



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Magyar Bálint István

**ALACSONY KÖZÉPFREKVENCIÁS
DIGITÁLIS JELFELDOLGOZÓ EGYSÉG
TERVEZÉSE RÁDIÓAMATŐR CÉLRA**

KONZULENS

Orosz György

BUDAPEST, 2012

Tartalomjegyzék

Kivonat.....	5
Abstract.....	6
1 Bevezetés	7
2 Rádióvevők működési elve	9
3 Alkatrészek kiválasztása	15
3.1 A jelfeldolgozó processzor kiválasztása	15
3.1.1 STM32F407VG bemutatása	17
3.1.2 Discovery board	20
3.2 ADC kiválasztása	21
4 Hardvertervezés	23
4.1 Jelkondicionálás	23
4.2 AD konverter illesztése	27
4.3 DA konverter illesztése	28
4.4 Nyomógombok illesztése	29
4.5 Enkóderek illesztése.....	31
4.6 Kijelző illesztése	32
4.7 Külső kommunikáció	33
4.8 A nyomtatott áramkör	34
4.9 Design, elrendezés	36
5 Illesztő szoftverek.....	38
5.1 Általános bemutatás, rendszerterv	38
5.2 AD konverter.....	41
5.3 DA konverter.....	43
5.4 Nyomógombok.....	44
5.5 Enkóderek	45
5.6 Kijelző.....	47
6 SSB jel I-Q demodulációja	48
6.1 Elméleti alapok	48
6.2 A választott demodulátor elrendezés bemutatása	50
6.3 Szűrők méretezése.....	54
7 Mérési eredmények.....	60

7.1 Hardver tulajdonságainak mérése	60
7.2 Demodulátor tulajdonságainak mérése	63
8 Összefoglalás, kitekintés	68
Irodalomjegyzék.....	69

HALLGATÓI NYILATKOZAT

Alulírott **Magyar Bálint István**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot 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 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, 2012. 12. 11.

.....
Magyar Bálint István

Kivonat

A dolgozat témája egy digitális jelfeldolgozó egység tervezése, mellyel az alacsony középfrekvenciára vagy alapsávba lekevert rádiófrekvenciás jelek dolgozhatóak fel. Cél egy olyan hardveregység létrehozása volt, amelyen könnyen és gyorsan lehet különböző jelfeldolgozó algoritmusokat implementálni és tesztelni. Tanulmányoztam néhány rádió felépítését és ezek alapján kiválasztottam egy, a rádiós alkalmazásokban leggyakrabban előforduló algoritmusok implementálására alkalmas platformot. A munkát a választott eszköz megismerésével folytattam. Megterveztem a működéshez szükséges hardveregységeket, melyek működésének helyességéről szimulációkkal bizonyosodtam meg. A kapcsolási rajzokból NYÁK tervet készítettem, amely legyártásra került. Az alkatrészek beültetése után teszteltem a megépített hardvert. Ezek után következett a hardveregységeket kiszolgáló szoftverkomponensek megírása. A megírt szoftverek segítségével egy egyszerű, SSB (Single Side Band) jelek demodulálására alkalmas algoritmust implementáltam. Végül méréseket végeztem, ahol a hardver, illetve az algoritmus tulajdonságait határoztam meg.

Abstract

The topic of the thesis is the design of a digital signal processor based platform which processes low intermediate frequency signals or RF signals which are downmixed to baseband. The goal was to create a hardware unit which allows us the easy and quick implementation and test of various signal processing algorithms. I studied some radio structures, and selected a suitable signal processing platform which is able to perform the most common signal processing operations emerging in radio applications. I designed the hardware units which are required for the operation, and I also simulated the circuit to make sure whether it operates as intended. The printed circuit board plan was created from the circuit's schematic diagram, and was manufactured. After building the circuit, I tested the complete hardware. Next I wrote the low level software components which handle the hardware units. Using these software modules, I implemented a simple algorithm which is able to demodulate SSB (Single Side Band) signals. Finally, measurements were carried out with the hardware and the algorithm in order to investigate whether they comply with the expected operation.

1 Bevezetés

A rádiótechnikában nagyon elterjedt manapság, hogy az alapsávba vagy az alapsávhoz közeli középfrekvenciára (tipikusan 12 kHz) lekevert rádiófrekvenciás jelet valamilyen digitális jelfeldolgozó egység segítségével alakítják a felhasználó számára használható jellé. Az ilyen elven működő rádiók gyűjtőneve: SDR (Software Defined Radio). A digitális feldolgozás azért előnyös, mert a megírt algoritmus nagyon könnyen változtatható és jelfeldolgozó egységet csak egyszer kell megépíteni. Ha például AM (Amplitúdó Moduláció) demoduláció helyett FM-et (Frekvencia Moduláció) szeretnénk használni, egy SDR-ben csupán egy másik algoritmust kell futtatni, teljesen analóg feldolgozás esetén építenünk kell egy FM demodulátor panelt. Továbbá itt nem kell számolni a felhasznált alkatrészek értékeinek szórásából származó pontatlanságokkal sem, egy jól megtervezett digitális szűrő paraméterei mindig ugyanazok. Hátrányként elmondható, a digitalizálás során kvantálási zaj kerül a rendszerbe, illetve fixpontos aritmetika esetén a paraméterek kerekítési hibái jelentősek lehetnek.

A feldolgozó egység lehet számítógép vagy valamilyen jelfeldolgozó processzor. Az előbbi lehetőség kétségkívül olcsóbb, ha egy számítógép rendelkezésre áll. Az utóbbi lehetőség drágább ugyan, de sokkal nagyobb szabadságot kapunk cserébe: a hardvert olyanra készíthetjük, amilyenre szeretnénk, illetve független lesz készülékünk a számítógépektől. Ez mobil eszközöknél előnyös.

Dolgozatomban az utóbbi lehetőséget választottam. Célom egy olyan eszköz létrehozása volt, amelyen igen gyorsan implementálhatók, illetve tesztelhetők különböző algoritmusok, melyeket egy SDR valósít meg. A kapott eredményeket kiértékelve lehet következtetni arra, hogy mit érdemes analóg áramkörökkel megvalósítani, illetve mit kellene szoftveresen implementálni.

Az egységet úgy terveztem meg, hogy legyen rajta kijelző, nyomógombok, illetve néhány enkóder, melyekkel a beprogramozott paraméterek könnyen állíthatók, illetve a kijelzőn láthatjuk a betáplált értékeket.

A 2. fejezetben bemutatok néhány vevőtípust, illetve azt, hogy az általam készített egység hol foglal helyet egy rádióban.

A 3. fejezetben leírom, hogy milyen jelfeldolgozó processzort, illetve AD konvertert választottam és miért.

A 4. fejezetben bemutatom a bejövő jel kondicionálását, illetve a többi hardver egység (kijelző nyomógombok, enkóderek) terveit, illesztésüket, a kapcsolási rajzokat, illetve nyomtatott áramkör terveket.

Az 5. fejezetben a hardver egységek illesztő szoftvereit mutatom be. Ezek fontos részei a rendszernek, hiszen elfedik az alsóbb szinteket, így egy beolvasás a nyomógombokról, enkóderekről nagyon könnyen és gyorsan történhet. Fontos szempont, hogy ezek az algoritmusok minél kevésbé foglalják le a processzor erőforrásait, így meghagyva azt a jelfeldolgozó algoritmusoknak.

Az 6. fejezetben leírom egy SSB jel I-Q elven történő demodulálásának algoritmusát. A rövidhullámú rádióamatőr sávokban gyakorlatilag csak ilyen módon lehet sugározni, és az általam készített egység majdan egy ilyen rövidhullámon üzemelő amatőr rádió része lesz.

A 7. fejezetben a hardvert, illetve a megvalósított algoritmust tesztelem és mérem le.

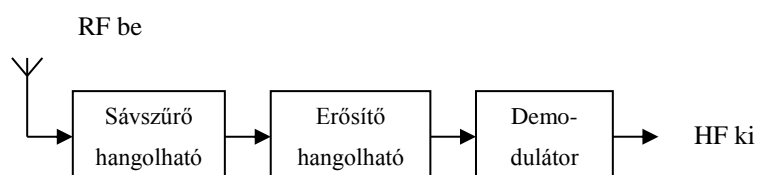
A 8. fejezetben összefoglalom a munkát, illetve leírom a felhasználhatóságot és a továbbfejlesztési lehetőségeket.

2 Rádióvevők működési elve

A rádió a fénynél alacsonyabb frekvenciájú elektromágneses hullámok modulációjával működő, jelátvitelre használt eszköz. Megkülönböztetünk rádióadót és rádióvevőt (illetve rádió-adóvevőt) a jelátvitel irányának megfelelően, mivel ezek működése és belső felépítése jelentősen eltérő. Dolgozatomban csak a rádióvevőkről lesz szó.

A vevő feladata tehát, hogy az antennára kerülő rádiófrekvenciás jelből kiválassza a venni kívánt adó jelét, és azt hallható (digitális adás esetén feldolgozható) jellé alakítsa. A következőkben bemutatok néhány vevőtípust, és ismertetem a működésüket is [1] és [2] alapján.

Egyenes vevő



2-1. ábra Egyenes vevő blokkvázlata

A sávszűrő, illetve az erősítő hangolásával kiválasztunk egy csatornát, és csak azt erősítjük, majd demoduláljuk.

Ennek a konstrukciónak egyetlen előnye van, az pedig az egyszerűsége.

Hátrányai:

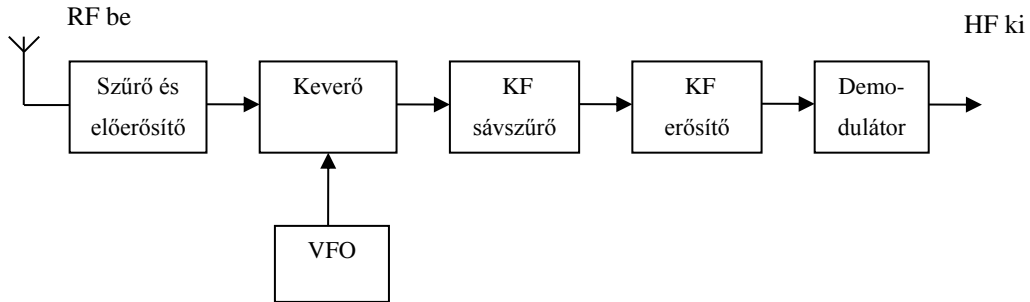
- A szűrőben üzemelő rezgőköröknek csak a kapacitását változtatjuk a hangolás során (ami általában forgókondenzátor), így a rezgőkör jósága is változik (nagyobb frekvencián egyre kisebb). Így előfordulhat, hogy a sáv alján még megfelelő a szelektivitás (tehát a szűrők pont egy csatornát eresztenek át), de a sáv tetejére annyira leromlik a jóság, hogy több csatornát is veszünk egyszerre.

- Több szűrőt kell hangolni egyszerre, illetve biztosítani kell azok együttfutását. Ezt nehéz megoldani, általában többkamrás forgókondenzátorral valósítják meg. Tehát minden rezgőkörnek a forgókondenzátor közelében kell lennie, így elkerülhetetlen bizonyos mértékű csatolás közöttük. Ez gerjedéshez vezethet.

Ilyen vevőtípust gyakorlatilag már sehol nem használnak.

Az egyenes vevőnél a probléma az volt, hogy nem lehet jó minőségű hangolható szűrőt készíteni. Ennek kiküszöbölésére fejlesztették ki a szuperheterodin vevőkészüléket.

Szuperheterodin vevő

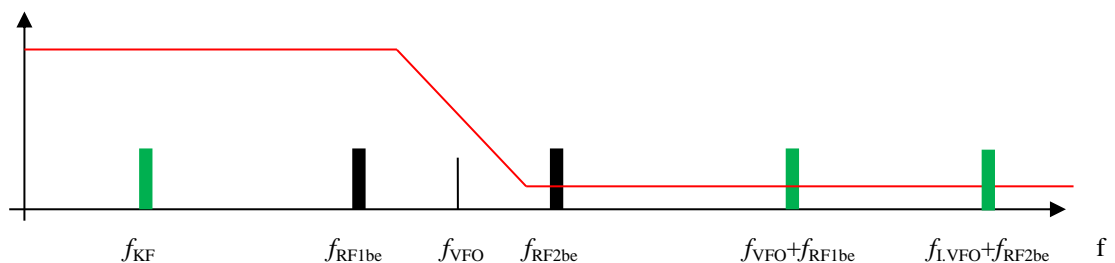


2-2. ábra Egyszeresen transzponált szuperheterodin vevő blokkvázlata

Ennél a konstrukciónál bejövő rádiófrekvenciás jelet erősítjük és szűrjük. Erősíteni általában csak a broadcast vevők, (broadcast: műsorszórás) illetve URH rádiók szoktak, rövidhullámon ez kimarad a nagyon erős adók által okozott esetleges túlvezérlések miatt. A szűrő általában egy sávszűrő, esetleg egy alul- vagy felüláteresztő szűrő, attól függően, hogy alsó, vagy felső keverést szeretnénk végezni. A keverő egyik bemeneti jele a VFO (Variable Frequency Oscillator), a másik pedig a szűrt rádiófrekvenciás jel. A VFO frekvenciájának állításával hangoljuk a rádiót. A keverő összeszorozza a bemenetere kerülő komponenseket. Így kimenetén megjelenik középfrekvenciára keverve: f_{RF1be} , f_{RF2be} , illetve $f_{L.VFO}+f_{RF1be}$, és $f_{L.VFO}+f_{RF2be}$ összegfrekvenciás komponens. f_{RF1be} , f_{RF2be} kiszámítható a következő összefüggések segítségével:

$$f_{RF1be} = f_{VFO} - f_{KF} \quad (1)$$

$$f_{RF2be} = f_{VFO} + f_{KF} \quad (2)$$



2-3. ábra A keverőben lezajló folyamatok (fekete: keverő bemenete, zöld: keverő kimenete, piros: bemeneti szűrő)

Belátható, hogyha (1)-t és (2)-t átrendezzük f_{KF} -re, akkor f_{RF2} és f_{RF2} ugyan oda keveredik le. Legyen f_{RF1be} a venni kívánt állomás, ekkor f_{RF2be} a tükörfrekvencia. Ha itt is dolgozik egy állomás, a jele ugyanúgy f_{KF} -re keveredik le. A két jel többé nem választható szét egymástól, a tükörfrekvencián sugárzó adó jele zavarja a venni kívánt állomást. Ezért szükséges a bemeneti szűrő. Ezt úgy kell méretezni, hogy a tükörfrekvenciát annyira elnyomja, hogy az ne zavarja a vételt.

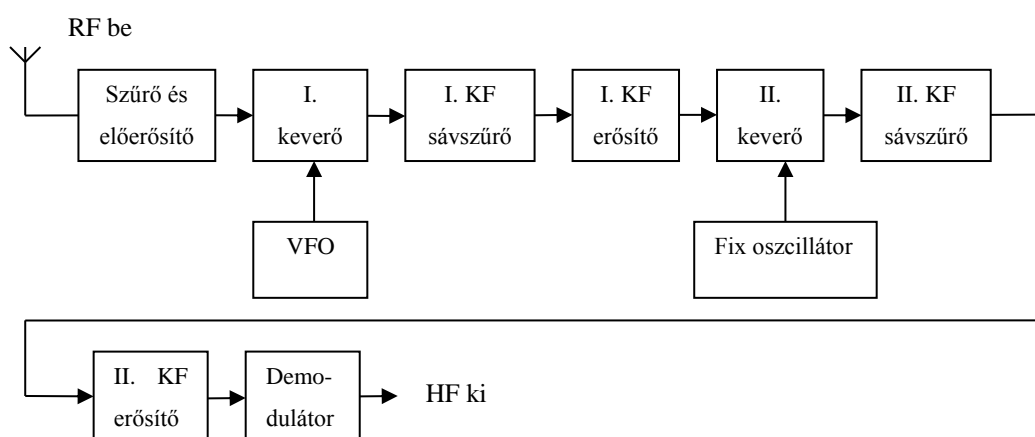
Keverést követően a jelet szűrjük egy jó minőségű szűrővel (általában kvarc- vagy kerámiaszűrővel). Itt jelentkezik a hatalmas előny az egyenes vevőhöz képest, ugyanis a konstans középfrekvenciára kell szűrőt készíteni, így jobb paraméterekkel rendelkezik, mint egy hangolható szűrő. A középfrekvenciás szűrő csak a venni kívánt állomást engedi át, minden mást kiszűr. Megfelelő erősítés után a jel ezután már demodulálható.

Ez a konstrukció jobb paraméterekkel rendelkezik, mint az egyenes vevő, azonban a középfrekvencia megválasztása nem egyszerű:

- Jó minőségű középfrekvenciás szűrő kisebb középfrekvencián könnyebben megvalósítható.
- Ahhoz hogy a bemeneti szűrő ne legyen bonyolult, nagy középfrekvenciára van szükség, hiszen így a tükörfrekvenciák távol kerülnek egymástól.

Ezt a problémát a kétszeresen transzponált szuperheterodin vevő megszünteti.

Kétszeresen transzponált szuperheterodin vevő



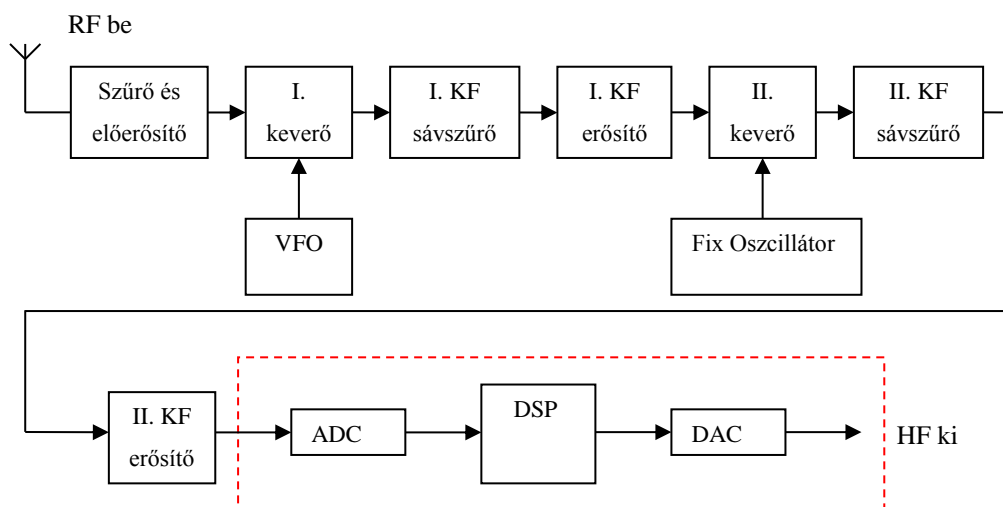
2-4. ábra Kétszeresen transzponált szuperheterodin vevő blokkvázlata

Manapság ez a legelterjedtebben használt vevőtípus. A broadcast vevők (broadcast: műsorszóró) gyakorlatilag mind, az amatőr rádiók nagy része így működik.

Működése az I. KF erősítőig megegyezik az előzőekben ismertetett egyszeresen transzponált változatával. Itt azonban az első középfrekvencia nem kompromisszum kérdése. Ez a frekvencia akkora, hogy a tükörfrekvencia elég távol kerüljön ahhoz, hogy egyszerűen kiszűrhető legyen. Az I. középfrekvencia nagyságrendekkel nagyobb, mint a venni kívánt jel sávzélessége. Ezért ide nem készíthető gazdaságosan egyetlen csatornát áteresztő szűrő, az I.KF szűrő utáni jel néhány csatornát tartalmaz. Ezért a jelet újból lekeverjük ezúttal egy jóval alacsonyabb frekvenciára. Itt az előző fokozat sávkorlátozott és erősített jelét keverjük, így tükörfrekvencián nincs zavaró jel. Ilyen tartományban már gazdaságosan készíthető jó minőségű szűrő, amely csak egy csatornát ereszt át. A II. KF szűrő utáni jelet erősítjük, majd demoduláljuk.

Az SDR-ek egy része is ilyen elvet használ. A rádió felépítése azonban módosul oly módon, hogy a demodulációt digitálisan valósítjuk meg. Ez a kétszer transzponált szuperheterodin SDR vevő.

Kétszer transzponált szuperheterodin SDR vevő



2-3. ábra Kétszer transzponált szuperheterodin SDR vevő blokkvázlata

A jel digitalizálása után a demodulálást egy jelfeldolgozó processzorral végezzük, majd a kész jelet újra analóggá alakítjuk. A második KF frekvencia ekkor tipikusan 12 kHz. Ez nem mindig teljesül, ilyenkor egy harmadik keverőegységet kell alkalmazni, ami 12 kHz-es középfrekvenciára kever le.

Az ilyen felépítésű rádiók előnye a teljesen analógokkal szemben:

- Könnyű változtatni az algoritmust. Ez analóg esetben az áramkör áttervezését jelenti.

- A jelfeldolgozó processzorban megvalósított szűrők nem öregsznek, nem hangolódnak el, és mindig ugyanolyanok (nem szükséges kalibráció).

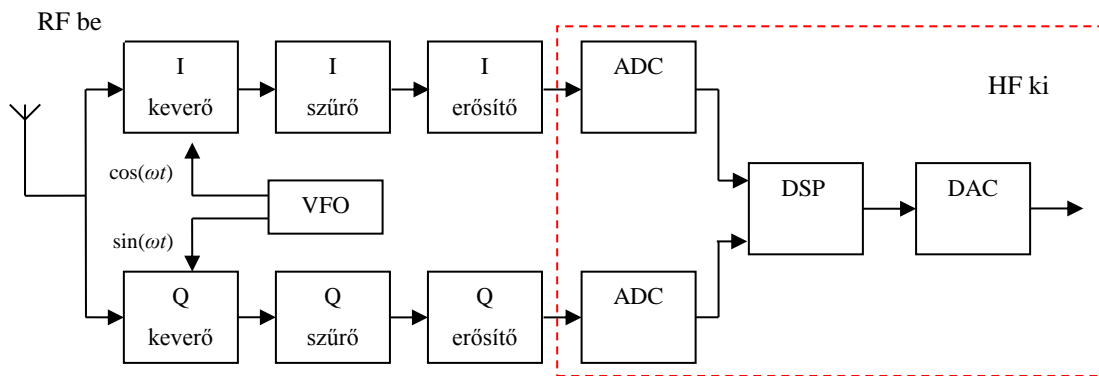
- Digitális adásmódok dekódolása csak így megvalósítható.

Hátrány:

- Az AD átalakító véges bitszáma következtében kvantálási zajt viszünk a rendszerbe.

A digitális feldolgozás előnyeit látva kifejlesztettek egy másik típusú elrendezést is:

Közvetlen I-Q keverésű SDR vevő



2-4. ábra Közvetlen I-Q keverésű SDR vevő blokkvázlata

Itt is két keverő található, de ezek nem egymás után, hanem egymással párhuzamosan működnek. Egy oszcillátor működik, ami két kimenettel rendelkezik, a két kimenet frekvenciája azonos, csupán a fázisukban van 90° -os különbség. Ezek a rádiók a vett jel modulációjától függően alapsávba vagy a 12 kHz-es középfrekvenciára keverik le a jelet. A VFO frekvenciája tehát nagyon közel van a venni kívánt csatornához. A tükrörfrekvenciák átlapolódása itt is megtörténik, azonban ebben az esetben egy komplex jel áll rendelkezésünkre (I, Q: Inphase, Quadrature), amiből a processzorban külön tudjuk választani a két, egymásra lapolódott jelet (ennek módját, és elméleti alapjait a 6. fejezetben részletezem). Ezt analóg körülmények között nehéz feldolgozni, mert a két csatorna sohasem egyforma (erősítőket, sávszűrőket nem tudunk tökéletesen egyformára készíteni), illetve az oszcillátor fáziskülönbsége sem pontosan 90° . Ez azt eredményezi, hogy a tükrörelnyomás nem lesz optimális. Ezt analóg módszerekkel lehetetlen kompenzálni, mert ezek az értékek változnak a hőmérséklettel, idővel, azonban egy DSP-n futtathatunk egy adaptív kompenzáló algoritmust, amely a szükséges kompenzálást elvégzi.

Egyéb lehetőségek:

Megtehetjük azt is, hogy egy nagysebességű AD konverter segítségével a spektrum egy nagy részét digitalizáljuk, és minden műveletet digitálisan végzünk. Erre is van példa: egy holland egyetem (University of Twente) rádióamatőr klubjában megvalósították azt, hogy egy 80 MHz mintavételi frekvenciájú AD konverterrel 0-30 MHz-ig digitalizálják az antennára kerülő spektrumot. A vett nagy mennyiségű adatot egy program feldolgozza, illetve felkerül az internetre is, ahol bárki meghallgathatja [3].

Az azonban elmondható, hogy így nem készíthető olyan jó paraméterekkel rendelkező vevő, mint a kétszeresen transzponált szuperheterodin, vagy I-Q keverésű elrendezésekkel.

Az ábrákon a piros szaggatott vonallal bekeretezett rész a digitális jelfeldolgozó egység. Ez az, melyet szakdolgozatomban megtervezek. A hardvert úgy készítem el, hogy a fent ismertetett mindkét elrendezésben lehessen majd használni.

3 Alkatrészek kiválasztása

3.1 A jelfeldolgozó processzor kiválasztása

Ez a feldolgozó egység elsősorban rádióamatőr célokra készül. Ennek tükrében az alábbiakban ismertetek néhány szempontot a jelfeldolgozó egység kiválasztására.

- Ár: Egyrészt azért használnak SDR-t mert kisebb az előállítási költsége, mint egy ugyan olyan minőségű teljesen analóg elven működő rádióknak. Másrészt saját magam állok a költségeket, és az nem szerencsés, ha egy több tízezer forintos, vagy még drágább hardvert kellene vásárolni. Mint a legtöbb termék fejlesztése során, itt is ez a legfontosabb szempont.

- Beszerezhetőség: Ez nem egy cég számára készülő project, fontos hogy egy esetleges meghibásodás vagy utánépítés alkalmával könnyen beszerezhetőek legyenek az alkatrészek.

- Beépíthetőség: Itt elsősorban a tokozásra gondolok. Visszautalva az előző pontra, ezt az áramkört amatőr körülmények között meg kell tudni építeni, így például egy BGA tokozású DSP szóba sem jöhet. Ha ilyen tokozásról van szó, akkor esetleg egy kisebb demo boardra előre beültetett integrált áramkör megfelelő lehet.

- Előre megírt könyvtárak: Sokat segít, ha van néhány előre megírt könyvtárunk. Ezeket a processzor fejlesztői készítették, és általában kihasználták az adott processzor minden lehetőségét. Ezt mi nem biztos, hogy meg tudjuk tenni, hiszen egy bonyolultabb architektúrát sokáig tart megismerni.

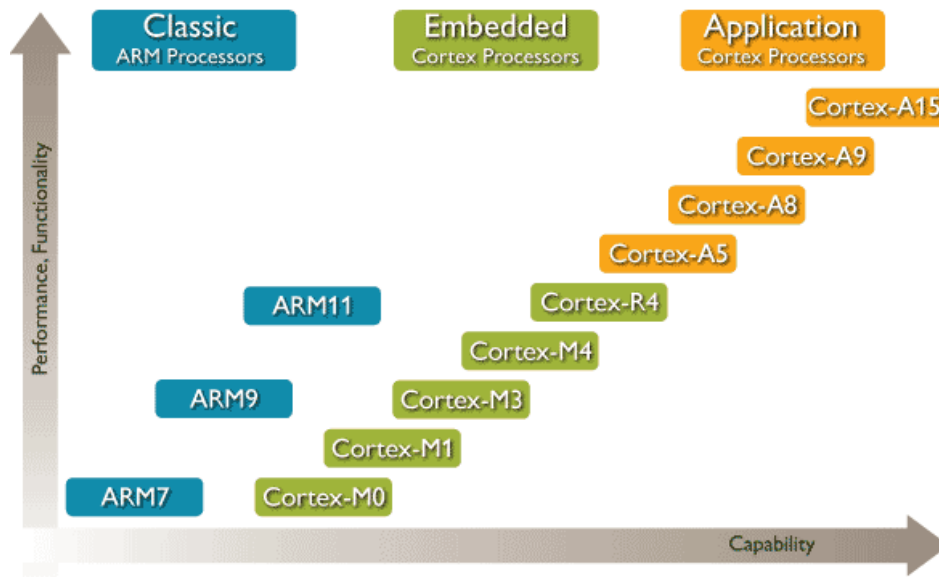
- Hardveres FPU (Floating Point Unit): Egyrészt a bejövő jelek nagy dinamikatarományja miatt, másrészt a megvalósítandó szűrők minősége miatt fontosnak tartom, hogy a processzornak legyen lebegőpontos aritmetikai egysége. E nélkül a lebegőpontos számítások jóval több ideig tartanak, mint FPU-val.

- Ismertség: Fejlesztés során fontos szempont, hogy egy termékről milyen könnyen érhető el internetről más fejlesztők tapasztalata. Ha egy termék széles körben ismert, jó eséllyel található megoldás számos, egyedül nehezen megoldható problémára.

- Szoftveres támogatottság: Itt a fejlesztőkörnyezetre, programozó, debugger eszközökre gondolok. Jó ha stabil, és különböző segítő funkciókkal rendelkezik (például IntelliSense).

- Számítási teljesítmény: Itt nem egy stúdió minőséget kielégítő eszköztől van szó, illetve a hasznos jeltartomány is néhány kHz tehát ezen a téren nem olyan nagyok a követelmények.

A fent felsorolt szempontokat figyelembe véve az STM32F407VG típusú mikrokontrollerre esett a választásom. Az ST Microelectronics gyártja. A kontroller processzora ARM CortexM4 architektúrájú. A Cortex kategóriában a leggyengébbnek számít, az M szériában viszont a legerősebb típus:

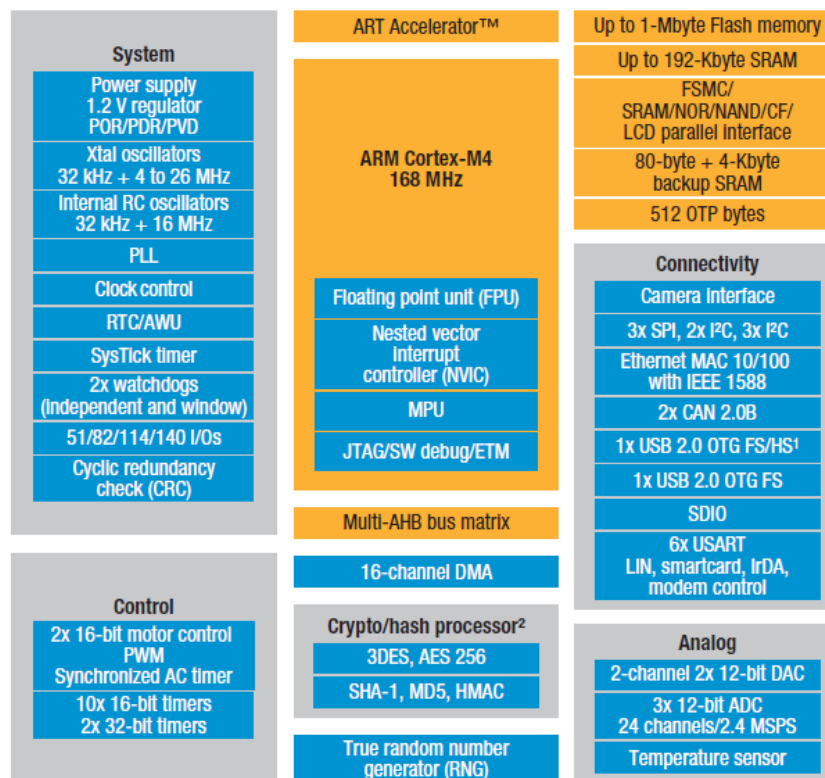


3-1. ábra Az ARM Cortex processzorok teljesítmény szerinti csoportosítása [4]

További ok, hogy ezt a típust válasszam, az az, hogy beszerezhető „STM32F4Discovery” kit formájában. Itt a panelon található a kontrollert programozó egység (STLink), amely átkapcsolható olyan módba is, hogy egy, a paneltól független kontrollert lehessen vele programozni. Így nem szükséges drága programozót venni. Az egész kit itthon hozzávetőlegesen 5000Ft-ba kerül. Maga a chip a gyártónál 7 dollár körül van. Ekkora összegért az Analog Devices is kínál jóval nagyobb számítási teljesítményű lebegőpontos DSP-t. Ehhez azonban egy demo board 500 dollárért kapható.

3.1.1 STM32F407VG bemutatása

Besorolása igen nehézkes, hiszen nem mondható egy általános célú mikrokontrollernek, de DSP-nek sem. Az ilyen kontrollereket az irodalom DSC-nek (Digital Signal Controller) nevezi. Ezek közös jellemzője, hogy van DSP utasításkészletük, és néhány (de nem az összes) DSP-kben megszokott hardveregység is; általában SIMD (Single Instruction Multiple Data), barrel shifter, MAC (Multiple And Accumulate), illetve sok viszonylag gyors PWM, illetve ADC egységük. Az ilyen kontrollereket szenzorok jelének feldolgozására, motorvezérlésre és egyéb szabályozási funkciók ellátására találták ki. Ezen a területen a 16 bites fixpontos architektúrától a 32 bites FPU-t tartalmazóig terjed a kínálat. A fent ismertetett adatok alapján ezt a kontrollert a DSC-k felső kategóriájába sorolnám. Ezek számítási kapacitásuknál fogva már használhatóak különböző audio jelek feldolgozására is.



3-2. ábra A választott controller blokkdiagramja [5]

A blokkvázlatból látszik, hogy a controller sok funkcióval és perifériával rendelkezik. Amelyek fontosak a megépítésre kerülő áramkör szempontjából, azokat részletesen ismertetem.

Hardver tulajdonságok:

- TQFP100 tokozás,
- 80 szabadon felhasználható I/O kivezetés,
- 3.3 V működési feszültség, 1.8 V magfeszültség,
- összes I/O kivezetés 5 V toleráns.

Memóriával kapcsolatos tulajdonságok:

- adat és utasításbusz szélessége 32 bit,
- flash memória: 1 Mbyte (programkód és konstansok tárolására),
- RAM memória: 192 kbyte (változók tárolására),
- A kontroller nem rendelkezik EEPROM memóriával. Erre nincs is szükség, hiszen képes írni a saját flash memóriáját programfutas közben, így az olyan adatok, melyeket a tápfeszültség megszűnése után meg kívánunk őrizni, ide menthetők.

- ART (Adaptive Real Time) memory accelerator: A chip maximálisan 168 MHz-es órajelen tud üzemelni. Ekkora frekvencián a statikus RAM megfelelően működik, azonban a flash memória már nem. Így a programmemória sebessége visszafogná a kontroller működését. Ezt úgy küszöbölik ki, hogy a programmemória egy részét bemásolják egy cache-be, (ami már megfelelő sebességű), és a kontroller innen hozza fel az utasításokat.

Processzor mag tulajdonságai:

- ARM Cortex M4 típusú mag, 32 bites architektúra. Ezt a processzort elsősorban mikrokontrollerekhez fejlesztette ki az ARM Ltd. Sok gyártó használja mikrokontrollereiben. Ez előnyös megírt kód hordozhatósága szempontjából.

- Beépített FPU: Lebegőpontos aritmetikai egység. Segítségével gyorsan végezhetők különböző aritmetikai műveletek egyszeres pontosságú lebegőpontos számokon.

- SIMD utasítások: egy utasítás egy nagyobb adathalmazon végzi el ugyan azt a műveletet. Ez itt csak korlátozott módon áll rendelkezésre: a 16 bites adatokon kettő, a 8 bites adatokon pedig négy adatot tudunk egyszerre kezelni az ilyen típusú utasításokkal.

- DSP utasításkészlet: Néhány a DSP-kben megszokott utasítás, például telítéses aritmetikai műveletek, szorzás és összeadás egy ciklus alatt.

- NVIC (Nested Vectored Interrupt Controller): Néhány fix prioritású megszakítást leszámítva, a többi megszakítást okozó eseményeket szabadon konfigurálhatjuk. Két fajta prioritást különböztetünk meg: a pre-emptive és a sub-priority. A különböző szintű pre-emptive prioritású megszakítások közül a magasabb megszakíthatja a már kiszolgálás alatt lévő alacsonyabbat. A sub-priority-nél a kiszolgálás sorrendben történik, természetesen a prioritásnak megfelelően. Minden egyes megszakítás prioritását 4 biten tároljuk, és megadhatjuk, hogy hány bitet szánunk az egyes típusú prioritásoknak.

- Multi Layer busz: A kontroller fejlett buszmátrixszal rendelkezik. Ennek köszönhetően több periféria cserélhet egymással adatot, illetve írhat az adatmemóriába, ezzel párhuzamosan a processzor megteheti ugyanezt.

Perifériák

- Három darab SPI (Serial Peripheral Interface) modullal rendelkezik, melyből kettő használható I2S (Integrated Interchip Sound) modulként. A legtöbb audio AD illetve DA konverter ilyen kommunikációs protokollt használ.

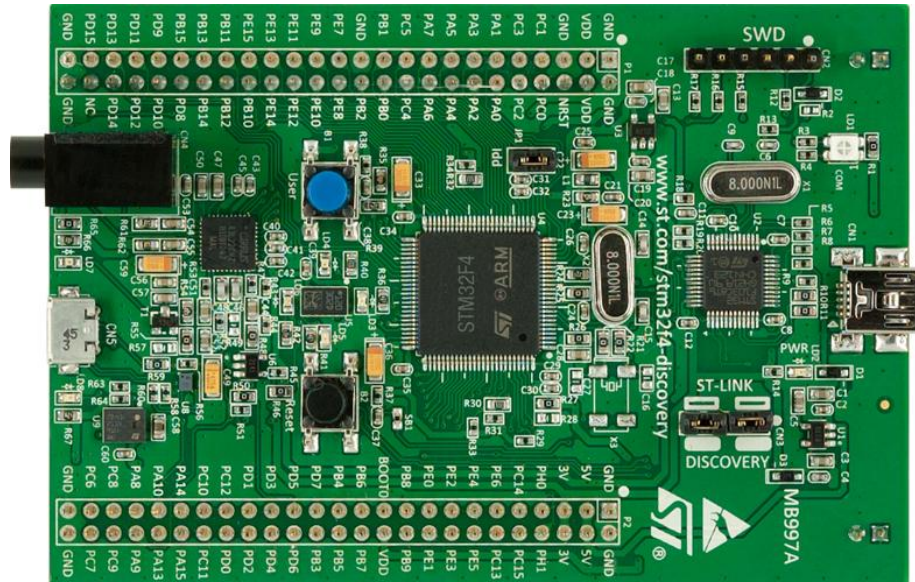
- PLLI2S: Az I2S buszt használó konvertereknek szüksége van rendszer órajelre. Ennek az egész számú hányadosával működik az I2S interface-jük. Ha nem azonos órajelforrásról üzemel a kontroller és a konverter, akkor az órajelük sem lesz tökéletesen azonos, így szinkronizálási hiba lép fel. A kontrollerben van dedikált PLL erre a célra, így szinkronizálhatjuk a kontroller I2S perifériájának működését a külső órajelre, vagy mi állíthatunk elő órajelet a konverter számára.

- Két darab I2C (Inter Integrated Circuit) busz. Az I2S busz mellett a konverterek különböző beállításaira gyakran ilyen buszt alkalmaznak.

- SysTick timer: Nagyon egyszerűen konfigurálható timer, megszakításában különböző rendszerfeladatokat végezhetünk el. Az általam épített áramkörben fontos szerepet kap [7].

3.1.2 Discovery board

A választott mikrokontroller beszerezhető az STM32F4Discovery elnevezésű fejlesztőpanelen is.



3-3. ábra STM32F4 discovery board [6]

Itt a mikrokontroller mellett megtalálható a programozó és debugger (STlink), a mikrokontroller működéséhez szükséges néhány alkatrész, illetve néhány integrált áramkör:

- mikrofon,
- gyorsulásérzékelő,
- sztereó szigma-delta DA konverter fülhallgató meghajtására alkalmas erősítővel,
- USB hardveres illesztője,
- néhány LED és nyomógomb.

A programozóval – két jumper kihúzása után – a paneltől független kontroller is programozható. Annak ellenére, hogy egy fejlesztő panelről van szó, kis méretének (97*66 mm) köszönhetően beépíthető a készülő áramkörbe. A beépítést a panel tervezői megkönnyítették, hiszen két darab 2*50-es tűksor található rajta, melyre a mikrokontroller lábai vannak kivezetve. Így az elektromos csatlakoztatás, illetve a mechanikai rögzítés is könnyen megvalósítható.

3.2 ADC kiválasztása

Az AD konverter kiválasztásánál is nagy körültekintéssel kell eljárni, mert ez szabja meg az egész modul minőségét. Ez esetben is ismertetek néhány szempontot:

- Ár: mint a mikrokontroller kiválasztásánál, itt is egy döntő szempont.
- Beszerezhetőség: hasonlóan az előző ponthoz.
- A kontrollerben is rendelkezésre álló kommunikációs interface-szel rendelkezzen.
- Nagy bitszám: a bemenetre kerülő jelek nagy dinamikatarományja miatt (70-100 μ V-tól 1-2 V-ig) fontos.
- Flexibilitás: a mintavételi frekvencia széles tartományban állítható legyen: például 12 kHz-es középfrekvencia esetén szükség van 48 kHz-es mintavételi frekvenciára, de egy alapsávba lekevert jel esetén 32 kHz is megfelelő.
- Két csatorna: az áramkörrel I-Q jeleket is fel kell tudni dolgozni, ezért elengedhetetlen hogy minimum két csatornával rendelkezzen.
- Szimmetrikus bemenet: Ennek az előnyeit ennél az áramkörnél még nem lehet majd tapasztalni, de a végleges verzióban a jelforrásnál szimmetrikussá alakítva a jelet, és úgy vezetve a konverter bemenetére a rendszer nem lesz annyira érzékeny a különböző zavaró hatásokra.
- Beépített átlapolásgátló szűrő: ezáltal megtakarítható a költséges és nagyméretű külső átlapolásgátló szűrő, amely a mintavételi frekvencia fele feletti komponenseket távolítja el. Ez az opció tipikusan Σ - Δ átalakítóknál áll rendelkezésre.

Az alábbi szempontok figyelembevételével a választásom a PCM1804 típusú Texas Instruments gyártmányú Σ - Δ AD konverterre esett. Minden paraméterében teljesíti az előírt követelményeket:

- két csatorna,
- 3.3 V-os digitális-, 5V-os analóg tápfeszültség,
- 24 bites felbontás,
- 32, 44.1, 48, 96, 88.2, 176.4, 192 kHz lehetséges mintavételi frekvencia,
- beépített átlapolásgátló szűrő,

- bekapcsolható felüláteresztő szűrő (DC komponens szűrésre),
- szimmetrikus analóg bemenet,
- beépített 2.5V-os referencia,
- I2S interface,
- overload jelzés,
- SSOP 28 tokozás.

Egyetlen hátránya az, hogy az egyes funkciók beállítását nem soros kommunikációs csatornán tehetjük meg, hanem az egyes kivezetéseinek alacsony illetve magas szintbe állításával. Így viszonylag sok mikrokontroller kivezetést kell felhasználni a kezelésére.

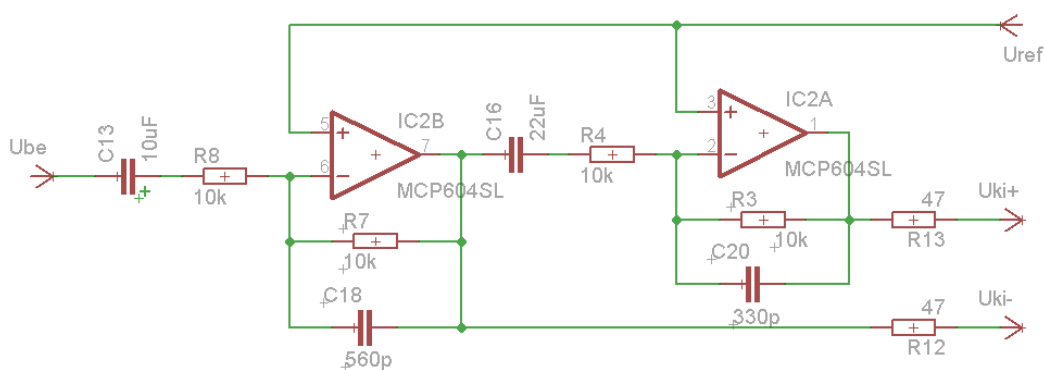
4 Hardvertervezés

Ebben a fejezetben részletesen bemutatom a megtervezett hardvert. A panel analóg és digitális jeleket is tartalmaz, ezek, illetve a tápfeszültségek elválasztására külön figyelmet fordítottam.

4.1 Jelkondicionálás

Az AD átalakító bemenete szimmetrikus, a tesztelhetőség érdekében azonban az áramkört aszimmetrikus bemenetűre terveztem meg. Így akár egy számítógép hangkártyája segítségével is generálhatunk mérőjeleket. Szükség volt egy olyan áramkörre, amely az aszimmetrikus jelekből szimmetrikus jeleket állít elő. Az AD átalakító analóg része egyszeres, +5 V-os tápfeszültségről üzemel. Ezért az előállított szimmetrikus jelek ofszetjét el kell tolni a referencia feszültséggel, amely jelen esetben $U_{Ref}=+2.5V$.

A két feladatot egyszerre, a következő áramkör segítségével valósítottam meg:



4-1. ábra Jelkondicionáló áramkör az AD átalakító számára

Az első fokozat átviteli függvénye:

$$U_{ki} = -U_{be} \frac{\frac{1}{sC_{18}} \times R_7}{\frac{1}{sC_{13}} + R_8} + U_{ref}$$

Az áramkör bemenetére kerülő jel a C_{13} kondenzátoron, és R_7 ellenálláson kerül a műveleti erősítő invertáló bemenetére. Ez a két elem felüláteresztő szűrőként viselkedik. A vágási frekvenciát úgy kell megválasztani, hogy a bemenetre kerülő DC komponens

vágja, de a hasznos jelet ne csillapítsa. A visszacsatoló ágban található C_{18} kondenzátor és R_7 ellenállás párhuzamos kapcsolása pedig egy aluláteresztő szűrőt alkot. Itt a vágási frekvenciát a mintavételi frekvencia felére választottam meg:

Feluláteresztő szűrő vágási frekvenciája: $f_{\text{cutoffHP}} \sim 1$ Hz

Aluláteresztő szűrő vágási frekvenciája: $f_{\text{cutoffLP}} \sim 24$ kHz

További követelmény még, hogy az erősítés abszolút értéke közepes frekvencián egységnyi legyen. Ebből következik, hogy a két ellenállás értékét egyformára kell választani. Az alkatrészek értékének kiválasztásánál egy szabad paraméterünk van. Így az ellenállások értékét 10 k Ω -ra választottam.

Ezekből az adatokból már meghatározhatjuk a két kondenzátor értékét:

$$C_{13} = \frac{1}{2\pi f_{\text{cutoffHP}} R_8}$$

C_{13} értéke 16 μF -ra adódott. Ilyen értékkel kondenzátor nem kapható, ezért 10 μF -osat választottam. Így a vágási frekvencia $f_{\text{cutoffHP}} = 1.69$ Hz-re módosul, de ez így is elfogadható.

$$C_{18} = \frac{1}{2\pi f_{\text{cutoffLP}} R_7}$$

C_{18} értéke 663 pF-ra adódott. Az AD átalakító rendelkezik átlapolásgátló szűrővel. Ez a szűrő azonban csak a Σ - Δ modulátor mintavételi frekvenciájának feléig működik. Ennél nagyobb frekvenciáknál ugyanúgy fellép az átlapolódás jelensége. Ez átlagos audio berendezésekben nem kritikus, de egy rádióban előfordulhat, hogy valamilyen csatolás folytán a bemenetre kerüljön néhány megahertzes jel. Az aluláteresztő szűrőnek ezen a tartományon van nagy szerepe. Ez a frekvencia 48 kHz-es mintavételi frekvencia és 128-szoros túlmintavételezés esetén 3.072 MHz. Ekkor a megtervezett szűrő már több mint 40 dB-es elnyomást produkál. Ezen megfontolások alapján a kondenzátor értékét nem a közelebb eső 680 pF-ra, hanem a kisebb 560 pF-ra választottam. Így a mintavételi frekvencia feléhez közeledve az átvitel simább lesz, és a nagyfrekvenciás tartományban sem lesz jóval kisebb az elnyomás. Ezek alapján a valós $f_{\text{cutoffLP}} = 28.4$ kHz-re módosul.

A műveleti erősítő neminvertáló bemenetére kötöttem a referenciafeszültséget. Mivel az invertáló bemenet egyenáramú szempontból „lebeg” (C_{13} miatt), továbbá a

visszacsatolás miatt, a kimeneten - bemenetet szabadon hagyva - pontosan a referenciafeszültség fele fog megjelenni. Így a szimmetrikus jel „-” felét előállítottuk.

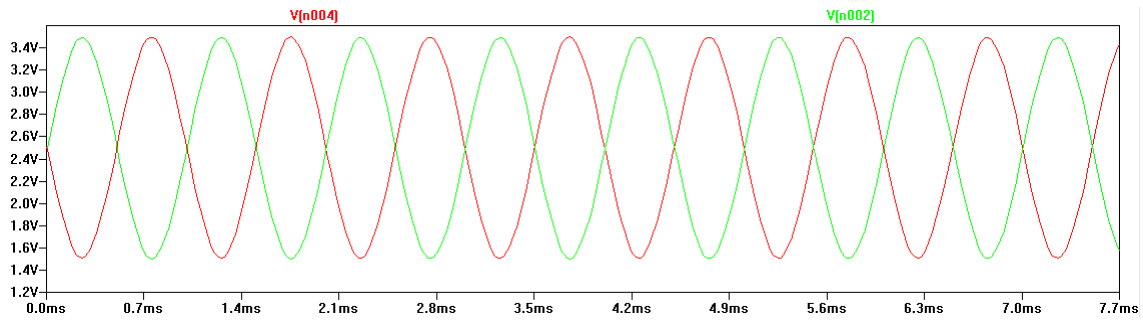
A „+” jel előállítása is hasonló megfontolások alapján történik, de ez esetben a bemenet az előző fokozat kimenete lesz. Így egy újabb invertáló erősítőt lehet alkalmazni, amely visszaforgatja a jel fázisát. Az előző fokozat által beállított ofszetet el kell távolítani, hogy ez a fokozat ismét eltolja a jelet. Erre egy 22 μF -os kerámia kondenzátort választottam. Azért szükséges kerámia, mert az előző fokozat kimeneti jele is ofszeten van, és ennek a fokozatnak a bemenete is, tehát a kondenzátoron csak a jel által okozott váltakozó feszültség lép fel. A tantál kondenzátor polarizált, ennél fogva nem léphet fel váltakozó feszültség a két kivezetése között, mert a fordított polaritás következtében meghibásodna. Azért nagyobb a kapacitás, mint az előző fokozatban, mert ennek az erősítőnek a bemenetén már érvényesül az előző fokozat felüláteresztő szűrőjének hatása. Nagyobb kapacitás esetén az aszimmetria csak kisebb frekvenciákon érvényesül. Ugyanilyen megfontolásból lett az aluláteresztő szűrő kondenzátora 330 pF. A két ellenállás ugyan az, mint előző esetben.

Mivel két csatorna van, ezért összesen négy darab műveleti erősítőre van szükség. Az 5 V-os tápfeszültség miatt csak rail-to-rail típus jöhet számításba. Választásom az MCP604 típusú Microchip gyártmányú műveleti erősítőre esett. Egy tokban négy darab műveleti erősítő foglal helyet. Be- illetve kimenetei is rail-to-rail típusúak mindkét esetben a jel 25 mV-ra közelítheti meg a negatív illetve pozitív tápot. A két kimeneten azért egy 47 Ω -os ellenálláson keresztül kerül a jel az AD átalakító bemenetére, mert a műveleti erősítő gyártója szerint egységnyi erősítés esetén a kapacitív terhelés rossz hatással van az erősítésre. A tápfeszültség szűréséről egy darab 100 nF-os kondenzátor gondoskodik.

A megtervezett áramkört szimuláltam az LTSpice elnevezésű programmal. Az alábbiakban ismertetem a szimulációs eredményeket.

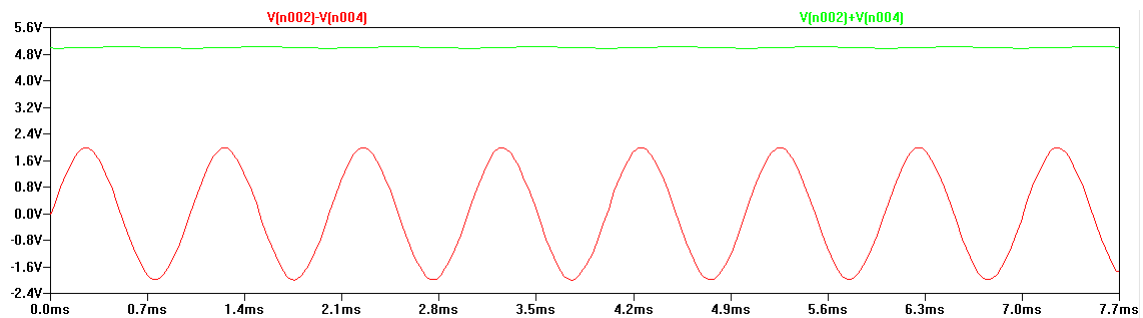
Szimulációs paraméterek:

- Tápfeszültség: +5 V
- Ofszet feszültség: 2.5 V
- Mérőjel: 2 V_{p-p} 1 kHz szinusz
- A generátorok, és az alkatrészek ideálisak



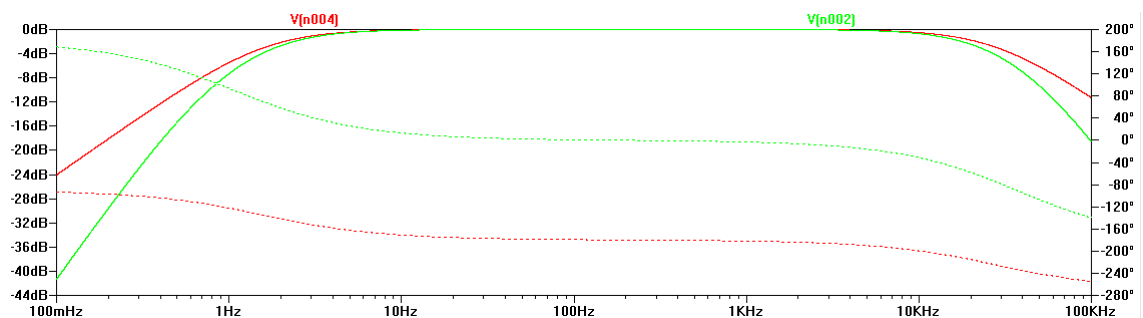
4-2. ábra Időtartománybeli szimulációs eredmények: két kimenet jelalakja

A zöld a „+”, a „-” kimenet pedig a piros jelzésű. Az ábrán jól látható a 2.5 V-os offset, illetve az, hogy a két jel ellenfázisban van. Továbbá leolvasható, hogy a különbségi jel az eredeti jel kétszerese lesz. Tehát annak ellenére, hogy egyik fokozat sem erősít, a szimmetrikus jel mégis kétszerese lesz az eredeti jelnek.



4-3. ábra Időtartománybeli szimulációs eredmények: összeg illetve különbségi jelek

Piros színnel a szimmetrikus jelet ábrázoltam (két jel különbsége). A kétszeres erősítés itt még szemetűnőbb. Zöld színű a két jel összege, ez a közösjel. Értéke ebből az ábrából nem meghatározható, de kinagyítva leolvasható, hogy a kétszeres offset feszültségre szuperponált $40 \text{ mV}_{\text{p-p}}$ szinusz. Ez hozzávetőlegesen 100-ad része a különbségi jelnek. A különbségi jel sem pontosan $4 \text{ V}_{\text{p-p}}$, hanem valamivel kevesebb. Oka a két csatorna aszimmetriája. Mivel itt nem egy mérőátalakítóról van szó, ezek az eredmények elfogadhatóak.



4-4. ábra Bemeneti fokozat szimulált átviteli karakterisztikája

közösjel kimenetet 10 μ F+100 nF-dal szűröm. Ezeket az értékeket a gyártó ajánlotta a megfelelő működés érdekében.

Az átalakító összes adat/konfiguráló kivezetését rákötöttem a mikrokontroller I/O kivezetéseire, így az összes funkcióját használhatjuk. Az összekapcsolásokat az alábbi táblázat tartalmazza:

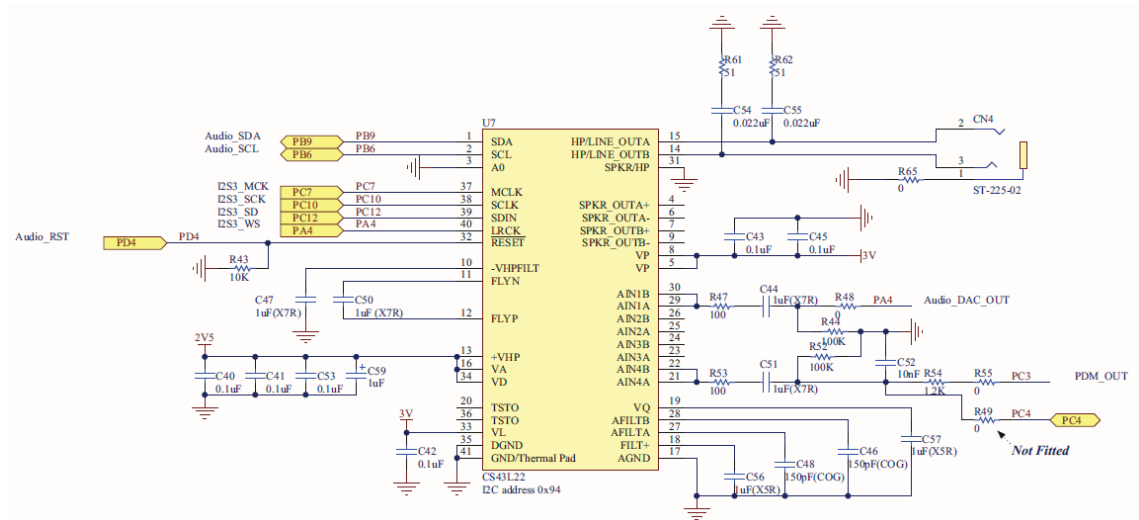
4-1. táblázat Az AD konverter, és a mikrokontroller lábainak összerendelése

Funkció	Neve az AD átalakítón	Helye a mikrokontrolleren
Jobb/Bal csatorna túlvezérlés	OVFR, OVFL	PE12,PE10
Reset	RST	PE14
Rendszer órajel	SCKI	PC6
I2S jobb/bal csatorna adat kiválasztás	LRCK	PB12
I2S busz órajel	BLK	PB13
I2S adat kimenet	DATA	PB15
Felüláteresztő szűrő ki/be kapcsolás	BYPASS	PD11
Túlmintavételezés és órajel kiválasztás	OSR2, OSR1, OSR0	PD9, PD10, PD8
Adattípus kiválasztás	FMT1, FMT0	PB14, PE15
Master/Slave mód kiválasztás	M/S	PB11

Az I2S buszhoz tartozó kivezetéseket (SCKI, LRCK, BLK, DATA) csak egy helyre köthettem a mikrokontrolleren, azokra a szabad lábakra, ahová multiplexelhetők. A többi kivezetés összerendelése nem volt kritikus.

4.3 DA konverter illesztése

A DA átalakító típusa CS43L22. Ez megtalálható az STM32discovery boardon, ezért kézenfekvő volt, hogy használjam. Itt csupán a kapcsolási rajzot, és a kommunikációs interface bekötését írom le. A kapcsolási rajz a 4-6. ábrán látható.



4-6. ábra DA átalakító kapcsolási rajza [8]

4-2. táblázat A DA átalakító és a mikrokontroller lábainak összerendelése

Funkció	Neve a DA átalakítón	Helye a mikrokontrolleren
I2C busz adat	SDA	PB9
I2C busz órajel	SCL	PB6
Rendszer órajel	MCLK	PC7
I2S busz órajele	SCLK	PC10
I2S busz adatbemenet	SDIN	PC12
I2S busz jobb bal csatorna kiválasztás	LRCK	PA4
Reset	RESET	PD4

Látható, hogy az I2C kommunikációs busz miatt ez a konverter jóval kevesebb szabad mikrokontroller lábat használ el.

4.4 Nyomógombok illesztése

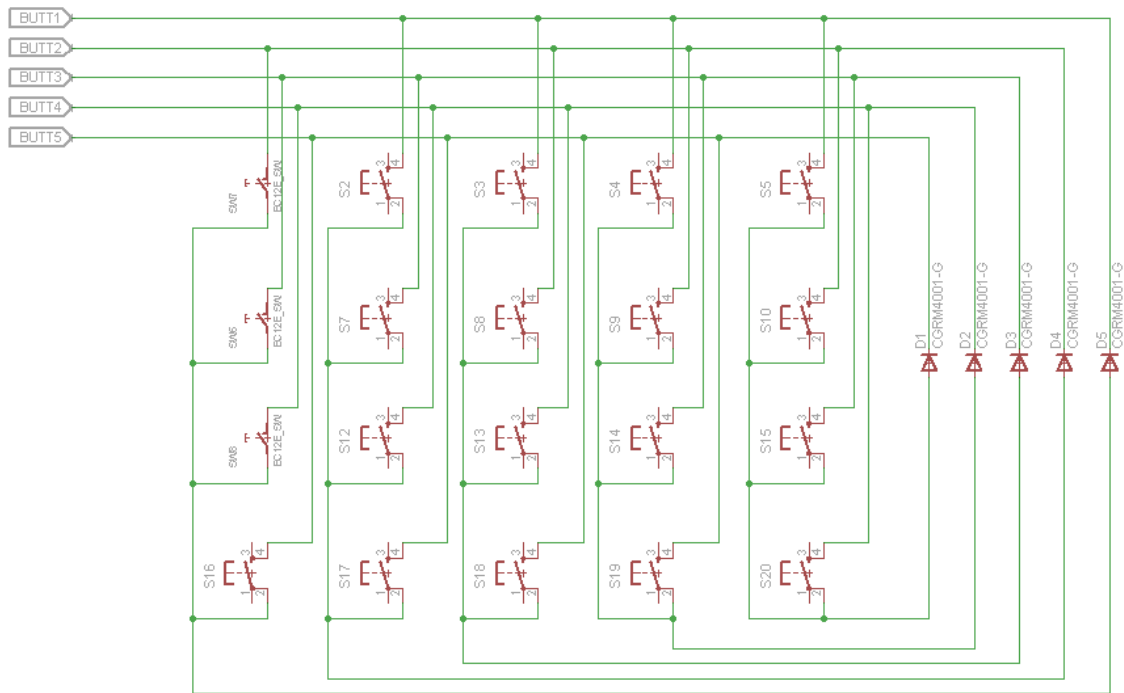
A készülő áramkörhöz sok nyomógombra volt szükség. Ha például olyan az algoritmus, hogy egy paramétert manuálisan lehet állítani, a nagy változásokat enkóderrel sokáig tartana beállítani. Ezért annyi gombra van szükség, hogy egy numpad kialakítása mellett még maradjon néhány szabad gomb, amire más programozható. Így végül 17

nyomógomb, került felhasználásra, továbbá három olyan enkóder, melynek tengelyének benyomása szintén egy kontaktust zár. Tehát 20 kontaktust kell összesen lekezelni. Az elegendő, hogyha a rendszer egy időben csak egy gomb lenyomását képes érzékelni.

Kézenfekvő hogy a gombokat 20 darab szabad mikrokontroller kivezetésre kötjük. Így azonban sok kivezetést használnánk el. Lehetséges lenne különböző digitális áramkörök alkalmazása, például dekódoló, de így még több IC kerülne a panelra. Helyette a charlieplexig megoldást alkalmaztam. Ez egy különleges fajtája az egyszerű multiplexelésnek. Ezzel a módszerrel lehet N db vezetékkel a legtöbb nyomógombot kezelni:

$$X=N^2-N$$

Ahol X a kezelt nyomógombok száma. Tehát a 20 darab nyomógomb kezeléséhez 5 db mikrokontroller I/O kivezetés szükséges.



4-7. ábra Nyomógombok kezelése

Működése:

Első esetben a BUTT1 nevű kivezetést a mikrokontrolleren kimenetnek állítjuk, BUTT2,3,4,5 pedig bemenetnek, úgy hogy a belső felhúzó ellenállást bekapcsoljuk. A BUTT1 kimenetet logikai 0 szintbe állítjuk. Megvizsgáljuk a többi bemenet értékét. Ha valamelyik értéke 0, akkor a hozzá tartozó gombot megnyomhatták. Ebben az esetben

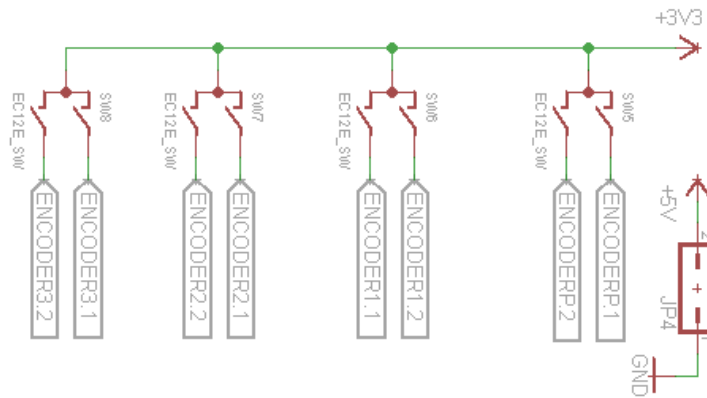
BUTT2 S2, BUTT3 S3, BUTT4 S4, BUTT5 pedig S5 megnyomását jelzi. Ezt követően BUTT2 kivezetés a kimenet, a többi pedig bemenet. A letapogatás az előzőekben leírtakkal megegyezik. Ezt követően az összes eredményt ellenőrizni kell. Ha egy gombnál több mutatott megnyomott értéket, akkor az áramkör sajátosságaiból következően nem tudjuk megmondani melyik gombok voltak azok. Az ilyen eseteket úgy kezeltem, mintha egy gombot sem nyomtak volna le.

4-3. Táblázat Nyomógombmátrix csatornáinak, és a mikrokontroller kivezetéseinek összerendelése

Csatorna neve	Helye a mikrokontrolleren
BUTT1	PA8
BUTT2	PA15
BUTT3	PC11
BUTT4	PD0
BUTT5	PD1

4.5 Enkóderek illesztése

Összesen négy darab enkódert terveztem az áramkörbe. Ezek közül három egyszerű mechanikus, 30 impulzus/fordulat típusú. Ezek olyan felépítésűek, hogy egy nyomógombot is tartalmaznak, ezt a tengelyük benyomásával használhatjuk. A negyedik enkóder is ilyen felépítésű lett, de a nyomógombja nem lett bekötve. A nyomtatott áramkört úgy alakítottam ki, hogy ide optikai elven működő enkóder is csatlakoztatható legyen. Ez azért szükséges, mert ha ezzel az áramkörrel vezéreljük egy rádió VFO-ját, akkor fontos, hogy finoman állítható legyen a frekvencia. Ilyen enkóder a magas költsége miatt nem került beépítésre, de a későbbi beépítés lehetőségét biztosítottam a két tápfeszültség odavezetésével. Az enkódereket úgy kötöttem be, hogy a közös kivezetést +3.3 V-on van, a másik két kivezetést pedig a mikrokontroller egy-egy bemenetnek definiált kivezetésén.



4-8. ábra Enkóderek kezelése

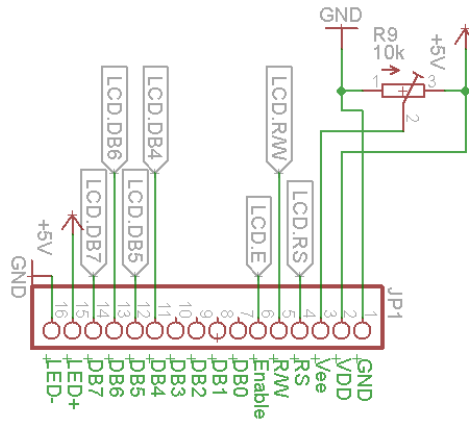
4-4. táblázat Enkóderek kontaktjainak összerendelése a mikrokontroller kivezetéseivel

Enkóder neve, kivezetés száma	Helye a mikrokontrolleren
ENCODERP1	PA2
ENCODERP2	PA3
ENCODER1.1	PE5
ENCODER1.2	PE6
ENCODER2.1	PE11
ENCODER2.2	PE9
ENCODER3.1	PB7
ENCODER3.2	PB8

4.6 Kijelző illesztése

Kijelzőnek 2004 típusú 4x20 karakteres HD47780 kompatibilis típust választottam. A kommunikációs interface 4 bites lett, így takarékoskodva a kivezetésekkel. A kijelző feszültségét egy helitrimmer segítségével állíthatjuk be egy kívánt értékre. Ezzel a kontraszt változik. A kijelző kommunikációs interface-ja 5 V-ról működik, a mikrokontrolleré 3.3 V-ról. Az LCD adatlapja szerint az IC már 2.2 V-tól logikai

„1”-nek érzékeli a bemenetére kerülő jelet, így összeköthetjük őket bármiféle szintillesztő nélkül.



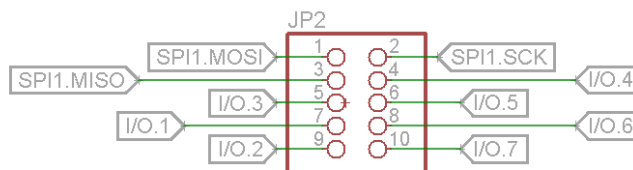
4-9. ábra LCD kijelző bekötése

4-5. táblázat LCD kijelző és a mikrokontroller csatornáinak összerendelése

Funkció	Helye a mikrokontrolleren
RS (Register select)	PA1
R/W (Read/write)	PC4
E (enable)	PC5
DB4 (Adatbusz 4. bit)	PB0
DB5 (Adatbusz 5. bit)	PB2
DB6 (Adatbusz 6. bit)	PE8
DB7 (Adatbusz 7. bit)	PE7

4.7 Külső kommunikáció

A panelra kivezettem egy SPI buszt, és 7 db I/O kimenetet. Ezek szabadon konfigurálhatók, az eszköz külvilággal történő kommunikáció megvalósítására.



4-10. ábra Külső kommunikáció bekötése

4-6. táblázat Külső kommunikáció és a mikrokontroller összerendelése

Funkció	Helye a mikrokontrolleren
SPI MOSI	PB5
SPI CLK	PA5
SPI MISO	PB4
I/O 1	PE4
I/O 2	PC13
I/O 3	PE3
I/O 4	PD7
I/O 5	PD2
I/O 6	PD3
I/O 7	PD6

4.8 A nyomtatott áramkör

A következőkben leírom az analóg és digitális áramkörök tervezésénél figyelembe vett szempontokat.

Analóg áramkörök

Lehető legrövidebb jelvezetékek, így kevésbé tud zaj kerülni az jelbe. Ezért az összes ellenállás, és azok a kondenzátorok, melyek előállíthatók ekkora méretben, 0603 méretűek lettek. Ennél kisebb méretűeket nem lehet kézzel beforrasztani. Az ellenállások fémrétegek, a kerámia kondenzátorok pedig NP0 anyagúak. A fémréteg ellenállás zajszegényebb, mint például a szénréteg. Az NP0 anyagú kondenzátorok pontosak, és alacsony átvezetéssel rendelkeznek. A nagyobb kondenzátorok tantál elektrolit kondenzátorok. Ezek jobb paraméterekkel rendelkeznek, mint az alumíniumból készült típusok. Kisebb a soros induktivitásuk és ekvivalens soros ellenállásuk, emiatt analóg jelfeldolgozásban gyakran használnak ilyen típust.

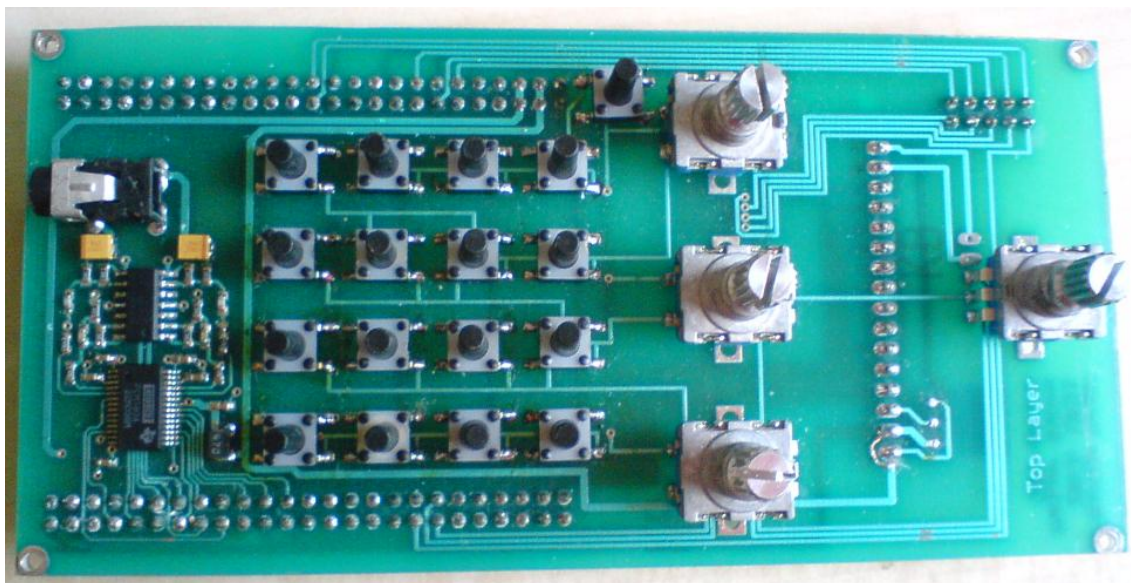
Az áramkör kétoldalas furatgalvanizált technológiával készült el. Az analóg áramkörök a felső rétegen (top layer) helyezkednek el, ez alatt található réteg (bottom) „teleföld” lett, és csak néhány vezeték megy itt, amit más módon nem lehetett volna vezetni, illetve néhány alkatrész, amelyek a másik oldalt nem fértek el. A földelés kialakításánál

a csillagpontos földelés és a föld síkok (ground plane) módszerét kombináltam. Az analóg rész alatt a fólia két részre oszlik: az analóg és a digitális föld részekre. Ezeket külön vezetem oda, ahol a panel csatlakozik a földhöz, így eleget téve a csillagpontos kialakításnak [10].

Az egész analóg részt úgy helyeztem el, hogy az STM32F4discovery panel DA átalakítójának jack kimenete, és az AD átalakító ugyancsak jack bemenete egy oldalon legyenek.

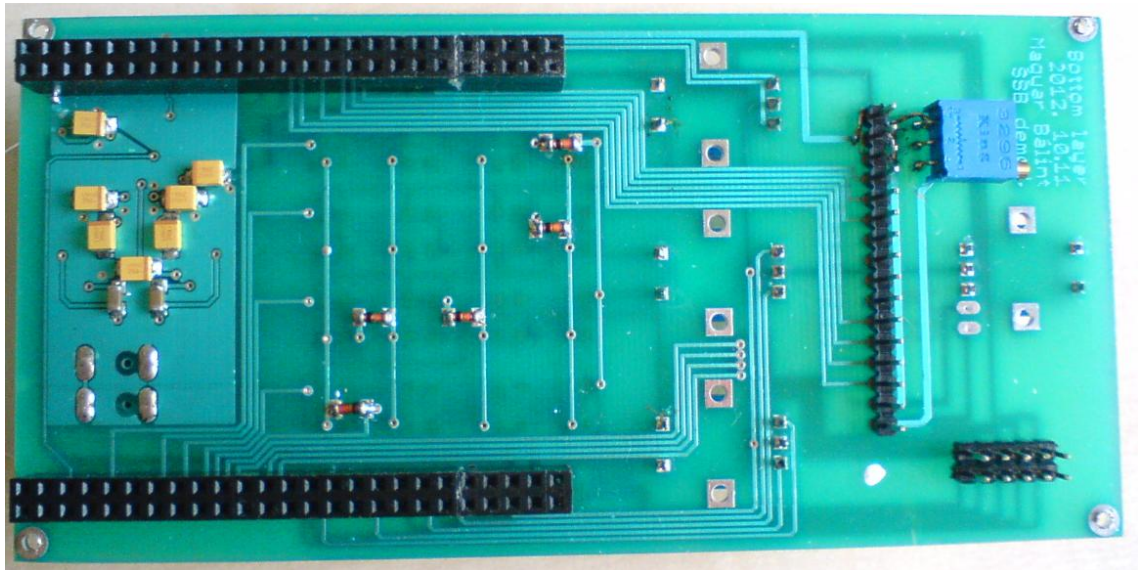
Digitális áramkörök

A digitális áramkörök nagy része a Discovery panelen volt rajta. Nekem a nyomógombok, a kijelző és az enkóderek elhelyezése volt a feladatom. Itt az ergonómia játszott döntő szerepet: a nyomógombok és az enkóderek kényelmesen kezelhetőek legyenek. A kijelző nagy mérete miatt nem kapott helyet a panelon, mert ez esetben a panel hozzávetőlegesen kétszer ekkora lett volna, ami a költségek jelentős növekedésével járt volna. Ezért a kijelzőnek egy tükörsort alakítottam ki, ide szalagkábellel csatlakoztatható.



4-11. ábra Az elkészült nyomtatott áramkör „Top layer”

4-11. ábrán bal oldalt látható a jack csatlakozó alatta a műveleti erősítő, az alatt az AD átalakító. Ennek a felső része az analóg, az alsó pedig a digitális rész. Az analóg résztől jobbra láthatók a nyomógombok, illetve az enkóderek. Az elkülönített enkóder helyére építhető be optikai típusú, ez látható a panelon szabadon lévő két furatból. Ezek egyike +5V a másik a GND. Így gyakorlatilag ide bármilyen típus csatlakoztatható.

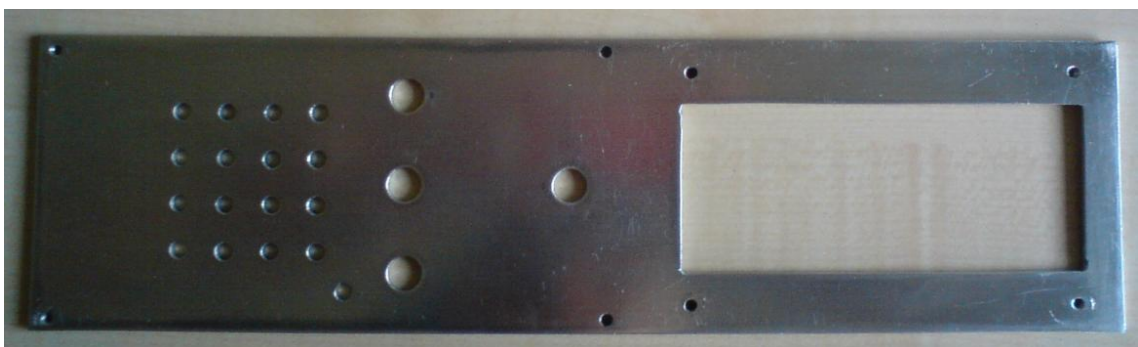


4-12. ábra Az elkészült nyomtatott áramkör, „Bottom layer”

A 4-11. ábrán látható az analóg áramkörök számára kialakított „teleföld” (bal oldalt közepén), a két földsíkot elválasztó vágás (bal oldalt fent a legfelső kondenzátortól indul) a két hüvelysor, melybe a Discovery panel csatlakozatható (bal oldalt fent és lent), az LCD kijelző számára készült túsoros (jobb oldalt közepén), illetve a külső kommunikációt megvalósító túsoros (jobb oldalt lent).

4.9 Design, elrendezés

Az áramkör célja az, hogy SDR jelfeldolgozó algoritmusait teszteljek és mérjek rajta. Ennek következtében az áramkört sokat kell mozgatni, például a számítógéptől, ahol a programozás történik, oda ahová az algoritmust tesztelni lehet. Ezért ha a kijelző egy szalagkábelen „lóg”, és semmilyen más rögzítése nincs, hamar kontakthibák léphetnek fel. Ezért egy alumínium lemezt használtam fel (4-13. ábra), melyre rögzíthető minden.



4-13. ábra Megmunkált alumínium lemez, melyre az áramkör és a kijelző rögzíthető



4-14. ábra Az alumínium lemezre rögzített áramkör (előlnézet)

A 4-14. ábra látható az elkészült áramkör előlnézeti képe. A nyomógombokhoz és enkóderekhez könnyű hozzáférni.



4-15. ábra Az alumínium lemezre rögzített áramkör (hátnézet)

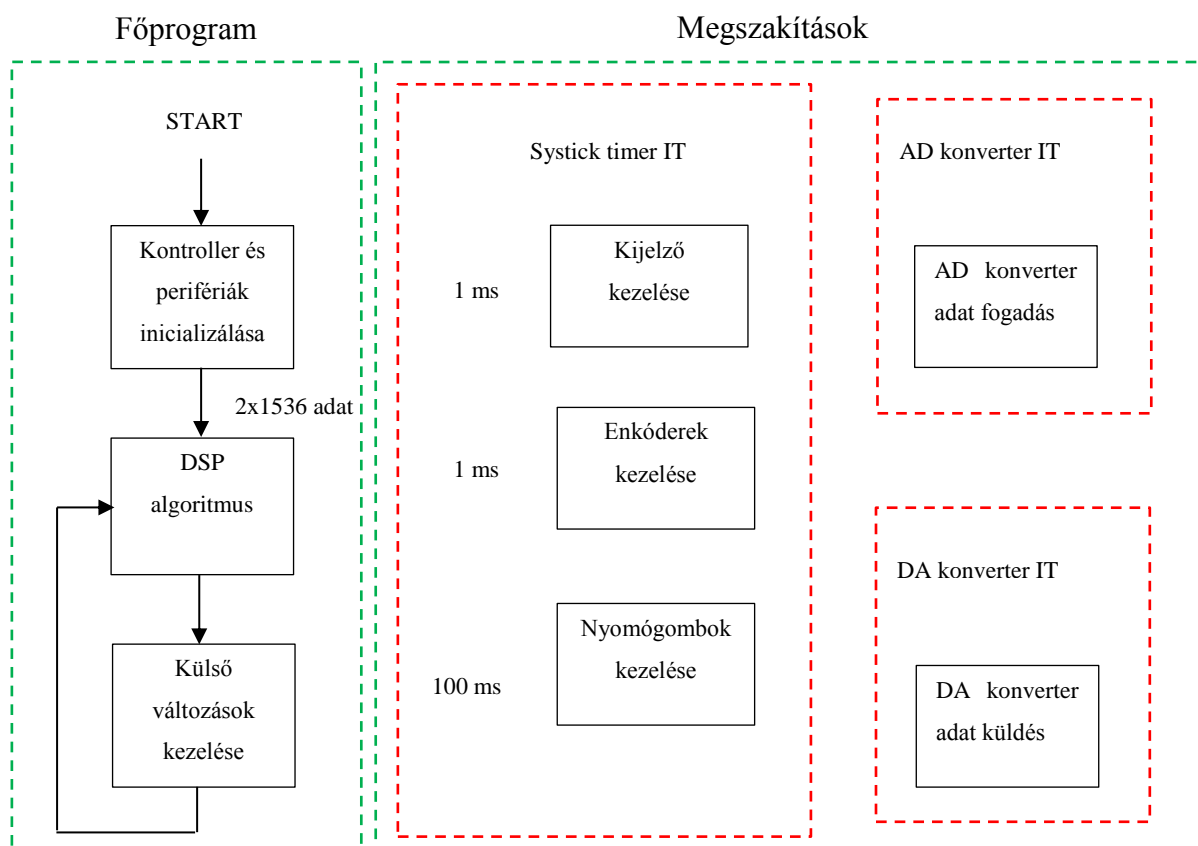
A 4-15. ábra bal oldalán látható az analóg jelek csatlakozási pontja, középen a programozásra és tápellátásra szolgáló USB csatlakozó.

5 Illesztő szoftverek

Ebben a fejezetben a hardverek működését kiszolgáló szoftverkomponenseket mutatom be. Ezek azért szükségesek, hogy elfedjék a legalsó réteget és így néhány függvény segítségével kezelhetjük a hardvert. Az alábbi komponensekről lesz szó:

- AD konverter inicializálás, olvasás,
- DA konverter inicializálás, írás,
- LCD kijelző inicializálás, írás,
- nyomógombokról történő beolvasás,
- enkóderekről történő beolvasás.

5.1 Általános bemutatás, rendszerterv



5-1. ábra Szoftver blokkdiagrammja

A kontroller inicializálása után elindul a főprogram. Itt inicializáljuk a hardvereket, és belépünk egy végtelen ciklusba, ahol a jelfeldolgozás és a külvilággal való kommunikáció lezajlik. Egyszerre 2×1536 mintát dolgoz fel a szoftver. A kétszeres szorzó a két csatornát jelenti. Az 1536-os blokkméret összefügg a mintavételi frekvenciával. Ez esetben 48 kHz. Ahhoz, hogy például 8 kHz mintavételi frekvenciára áttérjünk, 6-szoros decimálás szükséges. Decimáláshoz praktikus, ha az adattömb is osztható a decimálás mértékével. $1536/6=256$, így könnyen megvalósítható a decimálás. A tömbméret még így is kettő hatványa. Ez azért van, mert ilyen tömbön zero padding nélkül lehet FFT-t végezni.

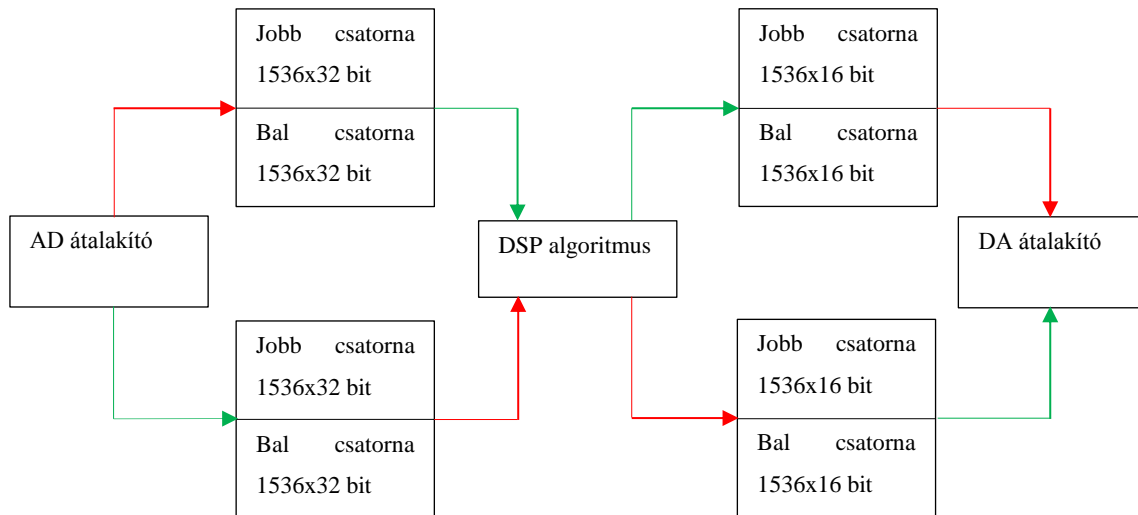
Miután feldolgoztunk egy adatsomagot, az egység kommunikál a külvilággal. Lekérdezi a nyomógombok, enkóderek állását, és ennek megfelelően módosítja a paramétereket, illetve ír az LCD kijelzőre, ha szükséges. Ezt követően dolgozzuk fel a következő adatsomagot.

Megszakítások

A nyomógombok, enkóderek és a kijelző kiszolgálása a SysTick timer időzítésének megfelelően történik. Ez a timer minden 1 ms-ként megszakítást generál. Az LCD kijelző és az enkóderek kezelése minden ilyen ciklusban megtörténik. Ez szükséges, mert a kijelzőre egy karakter kiírása 2 ms-t vesz igénybe, így elég karaktert ki tudunk írni. Az enkóderek gyors tekerésénél a periódusidő 4-6 ms, ezért itt is indokolt az ilyen gyakori lekérdezés. A nyomógombok kezelése minden századik megszakításban kerül sor. Ennél gyorsabban nem lehet a nyomógombokat működtetni, és itt azért is előnyös a ritka mintavételezés, mert a pergés jelenségeket kiküszöböljük. Az AD és DA konvertereknek saját megszakítási rutinnal rendelkeznek.

Kettős pufferelés, konverterek megszakításai

Az adatok mozgatását a két konverter megszakítása végzi el. Ezek a megszakítások eseményvezéreltek, és akkor következnek be, hogyha az I2S periféria puffere megtelt/kiürült. Az AD átalakító 24 bites adatait 32 bites regiszterbe tárolom. A DA átalakító 16 bites, ennek megfelelően az adattömb 16 bit széles. A két átalakító felbontása közötti különbségre magyarázat a bejövő jelek nagy dinamikatartománya. Az AD és a DA konverter is kettős pufferelésű: amíg az egyik puffer töltődik adatokkal (illetve a DA konverterre írjuk ki az adatokat), addig a másik puffert feldolgozzuk.



5-2. ábra Kettős puffrelés blokkdiagramja

Azt, hogy éppen melyik pufferből dolgozhat a DSP algoritmus, illetve a DA átalakító, azt az AD átalakító megszakításában állított flagekből lehet meghatározni.

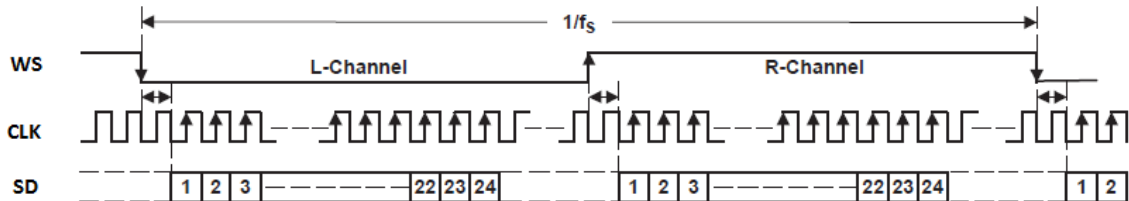
I2S (Integrated Interchip Sound) busz bemutatása

Mindkét átalakítón az adattovábbítás I2S buszon történik, ezért fontosnak tartom ennek a busznak a bemutatását. Integrált áramkörök közötti digitális sztereó audio jelátvitelre fejlesztették ki. Az adatok továbbítása sorosan történik. Létezik szimplex és duplex I2S busz is, az utóbbit általában codec-eken használják. A kommunikáció három vezetéken valósul meg (duplex átvitel esetén négyen):

- SD (Serial Data), vagy adatvezeték,
- ext. SD adatvezeték a másik irányba (csak duplex módban),
- CLK (Clock), órajelvezeték,
- WS (Word Select), adatkiválasztó vezeték

A WS határozza meg, hogy éppen melyik csatorna átvitele történik. Az, hogy WS egyes állapotai mit jelentenek, függ a használt módtól. Hasonlóan a WS jelhez, az hogy a CLK melyik élén érvényes az adatkimenet, a különböző módok függvénye. Egy csatorna által 16 vagy 32 bites csomagok küldhetők. Ha az adat rövidebb, - ez tipikusan a 24 bites átalakítók esetén áll fenn -, akkor a fennmaradó 8 bitet 0-ákkal egészítik ki. Azt, hogy az alsót vagy a felsőt, a különböző módok döntenek el. Tehát az, hogy milyen módban működik az átvitel, függ a szóhosszúságtól, az órajel polaritásától, a kiegészítő

bitek helyétől. Gyakorlatilag ezek bármilyen kombinációja előfordulhat, ezért viszonylag sok mód van. Mindkét konverter, amit használok Phillips standardot (egy-egy adatlapokban PCM standard-nak nevezik) használ. Ennek a módnak a jelalakjai az 5-5. ábrán látható. Érdekessége, hogy a WS jel váltása utáni első órajelciklusban még az előző csatorna utolsó bitje van a buszon.



5-3. ábra Egy tipikus I2S buszciklus (Phillips standard mód) [9]

5.2 AD konverter

A rendszert konfigurálni kell az adatfolyam megindítása előtt. Ennek menete:

Mikrokontroller I2S perifériájának konfigurálása:

```
PCM1804_I2S_Init();
```

- I/O engedélyezése, adatirányok beállítása
- 48 kHz mintavételi frekvencia
- Phillips standard I2S protokoll
- 24 bites adatok fogadása
- A kontroller a Master, és fogadni fogunk adatokat
- Az adat az órajel lefutó élénél érvényes
- MCLK órajelgenerálás bekapcsolása

A kapott adatokból a kontroller kiszámítja a szükséges a szükséges PLL regiszterek értékét, és $256f_s$ órajelet generál az MCLK lábra.

AD átalakító kivezetéseinek megfelelő állapotba állítása:

```
PCM1804_Config();
```

1. Az áramkör RESET kivezetésére alacsony szintet kell kapcsolni.
2. Master/ Slave beállítása, esetemben a konverter slave módban üzemel.

3. Adattípus kiválasztása: Az I2S protokoll több átviteli módot definiál, ezek közül meg kell adni azt, amelyiket a kontrollerben használjuk. Az AD konverter képes DSD módra. Ekkor a konverter adat kimenetére a Σ - Δ modulátor által előállított bitsorozat kerül. A konfigurációt az FMT1..0 kivezetések megfelelő szintbe állításával adhatjuk meg:

FMT1	FMT0	FORMAT	MASTER	SLAVE
Low	Low	PCM, left-justified, 24-bit	Yes	Yes
Low	High	PCM, I ² S, 24-bit	Yes	Yes
High	Low	PCM, standard, 24-bit	Yes	Yes
High	High	DSD	Yes	–

5-4. ábra Adattípus kiválasztása [9]

Én a 3. lehetőséget választottam, a mikrokontroller is ilyen módban működik.

4. Túlmintavételezési tényező beállítása: 128, 64, vagy 32-szeres lehet. Beállítani az OSR2..0 kivezetések megfelelő szintre kapcsolásával lehet. Az 5-5. ábra alapján kell beállítani. Én 128 szoros túlmintavételezést állítottam be, azért mert a szükséges 48 kHz-s mintavételi frekvencia csak ebben a beállításban érhető el.

OSR2	OSR1	OSR0	OVERSAMPLING RATIO	SYSTEM CLOCK RATE
Low	Low	Low	Single rate ($\times 128 f_s$)	Automatically detected
Low	Low	High	Dual rate ($\times 64 f_s$)	Automatically detected
Low	High	Low	Quad rate ($\times 32 f_s$) ⁽¹⁾	Automatically detected
Low	High	High	Reserved	–
High	Low	Low	Reserved	–
High	Low	High	Reserved	–
High	High	Low	Reserved	–
High	High	High	Reserved	–

5-5. ábra Túlmintavételezési tényező meghatározása, ha a konverter Slave módban van [9]

OVERSAMPLING RATIO	SAMPLING FREQUENCY (kHz)	SYSTEM CLOCK FREQUENCY (MHz)					
		128 f_s	192 f_s ⁽¹⁾	256 f_s	384 f_s	512 f_s	768 f_s
Single rate ⁽²⁾	32	–	–	8.192	12.288	16.384	24.576
	44.1	–	–	11.2896	16.9344	22.5792	33.8688
	48	–	–	12.288	18.432	24.576	36.864
Dual rate ⁽³⁾	88.2	–	–	22.5792	33.8688	–	–
	96	–	–	24.576	36.864	–	–
Quad rate ⁽⁴⁾	176.4	22.5792	33.8688	–	–	–	–
	192	24.576	36.864	–	–	–	–
DSD mode ⁽³⁾	44.1	–	–	11.2896	16.9344	–	–

5-6. ábra Túlmintavételezési tényező és mintavételi frekvencia függvényében a rendszerórajel meghatározása [9]

Mintavételi frekvenciát nem lehet megadni, ezt a WS jel frekvenciájából meghatározza a konverter.

5. Felül áteresztő szűrő bekapcsolása, ha szükséges. A bypass láb „1” értéke esetén nincs, „0” értéke esetén van felüláteresztő szűrő.
6. RESET magas szintbe állítása

Ezek után a megszakítást konfiguráltam be. Ahogy megtelik adattal az I2S modul, megszakítás generálódik. A kontroller I2S perifériája csak 16 bites adatok fogadására alkalmas, tehát egy 24 bites adat két megszakítást generál. A megszakítási rutin eldönti, hogy milyen adat érkezett: Jobb vagy bal csatorna, alsó vagy felső fele a szónak. Ezek figyelembevételével elhelyezi az adatot a memóriában, majd beállítja a megfelelő flageket. Ezek a következők:

- melyik puffert töltődik éppen,
- beérkezett csomag alsó vagy felső része a teljes adatnak,
- a puffer hányadik eleménél tartunk.

Ezt követően a főprogramba engedélyezni kell az I2S modult. Ennek hatására a mikrokontroller kiadja a szükséges órajeleket, az AD konverter pedig szolgáltatja az adatfolyamot.

5.3 DA konverter

Az AD átalakítóhoz hasonlóan itt is szükség van konfigurálásra működés előtt:

Mikrokontroller I2S perifériájának konfigurálása:

```
CS43L22_I2S_Init();
```

- I/O engedélyezése, adatirányok beállítása,
- 48 kHz mintavételi frekvencia,
- Phillips I2S protokoll,
- 16 bit hosszú adatok,
- a kontroller a Master, és fogadni fogunk adatokat,
- Az adat az órajel lefutó élénél érvényes,
- MCLK órajelgenerálás bekapcsolása.

Ezt követően a konvertert állítom be. Itt nem az egyes lábak megfelelő szintbe állításával lehet felkonfigurálni a konvertert, hanem I2C buszon keresztül a megfelelő regiszterek értékének beállításával. A konverter a 0x94-es címen van.

```
CS43L22_Config();
```

1. DA talakító RESET-be állítása
2. Master/Slave beállítása, itt szintén slavet állítottam be
3. DA konverter kikapcsolása (Regiszter: 0x02, Érték: 0x01)
4. Fülhallgató mód bekapcsolása, hangszóró mód kikapcsolása (regiszter: 0x04, érték: 0xAF)
5. Automatikus órajel detektálás bekapcsolása (regiszter: 0x05, érték: 0x81)
6. I2S protokoll beállítása: Phillips standard (regiszter: 0x06, érték: 0x06)
7. DA átalakító bekapcsolása (regiszter: 0x02, érték: 0x9E)

A működéshez nem szükséges mást beállítani. Az átalakító a bemenetére érkező adatfolyamból „kitalálja” a paramétereket.

Ezután a megszakítást állítottam be. A megszakítási rutin megvizsgálja az AD átalakító által állított flageket és az adatokból meghatározza hogy melyik puffer hányadik adatát kell elküldenie.

Hasonlóan az AD konverterhez, itt is a főprogramban helyeztem el az I2S modul engedélyező parancsát. Ügyeltem arra, hogy legyen késleltetés a két modul engedélyezése között, így a két megszakítás sem egy időben fog bekövetkezni.

5.4 Nyomógombok

Három függvény segítségével kezelhetjük a nyomógombokat:

```
BUTTON_HANDLE()
```

Ezt a függvényt a Systick timer megszakítási rutinjában helyeztem el. A függvény elmenti az előző értékét a nyomógomboknak, és lekérdezi az újat. Nem szükséges inicializálni, mert tesztelés közben változtatjuk az I/O portok adatirányát, ennek köszönhetően mindig amikor meghívódik ez a függvény, egyben inicializáljuk is a hardvert.

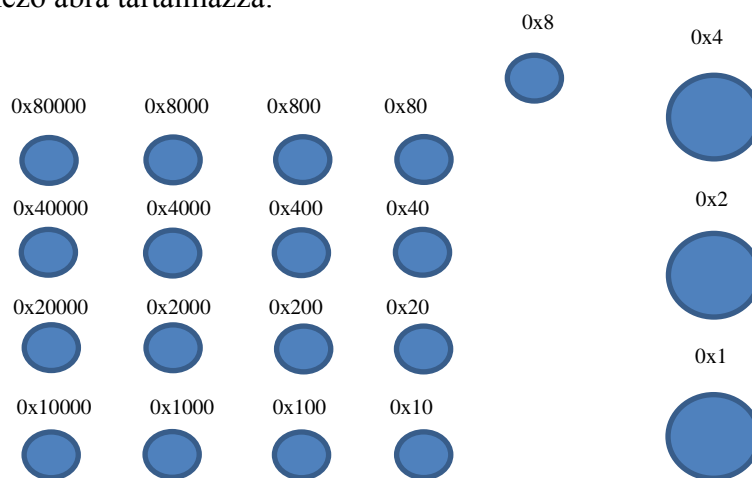
```
uint32_t BUTTON_GET_STATE(void)
```

Ennek a függvénynek a visszatérési értéke annak a nyomógombnak a száma, amelyen lefutó élt érzékelt. A függvény megvizsgálja, hogy melyik regiszter nagyobb: az előző, vagy az aktuális. Ha az előző érték, az azt jelenti, hogy valamelyik gombot éppen megnyomták. Ezután meg kell vizsgálni, hogy a regiszterben – binárisan ábrázolva – csak egy darab egyes van e. Ha ez teljesül, akkor csak egy gombot nyomtak meg. Ha nem, akkor egyszerre többet, amit nem tudunk megkülönböztetni a hardver sajátosságai miatt. Ha csak egy gomb megnyomását érzékelt, a függvény visszatér az éppen megnyomott gomb értékével, minden más esetben a visszatérési érték 0.

```
void BUTTON_SET_HANDLED()
```

A gombok lekérdezése 100 ms-ként történik meg, kezelésük körülbelül 30 ms-ként (ennyi idő alatt telik meg egy adat puffer). Így egy gombnyomást háromszor kezelnének le. Ezért a gombnyomás lekezelése után meg kell hívni ezt a függvényt. Ez frissíti a nyomógombok előző és aktuális értékét, nem kezeljük le többször ugyanazt a megnyomást.

Azt hogy hogy az egyes értékek fizikailag melyik nyomógombhoz tartoznak, a következő ábra tartalmazza:



5-7. ábra Nyomógombokhoz tartozó regiszter értékek

5.5 Enkóderek

Négy függvény szolgál az enkóderek kezelésére.

```
ENCODER_INIT()
```

Ezt a függvényt a program elején kell meghívni. Inicializálja az enkódereket.

```
ENCODER_HANDLE()
```

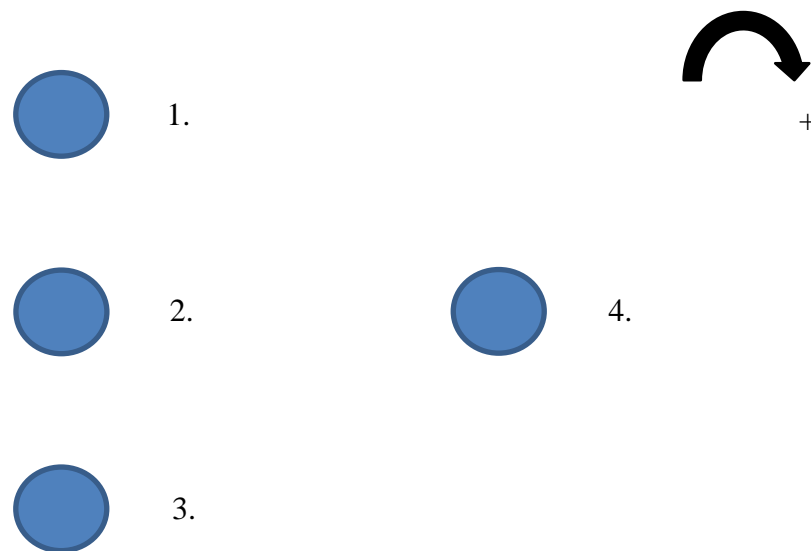
Ezen függvényt a SysTick timer megszakításban kell meghívni. Az összes enkódert megvizsgálja és megnézi történt-e elmozdulás valamelyik irányban. Ha igen, akkor növeli vagy csökkenti az érintett enkóder regiszterét, annak megfelelően, hogy melyik irányba történt. Az elmozdulás detektálás úgy működik, hogy az enkóderek egyik érintkezőjének aktuális és előző értékét mindig elmenti a függvény egy-egy regiszterbe. Egyszeres módban használom az enkódereket, ezért csak a felfutó éleket figyeli a függvény. Ha detektáltunk egy felfutó élet (az előző értéket tároló regiszter 0, az aktuálisat tároló pedig 1) akkor megvizsgáljuk a másik láb állapotát. Ha ez 0, akkor balra forgatták az enkódert, ha pedig 1 akkor jobbra. Ennek megfelelően első esetben levon egyet, második esetben pedig hozzáad egyet az érintett enkóder regiszteréhez.

```
int16_t ENCODER_GET_OFFSET(uint8_t Encoder_Name)
```

A paraméterként megkapott számú enkóder értékével tér vissza. A paraméter 1..4 közötti egész szám. Ha nem ilyen adunk meg a visszatérési érték 0.

```
ENCODER_CLEAR_OFFSET(uint8_t Encoder_Name)
```

A paraméterbe megkapott enkóder regiszterét törli ki. A paraméterre ugyanazok a szabályok vonatkoznak, mint az előző függvény esetében.



5-8. ábra Enkóderek párosítása a szoftverhez, illetve a forgatás pozitív iránya

5.6 Kijelző

Öt függvény szolgál az LCD kijelző kezelésére.

```
LCD_INIT_GPIO()
```

Inicializálja az LCD-hez I/O kivezetéseket. A program elején a többi inicializáló függvény között kell meghívni.

```
LCD_WRITE_STRING()
```

A Systick timer megszakításában kell elhelyezni ezt a függvényt. A kijelzőre küldött adatokat ez írja ki a kijelzőre.

```
LCD_INIT()
```

Ezt a függvényt szintén a főprogramban kell elhelyezni. Itt már történik kommunikáció a kijelzővel, ez a függvény állítja be a főbb paramétereket:

1. 4 bites üzemmód
2. 4 soros mód
3. kurzor kikapcsolása
4. inkrementális mód
5. kijelző törlése

Ezt követően a kijelző készen áll a használatra.

```
uint8_t LCD_Send_Data_Stream(uint8_t address, uint8_t *data)
```

Ezzel a függvénnyel egy maximum 20 elem hosszú, nullával lezárt karaktersorozatot írhatunk ki a kijelzőre, egy adott címtől kezdve. A függvény visszatérési értéke 1, ha fogadta az adatot, 0 ha az LCD rutin foglalt (egy előző kiírás még folyamatban van).

```
uint8_t LCD_Send_Instruction_Stream(uint8_t *data)
```

Itt utasítássorozatot küldhetünk a kijelzőnek, például kijelzőtörlés, sorshiftelés, kurzor bekapcsolása stb. Visszatérési értékének jelentése megegyezik az előző függvényével.

6 SSB jel I-Q demodulációja

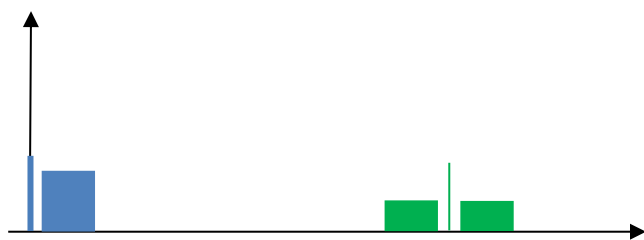
Ebben a fejezetben az amplitúdómodulált jelek néhány fajtáját mutatom be és egy egyszerű SSB/SC demodulátort tervezek meg az elkészített hardverre [11].

6.1 Elméleti alapok

AM-DSB (Amplitudo Modulation – Double SideBand)

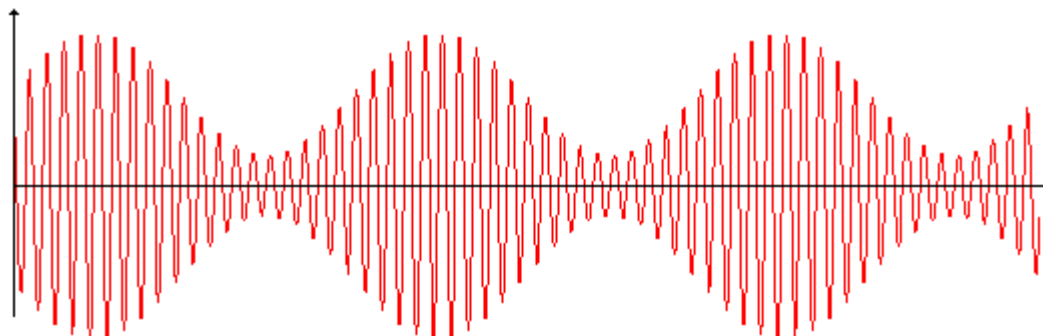
Adott egy alapsávi, sávkorlátos, DC komponens nem tartalmazó jel $x(t)$. Ez az a jel, amit továbbítani szeretnénk. Ehhez hozzáadunk egy DC komponens (U) , így kerül a modulátor bemenetére. A másik bemenetre pedig egy koszinusz függvényt kapcsolunk: $\cos(2\pi f_m t)$. A modulátor összeszorozza a bemenetére kerülő jeleket, melynek következtében előáll az amplitúdó modulált jel:

$$s(t) = (x(t) + U) \cos(2\pi f_m t)$$



6-1. ábra AM-DSB jel spektruma

Kék színű az alapsávi jel, zöld pedig a modulált jel. Látható a két oldalsáv, illetve az egyenkomponensből előálló vivő. A két oldalsáv gyakorlatilag ugyan azt az információt tartalmazza, a vivő pedig viszonyítási alapként szolgál demodulálásnál.



6-2. ábra Egy AM-DSB jel időfüggvénye

AM-DSB modulációs mód jellemzése:

- sok helyet foglal a rádiófrekvenciás spektrumban (a két oldalsáv, amely ugyanazt a jelet tartalmazza),
- rossz hatásfok: a másik oldalsávot is ki kell sugározni, de ez nem hordoz többlet információt,
- egyszerű demodulálhatóság (burkoló demodulátorral megvalósítható).

Egyszerű demodulálhatóságának köszönhetően középhullámú broadcast sávban használják (például Kossuth Rádió, és a BBC is ilyet használ középhullámon).

AM-SSB (Amplitudo Modulation Single Side Band)

Ha az egyik oldalsávot elnyomjuk egy alul- vagy felüláteresztő szűrő segítségével, akkor a jel információtartalmából nem veszít semmit, ellenben a hatásfok növekszik, a sávszélesség csökken. Annak alapján, hogy melyik oldalsávot hagyjuk meg, beszélhetünk AM-LSB (Low Side Band: az alsó oldalsávot tartjuk meg), illetve AM-USB (Upper Side Band: a felső oldalsávot tartjuk meg) modulációról. Ezt a módot használják a televízió képtartalom átvitelére. Demodulálásához bonyolultabb (szorzó) demodulátor szükséges, de a felhasznált sávszélesség a felére csökken.

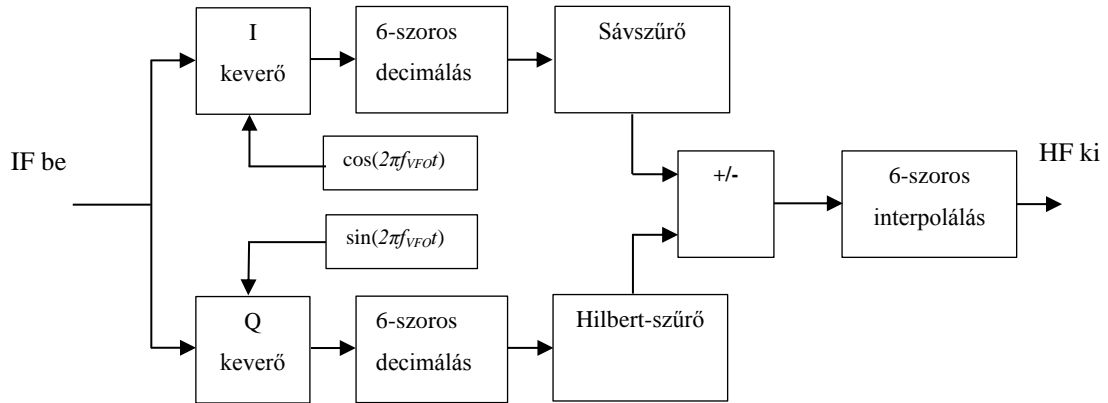
AM-SSB/SC (Amplitudo Modulation – Single SideBand, Supressed Carrier)

Megtehetjük azt is, hogy a vivőt sem sugározzuk ki. Ez a modulációs eljárás rendelkezik a legjobb hatásfokkal és legkisebb sávszélességgel. Az így előállított jel felfogható úgy is, mintha az alapsávi jelet eltoljuk a frekvenciatengelyen. Azzal hogy a vivőt is elnyomtuk, elveszítjük a viszonyítási alapot: azt hogy az alapsávi jel egyenszintje milyen frekvenciára transzformálódik. Demodulálása során az elnyomott vivőt nekünk kell előállítani a rádióban. Ha az előállított és az eredeti vivő különbözik, akkor a demodulált jel torzulni fog, például egy beszédet magasabbnak vagy mélyebbnek fogjuk hallani.

Jó hatásfoka és kis sávszélessége miatt ez az egy AM mód használható a rádióamatőr sávokba. Így ennek a modulációs fajtának a demoduláló algoritmusát terveztem meg.

6.2 A választott demodulátor elrendezés bemutatása

A választott eljárást I-Q, vagy Hartley demodulátornak nevezi a szakirodalom [2],[11], ennek blokkdiagramja a 6-3. ábrán látható.



6-3. ábra Demodulátor blokkvázlata

Az eljárás mindkét oldalsáv vételére alkalmas. A bemenetre 12 kHz-es középfrekvenciára kevert jelet kell kapcsolni, kimeneten pedig megjelenik a választott oldalsáv alapsávban. Ennek az elrendezésnek az előnye, hogy nem szükséges bonyolult és drága kvarcszűrővel analóg kiszűrni a nemkívánatos oldalsávot, mert azt a demodulátor kiszűri. Arról azonban gondoskodni kell, hogy a 12 kHz-es középfrekvencián átlapolódás még ne következzen be. Működését egy példán keresztül mutatom be.

A tesztjel legyen egy 10 kHz-es és egy 13 kHz-es szinuszjel. Ha egyszerű keverést alkalmaznánk, akkor a kimeneten megjelenne 1 kHz (a 13 kHz-es jel alapsávi megfelelője) és 2 kHz (a 10 kHz-es jel alapsávi megfelelője). Az első jel az LSB sávban van, a második pedig az USB-ben. A cél, hogy demodulálás után csak az egyik szinusz jusson a kimenetre.

A bejövő jelet megszorozom egy 12 kHz-es szinusz és egy ugyan ilyen frekvenciájú koszinusz jellel.

$$U_{be1}(t) = \sin(2\pi f_{be1} t)$$

$$U_{be2}(t) = \sin(2\pi f_{be2} t)$$

$$U_{VFO\sin}(t) = \sin(2\pi f_{VFO} t)$$

$$U_{VFO\cos}(t) = \cos(2\pi f_{VFO} t)$$

Ahol $f_{be1}=10$ kHz, $f_{be2}=13$ kHz, $f_{VFO}=12$ kHz.

A két próbafrekvencia szándékosan lett aszimmetrikusan elhelyezve a középfrekvenciára, mert így lekeverés után is meg lehet őket különböztetni.

Az első keverés eredménye:

$$\begin{aligned} U_{Qsignal1} &= U(t)_{VFO} \cos(U(t)_{be1} + U(t)_{be2}) \\ &= \frac{1}{2} [-\cos(2\pi(f_{VFO} + f_{be1})t) - \cos(2\pi(f_{VFO} + f_{be2})t) \\ &\quad + \cos(2\pi(f_{VFO} - f_{be2})t) + \cos(2\pi(f_{VFO} - f_{be1})t)] \end{aligned}$$

$$\begin{aligned} U_{Isignal1} &= U(t)_{VFO} \sin(U(t)_{be1} + U(t)_{be2}) \\ &= \frac{1}{2} [\sin(2\pi(f_{VFO} + f_{be1})t) + \sin(2\pi(f_{VFO} + f_{be2})t) \\ &\quad + \sin(2\pi(f_{VFO} - f_{be2})t) + \sin(2\pi(f_{VFO} - f_{be1})t)] \end{aligned}$$

Látható hogy a keverés eredményeként mindkét csatornán megjelenik a próbaspektrum mindkét komponense alapsávban, illetve az összegfrekvenciás komponensek. Az utóbbiakra nincs szükségünk, zavarják a jelet, ezért ezeket ki kell szűrni. Mivel a hasznos jel így már 0-3 kHz-ig terjed, érdemes áttérni alacsonyabb mintavételi frekvenciára. Erre egy hatszoros decimáló szűrőt készítettem. Ez kiszűri az összegfrekvenciás komponenseket, és lecsökkenti a mintavételi frekvenciát 8 kHz-re. Ez több szempontból előnyös:

- A kisebb mintavételi frekvencián az 1536 hosszú adattömb lecsökken 256-hosszúra, így hatszor hosszabb szűrőket tudunk futtatni ugyan annyi processzoridó alatt.

- Ha FFT-t szeretnénk végezni, akkor így a 8 kHz-es mintavételi frekvenciához viszonyítunk, és sokkal jobb felbontást érhetünk el.

Az aluláteresztő szűrés és decimálás után a két csatornán maradt jel a következő lesz:

$$U_{Qsignal2} = \frac{1}{2} [\cos(2\pi(f_{VFO} - f_{be2})t) + \cos(2\pi(f_{VFO} - f_{be1})t)]$$

$$U_{Isignal2} = \frac{1}{2} [\sin(2\pi(f_{VFO} - f_{be2})t) + \sin(2\pi(f_{VFO} - f_{be1})t)]$$

Észrevehetjük, hogy $U_{I\text{signal}2}$ -ben az első szinusz argumentumában szereplő $f_{VFO}-f_{be2}$ értéke mindig negatív, f_{be2} a felső oldalsávból való, így mindig nagyobb, mint f_{VFO} . Tekintettel arra, hogy a szinusz páratlan függvény, megtehetjük a következőt:

$$\sin(2\pi(f_{VFO} - f_{be2})t) = -\sin(2\pi(f_{be2} - f_{VFO})t)$$

Így csak az alakja változott meg az összefüggésnek, de ez a változtatás kulcsfontosságú. Ezt a változtatást megtehetjük a másik csatorna koszinusz függvényével is (amelyeknek az argumentumában szerepel f_{be2}), mivel azonban a koszinusz páros függvény, az előjel itt nem változik.

$$U_{Q\text{signal}2} = \frac{1}{2} [\cos(2\pi(f_{be2} - f_{VFO})t) + \cos(2\pi(f_{VFO} - f_{be1})t)]$$

$$U_{I\text{signal}2} = \frac{1}{2} [-\sin(2\pi(f_{be2} - f_{VFO})t) + \sin(2\pi(f_{VFO} - f_{be1})t)]$$

Az I csatorna jelét egy sávszűrővel szűröm. Ennek kettős szerepe van:

- Ugyanakkora késleltetést biztosít, mint a másik csatorna Hilbert szűrője
- Amplitudókarakterisztikája megegyezik a Hilbert-szűrő karakterisztikájával, így a két csatorna között nem lép fel kiegyenlítetlenség.

A Q csatorna jelét egy Hilbert szűrővel szűröm. Ennek sajátossága, hogy a bemenetére kerülő jel fázisát -90° -kal forgatja el. Ennek hatására a jel a következő lesz:

$$U_{Q\text{signal}3} = \frac{1}{2} [\sin(2\pi(f_{be2} - f_{VFO})t) + \sin(2\pi(f_{VFO} - f_{be1})t)]$$

Így előállítottuk a két szükséges jelet. Ezeknek az összege az alsó különbségük pedig a felső oldalsávot adják. Emlékeztetőkép az I csatorna jele:

$$U_{I\text{signal}3} = \frac{1}{2} [-\sin(2\pi(f_{be2} - f_{VFO})t) + \sin(2\pi(f_{VFO} - f_{be1})t)]$$

Így az alsó és a felső oldalsáv a következőképpen áll elő:

$$U_{USB} = U_{I\text{signal}3} - U_{Q\text{signal}3} = -\sin(2\pi(f_{be2} - f_{VFO})t)$$

$$U_{LSB} = U_{I\text{signal}3} + U_{Q\text{signal}3} = \sin(2\pi(f_{VFO} - f_{be1})t)$$

Itt látható, hogy mennyire hasznos volt az előjel kiemelése a szinusz argumentumából, ugyanis itt nem látszódna a kioltás.

Ezt követően a jelet interpolálom, azért hogy a 48 kHz-en működő DA konverter számára feldolgozható legyen.

Az konstrukción látszik, hogy a rendszer nagyon érzékeny a két csatorna kiegyenlítetttségére, mind fázis mind amplitudó szempontjából. Itt ez nem volt különösen nagy probléma, mert az összes műveletet a DSP-n belül végeztem el, és a lebegőpontos számábrázolásnak köszönhetően minden számítás nagyon pontos. Egyes megvalósításoknál az I-Q lekeverést nem a processzorba, hanem analóg keverőkkel valósítják meg. Így nem garantált, hogy a két keverőjel fáziskülönbsége pontosan 90° , illetve az esetleges analóg erősítések és szűrések során a két csatorna amplitudójában is lehet különbség. Ezek a kiegyenlítetlenségek nem tökéletes tükörelnyomáshoz vezetnek. Az ilyen hibákat különböző kompenzáló algoritmusok segítségével szokták csökkenteni.

6.3 Szűrők méretezése

Decimáló szűrő

A decimáló szűrő azért szükséges, hogy kiszűrje a spektrumból azokat a komponenseket, amelyek a mintavételi frekvencia csökkenése után átlapolódást eredményeznének, tehát gyakorlatilag egyszerű aluláteresztő szűrő.

A decimálást két lépcsőben végzem el. Ez jobb megoldás az egy lépcsőben történő decimáláshoz képest abból a szempontból, hogy a két szűrő együttesen is kisebb, mint az egy lépcsőben megtervezett szűrő. Az összes szűrőt a Matlab FDA toolboxában terveztem. Itt gyorsan tervezhető egy szűrő, és azonnal kijelzi az elkészült szűrő karakterisztikáját, így az esetleges változtatások könnyen elvégezhetők.

Itt nagyon hasznos, hogy az adatsor osztható a decimálási faktoral, hiszen ha nem lenne az, akkor nagyon nehezen lehetne elvégezni a decimálást (mindig az előző mintasorból meg kellene tartani annyit, hogy kiteljen az oszthatóság).

Mindkét fokozatot FIR szűrővel valósítottam meg. Annak következtében, hogy itt nincs visszacsatolás, megtehetjük azt, hogy felezésnél csak minden második, harmadolásnál csak minden harmadik mintát számoljuk ki, hiszen a többit úgyis eldobjuk. Ezáltal jelentősen lecsökken az algoritmus futási ideje. Ezt IIR esetben a visszacsatolások miatt nem tehetjük meg

1. fokozat: felező decimáló szűrő

Specifikáció (lásd 6-4. ábra):

Áteresztő tartomány: 0-4 kHz

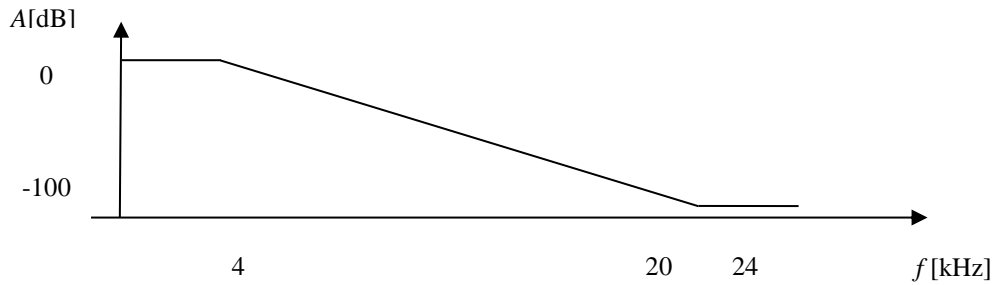
Zárótartomány: 20-24kHz

Elnyomás a zárótartományban: 100dB

Áteresztő tartománybeli ingadozás: 0.5dB

Mintavételi frekvencia: 48 kHz-ről 24 kHz-re csökken.

A követelményeket egy 15 tapes FIR szűrő teljesítette.



6-4. ábra Felező decimáló szűrő

A szűrő specifikációja nagyon „laza”, nagy az áteresztő tartomány. Ebben a tartományban a jelek egymásra lapolódnak a decimálás után. Ez nem probléma, hiszen abba a sávba történik átlapolódás, amit majd a következő szűrő eltávolít.

2. fokozat: harmadoló decimáló szűrő

Specifikáció (lásd 6-5. ábra):

Áteresztő tartomány: 0-3 kHz

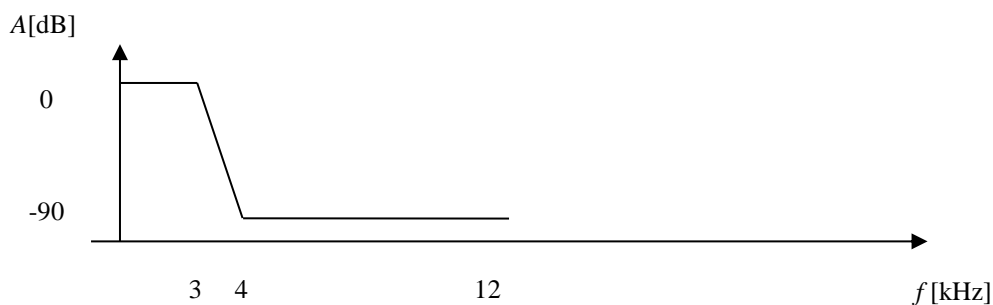
Zárótartomány: 4-12 kHz

Elnyomás a zárótartományban: 90dB

Áteresztő tartománybeli ingadozás: 0.5dB

Mintavételi frekvencia: 24 kHz-ről 8 kHz-re csökken

A követelményeket egy 131 tapes FIR szűrő teljesítette.



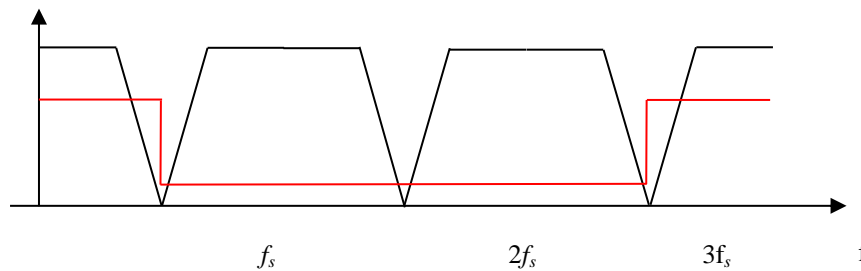
6-5. ábra Harmadoló decimáló szűrő

A két szűrő kaszkád kapcsolásából készül a hatszoros decimáló szűrő. Kipróbáltam azt is, hogyha egy szűrővel kellene megvalósítani a hatszoros decimálást. Akkor 200-nál

több tapre lett volna szükség a 90 dB-es elnyomás megvalósításához. Ennél a kétlépéses decimáló szűrő konstrukció kevesebbet használ.

Interpoláló szűrő

Egy mintavételezett jel spektruma perodikus, a periódus pedig a mintavételi frekvencia. Az interpoláló szűrő feladata az, hogy eltávolítsa ezeket a periódusokat, és csak egyet hagyjon meg, az alapperiódust, ahogyan a 6-6. ábrán látszik.



6-6. ábra Mintavételezett jel spektruma, interpoláló szűrő

Az interpolálást hasonlóan a decimáláshoz két lépcsőben végeztem el, itt azonban először kétszeresen majd háromszorosan interpolálok. Interpolálás menete: A mintasorba beszúrunk periodikusan $N-1$ darab 0-át (interpolálási faktor), és ezt vezetjük át a szűrőn. Így azonban sok olyan műveletet végzünk el feleslegesen, amelynek tudjuk a végeredményét előre: 0-val való szorzások eredménye 0. Ezért ha olyan szűrőt alkalmazunk, aminek a fokszáma az interpolálási faktor többszöröse, akkor ez a szűrő szétbontható N darab kis szűrőre. Ezeket a kis szűrőket egymásután futtatva megkapjuk az interpolált értékeket. Ezt nevezik polifázisú FIR szűrőnek.

1. fokozat: háromszorozó interpoláló szűrő

Specifikáció (lásd 6-7. ábra):

Áteresztő tartomány: 0-3 kHz

Zárótartomány: 4-12 kHz

Elnyomás a zárótartományban: 95dB

Áteresztő tartománybeli ingadozás: 0.5dB

Mintavételi frekvencia: 8 kHz-ről 24 kHz-re növekszik

A követelményeket egy 137 tapes FIR szűrő teljesítette.

A szűrő 137 tapes, így 138 együtthatója van. Ez pont osztható 3-mal, így a szűrő polifázisúvá alakítható.



6-7. ábra Háromszorozó interpoláló szűrő

2. fokozat: kétszerező interpoláló szűrő

Specifikáció (lásd 6-8. ábra):

Áteresztő tartomány: 0-4 kHz

Zárótartomány: 12-24 kHz

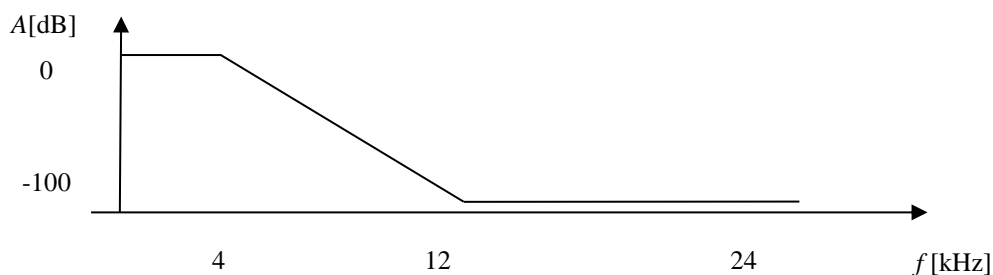
Elyomás a zárótartományban: 100 dB

Áteresztő tartománybeli ingadozás: 0.5 dB

Mintavételi frekvencia: 24 kHz-ről 48 kHz-re növekszik

A követelményeket egy 29 tapes FIR szűrő teljesítette.

Ez 30 együtthatójú, osztható 2-vel, tehát polifázisúvá alakítható.



6-8. ábra Kétszerező interpoláló szűrő

Hilbert szűrő és sávszűrő:

A Hilbert szűrő csak felüláteresztő vagy sávszűrő lehet. Ez megnehezítette a tervezést, mert nekem pont egy aluláteresztő jellegű szűrőre volt szükségem. Így egy olyan

sávszűrőt terveztem, ami lehetőleg minél kisebb frekvencián már átereszt. Megpróbáltam az eredeti 48000 Hz-es mintavételi frekvencián megvalósítani, hogy 300 Hz-en már ugyanaz legyen az átvitele a Hilbert-szűrőnek, mint a párjának a sávszűrőnek, de a két karakterisztika között több dB eltérés volt (azért pont 300 Hz, mert a rádióamatőr sávokon használt SSB jel – alapsávba lekeverve - 300 Hz-től 3 kHz-ig terjed). Ez megengedhetetlen, mert így a tükörelnyomás nagyon kevés lett volna (két csatorna közötti amplitudókiegyenlítetlenség miatt). Ez az eltérés az együtthatók számának drasztikus növelése esetén (néhány ezer tap) sem lett kisebb. Ennek az volt az oka, hogy a Hilbert-szűrő felüláteresztő jellegű, és relatív frekvenciába a 300 Hz nagyon kevés ekkora mintavételi frekvencián. Ezért is volt szükség a decimálásra. A decimálás következtében a 300 Hz relatív frekvenciába jóval nagyobb, mint az előző mintavételi frekvencián, sőt, így akár olyan szűrőt is lehetett tervezni, ami kisebb frekvenciákon is már átereszt.

A két szűrőnek ugyan az a specifikációja, csak ez egyik Hilbert típusú, a másik pedig egy egyszerű sávszűrő:

Áteresztő tartomány: 120 Hz – 3.8 kHz

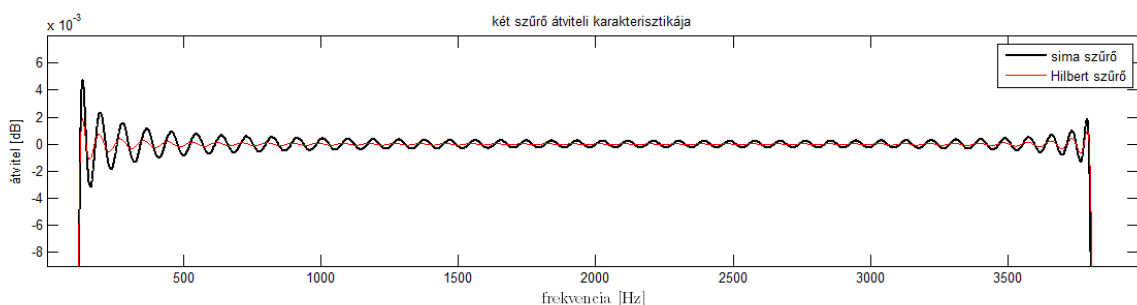
Zárótartomány: -

Áteresztő tartománybeli ingadozás: 0.005 dB

Mintavételi frekvencia: 8 kHz

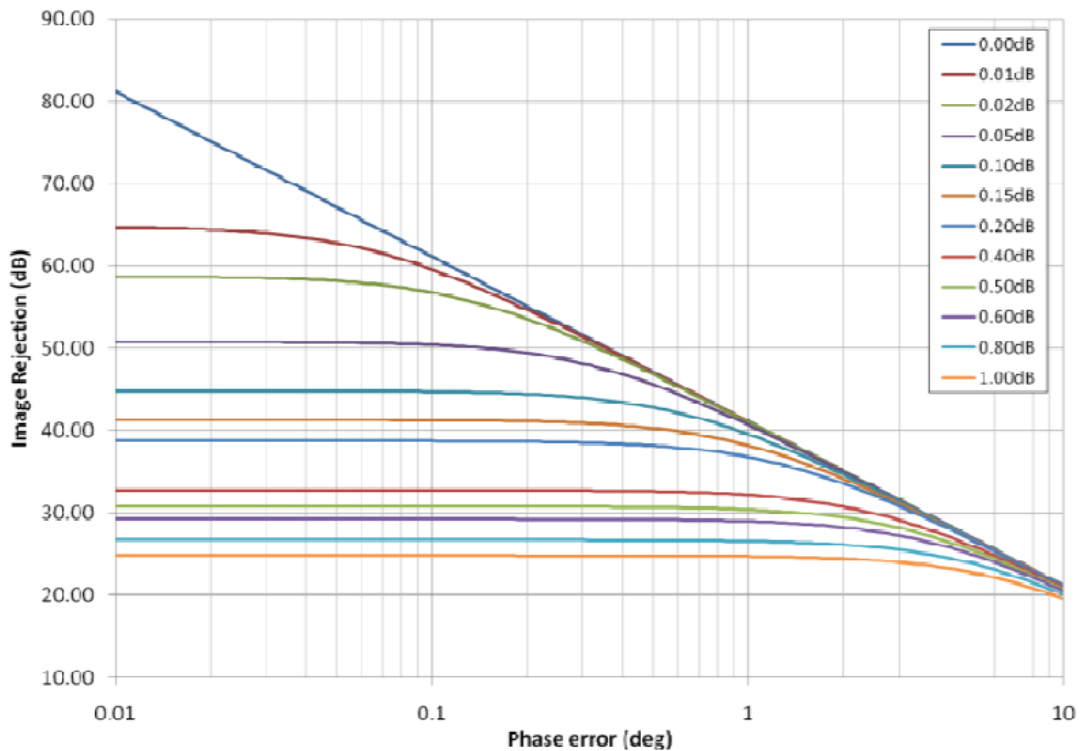
A követelményeket egy-egy 170 tapos FIR szűrő teljesítette.

Ezeknél a szűrőknél nem a teljes átviteli karakterisztika volt a lényeges, hanem az áteresztőtartományban mennyire egyforma az átvitelük.



6-9. ábra Hilbert és egyszerű sávszűrő átviteli karakterisztikája

A 6-9. ábrán látható hogy az átvitelek különbsége sehol sem nagyobb 0.003 dB-nél, így az amplitudó kiegyenlítetlenségből származó hiba nagyon kicsi lesz.



6-10. ábra Demodulátor tükörelnyomása a fázis illetve erősítéshiba függvényében [12]

A 6-10. ábrán látható, hogy hogyan változik a tükörelnyomás a fázis- és amplitúdóhiba függvényében. A fázishibát ebben a konstrukcióban a DSP-ben generált szinusz-, illetve koszinuszfüggvény hibája adja. Ez egy 256 hosszú tömbből felhozott adatokból negyedfokú interpolációval határozza meg a szükséges értéket. A 256 érték $-\frac{\pi}{2} \dots \frac{\pi}{2}$ tartományban tartalmazza a függvényt. Ha csak ezeket az értékeket használnánk, akkor is 0.16° ($180^\circ / (2 \cdot 256)$) lenne a fázishiba. Feltételezem, hogy a negyedfokú interpoláció egy nagyságrendet biztosan javít a pontosságon. Az amplitúdóhiba 6-9. ábra alapján maximum 0.003 dB. Tehát elméletileg a tükörelnyomás 70 dB vagy több lehet a megtervezett algoritmus használatával.

7 Mérési eredmények

Az áramkörrel kapcsolatos méréseket két részre bontottam:

- Hardver mérése

- átvitel mérése,
- bemeneti zaj mérése.

- Megvalósított algoritmus mérése

- működés ellenőrzése releváns teszttel,
- tükörelnyomás mérése.

Mérés során felhasznált eszközök:

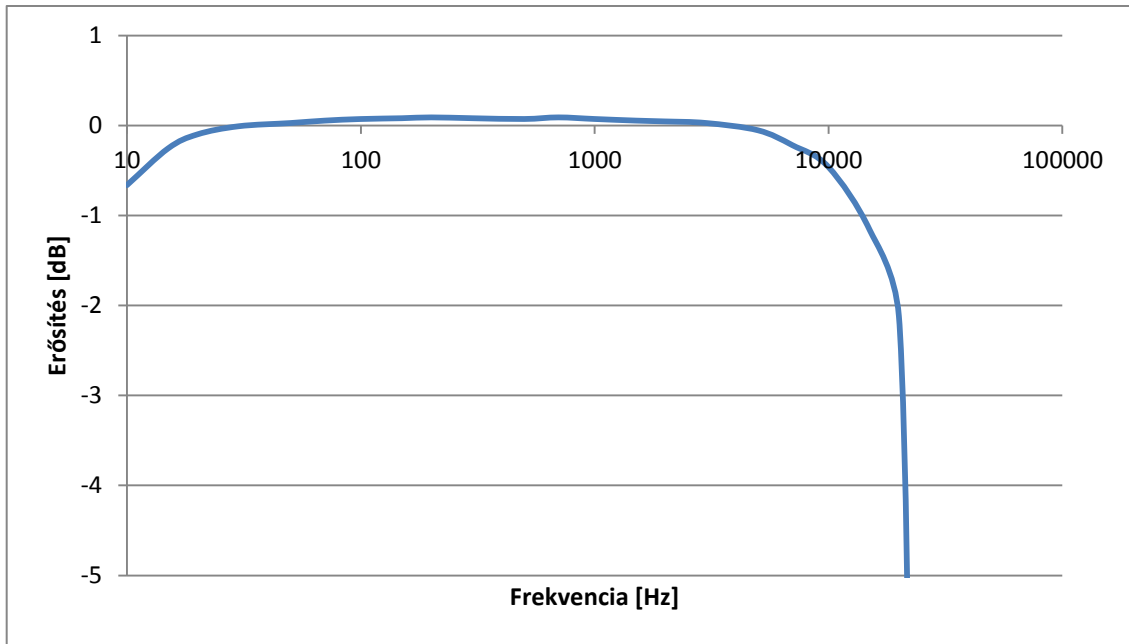
- Phillips PM5192 függvénygenerátor
- LeCroy WaveRunner LT342 digitális oszcilloszkóp
- Krohn-Hite 3322 programozható szűrő

7.1 Hardver tulajdonságainak mérése

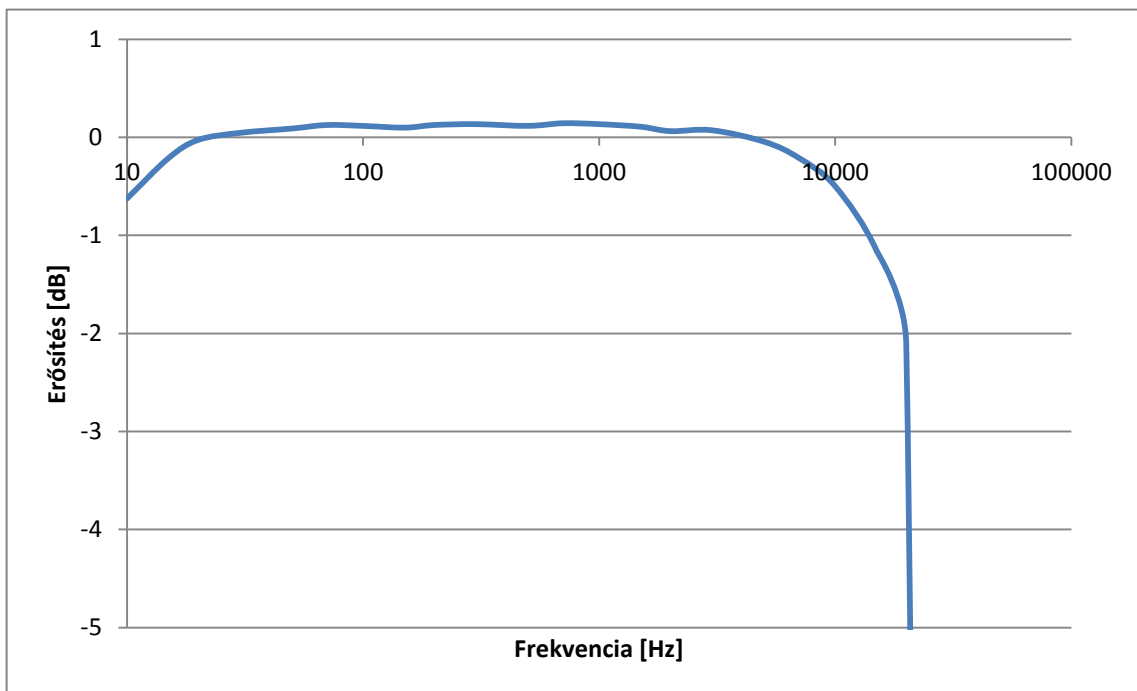
Átvitel mérése

Ez esetben olyan algoritmus futott a DSP-n, ami a mintavételezett jelet skálázás után a kimenetre küldte. A skálázás kiküszöböli a műveleti erősítők által okozott kétszeres erősítést. A skálatényező értéke 0.6219-lelt. Ekkor közepes frekvencián az átvitel éppen egységnyi. Látszik, hogy az erősítés nem pontosan kétszeres, ezért fontos hogy kiküszöböljük.

Az áramkör egyik bemenetére a jelgenerátort, a másik bemenetére az oszcilloszkóp 1. csatornáját csatlakoztattam. A 2. csatornán a jelgenerátor kimenetét mértem. Így mértem az átvitelt különböző frekvenciákon: 10 Hz-től 30 kHz-ig. A mért adatokból táblázatot készítettem, és grafikusán ábrázoltam. Az eredmények a 7-1. és 7-2. ábrán láthatóak.

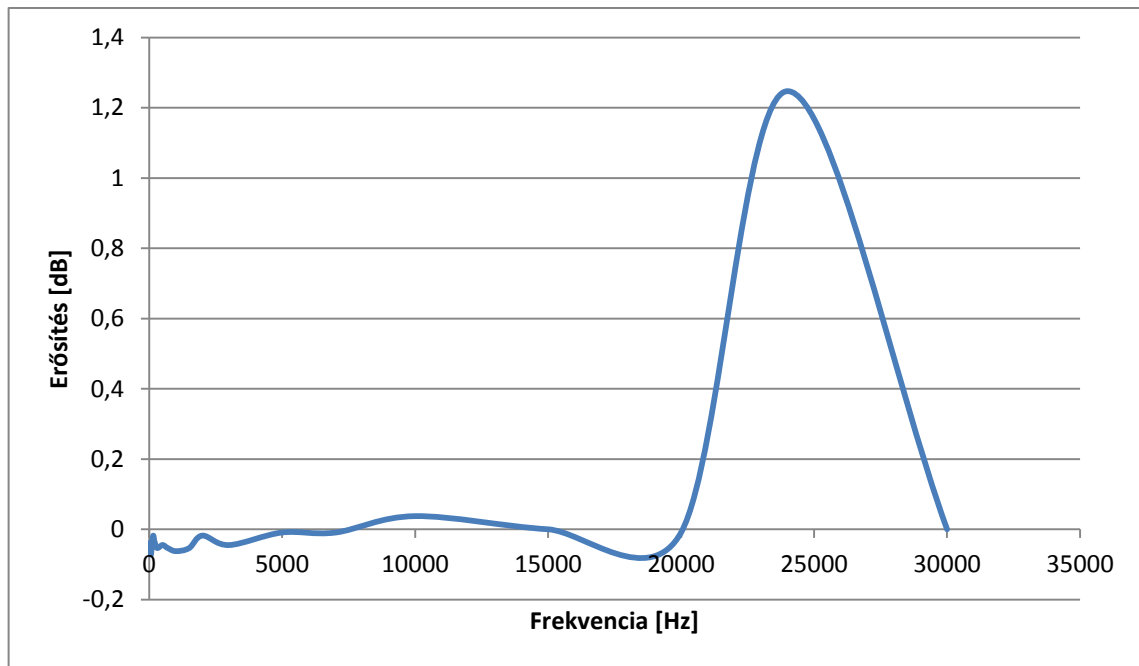


7-1. ábra Jobb csatorna átvitele



7-2. ábra Bal csatorna átvitele

Látható hogy a két csatorna hozzávetőlegesen egyforma. Ebben a felhasználásban azonban a két csatorna átvitelének különbsége sokkal többet mond el a rendszerről.



7-3. ábra Két csatorna átvitelének különbsége

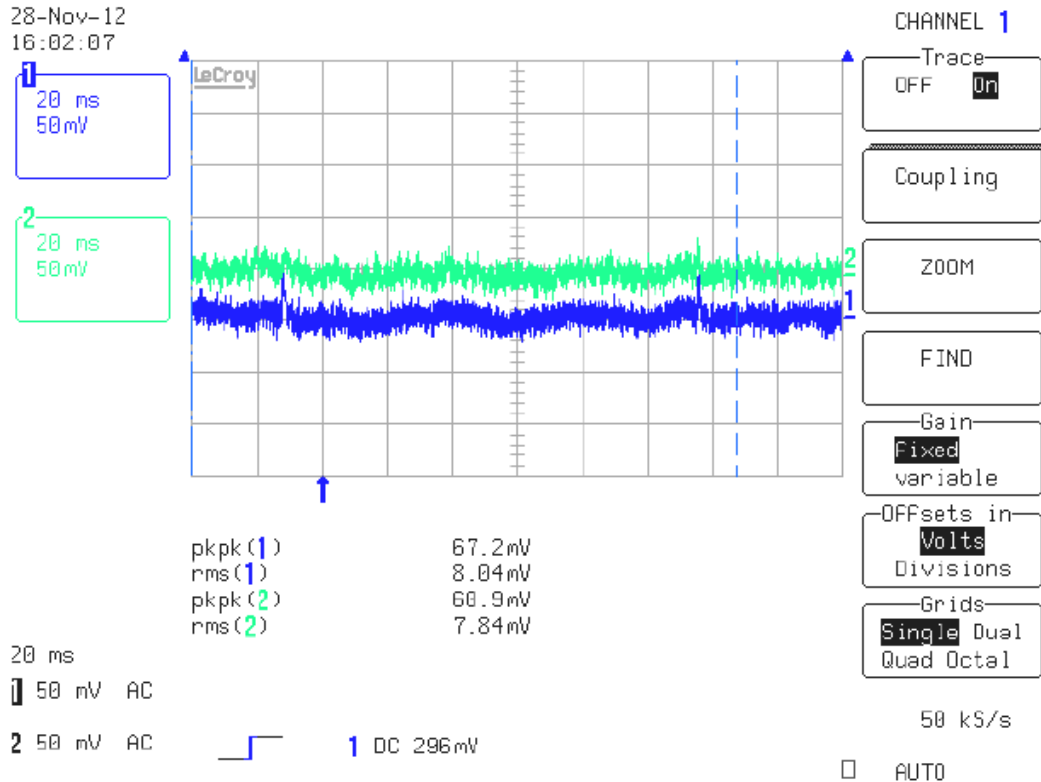
Ez a kép sokat mond a két csatornáról, ha külön-külön mintavételezzük az I-Q jelet. A hiba már a digitalizálás közben eléri a 0.05 dB-t (8-16 kHz-e sávban), így a 6-10. ábra tanulsága szerint a tükörelnyomás legfeljebb 50 dB lehet. Ezért ilyen konstrukciónál a kompenzálás kulcsfontosságú.

Zaj mérése

Olyan algoritmust futtattam, amely az előző mérésben említett skálázáson felül még $A=1000$ -szeresére erősíti a bejövő jelet. Így gyakorlatilag kiküszöbölünk minden hibát, és csak a bemenet zaját mérjük a kimeneten.

Mérési elrendezés:

Az áramkör mindkét bemenetét rövidre zártam és a két kimenetet a programozható szűrőre csatlakoztattam, ezzel a módszerrel próbáltam kiküszöbölni a hasznos sávon kívüli zajok hatását. Ezen egy 24 kHz vágási frekvenciájú szűrőt állítottam be. Ez látszólag nem változtat semmin. A DA konverter fülhallgató kimenete PWM-mel működik, ezért viszonylag nagy nagyobb frekvenciás (200 kHz körüli) zavarjel ül. Ezt nem halljuk, azonban a mérést megghamisítaná. A szűrő kimenetét csatlakoztattam az oszcilloszkópra.



7-4. ábra Bemeneti zaj mérése

A 7-4. ábrán látható, hogy a zaj effektív értéke mindkét csatornán $U_{NORMS} = 8 \text{ mV}$ körül van. A DA konverterre folyamatosan 0-át adva a zaj értéke $U_{NORMS0} = 2.5 \text{ mV}$, ez tekinthető tisztán a kimeneti fokozat zajának. A különbségüket visszaszámolva az AD konverter bemenetére a következő képlet alapján:

$$U_{NIRMS} = \frac{\sqrt{U_{NORMS}^2 - U_{NORMS0}^2}}{A}$$

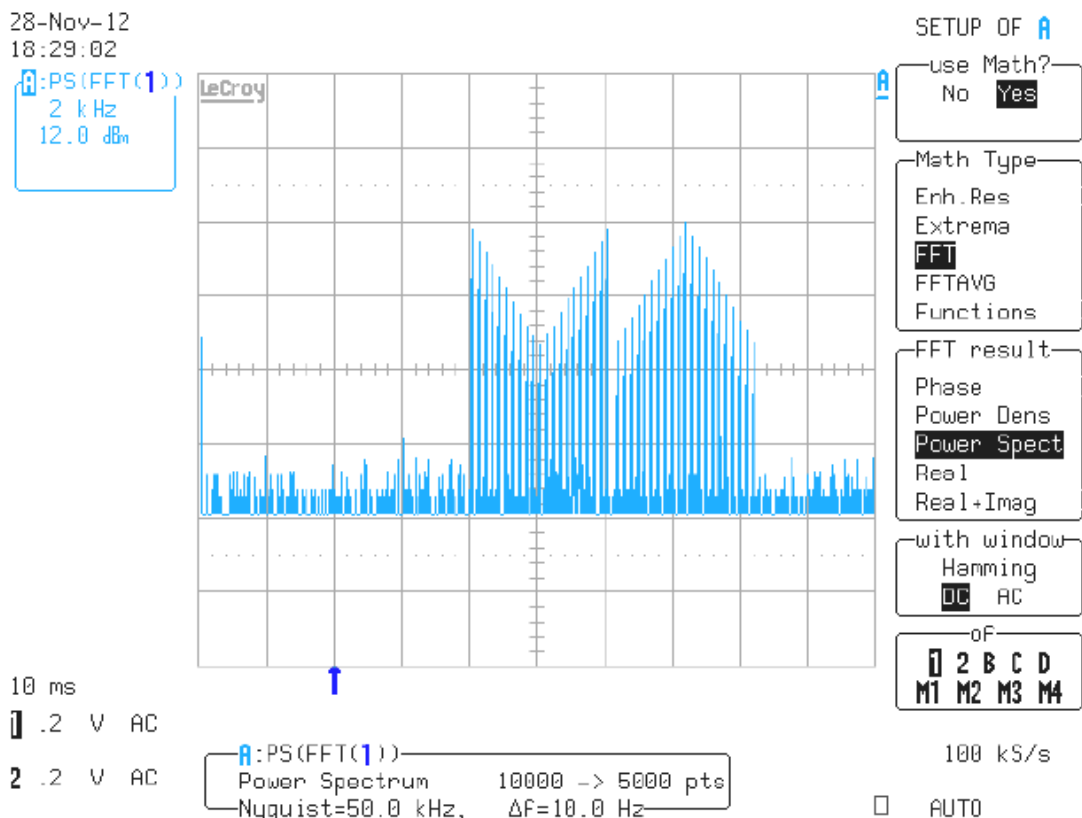
$U_{NIRMS} \sim 7.6 \mu\text{V}$ adódik. Az effektív értékek miatt van szükség a négyzetes különbségképzésre. Ennek a zajnak a nagy részét feltehetőleg a műveleti erősítő termeli. Későbbi fejlesztés során ide kis zajú típust kell majd beépíteni.

7.2 Demodulátor tulajdonságainak mérése

Algoritmus működésének ellenőrzése releváns tesztjellel

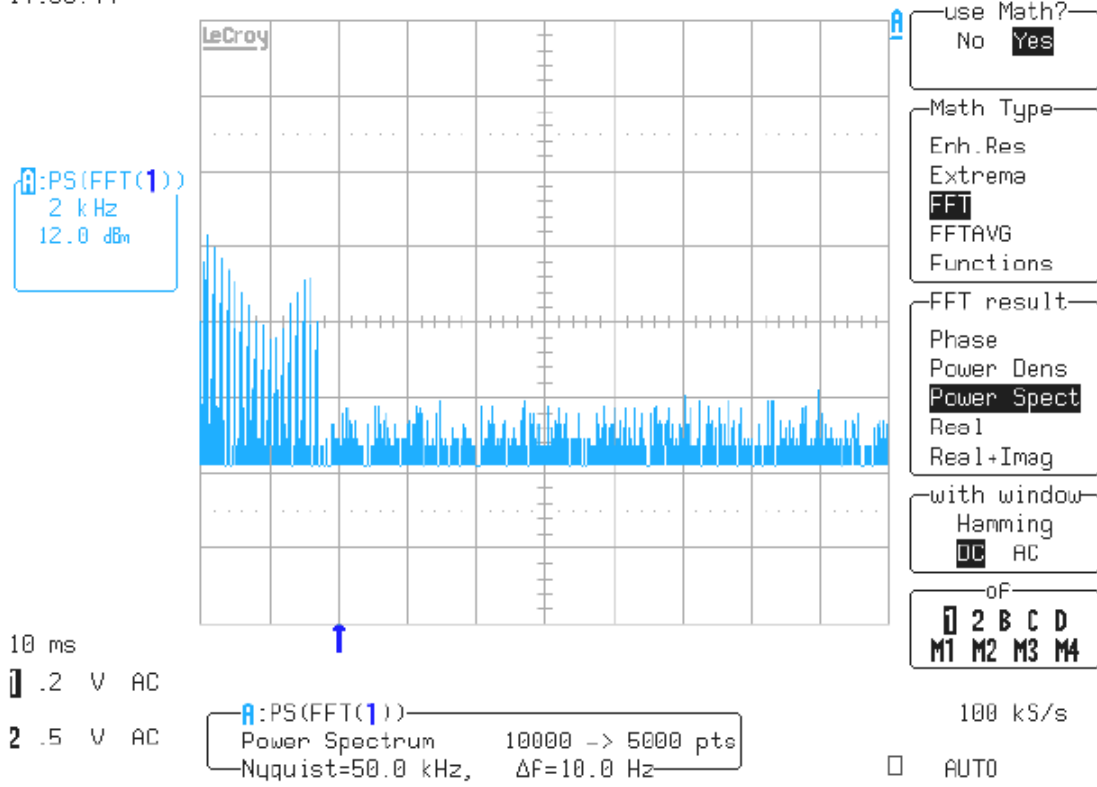
A tesztjel egy multiszinusz lett. Itt az egyes jelek úgy helyezkednek el, hogy betérítik a két oldalsávot. 8 kHz-től 12 kHz-ig 200 Hz-ként, illetve 12.1 kHz-től 16 kHz-ig szintén 200 Hz-ként. A tesztjel spektruma a 7-5. ábrán látható. Az amplitúdók úgy lettek megválasztva, hogy a két oldalsáv jól megkülönböztethető legyen. A jel generálásánál

fontos volt még a fázisok jó megválasztása. Ha az összes jelnek a kezdőfázisa 0 lenne, akkor a csúcstényező nagyon nagy lenne, és nem lehetne jó mérést végezni. Ezért a fázisok egy véletlengenerátor segítségével előállított értékek lettek, így a csúcstényező nem lett nagy. Ezt a jelet Matlabban generáltam, és a számítógép hangkártyáját kapcsoltam az áramkör bemenetére, a kimenetre pedig az oszcilloszkópot. Az oszcilloszkóp FFT funkcióját úgy állítottam be, hogy azt a mintasorozatot, amiből az FFT spektrumot állítja elő, Hamming-ablakkal szorozza be előzetesen. Ez az emelt koszinuszos ablak a nem koherens mintavételezés hatására létrejövő szivárgást csökkenti hatásosan. Látható is az összes spektrumot tartalmazó ábrán (7-5. 7-6. 7-7. 7-8.), hogy az összes szinusz spektrum egy egyenes vonalként látszik, nem „kenődik el” az alja.



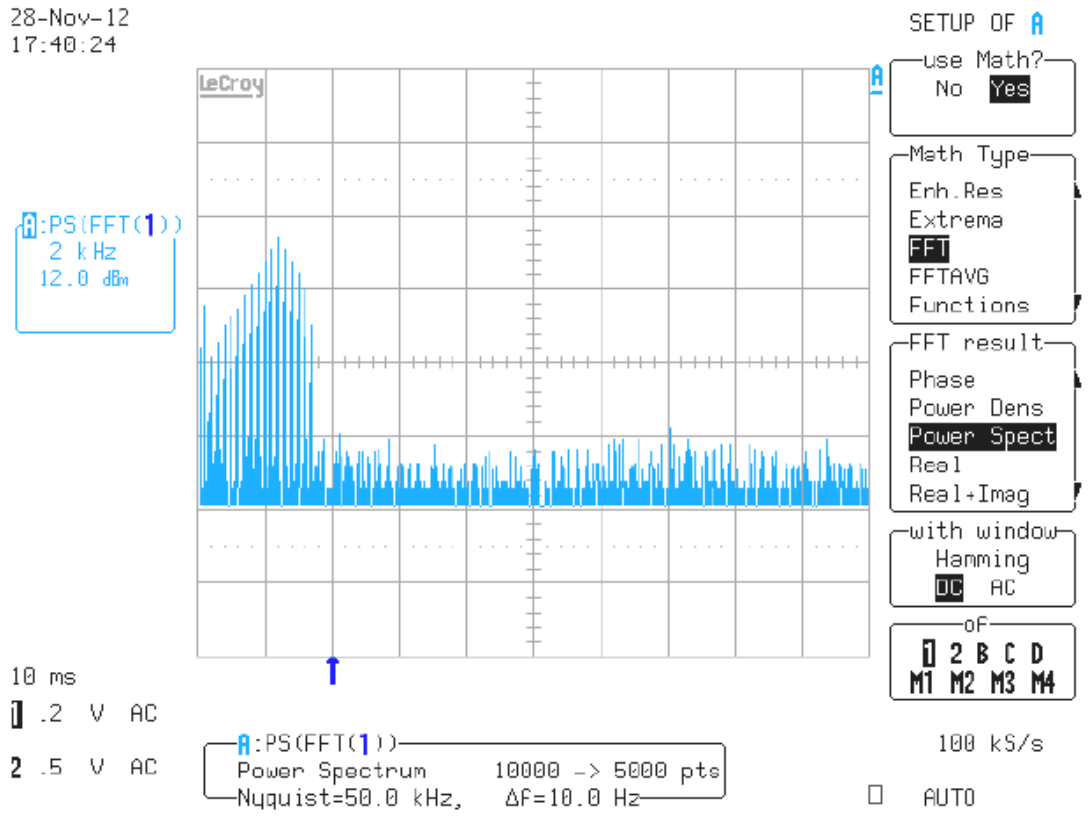
7-5. ábra A mérőjel spektruma [0-20 kHz, 2 kHz/div]

28-Nov-12
17:36:44

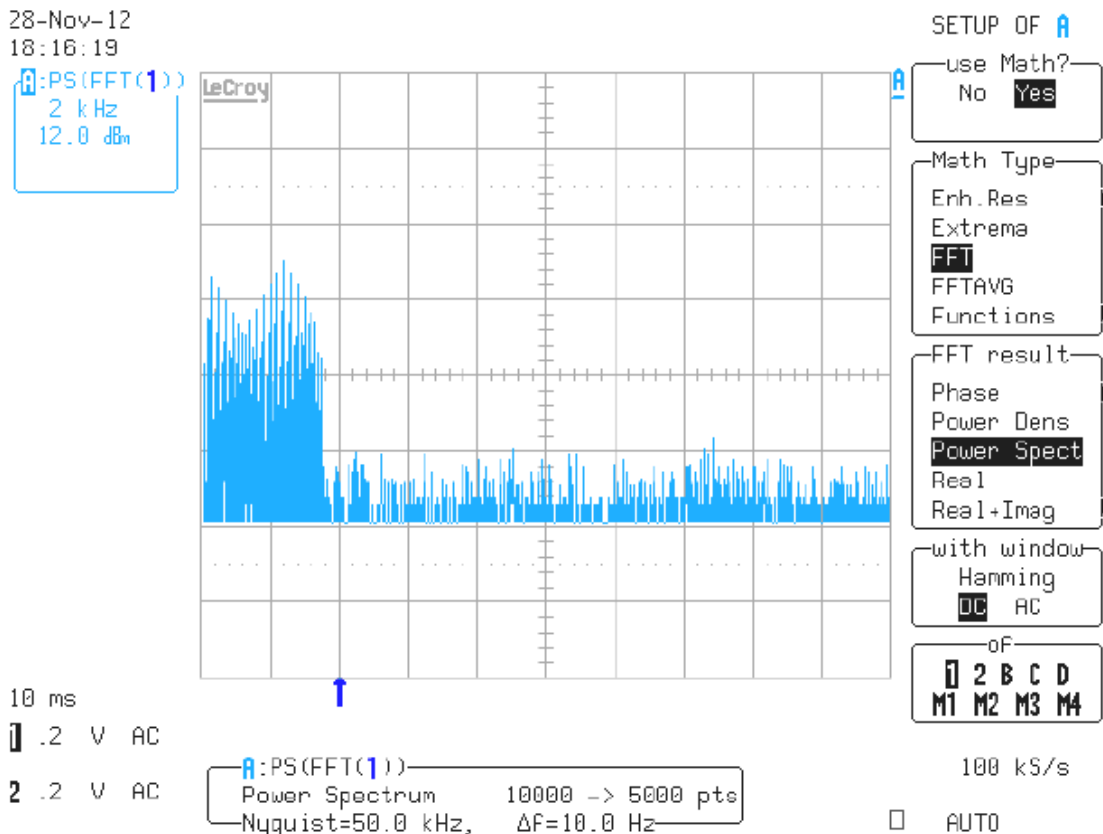


7-6. ábra Áramkör kimenete LSB beállítás [0-20 kHz, 2 kHz/div]

28-Nov-12
17:40:24



7-7. ábra Az áramkör kimenete USB beállítás [0-20 kHz, 2 kHz/div]

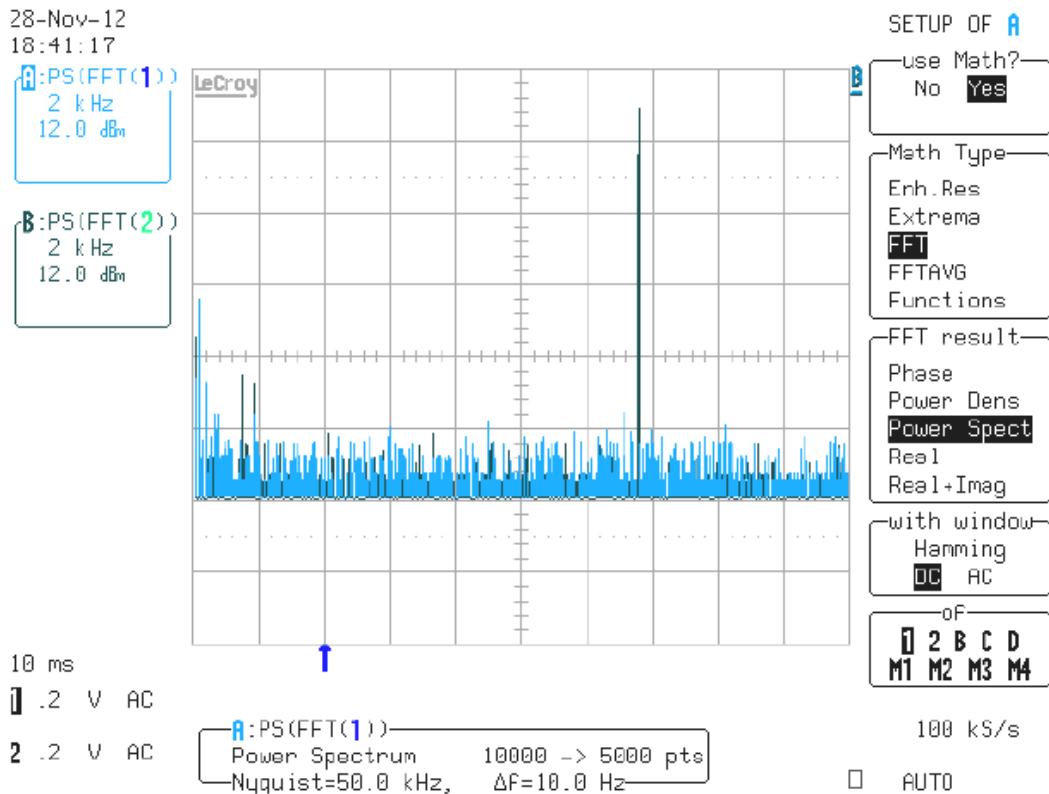


7-8. ábra Áramkör kimenete egyszerű keverést alkalmazva [0-20 kHz, 2 kHz/div]

Az ábrából a következő megállapításokra jutottam: A demodulátor megfelelően működik mind az alsó (7-6. ábra) mind a felső (7-7. ábra) oldalsáv demodulálása esetén. A kimeneti spektrumban nem látható a másik oldalsáv. Látható továbbá az is hogy a decimáló szűrő megfelelően működik, mert a bemenetre egy 4 kHz sávszélességű jel került, ami a kimeneten csomkolva van. Az interpolálás és a decimálás is megfelelően működik, mert a kimeneti spektrumban nem láthatóak 4 kHz feletti komponensek. A 7-8. ábra azt a jelet tartalmazza, amikor nem a Hartley-demodulátort alkalmazom, hanem egy egyszerű keverést. Itt az alsó és felső oldalsáv egymásra lapolódik, mert nincs tükörfrekvenciás elnyomás. Az összes ábrán látható, hogy a spektrum bal oldalán egy vonal van. Ez feltehetőleg az 50 Hz-es hálózati feszültség által keltett zavarjel.

Tükörelnyomás mérése

Itt egy 0.5 V p-p 13.8 kHz frekvenciájú szinuszjelet kapcsoltam az áramkör bemenetére. Az áramkört LSB beállításba tettem, így a kimeneten nem szabadna megjelenie semminek. Ha megjelenik valami, akkor annak és az eredeti jelnek a különbsége adja a tükörelnyomást.



7-9. ábra Tükörelnyomás mérése [0-20 kHz, 2 kHz/div]

A világoskék a kimenet, a sötétkék pedig a bemenet spektruma. Az 1.8 kHz-es frekvencián nem mérhető gyakorlatilag semmi. Ezért megállapítható, hogy a tükörelnyomás jobb, mint 60 dB. A másik oldalsáv esetében felesleges méréseket végezni, mert a működésről már az előző mérésben meggyőződünk, és a digitális feldolgozásnak köszönhetően a rendszer teljesen szimmetrikus, tehát ugyan ezt mérnénk.

Összegzésül elmondható hogy az algoritmus teljesíti a támasztott követelményeket. A processzoridőnek körülbelül a 60%-át foglalja le. Ez azért fontos, mert a keverés utáni rész egy jó alap, mert egy direkt I-Q jelet feldolgozó algoritmusban is szükség van ezekre a lépésekre. Belátható hogy minimális módosítással az algoritmus használható adás módban is, ekkor annyi a változás, hogy az egyes lépéseket fordított sorrendben kell elvégezni.

8 Összefoglalás, kitekintés

A szakdolgozatban sikerült megvalósítani a feladatkiírásban előírtakat, sőt még egy egyszerűbb demodulátor implementálására és mérésére is maradt idő. A megtervezett áramkör kiállta a próbát, a demodulátor megbízhatóan működött. A jövőben egy I-Q kiegyenlítetlenséget kompenzáló algoritmust szeretnék rá tervezni, így képes lenne direkt I-Q jel fogadására. A hardveregységek viszonylag jól működtek, de az AD konverter szimmetrikus jelét előállító műveleti erősítőt kisebb zajúra cserélném. Továbbá tervben van egy, a Discovery paneltől teljesen független áramkör megtervezése. Itt megvalósítható a két analóg rész teljes elektromos árnyékolása.

Irodalomjegyzék

- [1] Rádióvevők logikai felépítése
URL: http://wiki.ham.hu/index.php/Rádióvevők_logikai_felépítése
2012. december 5.
- [2] Robert C. Dixon: Radio Receiver Designs, Marcel Dekker Inc. 1998
- [3] University of Twente rádióamatőr klubja által üzemeltetett WebSDR honlapja
URL: <http://websdr.ewi.utwente.nl/> 2012. december 5.
- [4] Cortex processzorok osztályozása URL: <http://hschi.com>. 2012. december 5
- [5] STM32F4 mikrokontroller család felépítésének blokkvázlata
URL: <http://www.st.com> 2012. december 5
- [6] STM32F4 Discovery panel képe URL: <http://www.st.com> 2012. december 5
- [7] STM32F407VC típusú mikrokontroller referencia adatlapja
URL: <http://www.st.com> 2012. december 5
- [8] Az STM32F4discovery board adatlapja,URL: <http://www.st.com>
2012. december 5.
- [9] PCM1804 típusú AD átalakító adatlapja URL: www.ti.com 2012 december 5.
- [10] Hank Zumbahlen, Staying well grounded, Analog Dialogue 46-06 June 2012
- [11] Tony J. Roupael: RF and Digital Signal Processing for Software Defined Radio
Elsevier Inc. 2009
- [12] I-Q demoduláció elmélete URL: <http://www.microwavejournal.com>
2012. december 5.