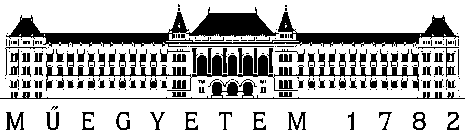
FELADATKIÍRÁS

A hallgató feladata egy gitárból (vagy más, hasonló jelszintű hangszerből) származó vonali jel fogadása, feldolgozása, átalakítása, és analóg jellé vissza konvertálása egy célhardver segítségével. A számításokat egy STM32 mikrokontrolleren kell elvégezni, és ehhez a szükséges periféria áramköröket is meg kell tervezni.

A munka során a következő részfeladatok megoldására van szükség:

* Bejövő jel vizsgálata, mérése
* Megfelelő illesztő áramkör tervezése, erősítők, szűrők méretezése
* Az átalakítások és számítások időkésleltetésének megállapítása, csökkentése
* A különböző transzformációk és néhány ismert effekt (pl. Delay, Chorus, Octave, stb.) szoftveres implementálása
* Célhardver megtervezése, élesztése, tesztelése



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Papp Áron

Digitális gitáreffekt megvalósítása mikrokontrollerrel

Konzulens

Kiss Ágoston

BUDAPEST, 2023

Tartalomjegyzék

[Összefoglaló 5](#_Toc136800807)

[1 Gitár effekt definíció 6](#_Toc136800808)

[2 Effekt specifikáció mérés 7](#_Toc136800809)

[2.1 Jel mérése időtartományban 7](#_Toc136800810)

[2.2 Jel mérése frekvenciatartományban 7](#_Toc136800811)

[3 Oktáv effekt modellezés 9](#_Toc136800812)

[3.1 Cél 9](#_Toc136800813)

[3.2 Naív próbálkozás FFT-vel 10](#_Toc136800814)

[3.3 FFT implementálása és problémái 11](#_Toc136800815)

[3.4 FFT algoritmus gyorsítása 12](#_Toc136800816)

[4 Filterbank alapú oktávkészítés 14](#_Toc136800817)

[4.1 ERB filterbank 14](#_Toc136800818)

[4.2 Gyorsított ERB filterbank 15](#_Toc136800819)

[5 Oktáv összehasonlítása különböző módszerekkel 20](#_Toc136800820)

[6 CMSIS függvények 22](#_Toc136800821)

[6.1 FFT használata 22](#_Toc136800822)

[6.2 ERB implementálása 22](#_Toc136800823)

[7 Tesztelési körülmények 24](#_Toc136800824)

[8 Jövőbeli tervek 26](#_Toc136800825)

[8.1 Nyomtatott áramköri terv 26](#_Toc136800826)

Hallgatói nyilatkozat

Alulírott **Papp Áron**, szigorló hallgató kijelentem, hogy ezt az önálló laboratóriumi beszámolót meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző, cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2023.05.08.

...…………………………………………….

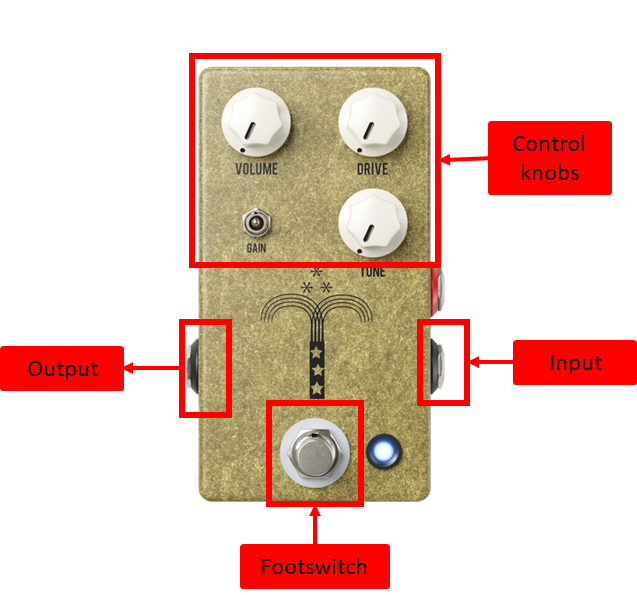
Papp Áron

Összefoglaló

Ebben az önálló laboratóriumi feladatban STM32H755ZI-Q fejlesztői panel segítségével digitális gitáreffekt létrehozása a cél. A feladat egyik kulcsfontosságú effektje az oktáver típusú effekt megvalósítása. Ahhoz, hogy megfelelő minőségű eszközt készítsek ismerni kell a „versenytársak” által nyújtott szolgáltatások, specifikációkat. A feladat jelentős részben digitális jelfeldolgozással is foglalkozik. Az elméletet különféle szakirodalmak, publikációk által leírt tapasztalatokat vett alapul, melyeket a gyors tesztelhetőség miatt elsőként „offline” teszteltem. Segítségemre volt ilyen téren a Matlab, melyben a különféle szűrők képzése gyors és átlátható volt. Amint megfelelő minőségű algoritmust tudtam készíteni, kezdetét vette az implementálás. A választott mikrokontroller nagy számításteljesítményű, így alapvető számítási korlát első ránézésre nem volt. Megismerkedtem a CMSIS könyvtár nyújtotta műveletek könnyű kezelésével, melyek elengedhetetlenek a követelmények teljesítéséhez.

# Gitár effekt definíció

A gitár effektek olyan berendezések, melyek lehetővé teszik zenészek számára, hogy különböző hangokkal és effektekkel egészítsék ki gitárjátékukat. Ezek eleinte analóg áramköröket jelentettek, melyek ma is igen keresettek, azonban a technika fejlődésével megjelent digitális effektek is igen nagy népszerűségnek örvendenek. A játék alatt akár többször ki-be kapcsolgatják az effekteket. Mivel a zenészek kezeit leköti a zeneművészet csúcstermékének eljátszása, a lábukat használják a berendezések működtetéséhez. Erre tipikusan egy DPDT vagy 3PDT kapcsolót használnak. A hangjel a gitárosok által jól ismert 6.3-as jack csatlakozón terjed. A lábkapcsoló megnyomásakor a zenei jel vagy direktbe kerül a kimenetre, vagy „moduláltan”. Az effektnek tehát van gitárjelhez ki és bemenete, kapcsolója, egy visszajelző fény, mely tudatja a kapcsoló állapotát, és kezelőszervek (potentiometer, jog). Ilyen hangzás az overdrive/distortion (torzító), a Delay (késleltetés), oktáver stb... A cél az, hogy a felhasználónak könnyű legyen elérni a fejében elképzelt hangot.



. ábra JHS gyártó Morning Glory V4 torzító effektje

# Effekt specifikáció mérés

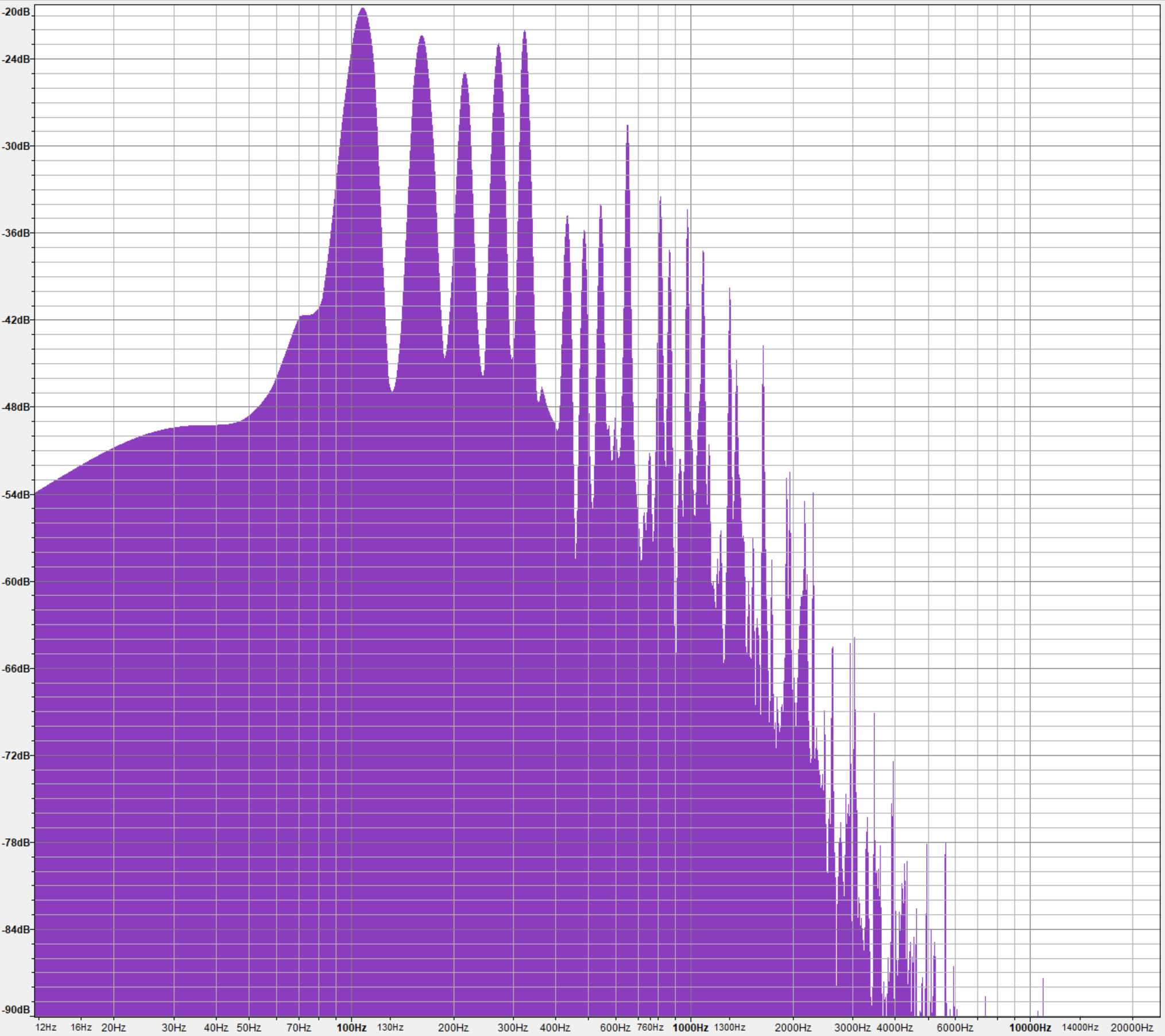
## Jel mérése időtartományban

A gitár húrjait pendítve, azok rezgőmozgást végeznek. Ha ezt a rezgőmozgást fémhúrok végzik egy mágnes fölött, akkor mágneses indukció lép fel. Ehhez a jelenséghez hangszedőt használnak, mely valójában egy vasmaggal ellátott tekercs. Mivel a jel frekvenciája megeggyezik, az akusztikai jellel, könnyen mérhetjük és feldolgozhatjuk. A fejlesztés során egy 2015 Epiphone Les Paul Plustop Pro típusú gitárt használtam. Ennek a hangszernek specialitása, hogy a hangszedője „felezhető”. A hangszedőket két csoporta lehet szedni. Az úgynevezett „single coil” és a „humbucker” típusú pickup. Az utóbbi jelentősen nagyobb (2x) jelszintet produkál. Ez a gyakorlatban azt jelenti, hogy a gitár hangját feldolgozó erősítő más torzításokat végez a jelen. Zenei ízlés igen egyénfüggő, egyik sem jobb mint a másik, egyszerűen más. A felezhető pickup azt jelenti, hogy egy kapcsoló segítségével képes vagyok a fele tekercset „deaktiválni” így a humbuckert egy single coil-ra tudom alakítani.

|  |  |  |
| --- | --- | --- |
| Nyaki hangszedő mód (típus: Epiphone Probucker) | Egy hang pengetése [V] | Akkord [V] |
| Humbucker | 0.3 | 1 |
| Single Coil | 0.13 | 0.5 |

## Jel mérése frekvenciatartományban

A frekvenciaméréshez egy Behringer Xenix 1222 típusú keverőpultot használtam. A keverőpult usb bemenettel kapcsolódik a számítógéphez, így akár ott is lehet mérni/lejátszani hangokat. A keverő képes 48 kHz-es mintavételi frekvenciára, amely manapság standard, sőt stúdiókban inkább magasabb 96, 192 kHz-es jelek is előfordulnak. Egy hang lefogása a gitár nyakán és pendítése a továbbiakban „hang”. Ha több hangot fogunk le és egyszerre szólaltatjuk meg azokat, akkor akkordot játszunk. A mérés során ezt a két típusú jelet vizsgáltam. A mérés során a keverőpulthoz csatlakoztatott gitár jelét Audacity zenei programmal tudtam elmenteni. Az elmentett állományt akár Audacityben akár matlabban könnyen feldolgozhattam. A mérések azt bizonyították, hogy a gitárok legnagyobb frekvenciakomponensei kb 10 kHz-ig terjednek. Fontosak a magas frekvenciás komponensek, de nem szabad elfelejteni, hogy a gitár jelútja könnyen 6-10 méter is lehet. Ilyenkor a kábel, mint aluláteresztő szűrő viselkedik, így ezek a komponensek nem biztos, hogy eljutnak az erősítő bemenetére. A jelcsökkenés ellen buffer erősítőket alkalmaznak, azonban azok karakterisztikái torzíthatják a jelet, vagy egyéb felharmonikus komponsens képeznek.

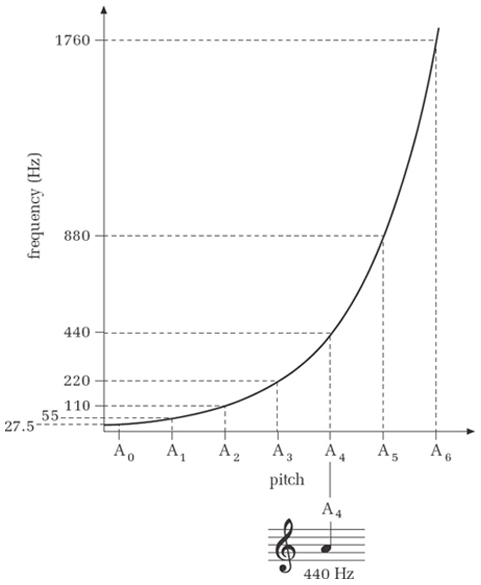


. ábra Egy „A” akkord spektrumja. Jól látható, hogy komponensek akár 6 kHz-ig is terjedhetnek.

# Oktáv effekt modellezés

## Cél

Egy frekvenciakomponens oktávja pontosan annak a kétszeresét jelenti. Ez az alapgondolat, amikor oktáv effektet kezdtem fejleszteni. Tehát valójában nincs más dolgom, mint a bemeneti jel frekvenciakomponenseit megduplázni. A zenei hangok két oktáv között 12 félhangra vannak bontva. Ez tulajdonképpen egy logaritmikus skálát jelent. Egy képzett fül 8-10 cent hibát is képes hallani. (két oktáv között 1200 cent van). Ebből az is következik, hogy alacsony hangok esetében 1 Hz eltérés nagyobb zenei hibát eredményez, mint magasokon.



. ábra Oktáv szemléltetése freknveciaspektrummal

Nézzünk egy példát:

|  |  |  |
| --- | --- | --- |
| Hang | Frekvencia [Hz] | 1 Hz relatív Hiba [\*10^-3] |
| A3 | 220 | 4.5 |
| A4 | 440 | 2.27 |

## Naív próbálkozás FFT-vel

Egy gitárjel frekvenciabeli analízisekor a legtöbb embernek a spektrum jut eszébe, én se voltam kivétel. Az FFT által előállított komplex frekvenciatagok abszolútértéke adja a jel spektrumát. A jel spektruma tartalmazza a frekvenciakomponenshez tartozó amplitudóértéket is. Érdekesség, hogy az emberi fül nem annyira érzékeny a fázishibára, mint két frekvenciakomponens relatív távolságára. Ez azt jelenti, hogy ha a jel megkapott spektrumának komponenseit mind megkétszerezem, akkor egy oktávval magasabb jelet kapok. Egy képzett zenész egy hangot 10 centnél pontosabban hall pontosnak. Mint a 3.1-es fejezetben említve, előjön a pontosság problémája. Ha az effektem minden hangra pontos szeretne lenni, akkor azokat mind 10 centnél pontosabban kell tudnom mérni. Ez a kalsszikus FFT-vel nem lehetséges. A válasz a spektrum jellegében rejlik. Egy DFT számítás során a kapott eredmény egy olyan tömb melyben a frekvencia komponensek a feldolgozott mintaszám és a mintavételi frekvencia függvénye. Az effekt „real-time”-ként kell, hogy fusson. Követelménynek érdemes felvenni egy késleltetés értéket, ami megszabja, hogy a moduláció milyen gyorsan követi le a beérkező hangokat. Ha ez túl sok idő (>0.03 s) akkor a ritmustartás nehézkessé válik. Az Electro Harmonix cég POG2 [15] típusú oktáver effektjéről azt hallottam, hogy 30 ms késleltetést okoz. Ilyen gyorsaságnál már a zenészek jelentős része szerette használni, úgyhogy én is ezt a cél állítottam a tervezés első fázisaiban. A DFT-hez szükséges mintákat összegyűjteni azonban időbe kerül. A legalacsonyabb gitárból kimérhető hang (standard hangolásban) az alsó E2 hang mely 83 Hz-es, a legmagasabb egy 24 bundos gitáron pedig az E6, aminek a frekvenciája 1396 Hz. A feladat az, hogy válasszuk ki azokat az DFT paramétereket, amelyek képesek a 83 Hz-es jelet 10 cent pontosságal megmérni.

-83 Hz=0.48 Hz

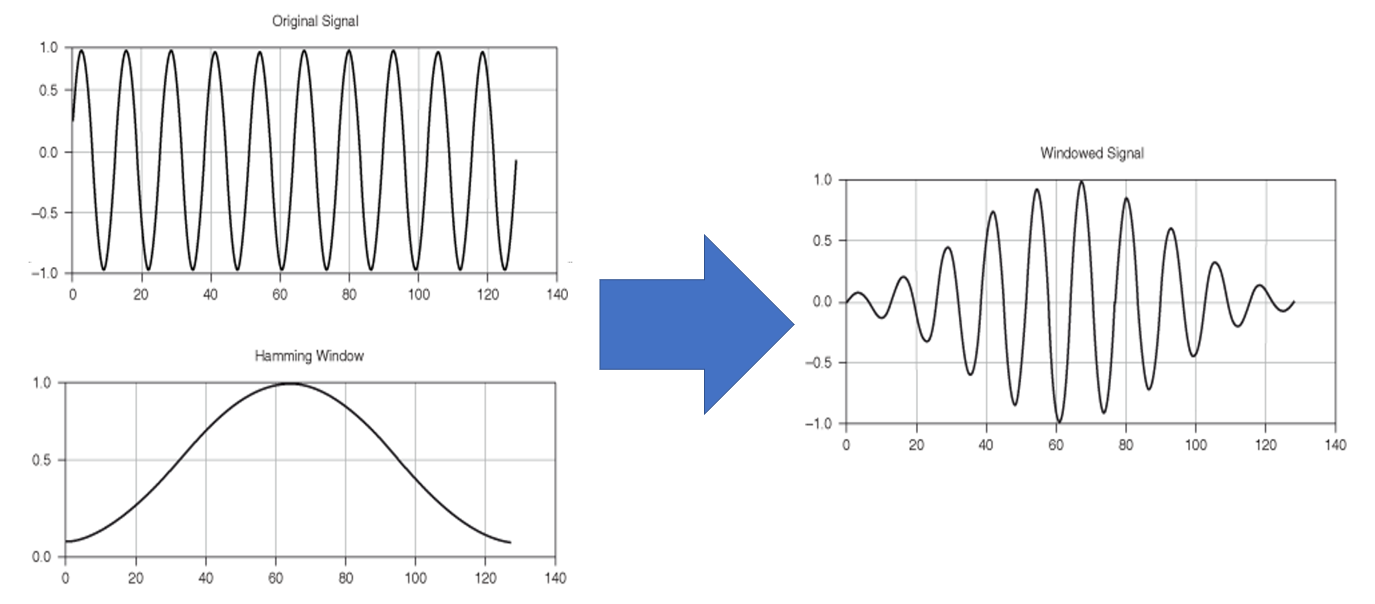
Tegyük fel, hogy a legnagyobb frekvenciakomponens 10 kHz. A mintavételi frekvencia minimum ennek a kétszerese kell, hogy legyen, tehát legyen . Az elöbb számított minimális eltérés és mintavételi frekvenciából kiszámítható a mintaszám.

Ez alapján egy FFT számításhoz ~41700 mintát kell venni. Ne feledjük, hogy ez egy Real-Time rendszer, azonban 41700 mintát 20 kHz mintavételi frekvenciával csak 2.085 s alatt lehet összegyűjteni. Ebből az látszik, hogy a naív FFT implementáció korlátokkal fog működni. Ugyanez a számítás a nagyobb frekvenciás E6-ra 2473.12-re jön ki. Triviális, hogy nem lehet megfelelni mindegyik hangra ugyanolyan követelményekkel.

## FFT implementálása és problémái

Kompromisszumos paramétereket használva a mintavételi frekvencia 30 kHz lett és mintaszám 4096. Ezek az STM32H7-re való implementálás során felmerült számítási korlát miatt alakult így. Az algoritmus „n” mintaszámú feldolgozott jel után FFT segítségével minden frekvenciakomponens a nála kétszer nagyobb indexre kerültek. Azért, hogy ne legyen üresjárás két buffert használtam, amelyet egy DMA mozgatott a DAC memóriába. Az eljárás előnye, hogy a processzor teljesmértékben képes a jelfeldolgozással foglalkozni, az egyszerű adatmozgatást így a DMA végzi. Legyen a két buffer neve „buffer1” és „buffer2”. (Ezek valójában egy bufferként definiáltak, csak az egyszerűség kedvéért félbevágva alkotják az előbb említett buffereket) Az algoritmus a következő módon működik. Beérkezett „n” minta után a DMA képes egy callback függvény segítségével jelezni, hogy áthaladt a bufferhossz felén. Ilyenkor a While() loopban kiértékelődik az FFT-s oktáv effekt, majd annak eredményét beírja a buffer1 memóriaterületére. Fontos, hogy mire buffer2 megtelik a buffer1-en végzett művelet véget érjen. Az algortimus +1 oktáv esetén bíztató eredményt hozott, azonban nem volt megfelelő minőségű. A visszaalakított jelek igen torzak voltak, és azt tapasztaltam, hogy csak egy bizonyos frekvencián voltak igazán tiszták. Oszcilloszkópos vizsgálattal, nem láttam olyan feszültségértéket az ADC bemenetén, mely a tápfeszültség 3.3 V maximumát átlépte volna.

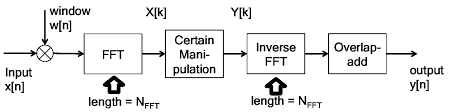
Fontos megemlítenem, hogy az első próbálkozás során a beérkező hangmintákat semmilyen ablakfüvénnyel nem alakítottam át. Ez azt jelentette, hogy a mintavétel nem koherens. Ezt úgy könnyű elképzelni, hogy ha a jelünk egy szinusz jel, akkor abból képzett minta nem 0-ban hanem valami egészen más értékben záródik. Ez abból fakad, hogy a mintavételi frekvencia nem egyenlő a vizsgált jel egészszámszorosával. Az ablakfüggvények ugyan torzítással, de képesek periódikussá alakítani a jelsorozatot. Ilyen ablakfüggvény például a Hamming, Blackman-Harris. A haranggörbe jelleg a jelsorozat elejét és végét csillapítja, közepét igyekszik megtartani. Az így feldoglozott jelek valóban sokkal kellemesebb hatást váltottak ki.



. ábra Ablakozott jelalak Hamming ablakkal

## FFT algoritmus gyorsítása

Az ablakozó függvény egyik problémája, hogy az ablakozott jel visszalakítva is egy amplitúdó modulált jelet alkot. Azért, hogy ezt csökkentsem úgynevezett „overlap-add” eljárást használtam fel. Ennek az eljárásnak a lényege, hogy a jelfeldoglozás nem csak új adatokból, hanem korábbiakból is történik. Ennek mértéke tipikusan 50%-ra esett. Ilyenkor a két haranggörbe egymásmellé helyezve és összeadva már egy jelentősen kisebb amplitúdó változást jelentett. Ha tovább növeljük az átlapolódás mértékét, akkor ugyan csökkentettük a „hullámzást”, de ehhez gyors hardver is szükséges. Az elöbb felvett paraméterekkel számolva, ez a számítás 4096 minta helyett annak fele kell csak, hogy beérkezzen. 2048 adat beérkezése alatt kell, továbbra is 4096 mintából álló FFT-t számítani. Mivel a kellő késleltetést nem értem el ezekkel az algoritmusokkal, más metódusok felé kezdtem kutatni.



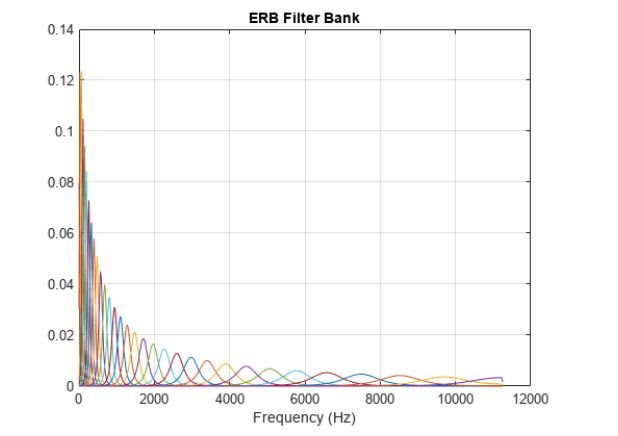
. ábra Overlap-Add algoritmus blokkvázlata [1]

# Filterbank alapú oktávkészítés

## ERB filterbank

Munkám során olyan algoritmusokat kerestem melyek az emberi fül viselkedését veszik alapul. Ilyen megoldást jelent az úgynevezett ERB (equivalent rectangular bandwith) típusú filterbank. Egy filterbank több szűrőt valósít meg, különböző paraméterekkel, akár mintavétellel is. Utóbbi úgynevezett multirate filter tudományágához vezet, ami számítási komplexitás csökkentéséért felelős. Ennek a dolgozatnak nem képezi tárgyát. Egy sáváteresztő szűrőt a központi frekvenciával és a szűrő meredekségével jellemzem. Etienne Thuillier szerint a megoldás a gyors polifónikus oktáv effekthez vezető út az ERB szűrők mutatják. Szerinte elégséges 43 darab rendkívől magas Q értékkel ellátott szűrőt alkalmazni. [2]

A tanulmány a 43 alsávi jelen Hilbert transzformációval fázistolást valósított meg. Az eltolt jel és az eredeti jel szorzata kétszeres frekvenciakomponens és egy nulla közeli frekvenciakomponens létrejöttét okozza. Egy jól paraméterezett felüláteresztő szűrővel az alacsonyfrekvenciás jelet képesek vagyunk eltüntetni, és meghagyni a „tisztán” oktáv jelet. A tiszta szó itt nem valódi tiszta, például szinuszt jelent. Ez legfőképp a sáváteresztő szűrők sávszélességén múlik. Az eredmény megfelel az elvártaknak, azonban az oktávozott jel rendkívül gyenge volt. Habár a hang megfelelő minőségű volt, bekell látni, hogy ez továbbra is egy erőforrásigényes eljárás. A bemeneti 43 szűrő, 43 fázistolás, szorzások, és ez mind csak 1 oktáv számítása, nem beszélve arról, hogy lefele oktávot nem valósítottunk meg.

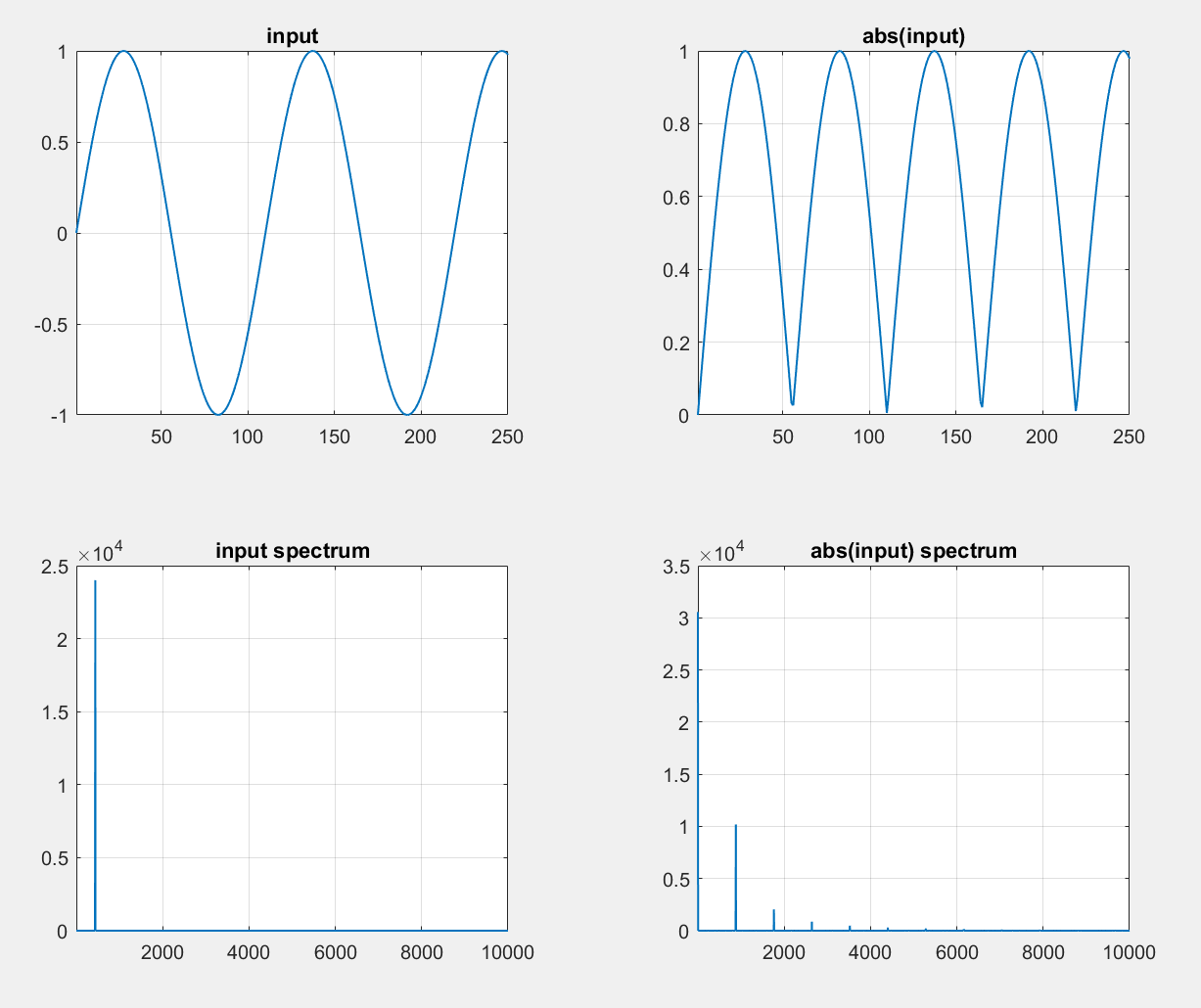


. ábra ERB filterbank szemléletes ábrázolása. Minden szín eltérő középfrekvenciával és Q értékkel rendelkezik. [3]

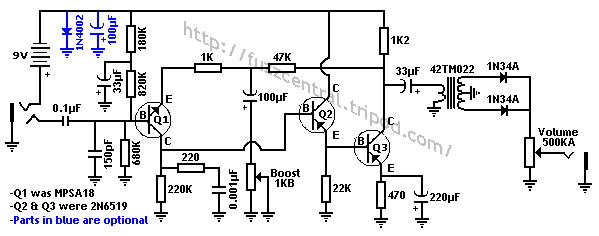
## Gyorsított ERB filterbank

Ebben a fejezetben az ERB filterbank számításai szeretném csökkenteni, főként mikrokontroller alapú rendszer szempontjából. A bemenetre engedett 43 szűrő elengedhetetlen, azonban a további részegységeket a következő módon lehet gyorsítani. Egy szinusz jel abszolút érték véve képesek vagyunk torzítva, de előállítani a jel egy oktávjellegű mását.

Ha megspórolom a valódi oktáv szűrését akkor már jelentős gyorsítást vittem a programomba. Az algoritmus végén a sávokat összeadva megkapom a jelfolyam oktávval eltolt mását (és torzításokat). A torzítás valójában kedvelt hatás. Az ötletet egy analóg effekt Tycobrahe Octavia adta, mely analóg módon valósította meg, azt amit én többsávon. Az effekt torzításba viszi a bemeneti gitárjelet, majd a kimeneten egy kétutas egyenirányítón átvezeti azt. Fontos megjegyezni, hogy az effekt csak egy hang kijátszásakor képes elérni az oktáv hangot. Az enyém ezzel ellentétben akár akkordok játszása esetén is jól működik.

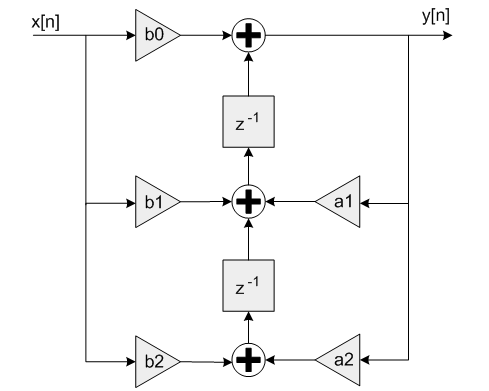


. ábra Abszolútérték képzés spektruma



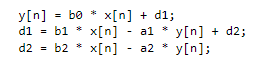
. ábra Tycobrahe Octavia kapcsolási rajza. Figyeljük meg a kimeneti egyenirányító áramkört. [5]

A CMSIS könyvtár előre definiált szűrői között mind FIR, CIC, IIR szűrők jelentek meg. Az említett szűrők kiszámításához az IIR szűrő Direct form II számítási metódusát használtam fel.

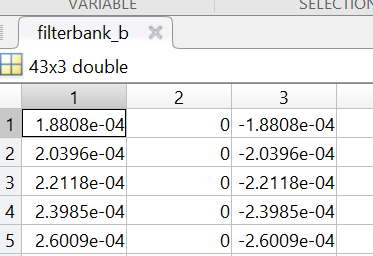


. ábra IIR szűrő Direct form II struktúrával számítva [6]

A szűrőparamétereket megvizsgálva egyértelművé vált, hogy a sáváteresztő szűrő eggyüthatói esetében a b2 mindig 0 értékű volt. Ez azt jelenti, hogy legalább 1 szorzás műveletet megspórolhatok, hiszen bármely szám szorzata 0 értéket ad.



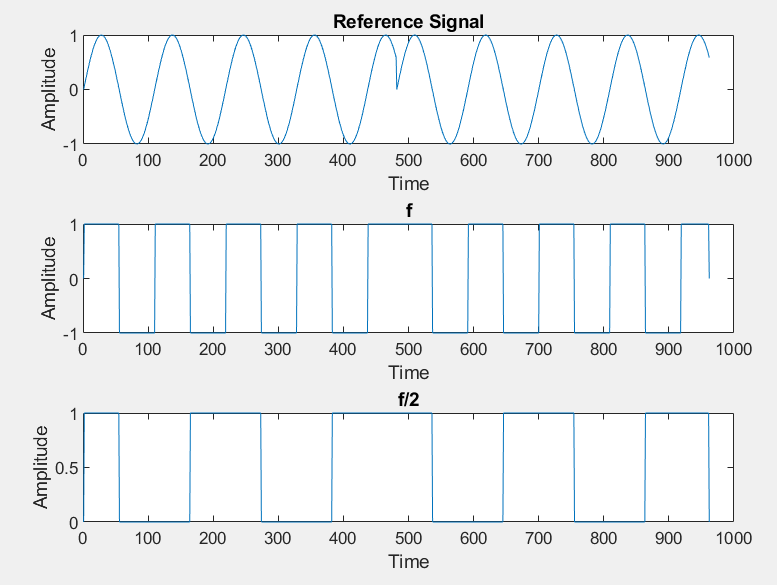
10. ábra IIR szűrő Direct form II differencia egyenlete [6]



. ábra Sáváteresztő szűrő „b” együtthatói

## Alsó oktáv előállítása

Szeretném elérni, hogy a játszott hangok egy oktávval lejjebb kerüljenek. Ez FFT esetében egy egyszerű mozgatás a spektrumon, viszont a korábban tárgyalt problémák miatt ezt elvetettem. Saját algoritmust készítettem, mely az egyszerű D tárolós frekvenciaosztó elvén alapul. A 4.1-es fejezetben említettem, hogy az alsávokban lévő jel tulajdonképpen szinuszos jelnek tekinthető. Ez azt jelenti, hogy képesek vagyunk detektálni a nullátmeneteket (amikor a jel negatívból pozitívba mozog vagy fordítva). Ha ezekre a váltásokra diszkrét értéket helyezünk képesek vagyunk egy négyszögjelet alkotni. Ennek a jeltípusnak frekvenciaosztását matlabban implementáltam oly módon, hogy a négyszögesített jel lefutó élére változott a kimenet értéke.



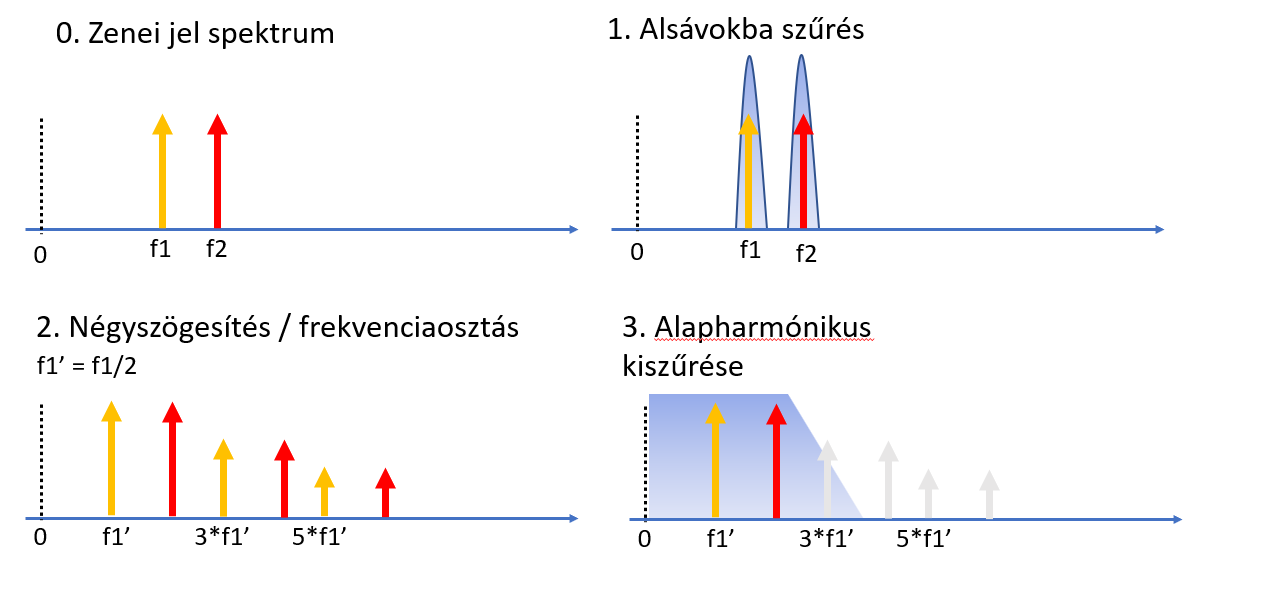
. ábra Alsávi jel „négyszögesítése” és frekvenciaosztása. Szándékos zavar esetén max fél periódusnyi „hiba” keletkezik

Ugyanakkor a négyszögjel ugyan tartalmazza a tisztán szinuszos komponenst, de ezen felül végtelen komponense van. Ezek a komponensek az alapharmonikus után 2n+1 frekvencián találhatóak meg. Ha 43 alsávom van, akkor a frekvenciaosztott jel is 43 sávból áll. Minden egyes alsávi érték szűrése időigényes feladat. Tehát a cél, egy olyan algoritmus készítése, mely kiszűri az alapsáv jelet, és lehetőleg minél több sávot egyszerre szűrjön. 13. ábra-n az algortimus menete található. Tegyük fel, hogy a bemeneten két frekvenciakomponenst detektálunk. Az ERB módszerrel (4.1) alsávokra szűrjük azokat, majd a tárolós módszerrel kialakítjuk a négyszögjeleket egy oktávval lejjebb.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fourier komponensek [Hz]** | **1.** | **2.** | **3.** |
|  | 40 | 120 | 200 |
|  | 60 | 180 | 300 |

A táblázatban jól látható, hogy alapharmonikusa kisebb, mint 2. komponense. Ez azt jelenti, hogy tudok definiálni egy olyan alul áteresztő szűrőt, mely képes alapharmonikusát átengedni, de a 1. komponensét nem. Tehát az összes olyan frekvencia, ami kisebb mint 120 Hz egyszerre szűrhető. Számolva a 43 alsávval végül 4 alszűrőt határoztam meg.

|  |  |  |
| --- | --- | --- |
| **Szűrő** | **Törésponti frekvencia** | **Alsávok száma [1:43]** |
| 1 | 140 | 1:8 |
| 2 | 430 | 9:19 |
| 3 | 1200 | 20:30 |
| 4 | 3000 | 31:43 |



13. ábra Alsó oktáv szűrő algoritmus

Az algoritmus megoldatlan része, hogy a nullátmeneti pont idejében ismeretlen a szinusz hullám amplitúdója. Ezért ennek implementálása során egy görgetett átlagot használtam amplitúdónak. Habár ez nem a valódi amplitudó, megfelelő hangolással elérte a kívánt eredményt. A konstans érték is jó megoldásnak bizonyult, viszont ez csak akkor használható, ha a dinamika (max-min jelszint) nem olyan fontos. További kutatás szükséges.

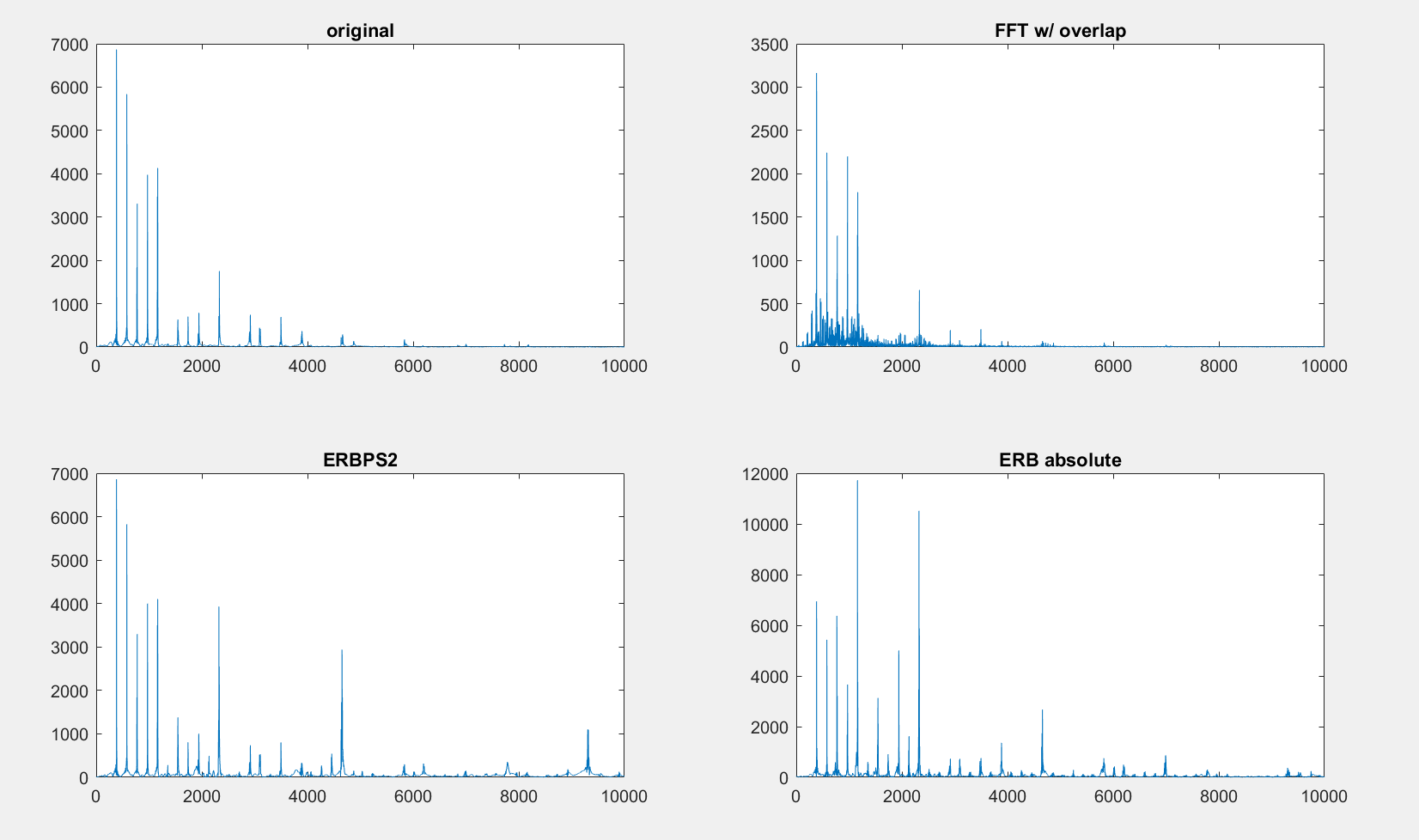
# Oktáv összehasonlítása különböző módszerekkel

Egy táblázatban összegyűjtöttem a vizsgált megoldásokat. A késleltetés akkor megfelelő, ha a zenész nem érzi a jelfeldolgozásból eredő késleltetést. Ez csak az ERB típusú feldolgozás esetében jelent meg.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Módszer** | **Számításigény** | **Minőség** | **Memóriahasználat** | **Késleltetés** |
| FFT | nagy | magas frekvenciákon elfogadható | nagy | nagy |
| FFT átlapolással | nagy | jobb, mint az FFT | nagy | kisebb, mint FFT |
| ERB-PS2 | kisebb, mint az FFT | kiváló | közepes | elhanyagolható |
| ERB Abszolút érték módszer | közepes | jó | közepes | elhanyagolható |

. ábra Különöző módszerek összehasonlítása

A 10. ábra-n látható frekvencia analízisen keresztül is betekintést nyerhetünk a különböző frekvenciák alakulására. Az „original” egy egyszerű akkordot jelent, melyet az algoritmusoknak bemenetként szolgáltattam. Jól látszik, hogy a „legtisztább” spektrum az ERB-PS2 metódussal érhető el. A legtöbb spektrumszivárgással az FFT típus rendelkezik. A saját fejlesztésű oktáv egészen jól követte az ERB-PS2 komponenseit, de látszik, hogy sok kis torzítás jelentkezik. Fontos megjegyezni, hogy az algoritmustól függően eltérő komponens amplitúdókat kapunk. Ezért az ábrát csak a frekvenciakomponensek helyzete miatt érdekes, azok amplitúdói nem. A különböző hangosság beállításához a kezelő potenciométerek alapján betudja állítani a megfelelő erősítéseket.



15. ábra Különböző módszerek összehasonlítása a frekvenciatartományon

# CMSIS függvények

A megvalósításhoz az STMicroelectronics által fejlesztett CubeIDE fejlesztői környezet alkalmaztam. Grafikus konfigurációval és kódgenerálással gyorsan haladtam a tesztekkel. A DSP függvények használatához külön hozzákell adni a könyvtárakat a projekthez, ehhez számos segédanyag található az interneten. A CMSIS könyvtár beépített függvények használatakor több szempontot is figyelembe kell venni. A használható függvények függenek a hardveres egység típusától. Az STM32H755ZI-Q fejlesztői panelen lévő mikrokontroller double lebegőpontos számítóegység található. Így akár float32\_t típusú változókkal is nagyon gyorsan tud számolni. A használt függvények gyakran sajátos struktúrákat használnak. Ugyan a használt mikrokontrollerben külön M4 és M7-es mikrokontroller is használható, csak az M7-est vettem használatba.

## FFT használata

A számításhoz az arm\_rfft\_fast\_f32() [9]függvényt kell meghívni. A függvényben az adattömb pointer-ét kell átadni, és a kimeneti buffer pointerét. Fontos megjegyezni, hogy a gyors számítás érdekében a bemeneti adattömböt is megváltoztatja. A kimeneti adattömb DC értéket és valós – komplex értékpárokat tartalmaz. Ha ebből amplitúdóspektrumot szeretnénk képezni, akkor a komplex függvények közül az arm\_cmplx\_mag\_f32() [10] függvényt kell használni.

## ERB implementálása

Az említett IIR filter direct form II típusú számítási módszer implementálását lehet a következő rövid C kódban olvasni. (8. ábra) Az arm\_mult\_f32() és arm\_add\_f32() függvények float32\_t típusú változókból álló tömböket képesek összeadni. Nem igényelnek speciális átalakítást, mint a komplexebb függvények, ezzel is számítási idő takarítható meg. Az implementálás során azt tapasztaltam, hogy a futtatási idő 10000-szeri meghívása esetén 400 Mhz-es órajellel az átlag végrehajtás 7 us, ami tovább csökkenthető, ha a függvényt Macro függvényként definiálom. Ilyen esetben ~6.1 us alatt hajtotta végre a szűrőműveletet. A szűrők együtthatóit matlab-bal generálom, és az értékeket a subbandfilter\_X változókban tárolom el. A kódsorokban található A1, B1… jelölések csupán az összeadások közötti részszámításokat tartalmazzák.

**void** **subbandfilter\_calculation**(**uint32\_t** input){

**float32\_t** input\_f32=(**float32\_t**)input-**2048**;

// set d[n], d[n-1], d[n-2]

**for**(**int** i=**0**;i<numberofsubbands;i++){

subbandfilter\_input[i]=input\_f32;

subbandfilter\_dn2[i]=subbandfilter\_dn1[i];

subbandfilter\_dn1[i]=subbandfilter\_dn[i];

}

// A1

arm\_mult\_f32(subbandfilter\_a1, subbandfilter\_dn1, subbandfilter\_A1, numberofsubbands);

// A2

arm\_mult\_f32(subbandfilter\_a2, subbandfilter\_dn2, subbandfilter\_A2, numberofsubbands);

// A1+A2

arm\_add\_f32(subbandfilter\_A1, subbandfilter\_A2, subbandfilter\_dn, numberofsubbands);

// d[n]=A0-A1-A2

arm\_sub\_f32(subbandfilter\_input,subbandfilter\_dn, subbandfilter\_dn, numberofsubbands);

// y\_n=b0\*d[n]+b1\*d[n-1]+b2\*d[n-2]

// B1

arm\_mult\_f32(subbandfilter\_b1, subbandfilter\_dn1, subbandfilter\_B1, numberofsubbands);

// B2

arm\_mult\_f32(subbandfilter\_b2, subbandfilter\_dn2, subbandfilter\_B2, numberofsubbands);

// B1+B2

arm\_add\_f32(subbandfilter\_B1, subbandfilter\_B2, subbandfilter\_output, numberofsubbands);

// B0

arm\_mult\_f32(subbandfilter\_b0, subbandfilter\_dn, subbandfilter\_B0, numberofsubbands);

// y=B0+B1+B2

arm\_add\_f32(subbandfilter\_output, subbandfilter\_B0, subbandfilter\_output, numberofsubbands);

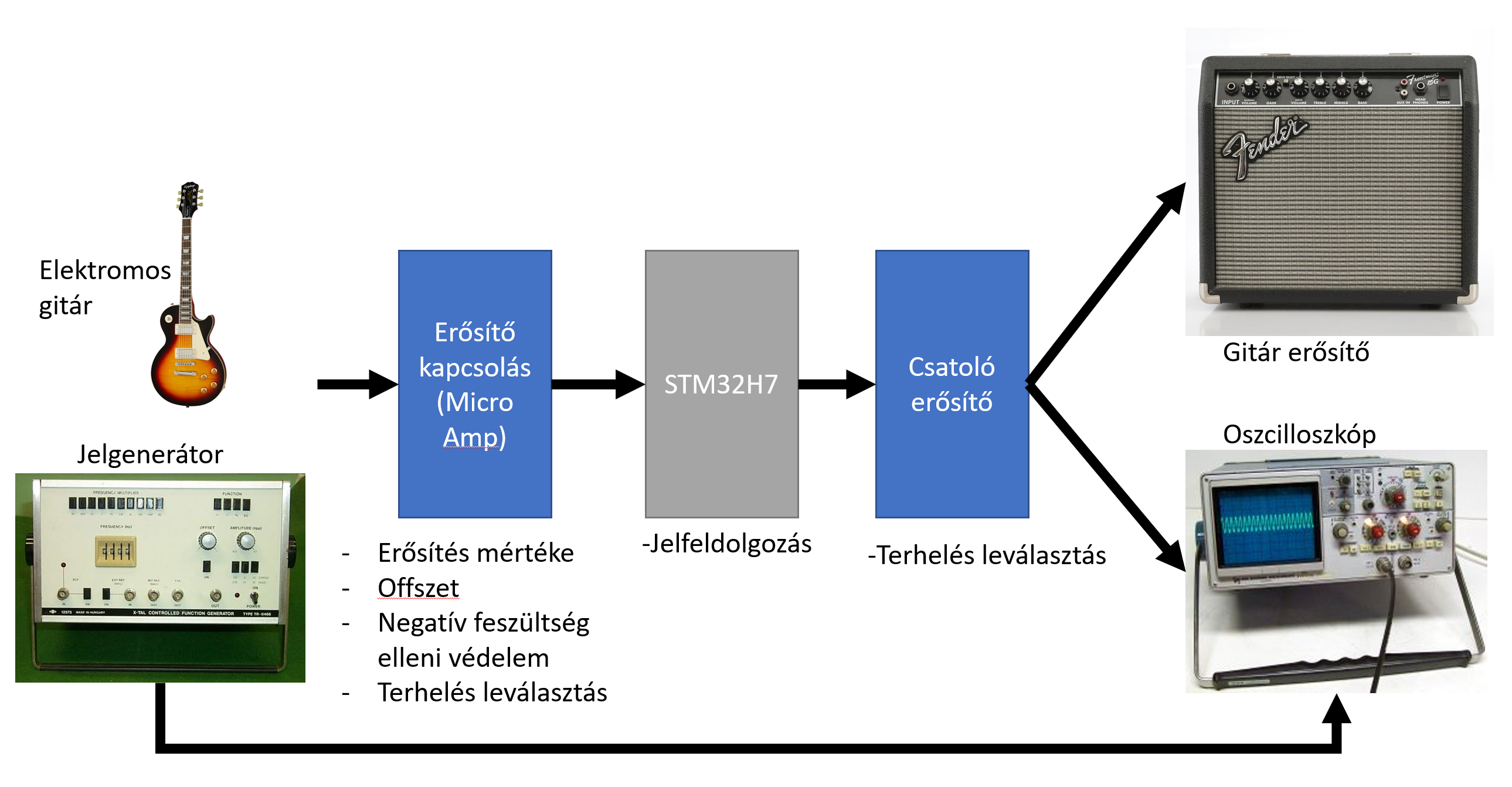
}

# Tesztelési körülmények

A matlabban történt szimulációk után a valóságban implementáltam az FFT overlap és az ERB abszolút megoldását. A tesztelés elején saját jelgenerátoromat használtam, azonban észrevettem, hogy kikapcsoláskor negatív feszültségeket is kiad, ami az MCU ADC csatornáját tönkretette. A hiba kiküszöbölésére diódás védőkapcsolást helyeztem a bemenetre. Az ADC akár 16 bites felbontást is lehetővé tud tenni, de én csak 12 biten használtam. Ennek oka, hogy a processzor DAC felbontása is 12 bit, így nem kellett még a skálázással is számolni. Mivel a gitárjel gyakran negatív feszültségértékeket is felvehet offszetelni kellett. A tesztkapcsolásban egy Rail-to-Rail erősítőt használtam, amely tápfeszültséget a fejlesztői kártyáról kapott. A be és kimeneti jeleket oszcilloszkópon követtem. A 12. ábra-ában említett eszközökkel csak időtartománybeli változásokat tudtam vizsgálni.

|  |  |
| --- | --- |
| **Pin** | **Funkció** |
| PA5 | DAC |
| PB0 | Beépítettt műveleti erősítő -> ADC |
| **Felhasznált eszközök** | **Funkció** |
| Type TR – 0466 | Jelgenerátor |
| Tektronix 434 storage oscilloscope | Jelvizsgálat |

16. ábra Teszt konfiguráció



. ábra Tesztösszeállítás

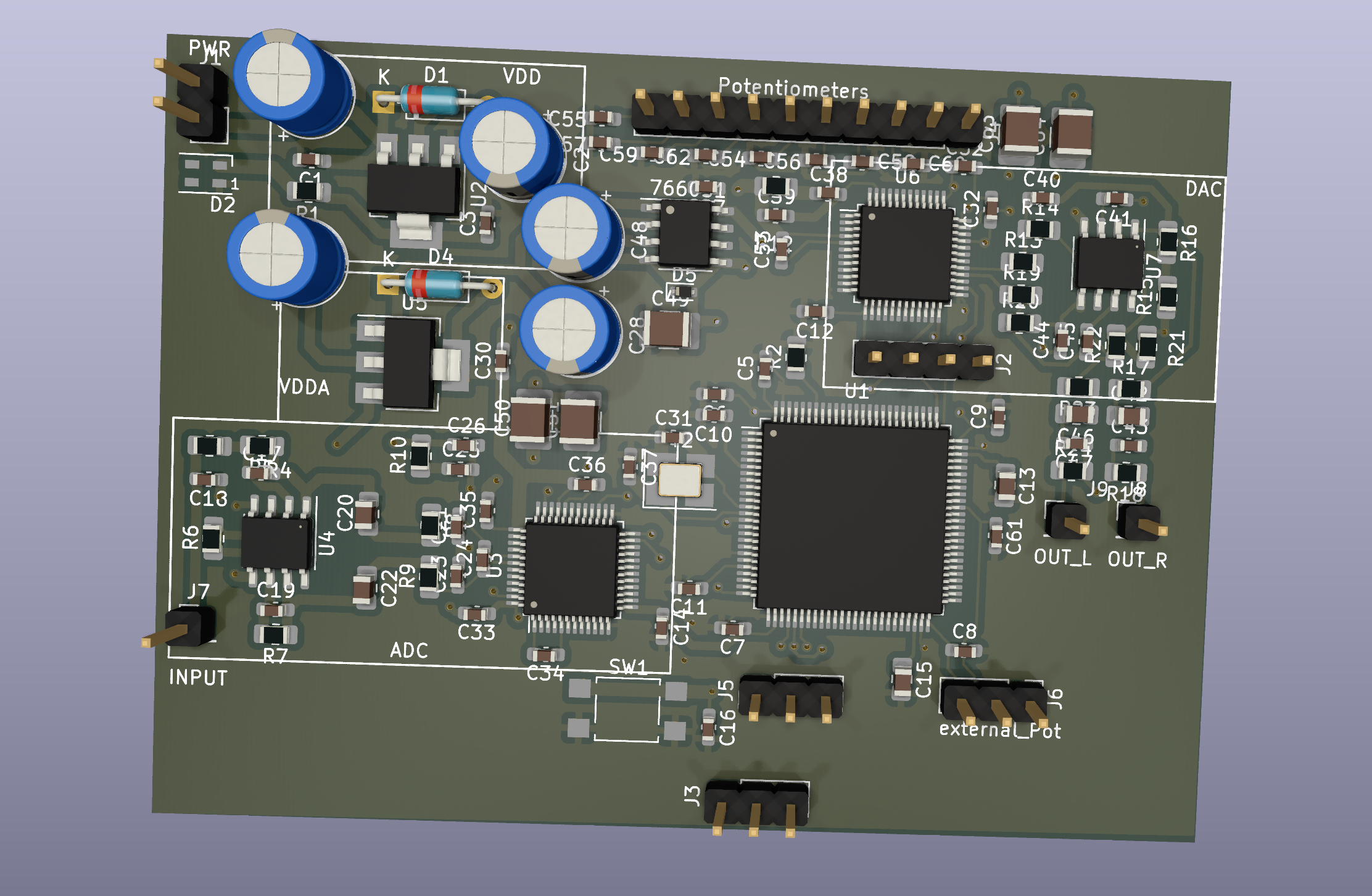
# Jövőbeli tervek

A feladatom során így csak egyetlen effektet tudtam megvalósítani, melynek fejlesztése során rengeteg jelfeldoglozási metódussal „trükkel” találkoztam. A gyakorlatban használt effektek jelentős része olyan alapokon nyugszanak, melyek megértését ebben a félévben szerzett mind szimulációs mind gyakorlati tapasztalat hasznosnak bizonyul. Szeretném továbbfejleszteni ezt a rendszert, több effektet fejleszteni, implementálni. A feladathoz valójában elég egy olyan kapcsolást tervezni, mely valójában egy ADC, DAC rendszert alkot. Itt olyan integrált áramköröket választottam, melyek lehetővé teszik az akár 24 bites feldolgozást 48 kHz-en a jelenlegi 12 bit helyett. Ezen felül sztereo kimenetet helyeztem el, hogy komplexebb hanghatásokat tudjak generálni. Ilyen például a híresebb ping-pong visszhang effekt, ahol a jobb és bal oldalról eltérő időközönként érkezik a késleltetett hang.

## Nyomtatott áramköri terv

A céláramkör feladata a beérkező jelek feldolgozása, majd vissza alakítása. A moduláció mértékét potenciométerek segítségével lehet hangolni. Mivel ebben a rendszerben mind analóg és digitális is megtalálható kifejezetten figyelni kell a megfelelő szeparációra. A digitális jelek könnyen indukálhatnak zajt azok közelében lévő analóg vonalakon, meghiúsítva a mérés hitelességét, vagy ami rosszabb a kimeneti jelünkben hallhatóvá válik. A PCB (nyomtatott áramkör) egyrészt kényelem másrészt szigetelés miatt 4 rétegűre választottam. Külön digitális és analóg tápfeszültséget és földet hoztam létre, melyek külön rétegekre lettek leosztva. Az 1. réteg a digitális föld és felszínen az analóg komponenseket kötöttem egymáshoz. A mikrokontroller 3.3 V feszültségen működik, ezért olyan ADC és DAC berendezéseket választottam melyek szintén működnek ilyen feszültségszinteken, így nem szükséges szintemelő áramkör köztük lévő SPI kommunikációhoz. Az analóg feszültség szintén ezeknek az eszközöknek készült, ugyanúgy 3.3 V, de az illesztőáramkör (műveletierősítő), melyet az adatlapból emeltem át a tervembe, +9 és -9 V feszültséget kap. A zeneiparban gitárosok körében két táplálási megoldás terjedt el. Az egyik egy 9 V-os elem, mely az analóg áramkörök rendkívül alacsony fogyasztásuk miatt hosszútávon használható. Óriási előnye, hogy az ilyen jellegű tápegység nem terhelt kapcsolási zajjal, így ezek ideálisak audió felhasználásra. Mivel az digitális effektek komoly számításigényes eszközök, gyakran elérik az 500 mA-es áramértéket is. Ilyen alkalmazásokhoz ezek az egyszerű elemes tápegység nem elég, dedikált tápegységet kell használni. Ezek az elemes korszak miatt továbbra is +9 V és +18 V értékeket képviselnek. További probléma, hogy effektek egymás után kötésekor a föld csatlakozik a jack csatlakozókon, és csatlakozik a tápegység csatlakozóján.

|  |  |  |
| --- | --- | --- |
| **Eszköz** | **Funkció** | **Választás oka** |
| STM32H730 [13] | mikrokontroller | 1 magos arm M7-es design, 550Mhz, nagy memória |
| AD1974[16] | ADC | SPI, 24 bit, 8-192 kHz, SNR = 107 dB |
| AD1934[15] | DAC | SPI, 24 bit, sztereo kimenet, SNR = 108 dB |
| ICL7660S[12] | negatív feszültség előállítása | Boost módban 35 kHz-es kapcsolási frekvencia, kevés alkatrész |
| OP275[14] | műveleti erősítő (illesztés) | Az adatlap alapján ajánlott műveleti erősítő |



. ábra Nyomtatott áramköri terv

1. The Overlap-Add (OLA) Systems for Audio Analysis and Synthesis - Yi-Wen Liu <https://ocw.nthu.edu.tw/ocw/upload/130/1608/10410%E5%8A%89%E5%A5%95%E6%96%87%E6%95%99%E6%8E%88%E6%95%B8%E4%BD%8D%E6%95%B8%E4%BD%8D%E8%81%B2%E8%A8%8A%E5%88%86%E6%9E%90%E8%88%87%E5%90%88%E6%88%90L4_Lecture-4-OLA-2015.pdf>
2. Real-Time Polyphonic Octave Doubling for the Guitar - Etienne Thuillier <https://aaltodoc.aalto.fi/bitstream/handle/123456789/20158/master_Thuillier_Etienne_2016.pdf?sequence=1&isAllowed=y>
3. Matlab ERB filterbank vizualizáció <https://www.mathworks.com/help/audio/ref/designauditoryfilterbank.html>
4. CMSIS DSP függvények listája <https://www.keil.com/pack/doc/CMSIS/DSP/html/modules.html>
5. Tycobrahe Octavia <https://fuzzcentral.ssguitar.com/octavia.php>
6. Biquad Cascade IIR filter <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__BiquadCascadeDF2T.html>
7. CMSIS v5 könyvtár  
   <https://github.com/ARM-software/CMSIS_5>
8. STM32H755ZI-Q developement board  
   <https://www.st.com/en/evaluation-tools/nucleo-h755zi-q.html>
9. CMSIS FFT <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__RealFFT.html>
10. CMSIS complex <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__cmplx__mag.html>
11. ST HAL könyvtár függvényei  
    <https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf>
12. ICL7660S adatlap  
    <https://docs.rs-online.com/1ff9/0900766b814d54b8.pdf>
13. STM32H730 adatlap  
    <https://www.farnell.com/datasheets/3154034.pdf>
14. OP275 adatlap  
    <https://www.analog.com/media/en/technical-documentation/data-sheets/OP275.pdf>
15. Electro Harmonix POG2 https://www.ehx.com/products/pog2/
16. AD1934 adatlap  
    <https://www.analog.com/media/en/technical-documentation/data-sheets/AD1934.pdf>
17. AD1974 adatlap  
    <https://www.analog.com/media/en/technical-documentation/data-sheets/AD1974.pdf>