP will be connected together for each digit, occupying pins 24 to 27. This way, instead of having to use 32 pins (8 segments x 4 digits), the display will only use 12, and a sequential system to display several digits. This means that the display will only display one digit at the time, then clear all LEDs and then go to the next digit. By doing this in a loop with little delay, it appears to be showing all LEDs ON at the same time with different outputs, although this would not be possible with only 12 pins. This is multiplexing the cathodes. Regarding how to know which pin is which, the technical datasheet provides this information (Figure 3). Additionally, an alternative to this circuit could be done using for example 2N2222 transistors and a ground connection (Figure 4) or a 74HC595 shift register, instead of using the digital pins as a temporal ground when on LOW and cutting the current flow when on HIGH. Of course, when connecting the digital pins to LEDs we put a resistor in between to avoid damage to the LEDs, in this case 220 ohms. Our circuit connections are shown in figure 5.

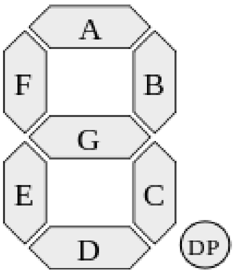


Figure 2: 8 Segments of a digit

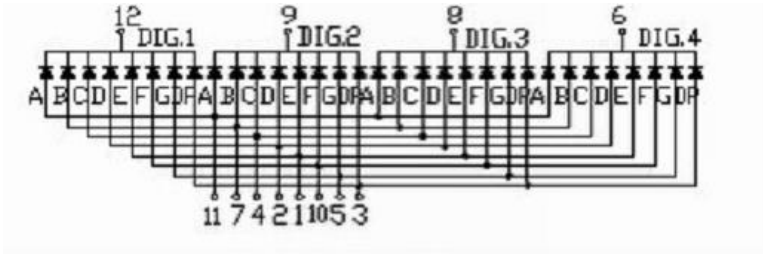


Figure 3: CC-5461AS pin settings

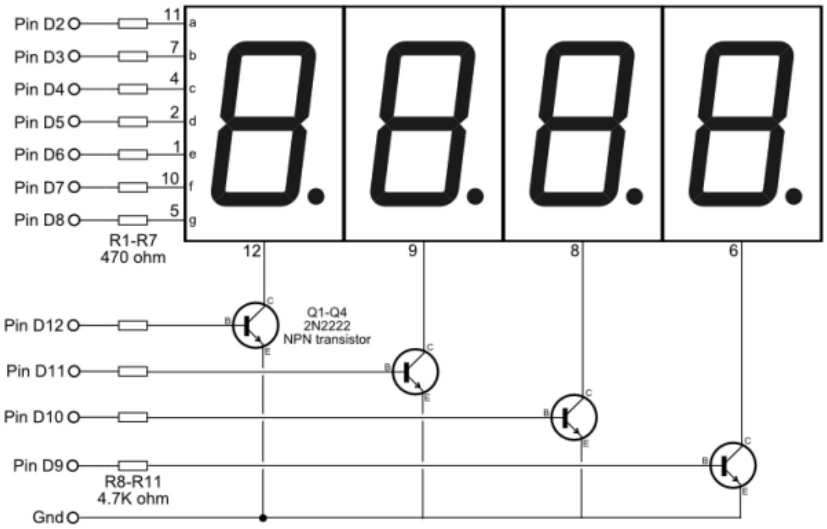


Figure 4: Alternative circuit using transistors

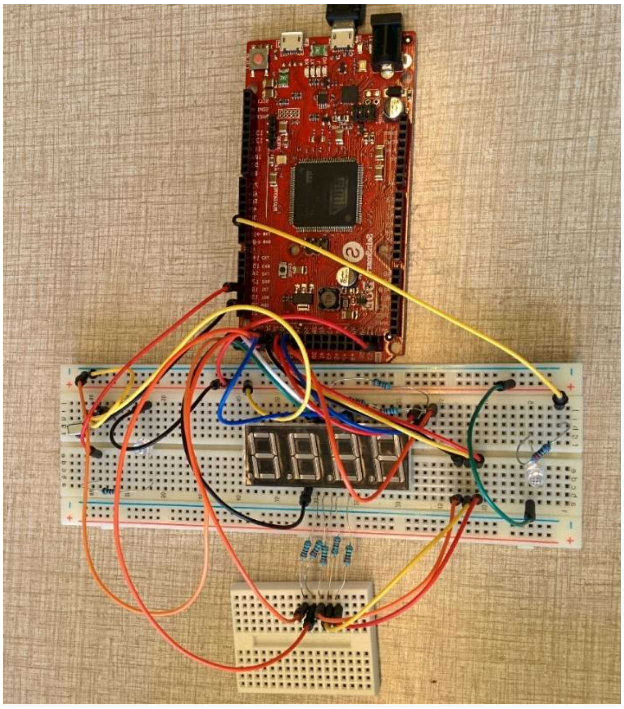


Figure 5: Our circuit for the timer

Software:

For the program first we create variables for the anode and the cathode, naming them a-g and p and d1-d4 so that they can be easily changed if we change the pin number, without having to look throughout the whole program for every place the variable is used. In the void setup section we make sure that all the pins are on output mode using the pinMode function.

In the void loop section, the timer circulates the digits and outputs the value one by one rapidly. To do this, it first uses the clearLEDs command, which changes the digital value of pins a-g and p to LOW, so that there are no residue lines ON from the previous digits, resulting in numbers not making sense, and eventually everything being 8, since it has all segment pins on HIGH. Next, we use the pick digit function with the input variable of the digit we want to edit. This function is also defined outside the loop, and it deactivates all digits by changing their value to high, and then depending on the input uses a switch flow control structure to activate the digit. In order to keep track of the current value, we use a variable n, which changes every loop, and is reset at 24:00, so it becomes 00:00. The value to display is calculated with the formulas shown, using division and modulo. The value gotten by the formula is recorded in currentd_ where the line is replaced by the current digit. Next, the digit is called using the pickNumber command, which is only a switch flow structure to trigger the right output function. The output pins that should be active are shown in the following table:

Number	a	b	c	d	e	F	g
0	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	LOW
1	LOW	HIGH	HIGH	LOW	LOW	LOW	LOW
2	HIGH	HIGH	LOW	HIGH	HIGH	LOW	HIGH
3	HIGH	HIGH	HIGH	HIGH	LOW	LOW	HIGH
4	LOW	HIGH	HIGH	LOW	LOW	HIGH	HIGH
5	HIGH	LOW	HIGH	HIGH	LOW	HIGH	HIGH
6	HIGH	LOW	HIGH	HIGH	HIGH	HIGH	HIGH
7	HIGH	HIGH	HIGH	LOW	LOW	LOW	LOW
8	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH
9	HIGH	HIGH	HIGH	HIGH	LOW	HIGH	HIGH

Concerning the dot, it uses the dispDec command, which is triggered separately, in this case only for the second digit, since it has hours:minutes. As the code shows, all it does is a digitalWrite command HIGH for pin p. Finally, there is a small delay, to make the LED ON long enough to be visible, low enough for the eye not to see blinking LEDs, and the right speed regarding time passing (change every minute).

4.1.2 Infrared remote control system:

Specifications and circuit:

In order to get inputs from the user, our system uses an infrared control (figure 1) and an infrared sensor. The decoded signal that the receiver gets is the following values:

Button	**Signal**
Power	16753246
Mode	16736925
Mute	16769565
Play/Pause	16720605
Previous	16723445
Next	16761405
EQ	16769055
Minus (-)	16754775
Plus (+)	16748655
0	16738455
Back/Arrows	16750695
U/SD	16756815
1	16724175
2	16718055
3	16743045
4	16716015
5	16726215
6	16734885
7	16728765
8	16730805
9	16732845
Repeat	4294967295

The main purpose of the remote system is to allow the user to adjust the clock to the correct time. In order to do so, the user will need to:

1. Activate the clock editing mode, which will pause the clock, and select digits one by one (with the exception of the hour digits) to edit, by pressing the MODE button.

2. Circulate the digits on both directions using the Previous/Next buttons, being able to loop from the first one to the last and vice versa.
3. Change the digit value either adding or subtracting with the +/- buttons or inputting a new number. Again, it is possible to loop using the +/- buttons.
4. Deactivate the clock editing mode by pressing the arrows button so that the timer will continue from the established time.

Regarding how the receiver works, the block diagram shows (Figure 2) the basic parts of it, and the connection pins required, which, in this case, are V_{CC} of 5 V, an output signal or V_{OUT} into a digital pin and a ground.

Figure 1: The Remote control that we are using

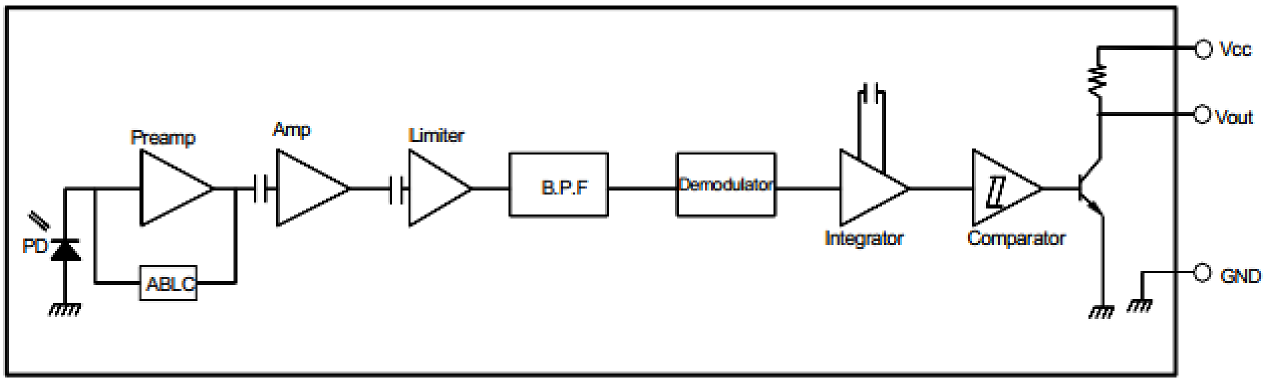


Figure 2: VS1838B block diagram

Software:

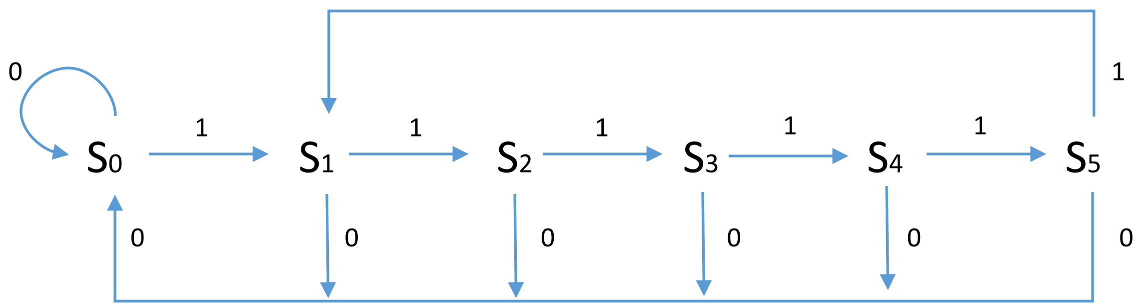
In order to use the signals received by the receiver we are using a modified version of the library called IRemote, renamed IRemote2 to avoid clashes between the original library and the modified version. The library decodes the signals and when it detects that a button is being pressed down for more than one input, it replaces the input with the repeat signal, so that the system does not do an intended action more than once. This function could be used to make the user to hold down the button for a certain amount of time. In our program, the first thing we do is import the library mentioned, via a #include < > command. Next, we define the variables we need, which are defining IRpin (the digital pin where Vout is connected) and two library defined variables with the types IRrecv and decode_results. The way the library works is that you write the IRecv variable name .resume() to get the next value. This triggers a switch structure, which verifies which button was pressed using the signal, and triggers a response depending on certain conditions. For example, button 9 will only work if the clock editing mode is on, and only change the values if the editing digit is either digit 4 (minutes) or the hour digits. These conditions avoid getting invalid times and causing errors.

4.5 State diagrams

Our system currently runs a software section that checks if it should lower the volume by adding and evaluating averages of values. Next, we will investigate the possibility of replacing it with a logic gate system. There are several ways to do this, and we will evaluate 2 ways.

4.5.1 X states, 1 input, all 1 system

This system has 1 digit input, either 1 for “too high" volume, or 0 for within the boundaries. In order to trigger the lower volume action, 5 consecutive “too high” inputs are required. This will create a 6 state system.



As the diagram shows, the number of states will be the amount of readings required plus 1, giving a maximum of 7 too high readings before the system will become very complex when doing Karnaugh maps. Next we do the State transition table:

Current State	Input	Next State
S_0	0	S_0
	1	S_1
S_1	0	S_0
	1	S_2
S_2	0	S_0
	1	S_3
S_3	0	S_0
	1	S_4
S_4	0	S_0
	1	S_5
S_5	0	S_0
	1	S_1

From this we can give each state its Q_0, Q_1 and Q_2 values. We need three Qs, because 3 can allocate a maximum of 8 states (2^3), while 2 can only have 4 states (2^2).

S	**Q_0**	**Q_1**	**Q_2**	**A = 0**			**A = 1**		
S_0	**0**	**0**	**0**	0	0	0	0	0	1
S_1	**0**	**0**	**1**	0	0	0	1	0	0
S_2	**1**	**0**	**0**	0	0	0	1	0	1
S_3	**1**	**0**	**1**	0	0	0	1	1	0
S_4	**1**	**1**	**0**	0	0	0	1	1	1
S_5	**1**	**1**	**1**	0	0	0	0	0	1

The next step is to do the Karnaugh maps for each Q, which are shown next:

$D_0 = A * Q_1' * Q_2 + A * Q_0 * Q_1' + A * Q_0 * Q_2'$

$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	0	1	0	0

$Q_0 = 1$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	1	1	0	1

$D_1 = A * Q_0 * Q_1' * Q_2 + A * Q_0 * Q_1 * Q_2'$

$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	0	0	0	0

$Q_0 = 1$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	0	1	0	1

$D_2 = A * Q_1' * Q_2' + A * Q_0 * Q_2' + A * Q_0 * Q_1$

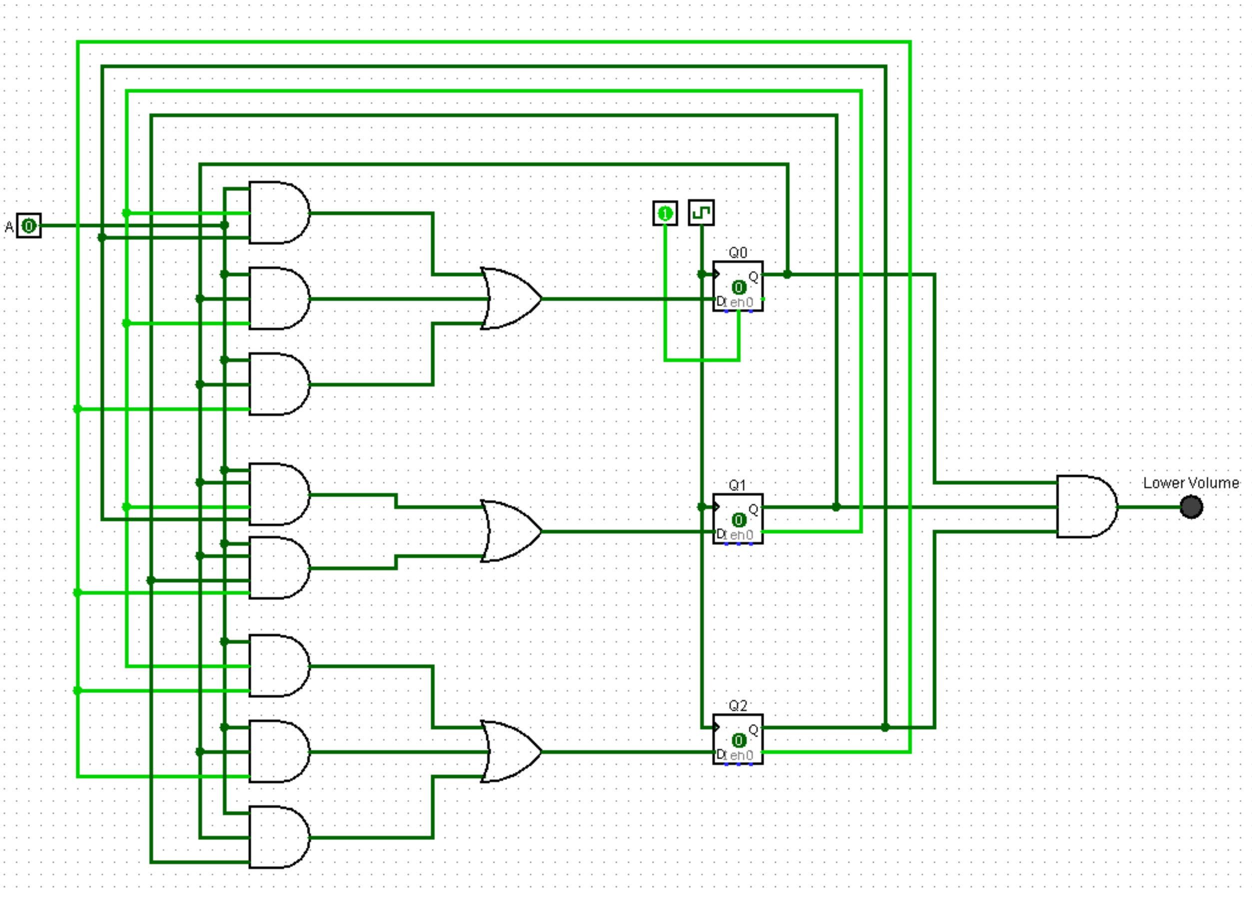
$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	1	0	0	0

$Q_0 = 1$

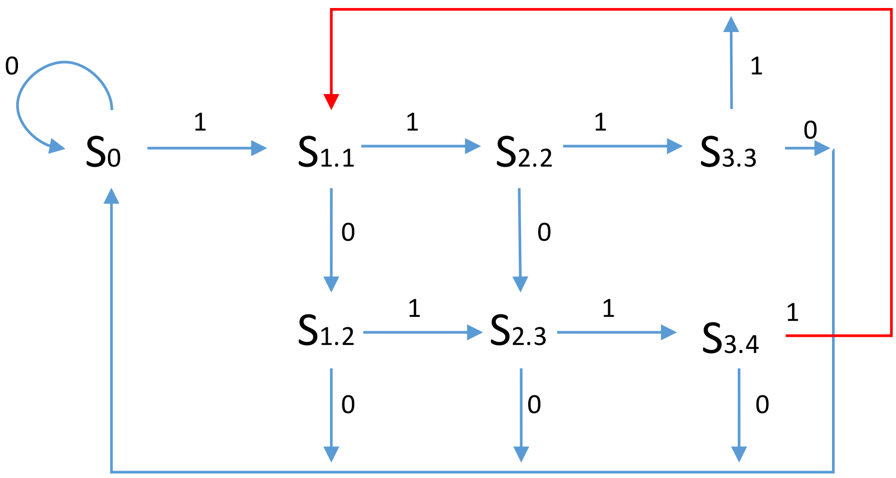
A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	1	0	1	1

With these formulas, we can now create the circuit that would perform this operation.



4.5.2 3/4 "too high" results

This way you only need 3 "too high" results out of 4 consecutive readings. It is the maximum amount possible without having to use 6 variable Karnaugh maps, since adding one more reading will add at least 2 states. As previously, the first step is to define the state diagram:



Next we do the State transition table:

Current State	**Input**	**Next State**
S_0	0	S_0
	1	$S_{1.1}$
$S_{1.1}$	0	$S_{1.2}$
	1	$S_{2.2}$
$S_{1.2}$	0	S_0
	1	$S_{2.3}$
$S_{2.2}$	0	$S_{2.3}$
	1	$S_{3.3}$
$S_{2.3}$	0	S_0
	1	$S_{3.4}$
$S_{3.3}$	0	S_0
	1	$S_{1.1}$
$S_{3.4}$	0	S_0
	1	$S_{1.1}$

From that table, we define the Q values:

S	Q_0	Q_1	Q_2	**A = 0**			**A = 1**		
S_0	**0**	**0**	**0**	0	0	0	0	0	1
$S_{1.1}$	**0**	**0**	**1**	0	1	0	1	0	0
$S_{1.2}$	**0**	**1**	**0**	0	0	0	1	0	1
$S_{2.2}$	**1**	**0**	**0**	1	0	1	1	1	0
$S_{2.3}$	**1**	**0**	**1**	0	0	0	1	1	1
$S_{3.3}$	**1**	**1**	**0**	0	0	0	0	0	1
$S_{3.4}$	**1**	**1**	**1**	0	0	0	0	0	1

And proceed to do the Karnaugh maps:

$D_0 = A * Q_1' * Q_2 + A * Q_0' * Q_1 * Q_2' + Q_0 * Q_1' * Q_2' * Q_0 + A * Q_1'$

$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	0	1	0	1

$Q_0 = 1$

A \ Q_1 Q_2	00	01	11	10
0	1	0	0	0
1	1	1	0	0

$D_1 = A' * Q_0' * Q_1' * Q_2 + A * Q_0 * Q_1'$

$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	1	0	0
1	0	0	0	0

$Q_0 = 1$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	1	1	0	0

$D_2 = A * Q_0' * Q_2' + A' * Q_0 * Q_1' * Q_2' + A * Q_0 * Q_2 + A * Q_0 * Q_1$

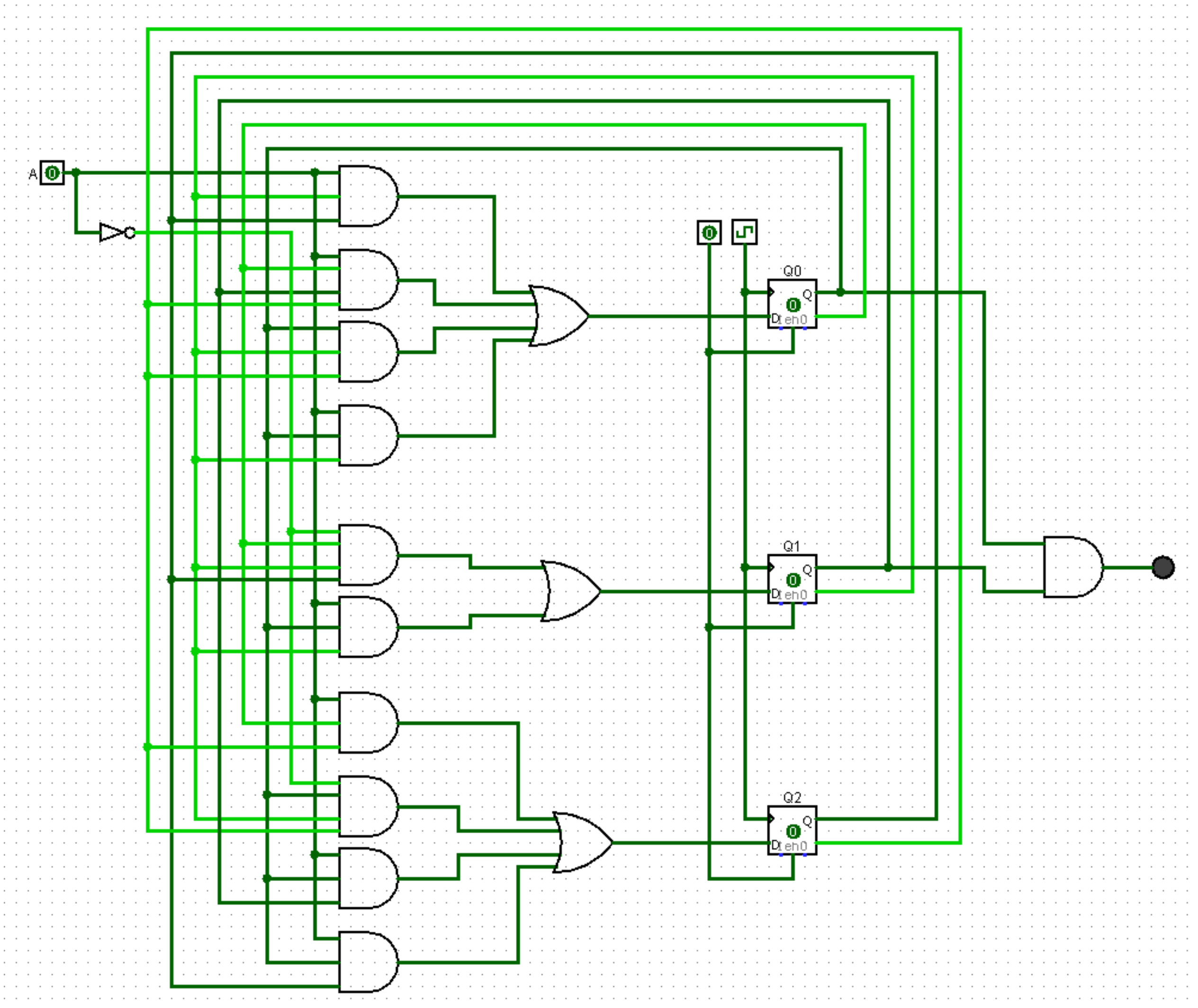
$Q_0 = 0$

A \ Q_1 Q_2	00	01	11	10
0	0	0	0	0
1	1	0	0	1

$Q_0 = 1$

A \ Q_1 Q_2	00	01	11	10
0	1	0	0	0
1	0	1	1	1

And finally get the circuit:



4.6 3dCAD usage in the project

All of our group members have decided to attend the optional 3dCAD course. The software that we were using was Inventor, used for many engineering purposes. It allows creating and managing 3D objects, as well as checking how much pressure they can endure. We had 10 lectures - 4 with basics and 6 with advanced techniques.
Soon after the first lecture we decided to use the gained knowledge for our project. We created a design of a box for our prototype in 3D in order to make it more presentable.

We used mostly the techniques acquired on the first few lectures; nevertheless, it was enough to create the prototype in 3D. However, we used some advanced techniques to create the screws for the box (namely the Coil function for the thread).

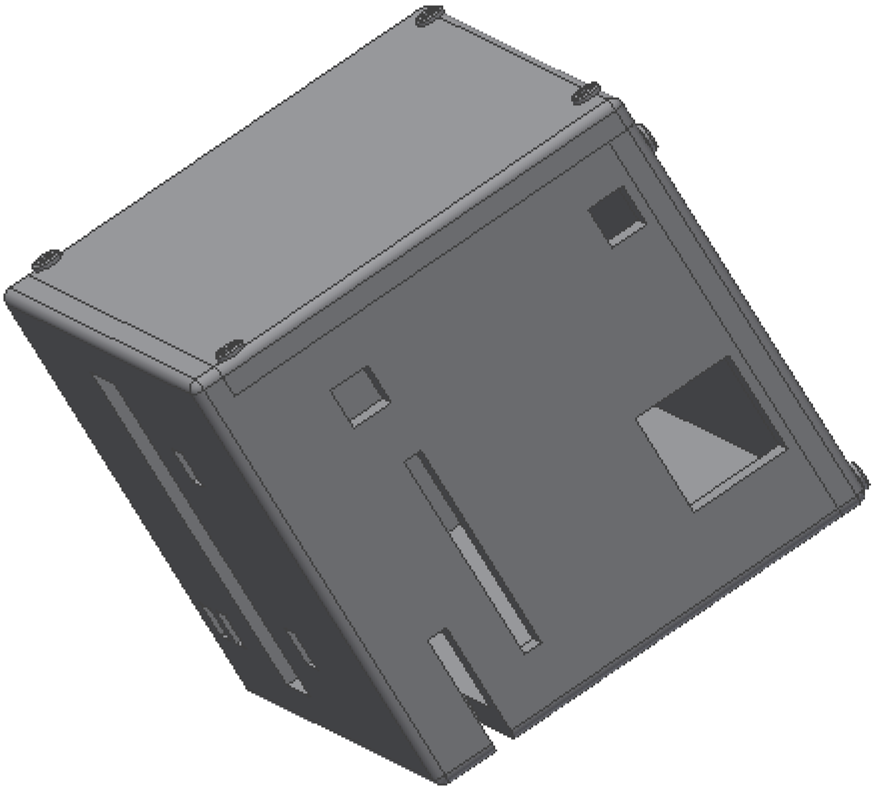


Figure 1: The assembled box

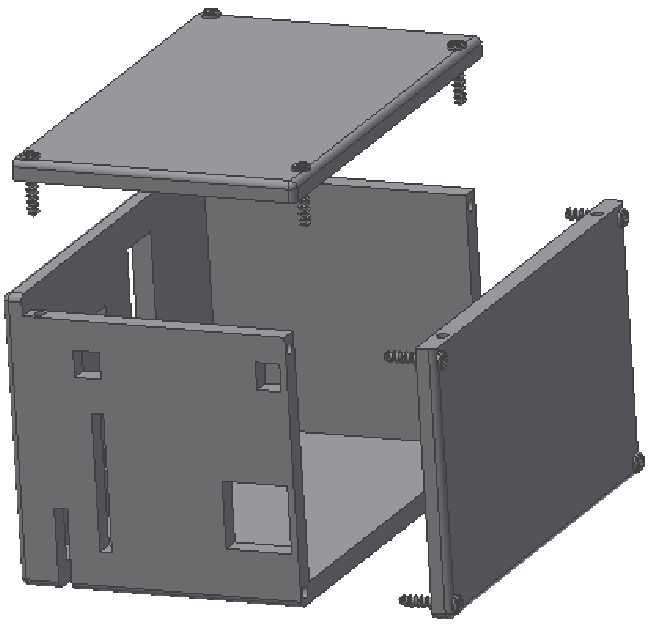


Figure 2: Structure of the box

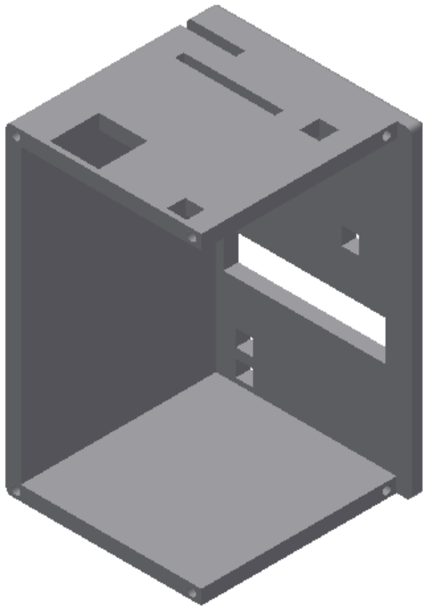


Figure 3

Figure 4: The screw

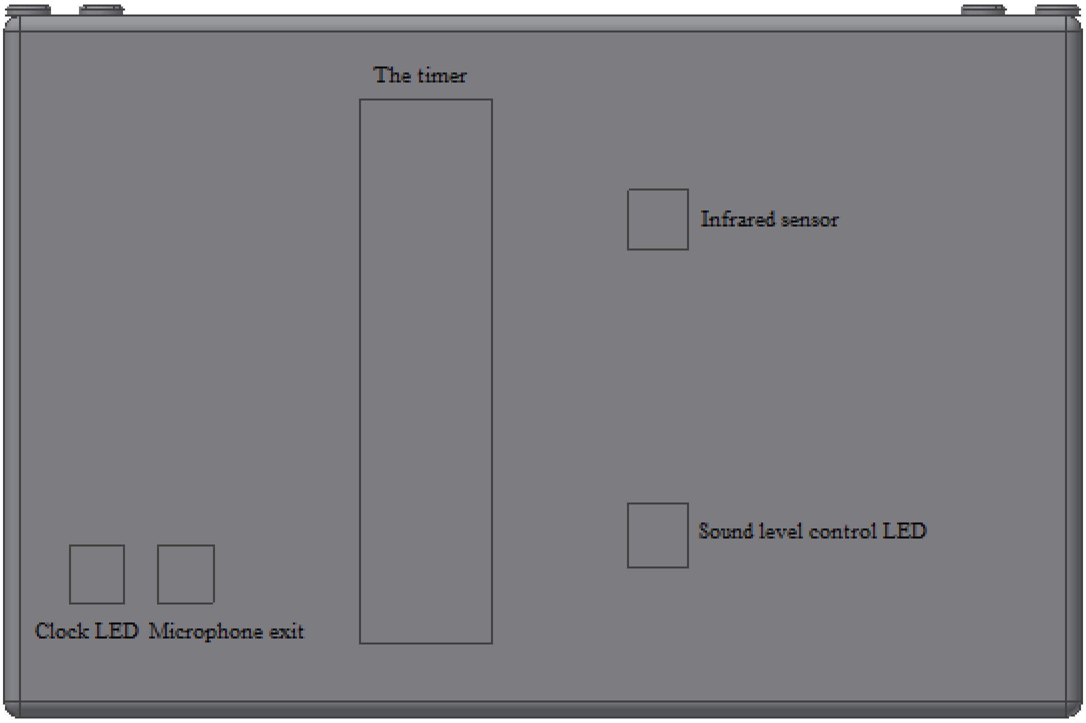


Figure 5: Top of the box

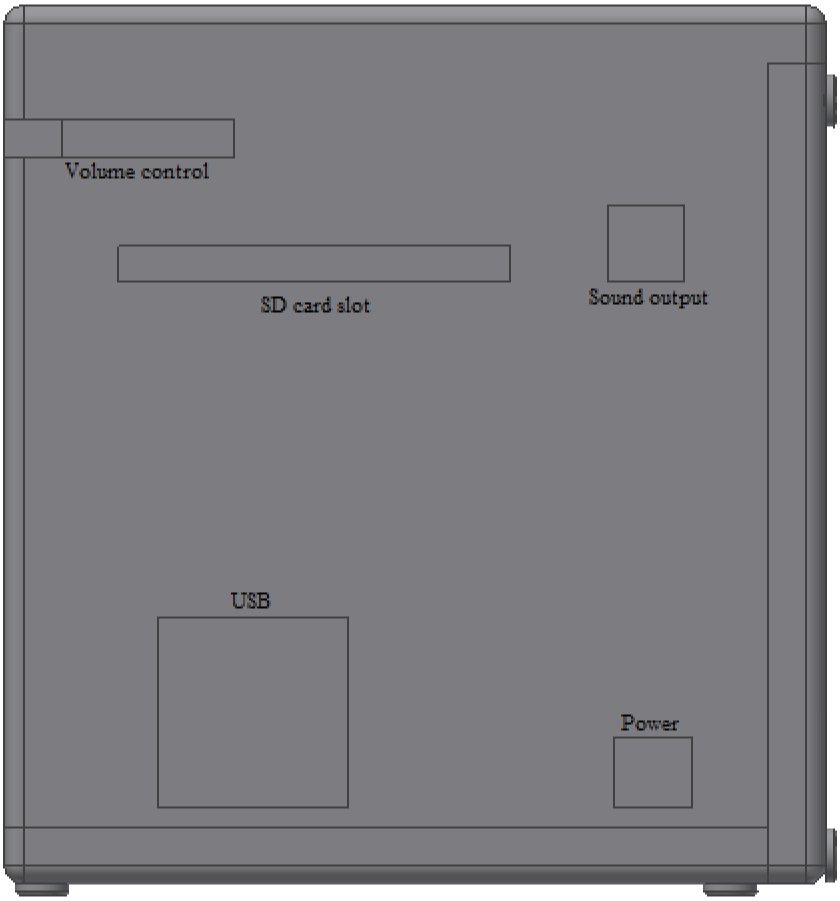


Figure 6: Side of the box

4.7 Wave shield

4.7.1 dB to Voltage transformation:

From the sound theory we have seen that we can calculate the voltage output of the microphone based on the sensitivity:

$$Transfer\ factor = 10^{-52/20}$$

This will offer us a value of 2.5119 mV/Pa or 2.5119mV output for a pressure of 94dBSPL at a voltage of 1V applied on the microphone, which is unfortunately too small for the Arduino's analogue to digital convertor to process. That is why we applied a gain of x200 or 46dBu to the pre-amp.

$$V = \frac{V_{out}}{V_{in}}\ Volts\ (gain\ factor)$$

Voltage (amplitude) gain			Power (energy) gain	
Voltage ratio Factor V_2/V_1	Voltage amplification G_V in dB		Power ratio Factor P_2/P_1	Power amplification G_P in dB
10^3	+60		10^6	+60
10^2	+40		10^4	+40
10^1	+20		10^2	+20
$\sqrt{10} = 3.16$	+10		10	+10
2	+6		4	+6
$\sqrt{2} = 1.414$	+3		2	+3
1	**±0**		**1**	**±0**
$1/\sqrt{2} = 0.7071$	−3		1/2 = 0.5	−3
1/2 = 0.5	−6		1/4 = 0.25	−6
$1/\sqrt{10} = 0.316$	−10		$10^{-1} = 0.1$	−10
$10^{-1} = 0.1$	−20		$10^{-2} = 0.01$	−20
$10^{-2} = 0.01$	−40		$10^{-4} = 0.0001$	−40
$10^{-3} = 0.001$	−60		$10^{-6} = 0.000001$	−60
$V_2/V_1 = 10^{(G_V \text{ in dB}/20)}$	$G_V = 20\times\log(V_2/V_1)$		$P_2/P_1 = 10^{(G_P \text{ in dB}/10)}$	$G_P = 10\times\log(P_2/P_1)$

From this formula we can also find out the amplification level:

$$L_V = 20\log\frac{V_{out}}{V_{in}}\ dB\ (Amplification\ level)$$

Voltage and power gain in amplifiers

Because the SPL is a logarithmic value we cannot create a linear function for the microphones output.

Therefor we will get to calculate the output voltage using the Sound pressure formulas as follows:

$$V = 10^{\frac{(mic\ sensitivity + L_P - 94)}{20}}\ Volts$$

In the microphone schematic we can see that we have a Voltage divider R2 right before the microphone. The purpose of this divider is not only to protect the microphone but also to decrease the input voltage in the amplifier.

From the LM386 datasheet we can see that the 10uF capacitor applied between the pins 1 and 8 offers us a gain of factor of 200 of level of 46dB.

Therefore we can conclude that the output voltage which will be read by the Arduino can be calculated by the following formula:

$$V = 10^{\frac{(sensitivity + L_P - 94)}{20}} * 200 * 0.9\ Volts$$

4.7.2 Software

For the wave shield and microphone input system, we are using a separate board, the Arduino UNO. This is because the wave shield library is incompatible with the Arduino UNO board, and because the microphone input and processing takes considerable time, thus creating an inconsistent speed in the timer and brightness issues. The default library for the wave shield is the WaveHC library, which comes from the AF_Wave library, which was discontinued in 2011. However, when changing library, the developer did not include the volume changing function, so we were forced to use the older version. In order to make a 3 year old library work, we had to go into the library's .h and .cpp files to modify some things, such as replacing the no longer existent WProgram.h library with the newer version Arduino.h, and editing some defines. The way the volume changing works is the following: instead of changing the signal, it changes the digital value of the outputs, reducing the voltage and thereby the sound level. However, this comes at the cost of having a limited range, from 0 to 12, where 0 is the maximum, and 5 is barely audible if the output when using value 0 was at an acceptable volume. Also, this way of doing it causes issues on the audio quality, mainly because when the volume value is changed, the speaker will generate a click sound, which can be annoying, especially if dealing with a song that has a very variable volume level, from very low to very high.

In practice, our program is playing a .wav file which because of limitations in the library must be mono (its output might still be in stereo though, although it will be the same sound on both sides) and a maximum of 22 050 Hz sample rate and 16-bit sample size. Then the microphone is picking up the sound level and averaging the values. In order to eliminate wrong values, we have eliminated the reading 0 and 1023 which are the maximum and minimum, since these are the ones that most often show wrongly. We are using average values to increase the accuracy of the readings, and once the average reaches a defined low value of 210 or a defined value of 450, the volume is changed. Also, from time to time, the average is reset, so that new values will always have a weight on the current average, instead of eventually being insignificant.

Finally, the UNO board receives an input from the Arduino DUE board, for it to know if the volume adjusting should be active or not.

We attempted to make the board take the microphone analogue inputs and convert them into decibel values, but there were too many wrong values for it to be accurate. If this was not the case we could make more personalized specifications, such as hard clipping, instead of the current soft clipping, or even some system to turn everything completely down if it reaches a threshold. However, for now our system does a soft clipping effect on the volume, as shown in the figure:

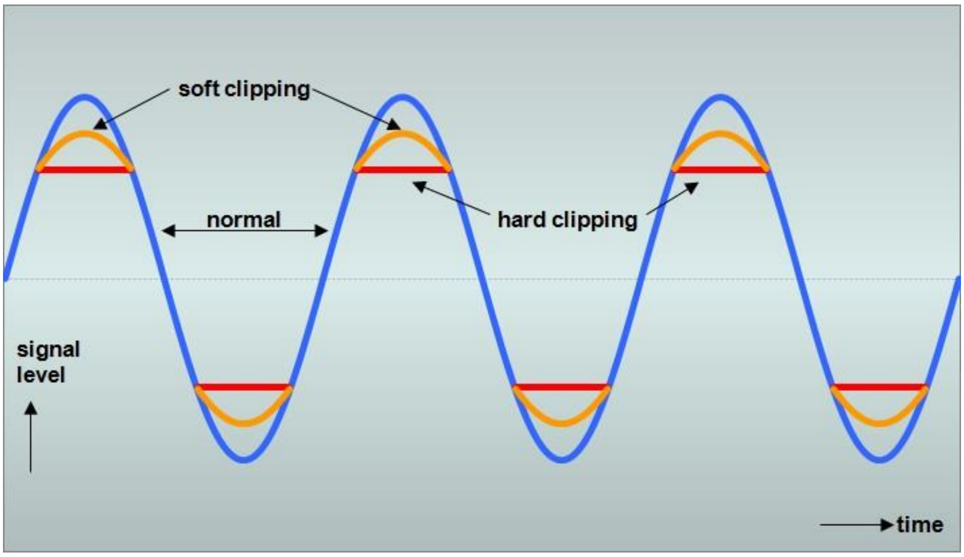


Figure: Different signal clipping modes on signal level.

4.8 Microphone

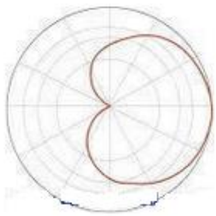
A microphone (MIC) is an acoustic-to-electric transducer that converts sound in air into an electrical signal. There are many microphone applications used in many devices such as telephones, recorders, radios and megaphones. [41]

The history of microphones dates back to 600 BC with the invention of masks with specially designed mouth openings that acoustically increase the voice in amphitheaters. In recent history, there were other experiments for the invention of the microphone, the first of which was by the English physicist Robert Hooke 1665. He invented "lovers' telephone" made of stretched wire with a cup attached at each end.

An early sound transmitter was invented by the German inventor Johann Philipp Reis by using a metallic strip attached to a vibrating membrane that would produce intermittent current. But in 1876 Scottish-American Alexander Graham Bell achieved better results by attaching the diaphragm to a conductive rod in an acid solution.

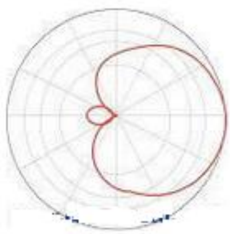
4.8.1 Types of microphones

- **Carbon**: This kind of microphones used carbon dust to interpret the audio signal. These microphones were commonly used in telephones and can still be found in some of today's telephones. A current is run through the carbon dust and the carbon is compressed by sound waves as they hit the diaphragm.
- **Dynamic:** In a dynamic microphone, the diaphragm moves a coil between two magnets (positive and negative) when sound waves hit it and the movement creates a small current**.**
- **Cardioids:** Has a strong sensitivity directly in front of the microphone, coupled with good sensitivity at the sides and rejection from behind (as in Fig 1)



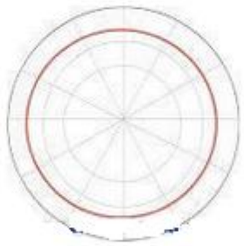
(Fig 1)

- **Supercardioid:** Similar to the cardioid, but with slightly reduced sensitivity around the sides. The supercardioid's pickup range includes a small area just behind the microphone (as Fig 2).



(Fig 2)

- **Omnidirectional:** a microphone that picks up sound equally from all directions. This is the only microphone not prone to a proximity effect (Fig 3). Proximity effect is an increase in sound response when the source is close to the microphone. [42]



(Fig 3)

4.8.2 Electret microphones [43]

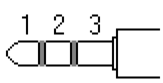
We used an electret microphone in the project, because it is sensitive, durable, extremely small in size, has low power requirements and inexpensive. Electret microphones are used in very many applications where small and inexpensive microphones with reliable performance characteristics are used.

The basic electret microphone is powering circuit which you can use as generic reference when receiving circuits which use electret microphones. The resistance is determined by R1 and R2. If you leave out R2, the output resistance is almost the resistance of R2.

Basic circuit

```
    +--------------------------- battery  -V     (3 to 12 Volts)
    |
   2k2 R1
    |
    o---------- 10uF ------o----- output
    |+                     |
 CAPSULE                  10k R2
    |-                     |
    +----------------------o----- GND       +V
```

Sound Blaster soundcards (SB16,AWE32,SB32,AWE64) use 3.5 mm stereo jack for the electret microphones. The microphone connector uses he following wiring:



1. Signal input
2. +5v
3. 0v

3.5mm plug

The following are specs for the Sound Blaster microphone input:

- Input Type: Unbalanced Low Resistance
- Input Sensitivity: Approx. -20dBV (100mV or 0.1Volt)
- Input Resistance: 600 to 1500. (Ohms)

- Input Connector: 3.5mm Mini plug (Stereo Jack)
- Input Wiring: Audio on Tip, Ground on Sleeve, 5Volts DC (power supply)

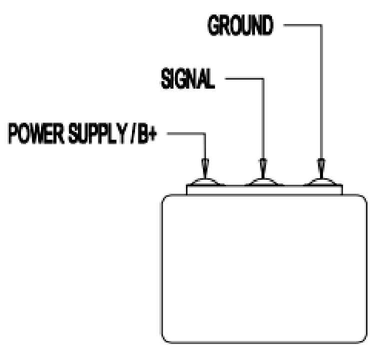
4.8.3 Electret microphone circuit [44]

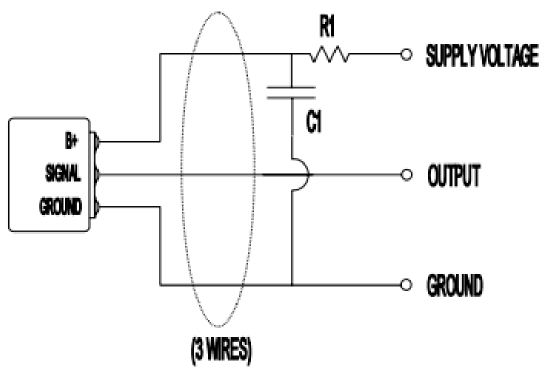
3-wire connection

The connection is in low voltage applications, typically operating with a supply voltage of 1.3 volts. This configuration uses three wires to connect the microphone signal output, supply and ground. The commonly used color scheme for cabled microphone assemblies is red for supply, white for signal and black for ground.

This type consists of a capacitor and a resistor which can be used on the supply line in the connecting circuit to make the microphone's pre-amplifier less susceptible to power supply disturbances.

The figures below show the microphone and a 3-wire connection with power supply filtering. The resistor R1 should be chosen such that the voltage drop across R1 does not reduce the operating voltage of the microphone pre-amplifier circuit (i.e. 1kΩ-2kΩ).





2-wire connection

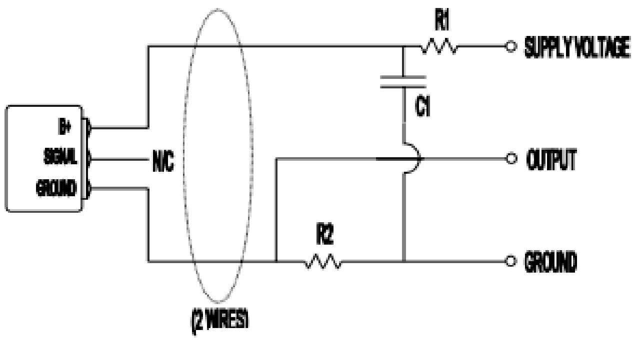
An electret microphone may also be wired for operation using only 2 wires in several different configurations, operating characteristics of the microphone pre-amplifier, and thus alters the performance of the microphone. For each of the configurations, an external resistor is needed. The power supply filtering mentioned above may also be included.

The 2-wire options as following

- Similar drain current (~20µ amps)
- Lower sensitivity
- R2 value 10kΩ – 250kΩ (w/ increasing B+)

- Higher power supply
- Characteristics similar to 3-wire configuration. w/ increasing R2

R2	Power Supply	1kHz sensitivity change
50kΩ	+5V	-3dB



4.9 Microphone sensitivity

International standards have established 1 pascal (Pa) as 94 dBSPL (sound pressure level).

This reference point is now accepted for specifying the sensitivity of microphones.

The µbar found in some non-European specifications refers to 74 dBSPL (20 dB less than 1 Pa) and the sensitivity or the sensitivity factor is not expressed as transfer factor in the usual form of "mV/Pa" as open circuit voltage rating. In the data sheets the sensitivity always applies to the frequency 1 kHz as reference, unless otherwise noted.

Microphones simply convert the sound pressure deviations p (pascals Pa) to audio voltage V (volts V).

Bear in mind that the output power of the microphone is not energy, but voltage.

Calculation formulas:

1. Sensitivity in dB re 1 V/Pascal
$$Sensitivity = 20 \times log(transfer\ factor)$$
2. Transfer Factor in mV / Pascal
$$Transfer\ Factor = 10^{sensitivity/20}$$

Example:

The microphone used for the prototype has a sensitivity of -52 dB re 1V/Pa. In this case we can calculate the transfer factor using the above formula as:

$$Transfer\ factor = 10^{-52/20}$$

This will offer us a value of 2.5119 mV/Pa.

Because of the small voltage value a preamplifier is required in order to have a better reading of the output values.

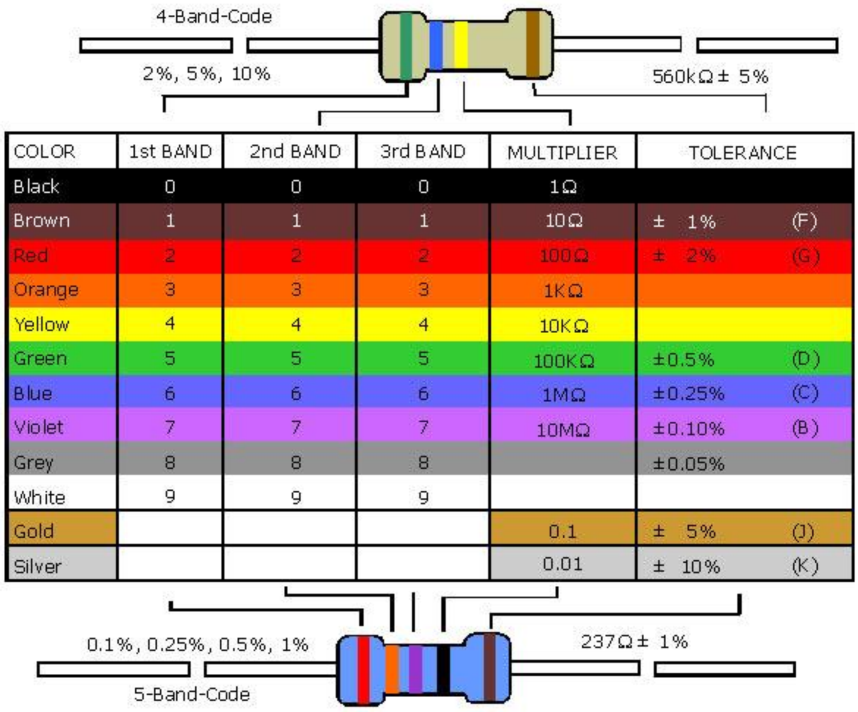
4.10 Components Description and Microphone Preamplifier circuit

4.10.1 Resistors

A resistor is a component of an electrical circuit that resists the flow of electrical current. A resistor has two terminals across which electricity must pass, and is designed to drop the voltage of the current as it flows from one terminal to the next. A resistor is primarily used to create and maintain a known safe current within an electrical component.

Resistance is measured in ohms, after Ohm's law. This rule states that electrical resistance is equal to the drop in voltage across the terminals of the resistor divided by the current being applied to the resistor. A high ohm rating indicates a high resistance to current. This rating can be written in a number of different ways depending on the ohm rating.

The amount of resistance offered by a resistor is determined by its physical construction. A resistor is coated with paint or enamel, or covered in molded plastic to protect it. Because resistors are often too small to be written on, a standardized colour coding systems used to identify them. The first three colours represent ohm value, and fourth indicate tolerance or how close by percentage the resistor is to its ohm value. This is important for two reasons: First, the nature of resistor construction is imprecise and second, if it is used above its maximum.



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE	
Black	0	0	0	1Ω		
Brown	1	1	1	10Ω	± 1%	(F)
Red	2	2	2	100Ω	± 2%	(G)
Orange	3	3	3	1KΩ		
Yellow	4	4	4	10KΩ		
Green	5	5	5	100KΩ	±0.5%	(D)
Blue	6	6	6	1MΩ	±0.25%	(C)
Violet	7	7	7	10MΩ	±0.10%	(B)
Grey	8	8	8		±0.05%	
White	9	9	9			
Gold				0.1	± 5%	(J)
Silver				0.01	± 10%	(K)

Maximum power handling capacity of resistors is:

¼ Watt	Max. 50mA
½ Watt	Max. 70mA
1 Watt	Max. 100mA
2 Watt	Max. 140mA
20 Watt	Max.440mA

4.10.2 Capacitors

The capacitor's function is to store electricity or electrical energy. The capacitor also functions as a filter passing alternating current (AC) and blocking direct current (DC).The symbol is used to indicate a capacitor in a circuit diagram. The capacitor is constructed with two electrode plates facing each other but separated by an insulator. When DC voltage is applied to the capacitor an electric charge is stored on each electrode. While the capacitor is charging up, current flows. The current will stop flowing when the capacitor has fully charged.

Disc Capacitor   Electrolytic Capacitor

Different kinds of capacitors use different materials for the dielectric.
Different kinds of capacitors are as follows:

- Electrolytic capacitors (electro chemical type capacitors)

- Tantalum capacitors
- Ceramic capacitors

In Disc capacitors, only a number is printed on its body so it is very difficult to determine its value in PF, KPF, uF, n, etc. In some capacitors, its value is printed in uF eg.0.1 in some others EIA code is used e.g. 104.

One or two numbers on the capacitor represents value in PF e.g. 8 = 8PF.
If the third number is zero, then the value is in P e.g. 100 = 100PF.
If the capacitor has three numbers and the third number is not a zero, it represents the number of zeros after the first and second digits e.g. 104 = 10 – 0000 PF.
If the value is obtained in PF, it is easy to convert it into KPF or uF.
PF / 1000 = KPF or n.
PF / 10, 00000 = uF.

For example, if the capacitor is 104, then it is 10-0000 PF or 100 KPF or n or 0.1 uF.

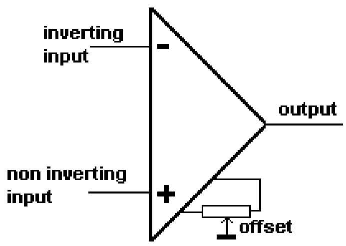
- Multilayer ceramic capacitors
- Polystyrene film capacitors
- Electric double layer capacitors (super capacitors)
- Polyester film capacitors
- Poly propylene capacitors
- Mica capacitors
- Metalized polyester film capacitors

- Variable capacitors

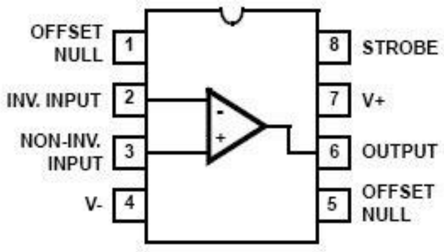
4.10.3 Integrated Circuits

Op Amp

The Op Amp was originally designed to carry out mathematical operations in analogue computers, such as bombsights, but was soon recognized as having many other applications. The Op Amp usually comes in the form of an 8 pin integrated circuit, the most common one being the type 741. It has two inputs and one output. The input marked with a “-“ sign produces an amplified inverted output. The input marked with a “+” sign produces an amplified but non inverted output. The Op Amp requires positive and negative power supplies, together with a common ground. Some circuits can be designed to work from a single supply. If the two inputs are joined together, then the output voltage should be midway between the two supply rails, i.e. zero volts.



If it is not, then there are two connections for adding a potentiometer, to remove this OFFSET. The Op Amp has a very high gain, typically (100 dB) 100,000 times. Looking at the diagram, an input with a swing of a fraction of millivolts produces an output that changes between + 12 volts and - 12 volts. In most cases this gain is excessive, and is reduced by negative feedback. Gain falls quite rapidly as the frequency increases. In fact the bandwidth (the point at which the output has fallen by 3 dB) is only 1 kHz. This is also improved upon by the use of negative feedback. The input impedance is high, 1M. The output impedance is low, 150 ohms.



4.11 Condenser/ Electret microphone

Condenser microphones

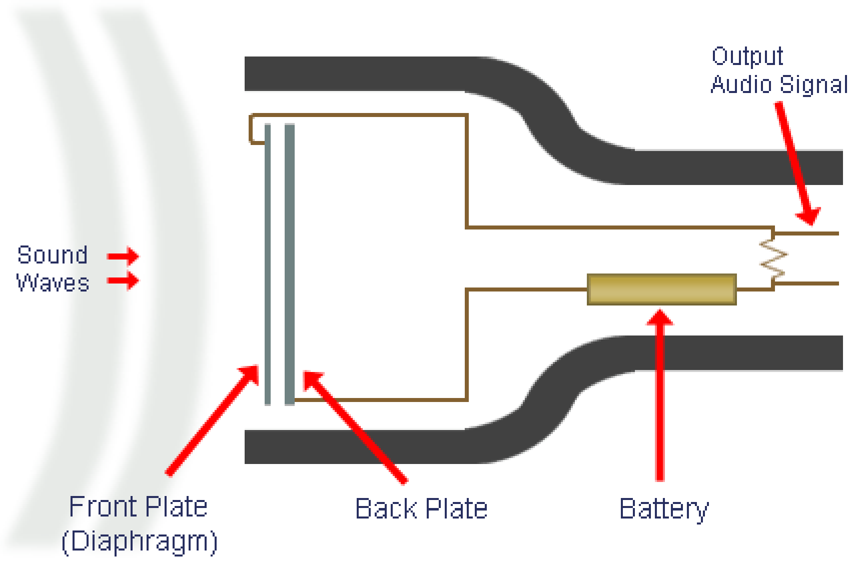
Condenser means capacitor, an electronic component which stores energy in the form of an electrostatic field. The term condenser is actually obsolete but has stuck as the name for this type of microphone, which uses a capacitor to convert acoustical energy into electrical energy.

Condenser microphones require power from a battery or external source, for example 48 volt phantom power from the mixing desk. The resulting audio signal is stronger signal than that from a dynamic. Condensers also tend to be more sensitive and responsive than dynamics, making them well-suited to capturing subtle nuances in a sound. They are not ideal for high-volume work, as their sensitivity makes them prone to distort.

How condenser microphones work

A capacitor has two plates with a voltage between them. In the condenser mic, one of these plates is made of very light material and acts as the diaphragm. The diaphragm vibrates when struck by sound waves, changing the distance between the two plates and therefore changing the capacitance. Specifically, when the plates are closer together, capacitance increases and a charge current occurs. When the plates are further apart, capacitance decreases and a discharge current occurs.

A voltage is required across the capacitor for this to work. This voltage is supplied either by a battery in the mic or by external phantom power usually +48v.



Cross-Section of a typical condenser microphone

The electret condenser microphone

The electret condenser microphone uses a special type of capacitor which has a permanent voltage built in during manufacture. This is somewhat like a permanent magnet, in that it doesn't require any external power for operation. However, good electret condenser microphones usually include a pre-amplifier which still requires power.

Other than this difference, you can think of an electret condenser microphone as being the same as a normal condenser.

Technical Notes:

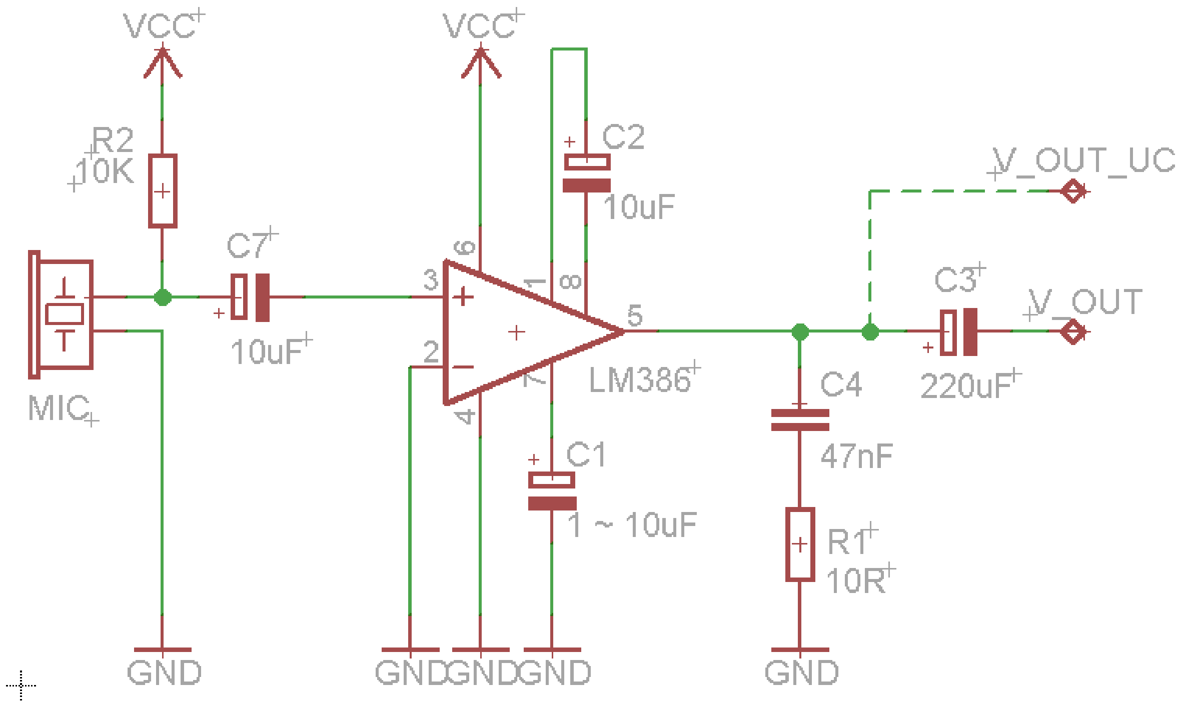
Condenser microphones have a flatter frequency response than dynamics.
A condenser microphones works in much the same way as an electrostatic tweeter (although in reverse).

4.12 The preamplifier circuit

One of the most used Op-AMP's in the case of Arduino is the LM386 amplifier which allows us to have a gain from x20 to x200.

4.12.1 Schematic used for the project

For our project we have used a basic schematic for a preamplifier:



Basic LM386 microphone preamplifier

In this schematic, a LM386-based audio amplifier takes its input from an electret microphone.

There are two output signals. V_OUT carries an AC-only voltage and should be used when interfacing with another piece of audio equipment. The voltage at V_OUT_UC is biased at 1/2 of the supply voltage V_{CC} and is the better option for microcontroller input.

If volume control is needed, the VR1 potentiometer from the original schematic can be added back, before the input pin 3.

4.12.2 Sound sensor for the Arduino

Arduino's analogue pins map a 0 ~ 5V voltage to a 0 ~ 1023 integer. In silence, an ideal sound sensor should consistently give readings of 511-512 – half of the 0 ~ 1023 range, corresponding to 1/2 VCC = 2.5V. When there is a very loud sound, the readings should sweep all the way to 0 and 1023. Moderate sounds should have readings somewhere in between. This makes it easier to set a threshold for detecting sounds of a given strength.

Sound	Readings range (ideal)	Amplitude (ideal)
(Silence)	511 ~ 512	1
Talking	300 ~ 723*	423*
Loud knock, broken glass	0 ~ 1023	1023

These particular values are just an example

4.12.3 LM386 pinout

Let's have a look at the pinout and see the role of each part in the schematic above.

1. Pins 1 and 8 control gain. When not connected (NC), the amplifier gain is 20. Adding a 10uF capacitor between them gives a gain of 200. Intermediate values are also possible, as described in the datasheet.
2. Pins 2 is the negative input – GND in our case.
3. Pin 3 is the positive input – i.e. the actual signal to be amplified. There is a 10K potentiometer before it, which adjusts the input signal level, acting as a volume control.
4. Pins 4 (GND) and 6 (Vs) provide the supply voltage for the amplification. For this setup a 4x AA NiMh battery pack is used, which provides ~5V.
5. Pin 5 is the output. It is biased to 1/2 of the supply voltage Vs. In simple terms, this means that the signal there has two components: An AC component, which is the amplified input signal, plus a DC component of 1/2 Vs = 2.5V. This biased voltage cannot be fed directly to a speaker, but is exactly what we need for a µC sound sensor. The 250uF electrolytic capacitor filters out the DC component and the remaining AC goes to the speaker.
6. Pin 7 is just named bypass, but the datasheet does not provide any further detail on it or its usage.

The 0.05uF capacitor and 10 ohm resistor pair from pin 5 to ground turns out to be called a "Boucherot cell" or "Zobel Network" and is used to prevent high frequency oscillations.

5 Evaluation of solution

5.1 Meeting the success conditions

While some of the success conditions that we set for our system were successfully met, others appeared to be more difficult than we expected them to be. Some were only partly met.

For example, the first condition works – system is able to detect the current sound level. However, it is not able to display the value in decibels, which creates an issue of calibration. We need an actual sound meter in order to see what values we are limiting the output to.

The second condition is fully met – the system does limit the sound when it reaches the sensor value that we can set ourselves. Since this function was basically the main goal we wanted to achieve, it is very good to have it working successfully.

The third condition is met in some way – people can actually set up the time on the system. However, they cannot directly change the value that the system is limiting the output to, although it can be done in the program. It is possible to make it more adjustable, but it requires a further software development.

The system is not quite economically viable for manufacturers, unless it gains a huge success on the market. However, it is not an issue if it is used only on concerts and alike, but not in people's residences.

The system is definitely embeddable in other audio systems, although it would require some work to do that. Namely, we would have to add a second pre-amplifier to the system in order to change the strength of the signal, possibly using a digital potentiometer.

5.2 Unexpected effects

Although our system should not have too many unexpected effects, some things could happen.

For instance, if a sound system is set to a maximum volume in its working hours (23 to 07 o'clock by default), and it is still working after those hours have passed, then the volume will increase substantially, which may be unwanted by the users. However, it should not be an issue for responsible users, who can simply turn the volume down when the system's working hours are gone.

Other than that we have not been able to think about any other unexpected effects that may happen because of the system.

5.3 Evaluation of sources

We have tried to choose the most reliable sources for our project. We selected the books written by reliable authors and Internet articles from trustworthy websites. We also got some useful inputs from professional people working in relevant fields – the sound company managers and owners.

6 Implementation

Before the system is implemented, there are a couple of things which must be changed, and some functions that could be added easily if desired. The most important change is the change from playing music in the Arduino wave shield, which is limited to mono 22 050 Hz 16 bit audio in wav format into any stereo system. As mentioned in the evaluation, this would mean replacing the change of the digital output to a digital potentiometer in a pre-amp in order to ensure compatibility.

It would also be a good idea to replace the microcontroller Arduino DUE with another, more economical model, since the processes involved in the system do not take up as much memory as it has and we do not need that many pins. That being said, a board like the Arduino UNO would be way too small, both capacity wise and concerning the amount of pins available. Finally, a more accurate microphone with a larger frequency range would also be beneficial, although it will increase the cost significantly.

Some of the functions which could be added if desired would be a manual override of the system, to bypass the system no matter the time, and a mute button, which could be redundant in some cases, since most sound systems already have that function.

7 Conclusion

7.1 SWOT Analysis

In conclusion, the system is working very well. But we would need some more time for developing an automated method of calibrating the sensor for each medium. During our research we have found out that the SPL and sound depend on more variables such as air density and acoustics of the measured area.

One weakness that we have discovered is that the limitation of the prototype lies mostly in the limitation of the microphone which has a low sensitivity and also overloads the preamplifier at a small SPL (according to our calculations a maximum of ≈115dB). One other problem is that the small electret microphone was created for near field and not far field measurements. Taking in consideration the prices of nominal sound systems, the price to implement a system such as this for them is quite low. Once the development is done, the system can be replicated with ease, with microcontrollers improving every year.

With future development the system will be able to detect the vibrations created by the resonance of wall in response to the frequency plied to it, thus being able to measure the sound outside the measured room. This will help to calculate the sound level which will be heard by the neighbours and also allow for a better control of the output SPL in order to decrease the disturbance for people who are outside the room in which the music is played.

We do not see any threats for the use of this system in personal and non-personal use. However, the sound controlling is not only based on the electronics part but it will also have to work in concordance with the acoustics of the environment.

7.2 Implementation and concerns

The prototype that we have created for the project is not ready yet to be implemented because of the previously mentioned limitations, but with further development it will be ready for implementation and also ready for use in more locations.

With the decrease of the electronics prices it will also be cheaper to create the system and have it implemented with more ease.

The first thing in the further development process has to be acquiring a better microphone with a more usable sensitivity for the system. Also a piezoelectric crystal sensor can be used to detect the vibrations created in the walls of the room.

7.3 Future work: Hardware

The future work on the hardware will have to be first on the sensor, because it will have to be more sensitive in order to detect the SPL with greater ease. The second part of the hardware future work is to create a preamplifier system that will be able to decrease the volume without adding any noise to the sound signal or without decreasing the sound quality.

As mentioned, this should be done with ease; just more research is needed in acoustics and the propagation of the sound. Once this is done, the system should be able to calculate and control the output so that the SPL will not be a disturbance to surrounding areas.

7.4 Future work: Software

The future work on software will have to be taking into consideration the specific equipment that the system is being implemented into, in order to evaluate which functions are relevant and which additional functions are needed. Also the timer functionality will have to be correctly adjusted for it to be equivalent to 1 minute passing, by both changing the delay and the amount of n variable changes before the system changes the integer. Also, we will have to evaluate the processing time of other functions in the system into the timing factor, in order to make the 4 digit display show the desired brightness.

References

1. 3. 5. 6. 7. 8. Abu Leyan. "DB" And The Unit Of Measure Sound. http://www.abuliyan.com/ (accessed 4.October.2013).

2. Praktisk Elektroakustik Jan Voetmann Eddy Bøgh Brixen 3 version 2011 Musikforlaget
4. D.Osama Abu Rub. Decibel. http://www.aljazeera.net/ (accessed 17/3/2014
9. The Institute of Electrical and Electronics Engineering, New York, 2000; ISBN 0-7381-2601-2; page 288)
10. Fathi Mohammed Meselhi, geography, health and medical, Dar al-Majid for publication and distribution, Cairo, 2008, pp. 161)

11. 12. 13. 14. Mother Nature world. The Harm of Noise Pollution for Human Health. http://deafnotes.com/ubb/ubbthreads/ (accessed 09/11/07)

15. Trent Rock. Understanding Noise Pollution. http://www.conserve-energy-future.com/ (accessed 2014)
16. Nicks J. Noise Pollution Effects on Animals. http://www.buzzle.com/articles/noise-pollution-effects.html (accessed 22.March. 2012)

17. 18. 19.SparkFun Electronics US. Arduino price. https://www.sparkfun.com/categories/103?page=all (accessed 2012)

20. Ross Mathis. *Eco-friendly battery-powered RFID.* http://secureidnews.com/news-item/eco-friendly-battery-powered-rfid/
21. "Noise Ordinance". Kinetics Noise Control, Inc. Retrieved 2012-06-29.
22. Adel Abdul Rahman. Europe decided to reduce the noise . http://classic.aawsat.com/ (accessed 8.February.2013)

23. 25. ARS Electronica. *Digital Communities. http://90.146.8.18/de/archives/prix_archive/prix_year_cat.asp?iProjectID=13638&iCategoryID=12420 (accessed 01.05.2014).* (11/05/14, in German)

24. Air and Radiation. *Noise Pollution. http://www.epa.gov/air/noise.html (accessed 01.05.2014).*10/05/14
26. Soundgearaustralia. *The Govnor SPL2 Limiter. http://www.soundgearaustralia.com.au/soundgear-catalogue/health-and-fitness-clubs/sound-limiters/guvnor (accessed 01.05.2014).*
27. 10 EaZy. Other Users. http://www.10eazy.com/cases/other-users/ (accessed 01.05.2014).

28. Soundgearaustralia. *The Govnor SPL2 Limiter. http://www.soundgearaustralia.com.au/soundgear-catalogue/health-and-fitness-clubs/sound-limiters/guvnor (accessed 01.05.2014).*
29. Math Open Reference. *Sine Waves. http://www.mathopenref.com/trigsinewaves.html (accessed 20-05-2014).*
30. Bøgh Brixen, Eddy & Jan Voetmann: Praktisk elektroakustik 3rd edition, 1st printing, 2011, p. 14

31. Fig. 1 taken from: http://www.slideshare.net/seidadem/basic-theory-of-sound
32. Ballou, Glen M.: Handbook For Sound Engineers, 3rd Edition, Elsevier Inc., 2005, p. 6
33. Ballou, Glen M.: Handbook For Sound Engineers, 3rd Edition, Elsevier Inc., 2005, p. 491

34. Crocker, Malcolm J.,: Handbook of Acoustics, John Wiley & Sons, Inc., 1998, p. 1389
35. Hyperphysics. *Faraday's Law. http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html (accessed 20-05-2014).*
36. Taken from: http://hyperphysics.phy-astr.gsu.edu/hbase/audio/spk.html#c2
37. Ballou, Glen M.: Handbook For Sound Engineers, 3rd Edition, Elsevier Inc., 2005, p. 1318
38. [1]"How many Arduinos are "in the wild?" About 300,000". Adafruit Industries. May 15, 2011. Retrieved 2013-05-26
39. Arduino. Arduino UNO. http:// http://www.arduino.cc/en/Main/ArduinoBoardUno *(accessed 20-05-2014).*
40. Arduino. *Arduino DUE. http://arduino.cc/en/Main/ArduinoBoardDue (accessed 20-05-2014).*
41. Zimmer, Ben (29 July 2010). "How Should 'Microphone' be Abbreviated?". The New York Times. Retrieved 10 September 2010
42. Al-Fajersite. *Types of microphone. http://alfajrsite.net/forum/showthread.php?t=3732 (accessed 31.05.2008)*
43. Tomi Engdahl. *Powering microphones. http://www.epanorama.net/circuits (accessed 2012)*
44. Tomi Engdahl. *Electret Microphone. http://www.intricontibbetts.com/pdf/1070895974.pdf (accessed 02.08.2003).*

45. 46. Stephen R.Fleeman. *Electronic Devices*. Englewood Cliffs, New Jersey 07632: Prentice Hall; 1990.

47. Glen M. Ballou. *Sound Engineers*, Third Edition ed. Elsevier Inc: John Wiley & Sonc, Inc; 2005.

Appendix

Appendix A

Program running on the Arduino UNO board:

```
#include <AF_Wave.h>                    //Input old wave shield library
#include <avr/pgmspace.h>
#include "util.h"                       //Input utilities: ramcheck and systemcheck
#include "wave.h"                       //Input playing capabilities and functions
#include <math.h>                       //Input math fucntions

AF_Wave card;                           //Internal variable for a library
File f;                                 //File name allocation
Wavefile wave;                          // Only one, library file

int avg = 300;                          //Average variable
long sum = 300;                         //Sum of readings variable
long num = 10;                          //Current reading variable
int count = 0;                          //Counter variable
const int SOUND_SENSOR = A0;            //Analogue microphone input pin
uint16_t samplerate;                    //Sample rate: 16-bit (library defined)
int IN = 6 ;                            //Input from other Arduino board

void setup() {
 Serial.begin(9600);        // set up Serial library at 9600 bps
                            //Start all pins as outputs/inputs
 pinMode(2, OUTPUT);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(IN, INPUT);

 if (!card.init_card()) {              //Attempt to inoitialize card
  putstring_nl("Card init. failed!"); return;
 }
 if (!card.open_partition()) {         //Detect partition
  putstring_nl("No partition!"); return;
 }
 if (!card.open_filesys()) {           //File system check
  putstring_nl("Couldn't open filesys"); return;
 }

 if (!card.open_rootdir()) {           //Directory check
  putstring_nl("Couldn't open dir"); return;
 }

 putstring_nl("Files found:");         //Successfully found
 ls();
}
```

```
void ls() {
  char name[13];                    //Name retrieval
  int ret;

  card.reset_dir();                 //Find next
  putstring_nl("Files found:");         //Found other file
  while (1) {
   ret = card.get_next_name_in_dir(name);   //Get next name
   if (!ret) {
      card.reset_dir();
      return;
   }
   Serial.println(name);
  }
}

uint8_t tracknum = 0;                   //Internal track number working or processing

void loop() {
   uint8_t i, r;
   char c, name[15];


   card.reset_dir();
   // scroll through the files in the directory
   for (i=0; i<tracknum+1; i++) {
     r = card.get_next_name_in_dir(name);
     if (!r) {
       // ran out of tracks! start over
       tracknum = 0;
       return;
     }
   }
   putstring("\n\rPlaying "); Serial.print(name);
   // reset the directory so we can find the file
   card.reset_dir();
   playcomplete(name);
   wave.volume = 0;                  //Reset volume
   tracknum++;                    //Get next track

}

void playcomplete(char *name) {
  playfile(name);
  while (wave.isplaying) {              //Library function for checking whether it is playing
    int sensorReading = analogRead(SOUND_SENSOR ); //Retrieval of analogue reading
  if (sensorReading == 0 || sensorReading == 1023){ //Eliminate extreme values
   }
   else{
  Serial.print("                ");
  Serial.println(sensorReading);         //Console debugging tool
```

```
 sum = sum + sensorReading;          //Adding up
 num++;                              //Counter
 avg = sum/num;                      //Average
 Serial.println(avg);                //Console for debugging
 delay(100);                         //Delay for smoothness
 count++;                            //Counter
 if (avg > 450 && digitalRead(IN) == HIGH){  //Defining maximum threshold and condition
  wave.volume++;                     //Decrease volume, 0 = maximum
  Serial.println("VOLUME UPOOOOOOO");    //Console print for debugging
  avg = 300;                         //Reset of variables for consistency and smoothness
  sum = 6000;
  num = 20;
  delay(500);
 }
 if (avg < 210 && digitalRead(IN) == HIGH && wave.volume > 0 ){    //Define minimum threshold and
condition
  Serial.println("VOLUME DOWNMNNNN");    //Console debugging
  wave.volume--;                     //Increase volume, 12 = minimum, 5 = practical minimum
  avg = 300;                         //Reset variable for consistency and smoothness
  sum = 6000;
  num = 20;
  delay(500);
  }
  if (num == 100){                   //Average reset for balance
   num = 10;
   sum = avg*10;
   Serial.print("values reset");     //Console debugging function
   }
 }

 }
 card.close_file(f);                 //End playing file
 }

void playfile(char *name) {
  f = card.open_file(name);
  if (!f) {
    putstring_nl(" Couldn't open file"); return;
  }
  if (!wave.create(f)) {
   putstring_nl(" Not a valid WAV"); return;
  }
  wave.play();                       //Start playing
}
```

Appendix B

Program running on Arduino DUE board

```
#include <IRremote2.h>   //Modified Remote library for compatibility with DUE board
// Remote system variables
int IRpin = 20;        // pin for the IR sensor
int LED = 18;          // LED pin for CLK
int LED2 = 2;          // LED pin for active system
IRrecv irrecv(IRpin);  // Library function for recieving results
decode_results results; // Result storage

// Flow control and miscelaneous variables
boolean CLK = false;   // Initializing clock editing as false, variable for state control
int OUT = 6;           // Output pin from DUE into UNO
int count = 0;         // Counter variable
int dig = 4;           // Current digit being worked on
int dnum = 0;          // Alternative pin position variable
int z = 0;             // 2nd counter variable

// Timer variables
// Cathodes
int a = 35;
int b = 33;
int c = 29;
int d = 31;
int e = 32;
int f = 34;
int g = 28;
int p = 30;

// Anodes (digits)
int d4 = 24;
int d3 = 25;
int d2 = 26;
int d1 = 27;

// Other timer variables
long n = 0;            // Current number (counter)
int del = 2000;        // Clock digit delay
int currentd1 = 0;     // Current digit value
int currentd2 = 0;     // Current digit value
int currentd3 = 0;     // Current digit value
int currentd4 = 0;     // Current digit value

void setup()
{
  pinMode (OUT, OUTPUT);   // Output pin setup
  //Timer pins setup
  pinMode(d1, OUTPUT);
  pinMode(d2, OUTPUT);
```

```
  pinMode(d3, OUTPUT);
  pinMode(d4, OUTPUT);
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(p, OUTPUT);
  // Other pins
  pinMode(13, OUTPUT);
  pinMode(LED2, OUTPUT);
  //Remote setup
  Serial.begin(9600);   // Data rate
  irrecv.enableIRIn();   // Start the receiver
  pinMode(LED, OUTPUT);   // Start LED
  digitalWrite(LED, LOW); // Make LED be off
}

void loop()
{
  if (CLK == false) {         // Check that the user is not editing the clock
  clearLEDs();
  pickDigit(1);
  currentd1 = (n/60000)%6;     // Storing of current value for later use
  pickNumber((n/60000)%6);
  delayMicroseconds(del);     //delay

  clearLEDs();
  pickDigit(2);
  currentd2 = (n/6000)%10;
  pickNumber((n/6000)%10);
  dispDec(2);
  delayMicroseconds(del);

  clearLEDs();
  pickDigit(3);
  currentd3 = (n/1000)%6;
  pickNumber((n/1000)%6);
  delayMicroseconds(del);

  clearLEDs();
  pickDigit(4);
  currentd4 = n/100%10;
  pickNumber(n/100%10);
  delayMicroseconds(del);

  n++;                    // Next value
  if ((currentd1 == 2 && currentd2 == 3) || (currentd2 < 8 && currentd1 == 0)){     // Time threshold for
system to turn on
```

```
 digitalWrite(LED2, HIGH);                  // Light System LED
 digitalWrite(OUT, HIGH);                   // Send command to other board
 }
else{
 digitalWrite(LED2, LOW);
 digitalWrite (OUT, LOW);
 }
if (currentd1 > 1 && currentd2 > 3)         // Time loop/reset at 24:00
{
 n = 1;
}
}
else{                   // If clock editing mode is on
 clearLEDs();
 pickDigit(dig);
 switch(dig){            // check which digit is being edited
 case 4:
 pickNumber(currentd4);        //current value in digit display
 dnum = 24;              // relevant pin
 break;
 case 3:
 pickNumber(currentd3);
 dnum = 25;
 break;
 case 2:
 pickNumber(currentd2);
 dnum = 26;
 break;
 }
 if(dnum == 24 || dnum == 25){     //if its the minute numbers blink individualy
 digitalWrite(dnum, HIGH);
 delay(300);
 digitalWrite(dnum, LOW);
 delay(300);
 }
 else{                 //if its the hour digits, blink both simultaneously
 while(z<600){              //counter for amount of loops
 clearLEDs();
 pickDigit(1);
 pickNumber(currentd1);
 delayMicroseconds(300);

 clearLEDs();
 pickDigit(2);
 pickNumber(currentd2);
 delayMicroseconds(300);
 z++;
 }
 z= 0;
 digitalWrite(26, HIGH);
 digitalWrite(27, HIGH);
```

```
 delay(300);

  }
 }
if (irrecv.decode(&results))
 {
  irrecv.resume();               // Receive the next value from remote
 }
switch(results.value) {            //check which input it is
 case 16736925:                   //mode button
 digitalWrite(LED, HIGH);
 CLK = true;
 delay(100);
 break;
 case 16750695:                    // arrows button
 if(CLK == true) {
 CLK = false;
 digitalWrite(LED, LOW);
 delay(100);
 }
 break;
 case 16712445:                    //digit arrow |<|<|
 if (CLK == true){
 if (dig == 2){                  //loop
  dig = 4;
 }
 else {
  dig = dig -1;
 }
 }
 results.value = 0;               //reset result
 break;
 case 16761405:                    //digit arrow |>|>|
 if (CLK == true){
 if (dig == 4){                  //loop
  dig = 2;
 }
 else {
  dig = dig +1;
 }
 }
 results.value = 0;
 break;
 case 16754775:                    // - button
  if (CLK == true){
  switch(dig){
   case 4:                       //relevant digit
   if(currentd4 > 0){
   n = n-100;                    //amount of n to change digit 4
   currentd4 = currentd4-1;
   }
```

```
        else{
          n = n+900;
          currentd4 = 9;
          }
        results.value = 0;
        break;
        case 3:
        if(currentd3 > 0){
        n = n-1000;
        currentd3= currentd3 -1;
        }
        else{
          n = n + 5000;
          currentd3 = 5;
          }
        results.value = 0;
        break;
        case 2:
        if (currentd2 > 0){
          n= n-6000;
          currentd2 = currentd2 -1;
          }
        else{
          if(currentd1 > 0){
            n = n-6000;
            currentd2 = 9;
            currentd1 = currentd1 -1;
            }
           else{
          n = n+23*6000;
          currentd1=2;
          currentd2=3;
          }

          results.value = 0;
          break;
          }
    }
    results.value = 0;
    break;
    }
    case 16748655:                    // + button
    if (CLK == true){
      switch(dig){
        case 4:
        if(currentd4 < 9){
        n = n+100;
        currentd4 = currentd4+1;
        }
        else{
          n = n-900;
```

```
    currentd4 = 0;
    }
   results.value = 0;
   break;
   case 3:
   if(currentd3 < 5){
   n = n+1000;
   currentd3= currentd3 +1;
   }
   else{
    n = n - 5000;
    currentd3 = 0;
    }
   results.value = 0;
   break;
   case 2:
   if ((currentd2 < 9 && currentd1<2)||(currentd1==2 && currentd2<3)){
   n = n+6000;
   currentd2 = currentd2 + 1;
   }
   else{
    if (currentd2 == 9){
     n = n + 6000;
     currentd1 = currentd1 +1;
     currentd2 = 0;
     }
    else{
     n = n - 23 * 6000;
     currentd1 = 0;
     currentd2 = 0;
     }
     results.value = 0;
     break;
    }
  }}
 results.value = 0;
 break;
 case 16738455: //0 button
 if (CLK == true){
  switch (dig){
   case 4:
   n = n- 100*currentd4;
   currentd4 = 0;
   break;
   case 3:
   n = n - 1000*currentd3;
   currentd3 = 0;
   break;
   case 2:
   if (count == 0){
    n = n - 60000*currentd1;
```

```
      currentd1 = 0;
      count = 1;
      results.value = 0;
      }
    else{
      n = n -6000*currentd2;
      currentd2 = 0;
      count = 0;
      }
    break;
    }
}
results.value=0;
break;

case 16724175://1 button
if (CLK == true){
  switch (dig){
    case 4:
    n = n- 100*currentd4 + 100;
    currentd4 = 1;
    break;
    case 3:
    n = n - 1000*currentd3 +1000;
    currentd3 = 1;
    break;
    case 2:
    if (count == 0){
      n = n - 60000*currentd1 + 60000;
      currentd1 = 1;
      count = 1;
      results.value = 0;
      }
    else{
      n = n -6000*currentd2 +6000;
      currentd2 = 1;
      count = 0;
      }
    break;
    }
}
results.value=0;
break;
case 16718055://2 button
if (CLK == true){
  switch (dig){
    case 4:
    n = n- 100*currentd4 +200;
    currentd4 = 2;
    break;
    case 3:
```

```
      n = n - 1000*currentd3 + 2000;
      currentd3 = 2;
      break;
      case 2:
      if (count == 0 && currentd2 < 4){
       n = n - 60000*currentd1 + 120000;
       currentd1 = 2;
       count = 1;
       results.value = 0;
       }
      else{
       n = n -6000*currentd2 +12000;
       currentd2 = 2;
       count = 0;
       }
      break;
      }
  }
  results.value = 0;
  break;
  case 16743045: //3 button
  if (CLK == true){
    switch (dig){
      case 4:
      n = n- 100*currentd4 +300;
      currentd4 = 3;
      break;
      case 3:
      n = n - 1000*currentd3 + 3000;
      currentd3 = 3;
      break;
      case 2:
      n = n -6000*currentd2 + 18000;
      currentd2 = 3;
      count = 0;
      break;
      }
  }
  results.value = 0;
  break;
  case 16716015 : //4 button
  if (CLK == true){
    switch (dig){
      case 4:
      n = n- 100*currentd4 +400;
      currentd4 = 4;
      break;
      case 3:
      n = n - 1000*currentd3 + 4000;
      currentd3 = 4;
      break;
```

```
    case 2:
    if(currentd1 <2){
    n = n -6000*currentd2 + 24000;
    currentd2 = 4;
    count = 0;
    break;
    }
    }
  }
  results.value=0;
  break;
  case 16726215 : //5 button
  if (CLK == true){
   switch (dig){
    case 4:
    n = n- 100*currentd4 +500;
    currentd4 = 5;
    break;
    case 3:
    n = n - 1000*currentd3 + 5000;
    currentd3 = 5;
    break;
    case 2:
    if(currentd1 <2){
    n = n -6000*currentd2 + 30000;
    currentd2 = 5;
    count = 0;
    break;
    }
    }
  }
  results.value=0;
  break;
  case 16734885 : //6 button
  if (CLK == true){
   switch (dig){
    case 4:
    n = n- 100*currentd4 +600;
    currentd4 = 6;
    break;
    case 2:
    if(currentd1 <2){
    n = n -6000*currentd2 + 36000;
    currentd2 = 6;
    count = 0;
    break;
    }
    }
  }
  results.value=0;
  break;
```

```
case 16728765 : //7 button
if (CLK == true){
 switch (dig){
  case 4:
  n = n- 100*currentd4 +700;
  currentd4 = 7;
  break;
  case 2:
  if(currentd1 <2){
  n = n -6000*currentd2 + 42000;
  currentd2 = 7;
  count = 0;
  break;
  }
  }
}
results.value=0;
break;
case 16730805 : //8 button
if (CLK == true){
 switch (dig){
  case 4:
  n = n- 100*currentd4 +800;
  currentd4 = 8;
  break;
  case 2:
  if(currentd1 <2){
  n = n -6000*currentd2 + 48000;
  currentd2 = 8;
  count = 0;
  break;
  }
  }
}
results.value=0;
break;
case 16732845 : //9 button
if (CLK == true){
 switch (dig){
  case 4:
  n = n- 100*currentd4 +900;
  currentd4 = 9;
  break;
  case 2:
  if(currentd1 <2){
  n = n -6000*currentd2 + 54000;
  currentd2 = 9;
  count = 0;
  break;
  }
  }
```

```
    }
    results.value = 0;
    break;
  } //button select closer
} //void loop closer

void pickDigit(int x)
{
  digitalWrite(d1, HIGH);
  digitalWrite(d2, HIGH);
  digitalWrite(d3, HIGH);
  digitalWrite(d4, HIGH);

  switch(x)
  {
  case 1:
    digitalWrite(d1, LOW);
    break;
  case 2:
    digitalWrite(d2, LOW);
    break;
  case 3:
    digitalWrite(d3, LOW);
    break;
  default:
    digitalWrite(d4, LOW);
    break;
  }
}

void pickNumber(int x)
{
  switch(x)
  {
  default:
    zero();
    break;
  case 1:
    one();
    break;
  case 2:
    two();
    break;
  case 3:
    three();
    break;
  case 4:
    four();
    break;
  case 5:
    five();
```

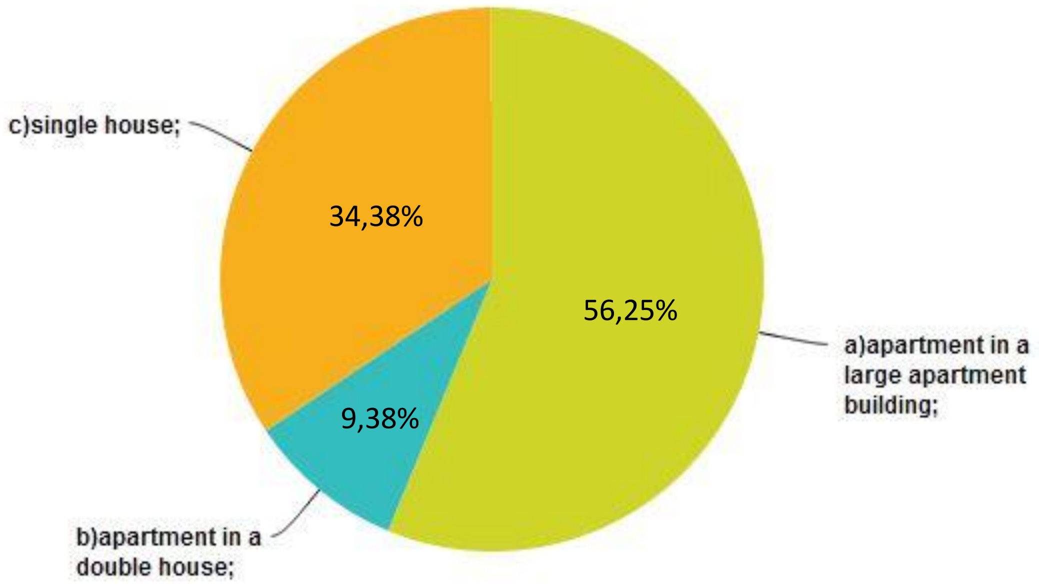
```
    break;
  case 6:
    six();
    break;
  case 7:
    seven();
    break;
  case 8:
    eight();
    break;
  case 9:
    nine();
    break;
  }
}

void dispDec(int x)
{
  digitalWrite(p, HIGH);
}

void clearLEDs()
{
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  digitalWrite(p, LOW);
}

void zero()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, LOW);
}

void one()
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
```

```
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}

void two()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}

void three()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}

void four()
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void five()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void six()
{
  digitalWrite(a, HIGH);
```

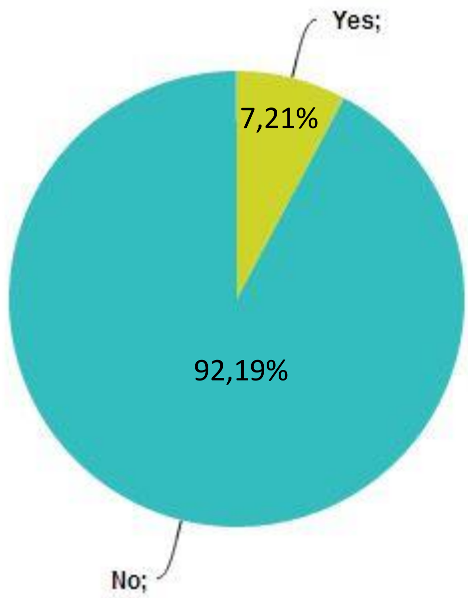
```
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void seven()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}

void eight()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void nine()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}
```

Appendix C

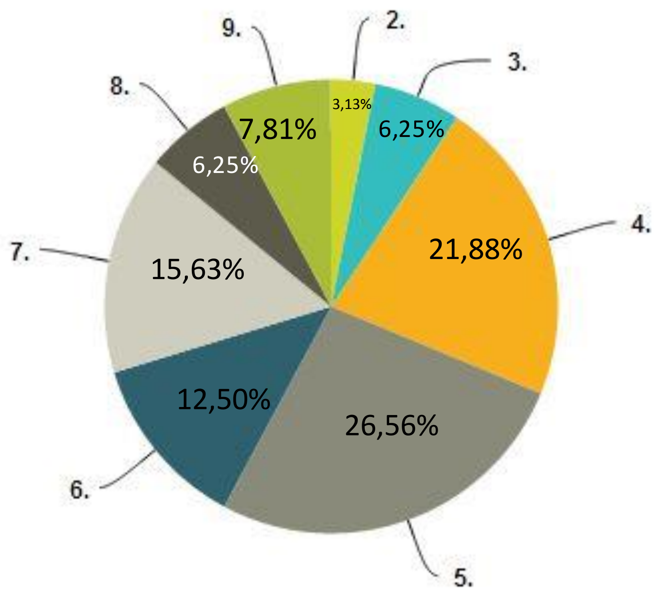
Question 1. In what kind of home do you live?



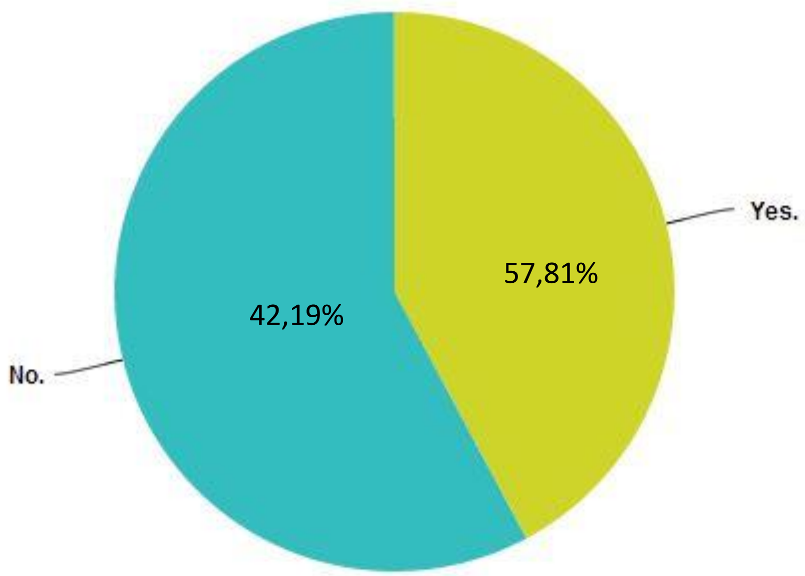
Question 2. Are you having any problems hearing in normal conditions?



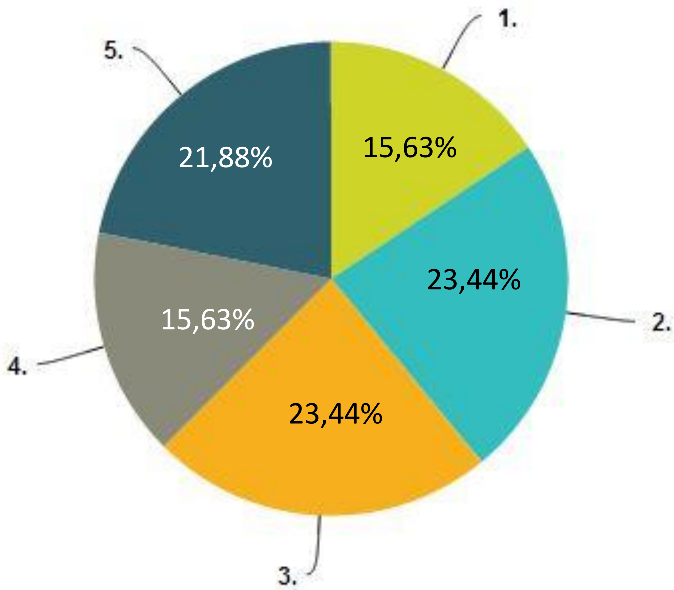
Question 3. On a scale from 1 to 10 how loud do you say that you listen to music? (1-low volume; 10-extremely loud volume)



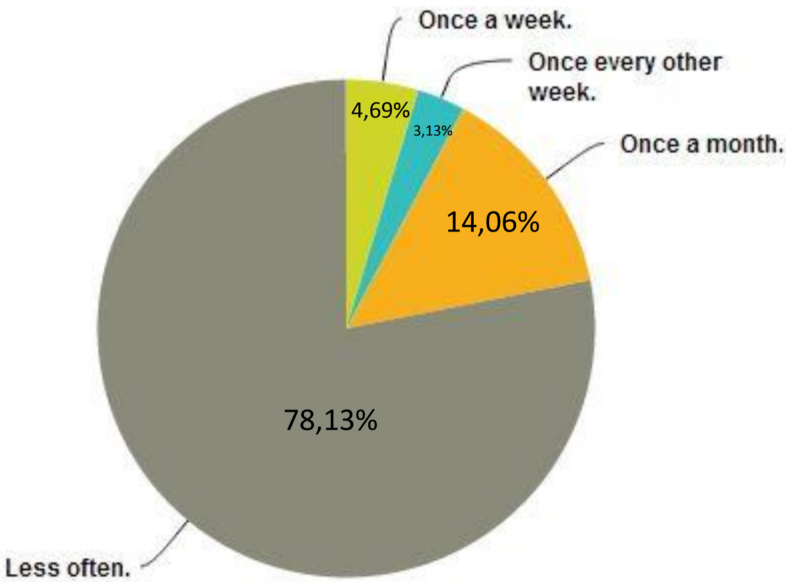
Question 4. Have you ever had problems sleeping because of loud parties close by?



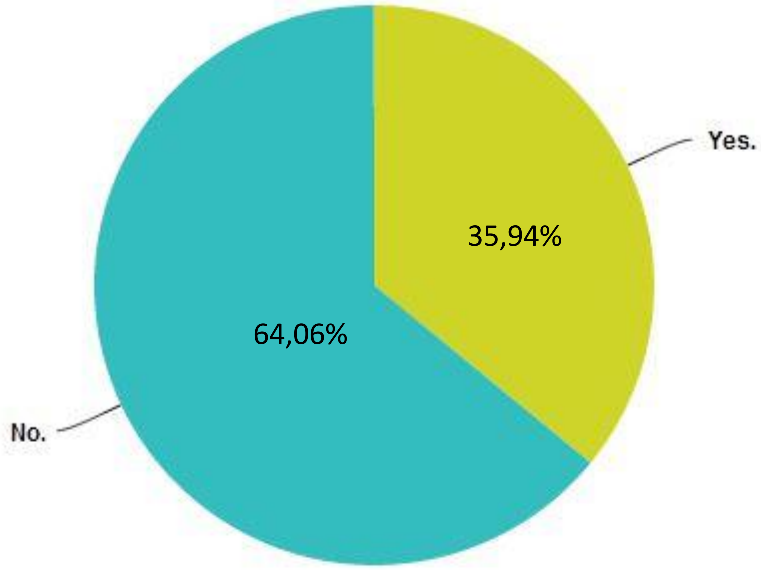
Question 5. On a scale from 1 to 5 how much would you like people to stop listening to loud music after certain hours? (1-I don't care; 5-I would like it to be impossible)



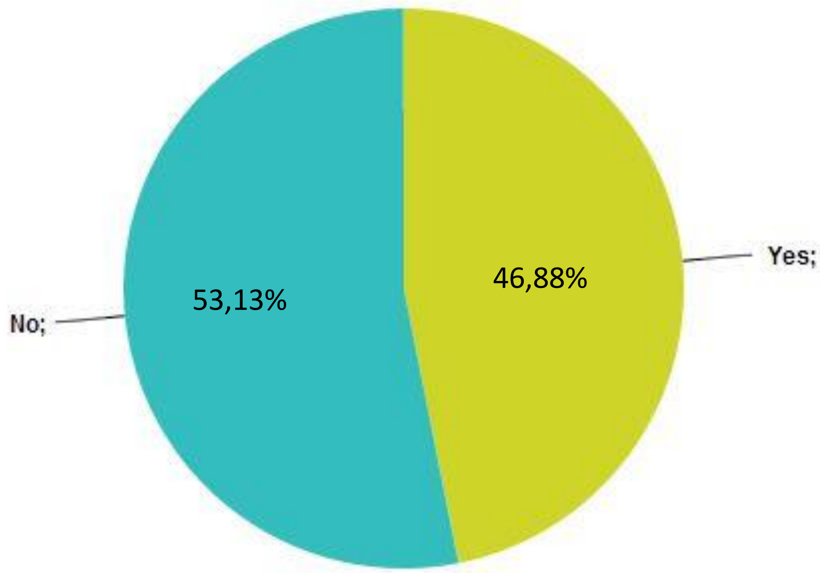
Question 6. How often do you have problems sleeping because of loud music from neighbors?



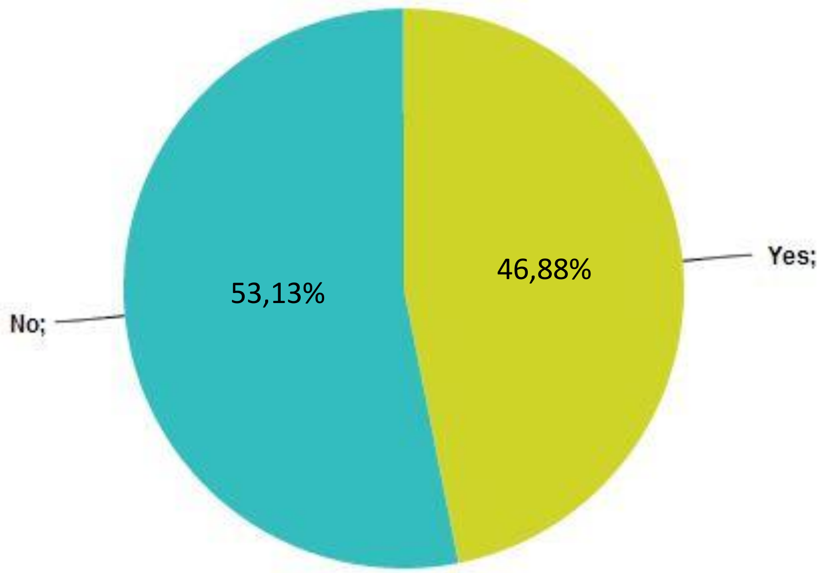
Question 7. After the last concert you went to, did you have any hearing impediments due to the loud music?



Question 8. Would you like for the sound systems to automatically control the sound level in order to stop the sound to becoming noise?



Question 9. Did you ever get complaints about listening to music to loud?



Question 10. On a scale from 1 to 10 how loud was the volume of the last outdoor concert you went to? (1- low volume; 10-extremely high volume)