AALBORG UNIVERSITY COPENHAGEN

MIO effect processor

Sound and Music computing

Supervisor: Daniel Overholt

Jelizaveta Zovnercuka, Kristians Konovalovs

25.06.2019

Table of contents

1. Introduction

Music is one of the most popular ways of self-expression. It has also gained very strong positions on a global market. While the revenue from the legal sales is falling, the popularity of the live shows has shown a growth. Only in UK the revenue from the live shows in 2014 was equal to 3.8 billion pounds, and music tourism in three years increased by 34% (Brown S. C., & Knox D., 2017). With such increasing popularity, the quality of the live shows is also growing. Now, in the digital era, it is possible to use backing tracks and instrumentals to support the show, as well as use effect pedals and amplifiers to transmit high-quality sound from the stage to the farthest part of the venue (Baxter-Moore N., & Kitts T. M.б 2016).

The expansion of this market led to massive development of different instruments, tools, digital technologies etc. Some of them have been forgotten, the others have gathered certain popularity and evolved. One of the shining examples is guitar and its variation – electric guitar, which gained huge popularity starting from 1960’s with 300,000 guitars sold per year and reaching 700,000 guitars seles in 2004 (Millard A, 2004). Due to its popularity, it has gained large attention on the market as well, ending up in multiple companies producing their own guitars with their own specific sounds. Not only does the sound of guitars from various manufacturers differ, also various effects allow altering sounds to match every specific need of guitar players. The opportunity to adjust the sound of the guitar led to the necessity to carry all the effects along for the show.

Heavy and numerous, the effect stompbox pedals have to be located on one place on the stage and left immovable. Any of the wires, connecting multiple pedals, disconnected during the performance, may lead to the absence of sound. Moreover, guitar players need to stand at the location of the pedals, to be able to switch on or off necessary effects at the right time.

With advantages such as stability, opportunity to adjust settings and intuitive design, the discomfort created by the stompbox effect pedals led to their development in the digital domain. At first, analog effects were significantly better than digital ones, since the digital signal lacked resolution for a proper sound recreation (Gilreath P., Aikin J., et al., 2004). However, with the development of the technologies, the quality of the digital signal became almost equal to the analog one (Wei L., & Moyer M. G. (Eds.), 2009). This opened completely new possibilities of making music and playing live shows. These possibilities combined with classic designs of popular music equipment could lead to the innovative redesigns of already existing products. As outlined by the Moore’s law, the amount of transistors in microprocessors doubles every 18 months, and this law has been true for the past 50 years (Stettler M., & Krishnapura S., 2016), which leads to the enormous increase of computational power of processors. Thus, the digital domain could become an alternative to analog effect pedals, also helping make the device less heavy and reduce all unnecessary wires.

All of the above leads to the **initial problem statement**:

How can we provide guitar players with the same level of functionality as the stompbox effect pedals, at the same time improving the mobility issues?

2. Analysis

2.1. Digital and analog domain comparison

Moore's law started in 1965, when he noticed that every year the amount of components in a chip was "roughly" doubling, numbers growing from 0 in 1959 to 64 components in 1965 (Mack C. A., 2011). Based on this, he predicted that there will be the same exponential growth of components in a chip for the next decade (Mack C. A., 2011). Even though the trend was keeping on, in 1975 Moore predicted a slowdown from components doubling each year, in reality keeping their growth each 18 months (Mack C. A., 2011). While the amount of transistors in a memory chip was constantly "roughly" doubling, the development of microprocessors being able to include such amount of transistors was much slower. The growth of amount of transistors in memory chips in 2000 was equal to 1.58x per year, e.g. 1 billion transistors in a chip, while the growth of transistors in microprocessors was equal to 1.38X per year, e.g. microprocessors having only 20-40 million transistors (Mack C. A., 2011). Moore's law not only covers the number of components, e.g. their size, but also their price, them becoming cheaper (Mack C. A., 2011). It means that more powerful devices can be developed for the lower price, what makes them more available on the market.

Digital domain has some issues compared to analog design, such as delay caused by processing, however, over the years these issues have been constantly improving. For example, nowadays the processing time in complex algorithms such as hearing aids may vary from only 1ms to 10ms (Frye G. J., 2001). It has been suggested that digital music devices should keep their latency below 10ms, as it is not perceivable by human ear, and this suggestion has become a standard up until now (McPherson, A. P. et al., 2016). Therefore, the latency less than 10ms in the digital domain caused by the processing is acceptable and may not be considered as a flaw.

When keeping in mind all of the above, it is obvious, that digital domain nowadays is almost as good as analog. Moreover, digital domain has some advantages compared to analog domain. It is much more comfortable in use, since it is much easier to build digital algorithm, than create an analog circuit. The same applies to the quality of the effects, since there are a lot of digital pre-made effect libraries, free to use (vst4free, 2019) (Free VST Plugins, 2019), which can also be updated or improved on the go. This makes process of building a music device much faster and easier compared to analog domain, where lots of wires, resistors and other sensors have to be attached. The costs are also much lower, while the functionality is increased (Mack C. A., 2011). Digital domain also allows more versatile change of the sound, e.g. multiple effects can be easier mixed, even modulators can be used, which help achieving the sound of any other instrument, while playing on the guitar (Jam Origin, 2019). Digital domain has wider dynamic range, since the best analog systems can only achieve around 80dB highest, while 16bit digital system can have around 90dB dynamic range (Fries B., & Fries, M., 2005). Making copies is also much easier with the use of digital domain, since it is fast and noise-free, while for analog domain it takes equivalent amount of time as the length of the audio file. Moreover, with every copy loss of 3dB of signal-to-noise ratio happens (Fries B., & Fries, M., 2005). Multiple analog copies lead to deterioration of the sound file, which cannot be improved or corrected, while digital domain contains error-correction codes that can automatically fill in the bits that were damaged or missing, thus, fixing the sound file (Fries B., & Fries, M., 2005). Finally, the digital storage is more durable than analog media storages, such as tapes or vinyls, and, therefore, keeps the original quality of the audiofile (Fries B., & Fries, M., 2005).

All of the above leads to the conclusion that digital domain can provide equally good sound quality as analog domain and, keeping in mind all of the advantages, it is meaningful to use digital domain for the alternative effect pedal design implementation.

2.2. Introduction to DSP

To be able to efficiently use digital domain for the alternative effect pedal design implementation, it is important to dig into basics of digital signal processing (DSP), to make sure that maximum sound quality is achieved with the minimum errors and quality loss.

Sound is a change of pressure in the air (Avison J., 2014). To be able to convert it from analog to digital domain, the air pressure has to be transferred into electric current, which can be understood by the computer. This is done by A/D converters, which are nowadays built in any digital electronic device using microphone (Hosken D., 2014). A/D converters basically have two steps - sampling and quantization (Reed J. H., 2002). Sampling is a process of recreating the signal by the discrete, evenly spaced points in time. Sample rate is the amount of samples in 1 second. If sample rate is too small, e.g. there is too small amount of points in a period, the loss of data may happen, which leads to wrong representation of the signal (Figure 1). To make sure that no aliasing happens, Nyquist theorem should be considered. It states that sample rate should be twice as fast as the highest frequency present in the signal, which, according to psychoacoustics, is equal to 22050 Hz (Park T. H., 2009). Some of the most popular sample rates that are currently used are 44,1kHz for CDs, 48kHz (or more) for professional audio, 32kHz for TV and Radio and 8kHz for the telephone (Morris T., & Tomasi C., 2017).

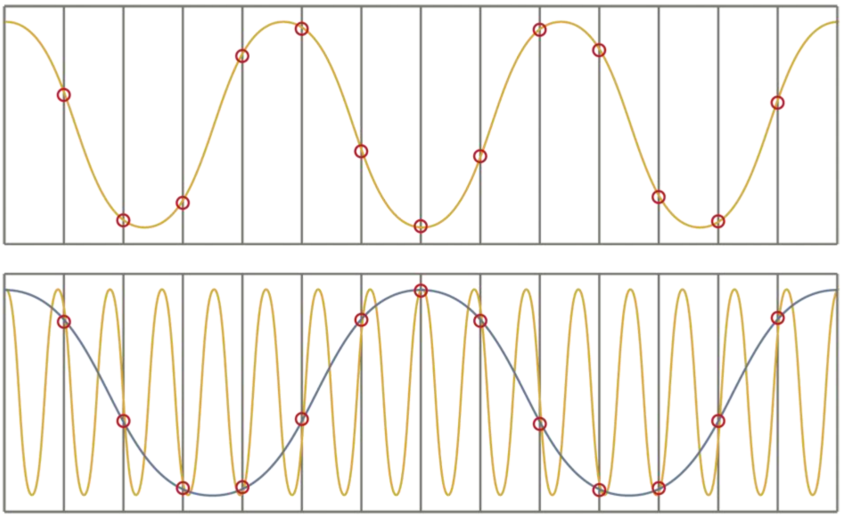

Figure 1. Example of aliasing

The second step in the A/D conversion is called quantization, which represents the resolution of the signal (Park T. H., 2009). It converts analogue voltage of a signal in a specific point of time to the binary number, e.g. bit. The higher resolution, the higher binary number, what means that the higher amplitude of the signal can be represented (n bits/sample) and the more precise signal can be recreated (Sauls S., & Stark C., 2016). However, with the increase of resolution, increases also the storage space required to store the signal (bitrate = fs*number of bits* number of channels) (Gan W. S., & Kuo S. M., 2007). When the bitrate is too small, there is a chance of quantizing error, e.g. when the signal falls between two digital “steps” and has to be represented by the closest value, what makes changes to the original waveform of a signal (Figure 2). In music quantisation error is usually heard as a quantization noise (Gan W. S., & Kuo S. M., 2007). Therefore, it is important to keep the resolution high enough, which for the CDs is equal to at least 16 bit (Millward S., 2003). CD bitrate would be equal to 44100*16*2, e.g. $1.411*10^6$ bit/sec (Fries B., & Fries M., 2005), which is a pretty high number. This is the reason why music is usually compressed and only high quality formats like .flac or .wav have higher bitrate with lots of storage space required.

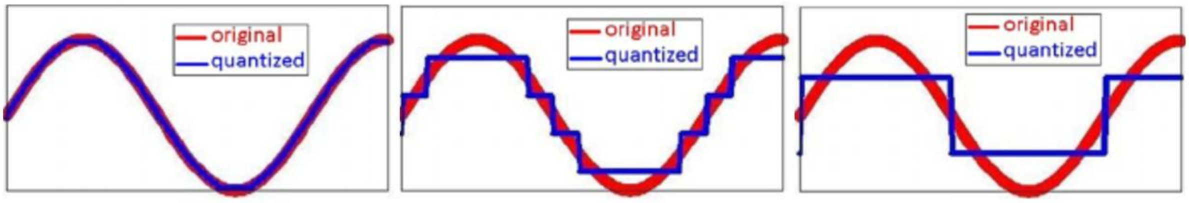

Figure 2. Quantization and quantization error

2.3. Effects

In order to be able to build up different effects it is important to gain knowledge about their nature and how these effects can be recreated in digital domain.

First of all it is important to understand the difference between frequency domain and time domain. Frequency domain describes frequency spectrum in one block of samples (Park T. H., 2009). It shows every frequency present in the signal in that specific block. Time domain, however, shows changes in frequency over the time (Park T. H., 2009). We can divide effects accordingly as frequency-based effects and time-based effects. The main difference is the principle of altering the sound, e.g. altering frequency components of the signal or its phase.

There are several effects, which have exactly the same principle, e.g. phase shift, such as echo, chorus, flanger, reverb, delay etc. The same signal is shifted in phase over some small period of time. The strength of the phase shift, e.g. delay time, determines the effect. Flanger has the smallest delay time, which is under 10ms and can be described with an equation:

$$y[n] = x[n] + x[n - g[n]]$$

where g[n]] is the delay time integer (Park T. H., 2009). It enhances specific frequencies and reduce others (Nonzee, V., & Poongbunkor, P., 2001). A block diagram of flanger effect can be seen in Figure 3.

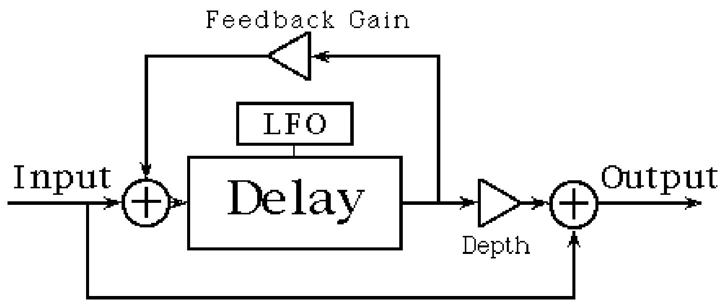

Figure 3. Block diagram of flanger

Chorus is very similar effect to flanger. They even share the same equation (Park T. H., 2009). The only difference is the delay time, which for the chorus lies in the range from 10 to 25ms (Park T. H., 2009). The name of the chorus effect is self-explanatory and reminds of multiple people singing the same line. One more time-based effect with the same principle is called echo with the longest delay time among before mentioned effects, e.g. 10-50ms (Park T. H., 2009). Echo is the effect we perceive in the cathedrals or large halls with signal being reflected from the surfaces and attenuated, with every reflection losing energy. Echo can be described with the following equation:

$$y[n] = x[n] + bN \cdot x[n - N]$$

Where bN is the coefficient of signal attenuation (Park T. H., 2009). Sometimes echo is called reverb. With the delay time exceeding the above mentioned limits, it becomes possible to distinguish between two signals. Block diagram of chorus can be seen on Figure 4. When this happens, it becomes a new separate effect called delay.

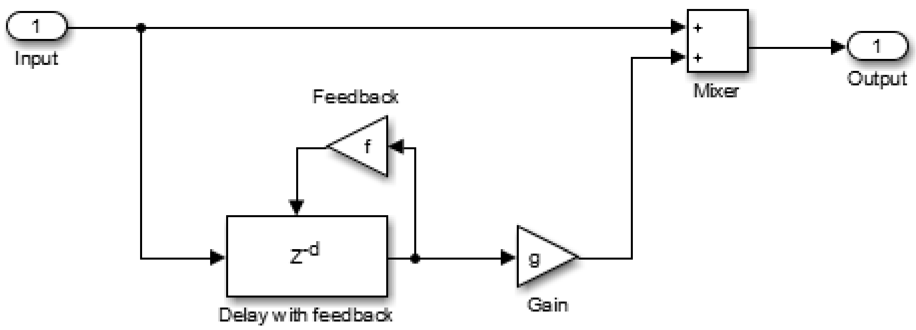

Figure 4. Block diagram of echo

When dealing with phase shift, it is important to keep in mind constructive and deconstructive interference. Constructive interference can be explained as a sum of two identical signals with the same phase and the same amplitude, which leads to the same signal with larger amplitude (Park T. H., 2009) (Figure 5.a). Deconstructive interference is the phenomenon behind the noise cancelling, which can be achieved by summing up two soundwaves with the same amplitude, but with the phase being reversed (called antiphase (Padhye R., & Nayak R. (Eds.)., 2016)) (Figure 5.b).

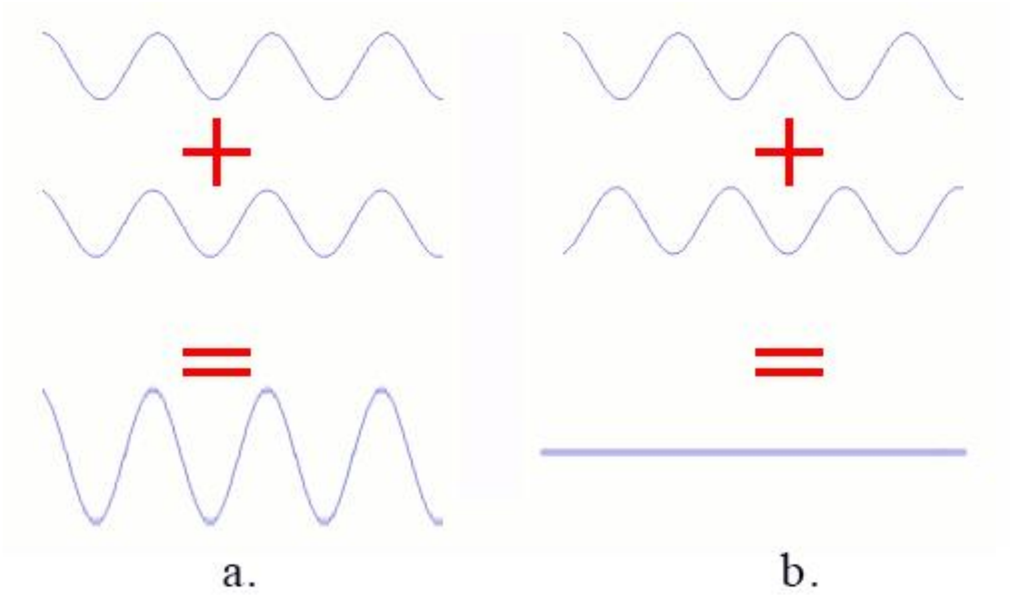

Figure 5. Constructive and deconstructive interference

Not only in noise cancelling can constructive and destructive interference be observed, but also in beating. Noise is being cancelled when signals match perfectly. However, when imperfections are present and the lengths of the sine waves differ slightly, the beating occurs. The resulting frequency is average frequency of two signals (Park T. H., 2009). This can become the basics behind modulations, for example, such as ring modulation (Park T. H., 2009).

Some of the effects are based on the simple filters, such as low-pass, high-pass or band-pass filters. Low-pass filter reduces high frequencies and only passes the low ones, as opposed to the high-pass filter (Fant-Saez G., 2006). Band-pass filter passes only specific frequencies (Bazil E., 2009). It is a bell-shaped curve that is determined by the center frequency and bandwidth. Band-pass filter is used in wah-wah effect, what creates the perception of the signal being increased in pitch by moving center frequency from lower to higher frequencies (Nonzee, V., & Poongbunkor, P., 2001). The effect allows controlling the band-pass filter over the time.

The waveform and its amplitude can also play an important role in the effect development. The sine wave is the natural shape of the wave. Pure tone that resembles one frequency is a single sine wave, and complex sound is a combination of multiple sine waves (Park T. H., 2009). Sometimes, the amplitude of the wave is too high to be handled properly, which is caused by too high amount of voltage for the capacitors (or other components) to be stored. When this happens, peaks of the sine wave are “cut”. This process is called clipping (Self, D. et al., 2009). There are two types of clipping - soft and hard. Soft clipping has the smoother wave shape, while hard clipping is changing the waveform so that it reminds of a square wave (Self, D. et al., 2009). When clipping occurs, the signal is being distorted. The more clipping happens, the more distortion is present in the sound. This is the basic principle behind the overdrive and distortion effects, first observed in tube amplifiers and then recreated in the digital domain by tweaking amplitude of a signal (Christophersen M., et al., 2007). Distortion became one of the most popular effects in music industry and is actively used in heavier music genres (Hunter, D., 2014).

2.4. Importance of visuals during the performance

Live music was always important part of our lives, but, perhaps, now it has reached its apogee (Baxter-Moore, N., & Kitts, T. M., 2016). Even though ticket prices for the live shows are constantly growing, the amount of people attending them is also increasing (Brown, S. C., & Knox, D., 2017). This tendency can be seen not only for the younger generations, but for people of all ages (Statista, 2018). There are multiple possible explanations for the people to be willing to attend expensive live shows. Some of the scientists admit that except for listening to music or supporting musicians, people may be willing to be there as a part of something special, to experience it with those, who share their minds (Brown, S. C., & Knox, D., 2017). This may lead to the feeling of unity. The others admit that another reason for people willing to attend live shows can be explained with the specific pleasure of capturing the moment, which happens only now and cannot be extended (Danielsen, A., & Helseth, I., 2016).

Not only the audio makes live shows special, but also the possibility to see performers with his/her own eyes. It is considered that the expressivity of the performance can be enhanced with the visual presence of the musicians on the stage (Danielsen, A., & Helseth, I., 2016). Seeing performers is important part of the concert, as they can interact with a crowd. It is considered that exactly visual part of the live show can help communicating with the auditory (Danielsen, A., & Helseth, I., 2016), which may be important for the immersion. Some scientists even found that the more static musicians are during the show, the more bored audience feels (Hammond M., Rabinowitz K., & Alldis D., 2009). It all leads to the conclusion that not only audio, but also visual aspects of the performance are important for the perception of the live concert, and a lack of the mobility on the stage may negatively influence the perception of the performance.

2.5. Microcontrollers / Portable computers

It is important to study what kind of microcontrollers and portable computers are available in order to be able to create a better prototype.

2.5.1. Microcontrollers

The most widely known microcontrollers are Arduino products. There are many different Arduino boards, and they all are great. Probably the biggest advantage is fast prototyping, since Arduino also produces different shields that allow connecting sensors straight to the board without any soldering. The number of digital and analog pins varies depending on product. For instance, Arduino Mega offers 54 digital and 16 analog I/O pins. Arduino probably has the best support in terms of different sensors and actuators compatibility. It also releases arduino IDE (Integrated Development Environment), which makes writing and uploading the code very simple and user-friendly. However the processing speed can be an issue. The most widely used board, Arduino Uno (Figure 6), has a clock speed of 16Mhz, which definitely might be not enough for some projects.

Figure 6. Arduino Uno microcontroller

NodeMCU (Figure 7) is another microcontroller which is based on ESP8266 Wi-Fi module. It is small, thin and cheap. It also offers 15 digital and 1 analog I/O pins. In comparison to Arduino boards, it is super-fast as well – it offers either 80Mhz or 160Mhz clock speed. Fortunately it is also possible to use Arduino IDE for the code development.

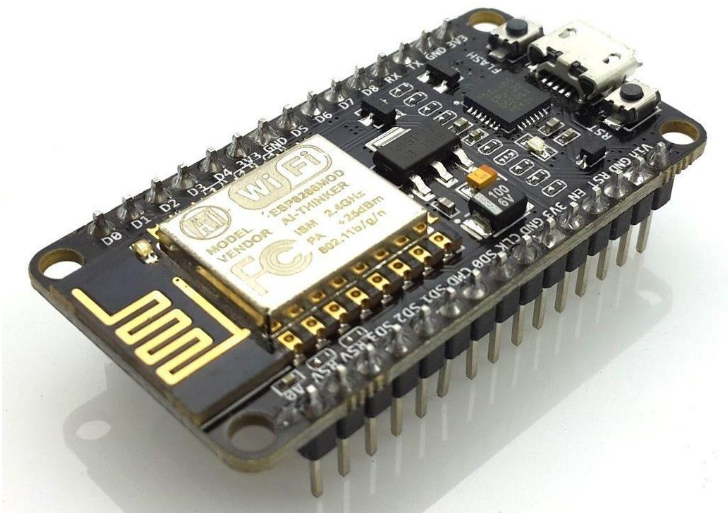

Figure 7. NodeMCU microcontroller

Teensy 3.5 board (Figure 8) is packed with different features. It runs at speed of 120Mhz, which is quite impressive. It comes with 25 analog inputs, 2 analog outputs, 62 digital I/O, SD card port, Ethernet port and 4 I2C ports.

Figure 8. Teensy 3.5 microcontroller

2.5.2. Portable computers

For this project, a portable computer was required for alternative design implementation. The main requirement was that it should be capable of running either any Linux distribution or Windows to be the proper development platform, and also be powerful enough to run an audio patch without significant latency and glitches.

One of the most popular portable computers is Raspberry Pi. The latest Raspberry Pi 3 (Figure 9) runs an optimized version of Debian called Raspbian, and it has 1 GB of Ram and 4 core CPU that runs @ 1.2Ghz. SD card is used for the storage space. This device has impressive connectivity, since it has 4 USB 2.0 ports, HDMI and Ethernet connectors. Finally, it comes with built-in Wi-Fi and Bluetooth modules. All these features make Raspberry Pi 3 an awesome development platform, especially for the price (Raspberry Pi, 2019).

Figure 9. Raspberry Pi 3 portable computer

Beaglebone (Figure 10) is another board in the same price range as Raspberry Pi 3. It comes with 512 MB of RAM and ARM Cortex-A8 processor that runs @ 1Ghz. Connectivity-wise it comes with HDMI, 2 USB ports and Ethernet. As an advantage, it comes with 4 GB of internal storage. It runs vanilla Debian, Android, Ubuntu and others (Adafruit, 2019).

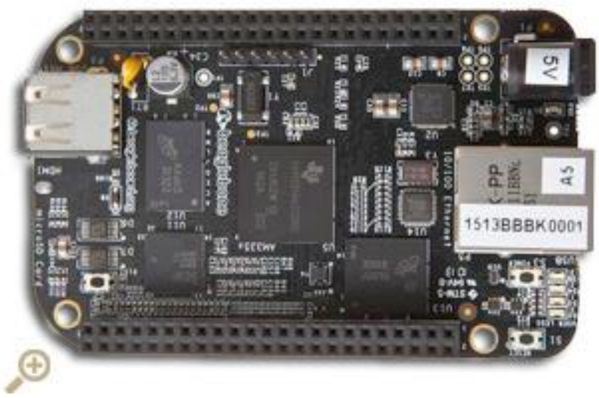

Figure 10. Beaglebone portable computer

Another, a little bit more expensive option is Asus Tinker Board (Figure 11), which costs almost twice as much as a single Raspberry Pi 3, but it also brings a lot of advantages. For instance it comes with 2 GB of ram and Cortex A-17 processor that runs @ incredible 1.8Ghz. It has 4 USB ports and HDMI. Wi-Fi and Bluetooth modules are included as well. It runs Debian-based operating system called TinkerOS. Price - approx. 60$ (Kurve A., 2017).

Figure 11. Asus Tinker Board portable computer

The last but not least, Odroid-C2 (Figure 12) is a small portable computer that offers great specifications for the price. It runs quad-core CPU @ 1.5Ghz and has 2 GB of RAM. Equipped with 4 USB ports, HDMI and Ethernet. It doesn't come with Wi-Fi or Bluetooth though. Price - approx. 66$ (Kurve A., 2017).

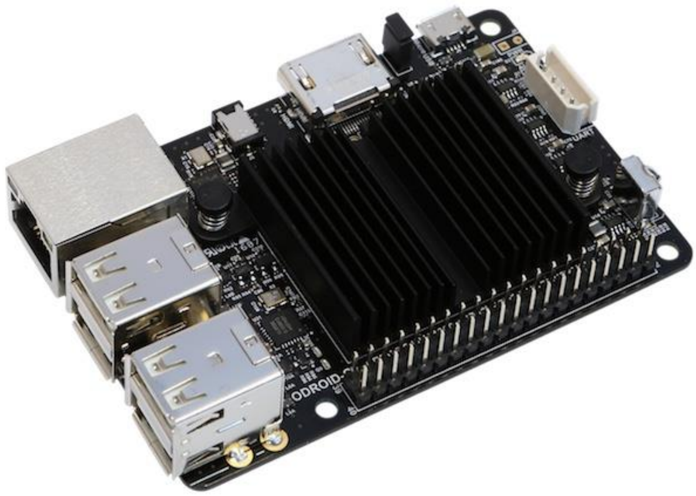

Figure 12. Odroid-C2 portable computer

For our project we have chosen to go with NodeMCU and Raspberry Pi 3 due to their low price, very high connectivity and portability. Both of them have built-in Wi-Fi module, which makes the connection between them very simple.

2.6. Wi-Fi vs Bluetooth

Wi-Fi and Bluetooth are both technologies for wireless data transmitting, and they have been around for quite a while now. Each of them has its own pros and cons, however they share the same principle - using radio waves for data transmission.

Wi-Fi has quite large bandwidth (up to 25mbps), and it is capable of sending large chunks of data in a matter of seconds. Since bluetooth was designed for sending small chunks of data between two mobile devices, its bandwidth is much narrower (around 800kbps) (Adepoju, A., 2018). Speaking of distance, Bluetooth working range is around 30 meters, whereas Wi-Fi is capable of sending data over quite large distances (up to 100 meters in an open space) (Sattel, S., 2016).

Wi-Fi is considered to be more secure due to its more complex infrastructure and encoding. However, since the signal is quite strong, it is more likely that there will be interference. Bluetooth, on the other hand, sends much weaker radio waves, which decreases the range, but makes the interference much less likely to happen. Moreover, Bluetooth can change the working channel more than 1000 times per second, which means that even if there is another Bluetooth device nearby, they will not interfere (Sattel, S., 2016).

Finally, when it comes to power consumption, Bluetooth is a winner. Since it transmits much weaker radio waves, the power consumption is reduced to a minimum. The simplicity of infrastructure makes it an advantage as well (Adepoju, A., 2018) (Sattel, S., 2016).

To sum up, Bluetooth seems like an incredible technology when one needs to send small chunks of data over a small distance with lower power consumption. However it was decided that Wi-Fi is more suitable for this project, mainly because it provides significantly better range.

2.7. Design principles

For a long time technology-centered design was considered to be the main way to design systems (Endsley, M. R., 2016). What is distinctive for the technology-centered design is that, first, the necessary functions are built, and only then the way how to interact with these functions is created. This may end up in having huge amount of unnecessary buttons, knobs, screens, which leads to the increase of the workload of the user, as well as waste of time and mental resources to perform the task (Endsley, M. R., 2016). The problem lies in the way people are able to process information. People are not able to concentrate on the large amounts of information at a time (Endsley, M. R., 2016). The incorrect and hard to use design can result in the mistakes, called *design-induced error* (Endsley, M. R., 2016).

As opposed to technology-centered design, user-centered design is built around the needs and capabilities of people. The information is structured in a way to efficiently fulfil desired goals, thus, minimizing design-induced errors (Endsley, M. R., 2016).

There are specific rules and guidelines for the good user-centered design. It is important that users are in control of the situation. When creating the design, it is important to remember, that it is a user who will be using the design, not the design that will do everything for the user (Endsley, M. R., 2016). A good user-centered design should help achieving goals, not achieve them. A digital camera can control focus or lightning, but should not automatically make pictures (Lowdermilk, T., 2013).

One possible way of reaching this goal could be by using predictive design, e.g. keeping the well-known functions as expected (Shneiderman, B., 2003). If there is an already existing well-known sharing function, there is no need to develop brand new sharing function that would create confusion for the new user. This can help keeping the design intuitive.

When designing brand new devices, one of the easiest ways to make a good user-friendly and intuitive design is to gather information from users about their needs and preferences. This can be done by surveys, interviews, A/B tests (if on the web) or case studies (if physical device is being tested), task analysis, e.g. conducting task-based experiment, or heuristic evaluation, where no rules or guidelines for the experiment have previously been set up (Lowdermilk, T., 2013). For testing applications on the early stages low-fidelity prototypes, such as paper-test or Wizard of Oz, can be used, but even before that a storyboard can help visually designating the interface (Lowdermilk, T., 2013).

Feedback, such as visual, auditory and haptic, play important roles in good user-centered design. The more feedback is implemented in the design, e.g. all three senses used, the higher are chances of building an intuitive and easy-to-use device (Erlhoff M., & Marshall T. (Eds.), 2007). The reasoning behind this lies in the fact that many sensors (for example, push button) return back to their original position after being pressed. Therefore, it is easy to forget the current state of the device. The user experience of using button could be improved by adding sound, such as click, when pressing the button, as well as a small LED next to the button, to help user understand, whether the button was actually pressed, whether the effect is active etc. However, both auditory and visual feedbacks have their limitations. In case when user is blind or deaf, sounds or LED’s can be not enough, therefore, a haptic feedback should be used (Erlhoff M., & Marshall T. (Eds.), 2007).

To conclude, good user-centered design should be based on user experience and expectations, it should be predictive and can be supported by the information, gathered from user testings. The design should include multiple types of feedback, such as visual, haptic etc. All of the above is an essential key to the user-friendly and easy-to-use design.

2.8. State Of The Art

In order to make a great product, it is important to study existing solutions within the similar area of interest. In case of this project we are going to take a look on different products that try to push the boundaries of traditional guitar effect management and bring brand new features to the market. We are not only keen on radically new ways of controlling the effects, but we are also interested in new interfaces that tend to optimize the utilization of guitar effects in general, making it more powerful, versatile, user-friendly and portable.

2.8.1. Advantages and disadvantages of effect pedals

With the increasing popularity of electric guitars consequently grows the scope of use of effect pedals. Even with the creation and development of Profiling Amplifiers and the increased average price of the effect pedals from 45 to 79 dollars per unit (Statista, 2019a), the popularity of the effect stompbox pedals (Figure 13) is not falling. In 2017 amount of pedals sold reached 1300000 units per year (Statista, 2019b). It means that there is something in the design and usability of effect pedals that makes them appealing for the musicians. Some of the most popular effect pedals are wah-wah (Saunders A. J., 2013), reverb (Orkin D., 2018), delay (Orkin D., 2018), looper (Orkin D., 2018), overdrive (Orkin D., 2018) etc.

Figure 13. Boss distortion stompbox effect pedal

Stompbox effect pedals are foot operated devices, thus, usually located on the floor. Basic effect pedals consist of one footswitch, up to three potentiometers for adjusting the parameters of effect and LED for the visual feedback (Gadol III, W. N., 2016). The footswitch also provides very clear haptic feedback, which resembles a click, produced by the mechanics of the system. Moreover, footswitch is very reliable and guarantees activation and deactivation of the effect. Potentiometers allow precisely adjusting parameters of the effect, however, for every song parameters have to be adjusted specifically, which takes some additional time during the performance, creates pauses and breaks between the songs. Since pedals are located on the floor, they can be associated with mobility issues during the performance.

10 experienced guitar players were asked to fill in the questionnaire that covered their experience of using effect pedals. 5 out of 10 respondents agreed that stompbox effect pedals fully satisfy their needs (M = 4.3, SD = 0.8) (Figure 14).

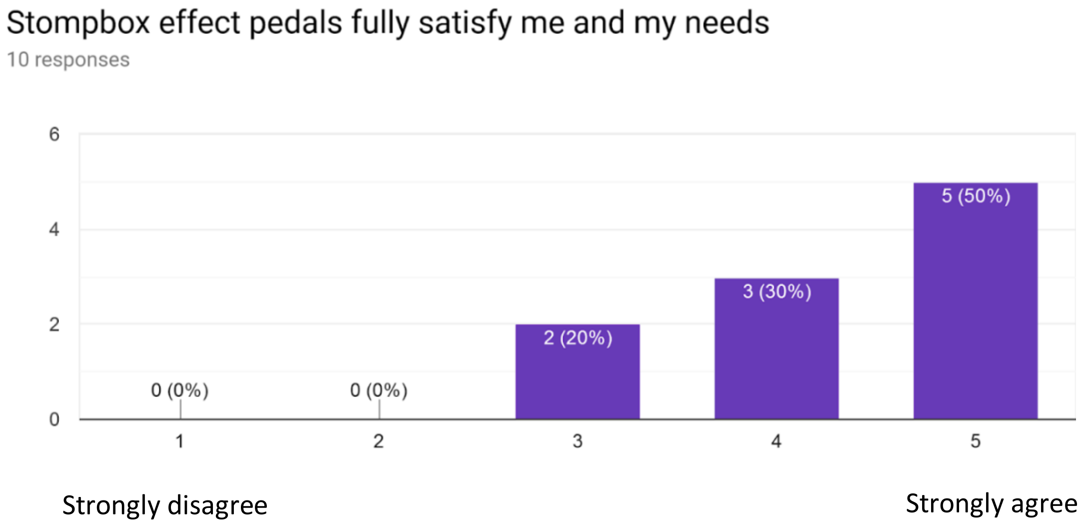

Figure 14. "Stompbox effect pedals fully satisfy me and my needs" results

However, a possible disadvantage was related to the analog domain. Since every effect requires a separate pedal, musicians usually end up having large pedalboards with multiple stompbox effect pedals, connected with wires. If at least one wire gets disconnected, the whole system stops working, and no sound is being produced, which may completely ruin the performance. Only 3 out of 10 participants have never experienced any technical issues caused by the effect pedals during the show, the other respondent answers varied significantly (M = 3.6, SD = 1.3). When digging deeper, it appeared that 8 out of 10 respondents with varying regularity have experienced a wire disconnecting from a pedal during their performance (M = 3, SD = 1.3). The opinion regarding the mobility issues during the show caused by the effect pedals differed significantly, covering all possible variants of the response (M = 3.1, SD = 1.5). 5 out of 10 respondents agreed with the lack of mobility caused by the effect pedals, 4 out of 10 disagreed with that statement and 1 respondent provided neutral results (Figure 15).

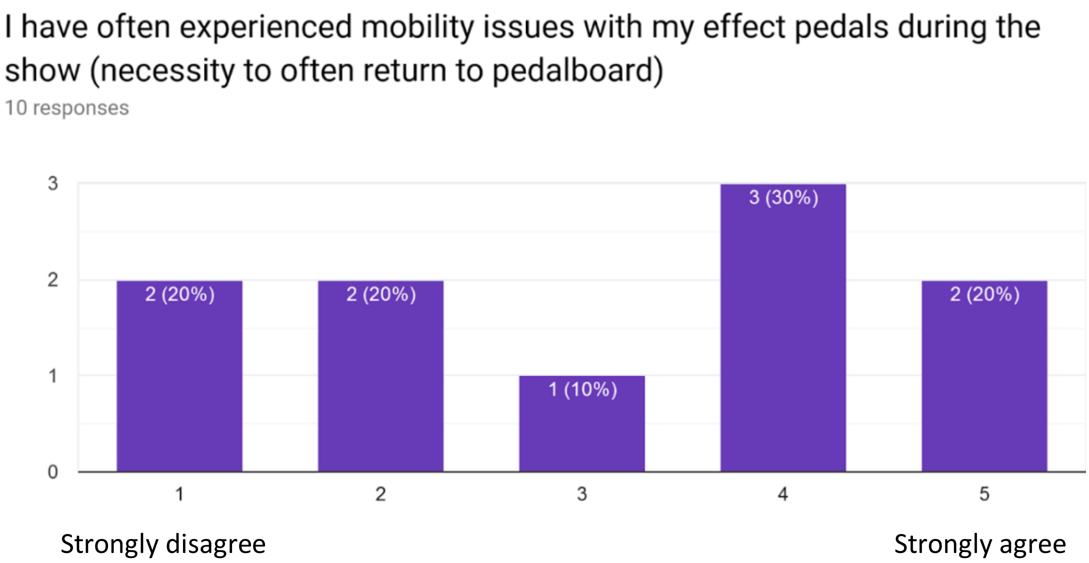

Figure 15. "I have often experienced mobility issues with my effect pedals during the show" results

This could be explained with the frequency of use of the effect pedals, since some of music styles do not require active use of effects. Neutral results were also gathered when trying to find out whether there are preferences of using analog or digital devices (M = 3.1, SD = 1.1), meaning that digital effects can be as much appreciated as the analog ones. The next step was to find out comfort level provided by the stompbox effect pedals when adjusting the effect parameters. 9 out of 10 respondents strongly agreed with the usefulness of adjusting settings at any place on the stage (M = 4.9, SD = 0.3) (Figure 16). This can be explained with the time consuming process of returning back to the effect pedals, sitting down and adjusting settings. An alternative design, removing those disadvantages, could be very meaningful and useful for the guitar players.

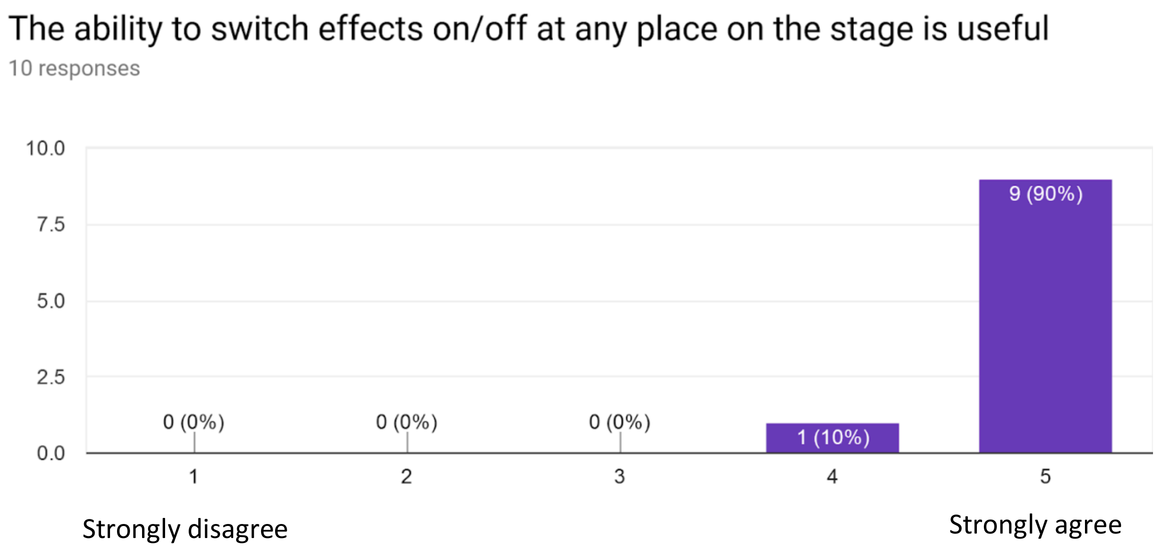

Figure 16. "The ability to switch effects on/off at any place on the stage is useful" results

The last question was related to the alternatives of controlling the effects on the stage. It was open question with an ability to write and explain respondent's thoughts on this case. Some of the respondents would prefer foot-activated effects, but not constrained to a certain place on a stage. The others would like to have a wireless control over the effects. Couple of respondents would like to avoid the necessity of setting up the effects themselves either with the use of a sound technician or presets. All of those responses could be combined in one design.

To sum up, stompbox effect pedals have both some great advantages and disadvantages. Effect pedals are extremely reliable foot activated devices that provide visual and haptic feedback. All of these advantages should be kept in the final design. At the same time, mobility should be improved, wires should be eliminated and the improved effect setting control and adjustment should be created.

2.8.2. Ibanez + KORG Kaoss / Guitar Wing / ACPAD

Essentially Ibanez + KORG Kaoss pad combination (Figure 17) is regular electric guitar that has a small effect processor attached to it. The guitar is plugged straight into the Kaoss pad, and the sound gets filtered by it. The Kaoss pad consists of a small touch screen, touch slider, 4 buttons and a tiny LCD screen. The effect library consists of 100 different time or frequency based effects, be it various low-pass of high-pass filters, different delays, flangers etc. The effects are controlled by touching a small touch screen, allowing to access different parameters of the effect simultaneously. Slider lets the user to scroll through the effects and choose one, which is very handy feature. LCD screen helps navigating through the effect library.

Figure 17. Ibanez guitar + KORG Kaoss pad

Another similar product is Guitar Wing (Figure 18). It is a MIDI controller which is possible to attach to an electric guitar or bass guitar. It features two sliders, five pressure-sensitive buttons, motion sensors and a number of regular buttons that help interacting with the device. As any other MIDI controller, it is designed to be used in conjunction with either effect processor that supports MIDI input, or with Digital Audio Workstation (DAW), where user maps the controller's sensors and buttons to different parameters of VST plugins or other effects. The device turned out quite compact, and it is supposed to be attached to the lower wing of the guitar, making it easy to access the controls and not worsening the playability in any way. What is quite important, the device is wireless, and it sends MIDI messages through Bluetooth, which means that the user can map all the controls before the show and then access them being anywhere on the stage.

Figure 18. Guitar Wing

Similar features are provided by another device called ACPAD (Figure 19). However, this device is designed for acoustic guitars. It is large, but ultra-thin plate that is attached to an acoustic guitar's front side and is equipped with large amount of pressure-sensitive buttons and sliders. It is also wireless and can be connected to a PC or tablet. If the Guitar Wing was more effect controller with high degree of freedom, ACPAD is rather a combination of effect controller and sampler. Thanks to 8 sensitive pads, it can easily be used as a sampler. However these pads are sensitive enough to be used as effect controller as well, so it all depends on what kind of configuration a user requires. Moreover, it is equipped with two touch sliders that also can be mapped to some effect parameters, be it frequency filters, modulators, pitch shifters etc. Finally, it has 10 buttons that allow the user to store up to 25 configuration presets.

Figure 19. ACPAD

2.8.3. Line 6 + Helix

Another perspective of a problem is not only how you control the effects, but also how and in what kind of form you maintain them. As it was mentioned earlier, it all started with stompbox effect pedals. Then it evolved into effect processors, which provided much more features and customization possibilities. However there are couple of products that pushed the boundaries even further, providing some unique functionality.

Figure 20. Variax technology and Helix by Line 6

Variax technology (Figure 20), produced by Line 6, is a piece of gear that is installed on an electric or bass guitar, and it allows manipulating the sound in various ways. First, it allows changing the tunings of the guitar just by turning the knob. This feature is enabled by integrating a separate piezo pickup into each string saddle. Then, DSP algorithm transposes up or down each string individually depending on the tuning. It is possible to store up to 11 different tunings and access them by just turning the knob on the guitar. It is possible to transpose each string one octave up or one octave down maximum. However the magic doesn’t end here. It is also possible to change the timbre and “type” of the instrument. For example, it is possible to choose between Les Paul, Stratocaster, Semi-Hollow and other types of guitar bodies (which all give different timbers), or to choose sitar or banjo. All processing happens by on-board electronics. Finally, Line 6 has released a guitar processor which is called Helix (Figure 21), and it is compatible with Variax technology. It stores tons of different guitar effects, and in combination with Variax, the versatility of the setup is extremely high. It allows creating presets that store not only the stack of the effects, but also the tuning of the guitar and the timbre. That means that it is possible to switch from brutal distorted electric guitar with super low tuning to a little banjo by just pressing a button on Helix. The tuning and guitar tone is configured by using Workbench HD software, and all the presets are managed by setting up Helix itself.

Figure 21. Helix guitar processor by Line 6

2.8.4. iRig Stomp I/O

iRig Stomp I/O (Figure 22) is the product of IK Multimedia, and at first glance it is just another effect processor. And, in fact, it is, but the biggest difference is that the whole processing is happening through the iPhone or iPad attached to it. This fact makes a big change in simplicity and user friendliness of the setup. Big touch screen of iPad makes it incredibly simple to make custom presets, manage effects etc., which dramatically changes the whole user experience. It also provides great visual feedback, as the main information is displayed on the screen. Finally, the device itself has four stomp buttons for accessing presets and expression pedal, which is possible to assign to different effect parameters.

Figure 22. iRig Stomp I/O

2.8.5. Hot hand

Hot Hand (Figure 23) is a very simple device, which potential is limited only by imagination of the user. It is a simple MIDI controller that is possible to attach ether to hand (or theoretically to any other part of the body), or to a musical instrument. Hot hand detects the change of its position in space (accelerometer), and it is possible to map this change to any effect that supports MIDI input (the most common use case is to map it to an effect in the DAW). It is possible to either use it as expression tool (continuous change), or as effect activator/deactivator. The last one is particularly interesting to us as it essentially allows switching effect on/off being anywhere on the stage. Usually for switching effect on and off the Hot Hand is attached to the guitar neck, and it needs to be shacked up in order to switch effect's state. This proves that there are alternatives for controlling effects without stompbox effect pedals lying on the floor.

Figure 23. Hot Hand

2.8.6. Keith McMillian's Soft Step

This little device is a MIDI controller that is designed to be controlled while standing, e.g. controlled by feet. The fact that Keith McMillian's Soft Step (Figure 24) is floor-based MIDI controller makes it already quite interesting, because in combination with laptop and proper VST plugin it can be much more versatile than the majority of effect processors. However each of this device's buttons are pressure sensitive, which makes it possible not only to switch effects on/off, but also continuously control some of the effect's parameters (like expression pedal). Moreover, each button has two axes, which makes it even more expressive.

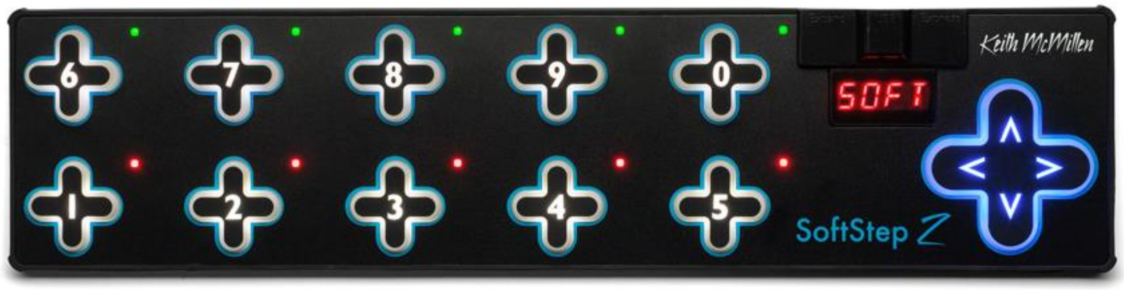

Figure 24. Keith McMillian's Soft Step MIDI controlled

2.8.7. Conclusion

All above mentioned products proved that there always exists a room for improvement, and that is especially the case with guitar effects. Many companies are trying to make the usage of the effects more user-friendly, affordable, versatile and expressive, one way or another. The high amount of great product that came out recently only contributes to the fact that the usage of effects can be optimized even more, and greater products are yet to come. One of the most relevant design elements from ones described above was the Hot Hand, and especially its ability to activate or deactivate effects with a specific activation movement. Also, the way of setting up the configuration was brilliantly implemented in iRig Stomp I/O design. The opportunity to make a set up using the touch screen makes the design user-friendly, intuitive, comfortable and fast. The key elements of these designs should be taken into consideration when designing the alternative effect device.

2.9. Target Group

To be able to come up with the good design for the device, it is important to know target group, which will be using this device. Since we are building redesign of the effect pedals, it is obvious that the main target group is concentrated around musicians and, specifically, those, who use effect pedals, e.g. play either electric or bass guitar. To narrow down potential users of the device, it is important to make sure that guitar and bass players are active users of the effect pedals. Moreover, in order to use this device properly, it is necessary to be a decent and skilled musician with experience in live performances, using stompbox effect pedals. This would make sure that issues related to original stompbox effect pedals are familiar, which would allow more objective comparison of the redesigned device. The age, sex or genre in which musicians perform does not matter.

2.10. Design requirements

When all necessary information has been gathered and analysed, it was important to create a list of requirements for the design and implementation. The list should be based on the analysis and include all the advantages of the state of the art, at the same time excluding the issues and disadvantages.

Design:

- Foot-controlled
- Wireless
- Reliable
- Small
- Guitar/body mounted
- Provide both visual and haptic feedback
- Possibility to adjust settings at any place without the necessity to sit down

Audio:

- Good-quality digital effects
- Effect library to choose necessary effect from
- Presets for the effect settings

Implementation:

- Raspberry Pi, NodeMCU, LCD screen, LEDs, vibration motor, potentiometers
- Wi-Fi for wireless connection
- Real-time (<10ms latency)

The above design requirements lead to the **final problem statement**:

How can we create an effect processor for guitar/bass players that would be competitive with classical stompbox effect pedal design, provide the same level of experience, meanwhile increasing mobility on stage?

3. Design

To be able to answer final problem statement, it is necessary to create a design that would solve all the issues of the classic effect pedal design. The design has to be supported with the list of requirements.

3.1. Initial design

3.1.1. Iteration I

First redesign of the stompbox effect pedals was guitar-mounted device. It consisted of nine potentiometers - three for each effect and three pressure sensitive resistors for activating and deactivating the effects. An LED was placed under each button to provide visual feedback (Figure 25). The case studies carried out on 30 guitar players revealed two main disadvantages of this device. First, it was a hand-activated device, what distracted guitar players from the performance by forcing them for a moment to stop playing to activate or deactivate effects. Second, it was very hard to adjust settings. There were nine potentiometers placed in one line, three per effect. Even though there were visual markers describing what is the parameter for which effect, it was impossible to read without tilting down.

Figure 25. Iteration I of an alternative effect device

3.1.2. Iteration II

The second redesign was focused on keeping the strong parts of the original stompbox effect pedals and at the same time reducing their disadvantages. Since first design proved hand control being hard and uncomfortable to use, it was decided to shift back to the original foot operated design, which has proved itself to be easy to use and reliable. However, it has been decided to only keep the movement itself, without any floor based devices. This could reduce the mobility issues and allow activating effects at any time and place. To make this possible, the activation movement was required. It would allow freely moving on the stage and then, when required, perform activation movement, that would give possibility to choose necessary effects. There were three possible effects to choose from - distortion, delay and reverb. The effects were chosen by their popularity. Three effects were related to three leg positions - left, middle and right. To make sure that the effect was switched on or off, visual and haptic feedback was implemented.

To improve the issues with the settings, it has been decided to move the setting device to the side part of the guitar, to make it clearly visible. At the same time it would eliminate the necessity to sit down to adjust the settings. The place on the guitar was chosen basing on the usability testing, where three options were provided, however, due to the fact that it has been decided to add the LCD screen, only one place on the side of the guitar could be used (Figure 26).

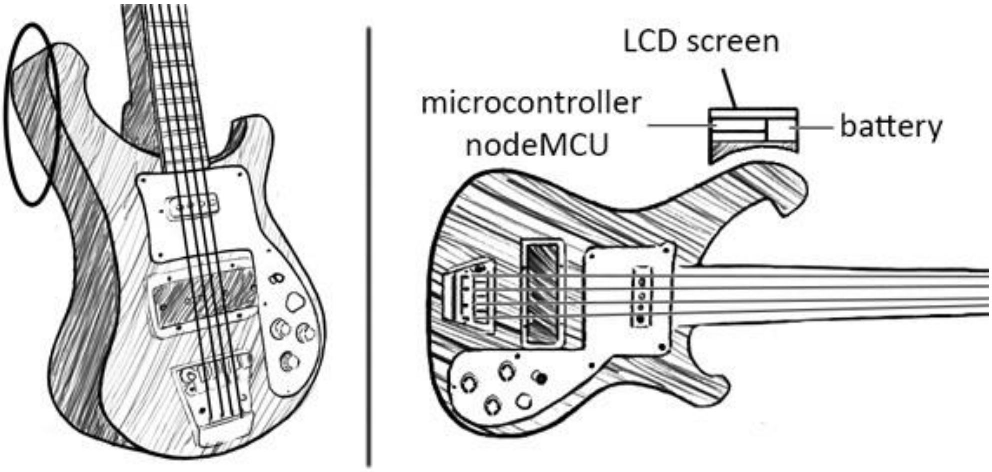

Figure 26. Iteration II: the placement of the setting device

According to the opinion of 30 test participants the comfort of this design was overall rated positively, it’s mean being equal to 4.1 out of 5, compared to 3.8 for the original effect pedal design. The reason for such high results might be the new way of adjusting the effects. Classic effect pedals require musicians to sit down to adjust the settings, while redesign eliminated that necessity. However, in the interviews after the testing some of the test participants admitted that setting device could still could be improved.

3.2. Final design

The final design is based on the previous iterations. It is supposed to take all the advantages from the previous iterations and classic defect pedals design and at the same time combine them in one multifunctional device that would improve the strengths and eliminate weaknesses of original and initial designs.

3.2.1. Concept

The device should be focused on keeping the same experience as from the classic effect pedals, while providing additional mobility during the performance. This can be achieved by removing the physical pedal that usually lays down on a floor immovable. Since previous designs showed hands being impossible to be used for the activation or deactivation of the effects, the controller should still be foot activated. Thus, it has been decided to keep the movement for switching effects on/off, but remove physical pedals. To switch effects on or off it is required to “press” an invisible button.

The maximum amount of the effects is limited to three due to the precision issues - bigger amount of the effects would lead to inaccuracy and difficulties in choosing the right effect. Thus, choosing the effect happens by turning toe left or right or keeping it in the middle. When choosing the effect, three different LEDs help navigating through the effects. Three different colours were used, one for each effect. When the effect was chosen and invisible button was “pressed”, a haptic feedback, e.g. the vibration, was performed.

3.2.2. Activation movement

To make sure that the effects are not activated or deactivated accidentally, the special movement for activating the whole system was chosen. It is called inversion and can be explained as tilting foot to the side (Figure 27). This activation movement was implemented and tested on the iteration II of the design and proved itself to be successful. The main principle behind this activation movement was to make sure that it cannot be made accidentally. It has, however, one disadvantage, related to the type of shoes user is wearing. Rough, stiff shoes cannot be used together with this movement, which, for the purpose of this paper, is acceptable.

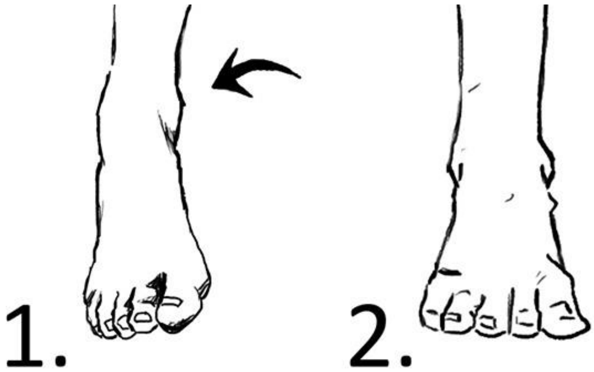

Figure 27. The activation movement called inversion

3.2.3. Setting device

Not only stompbox effect pedals allow activation and deactivation of the effects, but also the adjustment of the settings for each effect. In the classic effect pedal design the adjustment of the settings is very simple and user-friendly, since there is physical device that can have potentiometers, switches, knobs etc. However, in our case we have to create a separate design that will handle this task. There were three general ideas about the way of implementing setting device.

3.2.3.1. Floor-based setting device

This idea is a clone of the original design. Since it is only the activation and deactivation of the effects during the songs that reduce the mobility, in theory, setting device design could be kept without changes, especially since settings are usually adjusted before or in between the songs. This leads to the idea of creating a floor-based setting device.

It would have the same functions and design as a stompbox effect pedals, e.g. three potentiometers for the parameters of the effect, LED for the visual feedback, toggle switch to choose between type or timbre of the effect etc. Since we are combining multiple effects in one device, an LCD screen should be added. The device should be wireless to avoid any chances of breaking the system by pulling off wire during the show.

3.2.3.2. Guitar-mounted setting device

Since previous tests showed that it was uncomfortable to sit down to adjust the settings, the alternative to floor-based design could be guitar-mounted device, e.g., carrying the setting device with oneself during the whole show. The device should be small, to make sure that it is not disturbing musicians during their performance. Except from that, the hardware could be equal the floor-based design.

Another aspect of this design is location of the setting device on the guitar. Tests from the iteration II of the design showed the possibility of using the side part of the guitar, since the device located there is not interfering with the performance. The only issue was the size of the device. To make sure that it is simple and comfortable to use, the device should be relatively flat. This would reduce chances of accidentally rotating potentiometers, which was present in the design of iteration II. Device should also be wireless to exclude potential risks due to the pulled off wire.

3.2.3.3. Mobile app

Third possible design solution for the comfortable setting adjustments was creating a mobile application that would run on any mobile device. The idea of using applications in music industry is not new. There are multiple DAWs that provide such function to their users, such as Studio One (Google Play, 2019), Logic Pro (App Store, 2019), Abletone live (help.ableton.com, 2019) etc. The advantage of this design is that it does not require any additional physical device, which makes it easier during travelling and transportation.

3.2.4. Pilot test - Setting device

3.2.4.1. Test subjects

To find out which design would fulfil all the requirements for the setting device and at the same time provide the best user experience, user testing has been carried out. Case study with three conditions resembling each of the designs was carried out on 5 test participants using the within-group method. All of the test participants were male, their ages between 22-29 (M = 26, SD = 2.7). All test participants were semi-professional musicians chosen among Emergenza festival final contestants in Latvia, Riga.

3.2.4.2. Test setup and procedure

Pilot test was conducted in a quiet room with all the necessary equipment, e.g. guitar amplifier, effect pedals, wires etc. Test participants were introduced the goal of the experiment and were asked to think-aloud, explaining their actions and expectations. They were not given any suggestions or explanations about how to use the devices. Randomization method for the cases has been used to avoid bias. After each case test participants were asked to fill in short questionnaire. After the experiment they were asked to choose the best design.

3.2.4.3. Test results

The results have been collected through the questionnaire after each case. It was 5 point Lykert scale, where 1 was marked as “strongly disagree” and 5 was representing “strongly agree”. All of the designs were considered to be reliable with the mean for every design higher than 4.5 ($M > 4.5$). Mobile app was considered to be the most comfortable to adjust settings ($M = 4.2$, $SD = 0.8$) compared to guitar-mounted design ($M = 3.8$, $SD = 1.1$) and floor-based design ($M = 2.4$, $SD = 0.5$). The concept of the mobile app was also more appealing to test participants ($M = 4.6$, $SD = 0.9$) compared to guitar-mounted design ($M = 3.8$, $SD = 0.8$) and floor-based design ($M = 3.2$, $SD = 1.1$). 4 out of 5 test participants strongly agreed that would use such mobile app during their show.

After the experiment, all tests participants agreed that mobile app would be the most comfortable way of adjusting the settings. 2 out of 5 test participants also mentioned a potential for the guitar-mounted setting device. Therefore, it has been decided to keep only the mobile app for this design iteration. Guitar-mounted setting device could be kept as an alternative for the next design iteration.

3.2.5. Mobile app final design

The application should handle the same functions, e.g. adjust settings for the specific effects. However, since it is using mobile device screen, which is relatively large, a proper user interface can be built. Therefore, it would be meaningful not only to let users to adjust settings, but also provide possibility to choose the effects users would like to work with. Thus, a library of the effects would be required.

Moreover, during a short interview after the experiment, some of the guitar players revealed their ideas on having presets for the effect settings. This would allow saving specific settings for the specific songs, which would save time during the performance and make the application easier to use.

The above requirements were combined in the final design. When opening the app for the first time, main menu with no settings appears (Figure 28a). To choose settings, user has to press on the empty effect box. After that, effect library shows up. Three empty boxes represent three effects that can be chosen. When the effect is chosen, the effect library disappears, and settings of the chosen effect show up (Figure 28b). At the same time, the chosen effect box is highlighted to add clear visual feedback to the system. The same procedure can be done for each of the empty effect boxes. To adjust the setting, the user has to use slider to tweak parameters. For an additional ease of use, it is possible to tap on the parameters and write down the exact number.

To open up preset tab, a small arrow on top should be pressed. After that a horizontal scrollbar with multiple boxes shows up, each box representing an empty preset slot (Figure 28c). To save the current state of the effects, user should press and hold empty preset slot, after which a pop up window with the text “Preset saved” will appear. This will help user to understand the current state of his preset. To activate this preset, user has to click on it. When activated, the preset box is highlighted (Figure 28d). After that, changes to the effects and their parameters are possible. In case if user wants to discard these changes, s/he should press already existing preset. To save these changes, user should press and hold an empty preset slot. To update already existing preset, user should press and hold this existing preset slot.

To sum up, the alternative design of the stompbox effect pedals is a foot-activated wearable wireless device with two types of feedback present, e,g, visual and haptic. The design combines all strengths of the classic effect pedal design, at the same time reducing its weaknesses, such as discomfort from the setting adjustment, necessity to buy separate pedal for each effect or return to specific place on the stage. The last is supposed to improve the mobility of musicians during the performance. All of the mentioned above can be described as mobility increment-oriented effect processor (MIO effect processor).

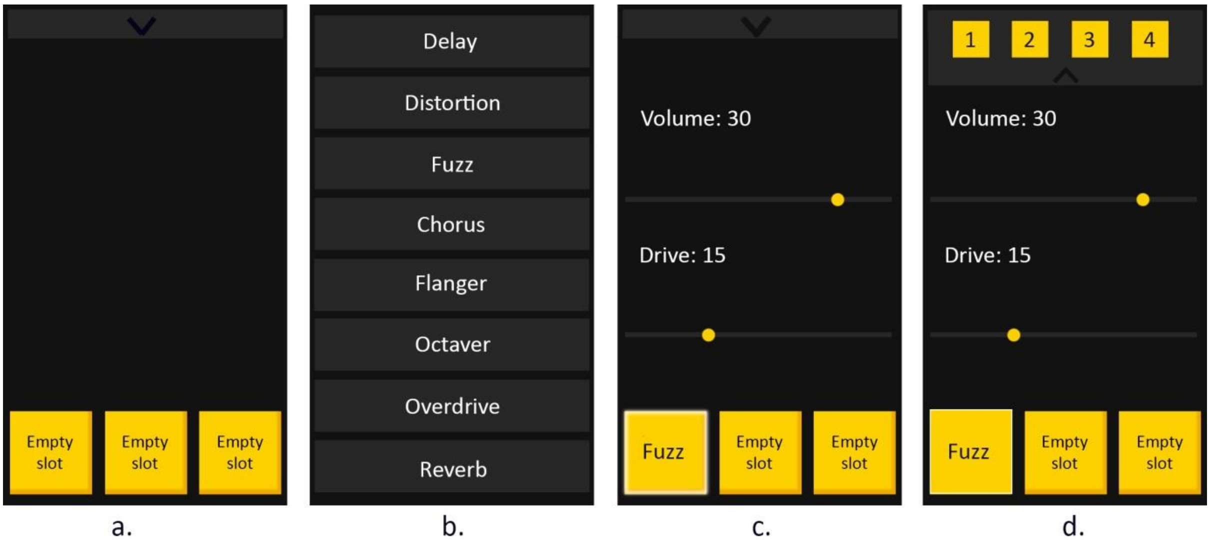

Figure 28. The design of the Mobile App

4.Implementation

4.1. Introduction / Basic description of the setup

In the scope of this project, the following objectives were established. First, the library of effects had to be implemented, so that the user could choose what effects s/he needs. Else, the requirement was that it should be possible to place them in any desired order. Moreover, presets had to be implemented, so that the setup could be saved into a preset and be accessed later.

Due to the results of the UX tests that were described above (see chapter 3.2.4), it was decided to remove the setting device and replace it with a mobile app, that was supposed to provide a user interface for both choosing the effects from library and then managing them.

Speaking of Foot Controller, the concept remained the same since the majority of test participants from previous UX tests appreciated it, however it was concluded that more visual feedback should be made. To be more specific, three additional LEDs should be added as described in chapter 3.2.1, which were supposed to reflect which effects were switched of and off.

4.2. The foot controller

The foot controller was built upon a metal frame that was supposed to enfold user's foot. One rubber strap was used to prevent it from falling off. NodeMCU microcontroller was chosen to be the core of this device due to its compactness, processing speed (160Ghz) and on-board Wi-Fi module. It was attached to the upper side of the device. Grove - IMU10DOF sensor was chosen for orientation tracking, and it was attached to the right side of the device, next to the right side of the user's heel. One RGB LED was placed right next to NodeMCU, and its role was to help the user to navigate through effects while choosing one. Three LEDs (red, green and blue) were attached near RGB LED, and they were supposed to inform the user about which effect is activated. Finally, a vibration motor was added to the system for the future haptic feedback implementation.

The main requirement for this device was to allow musicians to activate or deactivate effects being wherever on the stage. This feature was enabled by implementing two different states of the device. During the first state, the musician could freely walk, jump or run, while the device was waiting for the specific inversion movement with the foot the device was attached to (see chapter 3.2.2.). After the musician performed inversion movement while standing on one place, the device entered the second state, which allowed choosing the effect. The user could rotate his foot to either left or right side for choosing the effect s/he wants to activate. RGB LED was helping to navigate through the effects by changing the colour of the light, e.g. if user turns his foot to the left side, LED shined red, if to the right side, it shined blue. If the users' foot stayed in the middle, LED shined green (Figure 29). Each colour corresponded to one of the effect slots. Finally, in order to change the state of the effect, the user had to lift a toe and imagine that s/he has to press a button (which would resemble the use of usual pedal effects).

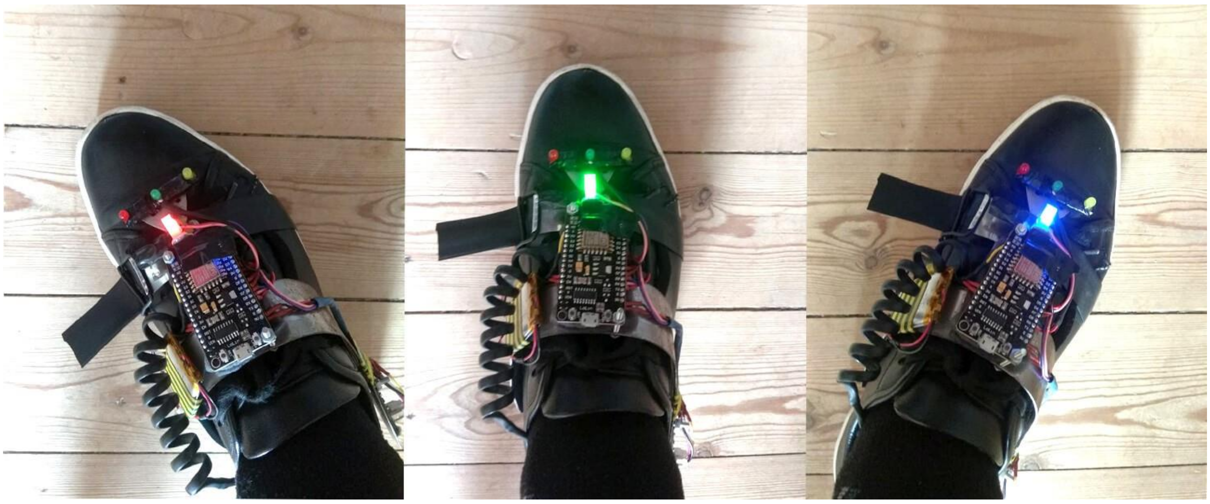

Figure 29. Foot positions resembling three effect slots

The proper position tracking algorithm had to be implemented. During the first state, the device had two main objectives: it had to distinguish when the user is in the move (e.g. s/he is walking, running, etc), and it had to distinguish the inversion movement. The key to implementing this feature was to be able to distinguish the inversion movement, and then consider everything else as "the person is walking". As mentioned above, The IMU10DOF was used for position tracking. This sensor can provide a lot of information, but only its functionality as a gyroscope was used. It showed the acceleration of the rotation of all three axes of the sensor. The IMU10DOF sensor was placed on the right side of the heel, and it was crucial because in this case when the user performed an inversion movement, only Y axis showed some significant acceleration (Figure 30).

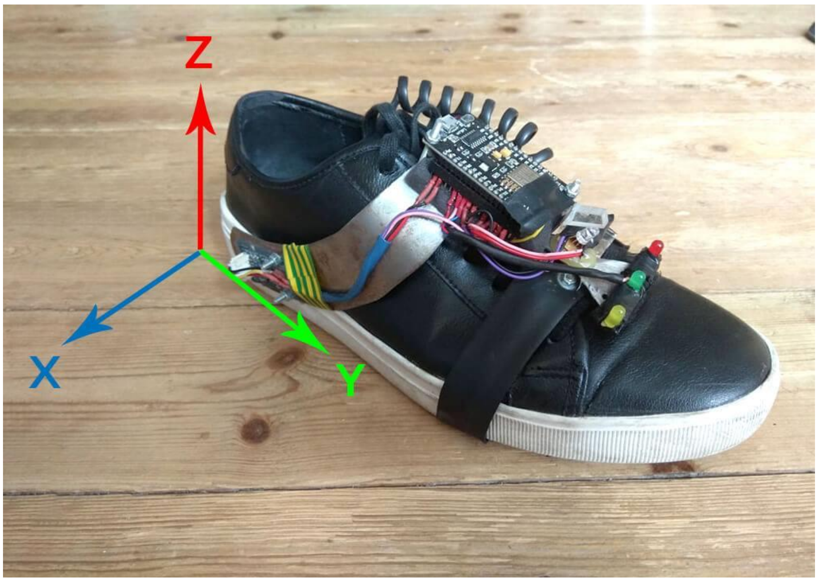

Figure 30. The axes directions of the IMU10DOF

The other two axes (X and Z) only showed insignificant noise. Basing on this fact, the following assumptions were made. If X and Z axis are showing some significant acceleration, that means that the person is moving and the device should be blocked. However, if X and Z axis are not showing any acceleration, that means that the person is not walking and his foot is standing on the ground. In this case we track Y axis and wait for a significant acceleration, while still checking X and Z axis for any activity. If we receive high enough values from Y axis, we let the device enter the second state.

Just right after the device has entered the second state it had to memorize its current location in the space as “zero angle position”. It was done by creating a variable that was equal to itself + the current acceleration of Z axis every loop iteration. This trick didn’t give the exact angle the user turns the device, but it gave good enough reference on where the foot is turned. All conditions are represented on Figure 31.

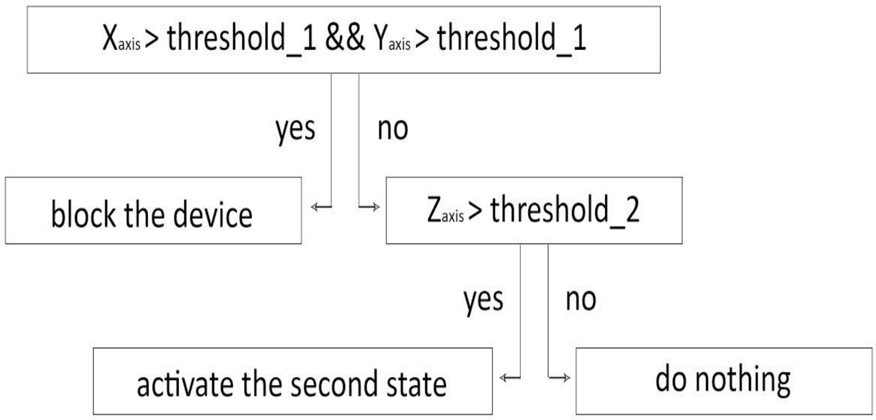

Figure 31. All possible conditions for the activation device

This foot controller was communicating with DSP unit using Wi-Fi. UDP protocol was chosen due to simplicity and data transferring speed. It is considered to be less secure since there is no check that the data was received, however it hasn’t failed a single time during testings, so it was decided to leave it as it is for now. The DSP unit was creating the network, and all devices received their static IP addresses on the first “handshake”.

Finally, a haptic feedback was implemented using a vibration motor. It was performing a short vibration during the activation of the device, and also when any of the effects was activated or deactivated.

4.3. The Mobile App

Xamarin was chosen as mobile app development platform due to various reasons. The biggest advantage of Xamarin is that it allows developing a mobile application simultaneously for three different platforms - Android, iOS and Windows Applications. Another advantage is that it is based on C#, therefore a developer doesn’t have to spend a lot of time learning a completely new framework, and the development can start right away. Xamarin offers many premade UI objects, which makes it faster and easier to develop. Finally, Xamarin strives to deliver an app that is almost identical to native API, which ensures the best possible performance and also allows using native UI, SDKs etc. (Leuschenko O., 2018).

The development of the app is divided into Front End and Back End. Front End is essentially the visual aspect of the app. On the other hand, Back End is the functionality of the app. The Front End code is written in .xaml, and it defines the type of layout, objects present in the page, their size and colour. It also provides a link to a method lying on the Back End that will be executed on a certain type of interaction with the object. The example can be observed on the Figure 31. In this case, the layout is "StackLayout", which stacks the objects one on top of another with respect to the orientation. This particular layout consists of three buttons. The names of the buttons are dynamically changed by the Back End. Different methods will be executed when the button was pressed and when it was released. It was necessary for implementing "press and hold" functionality, which will be explained later.

```
<StackLayout
    Orientation="Horizontal"
    VerticalOptions="EndAndExpand">
    <Button x:Name="first_effect_button"
            HorizontalOptions="StartAndExpand"
            WidthRequest="100"
            HeightRequest="100"
            Pressed="First_effect_button_Pressed"
            Released="First_effect_button_Released"
            />
    <Button x:Name="second_effect_button"
            HorizontalOptions="CenterAndExpand"
            WidthRequest="100"
            HeightRequest="100"
            Pressed="Second_effect_button_Pressed"
            Released="Second_effect_button_Released"
            />
    <Button x:Name="third_effect_button"
            HorizontalOptions="EndAndExpand"
            WidthRequest="100"
            HeightRequest="100"
            Pressed="Third_effect_button_Pressed"
            Released="Third_effect_button_Released"
            />
</StackLayout>
```

Figure 31.The layout of three effect slots

These three buttons represented the effect slots. They were placed on the bottom of the app and when the user opened the app, each of them said "Choose the effect". It was decided that these buttons should have the following functionality. At first, when the user clicked one of these buttons, a new layout popped up, showing the list of available effects. After the user chose the effect, the button changed the displaying text from "Choose the effect" to the name of the effect the user chose. However the user also had to have a possibility to discard chosen effect and take a new one. This was done by allowing to press and hold the button until the popup window showed up asking "Would you like to discard this effect?". This can be observed on the Figure 32.

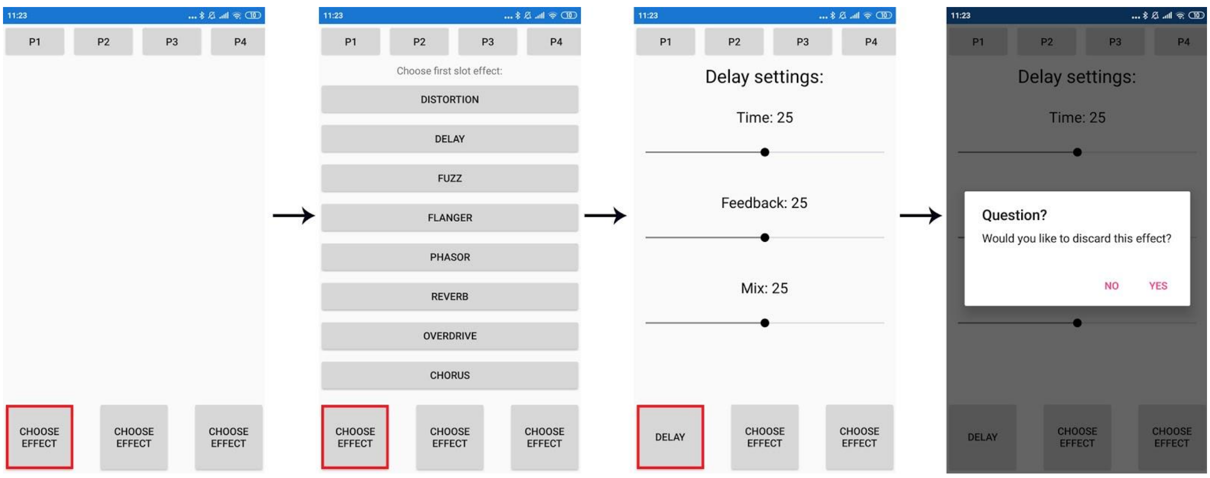

Figure 32. The sequence of choosing and removing the effect for a particular effect slot

In order to implement such functionality, a custom button renderer had to be made. Basically it was important to distinguish whether the button was just clicked, or it was pressed and held for 1 second. Xamarin offered two handy method triggers - “Pressed” and “Released”. The first one triggered a method on a Back End if the button was pressed, and other one triggered another method if the button was released. For the first button, the names of these methods were “First_effect_button_Pressed” and “First_effect_button_Released” (Figure 33).

```
private void First_effect_button_Pressed(object sender, EventArgs e)
{
    firstButtonIsReleased = false;
    firstButtonIsPressed = false;
    trackFirstButtonState();
}
```

Figure 33. “First_effect_button_Pressed” method

When the user pressed a button, first method was executed. It triggered another method called “trackFirstButtonState”, which started to monitor the state of the button from another thread. Its purpose was to wait 1000ms and then check if the button was still pressed. If it was, then it triggered another method called “firstButtonWasOnHold” (Figure 35), which essentially displayed the alert message. It should be noted that it worked only if the effect was already chosen. The code snippet of “First_effect_button_Released” can be seen on the Figure 34 down below.

```
private void First_effect_button_Released(object sender, EventArgs e)
{
    firstButtonIsReleased = true;
    if (firstButtonIsPressed == false)
    {
        if (effect_1_chosen == false)
        {
            if (EffectChoice1.IsVisible == false)
            {
                EffectChoice1.IsVisible = true;
                EffectChoice2.IsVisible = false;
                EffectChoice3.IsVisible = false;
                hideSecondButtonView(second_effect_selected);
                hideThirdButtonView(third_effect_selected);
            }
            else
            {
                EffectChoice1.IsVisible = false;
            }
        }
        else
        {
            hideAndDisplayFirstButtonView(first_effect_selected);
        }
    }
}
```

Figure 34. “First_effect_button_Released” method

However if the button was released earlier, then this action was interpreted as a simple click. In this case, if the effect was not yet chosen, it opened the effect library, or if the effect was chosen, then it either hid or opened settings of chosen effect. It also hid all other windows that potentially could be opened, which are the settings of the effects from the other two slots or libraries. The closure of these windows was done by accessing their "IsVisible" parameter, which allowed changing the visibility of the layout.

When any of the effects was either chosen or discarded, or some of the settings were changed, the app was sending the data to the DSP unit. UDP protocol was chosen for data transfer. The IP of the DSP unit was hardcoded. Different ports were used for different purposes. Overall, 30 different ports were used to transfer the data. It might seem too much for such a simple setup, but it was much easier to scale the code this way.

```
public async void trackFirstButtonState()
{
    await Task.Run(() =>
    {
        for (int i = 0; i <= 50; i++)
        {
            if (firstButtonIsReleased == false)
            {
                Thread.Sleep(20);
            }
            else
            {
                i = 51;
            }
        }
    });
    firstButtonIsPressed = true;
    if (first_effect_button.IsPressed)
    {
        if (effect_1_chosen)
        {
            firstButtonWasOnHold();
        }
    }
}
```

Figure 35. "firstButtonWasOnHold"method

Each effect was unique. That means that there were 8 effects implemented for each slot, so it was possible to place delay effect into all three slots and adjust them differently. Each effect had its personal view with settings. The view contained as many sliders as the effect required. For instance, the Fuzz effect setting view contained only two sliders, but Flanger effect setting view contained 4 sliders. Each setting view was linked to a specific method on the Back End that was listening for sliders and was sending the data to the DSP unit in case if some parameters were changed. It was also updating the number above the slider on the Front End.

```
void S1_Dist_Value_Changed(object sender, ValueChangedEventArgs args)
{
    if (sender == S1DistToneValue)
    {
        S1_Dist_Tone = S1DistToneValue.Value;
        S1DistToneLabel.Text = String.Format("Tone: {0}", (int)args.NewValue);
    }
    if (sender == S1DistDistortionValue)
    {
        S1_Dist_Distortion = S1DistDistortionValue.Value;
        S1DistDistortionLabel.Text = String.Format("Distortion: {0}", (int)args.NewValue);
    }
    if (sender == S1DistVolumeValue)
    {
        S1_Dist_Volume = S1DistVolumeValue.Value;
        S1DistVolumeLabel.Text = String.Format("Volume: {0}", (int)args.NewValue);
    }

    S1_Dist_Settings = Math.Floor(S1_Dist_Tone).ToString() +
        "|" + Math.Floor(S1_Dist_Distortion).ToString() +
        "|" + Math.Floor(S1_Dist_Volume).ToString() + ".";
    Send_Settings(S1_Dist_Settings, 4001);
}
```

Figure 36. The renderer of the distortion effect from the first effect slot

Before sending the data, all the numbers were rounded up to the lowest nearest integer. Before sending, all numbers were put into a string, split with "|" character. It was done to make it easier to split these numbers on the other side. Then this string was converted to bytes using ASCII Table. This string was created in the settings view renderer, and then sent to the "Send_Settings" method, which was used for sending the data. The example of the setting renderer of the Distortion effect from the first slot can be observed on the Figure 36. The "Send_Settings" method can be observed on the Figure 37.

```
public void Send_Settings(string input, int port)
{
    using (var client = new UdpClient())
    {
        try
        {
            client.Connect("172.24.1.1", port);
            Byte[] sendBytes = Encoding.ASCII.GetBytes(input);
            client.Send(sendBytes, sendBytes.Length);
            client.Close();
        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
    }
}
```

Figure 37. "Send_Settings" method

Finally, the presets were implemented. First, the separate horizontal-oriented layout was created and placed on top of the screen. It contained ten buttons placed in a row. In order to make it possible to scroll through the buttons, "DynamicStackLayout" was used. Each button had the same functionality as effect slot buttons, e.g. they were able to distinguish between "click" and "was pressed and held". It was decided that the user should be able to save a preset by pressing a button and holding it for one second. In this case a popup window was displayed informing that the preset was saved. To apply the preset, the user had to just click the button once. In order to reset the preset, the user had to press and hold the button again. Another popup window was shown asking if the user wanted to delete the preset.

In order to save the preset, Xamarin.essentials was used. It allowed saving basic types of data in the memory of the phone. To save a preset, the effects that were chosen were saved as well as their settings. The example of the code that performed the saving of the preset is shown on the Figure 38.

```
public void Save_First_Preset(bool save)
{
    if (save)
    {
        Preferences.Set("P1_first_effect_selected", first_effect_selected);
        Preferences.Set("P1_second_effect_selected", second_effect_selected);
        Preferences.Set("P1_third_effect_selected", third_effect_selected);

        if (effect_1_chosen == true)
        {
            if (first_effect_selected == "Distortion")
            {
                Preferences.Set("P1_S1_Dist_Tone", S1_Dist_Tone);
                Preferences.Set("P1_S1_Dist_Distortion", S1_Dist_Distortion);
                Preferences.Set("P1_S1_Dist_Volume", S1_Dist_Volume);

            }
            if (first_effect_selected == "Delay")...
            if (first_effect_selected == "Fuzz")...
            if (first_effect_selected == "Flanger")...
            if (first_effect_selected == "Phasor")...
            if (first_effect_selected == "Reverb")...
            if (first_effect_selected == "Chorus")...
        }
        if (effect_2_chosen == true)...
        if (effect_3_chosen == true)...
        Console.WriteLine("first preset saved");
    }
}
```

Figure 38. Preset saving

Finally, in order to apply the preset, all the settings and effect slots were cleared. Then, it was figured out which effects were stored in the preset and which slot they belonged to. Then, after the effects were placed into slots, settings were applied too. Obviously all changes were sent to the DSP device, so the right effects with the right settings were put into the sound chain.

The last touch was to make a button that would switch the DSP unit off. During the development of the prototype it was discovered that the SD card that was used by Raspberry Pi as the main memory storage can easily get corrupted if it is frequently switched off the hard way, e.g. by just by unplugging it from the power source. Since during the regular use of the prototype the Raspberry Pi was running headless, it had to be switched off in more convenient way. The solution was to add one more button in the Mobile App interface that would trigger Shell command on Raspberry Pi to switch it off.

4.4. The DSP Unit

The DSP unit was based on Raspberry Pi 3. It runs Raspbian operating system, which is adopted version of Debian. Behringer UCA202 was used as a sound card, which allowed having low audio latency (around 6 - 8 ms). Pure Data was used for creating all the audio effects. Small Java server was used for the communication between Pure Data, Mobile App and Foot controller.

First of all, the Raspberry Pi had to be converted into a Wi-Fi hotspot, so that it would be possible to connect other devices to it. In this case no additional/external network was required, the whole interface was standalone. Hostapd and dnsmasq programs were used for the hotspot setup. Static IP address was set to 172.24.1.1. Name of the network was “Pi3-AP”.

Next, the Java server was implemented. Basic type of UDP server was made. Essentially, its purpose was to listen to different ports, read data and make a certain action depending on which port it was listening to and what kind of data it received. The most generic scenario was that it had to receive data from Mobile App and Foot controller, process it and forward it to the Pure Data patch. To simplify things and make the code more scalable, a class “Send_Data” was created. It allowed creating multiple instances of this class, each for a specific port. The code can be observed on the Figure 39.

```
public Send_Data(int Receive_From_Port, int Send_To_Port, String thread_name){
    this.Receive_From_Port = Receive_From_Port;
    this.Send_To_Port = Send_To_Port;
    this.thread_name = thread_name;
}

public void run(){
    try{
        System.out.println("I am:"+ thread_name +", and I am listenning to port " + Integer.toString(Receive_From_Port));
        DatagramSocket LCDserverSocket = new DatagramSocket(Receive_From_Port);
        byte[] receiveData = new byte[100];
        byte[] LCDsendData1 = new byte[110];

        while(true){
            try{
                DatagramPacket LCDreceivePacket = new DatagramPacket(receiveData,receiveData.length);
                LCDserverSocket.receive(LCDreceivePacket);
                String LCDsentence = new String(LCDreceivePacket.getData());
                String name = LCDsentence.substring(0,LCDsentence.indexOf('.'));
                System.out.println(name);
                String [] LCDbarray = name.split("\\|");

                int[] LCDstringToNum = new int[LCDbarray.length];
                for(int g = 0; g<LCDbarray.length; g++){
                    LCDstringToNum[g] = Integer.parseInt(LCDbarray[g]);
                }
                for(int b = 0; b<LCDbarray.length; b++){
                    LCDsendData1[b] = (byte)LCDstringToNum[b];
                }

                InetAddress address1 = InetAddress.getByName("localhost");
                DatagramPacket packet  = new DatagramPacket(LCDsendData1,LCDsendData1.length,address1,Send_To_Port);
                packet.getPort();
                DatagramSocket dsocket = new DatagramSocket();
                dsocket.send(packet);
                dsocket.close();

            }
            catch(Exception ex1){
                System.out.println(ex1);
            }
        }
    }
    catch(Exception ex2){
        System.out.println(ex2);
    }
}
```

Figure 39. “Send_Data” class implementation

During the declaration, three parameters had to be passed to the object - the port that the object had to listen to, the port to which it had to send data and the name. The last one was needed just to check that everything works as it should and all objects are created. Each object was starting a separate thread to avoid interference. Overall, 30 different threads were started from 30 different objects. The declaration of some of the objects can be seen on Figure 40.

```
public static void main(String[] args){


    Thread t1 = new Thread(new Send_Data(4001,5001,"t1"));
    t1.start();

    Thread t2 = new Thread(new Send_Data(4002,5002,"t2"));
    t2.start();

    Thread t3 = new Thread(new Send_Data(4003,5003,"t3"));
    t3.start();

    Thread t4 = new Thread(new Send_Data(4004,5004,"t4"));
    t4.start();

    Thread t5 = new Thread(new Send_Data(4005,5005,"t5"));
    t5.start();

    Thread t6 = new Thread(new Send_Data(4006,5006,"t6"));
    t6.start();

    Thread t7 = new Thread(new Send_Data(4007,5007,"t7"));
    t7.start();

    Thread t8 = new Thread(new Send_Data(4008,5008,"t8"));
    t8.start();

    Thread t9 = new Thread(new Send_Data(4009,5009,"t9"));
    t9.start();

    Thread t10 = new Thread(new Send_Data(4100,5100,"t10"));
    t10.start();
```

Figure 40. The declaration of threads

When the code was done, it was saved into a .jar file and was set to be executed on every boot.

Finally, the Pure Data patch was implemented. Since it should have been possible to choose the same effect for each effect slot, three identical stacks of effects were implemented. Therefore it was possible to place, for instance, delay effect into all three slots, but essentially they were three different instances of delay effect. The schematic view can be observed on the Figure 41.

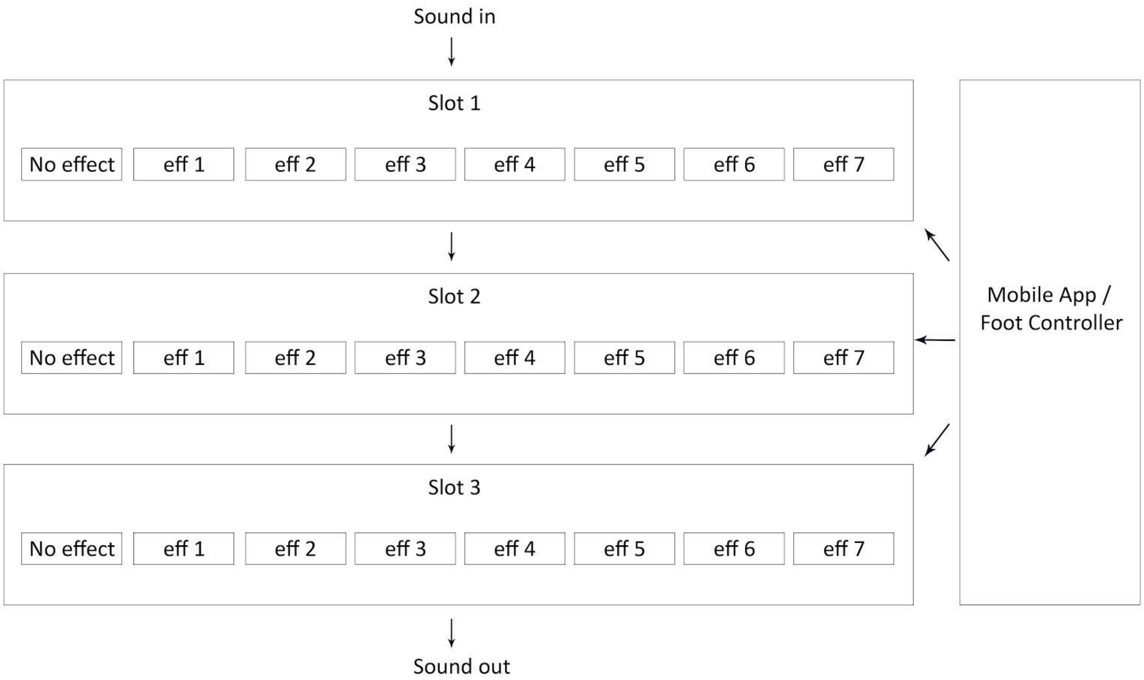

Figure 41. The schematic view of the DSP unit infrastructure

Each slot was listening to both Mobile App and Foot controller. Basically, the Mobile App was defining which app was placed in the slot, and the Foot controller switched this effect on and off. Each effect was listening to its own individual port, which is why so many threads were implemented in Java server. A bird's eye view on the Pure Data patch is presented on the Figure 42.

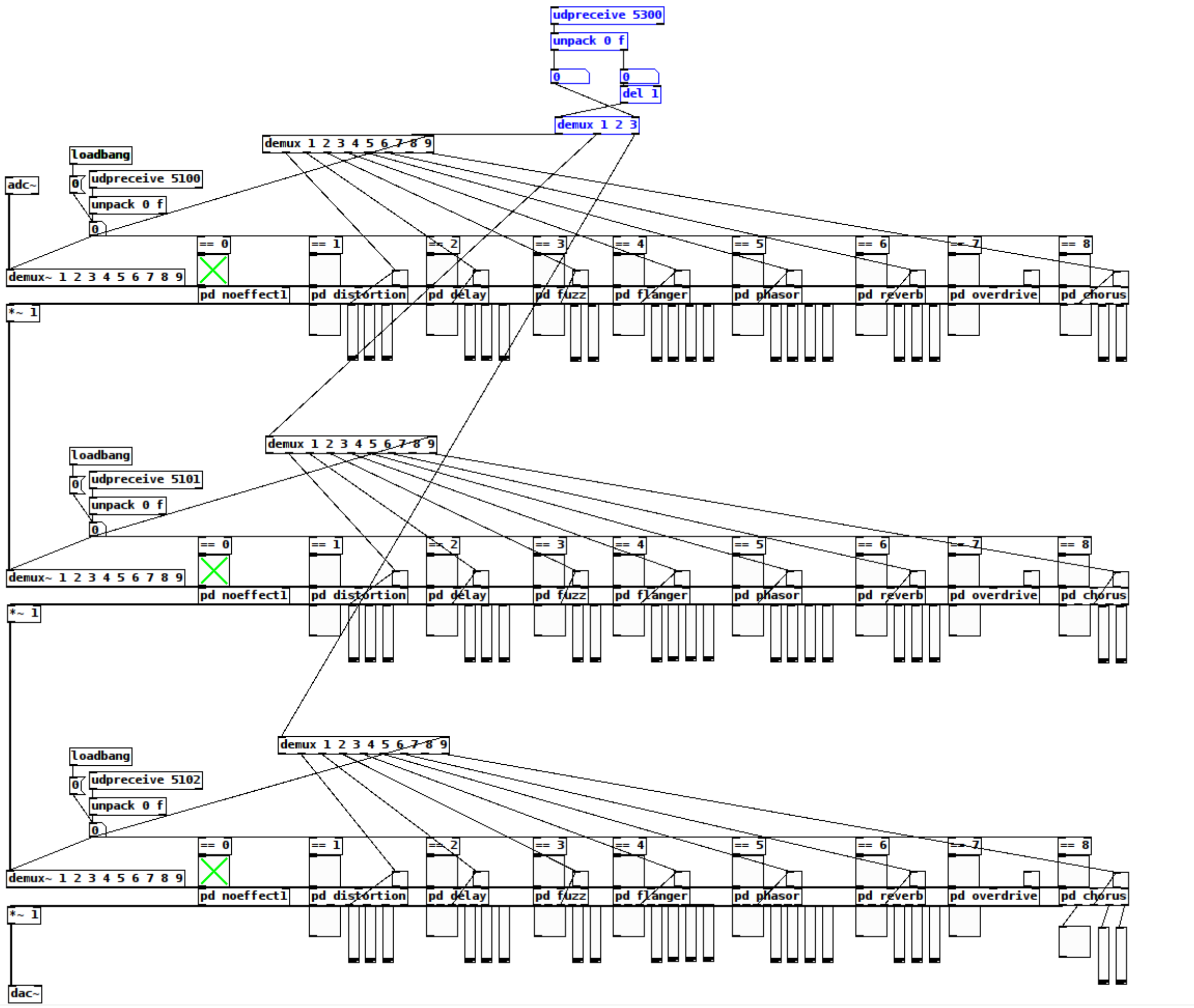

Figure 42. The top layer of the Pure Data patch.

Seven effects were implemented: distortion, delay, fuzz, flanger, phasor, reverb and chorus. Everything was prepared for the overdrive as well, but unfortunately it was not eventually implemented. One of the challenging things was that Pure Data patch could only utilize one core of a processor. Since there was quite a lot of code, it was heavily loading the processor even when there was no audio processing yet. When the audio processing was on, it was glitching a lot and the audio was unusable. This issue was solved by adding a "switch" object into each subpatch. It made it possible to choose when to process a particular subpatch, and when not. This can be observed in the distortion patch on the Figure 43. Most of the effects implemented use the same principle. Thus, not all of them will be explained. Some effects were built from scratch, whereas some open-source effects were obtained from the internet (Guitar Extended, 2019) (Pure Data forum~, 2019).

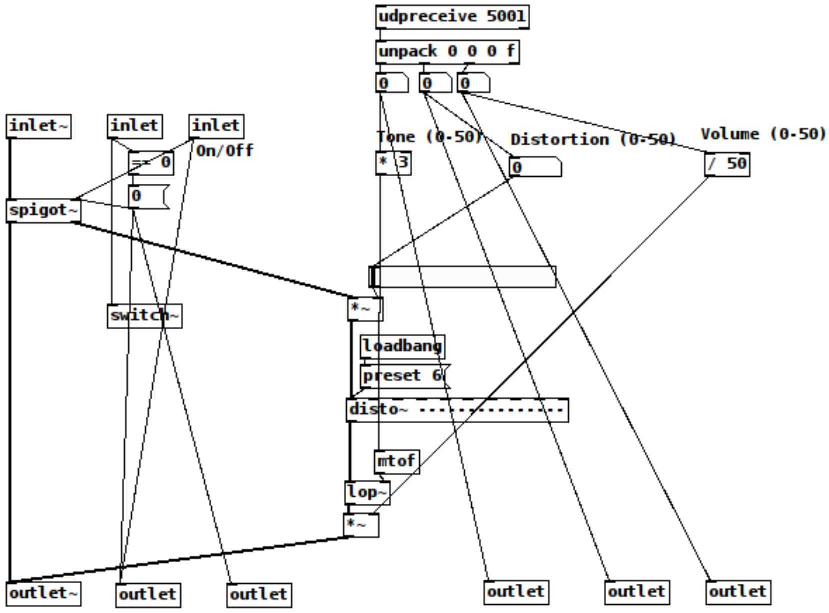

Figure 43. The distortion effect

The distortion patch was created using [disto~]. It is a premade object available in Pd-extended, and it offers six different types of distortion, from light to quite heavy. This patch had three parameters - output level, tone and distortion level. The “output level” parameter was done by placing a [*~] object, which multiplied every incoming audio sample by a number between 0.1 and 1. The “tone” parameter was implemented by using a simple low-pass filter. The “distortion level” parameter was adjusted by changing the level of a signal that attacked the [disto~] object.

The delay effect was implemented using [delwrite~] and [delread~]. [delwrite~”] object was writing the sound into a buffer, and [delread~] was playing this sound after a certain time set in milliseconds. As it can be observed on the Figure 44, the effect was created by sending the sound from [delread~] back into [delwrite~] with a certain reduce in volume. This effect had three parameters as well - time, feedback and mix. The “time” parameter was done by changing the time after which the [delwrite~] was playing the audio. The “mix” parameter was implemented by changing the damping factor of the sound that was coming from [delread~] back to [delwrite~]. The “volume” parameter was done in the same fashion as in distortion effect. The patch can be seen on the Figure 44.

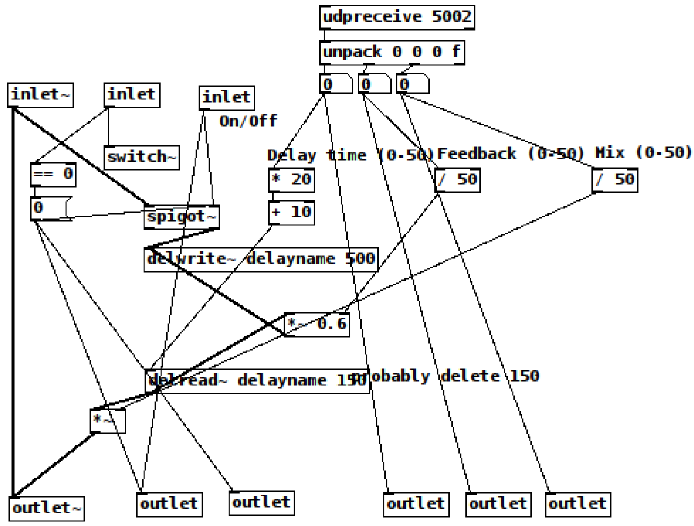

Figure 44. The delay effect

The chorus effect was implemented using multiple [vd~] effects, which basically had the same functionality as [delread~], but it could be used in conjunction with, for instance, oscillator or phasor. Oscillators of different frequencies controlled the time after which the [vd~] objects played the recorded sound. All that created a feeling of having multiple objects playing the same thing with slight delays, which essentially is a chorus effect. The implementation of the patch can be observed on the Figure 45.

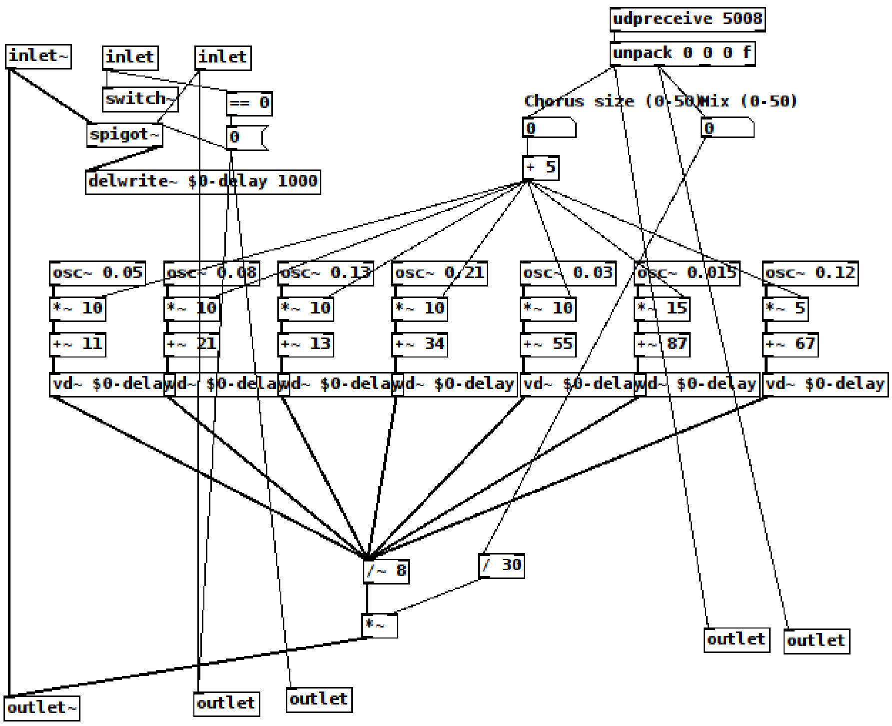

Figure 45. The chorus effect

4.5. Results

The prototype turned out fully functional. The Mobile App was very stable and fast. It was lacking the visual aspect, but functionality-wise it was strong. The Foot Controller has got much better visual feedback, but the code optimization still had room for improvement. The library of effect was ready to use, and the ability the save the whole setup as preset was available as well.

5. Methods

5.1. Conditions and hypothesis

In order to find out the advantages and disadvantages of the MIO effect processor, final experiment was conducted. It should have not only revealed if MIO effect processor design can provide the same level of comfort and functionality, but also if it could provide more mobility during the performance. To answer these questions, the experiment was divided into three steps.

5.2. Usability testing

5.2.1. Test procedure

Before carrying out the final experiment it was necessary to make sure that the prototype is fully functional. To do so, a usability testing was carried out. It was supposed to reveal any malfunctions, if such existed. The usability test does not require large amount of people. According to the Knapp (Knapp J., 2016), five people are enough to reveal 80% of all problems. Moreover, since it is only functionality of the prototype, these people may be not part of the target group. Thus, five test subjects with no previous experience in music and specifically guitar playing participated in the usability testing. 3 of them were men and 2 of them were women with their ages between 21-28 ($M = 24.8$, $SD = 2.59$). The usability testing was task based to make sure that every single function is tested out. The tasks covered stability of foot activated switch, e.g. the ability to move, walk, jump etc. without activating the effects, the ability to distinguish between different effects, adjust all the settings through the mobile app and use presets.

5.3. Case studies

The next step was to compare our effect pedal redesign with the original effect pedals to see, if they provide the same range of comfort, functionality and reliability. All of this could be summed up in one word - experience. The goal of the experiment was to find out whether the experience of two designs differed and if one of them could be considered better or worse. Thus, two conditions have been established - classic effect pedals design and MIO effect processor.

Null-hypothesis: There is no significant difference between the experience of the effect pedal design and MIO effect processor.

Alternative-hypothesis: There is significant difference between the experience of the effect pedal design and MIO effect processor.

A significant difference between the cases would mean that one design is significantly better than another and, depending on the mean, this could be considered either as a success or a failure of the design. No significant difference would mean that both designs provide similar experience and could be considered as competitive.

5.3.1. Test subjects

10 test subjects with a previous experience in playing the guitar and using effect pedals participated in the experiment, their years of playing between 3-20 (M = 10.7, SD = 6.43). 9 of them were males and 1 was female, their ages between 18-38 (M = 26.1, SD = 3.7). To gather more sensitive information within-group method was used (Field & Hole, 2002). Convenience sampling was used when selecting test subjects in order to save time and money, which would be much more difficult if using random sampling (Salkind, 2010). All test subjects were chosen from the group of guitar players on social media.

5.3.2. Set-up and apparatus

The final experiment was conducted in the quiet room with all necessary equipment present. First part of the experiment was conducted in Denmark, in Valgårdsvej 4, Copenhagen, in a rehearsal studio, and the second part was conducted in another rehearsal studio in Latvia, Brīvības gatve 193 d, Riga. The setup consisted of MIO effect processor device, three stompbox effect pedals – Boss DS1 distortion, Joyo analog delay and Behringer DR600 digital reverb; and electric guitar. The sound was delivered through Marshall guitar amplifier.

5.3.3. Test procedure

Test subjects were explained the task of the experiment, e.g. play Nirvana “Smells like teen spirit” song using both effect devices alternately. The song was chosen due to its popularity and ease, as well as necessity to use multiple effects. The controls for the activation of the effects were presented as well. Test subjects were allowed to try out the MIO effect processor device. After getting used to the design and functionality, the experiment started.

Before playing the song test participants had to choose the right effects in the app and adjust their parameters. No suggestions or explanations regarding the app were given. To imitate a real-life scenario of a stage performance, backing track with drums and bass was present. Thus, test subjects had to play in a certain tempo and activate and deactivate the effects at a specific point of time.

The randomization method was used, e.g. the experiment started with a random condition for every test subject to avoid bias and reduce the opportunity of learning. After every condition a questionnaire was filled in. The questionnaire covered such topics as functionality, comfort, stability and reliability, user-friendliness and ease of use. There were two inverse questions in the questionnaire to test out the attention of the test subjects and avoid response bias, when only highest or lowest values are chosen for all the questions.

5.3.4. Data analysis

First step in data analysis was to find out if data gathered was parametric. To make sure that data is analysed correctly, negatively constructed questions have been inversed. The data is parametric if it satisfies three assumptions.

First assumption: Data is interval or ratio

The data was gathered through questionnaire with 5-point Likert scale. It is still hard to clearly define type of the data, since the opinions of the researches are divided. Some of them suppose that Likert scale provides ordinal data, the others assume that it is interval data that can be gathered (Brown, 2011). Since some of the scientists assume that distance between Likert items is equal, it can be assumed that data is interval.

Second assumption: Homogeneity of variances

In other words homogeneity of variances means equal variance for both groups. This assumption can be proved or disproved with the use of Levene's test, which is very efficient for the sample sizes around 30 test participants (Field & Hole, 2002). The results (Table 1) showed no significant difference in variances of both conditions (p = .95).

Test of Homogeneity of Variances

value

Levene Statistic	df1	df2	Sig.
.004	1	18	.953

Table 1. The Levene's test results

Third assumption: Normally distributed population

Normal distribution is a bell-shaped symmetric distribution. There are multiple normality tests that can show the distribution of data, such as Kolmogorov-Smirnov, Anderson-Darling, Lilliefors and Shapiro-Wilk test, which are considered to be the most frequently used ones (Razali, N. M., & Wah, Y. B., 2011). As the test results may differ, it has been decided to use two tests with different approaches, e.g. Kolmogorov-Smirnov displays the data through histogram and Shapiro-Wilk displays the distribution through the Q-Q plot. Both tests showed no significant difference between the conditions, (p = .095) for the stompbox effect pedals and (p = .159) for the MIO effect processor design according to Kolmogorov-Smirnov test, and (p = .306) for the stompbox effect pedals and (p = .152) for the MIO effect processor design (Table 2).

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	case	Statistic	df	Sig.	Statistic	df	Sig.
value	1	.244	10	.095	.914	10	.306
	2	.226	10	.159	.886	10	.152

a. Lilliefors Significance Correction

Table 2. Kolmogorov-Smirnov and Shapiro-Wilk normality test results

When the data appeared to be parametric, a paired t-test with 95% confidence level was performed to find out significant difference between effect pedal design and MIO effect processor design. To get a better understanding of the information gathered, descriptive statistics was used. To support the quantitative data, qualitative interviews have been conducted after the test.

5.4. Mobility test

The final step was to find out if MIO effect processor design could provide more mobility on the stage compared to effect pedal design. The mobility issue was one of the main reasons for the development of the MIO effect processor design, and whole design was built around additional mobility, therefore, it was necessary to gather valid data about the possibilities of moving around during the performance.

Since mobility is a subjective measure, it has been decided to gather the data through interviews, but not from the musicians using the device, but from the audience, watching the performance. The opinion of the audience matters due to the fact that it is their perception of the performance that is influenced by the mobility on the stage, not the musician's. Therefore, only audience can decide if the performer was mobile during the performance, if the show was dynamic enough etc.

5.4.1. Test subjects

3 guitar players and 12 viewers participated in the experiment. The viewer ages were between 24-28 ($M = 25.5$, $SD = 1.38$). 9 of the test subjects were males, 3 were females. A within-group design was used, e.g. the audience listened to both songs of every guitar player.

5.4.2. Set-up and apparatus

The final experiment was conducted in the small club called Republika in Latvia, Riga, Mazā Pils iela 11, with all necessary equipment, such as stage lights, PA system amplifiers etc. The guitar players used their own electric guitars. The setup consisted of MIO effect processor and distortion, delay and reverb effect pedals. The guitar players performed on a stage with backing track playing, the sound amplified through PA system as on a real show.

5.4.3. Test procedure

Before the experiment, guitar players were asked to prepare two similar songs and a backing tracks for them to be able to perform in front of a small crowd. For one of the songs classic effect pedals were used, for the other they were asked to practice and perform using MIO effect processor. The guitar players were explained their main task, e.g. perform their songs being as mobile on the stage as possible. When they were ready to use both devices, audience has been found and brought to the venue.

The conditions of the event as close to real performance as possible were created. Each guitar-player had his own performance time. Guitar players started with a random design to avoid bias. After performing two songs, the next musician came on the stage.

After every performed song test participants were asked to fill in short questionnaire, covering such topics as visual appeal of the performance, amount of movement and walking on the stage etc. After every guitar player's performance the audience was asked to share their thoughts about which song's performance they enjoyed more. It was important that two of the performed songs were similar, e.g. their auditory perception remained unchanged, therefore, the only thing that differed was visual perception. The effect device design was the key element, and it was the only changing factor of the performance. Therefore, it was considered that all changes in the visual perception were influenced by the effect device used. The device that was used in the song that was enjoyed more was considered to provide more mobility. After the experiment test subjects were asked to participate in a short semi-structured interview to gather deeper understanding of the perception of the performances and reasons for the choices made. Moreover, guitar players were asked to share their thoughts regarding the experience of the use of MIO effect processor during the real show.

5.4.4. Data analysis

The data gathered from the songs played with stompbox effect pedals and the ones, performed with MIO effect processor, have been averaged per person respectively. The results were analysed with Levene ($p = .006$), Kolmogorov-Smirnov ($p = .08$) and Sharipo-Wilk ($p = .08$) tests. The data appeared to have unequal variances and, thus, being non-parametric. Therefore, a Wilcoxon test was used to find out if there was statistical difference between visual aspects of the performance when using stompbox effect pedals and MIO effect processor.

The information collected through the interviews allowed making correlations between the design of the effect device and mobility on the stage, mobility and the perception of the show etc. The information from the guitar players was also analysed for the future redesign.

6. Evaluation

Each of the methods used for the testings was supposed to reveal different aspects of the MIO effect processor design. The usability testing was supposed to show any possible technical issue regarding the code and construction of the prototype. The case studies compared the experience gathered from the use of MIO effect processor design with the classical stompbox effect pedals experience. Such side by side comparison allowed unveiling the main strengths and weaknesses of both designs and find out, if MIO effect processor design could be competitive with the stompbox effect pedals design. Finally, the results gathered during the mobility tests could show if two designs actually influenced differences in visuals of the performance, and if MIO effect processor design could actually provide guitar players more mobility on the stage.

6.1. Usability testing results

The results of the usability testing revealed a stability issue. The device didn't always enter the second state as expected. This was happening due to a couple of reasons. First, people had different physical ability of twisting their toe, therefore, some were performing the inversion movement quite successfully, but the others struggled a bit. This could be fixed by providing user a possibility to calibrate the device by changing the sensitivity of the device (calibration happening through the Mobile App). In case if a person struggled to perform an inversion movement, s/he would be able to try increasing sensitivity. This however could make it a little bit more likely that the device would accidentally activate on the go. Unfortunately, we were not able to create a successful calibration mechanism that would increase sensitivity without an accidental activation of the effects, therefore, it has been decided to move on as it is and leave a smarter calibration method implementation for the future works.

However much more often the problem was that the person was making the inversion movement too fast and the device was not keeping up with processing the data. Even though the person performed the right move, the device was not activated e.g. did not enter the second state. There were a couple of options how this problem could be solved. For example, NodeMCU could be changed to another microcontroller that could offer higher processing speed. However something much less time consuming was the code optimization and cleaning.

To do that, the "void loop" method had to be redone so that it calculated as little as possible per cycle. First of all, all the code was divided into two parts, each corresponding to the state of the device. The amount of methods used was reduced to a possible minimum. All the serial print statements and comments that were used for debugging were removed. Another problem that was decreasing the efficiency of the code was the use of sleep() method. Mainly it was used for the vibration feedback. Unfortunately it worked so that during the tiny moment then the vibration was on, the device was frozen. This issue was solved by adding a timer parameter, which instead of sleeping allowed it switching off the vibration after certain amount of void main cycles.

6.2. Case studies testing results

Paired t-test showed no significant difference ($p = .187$) between the means of the stompbox effect pedals and MIO effect processor. This means that classical effect pedal design is not significantly better than MIO effect processor design and, thus, they can provide similar experience. For a deeper understanding of the results descriptive statistics have been used.

First, means of both cases were compared to find out, which of the designs was subjectively rated higher. The stompbox effect pedal design had a slightly lower mean ($M = 3.99$, $SD = .635$) than MIO effect processor design ($M = 4.23$, $SD = .529$). This could mean that some of the elements of the classical pedal design were improved in MIO effect processor.

For a deeper analysis of the data, every question has been studied separately with a side by side comparison of both conditions. The results have been structured in the same way as they have been presented in a questionnaire.

6.2.1. Comfort and ease of use

Even though both designs used the same effect activation principle, the MIO effect processor design had its own specifics of the activation, e.g. first, the activation movement and, second, choice of the right effect. As this principle is a bit more complex than just switching on or off right pedal, some musicians needed more time to get used to the controls of the MIO effect processor design. However, one of the test subjects had serious issues with the activation movement. He was not able to understand and perform correct movement, which led to inability to properly use the device. His negative experience influenced the overall results for the comfort of activation and deactivation of the effect (Figure 46) and the mean for the MIO effect processor design ($M = 3.9$, $SD = .99$) compared to classical effect design ($M = 4.6$, $SD = .5$).

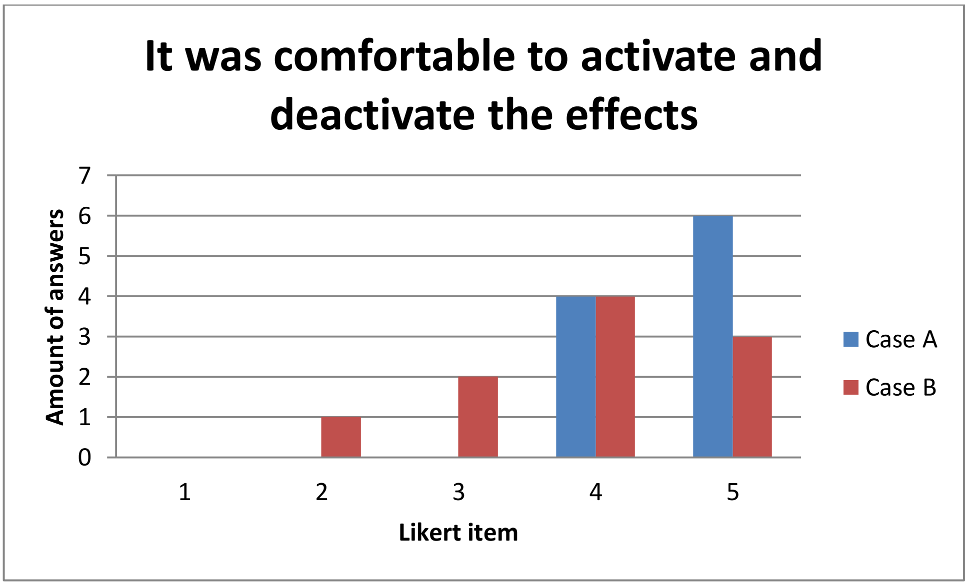

Figure 46. "It was comfortable to activate and deactivate the effects" results

The same issue was relevant to the ease of use of the devices. Since the inversion movement appeared to be hard for one of the test subjects, the use of MIO effect processor also became hard for him. However, it can be clearly seen that he was the only one who found the MIO effect processor to be hard to use (Figure 47), with the means for the stompbox effect pedal design (M = 4.5, SD = .88) and for the MIO effect processor design (M = 4.3, SD = .94) accordingly.

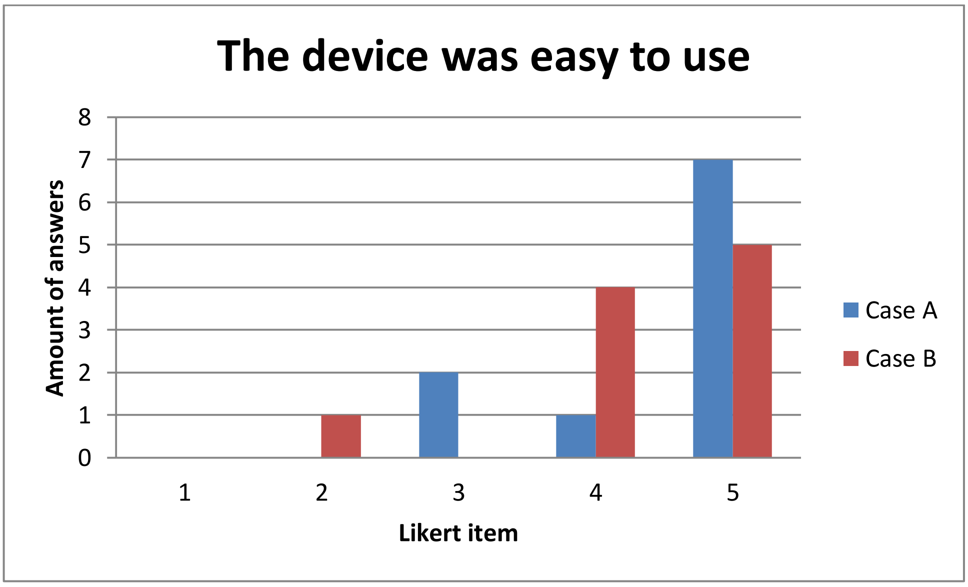

Figure 47. "The device was easy to use" results

6.2.2. Reliability

One of the most crucial and important questions of the whole questionnaire was related to the reliability of the device. Stompbox effect pedals are widely known because of their reliability and stability. It would be impossible to compete with the classical design and use MIO effect processor during the real show, if it would not be reliable enough. Even though its functions were previously tested on the usability testings, reliability can be revealed only through the constant use of the device.

Despite the improvement of the code, there were still rare errors present in usability of MIO effect processor (Figure 48). The results for the stompbox effect pedals and MIO effect processor design are present respectively (M = 4.2, SD = .48), (M = 3.7, SD = 1.33). From the observations it could be concluded that the reliability differed from person to person. This could lead to an idea that, since test subjects were using this device for the first time in their lives, some of the errors could be caused by incorrect or imperfect activation movement. Moreover, from the interviews that will be described in the chapter 6.2.5., it was possible to conclude that the comfort and effectiveness of the activation movement strongly depended on the musicians' general control over their body. A deeper analysis regarding the reliability is required.

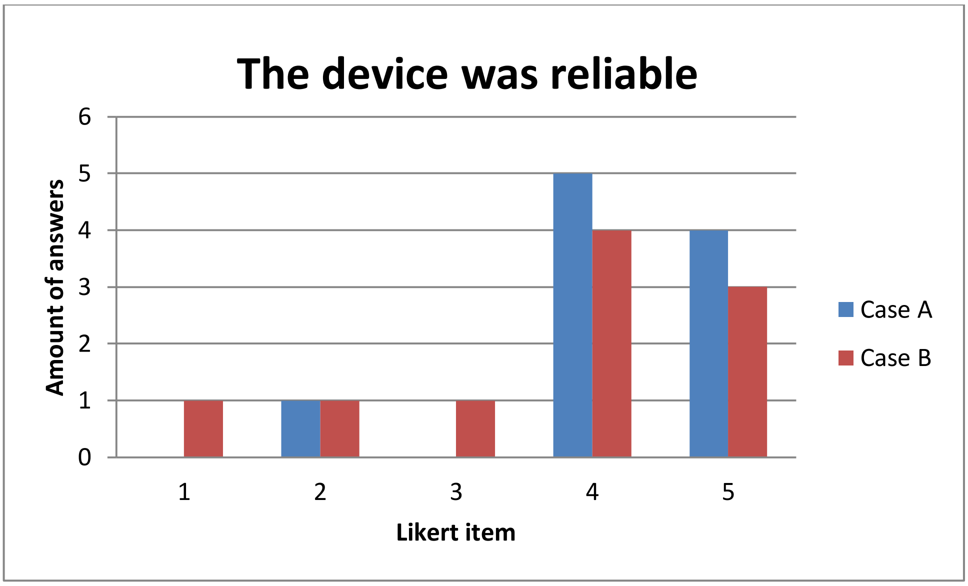

Figure 48. "The device was reliable" results

6.2.3. Feedback and intuitiveness

Due to the additional activation movement of the MIO effect processor design, which was supposed to be the one people tend not to perform on purpose, the intuitiveness of the design suffered compared to the classic effect pedal design (Figure 49). However, it was expected that it would be hard to compare intuitiveness of the brand new design and stompbox effect design, which became classics, used for a decades. It was obvious that, while stompbox effect pedals were previously actively used by all test subjects, MIO effect processor design was innovative and required feature presentations. However, the mean of the MIO effect processor design could still be considered high enough (M = 4.1, SD = .73), even when compared to classic effect pedal design (M = 4.7, SD = .48).

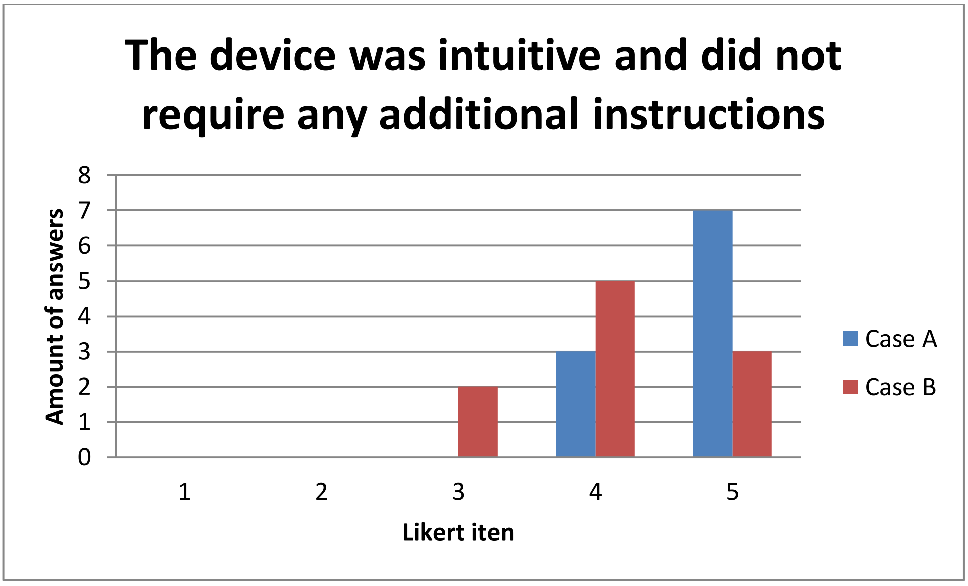

Figure 49. "The device was intuitive and did not require any additional instructions" results

One of the aspects of the intuitiveness of the MIO effect processor design was feedback. There were two types of feedback implemented in MIO effect processor design – visual and haptic feedback, not taking into consideration auditory feedback, which is caused by the principle of the device itself. Clear feedback can significantly improve user experience and make the design more user-friendly. At the same time, even if the design is clear and intuitive, lack of clear feedback can completely destroy the positive experience. Thus, questions regarding both haptic and visual (Figure 50) feedback were included in the questionnaire. The results of the questionnaire showed a significantly higher mean in visual feedback for the MIO effect processor design (M = 4.9, SD = .31) compared to stompbox effect pedal design (M= 3.8, SD = 1.31). The same principle applies to haptic feedback, being much better perceivable in MIO effect processor design (M = 4.5, SD = .7) then in stompbox pedal design (M = 3.3, SD = 1.56).

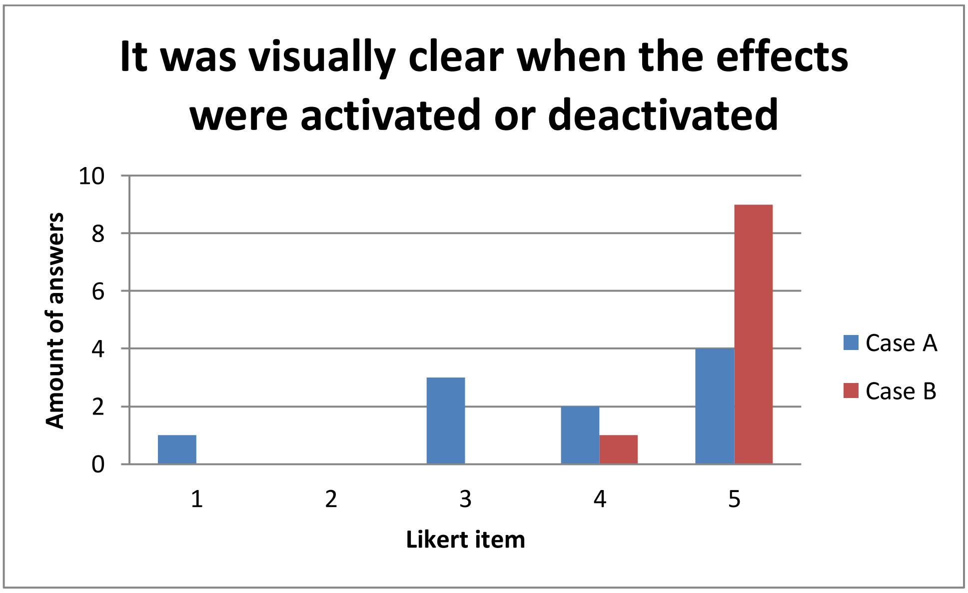

Figure 50. "It was visually clear when the effects were activated or deactivated" results

6.2.4. Setting adjustment

The last part of the questionnaire was focused on the setting adjustments. It was one of the main aspects of the project that should have been improved, and it was necessary to find out, whether the chosen direction improves the setting adjustment experience. The results (Figure 51) show easier process of setting adjustment for the MIO effect processor design (M = 4.3, SD = .94), compared to classic effect pedal design (M = 4, SD = 1.05).

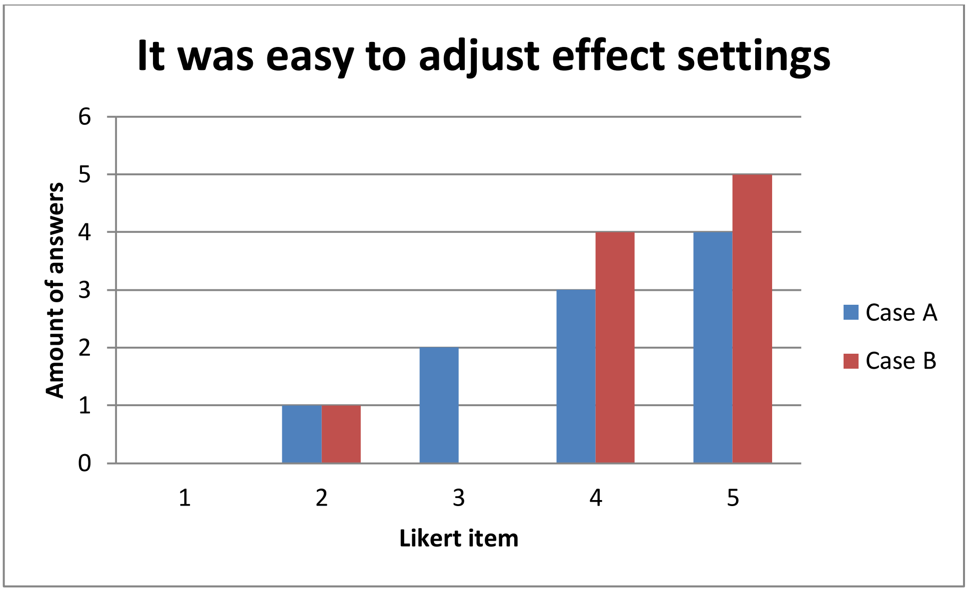

Figure 51. "It was easy to adjust effect settings" results

The comfort of the setting adjustments was also rated higher for the MIO effect processor design (M = 3.8, SD = 1.22) compared to the stompbox pedal design (M = 3.1, SD = 1.19). However, as it can be seen from the Figure 52, the variance is big for both cases, which means that there is a room for improvement of the setting adjustment, which was discussed during the interviews (see chapter 6.2.5.).

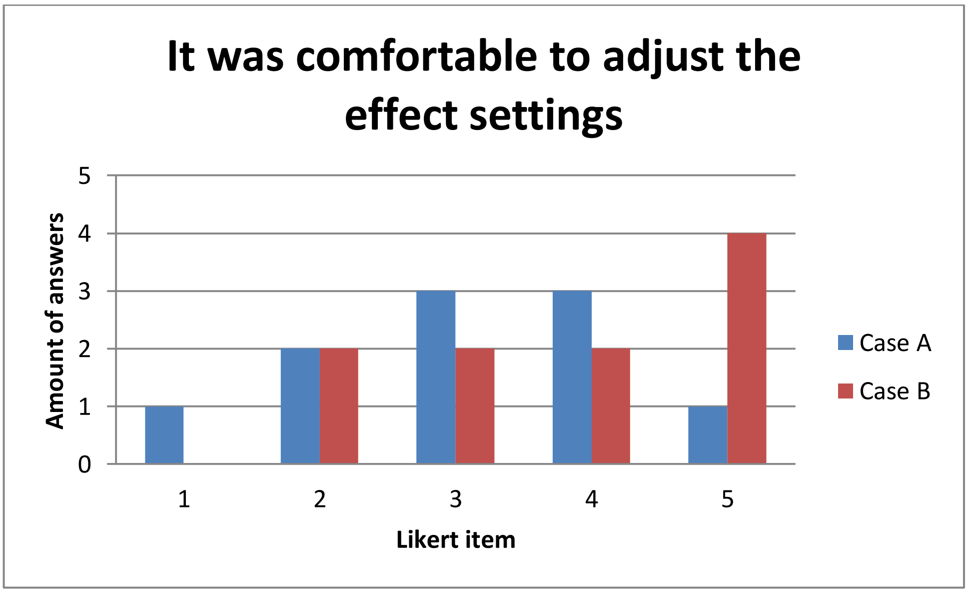

Figure 52. "It was comfortable to adjust the effect settings" results

Finally, it was interesting to find out, whether the MIO effect processor design functions could be satisfying enough to be competitive to classic effect pedal design and to be actually used during a real show. The results (Figure 53) appeared to be satisfying with the MIO effect processor design (M = 3.9, SD = .87) showing equal results to stompbox effect pedals (M = 3.9, SD = 1.2) and being possible to use during a real show.

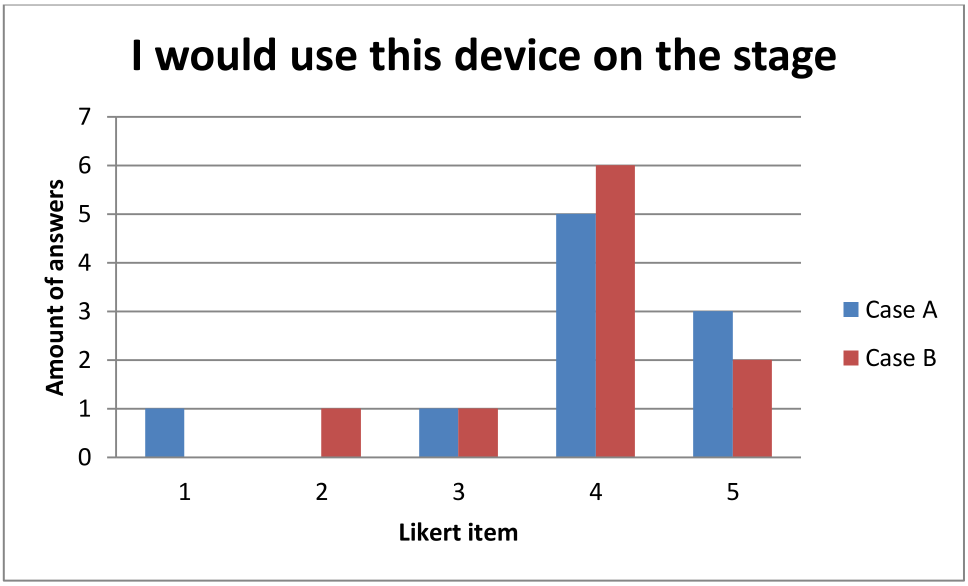

Figure 53. "I would use this device on the stage" results

6.2.5. Qualitative interviews

After the test every test subject has been asked to participate in a semi-structured interview to gather more in-depth view on the possible issues of the MIO effect processor design.

During the interview it has been revealed that it was relatively easy and fast to get used to the activation movement, which only increased the confidence that the choice of the movement was made correctly. Moreover, test subjects have expressed that they would like to use MIO effect processor design in a real setup during the show, as it indeed could solve some of problems they face during their shows. One of the test participants noted that, when playing solo, he tends to move closer to the sides of the stage to interact with a crowd, which forces him to run back to the pedal board right afterwards. This undoubtedly creates discomfort and frustration. In his opinion, MIO effect processor design could solve this issue.

However, to make MIO effect processor design possible to use during a real-life setup, it was necessary to increase the reliability level. To be more specific, sometimes the device didn't switch on after the inversion movement. This issue had two reasons. First, the test participants were performing the inversion movement too fast, and the device was not keeping up. The solution lies in the further code optimization, making it calculating the data faster, which would result in a better sensitivity of the device. Another problem was that different users were performing the inversion movement with the different level of success due to natural flexibility of their joints. The solution to this problem might be the possibility to calibrate the device (using the Mobile App for instance), allowing to change the sensitivity.

Next, some test participants suggested that the device could be made smaller and more robust. Their main concern was that even after they were told that they can move freely, they were still acting very accurate because they were unsure that the device will survive. Therefore it is important to make sure that the user trusts the device from both functionality and rigidity perspectives.

6.3. Mobility test results

Wilcoxon test showed significant difference between means of stompbox effect pedals and MIO effect processor ($p < .002$). When comparing means for both cases, the MIO effect processor shows better results ($M = 4.46$, $SD = .52$) than stompbox effect processor ($M = 3.76$, $SD = .58$). For a deeper understanding of the data, descriptive statistics were used. Since there were three songs per effect device and only five Likert items, it was decided not to average the results, but to sum up all responses per device. Thus, we ended up having 36 responses per each device.

6.3.1. Questionnaire

In general, song performed with MIO effect processor were considered to be more energetic ($M = 4.5$, $SD = .65$) compared to the ones performed with stompbox effect pedals ($M = 3.97$, $SD = .81$) (Figure 54).

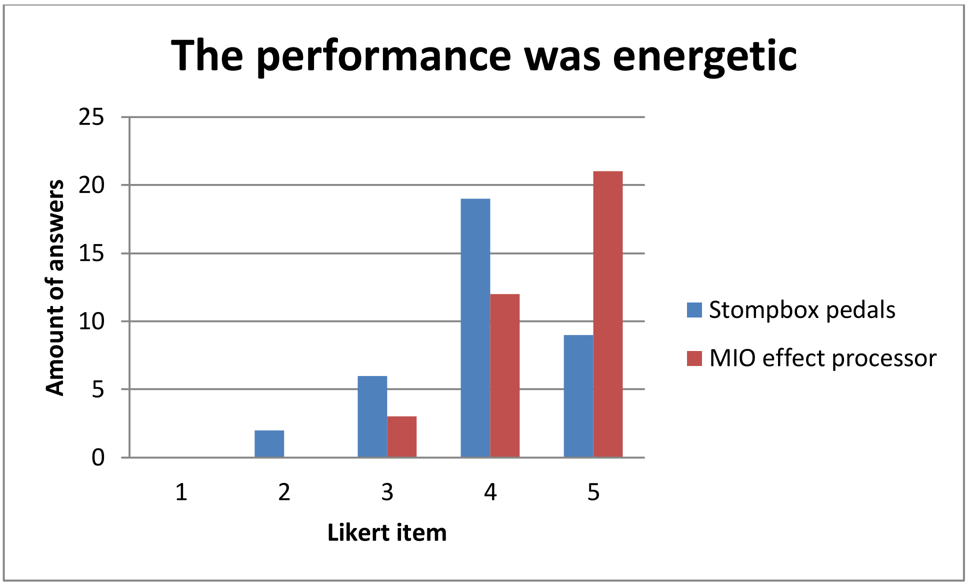

Figure 54. "The performance was energetic" results

The crowd also felt like musicians interacted less with them when using classic effect pedals design ($M = 3.4$, $SD = 1.1$), while during the songs played using MIO effect processor they felt like the interaction was stronger ($M = 4.35$, $SD = .76$) (Figure 55).

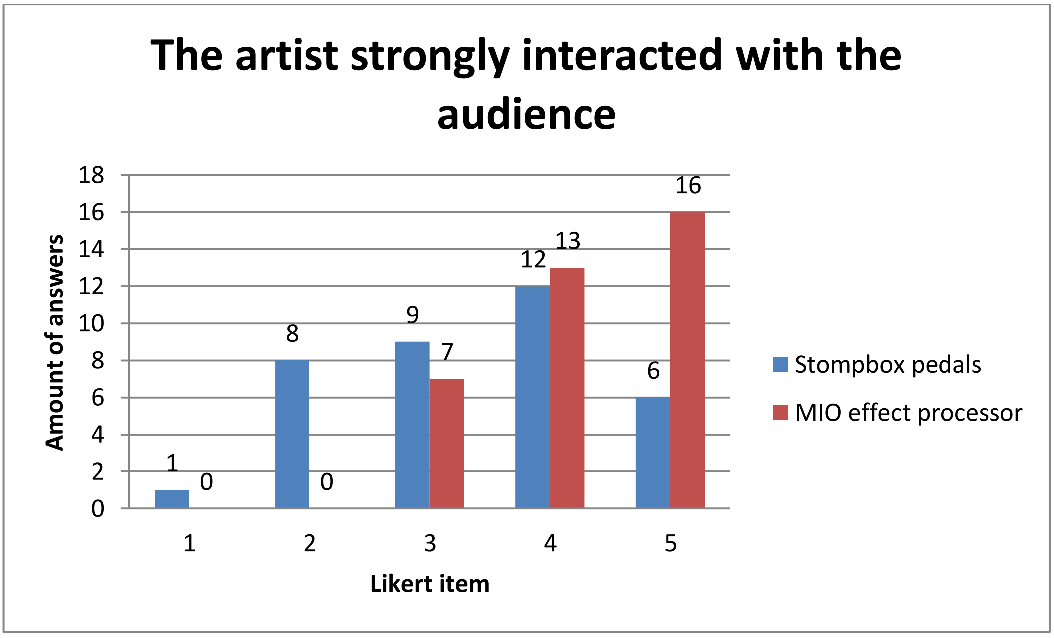

Figure 55. "The artist strongly interacted with the audience" results

Both energy and interaction with an audience on the stage can be used to convey feeling and emotions during the performance. Since both previous questions showed better results for the MIO effect processor design, it was predictable that MIO effect processor would show higher mean (M = 4.27, SD = .84) than stompbox effect pedal design (M = 3.7, SD = .74) (Figure 56).

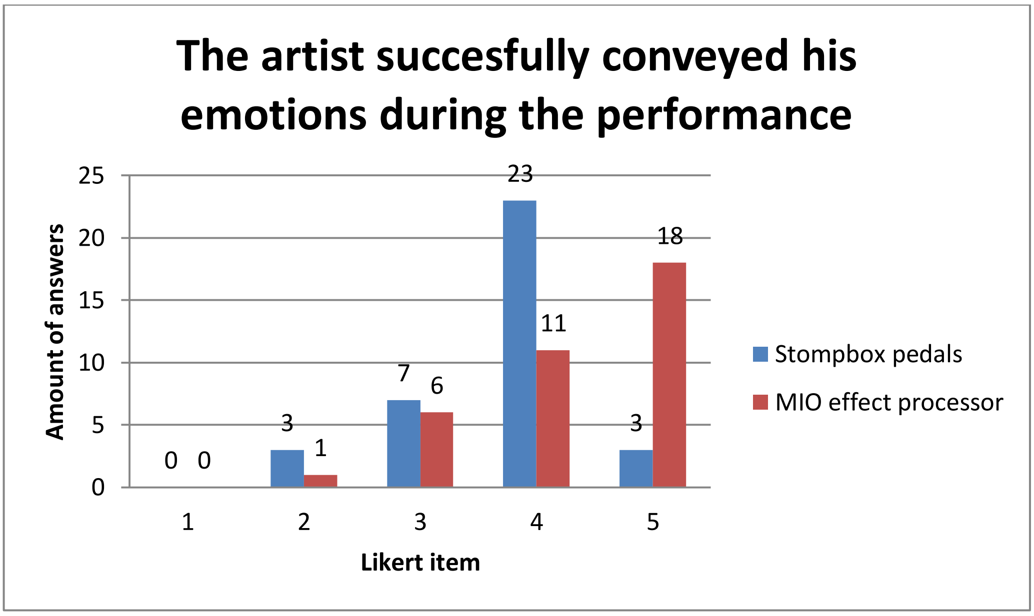

Figure 56. "The artist successfully conveyed his emotions during the performance" results

The main question of the whole mobility testing session was to find out, if musicians would be able to move around more on the stage without necessity to return to the same place again and again. The crowd was asked to share their opinion regarding mobility of the musicians on the stage. The MIO effect processor showed much higher rate of the mobility on the stage (M = 4.5, SD = .65) than classic effect pedal design (M = 3.7, SD = .94) (Figure 57).

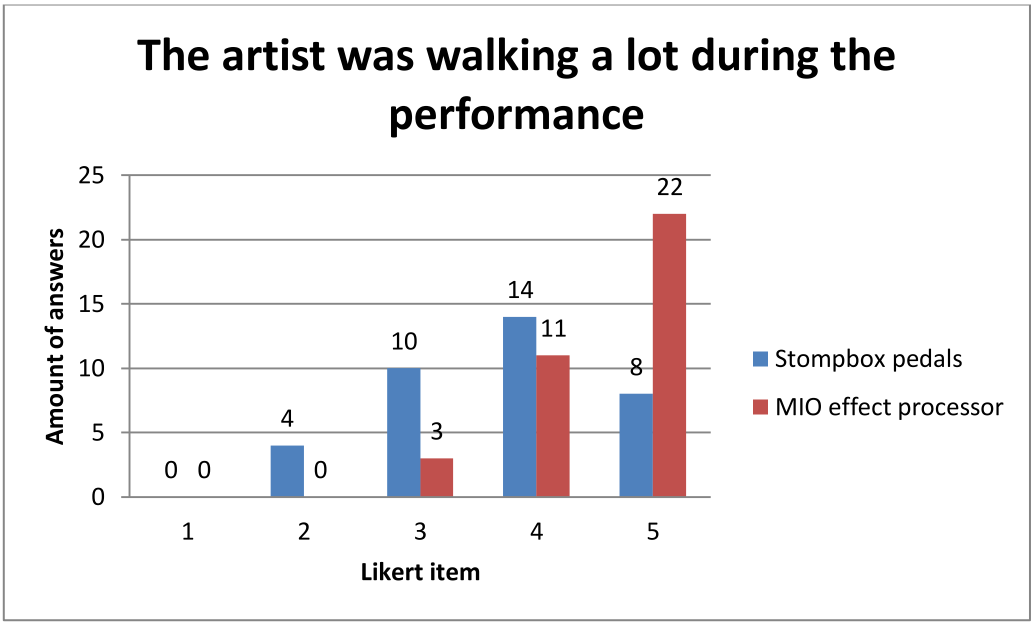

Figure 57. "The artist was walking a lot during the performance" results

Finally, it was interesting to find out, if increased mobility of the musicians on the stage could actually enhance the visual aspects of the show. The crowd rated the visual performance with the use of MIO effect processor much higher (M = 4.7, SD = .52) then performance, where stompbox effect pedals were used (M = 3.8, SD = 1.1) (Figure 58). This could be due to the reason that one of the guitar players actually jumped off from the stage into the crowd and performed like this, controlling his effect even while being out of the stage.

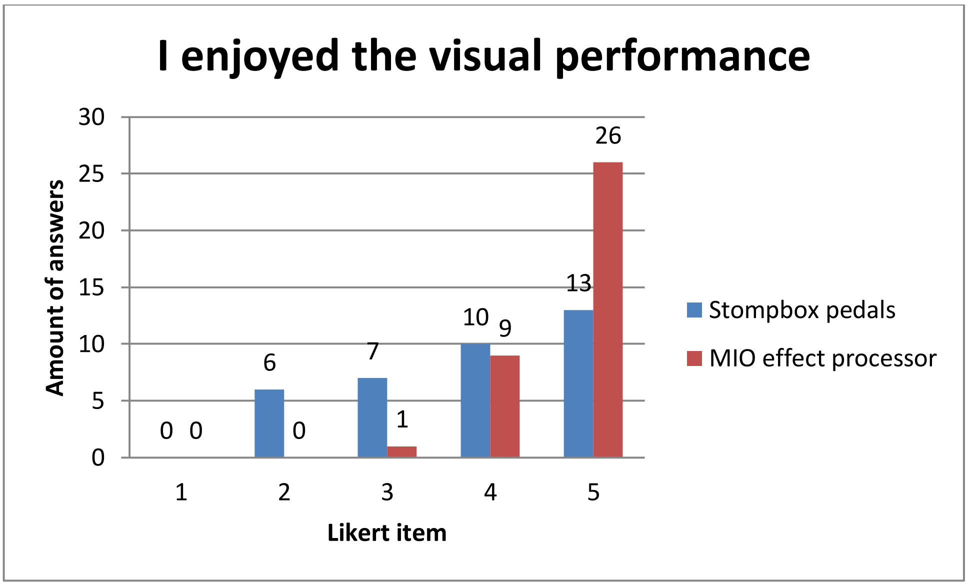

Figure 58. "I enjoyed the visual performance" results

6.3.2. Interviews with the test participants

After the performance test participants were asked to share their thoughts regarding the visual aspects of the show. The questions were related to the differences between the performances, their visual aspects and interaction.

It was interesting that none of the test participants were connecting enhanced visual appeal of the performance to the mobility of the musicians. They perceived it as if musicians were more confident, when playing some of the songs and, therefore, could convey more feelings and emotions to the audience. At the same time, when feeling less confident, it felt like they were too focused on their performance and were not able to interact with the crowd that much.

6.3.3. Interviews with the musicians

After the performance, musicians were asked to share their thoughts about two devices. Concept-wise, the MIO effect processor was enjoyed more due to the fact that it eliminated the necessity to keep in mind the position on the stage for a timely activation or deactivation of the effect. However, this device was lacking reliability. Twice during the six songs the device started glitching, creating a feeling that it entered the endless loop that was constantly sending some data to the DSP unit, which has never previously happened during both usability testings and case studies. Therefore, even though this device reduced the necessity of thinking about the position on the stage, it created another issue, which had to be thought of, e.g. if the device had accidently entered the activated state.

However, main goal of the MIO effect processor, e.g. increased mobility on the stage, was reached. The guitar players were less concerned about their position and could focus more on the performance itself, walk around, reach towards the audience and even jump off the stage into the crowd, which, consequently, allowed them much stronger interaction with the audience and greater degree of immersion in the show.

To sum up, the results of the mobility testing repeated the results of the case studies. Both tests revealed increased mobility on the stage, but lack of reliability of the MIO effect processor. The concept was appreciated both by musicians and by audience and positively influenced the performance of the guitar players.

7.Discussion

The testing of MIO effect processor has revealed both advantages and inconsistencies. First of all, summing up the results from all tests, it got clear that the Foot Controller had some stability issues. It was first revealed during the usability testing, and it was attempted to be fixed before the case studies and mobility test by optimizing the code. Unfortunately the issue was not completely solved, and the device was still not reliable enough to be 100% trustworthy. The main problem was that the Foot Controller didn't enter the activated state. The root problem was that the device still was not able to process the data fast enough. Therefore, the device was not capable of correctly interpreting fast movements. Moreover, the MIO effect processor cannot be used in a signal chain that has another unit that provides audio latency (for example, Variax technology). In this case, the latencies will sum up, becoming perceivable. The possible solution is the further code optimization and, perhaps, the utilization of another microcontroller that has faster CPU.

The general concept, however, was highly appreciated by the test participants. The activation movement once again proved itself to be the right choice, since test participants agreed that it was fast to get used to it and easy to adapt to. Speaking of visual and haptic feedback, MIO effect processor was an obvious winner. The majority of the test participants noted that it was very clear when the Foot Controller was activated, which effects were on and off etc., which, without any doubts, made a positive impact on the whole user experience. The effect setting adjustment using a Mobile App was estimated better than in case of classical effect pedal design. The majority of test participants found it very pleasant to be able to adjust setting using a mobile phone, without a need to constantly sit down.

One of the possible improvements that were requested by test participants was an ability to upload the necessary effects to the library. This would create limitless opportunities for the guitar players in terms of finding the right sound. Moreover, a great addition would be an ability to combine different effects together, thus, building a completely new and unique effect.

Finally, the preset options appeared to be highly appreciated and actively used during the performance. Some of the musicians had a separate preset for each song. To make long-lasting live shows even easier, a preset selector could be implemented, e.g. the opportunity to change the preset with the foot and not only the Mobile App.

To conclude, the MIO effect processor could become a true competitor to classical effect pedal design. Results have also revealed that musicians would equally like to use both designs on the stage, if the stability and reliability would be improved. This means that MIO effect processor indeed can be a potential substitute for classical pedal effects when it comes to a stage performance.

8. Conclusion

With the evolution of digital technologies and constantly increasing processing power, new horizons for audio effects related product development are open. This contributes to both brand new concepts, and redesigns of existing well-known devices. The possibilities created by technological growth led to our redesign of stompbox effect pedals, which, despite being time-tested and widely-used, still had some limitations. Some of the key problems of existing designs were the necessity to constantly return to the effect pedals during the show, sit down to adjust the settings of the effects and other small issues such as price, wired connectivity etc. All together, these have a negative influence on the visual aspects of the performance, as well as creating discomfort and reducing the mobility of the guitar and bass players on stage. This led to the final problem statement for this project:

How can we create an effect processor for guitar/bass players that would be competitive with classical stompbox effect pedal design, provide the same level of experience, meanwhile increasing mobility on stage?

As a possible solution, a wireless foot-activated mobility increment-oriented effect processor has been developed together with a mobile app for ease of effect setting adjustments. The device has been tested in terms of usability, concept and mobility. The usability testing revealed issues with the reliability and stability of the device. The comparison of the MIO effect processor design and classical stompbox effect pedal design showed no significant difference between the devices ($p = .187$), which could be seen as the competitiveness of both designs. The results also revealed the additional comfort in setting adjustment of MIO effect processor ($M = 4.3$, $SD = .94$), compared to classic effect pedal design ($M = 4$, $SD = 1.05$). Finally, the mobility test proved MIO effect processor to provide more mobility on the stage ($M = 4.7$, $SD = .52$) than traditional stompbox effect pedals ($M = 3.8$, $SD = 1.1$), which could possibly lead to increased interaction with the audience, easier conveyance of emotions, and improved visual aspects of the show (more data required).

To sum up, the concept of the MIO effect processor showed the ability to solve issues and limitations of the stompbox effect pedals, such as mobility on the stage and setting adjustment, at the same time providing the same level of functionality and comfort. However, for the device to be fully competitive with stompbox effect pedals, reliability and stability should be improved.

9. Future works

Even though MIO effect processor has performed well during the testing, it still had great potential for improvement. First of all, the stability of the Foot Controller should be increased. Unfortunately, despite all the attempts to optimize the code of the device, it still occasionally failed to either get activated, or it activated when it wasn’t supposed to. As it was mentioned earlier, perhaps a better microcontroller can be used to process information faster. Else, some smart calibration functionality can be implemented.

Another interesting idea was to make an “experimental mode”, by allowing the user to make a completely new effect by stacking existing effects one onto another, and save it as a new effect. This could make the whole setup much more versatile. In addition to that, it would be nice to be able to upload new effects to the effect library.

One more important feature that was requested by a couple of test participants was the ability to change presets using the Foot Controller. This comes from real life scenarios when musicians have individual preset for each song. As it is now, they would have to change presets using the Mobile App. Additional movement can be added to the Foot Controller that would allow changing presets on the go.

Some test participants noted that even though the Foot Controller works fine, it looks like it is fragile, and they subconsciously were trying to be very careful with it. It was proposed to make it smaller and more durable-looking, so that the user could act as crazy as h/she wants without being worried that the device will break into parts.

Finally, the quality of the effects still has room for improvement. All time-based effects, such as delay, reverb, chorus etc., sounded acceptable, but distortion and fuzz effect left something more to be desired. For example, it can be done by using impulse responses of popular distortion pedals / guitar amplifiers.

Bibliography

Adafruit (2019). Retrieved 21.06.2019 from: https://www.adafruit.com/product/1876

Adepoju, A., (2018). *The differences between Bluetooth and WiFi*. Dignited. Retrieved 21.06.2019 from: https://www.dignited.com/35868/the-differences-between-bluetooth-and-wifi/

Avison, J., (2014). The World of Physics. *Juvenile Nonfiction*

Baxter-Moore, N., & Kitts, T. M. (2016). The live concert experience: an introduction.

Baxter-Moore, N., & Kitts, T. M. (2016). The live concert experience: an introduction.

Bazil E. (2009). Sound Equalization Tips and Tricks. PC Publishing.
Brown, J. D. (2011). Likert items and scales of measurement. Shiken: JALT Testing & Evaluation SIG Newsletter, 1, 10–14.

Brown, S. C., & Knox, D. (2017). Why go to pop concerts? The motivations behind live music attendance. *Musicae Scientiae*, *21*(3), 233-249.

Brown, S. C., & Knox, D. (2017). Why go to pop concerts? The motivations behind live music attendance. *Musicae Scientiae*, *21*(3), 233-249.

Christophersen M., et al. (2007). *Guitar Effects Unit*

Danielsen, A., & Helseth, I. (2016). Mediated immediacy: the relationship between auditory and visual dimensions of live performance in contemporary technology-based popular music. *Rock Music Studies*, *3*(1), 24-40.

Endsley, M. R. (2016). *Designing for situation awareness: An approach to user-centered design*. CRC press.

Erlhoff, M., & Marshall, T. (Eds.). (2007). *Design dictionary: perspectives on design terminology*. Walter de Gruyter.

Fant-Saez G. (2006). Pro Tools for musicians and songwriters. Peachpit
Field, A., & Hole, G. (2002). How to design and report experiments. Sage.

Free VST Plugins (2019). Retrieved 15.05.2019 from: http://vstplanet.com/

Fries, B., & Fries, M. (2005). *Digital audio essentials*. " O'Reilly Media, Inc.".

Frye, G. J. (2001). Testing digital and analog hearing instruments: Processing time delays and phase measurements. *The Hearing Review*, *8*(10), 34-40.

Gadol III, W. N. (2016). *U.S. Patent No. 9,240,172*. Washington, DC: U.S. Patent and Trademark Office.

Gan, W. S., & Kuo, S. M. (2007). *Embedded signal processing with the micro signal architecture*. John Wiley & Sons.

Gilreath P., Aikin J., et al. (2004). The Guide To MIDI Orchestration. *Musicworks.*

Google Play (2019). Retrieved 21.06.2019 from: https://play.google.com/store/apps/details?id=com.presonus.dawremote&hl=en
App Store (2019). Retrieved 21.06.2019 from: https://itunes.apple.com/us/app/logic-remote/id638394624?mt=8

Guitar Extended, 2019. Retrieved 21.06.2019 from: https://guitarextended.wordpress.com/audio-effects-for-guitar-with-pure-data/?fbclid=IwAR1osohSCYWHt3Xyf3DUjYm4I64mZTL0QXPsOGue69R-ryKTlBvtNqCI19Q

Hammond M., Rabinowitz K., & Alldis D. (2009). Guide to performing: Singing. Guardian News and Media Limited or its affiliated companies, Retrieved 21.06.2019 from: http://www.theguardian.com/uk.

Help.ableton.com (2019). Retrieved 21.06.2019 from: https://help.ableton.com/hc/en-us/articles/209071989-Apps-for-controlling-Live-with-an-iOS-or-Android-device

Hosken, D. (2014). *An introduction to music technology*. Routledge.

Hunter, D. (2014). *Guitar Amps and Effects For Dummies*. John Wiley & Sons.
Jam Origin (2019). Retrieved 15.05.2019 from http://www.jamorigin.com/products/midi-guitar-1/

Knapp, J., Zeratsky, J., & Kowitz, B. (2016). *Sprint: How to solve big problems and test new ideas in just five days*. Simon and Schuster.

Kurve, A. (2017). *10 Best Raspberry Pi alternatives you can buy*. Beebom. Retrieved 21.06.209 from: https://beebom.com/best-raspberry-pi-3-alternatives/

Leuschenko, O., (2018). *The Pros and Cons of Xamarin for Cross-Platform Developmen.* Retrieved 21.06.2019 from: https://hackernoon.com/the-pros-and-cons-of-xamarin-for-cross-platform-development-2a31c6610792

Lowdermilk, T. (2013). *User-centered design: a developer's guide to building user-friendly applications*. " O'Reilly Media, Inc.".

Mack, C. (2015). The multiple lives of Moore's law. *IEEE Spectrum*, *52*(4), 31-31.

Mack, C. A. (2011). Fifty years of Moore's law. *IEEE Transactions on semiconductor manufacturing*, *24*(2), 202-207.

McPherson, A. P., Jack, R. H., & Moro, G. (2016). Action-sound latency: Are our tools fast enough?.

Millard A. (2004). The electric guitar: a history of an american icon. JHU Press.

Millward, S. (2005). *Fast Guide to Cubase SX*. PC Publishing.

Morris, T., & Tomasi, C. (2017). *Podcasting for dummies*. John Wiley & Sons.

Newnes.
Nonzee, V., & Poongbunkor, P. (2001). DSP audio effects. *ECE 320 Final Project Paper*.

Orkin D. (2018). *The best-selling pedals of 2018*. Retrieved 13.04.2019 from: https://reverb.com/news/the-best-selling-pedals-of-2018

Padhye, R., & Nayak, R. (Eds.). (2016). *Acoustic Textiles*. Springer Singapore.

Park, T. H. (2009). *Introduction to digital signal processing: Computer musically speaking*. World Scientific.

Press.
Pure Data forum~, 2019. Retrieved 21.06.2019 from: https://forum.pdpatchrepo.info/

Raspberry Pi (2019). Retrieved 21.06.209 from: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Razali, N. M., & Wah, Y. B. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. Journal of Statistical Modeling and Analytics(2(1)), 21-33.

Reed, J. H. (2002). *Software radio: a modern approach to radio engineering*. Prentice Hall Professional.

Salkind N. J. (2010). Encyclopedia of Research Design.

Sattel, S., (2016). *WiFi vs. Bluetooth: Wireless Electronics Basics.* Autodesk. Retrieved 21.06.2019 from: https://www.autodesk.com/products/eagle/blog/wifi-vs-bluetooth-wireless-electronics-basics/

Sauls, S., & Stark, C. (2016). *Audio production worktext: concepts, techniques, and equipment*. Routledge.
Saunders A. J. (2013), Finding Your Guitar Tone, Kindle Edition.

Self, D., Duncan, B., Sinclair, I., Brice, R., Hood, J. L., Singmin, A., ... &
Share of Americans who attended rock concerts in the past 12 months in 2018, by age. (2018). Statista Survey; Cint © Statista 2018.

Shneiderman, B. (2003). *Leonardo's laptop: human needs and the new computing technologies*. Mit Press.

Statista (2019a). *Average unit price for a guitar effects pedal in the United States from 2005 to 2017 (in U.S. dollars).* United States. The Music Trades © Statista 2019.

Statista (2019b). *Number of guitar effects pedals sold in the United States from 2005 to 2017 (in thousands).* United States. The Music Trades © Statista 2019.

Stettler M., & Krishnapura S. (2016). Moores law not dead and intels use of hpc to keep it alive. HPCWire.

Vst 4 free (2019), Retrieved 15.05.2019 from: http://vst4free.com/

Watkinson, J. (2009). Audio engineering: Know it all: Know it all. (vol. 1).
Wei, L., & Moyer, M. G. (Eds.). (2009). *The Blackwell guide to research methods in bilingualism and multilingualism*. John Wiley & Sons.

Figures

Figure 1. Retrieved 21.06.2019 from: https://www.fieldingdsp.com/alias

Figure 2. Retrieved 21.06.2019 from: https://www.researchgate.net/figure/Color-online-Quantization-of-a-sinusoidal-signal-and-the-corresponding-quantization_fig6_50250631

Figure 3. Retrieved 21.06.2019 from: http://mmi504.pbworks.com/w/page/9136646/Erik%20And%20Jeff

Figure 4. Retrieved 21.06.2019 from: https://se.mathworks.com/help/audio/examples/delay-based-audio-effects.html#d117e1364

Figure 5. Retrieved 21.06.2019 from: https://www.ligo.caltech.edu/LA/page/what-is-interferometer

Figure 6. Retrieved 21.06.2019 from: https://store.arduino.cc/arduino-uno-rev3

Figure 7. Retrieved 21.06.2019 from: https://www.quora.com/What-is-NODEMCU

Figure 8. Retrieved 21.06.2019 from: https://www.pjrc.com/store/teensy35.html

Figure 9. Retrieved 21.06.2019 from: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

Figure 10. Retrieved 21.06.2019 from: https://beagleboard.org/bone

Figure 11. Retrieved 21.06.2019 from: https://beebom.com/best-raspberry-pi-3-alternatives/

Figure 12. Retrieved 21.06.2019 from: https://beebom.com/best-raspberry-pi-3-alternatives/

Figure 13. Retrieved 21.06.2019 from: http://www.bestguitareffects.com/boss-ds-1-distortion-pedal-review-best-guitar-distortion-pedal/

Figure 17. Retrieved 21.06.2019 from: https://www.musicradar.com/news/guitars/ibanez-unveils-korg-kaoss-pad-equipped-guitar-and-bass-608183

Figure 18. Retrieved 21.06.2019 from: http://lividinstruments.com/products/guitar-wing/

Figure 19. Retrieved 21.06.2019 from: https://www.kickstarter.com/projects/588998327/acpad-the-electronic-orchestra-for-your-guitar

Figure 20. Retrieved 21.06.2019 from: https://line6.com/variax-modeling-guitars/shuriken/

Figure 21. Retrieved 21.06.2019 from: https://line6.com/helix/

Figure 22. Retrieved 21.06.2019 from: https://www.ikmultimedia.com/products/irigstompio/

Figure 23. Retrieved 21.06.2019 from: https://www.sourceaudio.net/hot-hand.html

Figure 24. Retrieved 21.06.2019 from: https://www.keithmcmillen.com/products/softstep/