Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Elektro- und
Informationstechnik

Electrical- and Information Engineering-
Information Engineering

Project

A digital Guitar Effects Pedal or what the modern Hendrix would use

Author: Markus Velm
Semester: Two
Professor: Prof. Dr. Manfred Litzenburger

List of Abbreviations

ADC Analog-Digital Converter
CLI Command Line Interface
CPU Central Processing Unit
DAC Digital-Analog Converter
DMA Direct Memory Access
EDA Electronic Design Automation
FFT Fast Fourier Transform
FIR Finite Impulse Response
IC Integrated Circuit
IDE Integrated Development Environment
IIR Infinte Impulse Response
I^2C Inter-Integrated Circuit
I^2S Inter-Integrated Sound
LTI Linear Time Invariant
LTV Linear Time Variant
MSB Most Significant Bit
PCB Printed Circuit Board
SPI Serial Peripheral Interface

Contents

1. Introduction

Guitar pedals are widely used. Classic ones are analog but there is an increasing number of digital effects due to processing power of PCs or microcontrollers. Both concepts have reasonable advantages. For the analog concept the unbeatable advantage is that there is minimal latency. The response of an analog system is the fastest you can get. The downside of analog processing is the lack of flexibility.

This project is about an implementation of a guitar effect on a microcontroller based platform. The effect which should be realized is variable bandpass-filter. Due to its acoustic characteristics it is also called »Wah-Wah«-effect. Guitar legends such as Jimi Hendrix, especially on the song Voodoo Child (Slight Return), and also Eric Clapton used this effect a lot, often in combination with a heavy distortion, also called *Fuzz* [1] [2].

2. Goal

The goal of this project is to implement a digital bandpass filter, which is variable in its center frequency. The core of this should be a STM32 microcontroller.

Additionally, an own Printed Circuit Board (PCB) should be developed to integrate all components onto one single board. There should also be the opportunity to change the software of the microcontroller in order to extend the functionality. Other effects like reverb, delay or other filters should be possible to implement.

3. Basics

3.1. Digital Signal Processing

3.1.1. Digital Filters

3.1.1.1. Finite Impulse Response Filter

Digital Filters are a huge field of digital signal processing. The most common filters are FIR-filters. These are also called »non-recursive filters«. The basic structure is a weighted shift register without feedback. The missing feedback is what makes the FIR-filter stable per construction.

In Figure 3.1 the structure of a FIR-filter is shown.

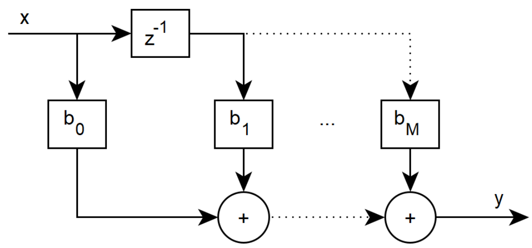

Figure 3.1: Basic structure of a M order FIR-filter in the first canonical form [3]

It can be seen that there are multiple components in this filter. The first component is the delay block, which is described as z^{-1}. This block is responsible for adding a delay of one clock cycle to the input data. Secondly, there are multiplication blocks, noted as b_M. These blocks weight the input data and give the result to an adder, which adds $M + 1$ weighted and delayed signal samples. This sum is the output of an FIR-filter. It also means, that the output is only dependend on the present and the M last input samples and not of the past output.

The input signal is x, whereas the output is y. The order of this Filter is M which means the amount of filter taps or delay blocks.

A special case of the FIR-filter is the *Moving Average Filter*, at which all coefficients are $\frac{1}{M+1}$.

The whole system can be described with the difference equation in the time-domain (Equation 3.1)

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + ... + b_M \cdot x[n-M] \tag{3.1}$$

,

or with the transfer function in the z-domain (Equation 3.2)

$$H(z) = b_0 + b_1 \cdot z^{-1} + ... + b_M \cdot z^{-M} \tag{3.2}$$

.

The output can be calculated with the convolution (noted as $*$) of the impulse response $h[n]$, which is simply the union of the filter coefficients, and the input signal $x[n]$ (Equation 3.3).

$$y[n] = x[n] * h[n] \tag{3.3}$$

Another way to describe the filterung result is in the frequency domain. If the frequency of the input is given with $X(z)$ and the frequency response of the filter is $H(z)$ than the output is $Y(z)$, which can be calculated like shown in Equation 3.4.

$$Y(z) = X(z) \cdot H(z) \tag{3.4}$$

The stability of the FIR-filter is characteristic, this can be seen in the pole-zero plane. All poles are in the middle of the unit circle, which is necessary for the stability of a system.

If the value of the z-plane is evaluated on the unit circle the result is the frequency response of the filter (Figure 3.2).

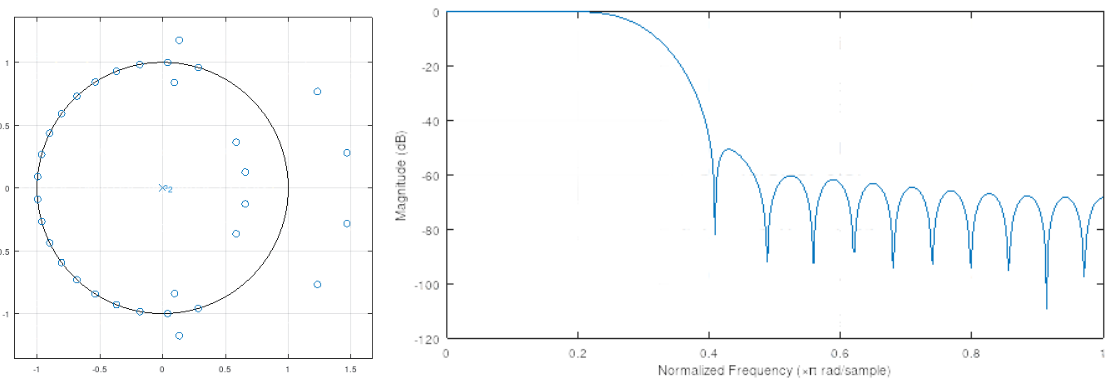

Figure 3.2: z-plane plot (left) and frequency response (right) of a FIR-lowpass-filter

3.1.1.2. Infinite Impulse Response Filter

Another filter type is the Infinte Impulse Response (IIR) filter. The structure is recursive as it is shown in Figure 3.3.

Due to this recursive structure, this type of filter is no longer necessarilly stable. This means, that it can oscillate and possibly the output can grow exponentally. To fullfill stability, all the poles of the IIR-filter have to be in the unit-circle [3].

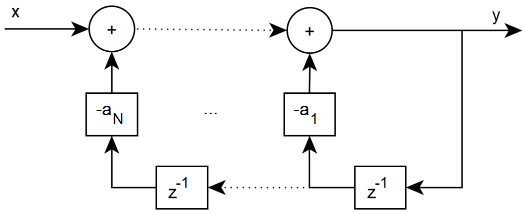

Figure 3.3: Basic structure of a M order IIR-filter in the first canonical form [3]

Its transfer function can be described as follows:

$$H(z) = \frac{1}{a_1 \cdot z^{-1} + ... + a_M \cdot z^{-M}} \tag{3.5}$$

In the time domain, the output is equal to

$$y[n] = x[n] - a_1 \cdot y[n-1] - a_2 \cdot y[n-2] - ... - a_M \cdot y[n-M] \tag{3.6}$$

.

The output is dependend on the current signal and all the M outputs before. In literature and other documentation, the a_0-coefficient is also referenced. This is the amplification of the output signal. Due to filter-stability-reasons, this coefficient is typically set to 1 [4] [5].

If the two types of filters are getting merged, the result is also an IIR-filter, plus a few additional feed-forward-terms. A widely used example of this is the *Biquad-filter* [4].

These filters use two filter-taps in the forward- and two taps in the feedback-path. The structure is shown in Figure 3.4.

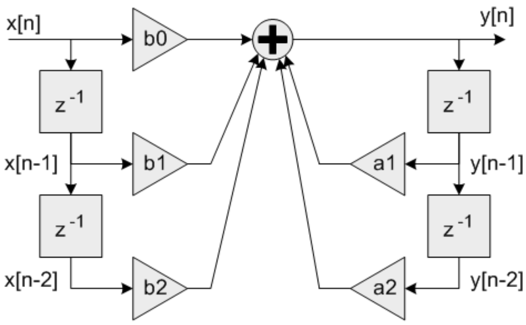

Figure 3.4: Structure of a biquad filter [4]

The transfer function of this filtertype is:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \tag{3.7}$$

Here it can be seen, why this filter is called »biquad«: it is the short form for »bi-quadratic«, which relates to the powers of the numerator- and denominator-terms. Also in this structure, the a_0-coefficient is set to 1.

This transfer function leeds to an output of:

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] - a_1 \cdot y[n-1] - a_2 \cdot y[n-2] \tag{3.8}$$

It is worth noting, that the denominator-coefficients have to be negated to match the structure in Figure 3.4.

3.1.1.3. Filter Design

This section deals with designing filters. There are a few ways to calculate the numerator and denominator-coefficients. But this shouldn't be part of this section, the goal is more like looking at how to design a filter in a high-level perspective.

First of all, the number of filtertaps will be discussed. As stated earlier, FIR-filters have the convenient property, that they are always stable. The downside of these filters are, that for example a narrow-band bandpass filter needs more filtertaps, than a comparable IIR-filter [3]. This should not be a huge problem in the most cases, but if the goal is to variate the filter-coefficients, this may lead to problems.

Secondly, the shape of FIR and IIR-filters differ. The recursive filter is more often used to model an actual analog filter, due to the fact, that analog filters are also of recursive structure (e.g. a second order LC-filter) [3].

In the following graphics, there is a FIR- on the left and a IIR-bandpass-filter on the right. All the parameters of these two filters, mid-frequency, bandwidth, quality-factor and number of coefficients are set equal in order to make them comparable.

The parameters which were chosen for this comparison were set to:

Table 3.1: Parameters to compare FIR- and IIR-filters

start-frequency	stop-frequency	number of coefficients
0.4	0.5	5

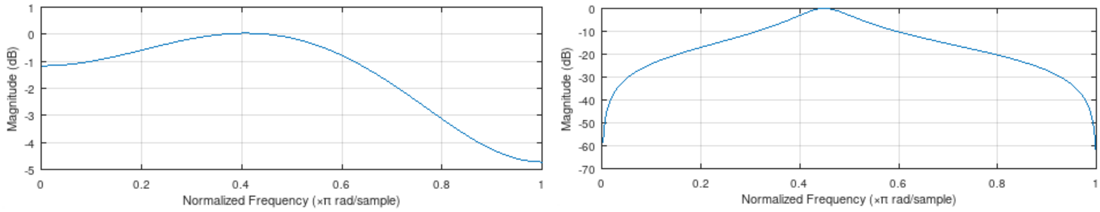

Figure 3.5: Comparison of the frequency response of a FIR- (left) and IIR-bandpass-filter (right)

As it can be observed, the IIR-filter has much sharper edges and the attenuation in the stopband is much higher. This is due to the fact, that IIR-filters have poles, which can produce sharper edges, but it can also lead to instability.

As a reference, an analog bandpass-filter is shown in Figure 3.6, designed with the Texas Instruments filter-design-tool [6].

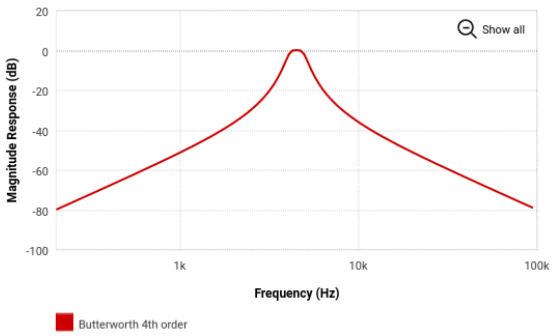

Figure 3.6: Analog bandpass filter frequency response

It can be seen, that the basic shape of the IIR-filter is much closer to the analog equivalent than the FIR-filter.

3.1.2. LTI and LTV-Systems

The filters above are all Linear Time Invariant (LTI)-systems. That means, they firstly have a linear behaviour, so filters can be combined in any way without changing the output. Secondly, the filters never change in time. This is not the case for a variable bandpass, as it changes its coefficients. Therefore it is linear, but no longer time-invariant. Such systems are then called Linear Time Variant (LTV). This may have an effect to the output signal, so this has to be evaluated.

3.2. Microcontroller-Peripherals

3.2.1. Inter-Integrated Sound

The Inter-Integrated Sound (I^2S) protocol was developed by *Philips* to share audio between Integrated Circuits (ICs). A similarity to the Serial Peripheral Interface (SPI) protocol can be recognized.

There are three signal lines:

- SCK: Clock signal
- SD: Data signal
- WS: Word select, for distinction between left and right channel

So the data is transmitted over the same signal line by time division multiplexing. To illsutrate the timing of the protocol the corresponding diagrams are shown in Figure 3.7 [7].

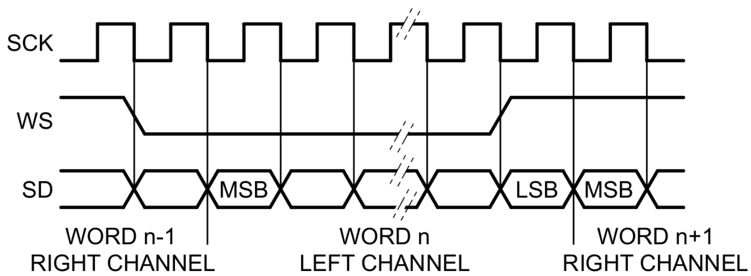

Figure 3.7: Timing diagramm of the I^2S protocol [7]

The word length can vary between transmitter and receiver. That is why the Most Significant Bit (MSB) is sent first in the standard configuration. Additionally the participants do not need to know the word length of the counterpart [7].

The wiring of this serial bus is possible in three basic configurations (shown in Figure 3.8). Thereby the master is always responsible for the SCK and the WS line.

3.2.2. Direct Memory Access

The Direct Memory Access (DMA) is a peripheral in the microcontroller, which allows to transfer data between memory destinations or peripherals without the need of the Central Processing Unit (CPU). This is advantagious, because this saves computation capacity which leads to a smoother data flow.

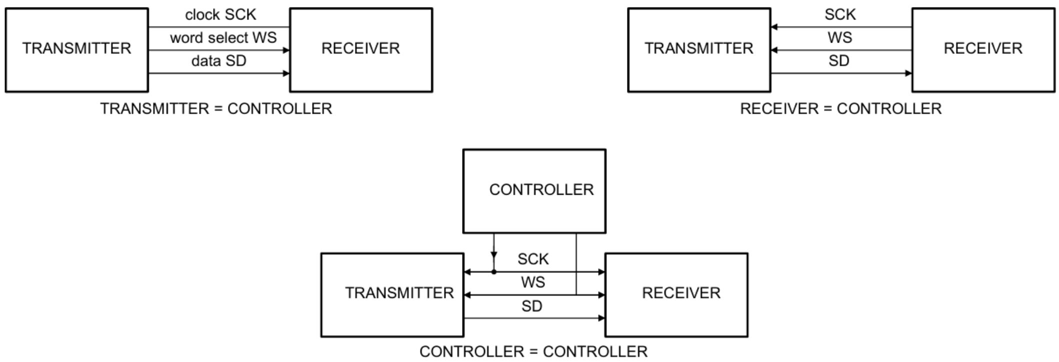

Figure 3.8: The three basic configurations of the I^{2}S protocol [7]

4. Requirements and concept

4.1. Challenges and requirements in Audio Signal Processing

4.1.1. Latency

As we have to sample a block of data in the digital domain, there is a latency between the input and output depending on some parameters. Firstly the buffer size has a large impact on the latency. The greater the buffer the greater the latency. It is application depending what latency is acceptable, this is why that parameter limits the system and should be evaluated carefully.

Secondly other components, which are application depending, adds latency to the whole signal path. For example filtering or other modifications take some time to be computed. Also the signal has to be converted from analog to digital and transmitted over an interface. All these components latencies add up to an overall system latency.

In this application, where a guitar signal is processed, the overall latency of the system should be less than 5 ms [8].

4.1.2. Packet losses and cracking noise

In some audio applications cracking noise can occur. This is due to some impulses on the audio signal. This can be caused by an error of the input sample, for example particles on a vinyl record, or by slow implementations of digital signal processing algorithms.

If the processing of the input data takes longer than the actual transfer, packets of audio data will then be dropped. An impulse or step on the output data will be seen, this can be heard as a cracking noise [9].

4.2. Concept

The basic concept is shown in Figure 4.1.

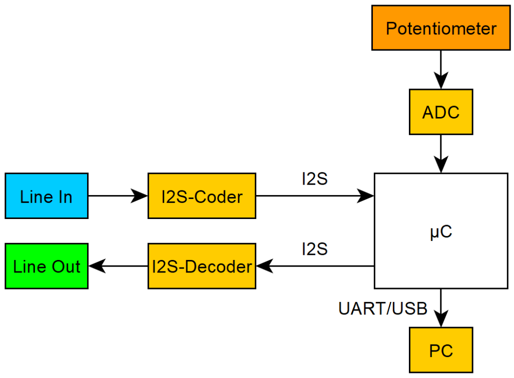

Figure 4.1: Basic concept

The input data is coming from the Line In, is sampled by a Analog-Digital Converter (ADC) and transmitted over I^{2}S to the microcontroller. There the data will be processed and the output will be transmitted again over I^{2}S to a Digital-Analog Converter (DAC). The output is again routed to a Line Jack.

Additionally a potentiometer is connected to set the midfrequency of the filter.

There is also a serial connection to a PC where a Command Line Interface (CLI) should be available to configure different parameters.

4.2.1. Signalflow

The basic idea is to filter the input signal with a bandpass filter. The mid frequency should be derived from a potentiometer which is simply connected to an ADC inside the microcontroller. It has to be evaluated how exaclty the filter is adjusted.

The first attempt will be a lookup-table and everytime the position of the potentiometer changes, the corresponding filter coefficients will be loaded.

The second approach is, to calculate the filter coefficients for a biquad-filter if the value of the potentiometer changes. This is reasonable to evaluate as the number of coefficients is small in comparison to a FIR-filter.

4.2.2. Double Buffering and DMA

To solve the problem with packet losses it is recommended to use the double buffering method. It is basically a buffer which is cut in half. If the first half is filled with data then this half will be processed while the other half is filled with the second half of the input data. Is the second half filled then this half will be processed and so on.

The advantage is that it is not necessary to wait for the whole input data block to process. This will reduce the time between processing and sampling data [10].

To take advantage of this, it is useful to combine this technique with a DMA, which is already discussed in section 3.2.2.

The whole procedure is shown in Figure 4.2.

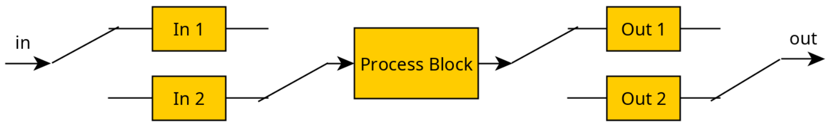

Figure 4.2: Double buffering concept [10]

5. Implementation

5.1. Introduction of the systems used

For the first implementation of basic filtering operations, the evaluation board *Nucleo-H743ZI* of STMicroelectronics is used. The breakout board *PmodI2S Stereo IN/OUT* is used in order to sample audio data from a 3,5 mm jack input.

The choice for the Integrated Development Environment (IDE) fell on the *STM32CubeIDE* as it provides an easy to use installation and development process.

Digital filters need filter coefficients. For the first approach these coefficients where computed beforehand and then stored in the memory of the microcontroller. For these computations, own Python scripts were developed.

In order to design the PCB, the Electronic Design Automation (EDA)-software *KiCAD* was used.

5.2. Implementation of the filters

As the goal is to develop a variable bandpass filter, not only a set of coefficients for one filter has to be stored, but many versions of this filter with different mid-frequencies.

As a reference, the frequency response of *Cry-Baby*-Pedal of Prof. Litzenburger has been measured, this is shown in Figure 5.1.

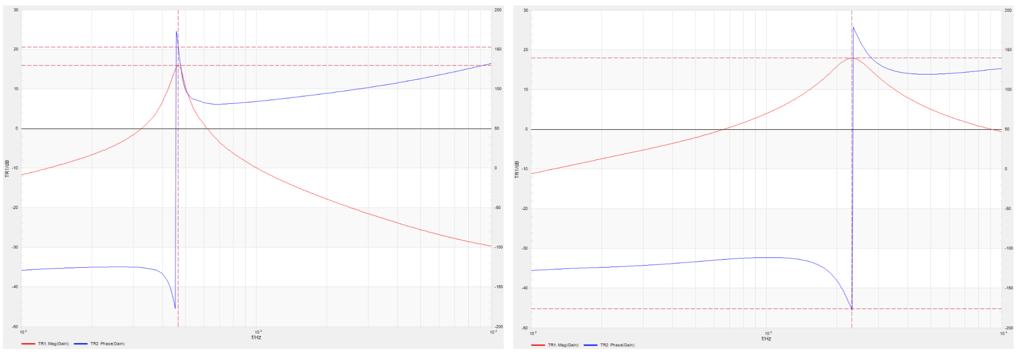

Figure 5.1: Measurement of the frequency response of a Wah-Wah-pedal at the lowest and highest frequencies

The resonance frequency of this filter varies between 466 Hz and 2,3 kHz.

It is also to be seen, that as the frequency changes, the bandwidth also seem to change. Firstly, it has to be mentioned, that this is a logarithmic scale. Nevertheless, in an ideal bandpass-filter, the quality factor is constant. This factor indicates how much energy is dissipated, or how high the loss of a oscillation circuit is. So to keep the Q-factor constant, with increasing the resonance frequency, the bandwidth also have to increase (Equation 5.1) [11].

$$Q = \frac{f_{res}}{f_{3\,dB}} \tag{5.1}$$

By measuring the bandwidths of this filter, it turns out, that the quality factor also changes.

The bandwidth at 466 Hz is measured to be 54 Hz, which leads to a Q-factor of:

$$Q = \frac{460\,Hz}{54\,Hz} = 8,518 \tag{5.2}$$

.

At 2,3 kHz the bandwidth is 600 Hz resulting in a quality factor of:

$$Q = \frac{2,3\,kHz}{600\,Hz} = 3,833 \tag{5.3}$$

.

To conclute this, there are two design criteria which can be derived.

- mid-frequency varies from 460 Hz to 2,3 kHz
- it is assumed, that a comparable filter is achieved by using a constant quality-factor of 5

5.2.1. Implementation using FIR-filters

For the first tries, FIR-filters were used, because they always guarantee stability [3]. The first prototype of the coefficients combines a set of 32 coefficient sets, which were calculated beforehand with the following parameters:

Table 5.1: Parameters for FIR-filterdesign

quality factor	number of coefficients
5	130

Figure 5.2 shows a FIR-model of the analog reference at the lowest and (460 Hz) and the highest frequency (2242 Hz).

It can be clearly observed, that the shape of the filters are not close to the analog reference. The filter edges, especially near the baseband, are not steep enough.

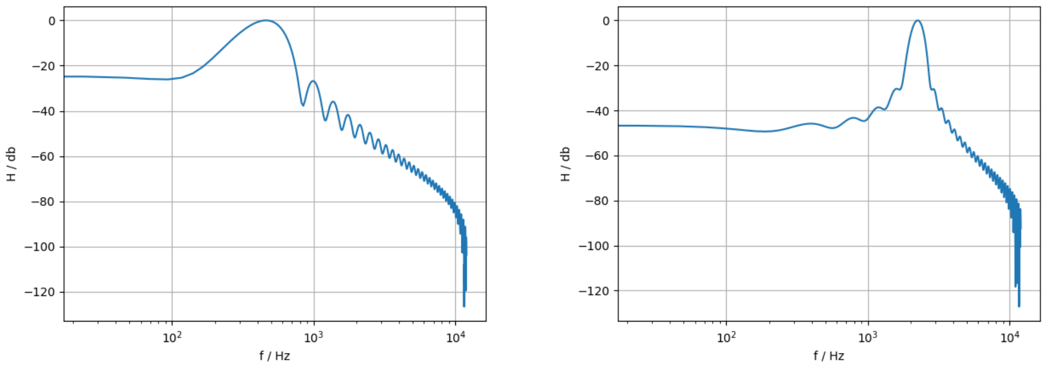

Figure 5.2: Simulation of the FIR-model frequency response at 460 Hz and 2242 Hz

5.2.2. Implementation using biquad-filters

The second approach to model the filter is by using a biquad-filter [4]. As already mentioned, the filter coefficients are being calculated at runtime.

Therefore following equations are used [5]:

$$\omega_0 = 2\pi \cdot \frac{f_c}{f_s} \tag{5.4}$$

$$\alpha = \frac{\sin \omega_0}{2 \cdot Q} \tag{5.5}$$

$$b_0 = \alpha \tag{5.6}$$

$$b_1 = 0 \tag{5.7}$$

$$b_2 = -\alpha \tag{5.8}$$

$$a_0 = 1 + \alpha \tag{5.9}$$

$$a_1 = -2 \cdot \cos(\omega_0) \tag{5.10}$$

$$a_2 = 1 - \alpha \tag{5.11}$$

, where f_c is the wanted center frequency, f_s the sampling frequency and Q the quality factor.

Designing a bandpass-filter with a biquad-filter results in the filters shown in Figure 5.3.

It shows, that the edges of a biquad-filter with five coefficients is much steeper than a FIR-filter with 130 coefficients. In general, the overall shape is closer to the analog reference in comparison to a non-recursive filter.

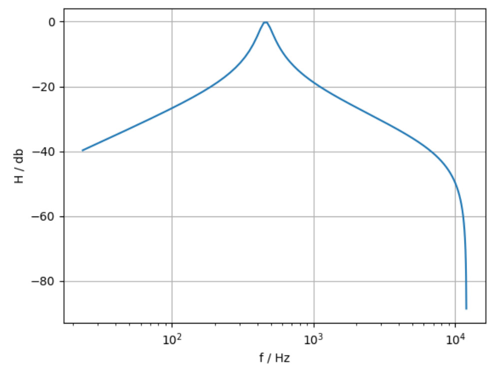

Figure 5.3: Simulation of the IIR-model frequency response at 460 Hz and 2242 Hz

5.3. Evaluation of the filters

In order to validate, that the implemented filters do in fact emulate a Wah-Wah-effect, the frequency response is measured as well as running through a hearing test.

5.3.1. Evaluation of the FIR-filters

Firstly the FIR-filter is evaluated.

To measure the frequency response, an oscilloscope with a signal generator is used. Therefore a sweep from 50 Hz to 10 kHz is applied to the input and a Fast Fourier Transform (FFT) of the output is calculated. The result of the filter at 460 Hz and 2242 Hz is shown in Figure 5.4. On channel 1 (yellow signal) there is the output and on channel 2 (blue signal) the input signal. The FFT result is shown in purple.

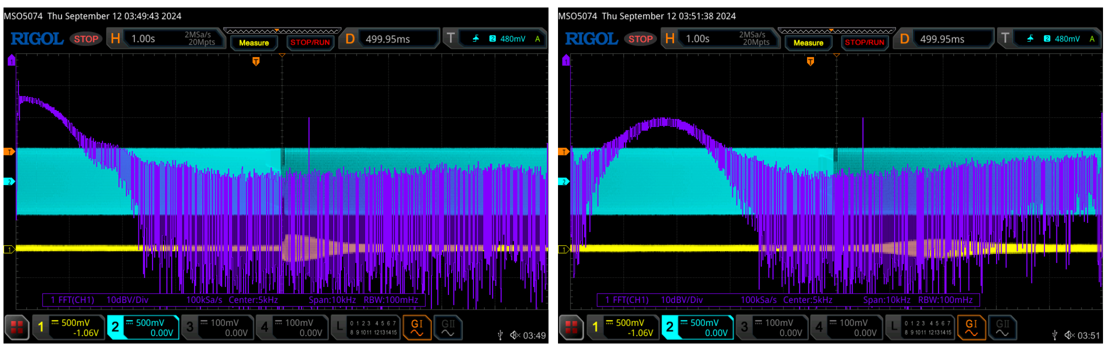

Figure 5.4: Measurement of the FIR-implementation

The first thing to be recognized is, that the filter at low frequencies is no longer a bandpass, but has lowpass characteristic. This is expected, if the simulation in Figure 5.2 is compared with this result.

Secondly, the filter has flat edges. This is also an expected result.

Lastly, the amplitude in the passband is a fraction of the amplitude of the original signal. This is not as expected and has to be evaluated in the future.

The first issue that will be recognized, is the cracking noise, when the filter coefficients are being updated. This can be reduced using fewer filter taps, but the consequence is, a frequency response with less sharp edges.

The hearing test provides the result, that the sound of this filter emulates a Wah-Wah-effect not sufficiently.

5.4. Development of the PCB

In order to get a handy pedal, the whole hardware should be integrated into one single PCB. The dimensions of this should not exceed the standard-sized size of effect pedals. As a reference the *Boss DS-1* distortion pedal were used to derive a reasonable size for the PCB. The dimensions of the distortion pedal are 73 x 129 mm, which should not be exceeded. The final PCB now has dimensions of 56,39 x 84,33 mm.

Furthermore, there should be enough inputs, to provide to ability to extend the functionality of this board. The idea is, to develop a platform, where any effect can be implemented. In order to fulfill this, three analog and four digital inputs are connected to solderpads for later use. Additionally, an Inter-Integrated Circuit (I^2C) connector is also provided, if for example a little display should be used.

The main components that were chosen are:

- STM32H725 microcontroller
- CS5343-CZZ Audio DAC
- CS4344-CZZR Audio ADC

The reason for the choice of these components is, that the these are also present on the Nucleo- and the breakout-board. Except for the microcontroller, therefore the smallest package with 68 pins were chosen.

5.5. Evaluation of the hardware

There are a few issues on the board, which have been found and fixed so far.

Issue 1:

The first issue is, that the core of the microcontroller is not connected to a source. This is, because the voltage regulator of this package is not connected to the core. This must be done externally. To fix this, a switched-mode power supply was added to the board, which outputs the 1 V for the core.

Issue 2:

The output voltage of the extra added power supply is noisy, which led to connection errors with the debugger. In order to fix this, extra capacitors were added to the supply of the core.

Issue 3:

The input of the op-amp needs an DC-offset voltage as it has only single supply. The offset voltage led to DC-coupling on the input port. To fix this a decoupling capacitor were added between to the input and the op-amp.

Current state:

It is possible to connect a J-Link debugger to the microcontroller.

A. Schematic

Bibliography

[1] D. J. Dailey, *Electronics for Guitarists*, 3rd ed. Springer, 2022.

[2] "Wah-wah." (), [Online]. Available: `https://de.wikipedia.org/wiki/Wah-Wah` (visited on 07/23/2023).

[3] M. Meyer, *Signalverarbeitung*, Analoge und digitale Signale, Systeme und Filter. Springer Vieweg, 2017.

[4] *Biquad cascade iir filters using direct form i structure*, Keil. [Online]. Available: `https://www.keil.com/pack/doc/CMSIS_Dev/DSP/html/group__BiquadCascadeDF1.html` (visited on 07/28/2024).

[5] R. Bristow-Johnson. "Cookbook formulae for audio equalizer biquad filter coefficients." (), [Online]. Available: `https://webaudio.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html` (visited on 09/12/2024).

[6] *Filter design tool*, Texas Instruments. [Online]. Available: `https://webench.ti.com/filter-design-tool/filter-response` (visited on 08/04/2024).

[7] *Um11732*, I^2S bus specification, Rev. 3, XNP Semiconductors, 2022. [Online]. Available: `https://www.nxp.com/docs/en/user-manual/UM11732.pdf` (visited on 07/23/2023).

[8] P. Beckmann. "Real-time embedded audio signal processing." (2008), [Online]. Available: `https://dspconcepts.com/sites/default/files/2008-10-05_real-time_embedded_audio_signal_processing.pdf` (visited on 07/23/2023).

[9] D. Stotz, *Computergestützte Audio- und Videotechnik*, Multimediatechnik in der Anwendung, 3rd ed. Springer Vieweg, 2018.

[10] D. Katz, R. Gentile, and T. Lukasiak. "Fundamentals of embedded audio, part 3." (2007), [Online]. Available: `https://www.eetimes.com/fundamentals-of-embedded-audio-part-3/` (visited on 07/23/2023).

[11] R. Kories and H. Schmidt-Walter, *Taschenbuch der Elektrotechnik*. Europa Lehrmittel, 2017.

List of Figures

List of Tables