



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

**PROCESSZOR ALAPÚ
RÁDIÓKÉSZÜLÉK HARDVER- ÉS
SZOFTVERKOMPONENSEINEK
TERVEZÉSE**

MSc Diplomaterv

KONZULENS

Dr. Orosz György

BUDAPEST, 2015

Tartalomjegyzék

Kivonat.....	5
Abstract.....	6
1. Bevezetés	7
2 Hardverkomponensek tervezése.....	9
2.1 Választott architektúra felépítése.....	9
2.2 Rendszerterv	11
2.3 Tervezési megfontolások	13
2.3.1 Alkatrészek kiválasztása	13
2.3.2 Szabványok használata	14
2.3.3 Nyomtatott áramkör rétegfelépítés kiválasztása	14
2.3.4 Footprintek tervezése	16
2.4 Processzor panel	19
2.4.1 Mikrokontroller kiválasztása	19
2.4.2 AD és DA konverter kiválasztása	22
2.4.3 Egyéb alkatrészek kiválasztása	24
2.4.4 STM Cube.....	25
2.4.5 Kapcsolási rajz tervezése	27
2.4.6 Nyomtatott áramkör tervezése	32
2.5 Szintézer panel	35
2.5.1 Bevezetés	35
2.5.2 Alkatrészek kiválasztása	38
2.5.3 Kapcsolási rajz tervezése	39
2.5.4 Nyomtatott áramkör tervezés	41
2.6 Keverő panel	43
2.6.1 Bevezetés	43
2.6.2 Alkatrészek kiválasztása	46
2.6.3 Kapcsolásirajz tervezése	48
2.6.4 Nyomtatott áramkör tervezése	50
2.7 Relé panel	51
2.8 Szűrő panel	53
2.8.1 Bevezetés	53

2.8.2 Szűrőtervezés	53
3 Szoftverkomponensek tervezése	57
3.1 Bevezetés	57
3.2 Szoftverarchitektúra bemutatása	57
3.3 Implementáció	59
3.3.1 Szintézer hangoló függvény.....	59
3.3.2 Memóriavezérlő konfigurálása, LCD, és SDRAM.....	61
3.3.3 Enkóder, SPI	63
3.3.4 AD/DA konverterek és a DMA	65
4 Algoritmusok bemutatása	67
4.1 SSB demodulátor algoritmus	67
4.1.1 Bevezetés	67
4.1.2 Az algoritmus bemutatása.....	69
4.2 RTTY demoduláló algoritmus	73
4.2.1 Bevezetés	73
4.2.2 Az algoritmus bemutatása.....	73
5 Elkészült eszköz bemutatása	77
6 Összegzés.....	82
Irodalomjegyzék.....	83
Függelék.....	85

HALLGATÓI NYILATKOZAT

Alulírott **Magyar Bálint István**, szigorló hallgató kijelentem, hogy ezt a diplomatervet 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(k), 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, 2015. 05. 24.

.....
Magyar Bálint István

Kivonat

A vezeték nélküli kommunikáció manapság életünk szerves részét képezi. Dolgozatom témája egy szoftverrádió tervezése és megvalósítása, mely a rövidhullámú rádióamatőr sávok vételére alkalmas. A BSc szakdolgozat és az Önálló laboratóriumok során egy ilyen rádió egyes komponenseinek elkészítése volt a feladatom, most pedig a megszerzett tudás és tapasztalat segítségével a teljes rendszer felépítését tűztem ki célul.

Először kiválasztottam a különböző szempontok alapján legmegfelelőbbnek vélt alkatrészeket, majd megterveztem a kapcsolási rajzokat. Ahol szükségesnek láttam, ott szimuláltam az áramköröket. Az alkatrészek footprintjeit szabványok segítségével terveztem meg, hogy könnyen és jól beferraszthatóak legyenek. Munkámat a nyomtatott áramkörök megtervezésével folytattam. Mivel az áramkörök viszonylag komplexek, továbbá a kiválasztott alkatrészek megkövetelik ezt, ezért az áramköröket 4 rétegű NYÁK (Nyomtatott Áramköri Lap) lemezre valósítottam meg. A NYÁK legyártása után beültettem az alkatrészeket, és élesztettem a panelokat. A tervezés során vétett néhány hibát sikeresen elhárítottam. Ezt követően a rendszer szoftverarchitektúráját terveztem meg és implementáltam. Ennek hatására még néhány hardverhiba kiderült, ezeket javítottam. Miután az alapvető működéshez szükséges szoftver komponenseket implementáltam, két algoritmust valósítottam meg, egy SSB demodulátort, illetve egy RTTY dekódert. Végül méréseket végeztem és néhány paraméterét meghatároztam az elkészült eszköznek.

Abstract

Nowadays, wireless communication is an integral part of our lives. The topic of my thesis is to design and accomplish a software defined radio which is able to receive the short wave radio amateur band. At my BSc thesis and project laboratories my task was to implement several component of this kind of radio. In this present thesis, with the gained experience and knowledge I aim to implement the whole system. First – based on various points of view – I chose parts considered to be best for the application, then I have designed the schematics. For critical components, I also simulated some of the schematics. The components' footprints were designed based on some standards, which guarantees the good solderability. I continued the work with the PCB (Printed Circuit Board) design. Because the circuits are complex enough, and some of the components needs, I used 4 copper layer PCB. After manufacturing, I have soldered the components, and tested the PCBs. I should have also correct some flaws which had been inadvertently included at the design phase. Then I designed the software architecture, and implemented it. During the implementation some minor hardware modifications were also necessary. After the basic software components were implemented successfully, I have designed and implemented two signal processing algorithm: an SSB demodulator and a RTTY decoder. Finally I measured some parameters of the radio.

1. Bevezetés

Diplomatervezés keretében folytattam munkámat, egy szoftverrádió elkészítését. A BSc szakdolgozat, illetve Önálló laboratórium tárgyak során egy-egy modul megépítése, tanulmányozása, most pedig a megszerzett tudásból, tapasztalatból, egy működő, használható rádió készítése volt a cél.

Az ilyen típusú rádióknál a rádiófrekvenciás jelet alapsávba vagy közel alapsávba transzponálják, majd digitalizálják, és a feldolgozás digitálisan történik. Ennek előnye, hogy a demodulátor fajtáját az algoritmus határozza meg, így csupán egy szoftvermódosítással teljesen más viselkedés valósítható meg. Ezt egy teljesen analóg rendszerű rádiónál panel cserével, esetleg alkatrészek értékének módosításával érhetjük el, ami jóval körülményesebb. További érv a digitális jelfeldolgozás mellett, hogy a megvalósított algoritmus pontosan úgy fog működni a valóságban, mint a szimulációban, azonban egy analóg áramkörnél ez koránt sincs így, hiszen a viselkedés változhat az alkatrészek szórása, hőmérséklet stb. függvényében.

Rádióamatőr célra több gyártó tradicionálisan gyárt rádiókat (Pl.: Yaesu, Icom, Kenwood), azonban az új típusok ára meglehetősen magas, a régebbi, olcsóbb használt típusok analóg rendszerűek, nem vehetik fel a versenyt fiatalabb társaival. További érv a saját rádió építése mellett, hogy a szoftver saját készítésű, így módosítható, és a megvalósítható algoritmusoknak csak a limitált számítási teljesítmény szab határt.

Az eszközök számítási kapacitásainak drasztikus növekedése miatt megjelentek olyan eszközök, melyek általános célúak, a jelfeldolgozó egy FPGA, esetleg valamilyen előfeldolgozás után egy PC. Az előbbire példa az USRP, utóbbira a HackRF. Ezek azonban áruk és korlátozott mobilitásuk miatt (például egy kitelepülés alkalmával egy számítógépet kellene magunkkal vinni) rádióamatőr célra csak korlátozottan alkalmas. Azonban találhatunk az általam készített rádióhoz architektúrában nagyon hasonló típust, ennek neve SDR Cube. A felsorolt típusok néhány évesek, és folyamatosan fejlesztenek újabbakat tehát bátran kijelenthető, hogy van létjogosultsága egy ilyen készülék tervezésének.

A következőkben bemutatom diplomamunkám felépítését.

A 2. fejezetben a választott struktúra felépítéséről majd a hardvertervezésről lesz szó. Először rendszertervet készítettem el. Ezt követően kiválasztottam az alkalmazáshoz

legmegfelelőbb alkatrészeket, és megterveztem a kapcsolási rajzokat. Ezek alapján terveztem meg a nyomtatott áramkört. A panelok legyártása, majd beültetés után funkcionális teszt következett. A megtalált hibákat kijavítottam.

A 3. fejezetben a szoftvertervezést mutatom be. Itt a hardver működéséhez szükséges komponenseket implementáltam. A munka során a gyártó függvénykönyvtárát használtam. Programozás során fény derült még néhány hardverhibára, ezeket javítottam. Miután a rendszerek működéséhez minden komponens adott volt, elkezdem a jelfeldolgozó algoritmusok fejlesztését.

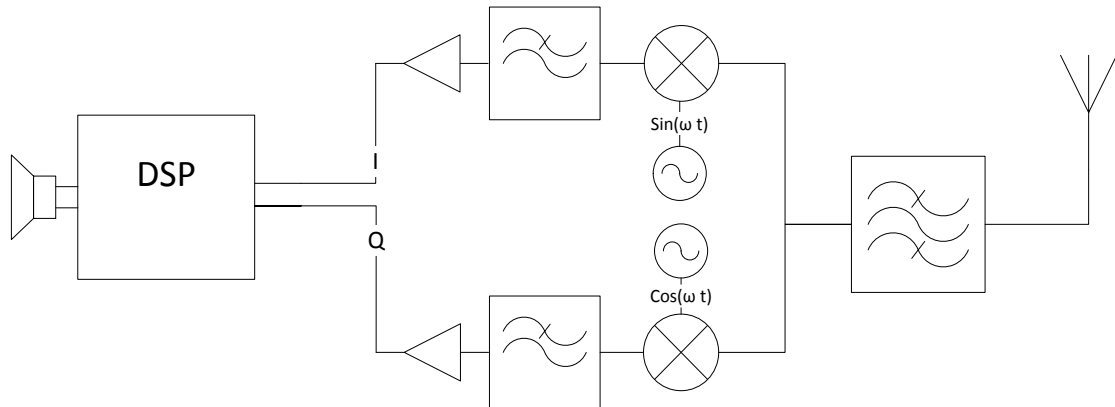
A 4. fejezetben az elkészült algoritmusokat mutatom be. Egy analóg illetve egy digitális adás vételére alkalmas algoritmust implementáltam: egy SSB demodulátort illetve egy RTTY dekódert.

Az 5. fejezetben bemutatom az elkészült rádiót, illetve néhány mérést, melyek az elkészült panelok jellemzőit mutatják be.

A 6. fejezetben összegzem az elvégzett munkát, és néhány továbbfejlesztési lehetőséget mutatok be.

2 Hardverkomponensek tervezése

2.1 Választott architektúra felépítése



1. ábra Direkt alapsávba keverő rádióvevő elvi felépítése

Az átláthatóság kedvéért csak a vételi oldalt mutatom be. Adás oldalon gyakorlatilag ugyan ez történik csak fordítva.

Az antenna jele egy sávszűrőre kerül. A sávszűrő csak a venni kívánt sávot, engedi át, kiszűri a sávon kívüli zavarjeleket. Ennek több kedvező hatása is van: megakadályozza a sávon kívüli nagy térerejű műsorszóró állomások okozta intermodulációt illetve kiszűri a lokáloszcillátor frekvenciájának páratlan felharmonikusain található állomásokat, amit a keverő – felépítéséből adódóan – szintén alapsávba transzponál. Problémaként róható fel hogy minden rádióamatőr sávra külön sávszűrőt kell készíteni, a szűrőket behangolni viszonylag időigényes, és nem kis többletköltséggel is jár.

A keverők egyik bemenőjele a lokáloszcillátor, a másik a szűrt jel. A két keverő lokáloszcillátora ugyanazon a frekvencián jár, csak a fázisuk különbözik, mégpedig 90° -kal. Ez egy nemlineáris áramkör, összeszorozza a bemeneteire kerülő jeleket. A kimeneten megjelenik az összeg és a különbségi frekvenciás komponens is. Előbbit kiszűrjük, az utóbbit pedig feldolgozzuk. Mivel direkt alapsávba keverünk, a lokáloszcillátor alatti, illetve feletti spektrum összekeveredik. Ezt küszöböli ki a kettős keverés, ugyanis az I és Q csatorna együttesen kezelve egy komplex jelet alkot, és a

digitalizálás után DSP-ben különféle jelfeldolgozó algoritmusokkal szétválasztható a két oldalsáv.

A következőkben összegyűjtöttem az ilyen típusú rádiók előnyeit, illetve hátrányait:

Előnyök:

- A jelfeldolgozó algoritmus tisztán numerikus, ezért jól szimulálható.
- Mivel a modulátor egy algoritmus, nincsen gyártási szórás.
- Viszonylag egyszerű frontend más architektúrákhoz képest (pl.: kétszeresen kevert superheterodin).
- A bemenő sávszűrő kivételével az egész architektúra integrálható egyetlen monolit chipbe.

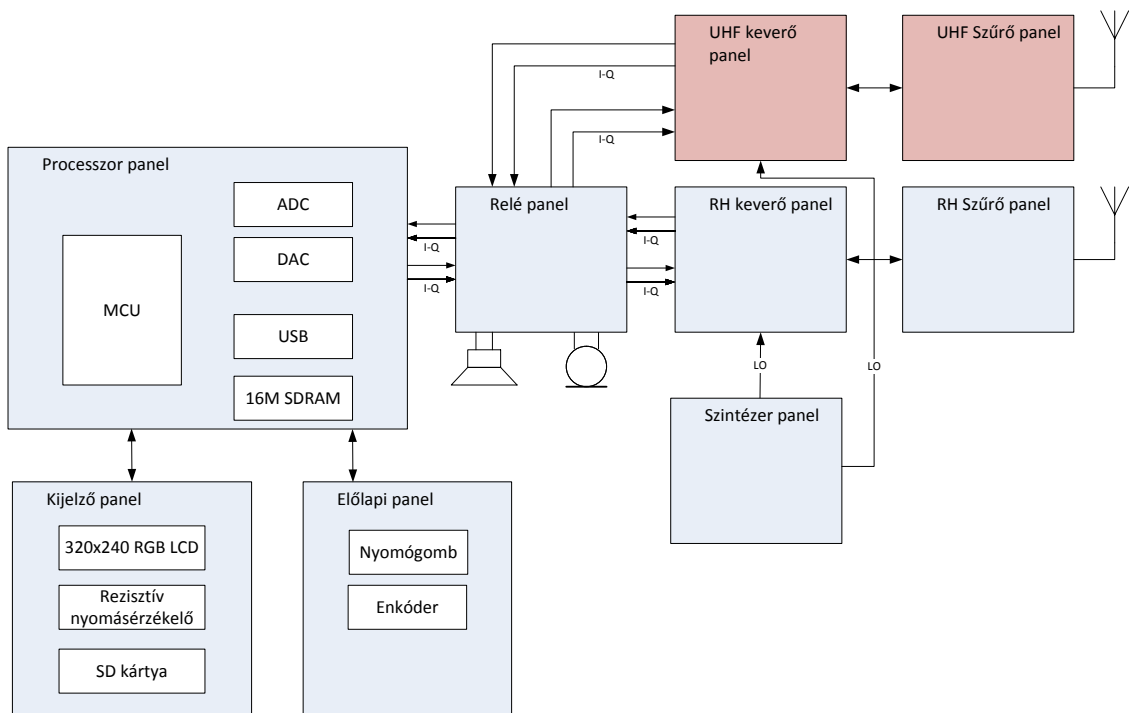
Hátrányok:

- Nagyon érzékeny az I-Q csatornák közti különbségekre.

2.2 Rendszerterv

A munkát egy rendszerterv elkészítésével kezdtem.

A fejlesztés egy iteratív folyamat, azonban a bemutatásban kell egy sorrendet tartani az egyes fejlesztési lépések között.



2. ábra Rendszerterv

A rendszerterv összeállítása során a hatáslánc, illetve az egyes modulok működésének definiálása volt a cél. Törekedtem arra, hogy az egyes modulok csak a legszükségesebb mértékben függjenek egymástól, és külön-külön is használhatók legyenek. Ezt soros buszok használatával értem el, illetve minden panel helyben állítja elő a számára szükséges tápfeszültségeket. A következőkben a fontosabb részekeségeket főbb tulajdonságait ismertetem röviden.

Processzor panel: Az első és egyben a legnagyobb egység. Ide fut be minden jel, illetve ez az egység vezérel mindent. Ezen található a mikrokontroller, az AD illetve a DA konverter, egy USB 2.0 OTG csatlakozó (OTG: On The Go, az eszköz működhet host és slave módban is), 16 Mbyte SDRAM, illetve csatlakozófelület a különböző perifériák részére:

- SPI, I2C buszok a többi panelon található IC-k vezérlésére

- 16 bites 8086 szerű busz az LCD kijelző részére
- Három darab enkóder interfész
- Fennmaradó szabad I/O-k kivezetése

Tápfeszültség igénye $\pm 5.5V$.

Kijelző panel: Ez az egyetlen olyan részegység, amelyet készen vettem. A panel tartalmaz egy 3.2' 320x240 felbontású színes LCD kijelzőt, illetve egy hozzá tartozó rezisztív nyomásérzékelőt. Található rajta továbbá egy SD kártya foglalat, sajnos az ebbe helyezett SD kártyát csak 1 bites SD, vagy SPI módban használhatjuk, ugyanis nincs minden kivezetve. A modul minden kivezetése a processzorkártyába fut.

Előlap panel: 12 nyomógombot, illetve két enkódert tartalmaz. A tervezés kezdeti fázisában még a konkrét program, illetve funkciók nem tisztázottak, ezért jobbnak láttam nyomógombokat is tervezni, mint csupán az érintőkijelzőre hagyatkozni. Az enkóderek viszont nélkülözhetetlenek, hiszen a hangolás, hangerő beállítás csak így oldható meg kényelmesen.

Relé panel: Ez a panel kizárólag kapcsolási feladatokat lát el. A jel igen sok irányba mehet állapottól függően (a keverők a mikrofon/hangszóró, illetve a processzorkártya között), ezért fontosnak tartottam egy olyan kis panel létrehozását, amibe minden jel befut, és beállítástól függően létrehozza a szükséges összeköttetéseket.

Keverő panelek: Ez a panel végzi a frekvenciatranszponálást, illetve az alapsávi erősítést. Bemenete vételi módban a szűrt RF jel, és a lokáloszcillátor jele, kimenete pedig az alapsávi jel, adás módban az irányok értelemszerűen változnak. Két keverőt tartalmaz, egyet adásra egyet vételre. Azért ezt a megoldást választottam, mert a keverő viszonylag olcsó a reléhez képest, továbbá sok vonalat kellene kapcsolni, ha adni és venni is kívánunk ugyan azzal a keverővel. A rendszertervben található egy URH keverő is, ezt más színnel jelöltem. Ennek oka, hogy ez nem készült el, de a későbbiekben a rádiót szeretném használni az URH sávokban is, ezért a rendszert úgy építettem fel hogy ez utólag könnyen beépíthető legyen.

Ezen panel minősége döntően befolyásolja a rádió paramétereit.

Szintézer panel: Feladata a lokáljel előállítás a keverő számára. Gyakorlatilag egy hőfokkompenzált referenciaoszcillátort (TCXO), illetve egy tört osztásarányú (Fractional-N) fajtájú PLL-t tartalmaz. Ennek különlegessége, hogy a frekvencia nagyon

finom lépésekben állítható (Egy egész osztásarányú PLL segítségével csak a referencia egész számú többszörösével). Az előállított lokáljel a keverőbe kerül. Szélessávú szintézer kell választani, hogy a szintézerpanel mind a rövid- mind pedig az ultrarövid hullámú sávokon is működőképes legyen.

Szűrő panel: Itt nyolc darab, háromfokozatú a rövidhullámú rádióamatőr sávokra hangolt, relével kapcsolható szűrőnek van helye. Hasonlóan az URH keverőhöz az URH sávokra hangolt szűrőbanknak is csak a helye került kialakításra.

2.3 Tervezési megfontolások

Még a tényleges tervezés előtt meghatároztam magamnak néhány szabályt. Ezeket rugalmasan kezeltem, ha betartásuk jelentős többletmunkával járna, vagy minimális többletköltséggel áthidalható lenne, akkor lehet enyhíteni rajtuk, de csak bizonyos keretek között.

2.3.1 Alkatrészek kiválasztása

A tervezést az alkatrészek kiválogatásával kezdtem. Itt több érdek feszül egymásnak:

- Beültethetőség: Legfeljebb QFN tokozású IC-eket válogatok ki, a BGA-t a házi beforraszthatóság nehézségei miatt kerülni kell.
- Beszerezhetőség: A választott IC beszerezhető legyen nagyobb disztribútoroknál (pl.: Digikey, Arrow, stb.), és ha lehetséges beszerezhető legyen belőle mintadarab (sample).
- Helyettesíthetőség: Ez legfeljebb a passzív alkatrészek illetve csatlakozók esetében tarthatók be, ugyanis a használt IC-k speciális cél IC-k, nagyon ritka, ha létezik belőle helyettesítő darab.
- Támogatás: Jó, ha egy olyan cég termékét választjuk, aki nagy hangsúlyt fektet a terméktámogatásra. Így ha valami olyan problémánk adódik, amit nem ír le az adatlap, akkor hamar választ kaphatunk.
- saját tapasztalat: Nagyban lerövidítheti a fejlesztési időt, ha olyan eszközt használunk, amit már ismerünk. Itt mérlegelni kell, hogy egy jobb, de ismeretlen eszköz használata mennyivel nyújtja meg a fejlesztési időt.

Ezek után listát készítettem a lehetséges alkatrészekről, majd a szempontok figyelembevételével kiválasztottam a számomra legmegfelelőbbeket. Ezekről az adott panelhoz tartozó fejezetekben térek ki.

2.3.2 Szabványok használata

Nagyon fontosnak tartottam, hogy ahol lehetőség van rá, ott alkalmazzak szabványokat a rádió elkészítése során.

Munkám során néhány IPC¹ által készített szabványt használtam. Ez a társaság kifejezetten az elektronikai gyártásra vonatkozó szabványokat állít elő. Az IPC szabványok használata az iparban igencsak elterjedt. Munkám során a microstrip vonalak, illetve az alkatrész rajzolatok (footprintek) tervezésénél vettem hasznát. Maga a szabvány általában pénzbe kerül, én nem is a szabványt vettem meg, hanem olyan programokat használtam, amelyek ezen szabványok alapján működnek.

2.3.3 Nyomtatott áramkör rétegfelépítés kiválasztása

A kapcsolási rajzok elkészítése után látszott, hogy a tervezett áramkörök nem valósíthatók meg a szokásos két rétegű NYÁK lemezen.

A kiválasztott integrált áramkörök döntő része QFN tokozású. Ez a típusú tokozás az aljára gőzölt úgynevezett thermal pad-en keresztül képes elvezetni a chipben keletkező hőt. Ez különösen fontos, mert a PLL illetve osztó IC nagyon sokat fogyaszt (1-1.5W), illetve a lineáris feszültség stabilizátor IC-k is egy kivételtől eltekintve ilyen tokozásúak. A thermal pad általában a földhöz van kötve a chip belsejében, így egy dedikált föld rétegen keresztül hatékonyan elvezethető a hő.

A processzorkártyán található a 176 lábú processzor, egy 48 lábú SDRAM, a viszonylag zavarérzékeny 24 bites AD és DA konverter. A chipek méretei, és lábszámai miatt két rétegen ennek behuzalozása gyakorlatilag lehetetlen.

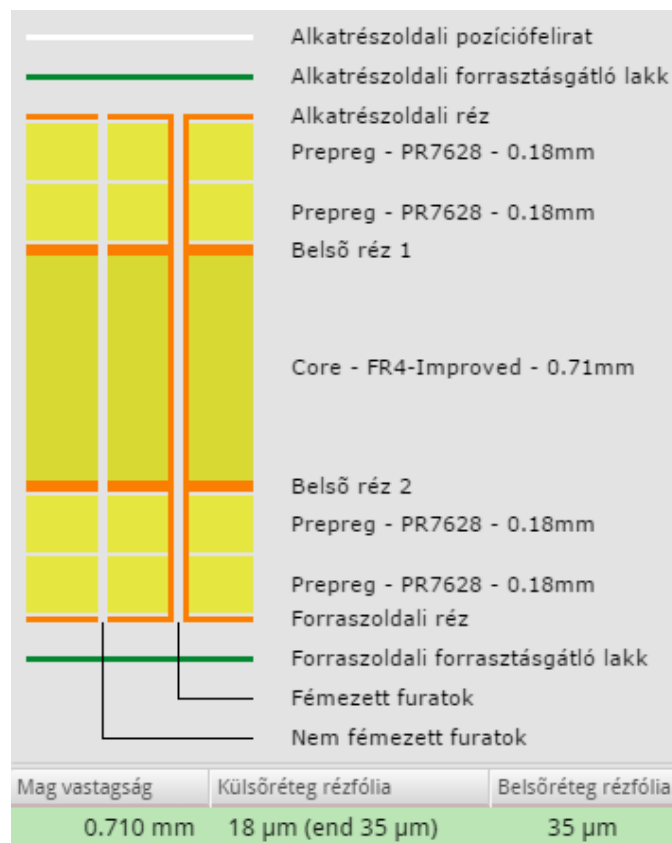
Megnéztem néhány kiválasztott IC próbapaneljét (evaluation board), ezek kivétel nélkül négy rétegen voltak megvalósítva.

Egy dedikált föld-réteg kialakításával egy panelon belül gyakorlatilag megszüntethetők a rossz földelésből eredeztethető problémák. Mivel esetemben vannak igen nagy

¹ Institute for Printed Circuits

frekvenciás áramkörök (a választott szintézer maximális kimenő frekvenciája 4.4 GHz) ahol a helytelen földelésből komoly problémák adódhatnak.

Ennyi érv elég volt, hogy meghozzam a döntést: 4 rétegű NYÁK lemez szükséges. A panelt az Eurocircuitsnál gyártattam le. Fontosnak éreztem már az elején eldönteni, hogy hol készíttetem el a nyákot, mert így előre ismerem az adott cég technológiai kereteit, illetve a rétegfelépítést. Az utóbbi azért fontos, mert a panel tartalmaz néhány microstrip vonalat, melyek impedanciája függ a rétegfelépítéstől. A használt rétegfelépítést a 3. ábra mutatja.



3. ábra A választott rétegfelépítés [1]

A rétegfelépítéstől független a rajzolatfinomság, amely megadja a legvékonyabb vezetőrétet (0.15 mm), szigetelőréteget (0.15 mm), és a legkisebb furatátmérőt (0.25 mm), illetve a hozzá tartozó rézgyűrűt (0.6 mm). Ez a még nem feláras finomsági osztály.

A 3. ábra minden lényeges információ tartalmaz az alkalmazott hordozó dielektromos állandóját kivéve. A prepregé $\epsilon_r=4.8$ az FR4-es magé pedig $\epsilon_r=4.6$. A rétegfelépítés minden panelon úgy lett kialakítva, hogy a belső rétegek közül egyik a föld a másik pedig

a tápfeszültségek. A két külső rétegen pedig a jelvezetékek helyezkednek el. Itt használtam microstrip vonalakat is. Ezeknek a méretezését az IPC-2141 szabványcsomag tartalmazza. Differenciális microstrip párt kellett méreteznem, mert ilyenén kell vezetni az USB differenciális jelvezetékét illetve a szintézer kimenetét is. Előbbi 90 Ω -ra, míg utóbbi 100 Ω -ra méretezett microstrip vonalat igényel. Az ehhez szükséges struktúrát a PCB Toolkit nevű, ingyenes programmal terveztem meg. Itt megadhatjuk a bemenő paramétereket és ebből a program kiszámolja az impedanciát a fent említett IPC-2141 szerinti összefüggések alapján. Néhány iteráció utána elérhető a kívánt érték.

Számolt értékek:

90 Ω :

W=0.3 mm

S=0.15 mm

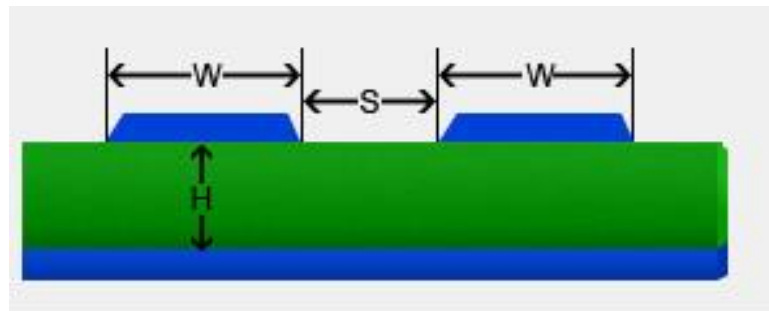
H=0.36 mm

100 Ω :

W=0.25 mm

S=0.15 mm

H=0.36 mm



4. ábra Microstrip differenciális vezetékpár

Viszonylag széles vezetők adódtak ki, ez annak köszönhető, hogy a belső réteg és a külső réteg között viszonylag nagy távolság van.

2.3.4 Footprintek tervezése

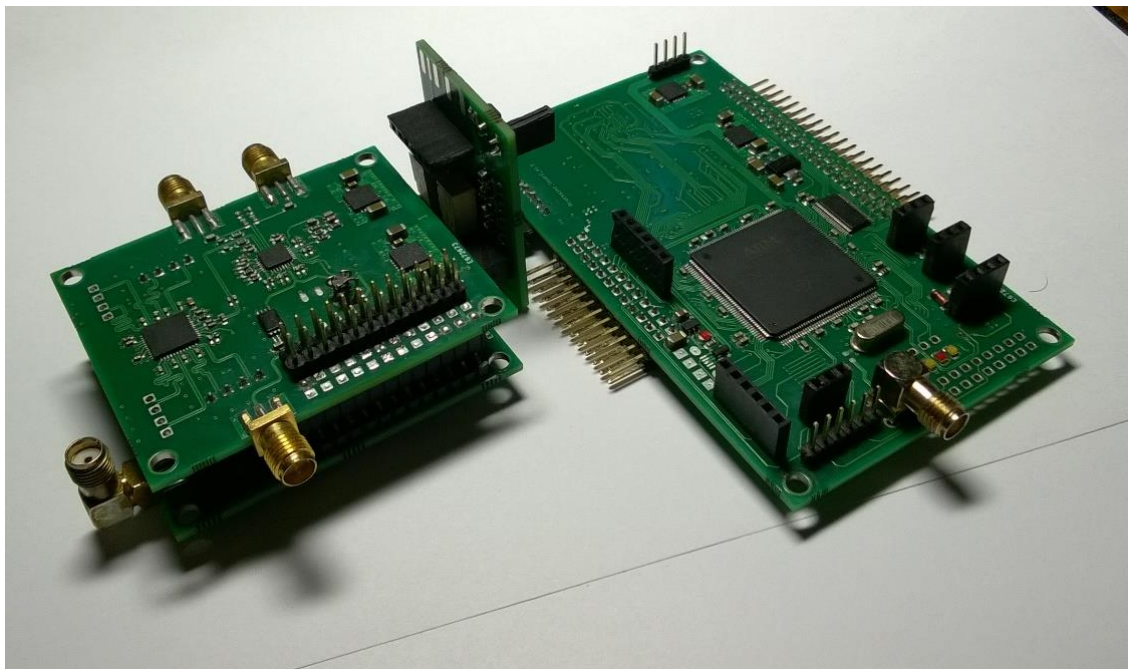
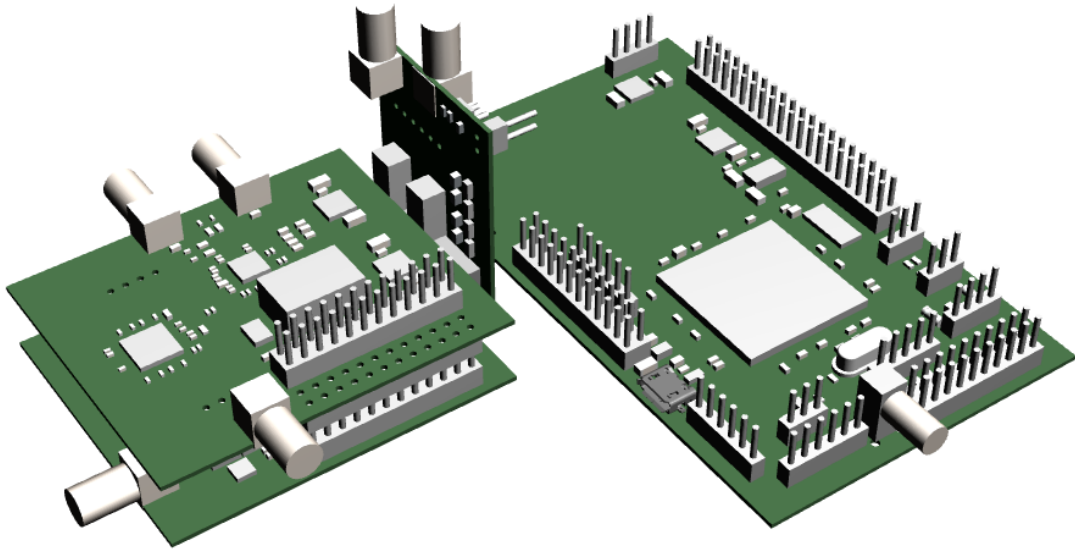
Footprintnek nevezzük a nyomtatott áramköri panelen az alkatrészek beförasztására kialakított rajzolatot. Ez a másik olyan pont ahol a szabványokra támaszkodtam. Szinte minden áramkörtervező program rendelkezik beépített, esetleg letölthető alkatrészkönyvtárral, amely az alkatrész sematikus rajzát illetve footprintjét is tartalmazza. Régebbi projectjeim során már előfordult, hogy egy ilyen footprint nem pontosan illeszkedett az alkatrészhez. Ez általában a panel legyártása után derül ki, ami

elég bosszantó. Tekintettel a 4 rétegű panel viszonylag magas költségeire nem kockáztattam, és minden használt alkatrész sematikus rajzát, illetve footprintjét magam rajzoltam be a programba. Ez elég időigényes, a felhasznált, nagyjából 60 féle tokozás körülbelül 30 órát vett igénybe, cserébe az adatlapok alapján, tényleg az adott alkatrésze szabott footprinteket kaptam cserébe. Az adatlapokon általában az alkatrész fizikai mérete, illetve egy ajánlott footprint található. Az utóbbit nem használtam, hanem a választott áramkörtervező program (Altium Designer 14) által felkínált IPC-7351 szabványcsomagot alapul vevő megoldást választottam. A szabvány a tokozás és fizikai méretek alapján különböző formulák szerint meghatározza az optimális méreteket, hogy az alkatrész a legkönnyebben beferrasztható legyen. Ennek ellenére a 0402-es alkatrészek illetve egy 0.4 mm lábtávolságú IC beferrasztása nem volt teljesen zökkenőmentes, ez azonban az otthoni technológia korlátait mintsem a tervezés hibáját jelenti.

Az Altium footprint generátora nem csupán footprintet generál a program, hanem egy háromdimenziós modellt is. Ez általában csak egy téglatest vagy kocka, azonban bonyolultabb formájú alkatrészek esetén (kvarc, USB csatlakozó) lehetőség van STEP² modellek importálására. Lehetőség van arra is, hogy megrajzolhatjuk mi magunk, azonban ez nem egy 3D modellező program ezért ez viszonylag nehézkes. Az Altium ugyanúgy jelzi a 3D alkatrészek ütközését, mint például a furatok vagy pad-ek ütközését, bármikor átválthatunk 3D nézetbe, továbbá exportálhatók a kész NYÁK-ok STEP modelljei is. Ez rendkívül hasznos, hiszen ha például egy dobozt akarunk készíteni a panelokhoz, akkor a modellező programban elég megnyitni a generált STEP modellt, és innen minden méret levehető. Az 5. ábra tetején az Altiumból exportált, összekapcsolt

² STEP: Standard for the Exchange of Product model data, azaz szabvány a termék modell adatok cseréjére. Ez egy szabványos adatformátum melyet minden modellező program támogat (AutoCAD, SolidEdge, stb.)

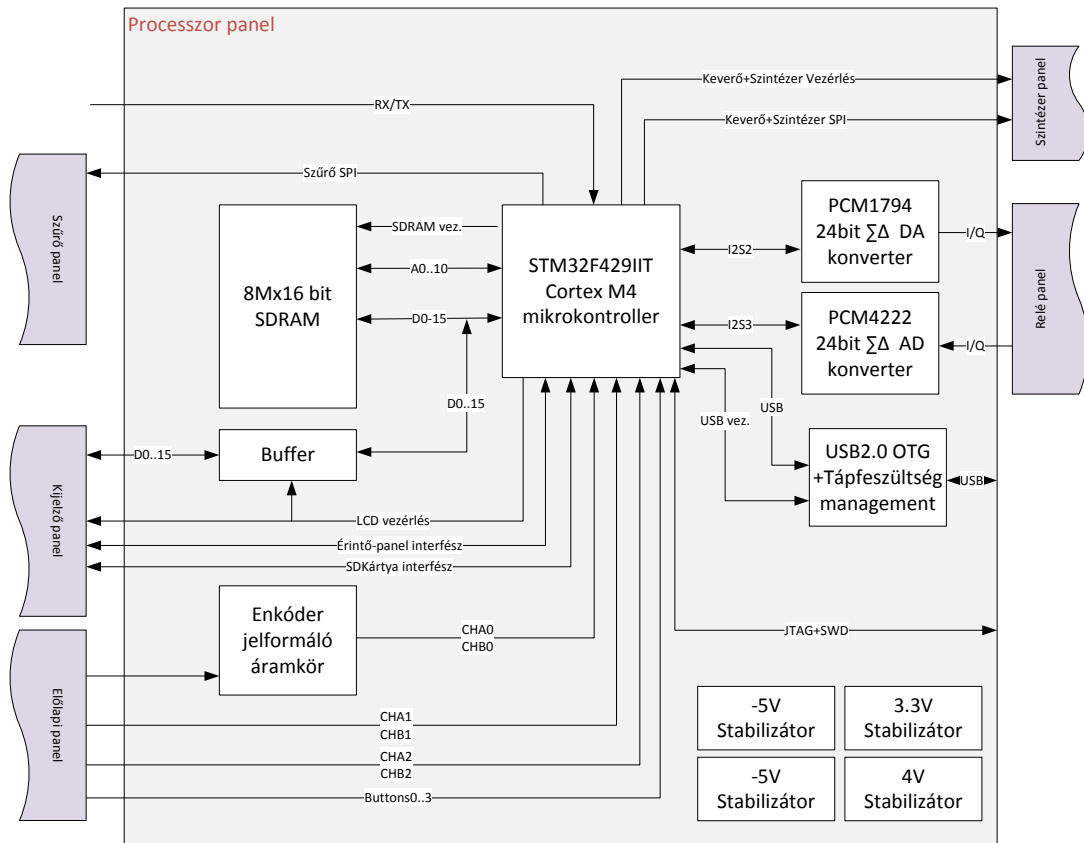
modellek láthatók, alatta pedig az elkészült áramkör. Némi változtatás történt a modellhez képest.



5. ábra A modell (felső ábra), és a valóság (alsó ábra)

2.4 Processzor panel

A processzor panel a legbonyolultabb panel, illetve ez a központi egysége a rádióknak, ezért a bemutatást ezzel kezdem.



6. ábra A processzorpanel blokkvázlata

Az 6. ábra mutatja a panel blokkvázlatát. A következőkben bemutatom az alkatrészek kiválasztásának menetét, illetve röviden elemzem a választott példányokat.

2.4.1 Mikrokontrollerek kiválasztása

Először a mikrokontrollerek kiválasztásával kezdtem a munkát, majd a választott eszköz köré építettem fel a kártyát. Mivel a jelfeldolgozó panel előző verziója STM32F4xxx platformra épült, ezért célszerű volt hasonló kontrollert választani. Egyrészt azért, mert az előző verzió tervezése során [2] sok hasznos ismeretet szereztem erről a családról, másrészt a már meglévő néhány ezer sornyi kódot (amelynek nagy része a mikrokontroller speciális funkcióit használja) nagyon nagy munka lett volna átültetni egy másik platformra. Fontos szempont volt még a tokozás, csak a TQFP tokozású típusok jöhettek szóba. Így az említett család második legnagyobb tagjára esett a

választásom, nevezetesen az STM32F429IIT-re. Ez egy ST Microelectronics által gyártott ARM CortexM4 alapú mikrokontroller. A család tagjai felfelé teljesen mértékben kompatibilisek, így a már megírt kód gyakorlatilag egy az egyben használható. A következőkben felsorolok néhány a projektem szempontjából releváns tulajdonságát: [3]

- Beépített FPU³

Segítségével egy ciklus alatt végezhető szimpla precizitású lebegőpontos műveletek (szorzás, összeadás).

- Maximum 180 MHz működési frekvencia

Ekkora működési frekvencia mikrokontrollerek esetén meglehetősen nagy, azonban problémákat vet fel.

- ART Accelerator™

A programkód a flash memóriában van tárolva, azonban az előző pontban említett sebességre nem képes ez a memóriatípus, ezen segít a fent említett egység. Gyakorlatilag egy cache, ami előre betölti a következő utasításokat, így a processzornak nem kell várnia a lassú flash memóriára.

- DSP utasításkészlet

Itt elsősorban a MAC (szorzás és összeadás egy ciklus alatt) illetve a SIMD (műveletvégzés több adaton egy időben) utasításokra kell gondolni.

- 2 MB Flash és 256 kB SRAM

Mind a flash (ahol a programkód mellett természetesen konstansok is tárolhatók) mind az SRAM tetemes mennyiségű. A flashban nyugodtan tárolhatunk fontkészleteket különböző szűrőegyütthatókat, a RAM-ban pedig viszonylag bonyolult jelfeldolgozási feladatokhoz is van elég hely. További sajátossága még a RAM-nak hogy tartalmaz egy 64 kByte-nyi területet (CCM⁴), amelyhez csak a processzor fér hozzá, azonban egy órajelciklus alatt. Ha a jelfeldolgozó algoritmust úgy írjuk meg, hogy a változók itt legyenek letárolva akkor komoly sebességnövekedéssel számolhatunk.

³ FPU: Floating Point Unit azaz lebegőpontos műveletvégző egység

⁴ Core Coupled Memory: e csak a CPU éri el, viszont nagyon gyorsan egy buszciklussal.

- Beépített memóriavezérlő (SRAM, SDRAM, NAND-, NOR Flash)

Ha a belső memória nem lenne elegendő, akkor különféle külső memóriákat is csatlakoztathatunk a mikrokontrollerhez. A választék nagyon széles, különböző Flash és RAM memóriák csatlakoztathatók. A memóriavezérlő nagyon kényelmessé teszi a külső memóriák használatát, mert felkonfigurálás után a külső memória a belső címbusz egy-egy címtartományán látszik, programozástechnikailag semmiben nem különbözik a belső memóriáktól. További előny még, hogy a ma kapható színes LCD kijelző modulok illeszthetők a memóriavezérlőhöz, így a kijelzés sebessége nagymértékben növelhető.

- Gazdag buszkészlet (SPI, I²S, I²C, SDIO, USB 2.0)

A szokásos SPI és I²C buszok mellett a kontroller rendelkezik I²S busszal is. Ez az audiotechnikában elterjedt busz, az audio AD illetve DA konverterek gyakran használnak ilyen interfészt. További előny még a teljesen integrált USB 2.0, illetve az SDIO interfész, mellyel SD kártyákat kezelhetünk.

- Fejlett buszmátrix, két darab DMA vezérlő

A kontrollernek két darab AHB⁵, kettő APB⁶ busza, illetve egy buszmátrixa van. A buszmátrix összeköttetést teremt az AHB mesterek (CPU, DMA, Ethernet, USB. stb.) és slave-ek között (SRAM1 SRAM2, Flash, FMC⁷, APB perifériák). A buszmátrixnak köszönhetően több nagysebességű master is működhet egy időben.

A mikrokontroller két DMA vezérlővel rendelkezik, segítségükkel gyakorlatilag bárholnan bárhova mozgathatunk adatokat a CPU közbeiktatása nélkül, a CCM memóriát kivéve, mert azt csak a CPU érheti el.

-TQFP 176 lábas tokozás, ebből 139 használható I/O

Tokozás tekintetében választhatunk a BGA illetve a TQFP tokozás mellett. A beforraszthatóság miatt, az utóbbit választottam. 176 kivezetésével ez a második

⁵ Advanced High-performance Bus nagy sebességű busz a processzor memóriák, és a DMA vezérlő csatlakozik ide.

⁶ Advanced Peripheral Bus: kisebb sebességű, mint az AHB a perifériák csatlakoznak ide.

⁷ Flexible Memory Controller: memóriavezérlő.

legnagyobb tokozás a szériában, bőséges I/O készletével pedig garantáltan be tudok huzalozni minden perifériát.

2.4.2 AD és DA konverter kiválasztása

A megfelelő konverterek kiválasztása nem egyszerű feladat, mert sok szempontot figyelembe véve kell meghozni a legjobb döntést.

A szempontok:

- magas bitszám,
- nagy dinamikartomány,
- max. mintavételi frekvencia: 40-200 kHz,
- a választott mikrokontrollerrel kompatibilis interfésszel rendelkezzen.

Ezek alapján két lehetőségem maradt: vagy $\Sigma\Delta$ AD és DA átalakítót használok vagy AD esetében SAR, DA esetében ellenálláslétra típusú átalakítót. Az előbbi mellett szól a magasabb bitszám, a kedvezőbb ár és a beépített átlapolásgátló szűrő, az utóbb mellett pedig kis késleltetés. Mivel néhány ms késleltetés nem probléma ebben az alkalmazásban, a beépített átlapolásgátló viszont kifejezetten előnyös, ezért mind AD mind pedig DA konverternek $\Sigma\Delta$ rendszerűt választottam. A keresést tehát ez irányban folytattam tovább. Próbáltam a beszerezhető legjobb paraméterekkel rendelkező típusokat választani, így a PCM4222 AD és a PCM1794A DA Texas Instruments gyártmányú konvertereket választottam. Ezek viszonylag egyszerűen beszerezhető példányok közül a legjobb paraméterekkel rendelkeznek.

A következő táblázatban bemutatom néhány paraméterét a választott konvertereknek.

1. táblázat Választott AD és DA konverterek néhány paramétere [4], [5]

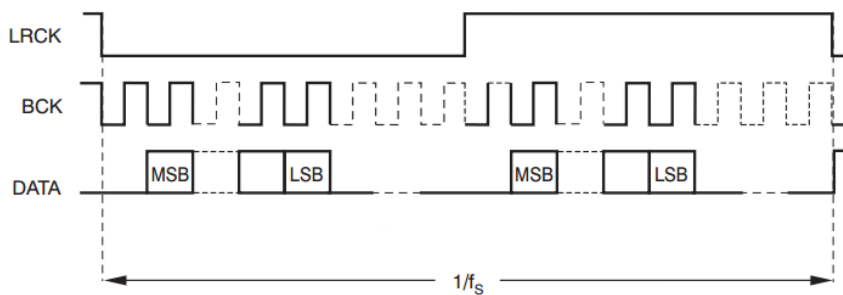
	PCM4222	PCM1794
Dinamikartomány [dB]	123	132
Felbontás [bit]	24	24
Analóg feszültség [V]	4	5
Interfész	I ² S, DSD, SAI	I ² S
Max. mintavételi frekvencia [kHz]	210	192
Be/kimenet	Differenciális	Differenciális

Az 1. táblázat mutatja az általam fontosnak vélt paramétereket. Az első és legfontosabb a dinamikartomány. Az elméleti maximum 24 bit esetén az elvileg elérhető dinamikartomány 144 dB lenne, azonban a különböző hibák miatt (nemlinearitás,

torzítások) ez mindig kisebb. A 123 dB-es érték már nagyon jónak mondható. Ezeknek a típusú átalakítóknak általában nagyon jó a linearitásuk a felépítésükből adódóan.

Az analóg feszültség a belső analóg áramkörök működéséhez szükséges feszültség. Ezen tápfeszültségek előállítására célszerű kiszajú stabilizátort beépíteni, mert erre a feszültségre ráülő zaj illetve zavarok nagymértékben befolyásolják az átalakítás minőségét. Sajnos a két konverter két különböző analóg feszültségről jár, tehát két külön stabilizátor szükséges, ami a költségeket növeli.

Mindkét átalakító rendelkezik I²S interfésszel amivel a mikrokontroller is. Ez a Phillips által a 80-as években kifejlesztett digitális sztereó audio jelek továbbítására használt egyszerű interfész.



7. ábra I²S interfész [4]

A 7. ábra mutatja az I²S busz idődiagramját. Az adatok 32 bites csomagokban érkeznek, az MSB jön először. Az adat az órajel felfutó élváltásakor érvényes. 24 bites a hasznos adat esetén a maradékot a protokoll szerint 0-kkal kell kiegészíteni. Az LRCK⁸ jelzi, hogy éppen melyik csatorna adata van a buszon (0 bal, 1 jobb csatorna). Az LRCK és a BCK⁹ között egy órajelnyi eltérés van.

Mindkét konverter viszonylag magas mintavételi frekvencián is képes üzemelni, bár az alkalmazásban a 48 kHz bőven elegendő.

Mindkét konverter analóg része differenciális. Ez hosszabb vezetéseken vitt jeleknél hasznos, mert a közös módusú zavarok kiküszöbölhetők ezáltal, esetemben viszont elő kell állítani a jelet ellenfázisban ami további pluszköltséget jelent.

⁸ LRCK Left Right Clock: jobb-bal csatorna órajel.

⁹ Bit Clock: bit órajel.

A jó AD és DA konverter tulajdonságai könnyen leronthatók rossz meghajtó fokozattal, ezért hasonlóan nagy hangsúlyt fektettem a jelkondicionáló fokozatok alkatrészeire is. Itt OPA2211, és OPA1604 típusú műveleti erősítőket alkalmaztam. Ez előbbi kettő az utóbbi 4 műveleti erősítőt tartalmaz egy tokban. Ezek a műveleti erősítők nagyon kis zajjal rendelkeznek, továbbá rail-to-rail be és kimenetűek.

2.4.3 Egyéb alkatrészek kiválasztása

Ebben az alfejezetben a kisebb nem annyira fontos alkatrészeket ismertetem.

Memória:

Ennél az eszköznél viszonylag egyszerű a választás, hiszen az SDRAM szabványos alkatrész. Egy 16 bites adatbusszal rendelkező ISIS gyártmányú IS42S16400J típusú 8 Mbyteos SDRAM került a panelre, végül azonban logisztikai problémák miatt egy ALLIANCE gyártmányú AS4C8M16S-6TIN típusú 16 MByteos került beépítésre. Hála a lábkompatibilitásnak az eszköz megfelelően működött.

LCD kijelző:

Egy 4.3 inches színes LCD kijelzőre esett a választásom. Ennek a mérete éppen megfelelő, egy rádióon kijelzendő összes információ olvashatóan elfér rajta. Adatbusza az ilyen kijelzőknél megszokott 8080-as processzorral kompatibilis busz, aminek kezelésére a processzor memóriavezérlője is alkalmas, így nagyon kényelmesen, kis processzoridővel írhatunk ki a kijelzőre. A kijelző tartalmaz egy rezisztív nyomásérzékelőt, és a hozzá tartozó vezérlő IC-t, illetve egy SDkártya foglalatot. Ezeket mind bekötöttem a mikrokontrollerbe a későbbi fejlesztési lehetőségek miatt.

USB tápfeszültség vezérlő és védelem:

Mivel OTG USB található a mikrokontrolleren ezért szükség van egy tápfeszültség felügyelő áramkörre. Ez az áramkör attól függően, hogy slave vagy host az eszközünk, kapcsolja a tápfeszültséget, host módban pedig gondoskodik, hogy az USB szabványban rögzített fél ampernél többel ne lehessen terhelni az áramkört. Erre a TPS2530 típusú áramkört választottam. Továbbá probléma még, hogy itt direktben vezetjük ki a mikrokontroller két kivezetését a külvilágba (D+ és D-). Ezek a kivezetések ki vannak téve mindennemű elektrosztatikus kisülésnek, amik akár tönkre is tehetik a kontrollert, ezért beépítettem túlfeszültségvédő áramkört, amit kifejezetten erre a célra ajánlanak. Az NXP gyártmányú PRTR5V0U2F típusra esett a választásom.

Komparátor:

Az amatőr rádióknál a gyártók próbálják utánozni a régi idők rádióit ahol egy forgókondenzátorral fokozatmentesen állították be a frekvenciát. Manapság kivétel nélkül szintézer a frekvenciaforrás, a forgókondenzátort pedig felváltotta a nagyfelbontású optikai enkóder. Egy ilyen enkóder azonban nagyon drága ezért más módszert választottam. A bipoláris léptetőmotorokat pontosan olyan jelalakkal kell vezérelni, mint amelyet az optikai enkóderek generálnak. Egyetlen probléma van a léptetőmotorokkal, hogy egy mágnesset forgatunk a tekercselt állórészben, így az indukált feszültség függ a forgatás sebességétől. Ezért szükség volt egy hiszterézises komparátorra, amely kis feszültségek esetén is megfelelő feszültséget ad. Választásom TLC272 típusú komparátorra esett. Ez a típus működik 3.3 V-ról így egy az egyben illeszthető a kimenete a mikrokontrollerhez.

Feszültségstabilizátorok:

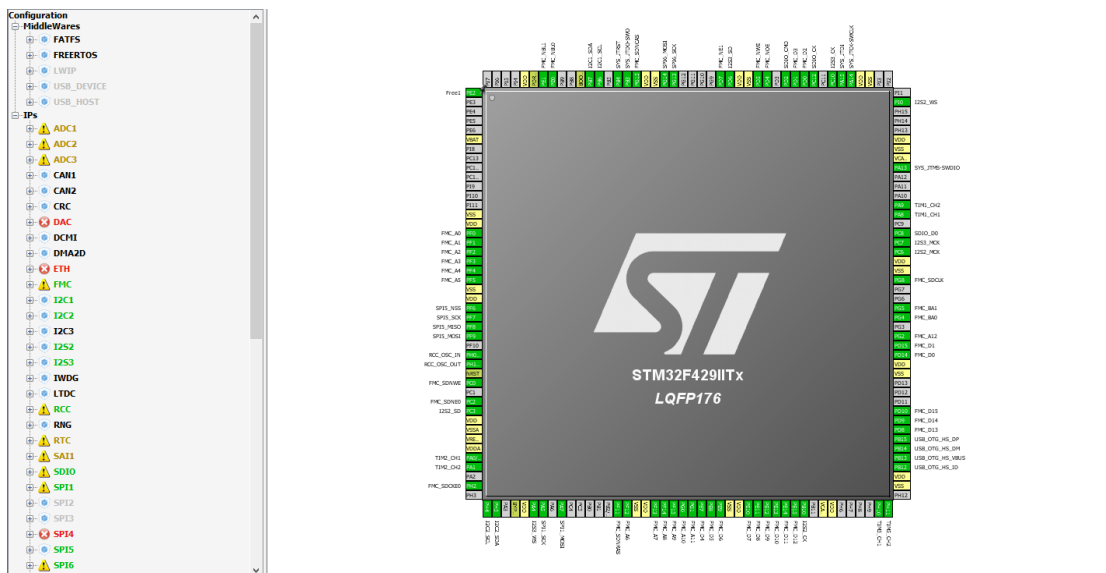
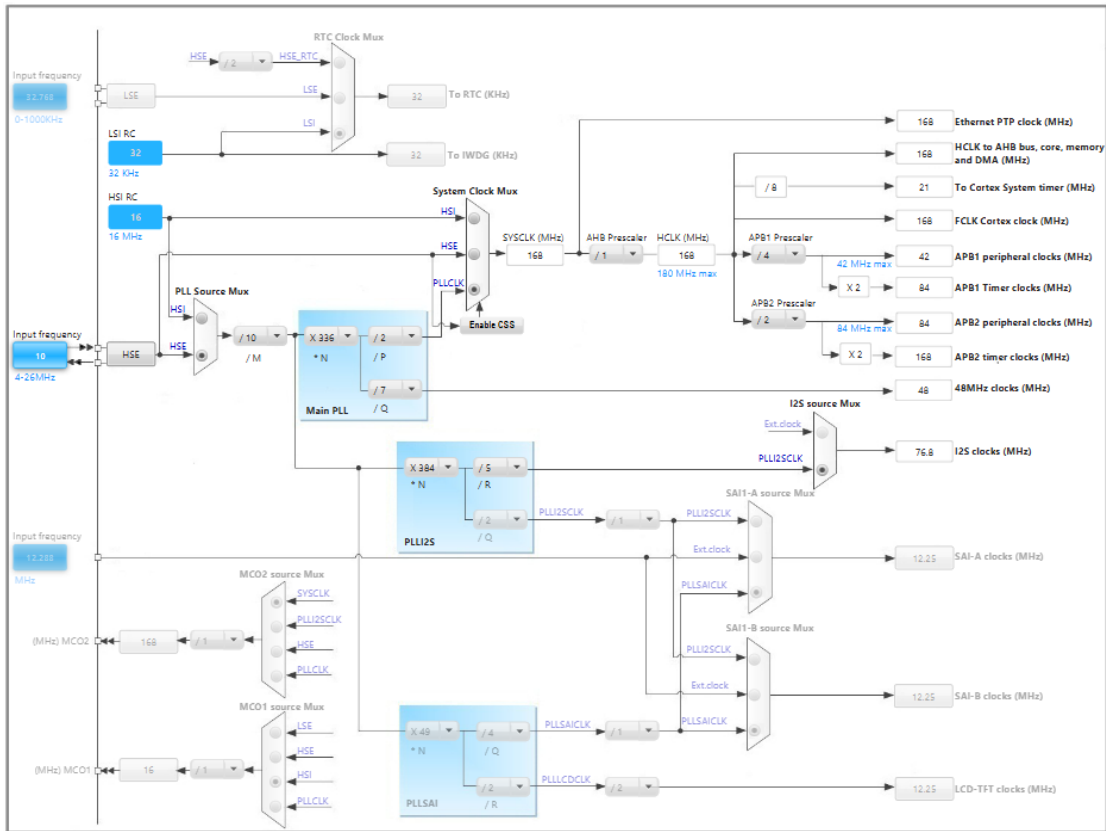
Az analóg áramkörök táplálására speciális kiszajú feszültségstabilizátorokat használtam. A pozitív oldalra a TPS7A4701 a negatívra pedig TPS7A3301 típusokat választottam. Ezek QFN tokozású stabilizátorok. Az előbbieik kimenőfeszültsége az egyes kivezetések földelése vagy szabadon hagyásával állítható be 0.1 V pontossággal. Az utóbbihoz egy feszültségosztó szükségeltetik. Fordított polaritás, túláram, és hőmegfutas ellen is védettek. A maximális kimenő áram 1 A lehet. A digitális áramkörök számára nem szükséges ilyen szintű stabilizálás, ezért ide a jóval olcsóbb LD1117 típusú fix 3.3 V-os feszültség stabilizátort használtam.

2.4.4 STM Cube

Bátran kijelenthető hogy, a választott mikrokontroller bonyolult, a klasszikus módszerekkel a kapcsolási rajz elkészítése viszonylag sok időt venne igénybe. Erre egyre több gyártó kínál megoldást különböző programok segítségével. Az ST Microelectronics az STM Cube nevű programmal rövidíti le a tervezési időt. Ebben a programban megadjuk a mikrokontroller típusát, ezt követően pedig beállíthatjuk a szükséges perifériákat. Ezeket a program a lehető legjobb elrendezésben elhelyezi nekünk a kontrolleren (sok periféria multiplexelhető több I/O-ra is, és több periféria esetén nem könnyű manuálisan megtalálni a legjobb lehetőséget).

A mikrokontrollernek elég bonyolult az órajelgeneráló rendszere. Számtalan osztót, illetve két PLL-t tartalmaz, hogy a processzormag, a buszok, illetve a perifériák

megfelelő órajellel legyenek ellátva. Ez beállítható az STMCube-ba felügyelettel, tehát rossz értékek esetén hibajelzést kapunk. További nagy előny még, hogy a beállításokból a program kódot generál, ami jó kiindulás lehet a fejlesztésnél.



8. ábra STM Cube órajelgeneráló PLL beállításai (fent) és a perifériák kiosztása (lent)

Itt sajnos azt tapasztaltam, hogy a generált kód más periféria könyvtárat használ, mint a régebben megírt kódok [2]. A cég honlapját tanulmányozva azt vettem észre, hogy

az új könyvtárat reklámozzák, tehát valószínűleg a régit nem fejlesztik tovább, csak az újat. Ezt sajnos el kell fogadni, és áttérni az új periféria könyvtárra minél előbb, mert a kódmennyiség növekedtével ez egyre nehezebb lesz.

Ezek alapján összeállítottam egy előzetes kapcsolási rajzot. Sok I/O volt amit csupán általános célra használtam (ADC DAC módjainak állítása, SPI slave select, stb.), ezek helye változott a nyomtatott áramkör tervezése során a minél könnyebb behuzalozhatóság érdekében.

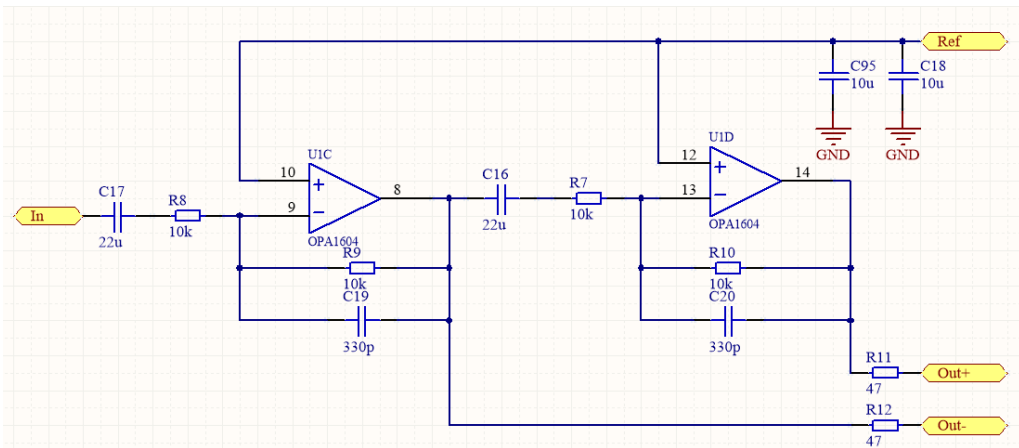
2.4.5 Kapcsolási rajz tervezése

A kiválasztott alkatrészek sematikus rajzát majd footprintjét megterveztem Altiumban. Ezt követően tanulmányoztam a kiválasztott alkatrészek adatlapjait, amik alapján elkészítettem a kapcsolási rajzokat. Az elkészült kapcsolási rajzokat az 50. ábra - 55. ábra mutatja. A digitális áramkörök esetén sok választási lehetőség nem volt, gyakorlatilag az azonos nevű vezetékeket kötöttem össze (az LCD kijelző kivétel volt). Ezen az STM Cube ajánlása nagyon sokat segített. Az analóg áramköröknél viszont sok időt elvett a megfelelő kapcsolat létrehozása. Ezek tipikusan az AD és DA konverter, illetve a komparátor jelének kondicionálása volt. Itt csak nem magától értetődő részletekre térek ki.

Általában az adatlapban ajánlott kapcsolásból indultam ki. Ezek a gyártó által kipróbált és ellenőrzött kapcsolások, tehát ezeket alkalmazva nagy valószínűséggel működni fog az áramkör. Egyes esetekben az alkatrészek értékén változtattam, máskor, indokolt esetben pedig a kapcsolást is módosítottam, ez esetben azonban szimuláltam a kapcsolást LTSpice-ban. Ennek ellenére előfordulhat, hogy a kapcsolat nem működik a valóságban, ennek az esélye azonban meglehetősen csekély.

AD konverter jelkondicionálás

Először az AD konverter jelkondicionálását mutatom be, amit részletesen a 9. ábra mutat.

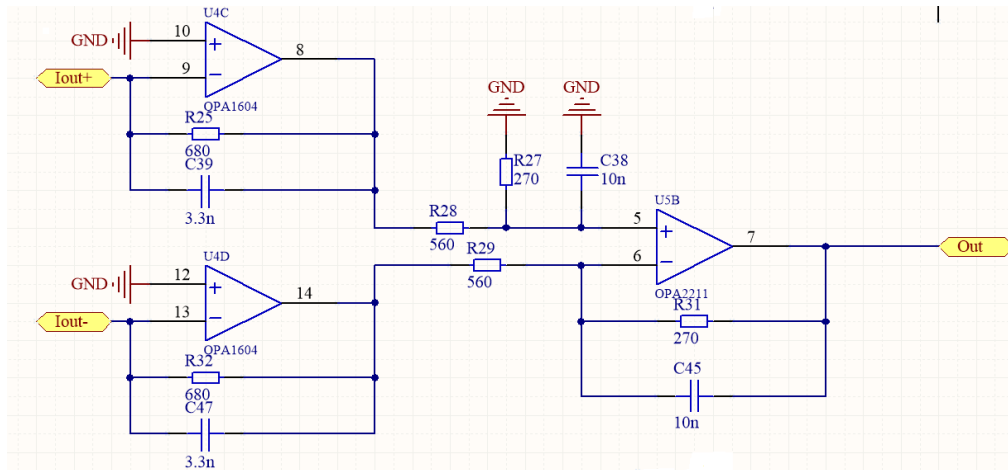


9. ábra AD konverter bemeneti jelkondicionáló áramkör

Az áramkör gyakorlatilag két darab invertáló kapcsolás egymás után kötve. A jelfeldolgozó egység előző verziójában [2] is ezt a kapcsolást alkalmaztam. A bemeneti szűrőkondenzátorokra azért van szükség, mert a műveleti erősítők invertáló bemenete a referenciafeszültségre kapcsolódik, így ha nem lenne kondenzátoros leválasztás, a kimenet kiülne a tápfeszültségre. A visszacsatoló ellenállással párhuzamosan található egy-egy kondenzátor. Ezek úgy vannak méretezve, hogy a párhuzamos RC kör vágási frekvenciája 48 kHz-re essen. Így bátran alkalmazhatunk 96 kHz-es mintavételi frekvenciát is. A bemeneti kondenzátor meglehetősen nagy, a vágási frekvencia 0.7 Hz-nél van. A gyártó előírja, hogy a referenciafeszültséget jól kell szűrni, erre szolgál a két darab 10 μ F-os kondenzátor. Ezek kerámiák, X7R típusúak, ennek köszönhetően nagyon széles frekvenciatartományban viselkednek kapacitásként.

DA konverter jelkondicionálás [5]

A következőkben a DA konverter segédáramkörét mutatom be. A konverter áramkimenetű, és 7.8 mA kimenő áram jelenik meg a kimenetére kötött ellenálláson teljes kivezérlés esetén. Az áramkimenet miatt a megszokottól eltérő kapcsolást kell alkalmazni, amelyet a 10. ábra mutat. Az áramkimenet munkaellenállása a műveleti erősítő visszacsatoló ellenállása is egyben, illetve ennek az ellenállásnak a nagyságával állítható be a teljes kivezérléshez tartozó feszültség is, ami esetemben körülbelül 4.3 V lett.



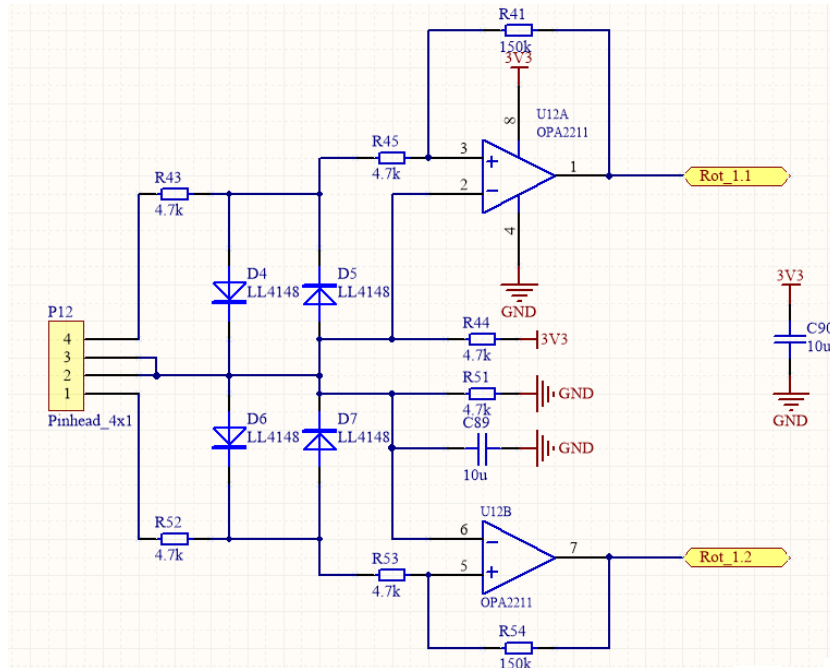
10. ábra DA konverter segédáramköre

A kimeneti jel szűréséről ez esetben is a visszacsatoló ellenállással párhuzamosan kapcsolt kondenzátorok gondoskodnak. Itt a vágási frekvencia 50 kHz-re adódott. Ezt a fokozatot egy különbségképző áramkör követi (vágási frekvencia 48 kHz), így alakul ki a kimenő jel. Ez a kapcsolás egy az egyben az adatlapban bemutatott, csupán az alkatrészek értékein változtattam, mert ott az ellenállások értéke számomra túl nagy kimenőfeszültséget eredményezett volna. Sajnos ebben az alkalmazásban a műveleti erősítők kettős tápfeszültségről üzemelnek, csak itt, sehol máshol az áramkörben. Erre azért van szükség, hogy az áramkimenet 0 alá is tudjon menni. Próbálkoztam különböző kapcsolásokkal, amivel elég csak egy tápfeszültség, de sajnos nem ismerem a chip belső felépítését, így nehéz bármit is tervezni köré. Végül a gyártóval történő egyeztetés eredményeképpen az adódott, hogy az szimpla tápfeszültséges megoldások jelentős mértékben csökkenti a konverter SNR-jét, ezért vagy használjam a gyári kapcsolást vagy használjak olyan konvertert, ami kiegészítő áramköre működtethető egyszeres tápfeszültségről. Átnéztem ezeket a konvertereket, azonban minőségi jellemzőik meg sem közelítik ennek a konverternek a paramétereit. Ezért maradtam ennél az IC-nél.

Enkóder áramkör

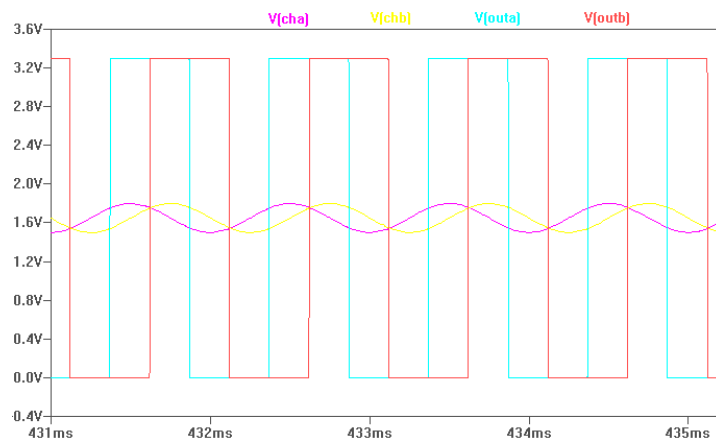
Következőkben a kvadratúra enkóderként használt léptetőmotor jelkondicionáló áramkörét mutatom be, melyet a 11. ábra mutat. A léptetőmotor a P12-es csatlakozóra van bekötve. A két tekercs egy-egy kivezetését összekötöm, és fél tápfeszültségre emelem. Erre a feszültségre kapcsolódik a két komparátor invertáló bemenete is. Az R43 és R52 a nagyobb áramok átfolyását gátolja a tekercseken, míg a diódák a feszültséget maximalizálják hozzávetőlegesen 0.7 V-ban. Ha nem lenne ott az előbb említett két

ellenállás, akkor nagyobb forgatási sebesség esetén a dióda gyakorlatilag söntölné a tekercseket, ami nem kis fékező hatással járna.



11. ábra Enkóder jelformáló áramköre

A komparátor visszacsatolása pozitív, ezt a kapcsolástechnikát alkalmazzák a hisztetézises komparátorok előállításához. Ezekkel az értékekkel nagyjából 150 mV a hiszterézis értéke. Ezt a kapcsolást szükségesnek éreztem szimulálni is, hogy megbizonyosodjam a működéséről.



12. ábra Az enkóder jelkondicionáló áramkör szimulációs eredménye

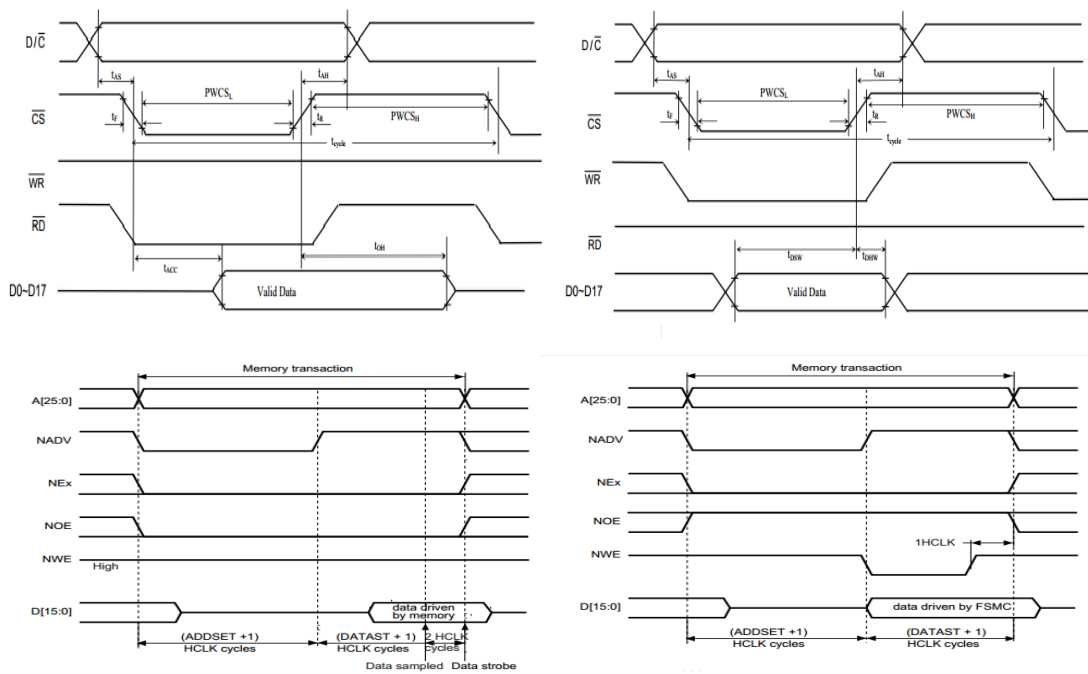
A szimuláció eredményeit a 12. ábra mutatja. A bemenőjel a V(cha) és V(chb) volt a kimenet pedig V(outa) és V(outb). A bemenetek egymáshoz képest 90°-kal eltolt 200 mVp-p szinuszos jelek voltak. Azért ennyi, mert a léptetőmotort kimértem

oszilloszkóppal, és nagyon lassú forgatásra is produkált ekkora jelszintet. A szimuláció meggyőző volt, ezért alkalmaztam ezt a kapcsolást.

LCD áramkör

Végezetül az LCD kijelző illesztését mutatom be. Ez nem egy szokványos illesztésnek mondható, mert a mikrokontroller memóriavezérlőjére csatlakozik az LCD modul, mégpedig úgy, mintha egy párhuzamos Flash memória lenne. Ez azért előnyös számomra, mert így a kijelzőre írni a programból csak egy adott memóriacímre történő írást jelent, a memóriavezérlő pedig elvégzi a komplett buszciklust. Ezáltal kevesebbet kell kódolni, és a kijelzés is gyorsabb lesz. A kijelző és az SDRAM közös adatbuszon helyezkedik el, és a kijelző viszonylag hosszú (~25 cm) szalagkábelon csatlakozik a processzor kártyára. Amikor az MCU az SDRAM-nal kommunikál, ott „lóg” az adatvonalon a kijelző kábele. Mivel a mikrokontroller az SDRAM-mal az AHB busz frekvenciájának felével képes kommunikálni, ami esetemben 90 MHz, a vezetéken olyan reflexiók jelennének meg, amik gyakorlatilag ellehetetlenítenék a kommunikációt. Ezért döntöttem buffer alkalmazása mellett, így amikor a kijelző nem aktív, akkor azt leválasztja a busztól.

Ezt követően a memóriavezérlő és a kijelző jeleit kell összeegyeztetni.



13. ábra LCD (felül) [7] és memóriavezérlő (alul) [6] olvasás (balra) és írás (jobbra) ciklus

Megvizsgáltam a memóriavezérlő buszciklusait az egyes módokban, és ezt összevettem az LCD által használt jelformákkal. Azt tapasztaltam, hogy NOR Flash módban megfeleltethetők a jelek egymásnak. Ezt a 13. ábra mutatja. A felső ábrák a kijelző az alsók a memóriavezérlő buszciklusai, a baloldali olvasás, a jobb pedig írás. A megfeleltetéseket a 2. táblázat tartalmazza.

2. táblázat LCD és memóriavezérlő jeleinek megfeleltetése

Funkció	Mikrokontroller	LCD
Adat/utasítás kiválasztás	A13	$\overline{D/\overline{C}}$
Chip select	NEx	\overline{CS}
Olvasás ciklus	NOE	\overline{RD}
Írás ciklus	NWE	\overline{WR}

A megfeleltetések a 13. ábra alapján könnyen beláthatók, egy kivétellel, ez pedig az adat/utasítás kiválasztás, ahová az egyik címvonalat vezettem. Ez azt eredményezi, hogy programban más címre kell hivatkozni utasítás illetve adat továbbítás esetén. Célszerű volt olyan címvezetékkel használni, amit az SDRAM nem használ, így elkerülhető volt a leválasztás. Én az A13-at választottam, ami azt eredményezte, hogy a programban a két funkció közötti címben 0x2000 offset van. A buffert ezek után már egyszerű volt betervezni, mert a DIR-t az \overline{RD} az \overline{OE} -t pedig \overline{CS} kivezetésre kellett kötni.

Ezzel végére értem a magyarázatot érdemlő komponenseknek. A processzorpanel kapcsolási rajzai a Függelékben találhatóak (50. ábra - 55. ábra).

2.4.6 Nyomtatót áramkör tervezése

Itt csak néhány egyszerű szabályt kellett betartani. Mivel a panelon vannak analóg és digitális áramkörök is, fokozottan ügyelni kell az alkatrészek elhelyezésére és a huzalozásra.

A konvertereket és a hozzá tartozó áramköröket az alsó oldalra helyeztem el lehető legtávolabb a digitális részekről. Ehhez az oldalhoz közelebb eső belső réteg lett a föld réteg. E fölötti rétegen futnak a tápfeszültségek, végül legfelül a digitális áramkörök. Mivel a panelméret limitált volt (a 4 rétegű NYÁK nagyon drága takarékoskodni kellett a hellyel), ezért az SDRAM átkerült az alsó oldalra, de amennyire lehet távol tettem az analóg áramköröktől. A 14. ábra mutatja az elkészült processzorkártyát.



14. ábra Az elkészült processzor kártya

Általában ahogyan az lenni szokott egy prototípusnál – a legkörültekintőbb tervezés esetén is – kerülhetnek hibák a tervekbe. Ezen panel esetén két apróbb eltervezés volt: az LCD illetve a szintézer felé menő szalagkábel csatlakozó tükrözve van. Ezen a problémán szerencsére könnyen lehetett segíteni: nem felülről, hanem alulról forrasztottam be a tűskéket, így gyakorlatilag újra megtükröztem.

Előfordult még egy nagyon érdekes hiba. A kijelző nem tudta megjeleníteni az egymás után következő fekete fehér pixeleket. Néhány pixelt kirajzolt, utána pedig lefagyott. Más színekombinációnál a probléma nem jelentkezett. A hiba felderítése nem volt egyszerű, mert lehet hibás az LCD a processzorkártya vagy a működtető szoftver is.

A hibás működést a kijelző és a mikrokontroller közé beiktatott buffer nagyon nagy felfutási meredeksége okozta. Ha nem alkalmaztam volna buffert akkor egyszerűen programból megoldható lett volna a probléma, mert a mikrokontroller I/O-inak sebessége állítható. Mivel a fekete-fehér átmenetnél mind a 16 bit változik egy irányban, nagy

felfutási meredekséggel, ez a változás akkora lökést okozott a szalagkábelen, hogy a strobe jelek is átbillentek, és az LCD vezérlő valamilyen ismeretlen állapotba került. Fontos megjegyezni, hogy ez nem függ az órajelfrekvenciától, csak a felfutási meredekségtől. A problémát úgy orvosoltam, hogy az LCD kijelzón (mivel a processzor panelre nem fért oda) felvágtam mind a 16 adatvezeték fóliáját, és 220 Ω -os ellenállásokon keresztül vezettem a jeleket. Ez az ellenállás egy aluláteresztő szűrőt alkot az LCD vezérlő bemeneti kapacitásával (adatlap szerint 5pF), ezzel korlátozva a felfutási sebességet. Így a szűrő vágási frekvenciája hozzávetőlegesen 144MHz, a kijelzőre akár 20 MHz-es frekvenciával is küldhetők az adatok.

Más ismert hibája jelen pillanatban nincs az áramkörnek.

2.5 Szintézer panel

2.5.1 Bevezetés

A bemutatást a szintézer panellel folytatom. Általánosságban az oszcillátoroknál, szintézereknél három minőségi paramétert szokás megkülönböztetni [22]:

- Hosszú távú stabilitás

Ez alatt azt értjük, hogy az oszcillátor frekvenciája mennyire stabil az idő múlásával. Az idő itt jóval nagyobb léptékű, mint a kimenőjel periódusideje, órák, napok vagy még nagyobb időtartam alatt vizsgáljuk, hogyan változik a kimenő frekvencia. Tehát frekvencia driftet mérünk. Paraméter nem csak az idő lehet, hanem például a hőmérséklet is.

- Rövid távú stabilitás

Itt az időtartam jóval kisebb, mint az oszcillátor kimenőfrekvenciájának periódusideje. Ilyenkor az egyes periódusok közti fluktuációt, fázisugrásokat vizsgáljuk. Ezt időtartományban jitternek, frekvenciatartományban pedig fáziszajnak hívják, gyakorlatilag mondhatjuk azt is, hogy a jel tisztaságát vizsgáljuk itt.

- Frekvenciapontosság

Itt azt vizsgáljuk, hogy a névleges frekvenciától mennyiben tér el a kimenőfrekvencia. Ezen paraméterek határozzák meg egy oszcillátor minőségét. Általánosságban elmondható, hogy ha a paramétereket javítani kívánjuk, akkor az az egyszerűség és az alacsony ár rovására megy. Céлом megtalálni az optimumot, hogy viszonylag olcsón jó szintézert készítek.

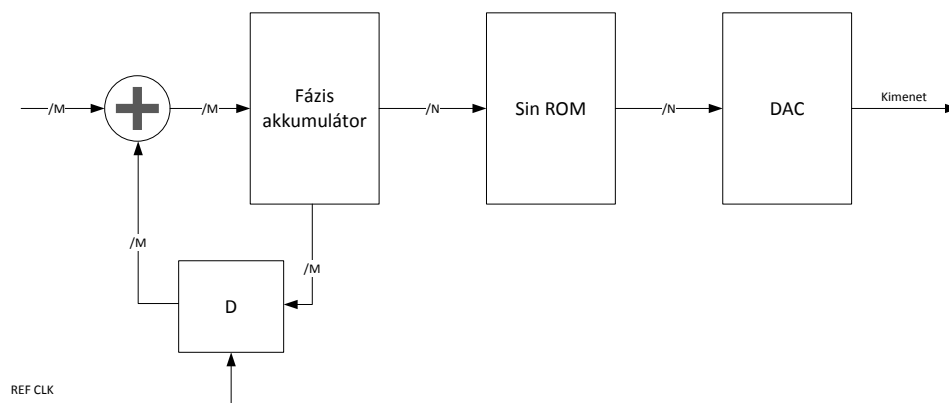
3. táblázat DDS és PLL struktúra összehasonlítása

	DDS	PLL
Felbontás	Nagy	Kicsi (FractN striktúrával jobb)
Sávszélesség	kicsi	nagy
Jel tisztasága	zavarjelek távol	zavarjelek közel
Ár	magas	alacsony
frekvenciaváltoztatás	gyors, sima	lassabb, glitches

Először el kellett dönteni, hogy milyen típusú legyen a szintézer. Két lehetőség kínálkozott, vagy DDS-t használjak vagy pedig PLL-t.

A 3. táblázat összefoglalja a PLL és DDS struktúra fő jellemzőit. Megjegyezném, hogy léteznek kevert struktúrák is, de ezeket az áruk és bonyolultságuknál fogva, illetve amiatt hogy a szintézerem előző verziója ilyen volt [8], nem részletezem.

A DDS¹⁰ leegyszerűsítve, egy fázisakkumulátorból egy ROM-ból (amibe a szinuszfüggvény van beprogramozva), és egy nagyon gyors DAC-ból áll. Blokkvázlatát a 15. ábra mutatja. Az akkumulátorhoz minden ciklusban hozzáadódik a beállított frekvencia értéke, majd a keletkezett (esetleg csonkolt) érték szolgál címként a ROM számára. A ROM-ból kiolvasott adat a DAC-ra kerül. Ez viszonylag új keletű technika, mert nagyobb frekvenciák esetén nagyon gyors DA konverterre van szükség, ami körülbelül 30 éve még nem állt rendelkezésre. Az ilyen struktúra előnye hogy a frekvencia nagyon pontosan állítható, és a váltás közben nem tapasztalhatók glitchek mint egy PLL esetében. Ezért a DDS-sel direkt frekvencia illetve fázismoduláció állítható elő (a DAC referenciájának változtatásával akár amplitudomoduláció is). Mivel ez gyakorlatilag egy vezérelt DA konverter, ezért a mintavételi törvényt be kell tartani, így a kimenőfrekvencia nem lehet nagyobb, mint a referenciafrekvencia fele (egyéb technikai okok miatt a gyártók a 0.4-es értéket javasolják). Továbbá a DA konverterben található nulladrendű tartó miatt megjelennek a kimenőjel felharmonikusai is. Ezeket ki kell szűrni a jelből, ami pluszköltséggel jár. Egyébként a vivőjel kis környezete nagyon tiszta. Mivel nagyobb frekvenciákra gyorsabb DAC kell ezért a DDS-ek ára arányos a maximális kimenőfrekvenciájukkal.

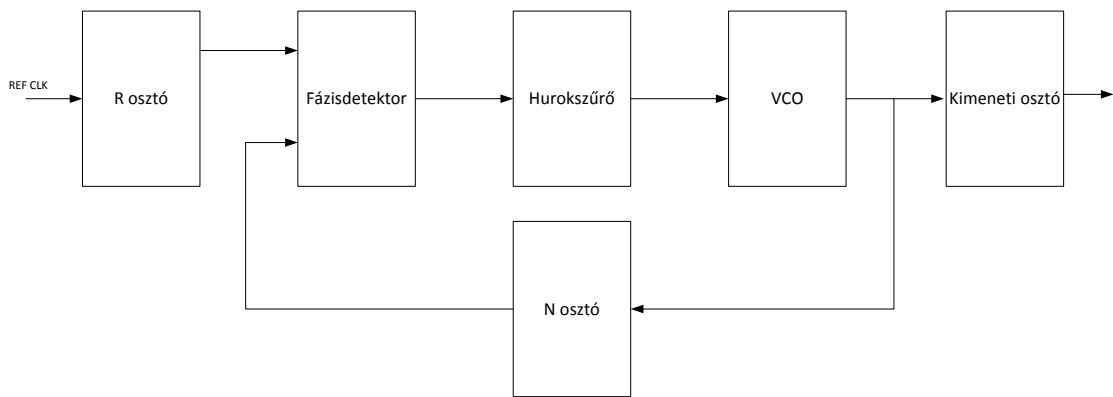


15. ábra DDS egyszerűsített blokkvázlata

¹⁰ Direct Digital Synthesizer

Ezzel szemben a PLL egy zárt szabályozási kör. Ennek következtében, ha változtatjuk a frekvenciát, akkor a beállítás ideje alatt különböző zavarok jelennek meg a kimeneten. Ez rádiós alkalmazásokban probléma, mert ilyenkor a kimeneti jel meghatározhatatlan.

Blokkvázlata a 16. ábrán látható. A bemenő frekvencia egy osztón keresztül a fázisdetektorra kerül. Ez az áramkör összehasonlítja a bemeneteire kerülő jelek fázisát, és a különbségükkel arányos hibajelét állít elő. A hibajel a hurokszűrőre kerül, ami kiszűri a nemkívánatos komponenseket.

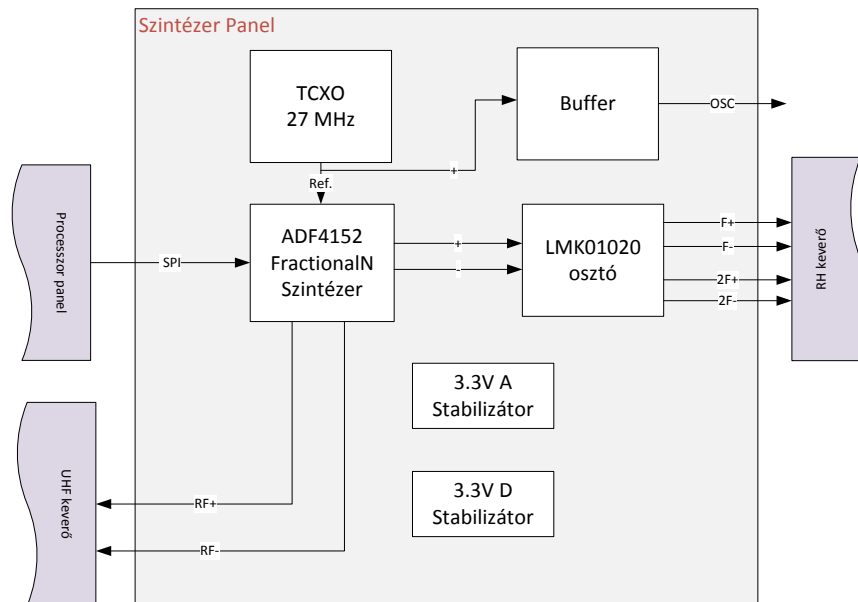


16. ábra PLL egyszerűsített blokkvázlata

A szűrt jel a VCO-ra kerül. Ez a bemeneti jellel arányos kimenő frekvenciát állít elő, amelyet leosztunk, majd visszacsatolunk a fázisdetektorra. A kimenőfrekvencia tehát a következőképpen alakul: $f_{ki} = f_{be} \frac{N}{R}$. Az N és R széles határok között mozoghat, tehát nagy frekvenciaátfogás érhető el, viszonylag olcsón. Az ilyen típusú PLL-eket egész osztásarányúnak, szakirodalomban csak IntegerN típusú PLL-nek nevezik. Ahhoz hogy a felbontás jó legyen belátható, hogy az R osztónak nagynak kell lennie. Ekkor azonban túlságosan leosztjuk a bemenőfrekvenciát ami a fázisdetektorra kerül. Ezt szorozza fel a PLL N-szeresére. Nem csak a frekvencia szorozódik fel, hanem vele együtt a zaj is. Ezért az IntegerN struktúrával nem érhető el nagy felbontás. Ennek feloldására találták ki a tört osztásarányú más néven FractionalN stuktúrát. Digitális osztóval csak egész számmal tudjuk leosztani a frekvenciát. Ebben a struktúrában két paraméterrel több van az M (modulus) és a F (fraction). Az F a számlálója az M pedig a nevezője a törtnek. A rendszer ilyenkor úgy működik, hogy az N számláló egy M hosszú ciklusban F-szer N+1-ig, majd M-F-szer N-ig számol. Ezzel átlagosan tényleg sikerült elérni a tört osztásarányt, azonban

a számláló technikailag két egész szám között ugrál. [9] Ez sajnos megjelenik a kimeneti jelben. Manapság ez a struktúra nagyon elterjedt, nagy felbontása és alacsony ára miatt.

A fent leírtak figyelembevételével elhatároztam, hogy tört osztásarányú PLL-lel fogom megvalósítani a szintézert.



17. ábra A szintézer kártya logikai felépítése

Az áramkör blokkvázlatát a 17. ábra mutatja.

2.5.2 Alkatrészek kiválasztása

Mint az előző fejezetben, így itt is az alkatrészek kiválasztásának kezdtem.

Először a szintézer IC-t választottam ki. Itt a céloom az volt, hogy egy chipben lehetőleg a PLL minél több alkatrésze benne legyen. A hurokszűrő és a VCO az a két részegység, amit nem szoktak a gyártók integrálni. A hurokszűrőt mindig az adott alkalmazáshoz kell hangolni, igazítani, másrészt pedig nagy értékű kapacitásokat ellenállásokat tartalmaz, ami sok helyet foglalna a chipen. A hurokszűrő fix részeit, amik nem foglalnak, sok helyet általában integrálják (tipikusan az erősítőt). A VCO esetében is szilíciumon elfoglalt hely lehet egy ok, amiért nem kerül a chipbe az áramkör. Tipikusan a nagyobb frekvenciájú illetve szélessávú szintézerek azok ahol integrálják a VCO-t. Én egy ilyen típusút választottam nevezetesen az Analog Devices által gyártott ADF4351 típusút. Néhány paraméterét bemutatom alább[10]:

- Integrált VCO, 2.2-4.4 GHz-ig hangolható, illetve ez tovább osztható 1-2-..64 osztó, mellyel az átfogható tartomány 35-4400 MHz-ig terjed
- maximum 4096 modulus érték
- $\Sigma\Delta$ modulátorral bővített tört osztásarányú architektúra: segítségével a zavarjelek távolabb kerülnek a vivőtől, oda ahol már a hurokszűrő is csillapít, opcionálisan ditherelés is beállítható (ekkor a zavarjelek nem tüskék formájában jelennek meg hanem, „szétkenődve”, mintha a zajszint emelkedne)
- két darab differenciális kimenet max. 5 dBm jelszinttel

A kimenő frekvencia minimuma (35 MHz) a rövidhullámú rádióamatőr sáv teteje, ezért a PLL egymagában nem képes feladatát ellátni. Erre a célra a TI gyártmányú LMK01020 típusú integrált áramkört választottam. Ez egy bemenettel és nyolc differenciális kimenettel rendelkezik, mindegyik kimenetre 1-510 frekvenciaosztás állítható be egymástól függetlenül. Ez nagyon hasznos az alkalmazott keverő miatt, erre a következő fejezetben részletesen kitérek. Ezt a konstrukciót már lehet használni a rövidhullámú sávok vételére is.

Egy eszközt kell még kiválasztani, mégpedig a frekvenciareferenciát. Gyakorlatilag ez az alkatrész szabja meg a hosszú távú stabilitást. Erre a célra egy 27 MHz-es TCXO-t választottam.

A szükséges feszültségstabilizátorok az előző fejezetben megismert TPS4A4700 típusúak lettek.

Beépítettem még egy órajelbuffert is, aminek bemenete a TCXO-ra, kimenete egy SMA csatlakozóra kapcsolódik. Így a referenciát más panelen is fel lehet használni (például a processzorkártyán a mikrokontroller órajelének). Órajelbuffernek a CDCLVC1102PWR típusút választottam.

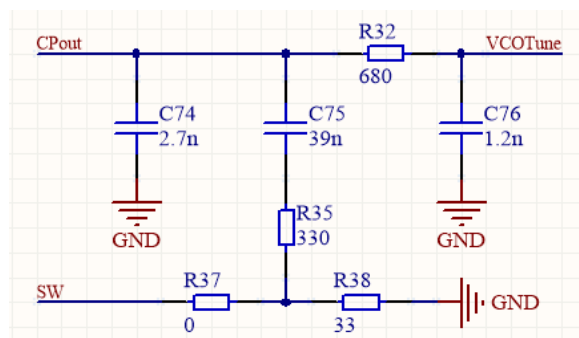
2.5.3 Kapcsolási rajz tervezése

A munkát az adatlapok alapos átolvasásával kezdtem. A PLL és az osztó adatlapja felhívja a figyelmet a tápfeszültség, és egyéb kivezetések megfelelő szűrésére. Mivel a VCO frekvencia nagy, kisebb kapacitás szükséges hogy a megfelelő frekvencián hidegítsenek a kondenzátorok. A kerámiakondenzátorok is rendelkeznek némi induktivitással, ami a VCO működési frekvenciáján (2.2-4.4 GHz) már nem

elhanyagolható. Ezen a tartományon a 10- 100 pF kapacitású NP0 típusú kondenzátorok hidegítenek a legjobban. A megfelelő kondenzátorok kiválasztása mellett a tápfeszültség hozzávezetésére is ügyelni kell, hogy minél kisebb legyen (ez is parazita induktivitást visz a rendszerbe), azonban az IC viszonylag kis tokozású ezért nagy alkatrészeket nem lehet megfelelően közel tenni. Ezért a lehető legkisebb, de még kézzel forrasztható 0402 méretű szűrőkondenzátorokra esett a választásom.

További feladat a hurokszűrő méretezése volt. A szűrő sáv szélessége befolyásolja a fáziszajt, ami a kimenet spektrumában jól látszik, erről bővebben beszámolok a mérési eredmények bemutatásánál. A nagy sáv szélesség nagyobb fáziszajt eredményez, de gyors beállást, a kis sáv szélesség jobb kimeneti paramétereket, viszont lassabb beállási időt ad. Mivel nekem mindkettő fontos (a lassú beállítás zavart okoz a kimenő jelben, a fáziszaj pedig a vételi minőséget rontja), ezért meg kell találni az optimális megoldást. A PLL elég fejlett és ezt hardveresen is támogatja. Beállítható egy ún. fast lock mód. Ennek lényege az, hogy található benne egy kivezetés, amit a fáziszárt állapot elérése után földre húz az áramkör. Így olyan szűrőt kell készíteni amelyben ha az egyik pontját földre húzzuk, a sáv szélesség jelentősen csökken. Így biztosítható a nagy sáv szélesség a beálláshoz, illetve a kisebb a fáziszárt állapothoz. A hurokszűrő méretezését az adatlap alapján az ADISimPLL nevezetű programmal végeztem el. Ez a gyártó saját programja szintézereik méretezésére illetve szimulációjára.

Kezdetben a gyári referenciakártya értékeit használtam a hurokszűrőre, mert ez biztosan működőképes. Mivel a kapcsolás nem változik csak az alkatrész értékek ezért az optimalizálni az elkészült panelen is lehet. A hurokszűrőt a 18. ábra mutatja.

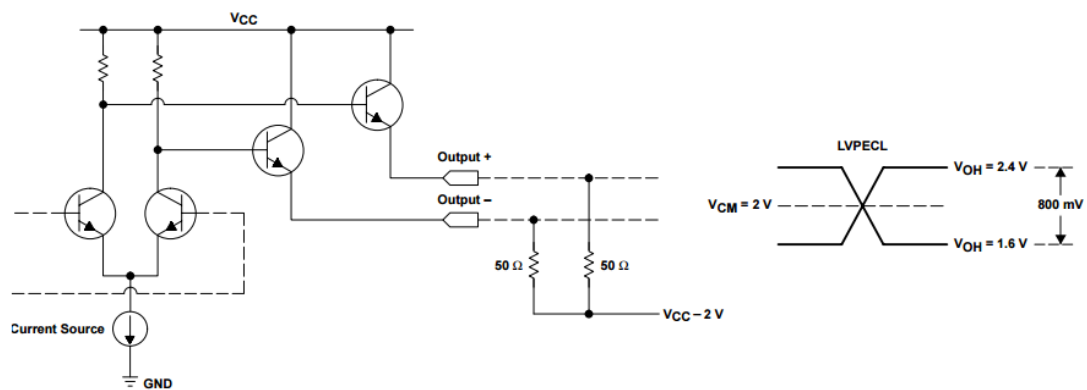


18. ábra A hurokszűrő

A következőkben a kimenetek lezárását mutatom be. Erre több megoldás kínálkozik. A kimenetre egy FET drain-je van kivezetve, ezt kell felhúzni tápfeszültségre a megfelelő működés érdekében. Itt használhatunk induktivitást, vagy ellenállást. Az induktív

előnye hogy nem disszipálódik rajta teljesítmény, cserébe csak egy adott tartományban elfogadható az általa képviselt impedancia, ezért ezt keskenyebb sávú alkalmazásokhoz ajánlják. Ha a kimeneti frekvenciát széles sávban kívánjuk változtatni, akkor az ellenállás az egyetlen megoldás, én is ezt használtam.

Az egyik kimenetet SMA csatlakozókra vezetem, innen kaphatja a jelet a későbbiekben megépítendő URH keverő, a másik kimenet pedig a programozható osztóra vezettem. Ez az áramkör saját magának állítja elő az előfeszültséget, ezért a bemenőjelet kondenzátorokkal le kell választani. Gondoskodni kell a megfelelő lezárásról is, amely jelen esetben $100\ \Omega$. Az osztó kimenetei LVPECL típusúak. A 19. ábra mutatja egy ilyen kimenet felépítését, illetve a szabvány által előírt jelszinteket is.



19. ábra LVPECL kimenet és a szabványos jelszintje [12]

A tápfeszültség (V_{CC}) rendszerint 3.3 V. Ebből kell előállítani 2 V-os feszültséget úgy, hogy az ellenállások single ended módban $50\ \Omega$ differenciális módban pedig $100\ \Omega$ -ot képviseljenek a kimenet felé. Megoldás, ha egy $82\ \Omega$ -ot a tápra egy $120\ \Omega$ -ot pedig a földre kötünk. Az ellenállások közös pontját a kimenetre kapcsoljuk. Így a fentebb írt minden követelmény teljesül [11].

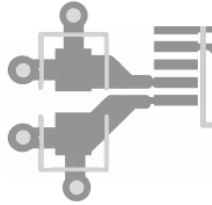
Az elkészült kapcsolási rajzokat az 56. ábra - 58. ábra mutatja.

2.5.4 Nyomtatott áramkör tervezés

Tervezésnél hasonlóan jártam el, mint a processzor panel esetében. A panel felső részén vezettem a nagyfrekvenciás jeleket, ehhez közelebbi belső réteg dedikált földréteg volt. Alatta futottak a tápfeszültségek, majd legalul az IC-k programozásához szükséges vezetékek. Az IC-k thermal padjei furatokon keresztül mind a föld réteghez vannak kötve

a jó hőelvezetést biztosítva. A PLL kimeneteit 100 Ω -os differenciális microstrip vonalon vezettem.

A PLL és az osztó hidegítő kondenzátorait közel tettem az IC-hez, és a belső rétegből a felsőbe a tápfeszültséget felhozó furatot is a 20. ábra szerint raktam le.

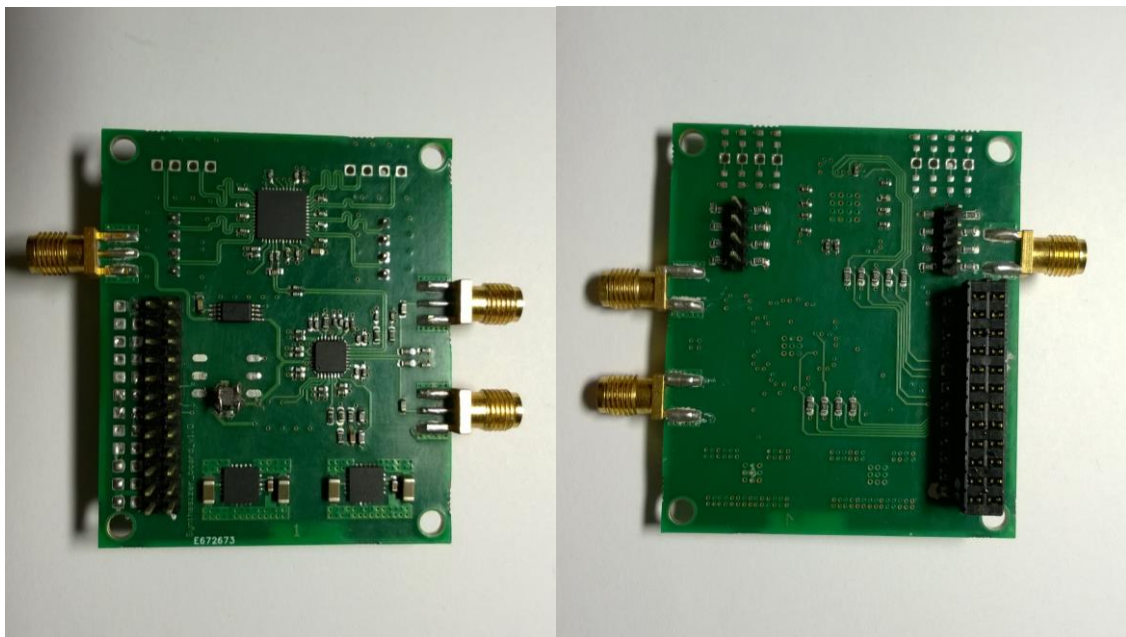


20. ábra Hidegítő kondenzátor helyes elhelyezése [13]

Összeszerelés és élesztés közben kiderült, hogy ezen a panelon is sikerült néhány hibát véteni:

- A referenciaoszillátort megtükrözve terveztem a panelre. Ezért a helyes működés érdekében „fejjel lefelé” kellett beforrasztani
- Az órajel buffer engedélyező kivezetését logikai magas helyett logikai alacsony szintre kötöttem, ehhez fóliát kellett vágni és egy vékony huzallal odavezetni a megfelelő feszültséget

Az elkészült áramkört a 21. ábra mutatja.



21. ábra Az elkészült szintézer panel

2.6 Keverő panel

2.6.1 Bevezetés

A keverő egy nemlineáris áramkör, ez valósítja meg a frekvencia transzponálást. Sokféle csoportosítás, és ezeken belül sokféle megvalósítás létezik. A teljesség igénye nélkül néhány csoportosítás:

- aktív vagy passzív,
- egyszeresen vagy kétszeresen kiegyenlített.

Azt hogy aktív vagy passzív a keverő, az attól függ, hogyan aránylik a bemenő RF jel nagysága a kimenő jelhez. Ha a keverő erősít, akkor aktív, ha csillapít, akkor passzív keverőről beszélünk.

A kiegyenlítés azt jelenti, hogy a keverőre kerülő jel, illetve az onnan elvezetett jel szimmetrikus-e. Ha csak az egyik szimmetrikus, akkor egyszeresen kiegyenlített keverőről beszélünk. Ennek előnye, hogy – attól függően melyik oldalon történik a szimmetrizálás – a kimeneten nem jelenik meg az LO vagy az RF jel. Kétszeres kiegyenlítés esetén egyik sem [20].

A keverőknek két bemeneti jele van: egyik a lekeverni kívánt jel (U_{be} ez általában nem periodikus), illetve a LO jele (U_{LO} ez periodikus). A keverő összeszorozza a két jelet, és ez jelenik meg a kimenetén. Az egyszerűség kedvéért a lekeverni kívánt jel is legyen periodikus.

$$U_{be} = \cos(2\pi f_{be}t) \quad (1)$$

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

A keverő összeszorozza a bemenetére kerülő két jelet, ami megjelenik a kimeneten, ez legyen U_{ki} .

$$\begin{aligned} U_{ki} &= U_{be}U_{LO} = \cos(2\pi f_{be}t)\cos(2\pi f_{LO}t) = \\ &= \frac{1}{2}\cos(2\pi[f_{be} + f_{LO}]t) + \frac{1}{2}\cos(2\pi[f_{be} - f_{LO}]t) \end{aligned} \quad (3)$$

A (3) egyenlet alapján látható, hogy a kimeneti jelben megjelenik a két frekvencia összege, illetve különbsége is. Mivel vevők esetében lekeverés történik

(nagyfrekvenciáról kisebbre), ezért az összegfrekvenciás komponens rendszerint kiszűrjük, csak a különbségit tartjuk meg. A különbségi frekvenciát középfrekvenciának nevezik (IF intermediate frequency). Ez a középfrekvencia akár lehet DC is ekkor a keverő direkt alapsávba transzponál.

Ez a jelenség azonban csak igazi szorzóáramkörökben valósul meg. Ilyen áramkörök azonban drágák, és limitált a működési frekvenciájuk is. Sokkal gyakoribb az a megvalósítás, amikor a keverőben valamiféle kapcsolóelem van, például dióda. Így gyakorlatilag mindegy, hogy az LO szinuszt, vagy négyszögjelet állít elő, mert miután a keverőben alkalmazott nemlineáris elem, például dióda kinyitott, a feszültség gyakorlatilag nem növekszik rajta, így a szinuszjel is úgy mond négyszögösedik. Az ilyen típusú keverőkre (a gyakorlatban szinte csak ilyet használnak), nem érvényes a (3) egyenletben definiált működés, mert nem szinuszos jellel, hanem négyszögjellel keverünk. Egy ilyen keverőjel a (4) egyenletben látható.

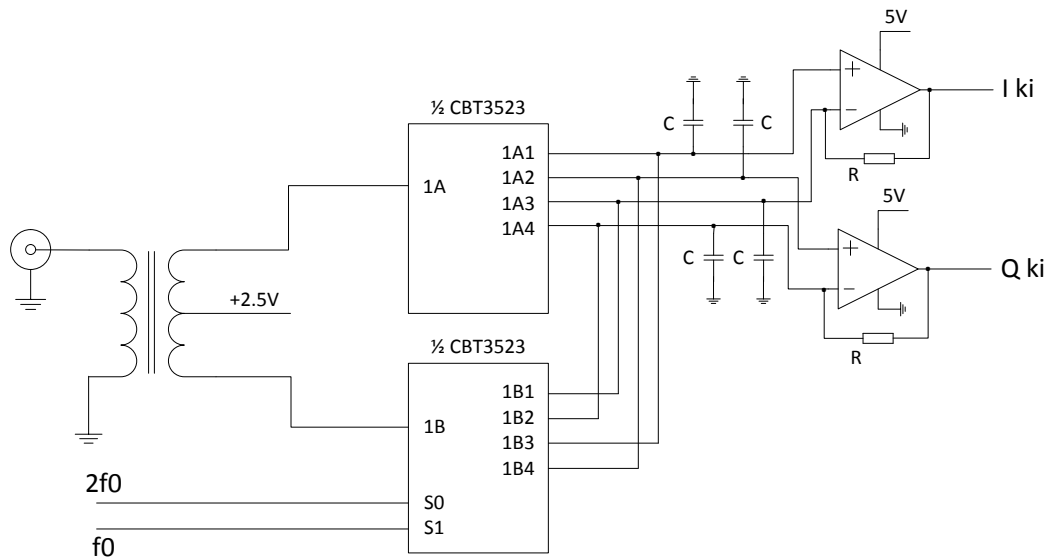
$$U_{LO2} = \cos(2\pi f_{LO}t) + \frac{1}{3}\cos(2\pi 3f_{LO}t) + \frac{1}{5}\cos(2\pi 5f_{LO}t) \dots \quad (4)$$

Így a keverő kimenetén a következő jelenik meg:

$$\begin{aligned} U_{ki2} = & \frac{1}{2}[\cos(2\pi[f_{be} + f_{LO}]t) + \cos(2\pi[f_{be} - f_{LO}]t)] + \\ & + \frac{1}{6}[\cos(2\pi[f_{be} + 3f_{LO}]t) + \cos(2\pi[f_{be} - 3f_{LO}]t)] \\ & + \frac{1}{10}[\cos(2\pi[f_{be} + 5f_{LO}]t) + \cos(2\pi[f_{be} - 5f_{LO}]t)] \dots \end{aligned} \quad (5)$$

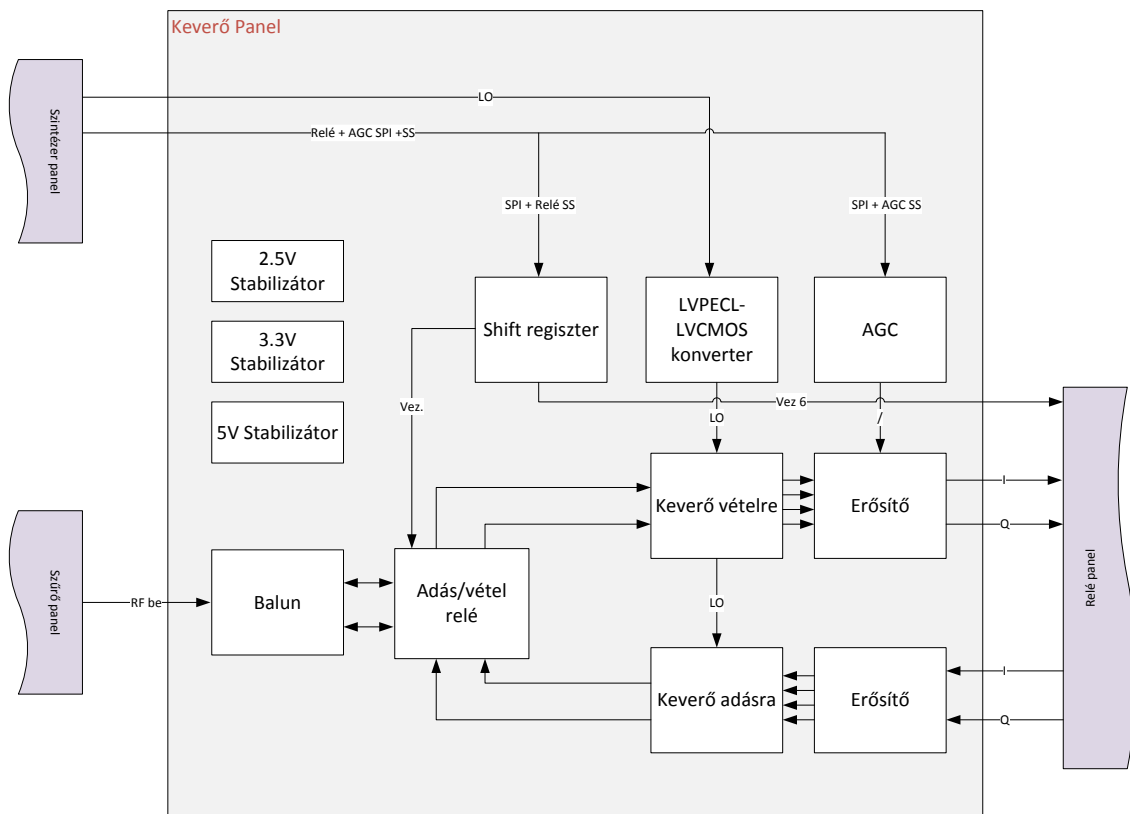
Látható hogy az (5). kifejezésben csak az összeg második tagja ($\cos(2\pi[f_{be} - f_{LO}]t$) hasznos, a többi vagy összegfrekvenciás, és/vagy a felharmonikusból származó komponens, melyeket alapsávban egyszerűen kiszűrhetünk. Ha a bemenőjel a (4)-hez hasonló, akkor a kimenőjel a két jelben található szinuszos komponensek lineáris kombinációjából adódik. Ekkor a keverő, a venni kívánt jel páratlan felharmonikusain található jelet is alapsávba transzponálja. Ezért kell gondoskodni arról, hogy ezek a jelek már a keverő előtt ki legyenek szűrve.

Keverőnek egy analóg multiplexert fogok használni. Ez egy kissé rendhagyó megoldásnak tűnhet. Ezt Dan Tayloe publikálta a 2000-es évek elején. Jobb név híján ezentúl csak Tayloe keverőnek fogom nevezni [14].



22. ábra Taylor keverő logikai felépítése [14]

Logikai felépítését a 22. ábra mutatja. A bejövő rádiófrekvenciás jelet szimmetrizáljuk egy transzformátor segítségével, így a keverő bemenő oldalról kiegyenlített lesz. Itt mindegyik tekercs egyforma menetszámú, tehát a kimeneten a feszültség kétszeres, az impedancia pedig négyszeres lesz. A transzformátor szekunder oldalának közös pontját felemeljük 2.5 V-os feszültségre. Erre azért van szükség, mert az analóg multiplexer 0-5 V-os tartományban fogad jelszintet. Negatív feszültség esetén kinyit a beépített védődioda. A transzformátor másik két kimenete a multiplexer egy-egy bemenetére kerül, az alacsonyabb kiválasztó bemenetére kétszeres, a magasabbra pedig egyszeres frekvenciájú keverőjel kerül. Ennek hatására a multiplexer az 1-2-3-4 sorrendben kapcsolja össze a bemeneteket a kimenetekkel. Itt célszerűbb lett volna Gray kódolást használni tehát 1-2-4-3 sorrendet tartani, mert így egyszerre csak egy bemenet változik, azonban a szintézerben található osztóval a szimpla illetve kétszeres frekvencia előállítás volt megoldható. A kimeneteket csoportosítjuk, és az összetartozók kerülnek a műveleti erősítő bemeneteire. Ez az oldal is láthatóan szimmetrikus, így egy kétszeresen kiegyenlített keverőről beszélünk. A multiplexer kimenetein található kondenzátorokkal a keverő sávszélességét állíthatjuk be, a műveleti erősítő visszacsatoló ágában lévő ellenállással pedig az erősítést állíthatjuk be. Belátható hogy a struktúra adás módban is működőképes.



23. ábra A keverő panel blokkvázlata

A 23. ábra mutatja a keverő blokkvázlatát. A következőkben az alkatrészek kiválasztását mutatom be.

2.6.2 Alkatrészek kiválasztása

Először a Balun transzformátor toroid gyűrűjét választottam ki. A tekercsek/transzformátorok a vizsgált frekvenciatartományban általában nem állnak rendelkezésre készen, el kell készíteni őket. Transzformátor célokra nagy permeabilitású anyagot kell választani. Nem szükséges túl nagy gyűrű, mert az átvitt teljesítmények viszonylag csekélyek (nem csak vételen, adás módban is). Így a választásom az FT23-43 típusú gyűrűre esett. Az első szám a méretet a második pedig az anyagot jelenti. Ez alapján ez egy 0.23 inch külső átmérőjű (kb. 6 mm), 43-as anyagú gyűrű. Ezt az anyagot $\mu_r = 800$ -as permeabilitás jellemzi a 0.5 – 30 MHz-es tartományban, ami nagyon kedvező. A gyűrű $A_L = 158 \text{ nH}/n^2$ fajlagos induktivitással rendelkezik. Az A_L egy vasmagra jellemző állandó, azt mutatja meg, hogy adott menetszám esetén mekkora induktivitású tekercset kapunk. Én 6 menetet tekertem a magra trifilárisan (három vezeték szorosan egymás mellett tekerve). Mivel a transzformátor átfogása több nagyságrend, ez az alacsonyabb frekvenciákon kevés lehet a nagyobb frekvenciákon viszont túl sok

(elsősorban a nagyobb menetszám miatt a tekercsek közti szórt kapacitás hatása egyre nagyobb mértékben jelentkezik).

A transzformátorról a jel az adás/vétel kapcsoló relére kerül. A relé kiválasztásánál nem csak ennek a panelnek a szempontjait kellett figyelembe venni, hanem a többiét is, ahol relét alkalmazok, ugyanis az előnyös, ha kisebb kompromisszumok árán, de sikerül mindenhová egy fajta relét használni, mert a nagyobb mennyiségek általában olcsóbbak, továbbá időt spórolok, ha a tervezőprogramban nem kell több alkatrészt felvinni.

Egyik fő szempont volt, hogy a relé megfelelő csillapítással rendelkezzen a használni kívánt frekvenciasávban, valamint az, hogy dupla morze érintkezőkkel rendelkezzen (a szimmetrikus jel miatt). Lehetőleg bistabil legyen, ami csak kapcsoláskor fogyaszt áramot, illetve legyen minél olcsóbb. A TE gyártmányú IM42 relé teljesíti ezeket a feltételeket. Szerettem volna árnyékolt relét beépíteni, de ezeknek az ára többszöröse a nem árnyékolt fajtáknak, ezért ezt a lehetőséget elvettem.

Ezt követően a multiplexert választottam ki. Itt igyekeztem minél gyorsabbat választani, hiszen ha gyorsan kapcsol, akkor kevesebb teljesítmény veszik el kapcsolás közben. A választott multiplexer a 74CB3Q3253 típusú lett. A CB3Q kapcsolócsalád sajátja, hogy 3.3V-os tápfeszültségről működnek, de a multiplexer be és kimenetein 5V-os is képesek kapcsolni. Ezt úgy érik el, hogy a kapcsoló FET gate-jét egy töltéspumpa segítségével a tápfeszültség fölé emelik [23]. Ez azért előnyös, mert így az erősítő működhet 5 V-ról, a multiplexer pedig 3.3 V-ról így az 5 V-os analóg tápvonalat nem terheli az esetleges zavarjelekkel.

A multiplexerek LCV MOS jeleket várnak a szintézer felől (lokáljel) viszont a szintézer LVPECL jelszinttel dolgozik. A két szint nem kompatibilis egymással, ezért illesztőre van szükség. A szintillesztést azért itt, és nem a szintézeren végeztem el, mert az LVPECL egy differenciális jel, ennél fogva a közös módusú zavarokra érzéketlen, míg az LVC MOS nem. A választott szintillesztő a TI gyártmányú SN65ELT23 lett. Ez két bemenetű konverter, így külön kellett egy a vételi és adási oldalra is.

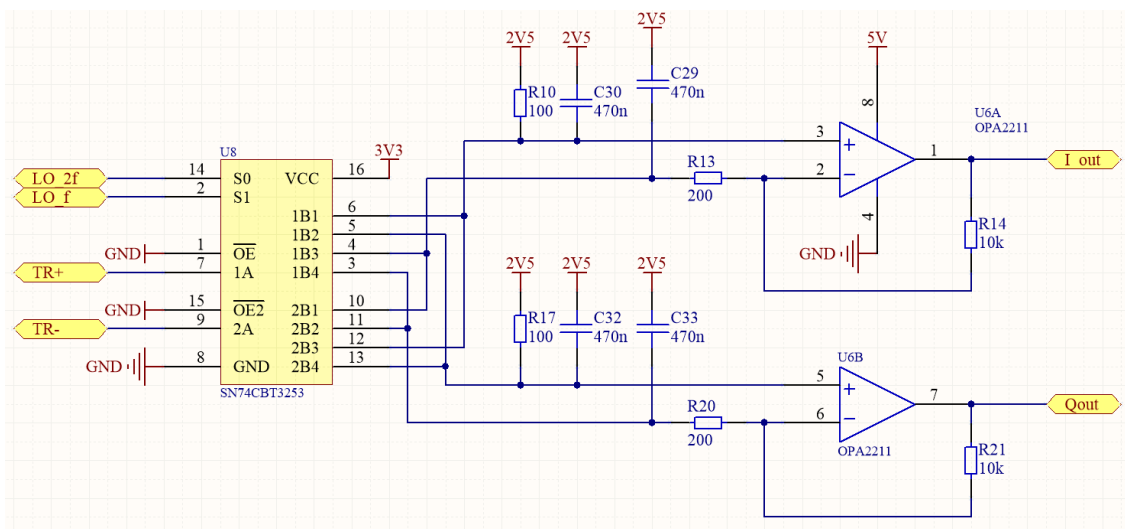
Műveleti erősítőknek a processzorkártya bemutatásakor ismertetett OPA2211 és OPA1604 típusokat használtam.

Az antennáról érkező jel dinamikatarományja meglehetősen nagy, így ahhoz hogy az AD konverterek bemenetére mindig optimális nagyságú jel kerüljön, szükséges változtatni a műveleti erősítők erősítését a bemenő jelszinttől függően. Erre egy sztereó digitális

potenciométert alkalmaztam, így az erősítés szoftveresen állítható gyakorlatilag bármilyen algoritmus használható az erősítésszabályozáshoz. A választott típus az AD5142. Ez egy 10 k Ω -os 256 lépésben állítható lineáris digitális potenciométer, amely SPI buszon keresztül programozható.

2.6.3 Kapcsolásirajz tervezése

Tayloe által közölt keverőt [14] némiképp átalakítottam. Az eredeti kapcsolásban az antenna nagy impedanciát látott a keverő túlóoldalán. Ez reflexiók kialakulásához vezet. Célom egy olyan kapcsolat kialakítása volt, melyben a keverő bármely állapotában az antenna felé a rendszer 50 Ω -ot mutat. Mivel az antenna közvetlenül egy középmegecsapolású transzformátorra kerül, aminek áttétele 1:2 a transzformátor pedig az impedanciát az áttétel négyzetével arányosan transzformálja, 200 Ω eredő ellenállást kell produkálni. A módosított áramkört a 24. ábra mutatja.

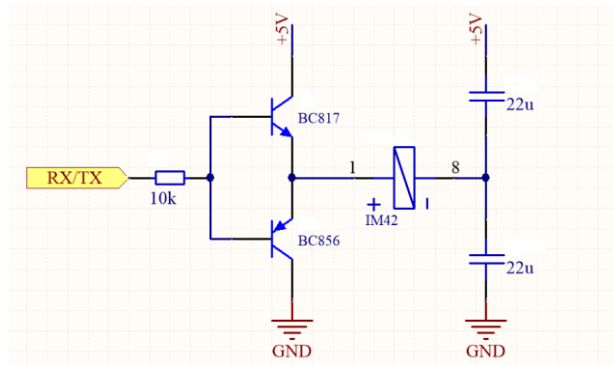


24. ábra Módosított keverő áramkör

Látható hogy a multiplexer mindegyik állásában a két bemenetére kerülő jelet (TR+ és TR-) rákapcsolja egyik műveleti erősítő invertáló illetve neminvertáló bemenetére, tehát e két pont között kell kialakítani a szükséges ellenállást. A neminvertáló bemenetet egy 100 Ω -os ellenálláson keresztül 2.5 V-ra kötöm (ne feledjük a transzformátor közös pontja is 2.5 V-ra van kötve). Az invertáló oldalon ez az ellenállás egyben az erősítés beállításában is részt vesz. Értékét azonban 200 Ω -ra választottam. Ennek az az oka, hogy a neminvertáló bemeneten lévő feszültség megjelenik az invertáló bemeneten is a feszültségkényszer miatt. A bemenő jel szimmetrikus az ellenállás két oldalán a jel ellenfázisban van, ezért az ellenálláson eső feszültség kétszeres. Ahhoz hogy

ugyanakkora áram folyjon, kétszeres ellenállásérték kell. Az R14 és R21 ellenállások helyett természetesen digitális potenciométer van a végleges kapcsolásban, csupán az átláthatóság kedvéért alkalmaztam itt fix ellenállást.

A következőkben a relé vezérlését mutatom be. A választott eszközt körülményes vezérelni, ugyanis egy tekercse van, és attól függően, hogy melyik irányba folyik rajta az áram vált állapotot. A gyártó ajánl hozzá egy kapcsolást [15], amit a 25. ábra mutat.



25. ábra A relé meghajtása

Itt egy NPN-PNP tranzisztorttal kapcsoljuk a tekercs egyik oldalát. A tekercsen keresztül az egyik kondenzátor elkezd töltődni (ha az alsó tranzisztor vezet, akkor a felső kondenzátor, és fordítva). A tekercsen átfolyó áram hatására átbillen a relé. A kondenzátorok nagyságával állítható be a csúcsáram. Ehhez a reléhez a gyártó 22 μ F-ot javasol.

A tranzisztorok vezérlésére shiftregisztert alkalmaztam. Így a mikrokontroller felől csak egy SPI (amit már amúgy is használtam több IC vezérlésére), illetve egy kiválasztó I/O szükséges. Ha egynél több relét kell kapcsolni, ez a megoldás kevesebb I/O-t igényel. Mivel a keverő panelon csak egy relé van, ezért a shiftregiszter többi kivezetését a relé panelre vezettem, ahol viszont 4 relé és egy hangfrekvenciás erősítő vezérlését kell megoldanom. További érv a shiftregiszter mellett, hogy adatlapja szerint 5 V-os tápfeszültség mellett képes működni 3.3 V-os rendszerben is. Mivel a relék működéséhez 5 V szükséges, ezért a tranzisztorok bemenetét is ekkora feszültséggel kell vezérelni a biztos ki/be kapcsolás érdekében, így szintillesztést is végez. Ez pusztán mikrokontrollerrel nem lenne megoldható.

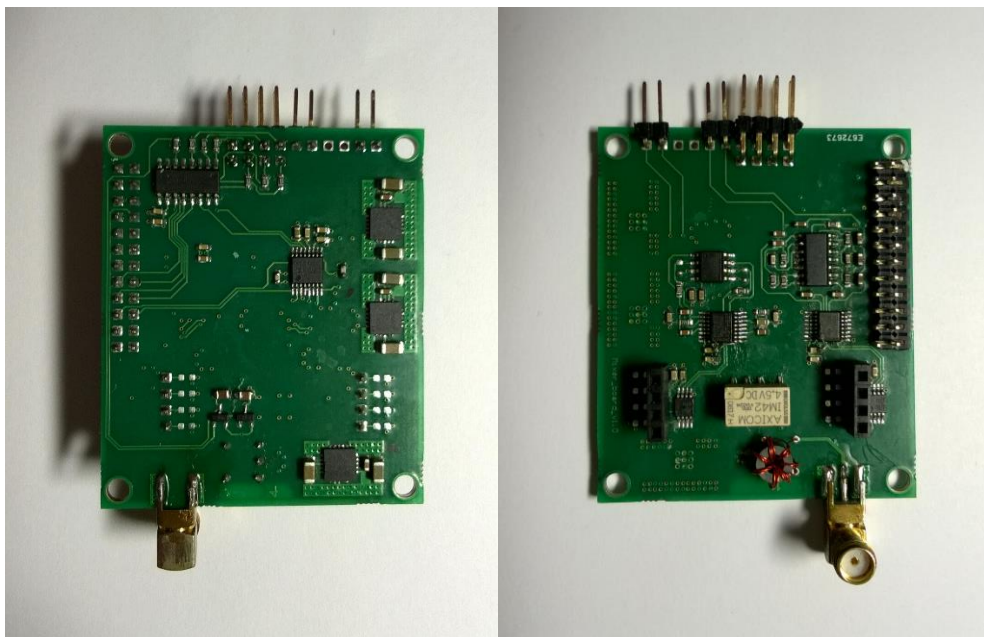
2.6.4 Nyomtattott áramkör tervezése

Ennél a panelnél a szintézer panelből indultam ki, mert ez a két panel „összedugható”, ezért minden egyes átvezetésnek pontosan passzolnia kell.

Hasonlóan az előző panelhoz itt is a felső rétegen vezettem az analóg jeleket, ehhez közelebbi belső réteg a föld, alatta lévő a tápfeszültség, végül az alsó rétegen a digitális jelek kerültek elvezetésre. A továbbiakban ugyan azokat az elveket követtem, mint az előzőekben ismertetett panelek esetén.

Sajnos itt is sikerült hibát véteni, sőt ezen a nyákon a legtöbbet és legnehezebben javíthatót. A szintézer panel felől jövő szalagkábel csatlakozót sikerült megtükrözni, így minden jelet keresztbe kellett vezetnem. Ezt a hibát úgy javítottam, hogy a kijövő tüskéket S alakban meghajlítottam, a szintézer panelon lévő hüvelysort pedig középre forrasztottam.

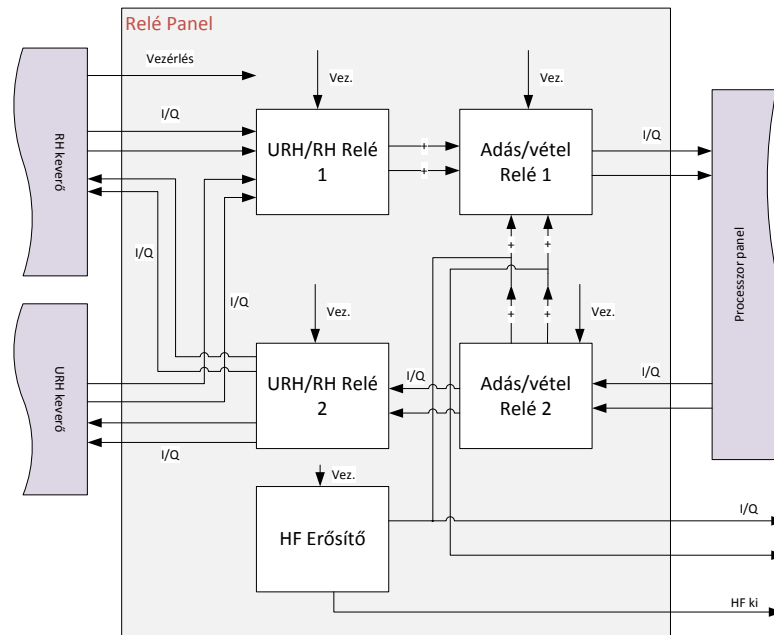
Továbbá sikerül az adó oldali keverőt meghajtó műveleti erősítő tápfeszültségeit felcserélni. Ezt fóliavágással majd átvezetékezéssel hoztam helyre. Az elkészült panelt a 26. ábra mutatja, a kapcsolási rajzokat pedig az 59. ábra - 60. ábra mutatja.



26. ábra Elkészült keverő panel

2.7 Relé panel

Ez a panel mindközül a legkisebb négy darab relét tartalmaz, a hozzá tartozó meghajtó áramköröket, illetve egy hangfrekvenciás erősítőt. Ez teremt kapcsolatot a keverők, a mikrofon és hangszóró, illetve a processzorkártya között. Az áramkör blokkvázlatát a 27. ábra mutatja.



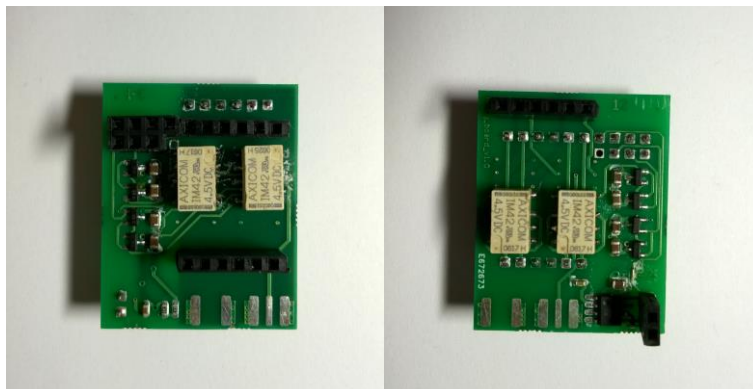
27. ábra Relé panel blokkvázlata

A panel úgy lett kialakítva, hogy van rajta hely az URH keverőnek is, így bővítés esetén ebből nem kell újat készíteni. E kis panel segítségével tehát kiválasztható, hogy RH vagy URH keverő jelét vezessük a processzorkártyára, illetve azt hogy adás vagy vételi módot is. Vételi módban a vevő keverőről érkező jelet digitalizáljuk, és a demodulált jel a DA konverterről a relé panel kimenetére kerül. Opcionálisan bekapcsolható egy hangfrekvenciás erősítő, ami egy kisméretű hangszóróra csatlakozik. Adás módban az AD konverterre mikrofon csatlakozik, a DA konverteren pedig megjelenik a modulált jel, ami az adó keverőre kerül. A reléket és az erősítőt a keverő panelon elhelyezett shift regiszter vezérli.

Erősítőnek az LM4875 típusú IC-t választottam. Ez maximum 1W kimenőteljesítmény leadására képes. Erősítése állítható, illetve egy engedélyező lábbal ki/be kapcsolható az egész áramkör.

A kapcsolás teljesen szokványos, már a dolgozatban ismertetett kapcsolástechnikai megoldásokat használ. A nyomtatott áramkör tervezésnél a csatlakozók pontos

elhelyezésére különös tekintettel voltam, hiszen ez a panel fogja össze az összes többi, ezért a csatlakozóknak pontosan a helyükön kell lenniük.



28. ábra Az elkészült relé panel

2.8 Szűrő panel

2.8.1 Bevezetés

A sávszűrők azok a passzív négypólusok, amelyek csak a kívánt frekvenciatartományt eresztik át. A 2.6.1 fejezetben bemutattam, hogy az alkalmazott keverőtípus a venni kívánt frekvencia páratlan harmonikusain lévő jeleket is alapsávba transzponálja, ez azonban kiküszöbölhető megfelelő sávszűrő alkalmazásával. További ok az használatukra, hogy kiszűrjük a sávon kívüli, erős műsorszóró adókat, amelyek miatt visszaveszi az AGC az erősítést, így a hasznos jelet nem erősítjük kellő mértékben, vagy ha az adó olyan erős, akár intermodulációt is okozhat a keverőben. Mivel vételen csak a szűrő van az antenna és a keverő között, ezért fontos hogy a hasznos jelet minél kevésbé csillapítsa, továbbá hogy a hőfokfüggése és az esetleges nemlinearitásai is a lehető legkisebbek legyenek.

Tervezésükhöz viszonylag sok irodalom és tervezőprogram áll rendelkezésre. A nehézség abban rejlik, hogy kiválasszuk a számunkra legmegfelelőbb topológiát és megvalósítási módot. Sok esetben a programok a gyakorlatban megvalósíthatatlan szűrőket generálnak, ezért ezt külön vizsgálni kell. További probléma, hogy ezeket a szűrőket a megtervezés után meg is kell építeni, és az alkatrészek szórása miatt be is kell hangolni. A beállítás időigényes, és drága műszer (hálózatanalizátor) szükséges hozzá.

2.8.2 Szűrőtervezés

A 40 m-es amatőrsávra történő szűrő megtervezését mutatom be. A többi sávra is hasonló megfontolások alapján történik a tervezés. Ez a sáv 7-7.2 MHz-s tartományon foglal helyet. Látható hogy a középfrekvenciához viszonyított sáv szélesség meglehetősen kicsi, így nagy meredekségű szűrőt kell tervezni.

A szűrőtervezésnél legelőször azt kell eldönteni, hogy milyen karakterisztikájú szűrőt akarunk alkalmazni. Három karakterisztika van, amit érdemes megemlíteni. Ezek:

- Butterworth (maximális lapos),
- Csebisev (ingadozás az áteresztő tartományban, de nagyobb meredekség),
- elliptikus vagy Cauer (ingadozás az áteresztő és záró tartományban is továbbá a legnagyobb meredekség) [21].

Annak ellenére, hogy a Cauer karakterisztika adja a legnagyobb meredekséget, az áteresztőtartományban okozott ingadozás nem jó tulajdonság. Ha ehhez hozzávesszük, hogy a megvalósítás miatt nem tudjuk pontosan előállítani a megtervezett szűrőt, ez további ingadozást vihet az áteresztő tartományba. Úgy döntöttem, hogy fontosabb hogy a szűrő egyenletes átvitelt produkáljon az áteresztőtartományban, mint a nagyobb meredekség, ezért választottam a Butterworth karakterisztikát.

A szűrő specifikációja:

- Áteresztőtartomány: 6.85-7.35MHz
- Csillapítás az áteresztőtartományban: max. -3dB
- Csillapítás a zárótartományban: legalább -40dB
- Be és kimeneti impedancia: 50 Ω
- Átmeneti tartomány: max. 1 MHz

Először kiválasztottam egy programot, amivel megtervezhetem a szűrőket. Választásom az AADE filter designer nevű programra esett. Viszonylag régi (2000-es évek eleje), az ingyenesen használható programok közül messze ez nyújtja a legtöbb szolgáltatást.

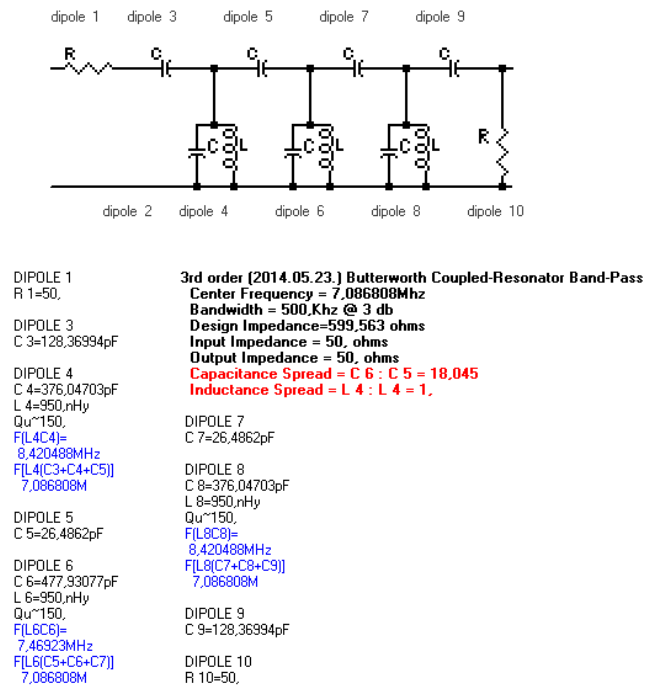
Ezt követően meghatároztam, hogy milyen struktúrában valósítom meg a szűrőt. A csatolt rezonátoros struktúrát választottam, mert ez a legalkalmasabb, ha a szűrő sáv szélességének és középfrekvenciájának aránya kisebb, mint 0.1. Ebben a szűrőben párhuzamos rezgőkörök találhatók, melyeket kondenzátorokkal csatoljuk ki be, illetve egymás között.

Meg kellett határozni, hogy hány rezgőkörös szűrőt fogok használni. Ha keveset, akkor a meredekség nem lesz megfelelő, ha sokat, akkor a szűrőt lehetetlenség lesz behangolni.

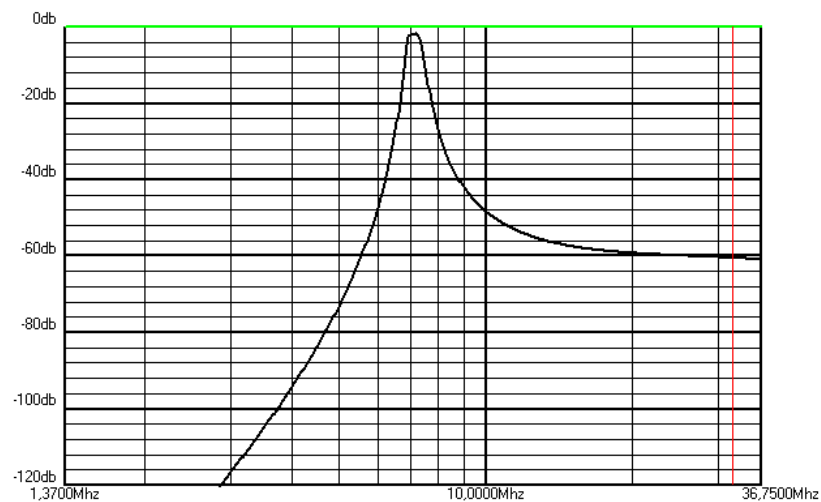
Kompromisszumként olyan szűrőt terveztem, amely három rezgőkört tartalmaz.

A szűrő kapcsolási rajzát illetve az alkatrészértékeket a 29. ábra a szűrő átvitelét pedig a 30. ábra mutatja. A kapcsolási rajzban található két ellenállás a be- és kimeneti impedanciát szimbolizálja. A tervezéskor az induktivitások értéke szabad paraméter volt. Az alkatrész kiválasztás és a tervezés itt egymással párhuzamos feladatok. Olyan induktivitás értéket kellett megadni, amely a kiválasztott vasmagon egész menetszámú, emellett a kiadódott kapacitásértékek közel vannak valamely szabványos értékhez (itt az

E12-es sorból válogattam). Néhány iteráció után megszületett a megoldás, láthatjuk, hogy a kapacitásértékek valóban közel vannak a szabványos értékekhez.



29. ábra A megtervezett szűrő kapcsolási rajza és alkatrészértékei



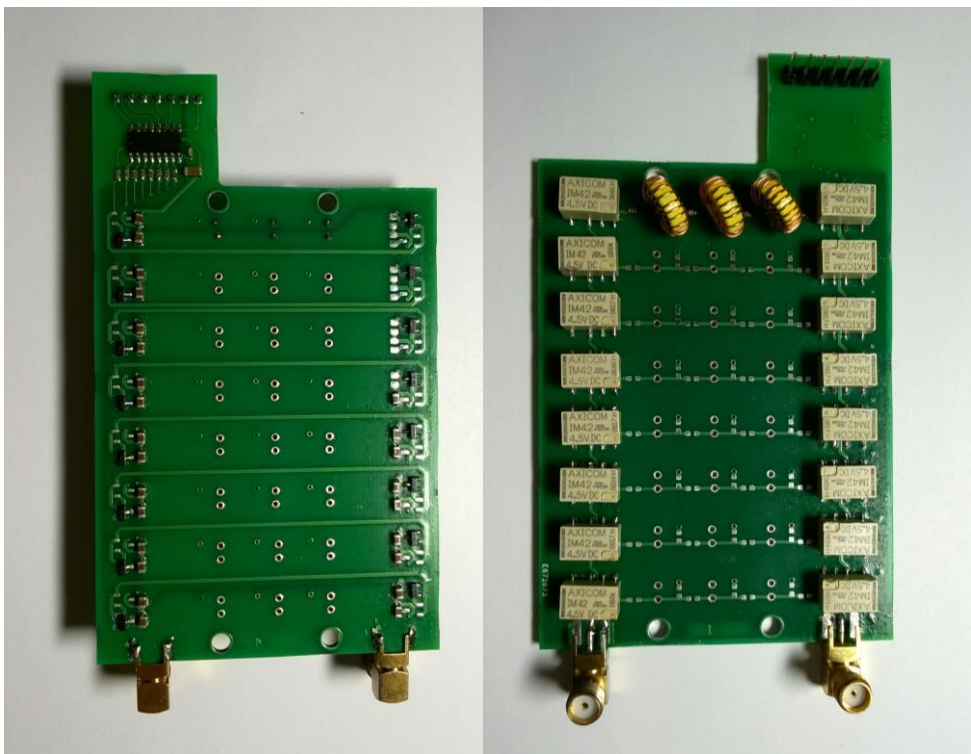
30. ábra A szűrő átvitele

Ezzel párhuzamosan válogattam az alkatrészeket. Az induktivitást T37-6 elnevezésű toroid gyűrűre tekertem. Az elnevezés itt is ugyan azt a szisztémát követi, mint a 2.6.2 fejezetben leírt Balun transzformátor esetében. A -6-os anyag kis permeabilitású a 3-40 MHz-es tartományban üzemel a legkisebb veszteséggel. Viszonylag nehezen megy

telítésbe, ezért erre az alkalmazásra ideális választás. A választott toroid mag $A_L=3 \text{ nH/n}^2$ értékkel rendelkezik. Ahol kis menetszám szükséges ott ezüstözött rézhuzalból tekertem meg az induktivitást ezzel is csökkentve a veszteségeket.

Kapacitásnak NP0 típusúakat választottam. Ez a fajta kis kapacitássűrűséggel, és magas árral rendelkezik. Kapacitását széles feszültség, frekvencia és hőmérséklet tartományban megtartja, továbbá öregedésük is elhanyagolható, ezért szűrőkbe kizárólag ilyen célszerű alkalmazni.

A szűrők kapcsolására a már 2.6.3 fejezetben ismertetett shift regisztert, relét, és relé meghajtó megoldást alkalmaztam. Az elkészült panelt 31. ábra mutatja.



31. ábra Az elkészült szűrő panel

3 Szoftverkomponensek tervezése

3.1 Bevezetés

A következőkben bemutatom a rádióhoz megírt programot. Először a hardverközeli, az egyes hardverelemek működését biztosító komponenseket ismertetem, majd bemutatok néhány jelfeldolgozó algoritmust, amelyeket implementáltam. Ahhoz hogy az algoritmusok gyorsan fussanak, itt is szükség volt a hardver (elsősorban a CPU belső felépítése) megismerése és működésének megértése.

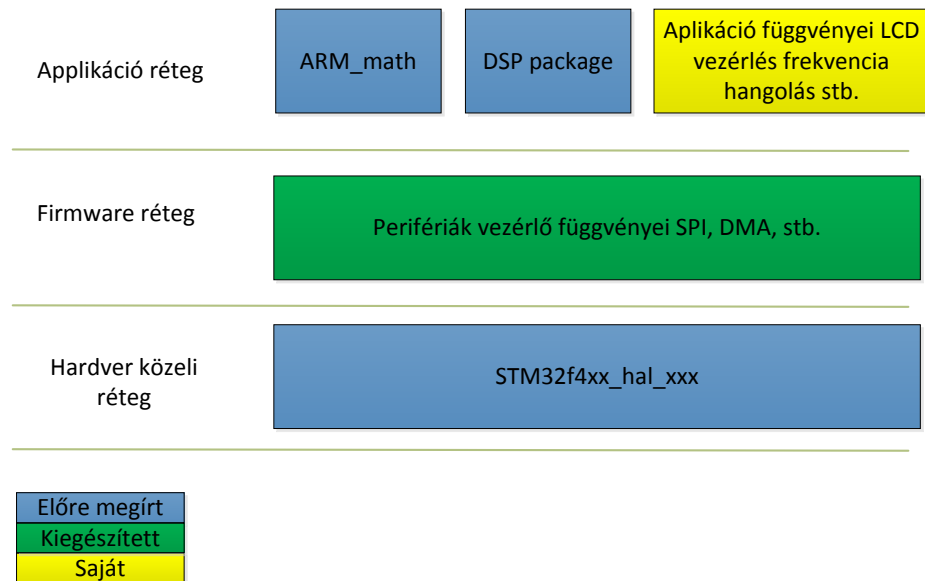
Már a 2.4.4 fejezetben bemutattam a mikrokontroller gyártójának a programját, amely arra hivatott, hogy meggyorsítsa a fejlesztést. Ez a program nem csupán a hardver, de szoftverfejlesztésben is segítséget nyújt. Azokból a perifériákból, melyeket beállítottuk a programban egy template-t generál, ami jó alapul szolgál a fejlesztéshez. Továbbá a gyártó nagyon sok példakódot is írt, ez némiképp ellensúlyozza a dokumentáció hiányát (a programkódból Doxygennel készített dokumentáció áll rendelkezésre, ami sokszor kevés). Eddig a gyártó egy firmware csomagot bocsátott a fejlesztők rendelkezésére, melyben a perifériák kezelése van implementálva. Ezt már nem támogatja, hanem egy HAL¹¹-t fejlesztett ki, és az STM Cube is ez alapján generál kódot.

3.2 Szoftverarchitektúra bemutatása

Már az elején látható volt, hogy a megírt program viszonylag nagy lesz, muszáj valami rendszert, rétegeket kialakítani, hogy a kód átlátható legyen. További érv emellett, hogy az STM Cube által generált kód lehet a rendszer alapja, és erre építhetők a felsőbb rétegek. A program által generált kódot érdemes felhasználni, annak ellenére hogy helyenként nem működött, mert ezen részek megértése és korrigálása kevesebb idővel jár, mintha mi magunk írtuk volna az egészet előről.

Beágyazott operációs rendszert nem használtam, mert enélkül maximálisan kihasználhatók a hardver lehetőségei, csinálhatunk olyat, ami egy beágyazott operációs rendszerben nem megengedett, viszont egy natív programban megállja a helyét. Továbbá nincs overhead, ami erős érv, mivel az elvégzendő feladatok nagyon számításigényesek.

¹¹ Hardware Abstraction Layer



32. ábra A rendszer szoftverarchitektúrája

A 32. ábra mutatja a rendszer szoftver architektúráját. Három réteget különböztetek meg, ezek a hardver közeli, a firmware, illetve applikáció réteg. A dobozok színe jelöli, hogy előre megírt, saját, vagy kiegészített komponensekről van szó. Törekedtem arra, hogy az egyes rétegek csak a felettük illetve alattuk lévő rétegekkel kommunikáljanak, illetve ne használjanak callback-eket. Ezáltal az egyes komponensek könnyen cserélhetők, és a kód más alkalmazásban is használható ahol hasonlóan van felépítve a szoftverarchitektúra. Ezt az elvet követni kicsit több munkával jár, de az átláthatóság, a flexibilitás és a kód újrahasznosíthatósága hosszú távon visszahozza a befektetett munkát.

A hardver közeli rétegben az STM Cube által használt függvénykönyvtárat soroltam. Ez a réteg gyakorlatilag letölthető a gyártó honlapjáról. Fontos szabály, hogy ezt csak használni lehet, módosítani azonban nem, mert ezzel sérülne a fentiekben leírt elv.

A HAL állományokban a perifériák konfigurálását, illetve kezelését leíró függvények (pl. SPI init, send, receive, DMA és FMC függvényei stb.) találhatóak.

E réteg felett találhatóak a perifériák vezérlő függvényei. Ez az a kód, amit az STM Cube generál az előzetesen beállított perifériákból. Ez egy félkész kód, ezt kell kiegészíteni, kijavítani, az applikációra szabni. Kivételesen az alsó rétegben található függvényeket hívjuk innen. Itt például több függvény található az SPI konfigurálására, külön-külön egy SPI küldés az integrált áramkörök vezérléséhez, itt található a memóriavezérlő felkonfigurálását tartalmazó függvény, illetve az SDRAM különböző módjait

implementáló függvények. Ez a réteg függ az elkészült hardvertől és hardvermódosítás esetén itt kell változtatni a programban.

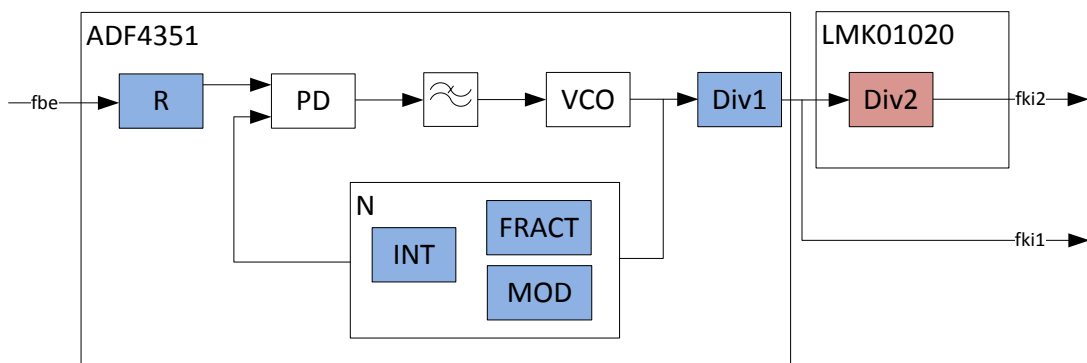
Az architektúra legtetőjén található az applikáció könyvtárai, illetve a matematikai és DSP könyvtár. Az applikáció függvényei teljesen függetlenek a hardvertől, a DSP és a matematikai függvények viszont nem. Az ok, amiért ide sorolom őket az az, hogy ezeket a könyvtárakat ezen a szinten használom. Ezeket a függvényeket nem is a mikrokontroller gyártója, hanem a kontrollerben található CPU fejlesztője írta meg (ARM Ltd.). Ezek a könyvtárak ezért némiképp kilógnak a sorból, bár hardverközeliek, de mégsem úgy, mint a HAL könyvtárak, ezért foglalnak helyet a legfelső rétegen. Például egy TI vagy NXP gyártmányú de CortexM4 magot tartalmazó mikrokontroller is pontosan ugyanezt használja.

3.3 Implementáció

Ebben a fejezetben bemutatok néhány részletet a megírt szoftverből. Itt megtalálhatóak mind az applikációs mind az alacsonyabb szintű komponensekből származó függvények. Azokat mutatom be melyek nem triviálisak, illetve valami probléma volt velük fejlesztés során. Sok kódrészletet fel lehetett használni az előző munkámból [2], ezeket nem részletezem itt.

3.3.1 Szintézer hangoló függvény

Ebben a fejezetben a szintézer felprogramozását és behangolását megvalósító függvényt mutatom be.



33. ábra A szintézer vázlatos felépítése

A 33. ábra mutatja a szintézer vázlatos felépítését. Megjelöltem rajta azokat a részegységeket, melyek a frekvencia beállításáért felelnek. Megjegyzem, hogy a konkrét alkalmazásban f_{ki2} kerül felhasználásra, f_{ki1} az URH keverő számára van fenntartva.

$$f_{ki2} = f_{be} \cdot \frac{(INT \cdot \frac{FRACT}{MOD})}{R \cdot DIV1 \cdot DIV2} \quad (6)$$

Ahol:

- f_{be} : Bemenő frekvencia (esetemben 27 MHz)
- INT: (Integer) PLL egészrész osztó (N osztó egészrész) (90-4096)
- FRACT: (Fractional): Törtrész számláló (1-MOD-1)
- MOD: (Modulus) Törtrész nevező (1-4096)
- R: R osztó
- DIV1: (Divider) Belső osztó ($2^{0...6}$)
- DIV2: külső osztó

A (6) összefüggésben látható, hogy a PLL milyen módon állítja elő a kimenő frekvenciát. Ezt az összefüggést kell gyakorlatilag leprogramozni, illetve ez alapján feltölteni a PLL regisztereit. Az erre megírt függvény a következő:

```
void SetFreq(uint64_t Freq, uint8_t RXTX, uint8_t MixerHFUHF, uint16_t *FreqError);
```

A függvénynek meg kell adni a frekvenciát Hz-ben (4.4 GHz és Hz-es felbontás esetén kifutunk a 32 bit ábrázolási tartományából ezért van szükség 64 bites paraméterre), majd azt hogy adás vagy vétel mód, illetve hogy RH vagy URH keverőt használjuk. A függvény ennek megfelelően kapcsolgatja ki vagy be az osztó és a szintézer kimeneteit. További paraméter még a frekvencia hiba. Itt egy címet kell megadnunk, és a függvény a címen található változóba írja a hibát. Erre azért van szükség, mert a PLL bár tört osztásarányú, de felbontása elmarad egy DDS-étől a magasabb rádióamatőr frekvenciákon (25-30 MHz), ezt a hibát a jelfeldolgozó algoritmusban korrigálni kell, ezt azonban csak a hiba ismeretében lehetséges.

Néhány változót fix értékűre vettem, ezek:

- R = 1; tehát a referencifrekvenciát nem osztjuk le.

- MOD = 4000; a törtrész nevezőjét 4000-re állítottam, ezáltal a frekvenciafelbontás 6.75 kHz lesz a 2.2-4.4 GHz-s tartományban. Az osztók ezt a felbontást tovább finomítják.

Az algoritmus először teszteli, hogy egyáltalán kell-e változtatni a paramétereken. Ha nem, akkor kilép bármiféle beavatkozás nélkül. Ha igen akkor végigfut egy if else szerkezeten, ahol megállapítja, hogy milyen osztót kell beállítani. A PLL minimum 35 MHz-re tud lemenni, ennél kisebb frekvencia csak az osztó IC-ről vehető le. Így megkaptuk DIV értéket. Ezt követően meghatározzuk INT értékét. Itt a (6) egyenletet átrendezve, és az osztásokat egészrészben végezve megkapjuk az eredményt.

Ezek ismeretében meghatározhatjuk a FRACT értéket. Itt már szükséges az egyes értékeket duplapontos formátumban kezelni a megfelelő pontosság érdekében. Ha megvan a FRACT érték is, ezt behelyettesítve a képletbe, majd kivonva a beállítandó frekvenciából megkapjuk a hibát.

A meghatározott értékeket az algoritmus betölti a PLL és az osztó megfelelő regisztereibe.

3.3.2 Memóriavezérlő konfigurálása, LCD, és SDRAM

LCD

Az LCD vezérlő felkonfigurálása után működőképes volt. A fekete-fehér pixelek egymás utáni megjelenítésével volt probléma, ami hardveres eredetű volt, ezt a 2.3.3 fejezetben bemutattam már.

Az LCD-hez használt memóriacímek a következők:

- Adat esetén: 0x60000000;
- Utasítás esetén: 0x60002000;

A konfigurálás után hozzávetőlegesen 20 MHz-es pixelfrekvenciával lehet a kijelzőre adatokat küldeni.

Az adatok küldése úgy történik, hogy készítünk egy pointert, ami a megadott címre mutat, és ennek adjuk értékül a kiírni kívánt adatot.

Egy ekkora méretű kijelzővel és a rendelkezésre álló hardveres támogatással gyakorlatilag mindent meg lehet jeleníteni, ami egy rádióhoz szükséges (pillanatnyi spektrumkép, illetve vízésés ábra), azonban most csak a működéshez legszükségesebb

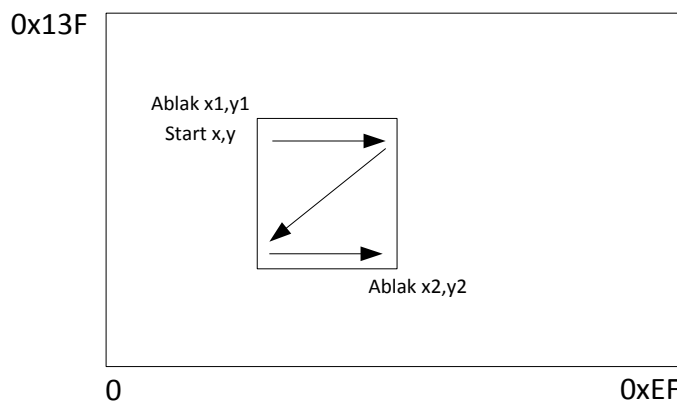
komponenst implementáltam, mégpedig az ASCII karakterek ábrázolását. Ennek segítségével kijelzhetővé válik a beállított frekvencia moduláció, stb.

```
void drawString(uint8_t* string, uint8_t font, uint16_t xpos, uint16_t ypos,
uint16_t Background, uint16_t Foreground)
```

A függvény bemenő paraméterei:

- string: A nullával lezárt karaktertömbre mutató pointer
- font: milyen fontot szeretnénk használni (4 féle van)
- xpos, ypos: a szöveg bal alsó sarka a kijelzőn melyik pixelre essen
- background, foreground: háttér, előtér színe

A függvény a legkönnyebben elkövethető hibákat kezeli (x, y pozíció túlcímzés, nem nullával lezárt karaktertömb stb.). Kihasználtam az LCD vezérlő adta lehetőséget, az ablakozást. Normál esetben úgy írunk a kijelzőre, hogy megadjuk a változtatni kívánt pixel x-y koordinátáját, majd a pixel színét. Így a kiírás nagyon hosszadalmas lenne. A másik módszer hogy megadjunk két koordinátát, ezek egy téglalapot jelölnek ki, majd a kezdőpontot és a pásztázási irányt. Ekkor csak adatokat kell írunk, és a vezérlő minden írás után inkrementálja a pixelcímet. A leírtak a 34. ábra mutatja. A kijelző az ábrán látható módon került beépítésre. Ez azért előnyös mert így ránézésre olyan, mint egy Descartes-koordináta rendszer, így könnyű tájékozódni benne. Ez a GUI fejlesztésnél nagy előny.



34. ábra Az LCD kijelző koordinátarendszere, és az ablakozás

Az adatokat a processzor írja ki a kijelzőre, ami ekkora órajelfrekvencián is komoly processzoridőt foglal. Hogy miért nem DMA-t használtam, azt a következő részben mutatom be.

SDRAM

A processzorkártyára viszonylag nagyméretű (16 Mbyte) SDRAM-ot terveztem. Itt tárolhatunk rögzített hangot, spektrumokat, kifejtett fontokat, szűrőegyütthatókat stb. A nagymennyiségű adathoz a DMA segítségével gyakorlatilag processzoridő felhasználása nélkül hozzáférhetünk. A memóriavezérlőt az STM Cube által generált kód helyesen felkonfigurálta. A memóriába beírt adat egy perc eltelte után is kiolvasható volt, tehát az automatikus frissítés is megfelelően működött.

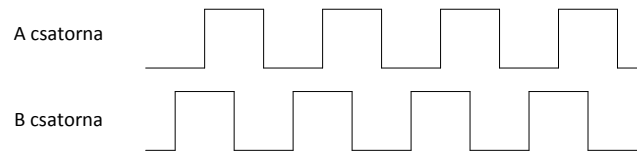
A célom az volt, hogy ebbe a memóriába eltárolom a karakterek képeit, és a szövegnek megfelelő karaktereket csupán átmásolom a kijelzőre. Ha mindezt DMA-val teszem, akkor jelentős processzoridő megtakarítás érhető el. Először próbaként nem DMA-val, hanem a processzoron keresztül próbáltam adatokat mozgatni az SDRAM-ból a kijelzőre. A kód hosszas hibakeresés ellenére sem működött. Végül a megoldást a mikrokontroller erratájában [16] találtam meg. A használt mikrokontroller chip revíziója „Y”. Ezekben a példányokban a memóriavezérlő nem képes magától „váltani” a dinamikus és statikus memóriák között, ezt magunknak kell, úgy hogy ha befejeztük az SDRAM használatát, akkor Self-refresh módba, ha pedig elkezdjük használni, akkor Normal módba kell tenni. Ezt a „3” jelű chip revízióba már kijavították. Tehát jelen állapotban nem használható DMA az SDRAM-ból a kijelzőre. Megoldható lenne, hogy az SDRAM-ból a mikrokontroller belső memóriájába, majd onnan a kijelzőre írjuk az adatot, azonban a hibakeresés időigényes volt, és inkább a fontosabb dolgokra koncentráltam, hiszen ez a rádió szempontjából nem kulcsfontosságú. A másik megoldás hogy kicserélem a mikrokontrollert a legfrissebb fajtára, ezt azonban azért vettem el, mert a mikrokontroller nagyméretű, meleg levegős forrasztóval nem lehet egyszerre mindenhol átmelegíteni, így a kiforrasztás során felszakadhat a rézfólia. Az SDRAM egyébként a 0xC0000000-0xC0FFFFFF címtartományon helyezkedik el.

3.3.3 Enkóder, SPI

Enkóder

A rádióhoz három darab enkóder csatlakozik, ebből az egyik egy léptetőmotor, amely a megfelelő jelkondíciónálással képes ellátni az enkóder feladatát. Az enkóder által generált jelformát a 35. ábra mutatja. Itt két egymáshoz képest 90 –kal eltolt négyyszögjel látható. Az élek számolásával az elfordulás nagyságát, a két jel egymáshoz képesti helyzetéből pedig az elfordulás iránya állapítható meg. Az előző verzióban szoftveresen

kezeltem az enkódereket. Ez a mostani verziónál azonban nem lehetséges. A használt léptetőmotorból készített enkóder 200 lépéses, 4-szeres mintavételnél 800 élváltás történik fordulatonként. Ha gyorsan tekerjük, akkor a sebessége 2-3 fordulat/perc is lehet. Ehhez nagyon gyakran kellene mintavételezni a vonalat (~10 kHz), ami lefogja a processzort.



35. ábra kvadratúra enkóder kimeneti jelformái

A szoftveres megoldás helyett hardvereset választottam, ugyanis a mikrokontroller Timer moduljai közül néhány képes fogadni és feldolgozni az enkóderek jelét. A konfiguráló kódot az STM Cube legenerálta a megadott adatokból. Problémaként adódott, hogy a generált kód nem 0-ra állította a számláló maximális értékét, így a modul nem számolt.

A beállított érték esetemben 0xFFFF lett, ebből jelen alkalmazásban nem futunk ki. A periféria jelzi a forgás irányát is, erre azonban nincs szükség, mert negatív irányban forgatáskor az elforgatott érték levonódik a regiszter tartalmából, így nagyon egyszerűen a változtatni kívánt regiszterhez egyszerűen csak hozzáadjuk a számláló értékét.

SPI

A szintézer és a keverőpanelen használt áramkörök (PLL, osztó, shiftregiszter, digitális potenciométer, stb.) egy közös SPI buszt használnak. Ezek az eszközök különböző SPI módokat használnak, ezért az összes IC-hez külön implementáltam a küldő függvényt. Ebben először beállítom az SPI perifériát a vezérelt eszköz által megkívánt módba, kapcsolom az adott eszközhöz tartozó SS (Slave Select) lábat, majd kiküldöm az adatot, végül az SS lábat logikai '1' be állítottam. A küldéshez a HAL-ban található függvényt használtam, aminek hibája, hogyha például 32 bitesek a slave regiszterjei, akkor a paraméterben kapott 32 bites típust fel kell bontani 8 bitesekre, és meg kell cserélni a sorrendjüket. Igaz ez többletmunkával járt, de nem sérült a 3.2 fejezetben bemutatott architektúra szabályrendszere.

3.3.4 AD/DA konverterek és a DMA

AD és DA konverter

Szándékosan hagytam a végére ezt a fejezetet, mert az itt leírtak kapcsolódnak majd az algoritmushoz.

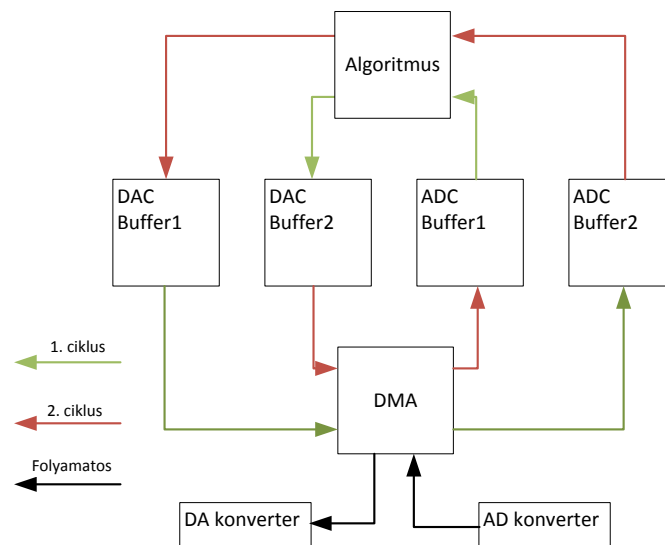
Mindkét konverteren a következő paramétereket állítottam be:

- Slave mód
- 48 kHz mintavételi frekvencia
- 24 bites felbontás
- Phillips I²S protokoll

Arra kellett figyelni, hogy mind a mikrokontroller perifériáiban, mind a konverteren is ugyan ezeket a paramétereket állítsuk be. A $\Sigma\Delta$ átalakítóknak a túlmintavételezett jelet digitális szűrők decimálják. Ezeknek szükségük van egy órajelre a működéshez. A mikrokontroller rendelkezik dedikált PLL-lel, amely előállítja ezt a frekvenciát. A mikrokontroller csak $256f_s$ előállítására képes.

DMA

Az adatmozgatást mind a két konverter esetében DMA-val végeztem. Kettős bufferelést alkalmaztam, amíg az egyik buffer párral a DMA dolgozik, a másikkal pedig a jelfeldolgozó algoritmus. Ezt a 36. ábra mutatja.



36. ábra Kettős bufferelés

A bufferek mérete minden esetben 1536, tehát 48 kHz-es mintavételi frekvencia esetén 30 ms-onként vált buffert a vezérlő, tehát az algoritmusnak, a kijelzésnek és minden egyébnek bele kell férnie ekkora időszelvénybe. A buffer hossza igazodik az algoritmus igényeihez, erre a következő fejezetben részletesen kitérek. A DMA cirkuláris, tehát ha befejezte az adatmozgatást az egyik bufferen, automatikusan a másikra vált.

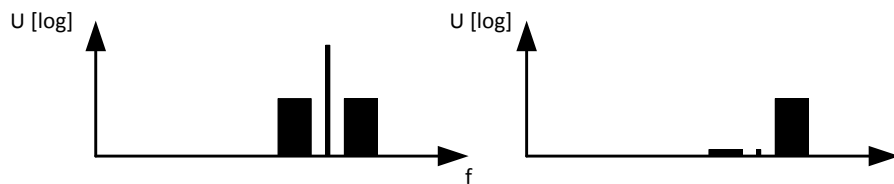
4 Algoritmusok bemutatása

Ebben a fejezetben bemutatok egy analóg és egy digitális jelek vételére szolgáló demodulációs eljárást: egy SSB és egy RTTY dekódoló algoritmust. Mindkét eljárás a rádióamatőr gyakorlatban nagyon elterjedt.

4.1 SSB demodulátor algoritmus

4.1.1 Bevezetés

Egy ilyen algoritmust előző munkámban [2] már megvalósítottam, itt egy jelentősen továbbfejlesztett változatát mutatom be.



37. ábra AM (balra) és SSB jel spektruma

Az SSB¹² jel legegyszerűbben az AM jelből származtatható (37. ábra), ha elnyomjuk a vivőt, illetve az egyik oldalsávot. Elnyomhatjuk az alsó vagy a felső oldalsávot ennek megfelelően lesz a jel USB¹³ vagy LSB¹⁴

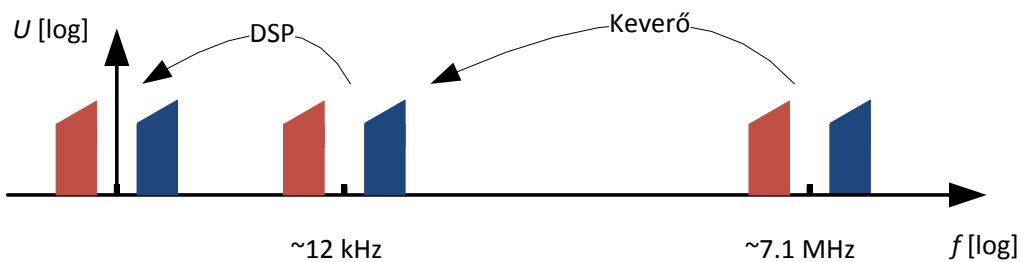
Ezáltal csökken a jel sávszélessége, illetve nő a „hatásfok”, ugyanis az adó nem sugározza ki a másik oldalsávot (ami ugyanazt az információt tartalmazza), illetve a vivőt. Cserébe viszont a modulátor és a demodulátor is jóval bonyolultabb. További hátrány hogy nincs vivő, így elviekben nem tudjuk meghatározni az állomás pontos helyét (a régi analóg TV műsorszórásban csak az egyik oldalsávot nyomták el a vivőt azonban megtartották, pont emiatt). Azért írtam, hogy elnyomott az egyik oldalsáv, illetve a vivő, mert a valóságban csak csökkenteni lehet a szintet, de tökéletesen megszüntetni nem. Különösképpen igaz

¹² Single Side Band; Egyoldalsávós jel általánosan

¹³ Lower Side Band; Egyoldalsávós jel, az alsó oldalsáv megtartásával

¹⁴ Upper Side Band; Egyoldalsávós jel, az felső oldalsáv megtartásával

ez az általam elkészített rádióra, ahol a két csatorna (I-Q) kiegyenlítetlensége miatt jelentkezik ez a hatás. A következőkben a rádió keverési elvét mutatom be.



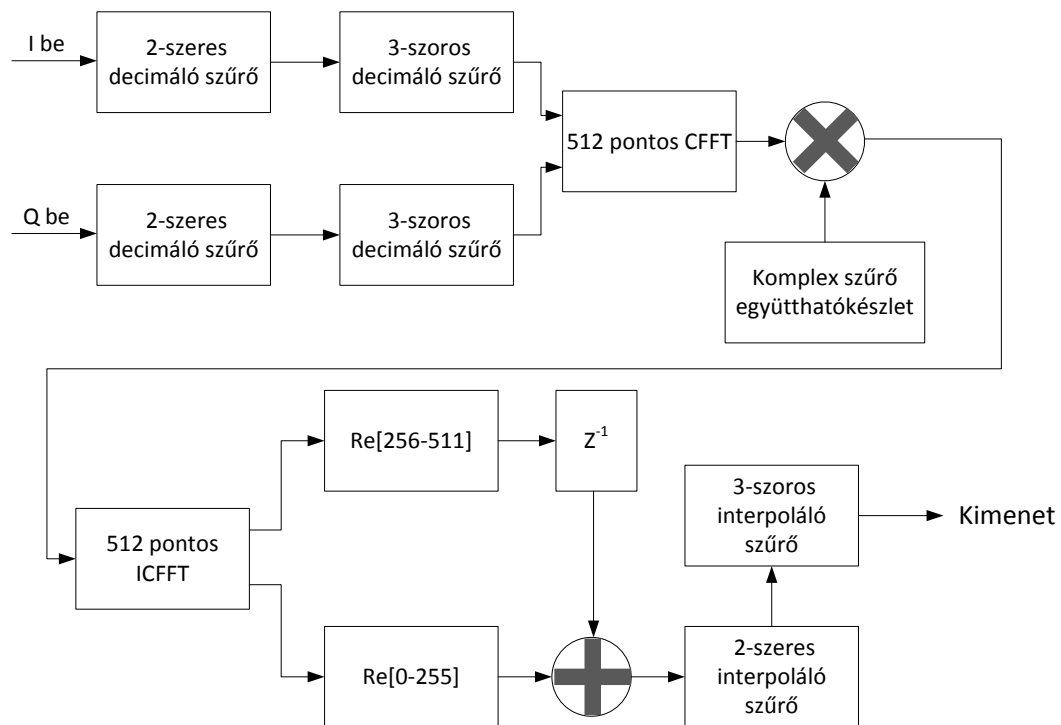
38. ábra A keverés elve

Az elvet a 38. ábra mutatja. Az ábra kissé torzított ezért is jelöltem a vízszintes tengelyt logaritmikusként. Az ábra jobb oldalán látható a venni kívánt rádiófrekvenciás jel. A kék a hasznos jel, a piros pedig a szomszédos csatornán dolgozó zavaró állomás. A jelet a keverő segítségével alacsony középfrekvenciára transzponálom, majd digitalizálom. Alacsony középfrekvenciát két okból alkalmazok:

- Az alkalmazott műveleti erősítőknek egyenáram közelében nagy a zajuk, és a specifikációban megadott zajszint csak 1 kHz-es frekvencia felett tekinthető mérvadónak.
- Az alkalmazott szintézer felbontása még túl kicsi ebben a tartományban (néhány száz Hz), így a finomabb hangolásokat digitálisan a mikrokontrollerben kell megoldani.

A jelfeldolgozó processzorban végül digitálisan alapsávba keverem a jelet. Mint az ábrán is látható, a nemkívánatos piros állomás átkerül a negatív frekvenciatartományba. Mivel a keverő komplex jelet állít elő, ebből a jelből kiválasztható a kívánt felső oldalsáv (kék). Ha egyszeres keverés lenne, a jel valós lenne, mivel azonban minden valós jel páros spektrumú, ezért az alsó és felső oldalsáv összekeveredett volna. Ezt matematikai formulák segítségével előző munkámban [2] már bemutattam. Az algoritmus feladata tehát eltávolítani a nemkívánatos ellenállomást.

4.1.2 Az algoritmus bemutatása



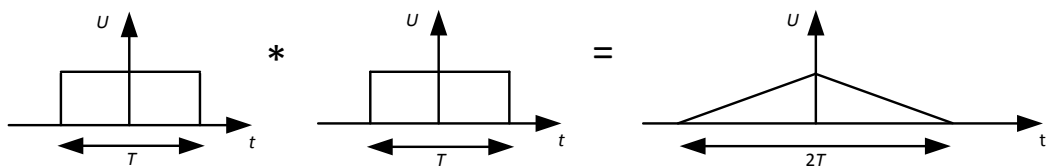
39. ábra A megvalósított algoritmus blokkvázlata

A 39. ábra mutatja az elkészült demodulátor algoritmus blokkvázlatát. A DMA által a tömbökbe töltött értékeket először lebegőpontossá alakítom, majd lekeverem alapsávba, ez azonban mindig megtörténik az alkalmazott algoritmustól függetlenül, így nem tekintem az algoritmus részének. A mintavételi frekvenciát a kezdeti 48 kHz-ről 8 kHz-re csökkentem két darab decimáló szűrővel. Ennek két oka van:

- A rádióamatőr gyakorlatban alkalmazott SSB adások sávszélessége 3.5 kHz, így a szűrők eltávolítják a csatornán kívüli felesleges, zavaró jelet.
- Így az adatmennyiség a hatodára csökken, ezért az algoritmus futási ideje is csökken.

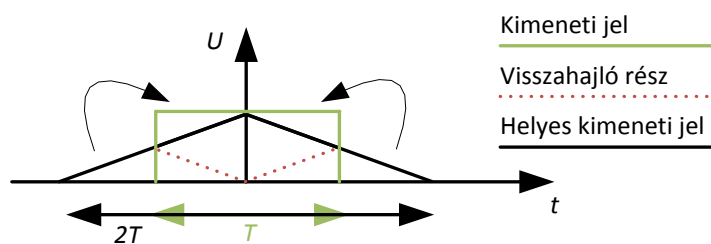
A DMA 1536 elemű blokkokban adja az adatot, így decimálás után 256 db értéket viszünk tovább. A decimált I és Q csatornát ezután komplex gyors Fourier transzformálom. A Fourier transzformáció 512 pontos. Mivel a rendelkezésre álló adatmennyiség csak 256, ezért a fennmaradó részt nullákkal töltöttem fel. A memóriában eltároltam egy 256 tapas komplex szűrő Fourier transzformáltját, a melyet hasonlóképpen állítottam elő (tehát a tárolt regisztrátum 512 elemű). E két tömböt összeszorozva megvalósítom a szűrést a

frekvenciatartományban. Ez azért előnyös mert időtartományban a FIR szűrővel történő szűrés egy konvolúció, ami frekvenciatartományban szorzásnak felel meg. Míg a konvolúció az elemszám N^2 -tel arányos számítási igényű addig a gyors Fourier transzformáció $M \log_2 N$ -nel arányos. Ezt az arányt picit rontja, hogy Fourier transzformálni kétszer kell (oda vissza), illetve a szűrés N darab komplex szorzással jár, tehát a valós számítási igény ennél több, de $N=512$ -nél jóval kisebb, mint N^2 , tehát ezzel a megoldással igen nagy sebességnövekedés érhető el. Feltűnő, hogy a Fourier transzformáció 512 pontos, viszont az alkalmazott szűrő, illetve a jel hossza is 256 minta. Erre a 40. ábra ad magyarázatot.



40. ábra Konvolúció bemutatása

Ebben az esetben két darab T hosszúságú négyszögjelet konvolúciója látható. Az eredmény $2T$ hosszúságú lesz. Belátható, hogy a tömbök megkétszerezése és nullákkal feltöltése nélkül (ezt nevezik zero padding-nek) a kimeneti jel nem „férne el” a neki szánt helyen. Ilyenkor a 41. ábra által mutatott képet kapjuk, a nem ábrázolható rész „visszahajlik”, és hozzáadódik a jelhez. A két komponens nem választható szét egymástól.



41. ábra Cirkuláris konvolúció zero padding alkalmazása nélkül

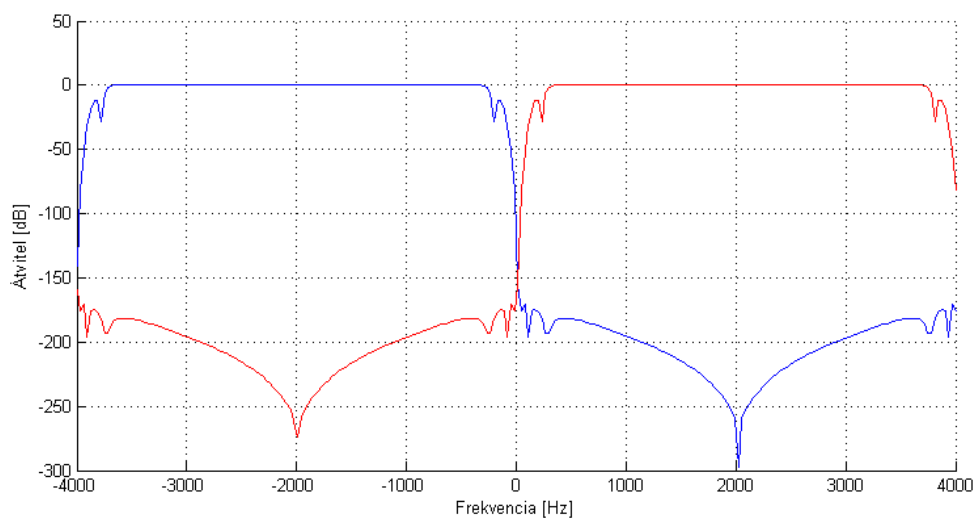
Az inverz Fourier transzformáció során kapott eredménynek vettem a valós részét, majd az eredményt két blokkra osztottam. Az első feléhez hozzáadom az előző ciklusban készült blokk második részét, ez kerül a kimenetre, majd eltárolom a második blokkot,

hogy majd a következő ciklusban a blokk elejéhez hozzáadhassam. Az alkalmazott módszer módszer neve OLA¹⁵.

A kapott kimenőjelet hatszorosan interpolálom (egy kétszeres és egy háromszoros interpoláló szűrő egymás után kapcsolva), majd típuskonverzió után a DA konverter kimeneti bufferébe töltöm.

A decimáló/interpoláló szűrők az előző algoritusból valók, azokat megtartottam. Azért alkalmazok két külön szűrőt, mert a kettő együtt kevesebb erőforrást igényel, mint egy darab hatszoros interpoláló/decimáló szűrő.

A komplex szűrő elkészítését MATLAB-ban végeztem. Először egy aluláteresztő szűrőt terveztem, a tervezéshez a `firls` függvényt használtam. Egy olyan 256 tapos szűrőt terveztem, amely a tartomány 0.25-ig átereszt, felette levág. Majd a szűrőegyütthatókat szoroztam $e^{j2\pi S(\frac{n}{N_f})}$ -el, ahol S a kívánt tolust (relatív frekvenciában), n az adott szűrőegyüttható, N_f pedig a szűrő hossza. Két szűrőt implementáltam egyet az alsó, egyet pedig a felső oldalsáv vételére.



42. ábra Az elkészült komplex szűrők a kék az alsó, a piros pedig a felső oldalsáv vételére

Az algoritmus megvalósításához kihasználtam a mikrokontroller adta lehetőségeket. Az összes tömböt amit használok a jelfeldolgozás során, a mikrokontroller CCM

¹⁵ Overlap add; azaz átfedés és összeadás.

memóriájába tároltam. Ez egy 64 kByte-os rész, amelyhez csak a CPU fér hozzá, más periféria nem. Ennek következtében ezt a területet a CPU mindig (nincs más busmaster aki fenntartaná a buszt, illetve ennek a busznak azonos az órajele a CPU-val) egy ciklus alatt eléri, ez már önmagában is jelentős sebességnövekedéshez vezet. Ez a memóriaterület a 0x10000000-0x10010000 tartományon található. Ahhoz hogy ezt használhassam, módosítanom kellett az ún. scatter filet, ami tisztázza, hogy az egyes memóriaterületek milyen típusúak (RAM vagy ROM), illetve mely változók használhatják azt:

```
RW_IRAM2 0x10000000 0x00010000 { DSP.o (CCM) }
```

Ezt követően a tömbök deklarálása a következőképpen történik:

```
float Idataf[1536] __attribute__((section("CCM")));
```

A jelfeldolgozó algoritmushoz szükséges minden függvény implementálva volt a DSP könyvtárban. Ezek a függvények optimalizáltak a CPU felépítéséhez, ezeknél aligha tudnék gyorsabban lefutó algoritmust írni.

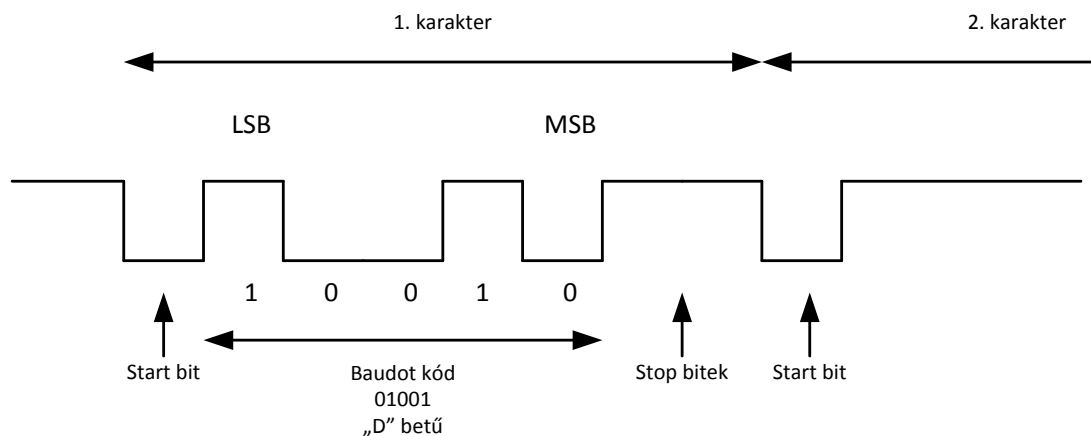
4.2 RTTY demoduláló algoritmus

A következőkben egy RTTY demoduláló, illetve dekódoló algoritmust mutatok be.

4.2.1 Bevezetés

Az RTTY (Radio TeleTYpe) a XX. század elején kifejlesztett digitális adatátviteli forma mely széleskörűen elterjedt, a katonaság és a posta is használta, illetve a rádióamatőrök használják a mai napig. Sok típusa van, de én most csak a rádióamatőr gyakorlatban alkalmazottat mutatom be [18] [17].

A kódolása FSK, 170 Hz a két jelzés közötti frekvencia (shift), illetve Baudot kódolást használ. Egyes karakterek átvitele (hasonlóan az RS-232-höz) egy startbittel kezdődik, majd 5 bitnyi Baudot kód [19], végül 2 bitnyi stopbit. A jelzési sebesség 45.45 baud. A protokoll az LSB bitet küldi először. Egy „D” karaktert a 43. ábra mutat illusztrációként.



43. ábra Egy karakter kódolása az RTTY kommunikációhoz

Általában a logikai „1” értékhez tartozó frekvenciát szokás megadni ez a MARK, ehhez képest 170 Hz-el lejjebb található a „0” jelzés vagyis SPACE. A start, illetve stopbitek megkönnyítik a karakter szinkronizálását.

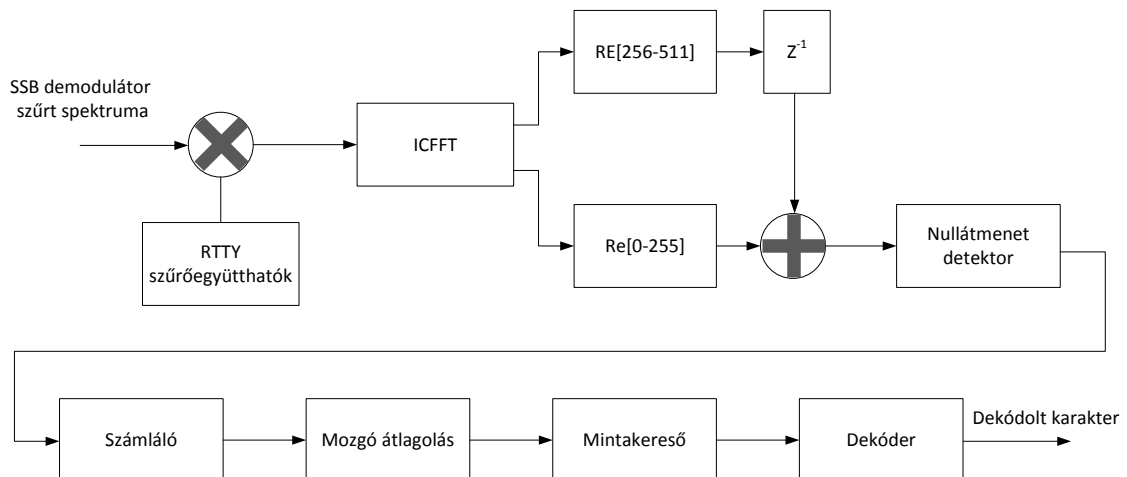
4.2.2 Az algoritmus bemutatása

Az algoritmus az előzőekben bemutatott SSB demoduláló algoritmus kiterjesztésének tekinthető. Az FFT-zett jelet ez esetben egy másik szűrővel szűröm meg. Ez is egy komplex szűrő, de ez csak az RTTY jel két jelzőfrekvenciájához tartozó frekvenciákat, és kicsiny környezetét engedi át. Az adott állomásra hangolás „fültre”

történik. Belátható hogy a rádióban a két algoritmusnak egymással párhuzamosan kell működni, hiszen hallanunk kell, hogy az adott jelre megfelelően ráhangolódtunk. A szűrő megtervezése előtt azonban el kellett dönteni, hogy az FSK jelet milyen módszerrel demodulálom, és a különböző eljárások különböző frekvenciatartományt kívánnak meg. Három lehetőséget fontoltam meg:

- DPLL, azaz a mikrokontrollerben leprogramozok egy PLL-t, amelynek a bemenete a demodulálni kívánt jel, kimenetet pedig a NCO bemenetéről veszem le.
- Demodulátor algoritmus alkalmazása. Manapság sokféle FM demodulátor algoritmus létezik, ezek viszont bonyolultak, valamilyen szögfüggvényt, kell számolni, (tipikusan $\arctg()$ -t), vagy osztani kell benne, ami nagy számítási igénnyel jár.
- Számláló alkalmazása. Mivel nem egy FM jel demodulálása a cél, csupán el kell dönteni, hogy két frekvencia közül éppen melyiket vesszük. Így elég, ha megmérjük a jel periódusidejét/frekvenciáját, ennek megfelelően döntjük el, hogy éppen milyen frekvenciájú jel jön.

Én a harmadik lehetőséget választottam, mert ezt a legkönnyebb implementálni. A megvalósított algoritmust a 44. ábra mutatja.

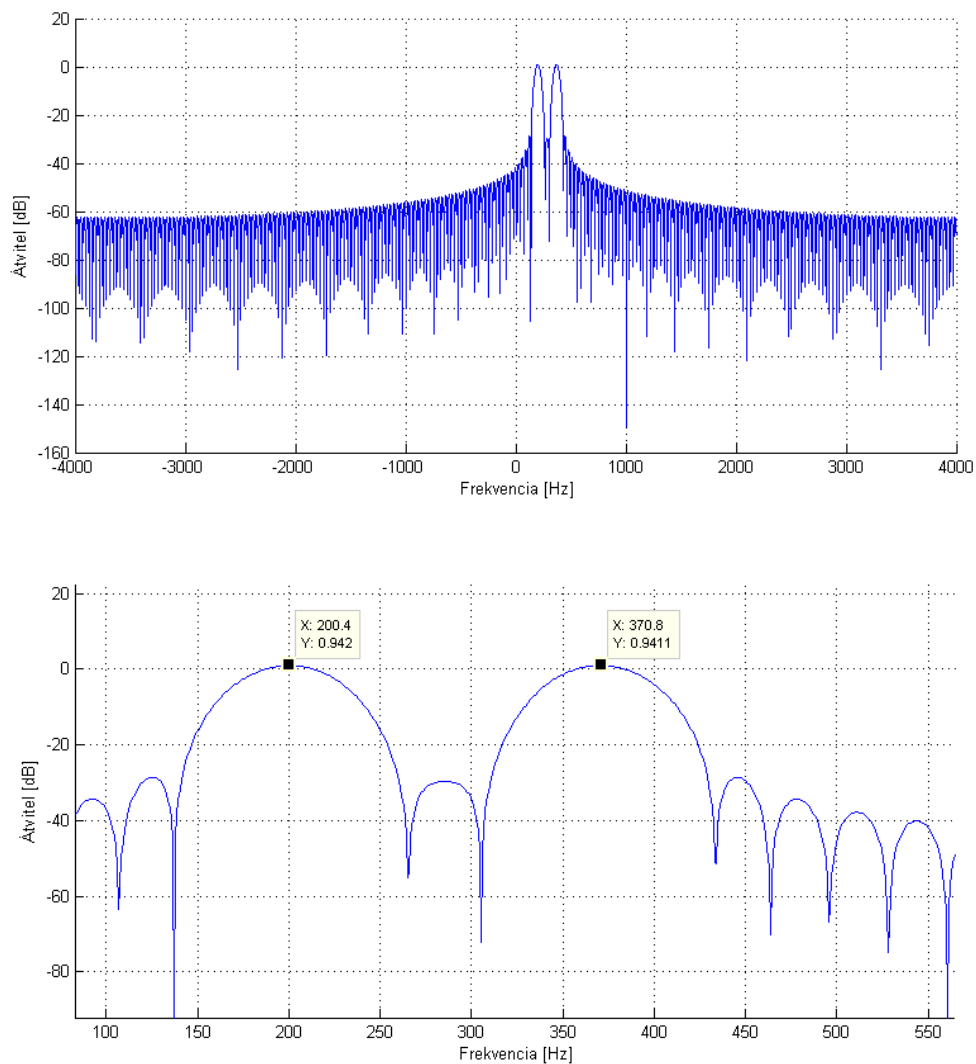


44. ábra Az RTTY dekódoló lgoritmus

A módszer ismeretében már meg lehet választani a szűrő átvitelét. Adott a 170 Hz-es frekvenciakülönbség a két jelzés között, és egy olyan frekvenciát kell választani, ahol a periódusidő különbség még szignifikáns, emellett könnyen behangolható fülre a rádió.

Választásom a 200-370 Hz frekvenciapárosra esett. A periódusidő ilyenkor 5 illetve 2.7 ms, ezt a különbséget még zajos jel esetén is könnyen detektálhatunk. Erre a két frekvenciára készítettem egy komplex szűrőt a 4.1.2 fejezetben bemutatott módszer segítségével. Itt azonban egy sávszűrőből indultam ki melynek középfrekvenciája 85 Hz sávzélessége pedig 50 Hz. Ezt a szűrőt kellett eltolni frekvenciatartományban 285 Hz-el, így a szűrő középfrekvenciái a kívánt 200 és 370 Hz-es frekvenciára kerülnek. Az elkészült szűrő átvitelét a 45. ábra mutatja. Látható hogy 256 tapes FIR szűrő ilyen kis áteresztő tartományon nem tud olyan meredek lenni. Viszont kissé távolabb az áteresztő tartománytól a csillapítás már 60 dB, ami jónak mondható.

Megegyezés szerint RTTY adást csak felső oldalsávon sugároznak, ezért csak egy szűrőt készítettem el.



45. ábra RTTY szűrő átvitele

A szűrő a bemeneti jelét az SSB demodulátor szolgáltatja, mégpedig az FFT-zett, szűrt jelet kapja meg. Ezt megszűrjük az RTTY szűrővel, majd az OLA módszer segítségével visszatranszformáljuk időtartományban, ezt a jelet dolgozom fel.

4. táblázat RTTY dekóder adatok

Frekvencia	Időköz	Mintaszám ($f_s = 8 \text{ kHz}$)
„0” jelzés 200 Hz	5 ms (két él között 2.5 ms)	20
„1” jelzés 370 Hz	~2.7 ms (két él között 1.3 ms)	10.4
bitidő (45.45 baud)	22 ms	176

Mivel digitális rendszerről van szó, ahol a jel mintavételezett, érdemes az időmérést mintaszámban végezni. A dekóderhez használt fontosabb időtartamokat összegzését mutatja a 4. táblázat.

Az időtartományba transzformálást követően egy nullátmenet-detektort használok. Itt meghatároztam egy 0 körüli intervallumot, amelyen belül kerülve az algoritmus nullátmenetet jelez. Nullátmenet jelzés után az algoritmus átugorja a következő 6 mintát, ennyi időn belül elvileg nem történhet nullátmenet, ezzel növeltem a zajimmunitást. A nullátmeneteket számlálom egy mozgó átlagolóval. Az átlagoló egy öt elemű cirkuláris puffer. A tömb egy elemébe 44 mintányi ideig gyűlik a nullátmenetek száma, majd a legrégebbi elemet eldobjuk, és a helyére gyűjtjük a legfrissebb nullátmeneteket.

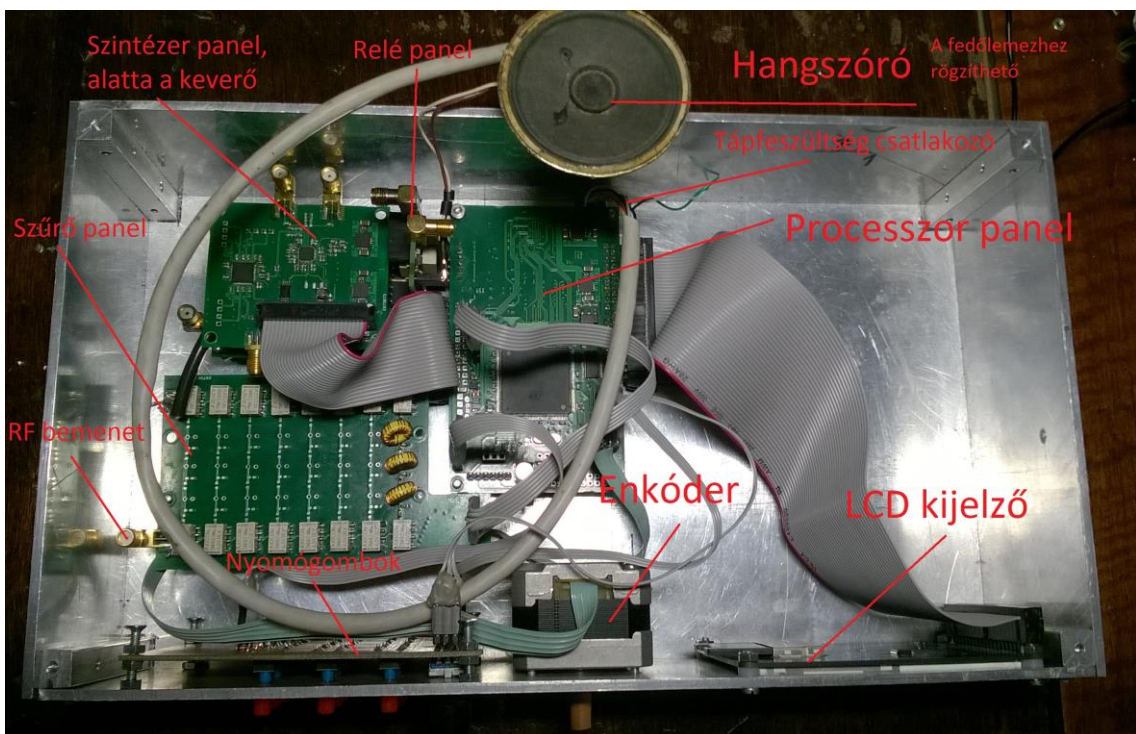
1-2 nullátmenet esetén alacsony, 3- nullátmenet esetén pedig magas szintnek veszi a tároló eredményét a program. 176 minta után az 5 tárolót megvizsgálja az algoritmus, és amelyik szintből több van, az lesz a bit értéke. A tárolók összesen 220 mintányi nullátmenetet tárolnak, így a bitmintavétel helyzete nem számít.

Ezt követően az elkészült biteket egy 16 bites regiszterbe shiftelem. Ekkora helyre pontosan két karakternyi információ fér (5 bit baudot kód +1 start, 2 stop bit). Az adott karakter csak akkor kerül kiírásra, ha a start és stop bitek megfelelőek mindkét mintában. Erre azért van szükség, mert 8 bitnyi mintában jóval nagyobb az esély arra, hogy a szükséges 3 bit megegyezik, de mégsem egy valós karaktert vett a dekóder. 16 bit esetén ennek valószínűsége már jóval kisebb.

5 Elkészült eszköz bemutatása

Az alábbi fejezetben bemutatom az elkészült eszközt, illetve bemutatom a rádióval kapcsolatos néhány mérést. Ezeket úgy válogattam össze, hogy egy-egy elkészült panel jellemzőjét mértem le.

Mechanikai felépítés és demonstráció



46. ábra A rádió elől-, illetve felülnézetből

A 46. ábra mutatja az elkészült rádiót. A műszerdobozt saját magam készítettem. Az előlap megtervezésénél ügyeltem arra, hogy az elrendezés ergonomikus legyen. Az ábra alsó részén látható a belső elrendezés, illetve az ábrába beleírtam az egyes komponensek

neveit is. Jobb oldalt szándékosan hagytam helyet, ide elfér egy tápegység, esetleg egy akkumulátor, illetve egy néhányszor 10 W-os RF erősítő is.

Ahol lakom, még nem sikerült egy rövidhullámú antennát telepítenem, viszont szükségesnek láttam működés közben bemutatni az elkészült rádiót. Mivel a rádióval a középhullámú állomások is gond nélkül vehetők, ezért az 540 kHz-en sugárzó Kossuth rádió vételét mutatom be. Az adóteljesítmény óriási (1 MW), ezért egy fél méter hosszú vezetékdarabot használva „antennaként”, jó vétel érhető el.

Az elkészült videó a <http://1drv.ms/1emiciK> linken érhető el.

AM adás gond nélkül vehető SSB demodulátorral is, azonban ha félrehangoljuk a rádiót, akkor hallani fogjuk a vivőt is. Picit félrehangoltam mindkét irányban és a videón jól hallható hogy egyik esetben a vivő halkabb, tehát van valamekkora tükörelnyomás. Ennek pontos műszerekkel való mérését a későbbiekben ismertetem.

Fáziszaj mérése

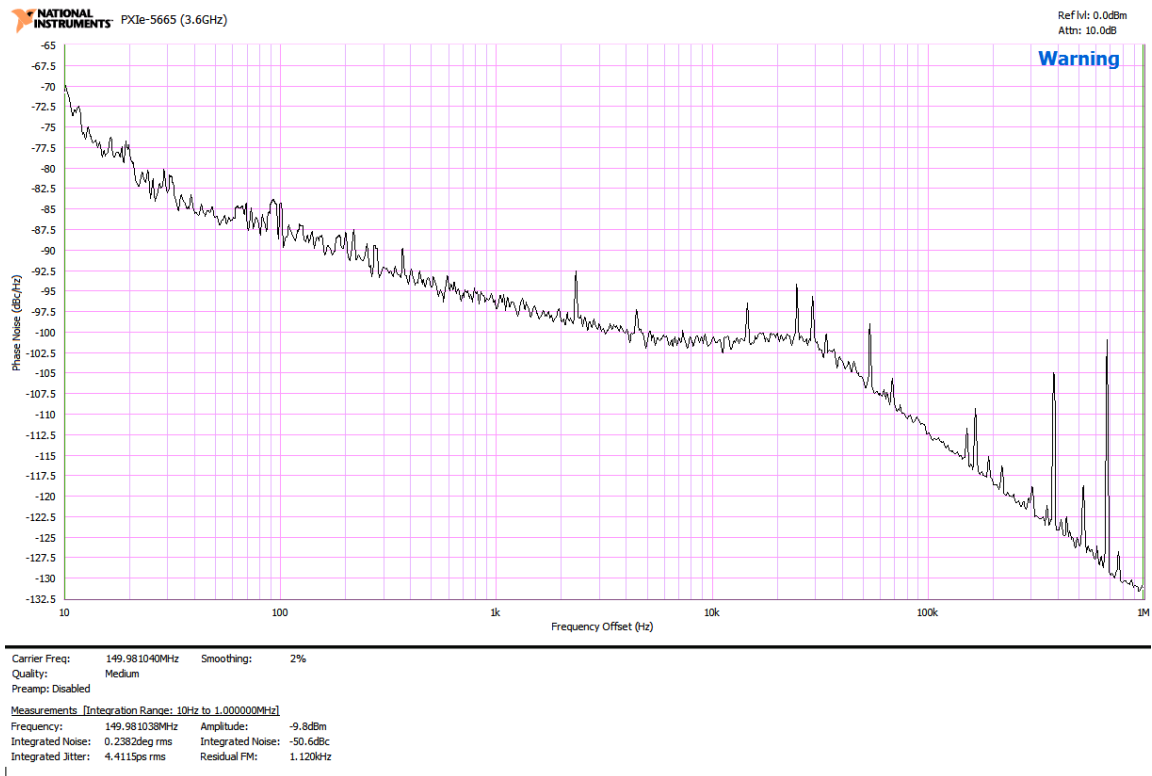
A fáziszaj egy jellemzője az oszcillátoroknak

A fáziszaj méréshez speciális műszer szükséges, egy átlagos spektrumanalizátorral nem lehet megmérni, mert az analizátor saját zaja jóval nagyobb, mint a fáziszaj. A mérést egy PXIe-5665 típusú National Instruments gyártmányú VSA¹⁶-val mértem meg.

A mérési frekvencia ~150 MHz volt. Több okból végeztem ezen a fekvencián a mérést:

- A jelet direkt a szintézerből vezettem ki, mert ez van kivezetve SMA csatlakozóra.
- A PLL felszorozza a referenciafrekvenciát, és a fáziszajt is. Túl alacsony frekvencián (néhányszor 10 MHz) a fáziszaj olyan kicsi lenne, hogy nem lehetne mérni.

¹⁶ Vector Signal Analyzer



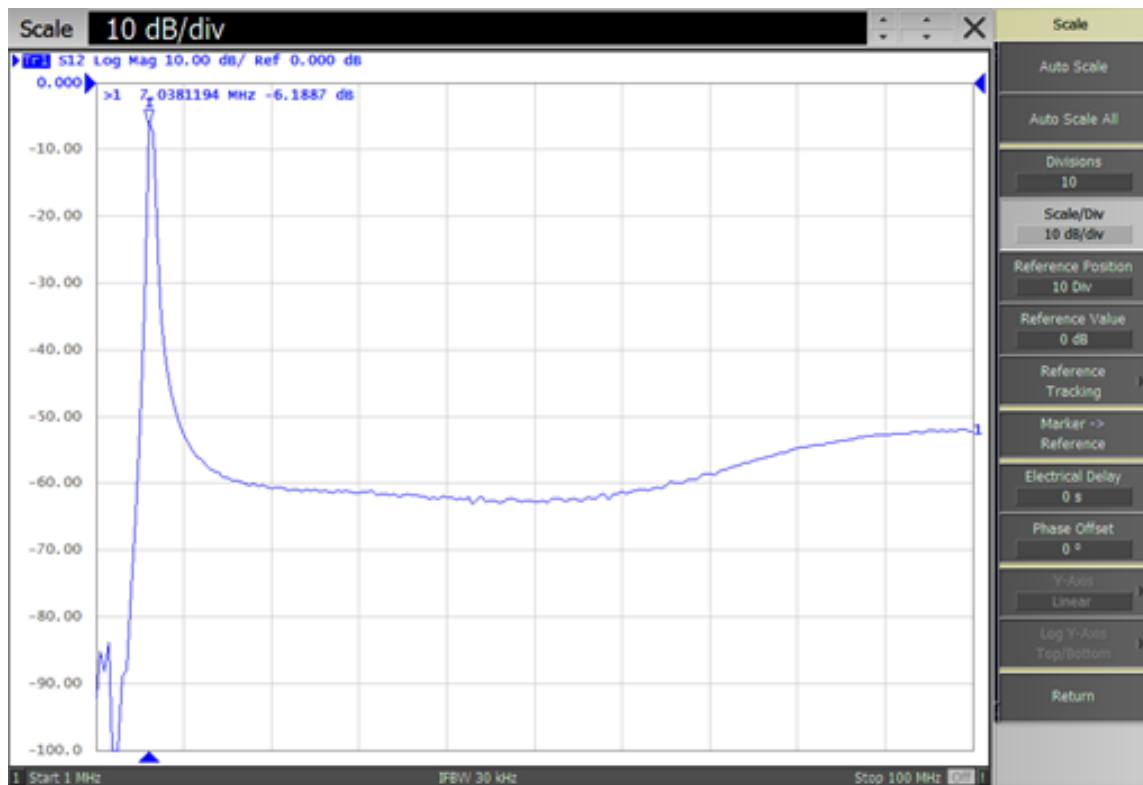
47. ábra PLL fáziszaja

A 47. ábra mutatja a PLL fáziszaját. Az ábrán jól látható a hurokszűrő sávszélessége (30 kHz), e felett a fáziszaj drasztikusan csökkenni kezd. Láthatók továbbá a tört osztásarányú PLL által termelt zavarok (spuriousok) (jobbaldalt). A kapott görbe rosszabb, mint ami a szintézer adatlapjában található. Hogy miért, annak számos oka lehet, ezek közül néhány:

- Rosszul megtervezett NYÁK lemez.
- Nagy referenciaoszillátor fáziszaj.
- Helytelen hurokszűrő méretezés.

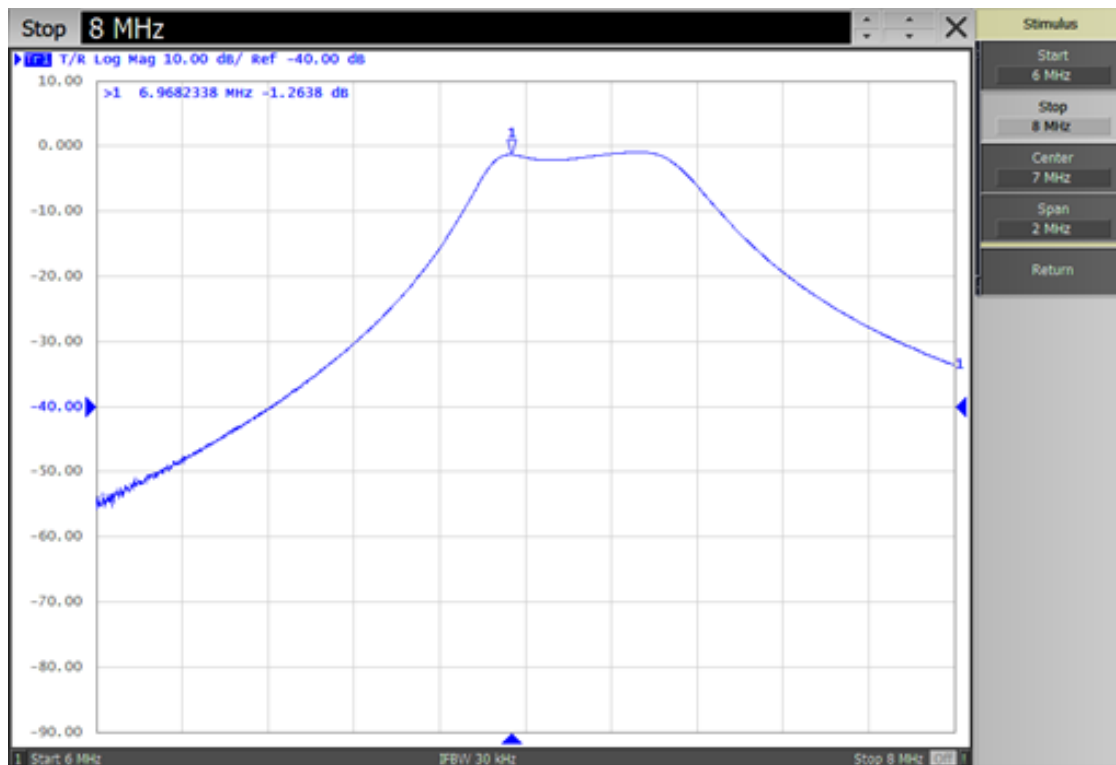
Sávszűrő mérése

A 7-7.2 MHz-ig terjedő rádióamatőr sávra tervezett sávszűrő átvitelének méréséhez az Agilent gyártmányú E5061B típusú hálózatanalizátort használtam. A behangolás viszonylag hosszú folyamat volt, itt csak a végeredményt mutatom be.



48. ábra A 7-7.2 Mhz-es rádióamatőr sávra tervezett szűrő átvitele széles tartományban

A szűrő átvitelét a 48. ábra és 49. ábra mutatja. Látható hogy a szűrő széles frekvenciatartományban -60 dB-es elnyomást képvisel. Az átvitelt közelről vizsgálva látható némi csillapítás az átérésztőtartományban, ez a szűrőben használt alkatrészek veszteségeiből tevődhet össze. Látható némi ingadozás is, amit nem sikerült kiküszöbölni a hangolás során, mivel azonban ez viszonylag kicsi (max. 2 dB), ezért nem befolyásolja számottevően az átvitelt. Összességében elmondható hogy a szűrő jól sikerült, a számolt karakterisztikára (30. ábra) nagyon jól illeszkedik a valós karakterisztika.



49. ábra A 7-7.2 Mhz-es rádióamatőr sávra tervezett szűrő áteresztő tartományban

Tükörelnyomás mérése

A jellemző méréséhez spektrumanalizátorra (HP3585B) és szignálgenerátorra (Agilent E4434B) volt szükség. A szignálgenerátoron beállítottam 7.101 MHz-et és -60 dBm-es jelszintet. A spektrumanalizátorral a rádió kimenetét vizsgáltam úgy, hogy USB módban van, a szintézer pedig 7.1 MHz-re állítottam. Ekkor a kimeneten 1 kHz-es jelnek kell megjelennie. A jel szintjét feljegyeztem, -30 dB volt. Ezután a szignálgenerátoron a frekvenciát 7.099 MHz-re állítottam. Ez pont a tükörfrekvencia, itt elméletileg nem kellene látnunk semmit, a jelszint azonban -63 dB volt. A tükörelnyomás a két szint különbsége, azaz 33 dB. Mivel az algoritmus közel tökéletesnek mondható (42. ábra szűrője -150 dB alá megy), ez teljes mértékben az I és Q csatorna kiegyenlítetlenségéből, vagy a szintézer által szolgáltatott keverőjel fázishibájából származhat. Ez elég rossznak mondható, és sajnos ez a direkt alapsávba keverő rádiók gyengéje.

6 Összegzés

A kitűzött célt sikerül elérni, terveztem egy rádiót, ami analóg és digitális adás vételére képes a 40 m-es rádióamatőr sávban. Eközben nagyon sok hasznos ismeretet szereztem a hardver- és szoftverfejlesztés, a rádiótechnika, jelfeldolgozó algoritmusok és mérés technika területén. A fejlesztés közben különös hangsúlyt fektettem a bővíthetőségre illetve a flexibilitásra. Például a processzor panelt a megfelelő programkomponensek implementálása után használni lehetne USB-s hangkártyaként, amely manapság nagyon divatos. A rádióban jóval több hardverelem lett implementálva, mint amennyit jelenleg használok, így a rádió fejleszthető további anyagi ráfordítás nélkül, mert a hardverben felmerült hibákat sikerült kijavítani.

A bővítési lehetőségek száma nagyon nagy, mind programozás, mind pedig hardverelemek tekintetében, a teljesség igénye nélkül néhány:

- URH keverő,
- URH szűrő,
- erősítő az adás módhoz,
- tápegység (akkumulátor, illetve hálózati táplálásra is),
- érintő kijelző, SD kártya élesztése,
- GUI továbbfejlesztése, spektrum és waterfall kijelzése,
- további demodulátor algoritmusok implementálása.

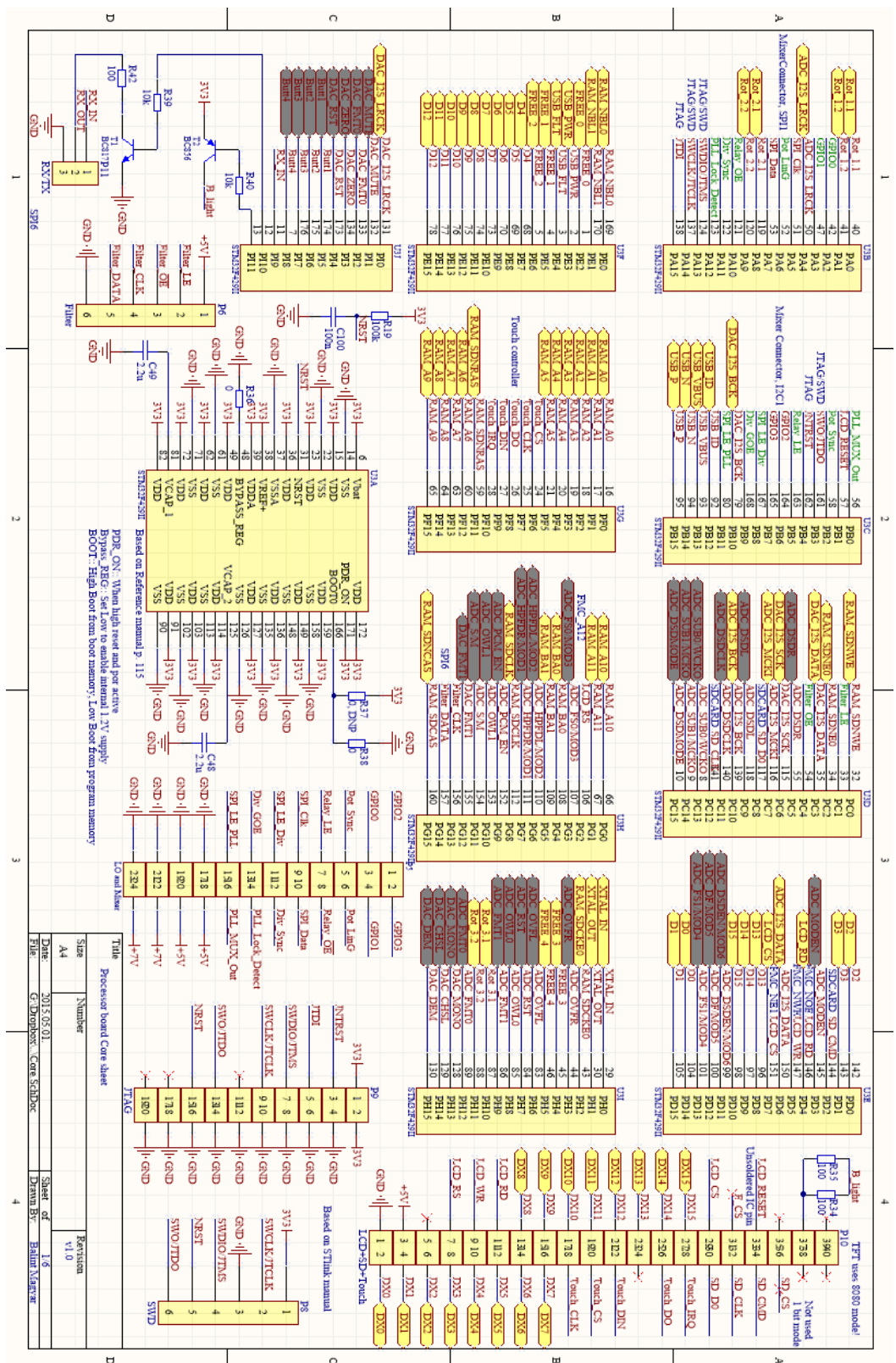
Ezek alapján a fejlesztésnek még korán sincs vége, és ehhez az elkészült eszköz nagyon jó alapul szolgál.

Irodalomjegyzék

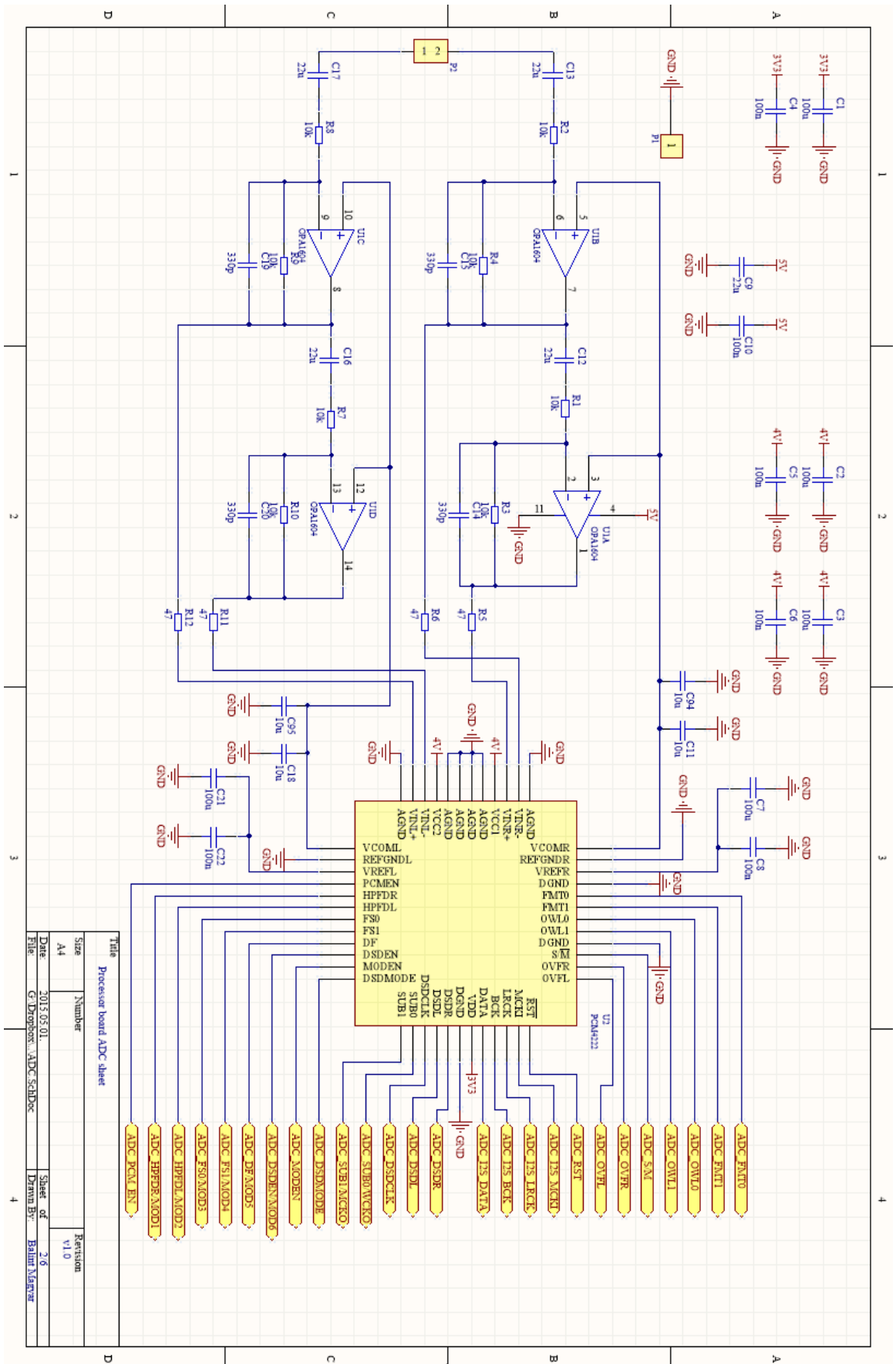
- [1] Eurocircuits Buildup selector. www.eurocircuits.com
- [2] Magyar Bálint Alacsony középfrekvenciás digitális jelfeldolgozó egység rádióamatőr célra BSc szakdolgozat 2012.
- [3] RM0090 STM32F429 Reference manual. www.st.com
- [4] PCM4222 datasheet. www.ti.com
- [5] PCM1694A datasheet. www.ti.com
- [6] AN2790 App. note: TFT LCD Interfacing with the high density STM32F10xxx FSMC. www.st.com
- [7] SSD1289 datasheet. www.solomon-systech.com
- [8] Magyar Bálint Analóg frontend készítése szoftverrádióhoz Önálló laboratórium zárójegyzőkönyv 2013.
- [9] Négyesi Gábor: Szintézeres lokáloszcillátor tervezése szoftverrádió alkalmazásához diplomamunka 2002.
- [10] ADF4351 datasheet. www.analog.com
- [11] LMK01020 datasheet. www.ti.com
- [12] SLAA120 Application report Interfacing Between LVPECL, VML, CML, and LVDS levels. www.ti.com
- [13] SCAA082 Application report High-Speed Layout Guildelines www.ti.com
- [14] Ultralow noise, high performance, zero IF quadrature product detector and preamplifier, Dan Tayloe 2003.
- [15] Application note #5 Latching relays 2009. www.te.com
- [16] DM00068628 STM32F42xx, and STM32F43xx errata sheet. www.st.com
- [17] RTTY specifikáció <http://www.ham.hu/radiosatvitel/szoveg/RTTY/>
- [18] RTTY specifikáció <http://en.wikipedia.org/wiki/Radioteletype>
- [19] Baudot kódolás http://en.wikipedia.org/wiki/Baudot_code
- [20] Practical RF Circuit Design for Modern Wireless Systems, Vol. 2, Active circuits and systems, Rowan Gilmore, Les Besser, Arctec House, London 2003
- [21] Practical RF Circuit Design for Modern Wireless Systems Vol 1 Passive circuits and systems Rowan Gilmore, Les Besser, Arctec House, London 2003

- [22] Rádiótechnika Évkönyve 2000, 201. oldal Nagy Gyula: Oszcillátorok fáziszaja
Rádióvilág Kft. Budapest 2000
- [23] SCDA008 Application Report CBT-C, CB3T, and CB3Q Signal-Switch Families
www.ti.com

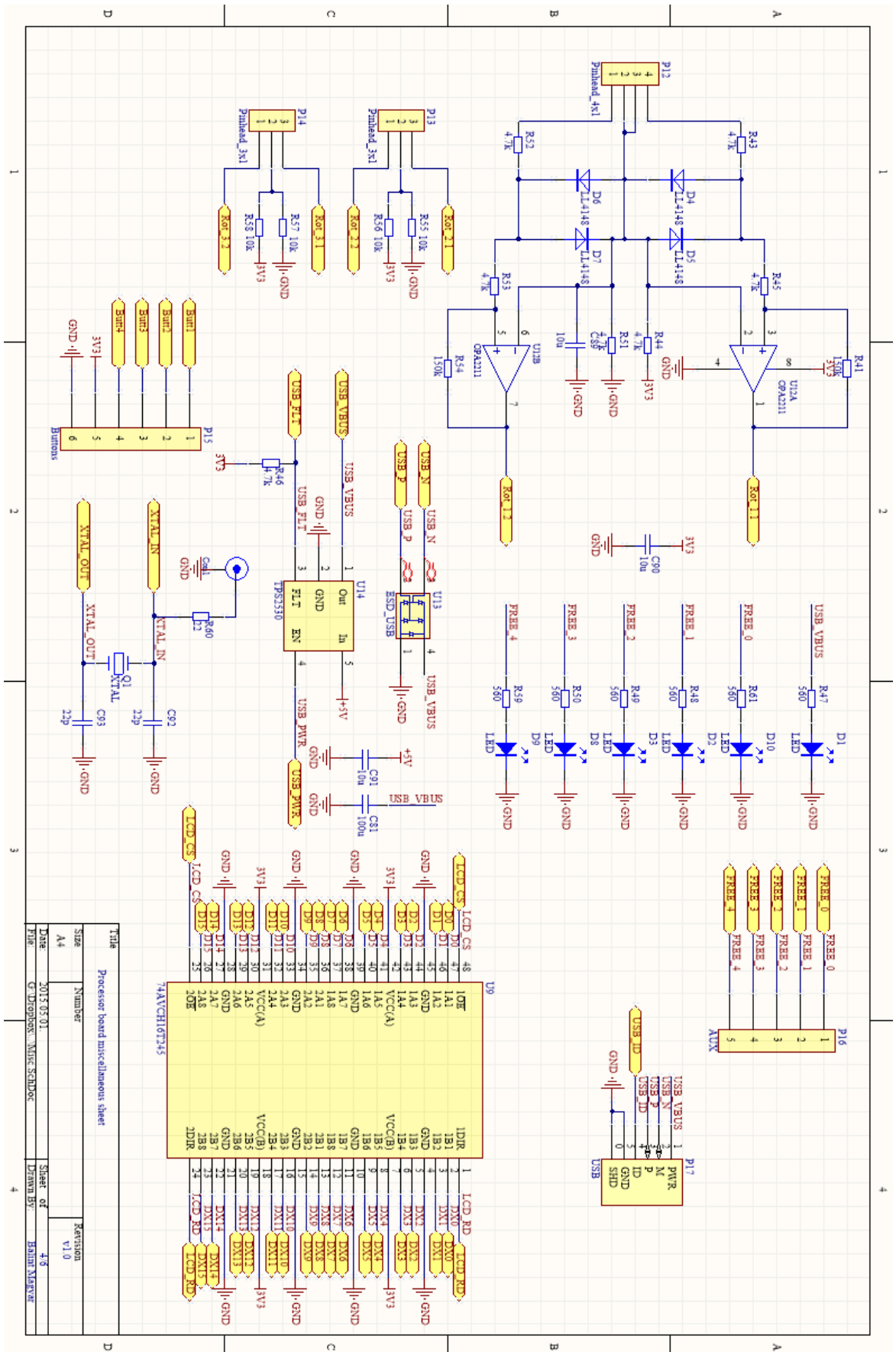
Függelék



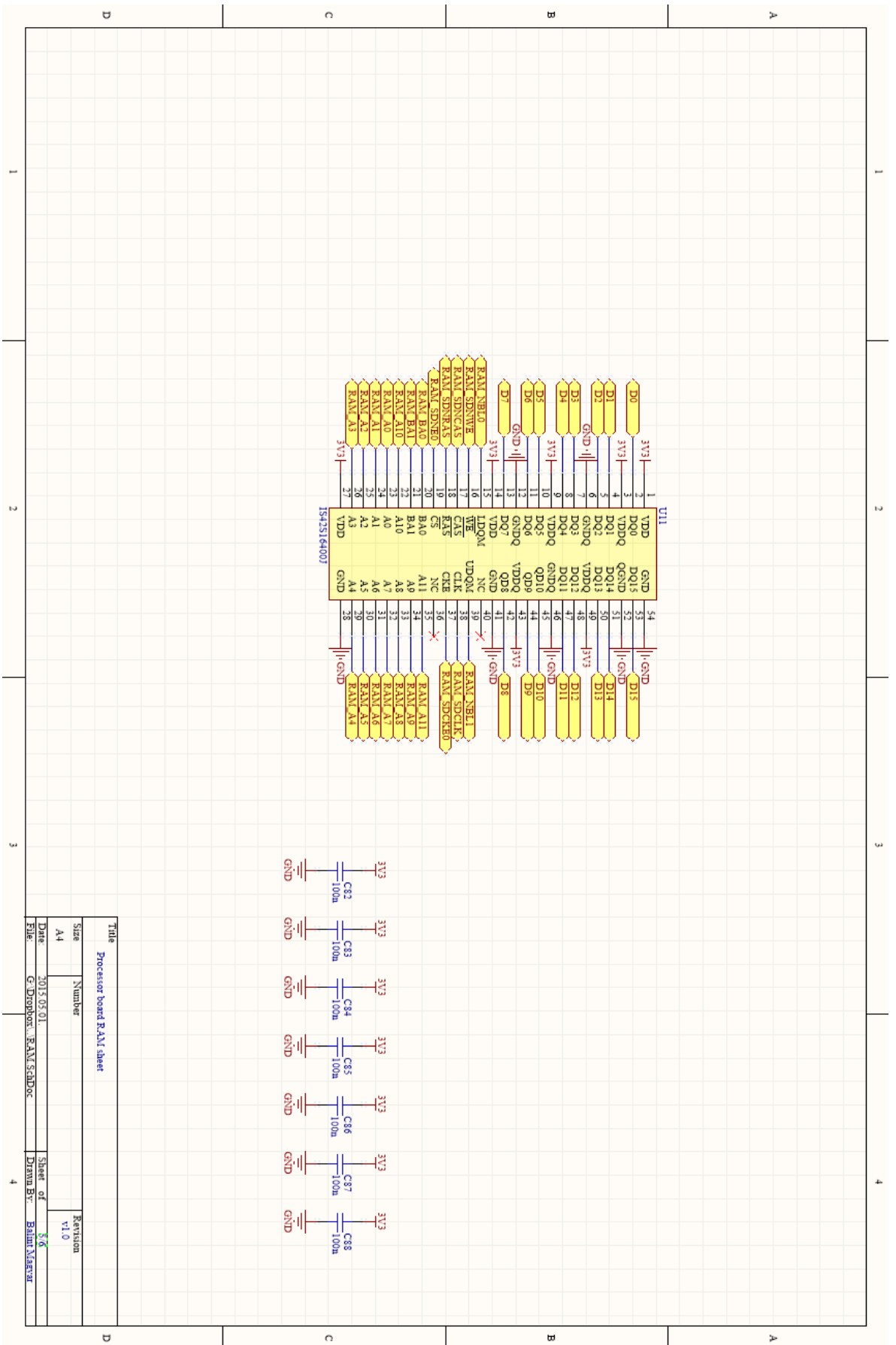
50. ábra Mikrokontroller kapcsolási rajza



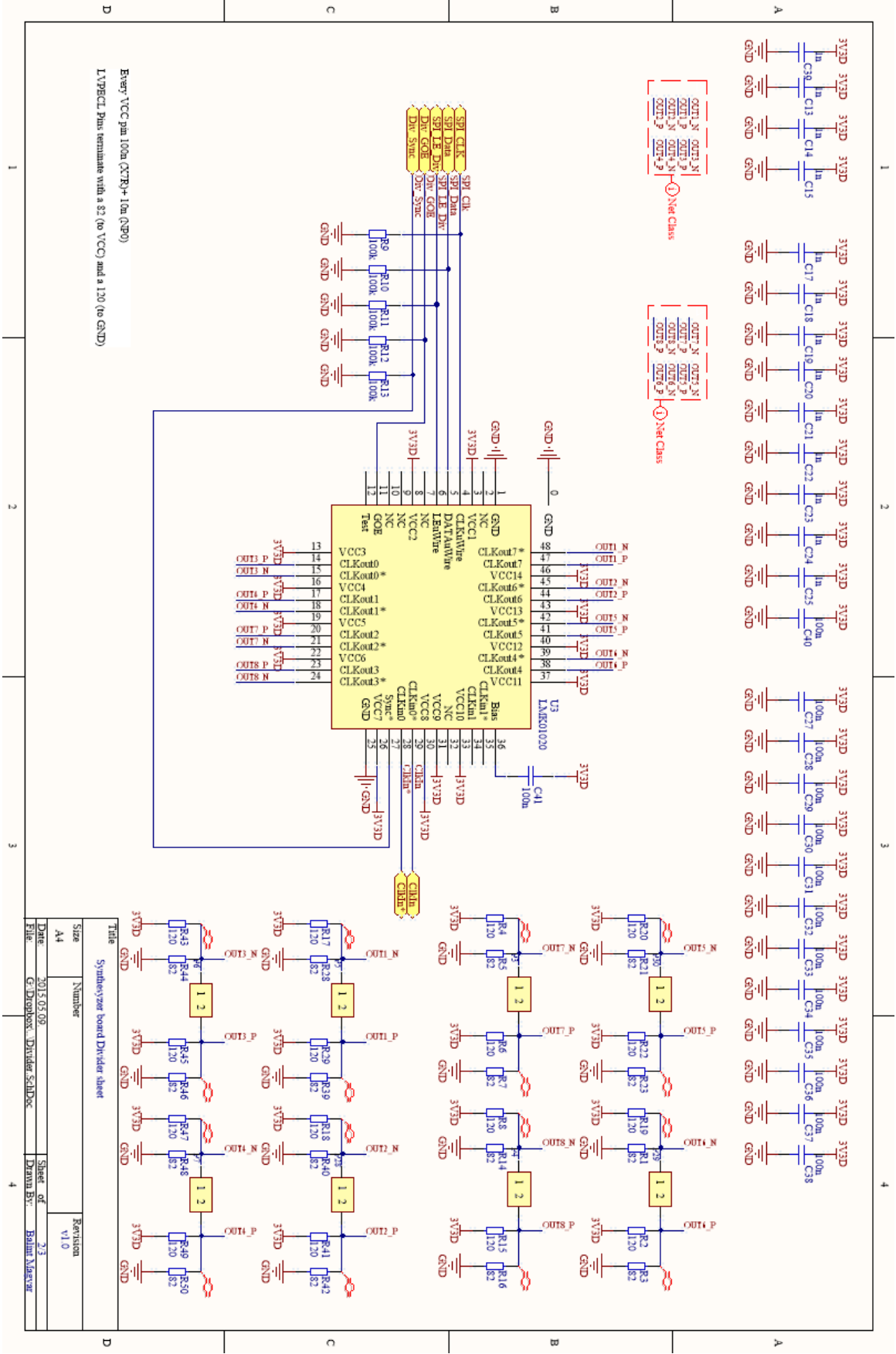
51. ábra AD konverter kapcsolási rajza



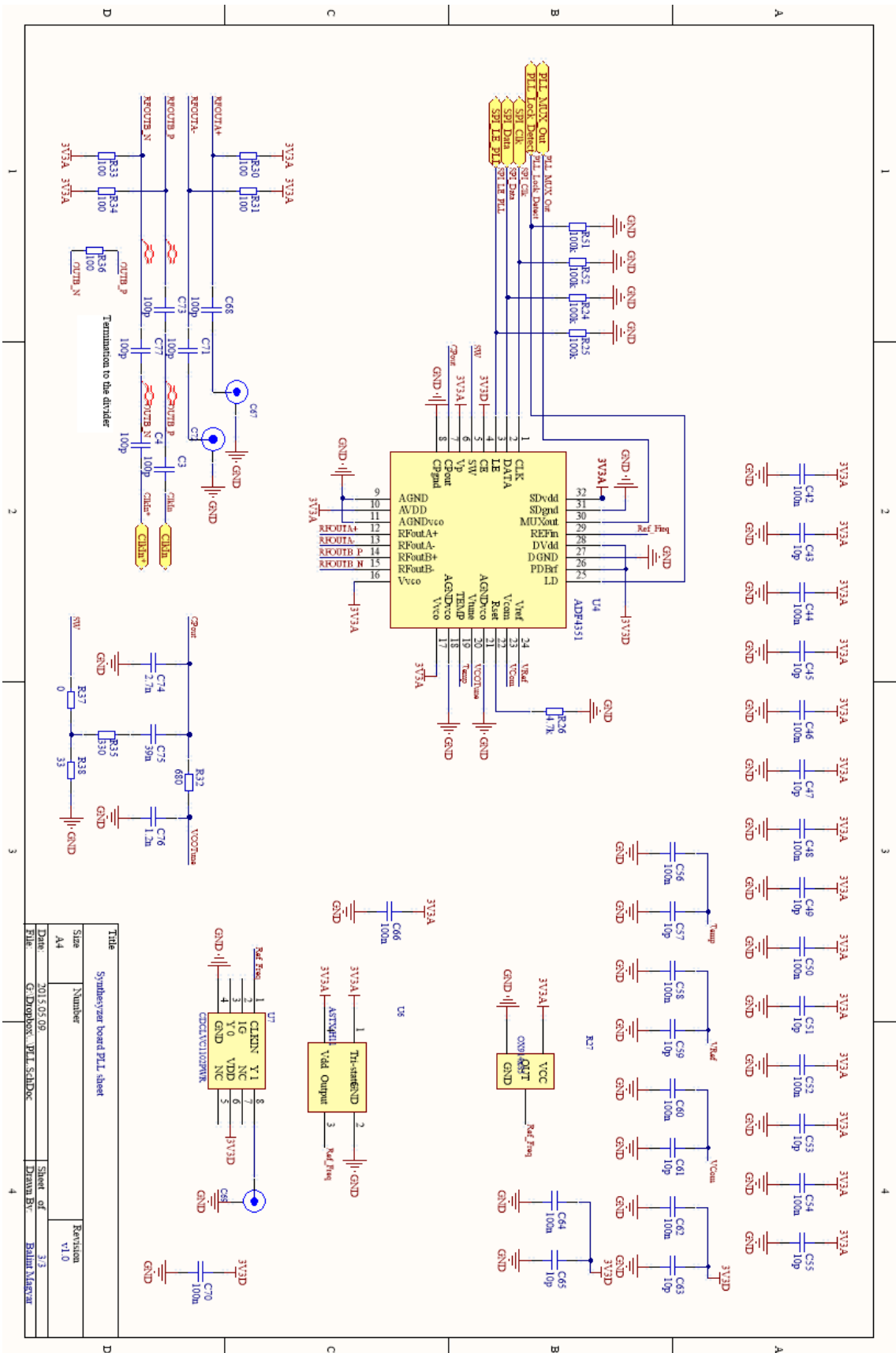
53. ábra Egyéb áramkörök kapcsolási rajzai



54. ábra SDRAM kapcsolási rajza

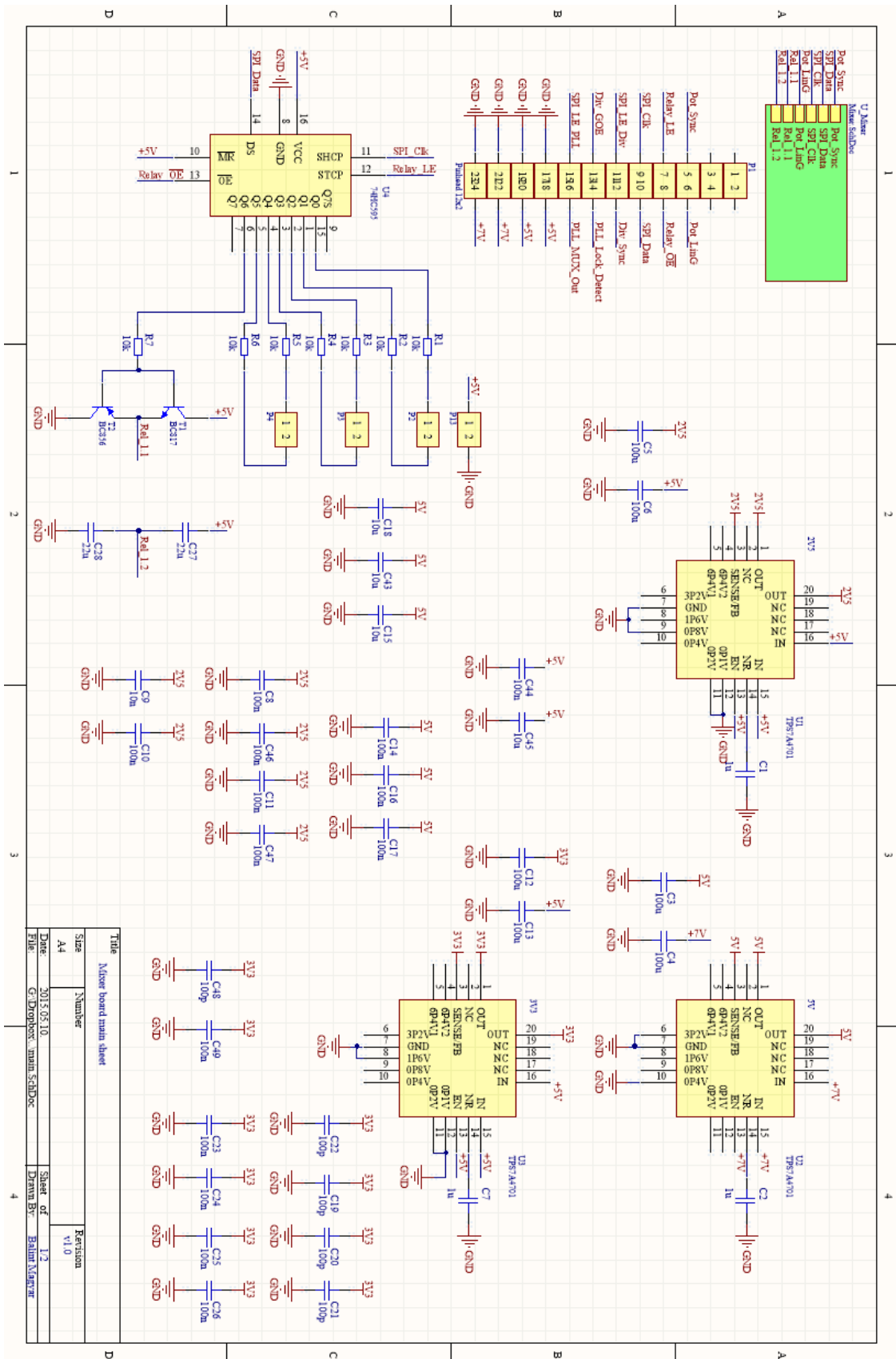


57. ábra Osztó kapcsolási rajza

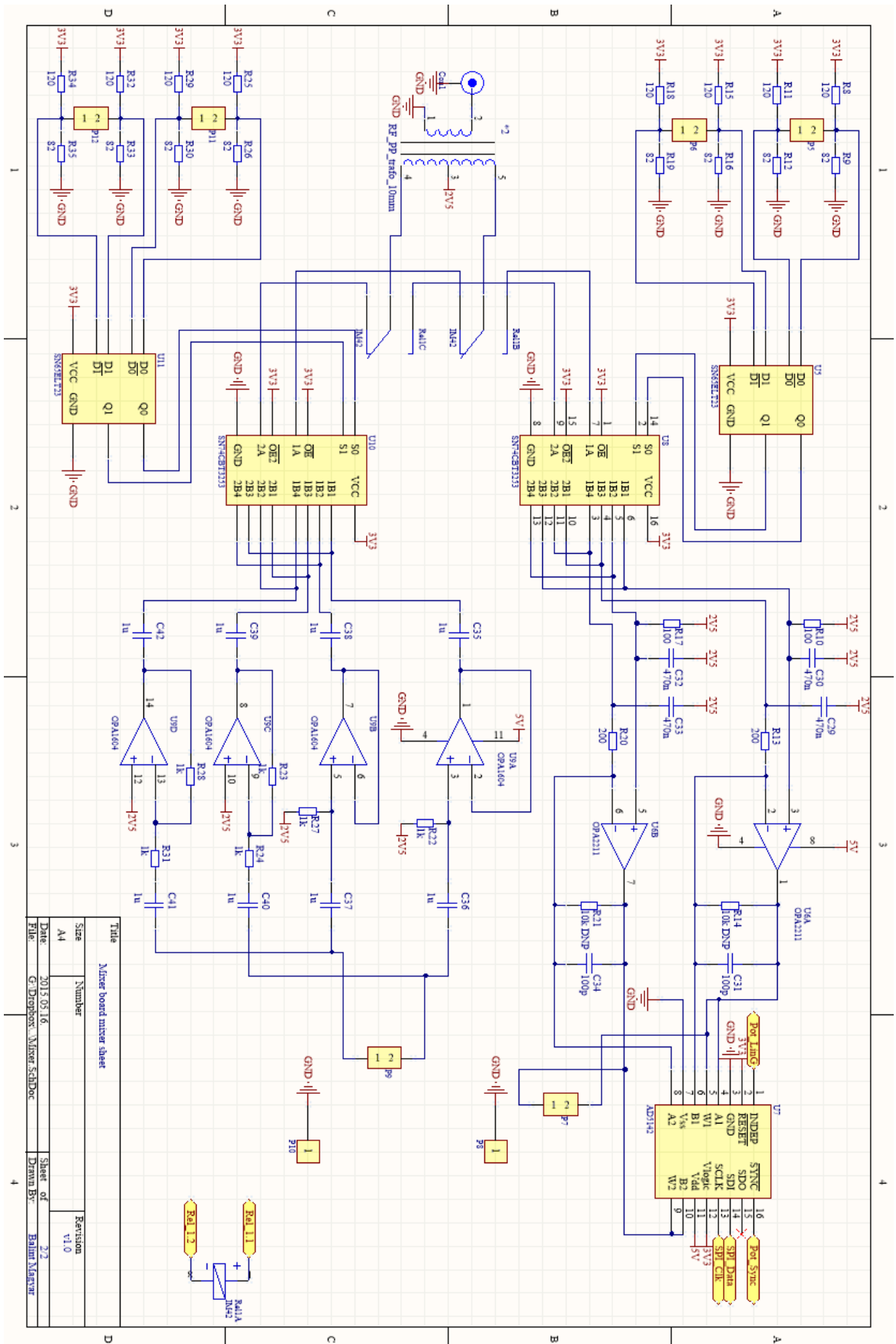


Title		Revision	
Synthesizer board PLL sheet		v1.0	
Size	Number	Revision	
A4		3/3	
Date	2015.05.09	Sheet of	
Drawn By	G. DRONOVSKYI, P.L.L. SCHLICK	DRONOVSKYI	

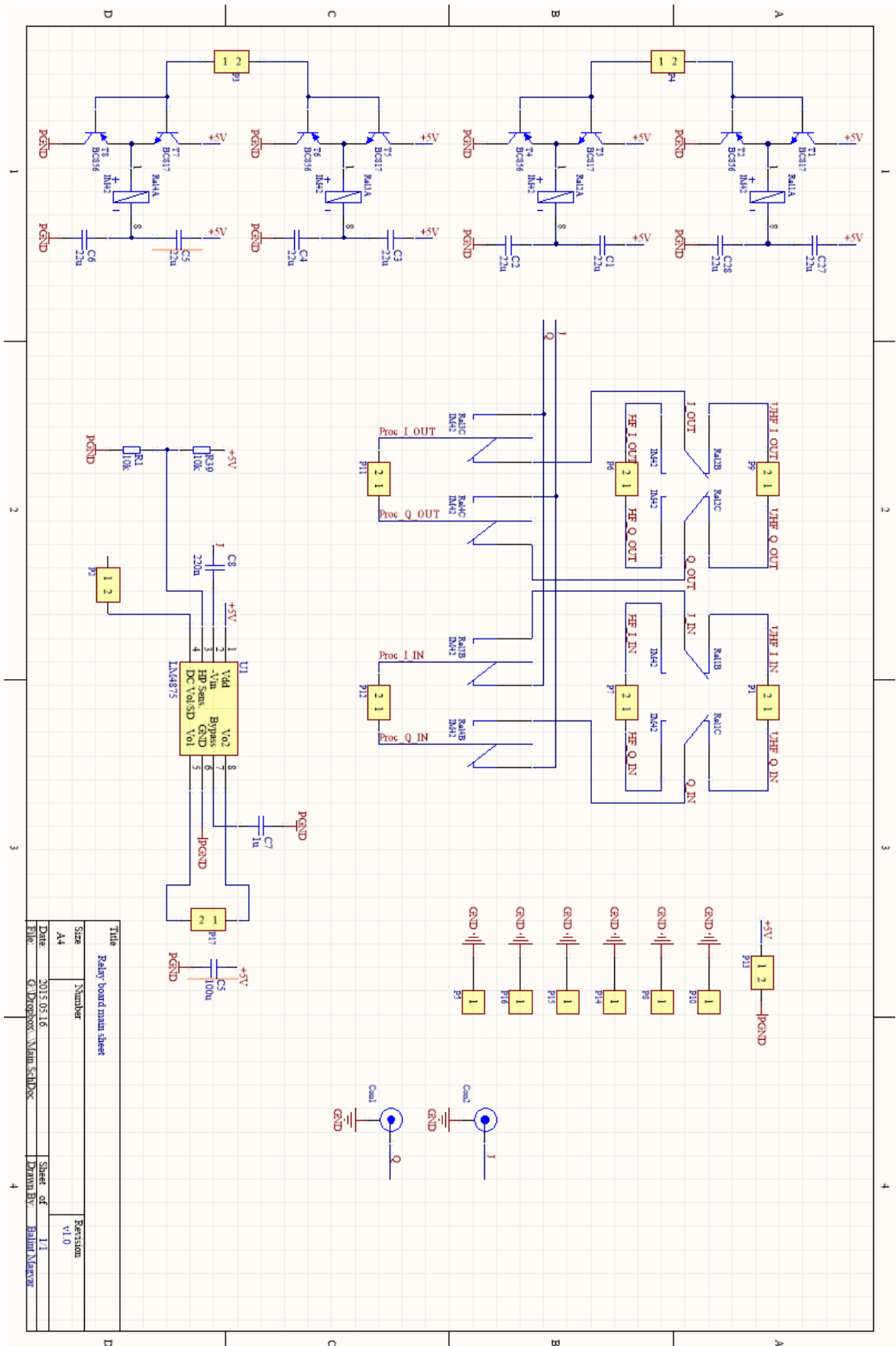
58. ábra PLL kapcsolási rajza



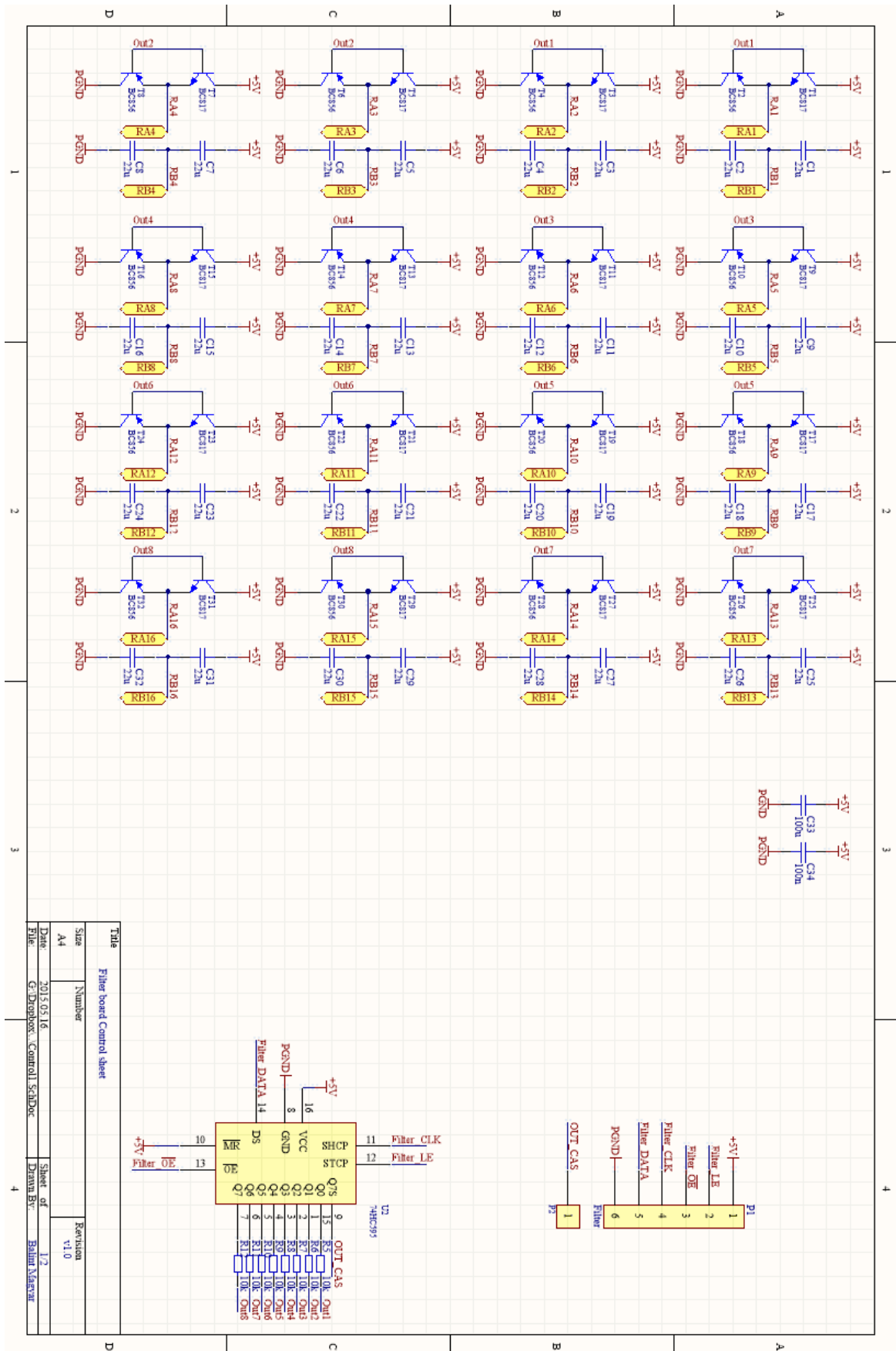
59. ábra Tápegységek kapcsolási rajza



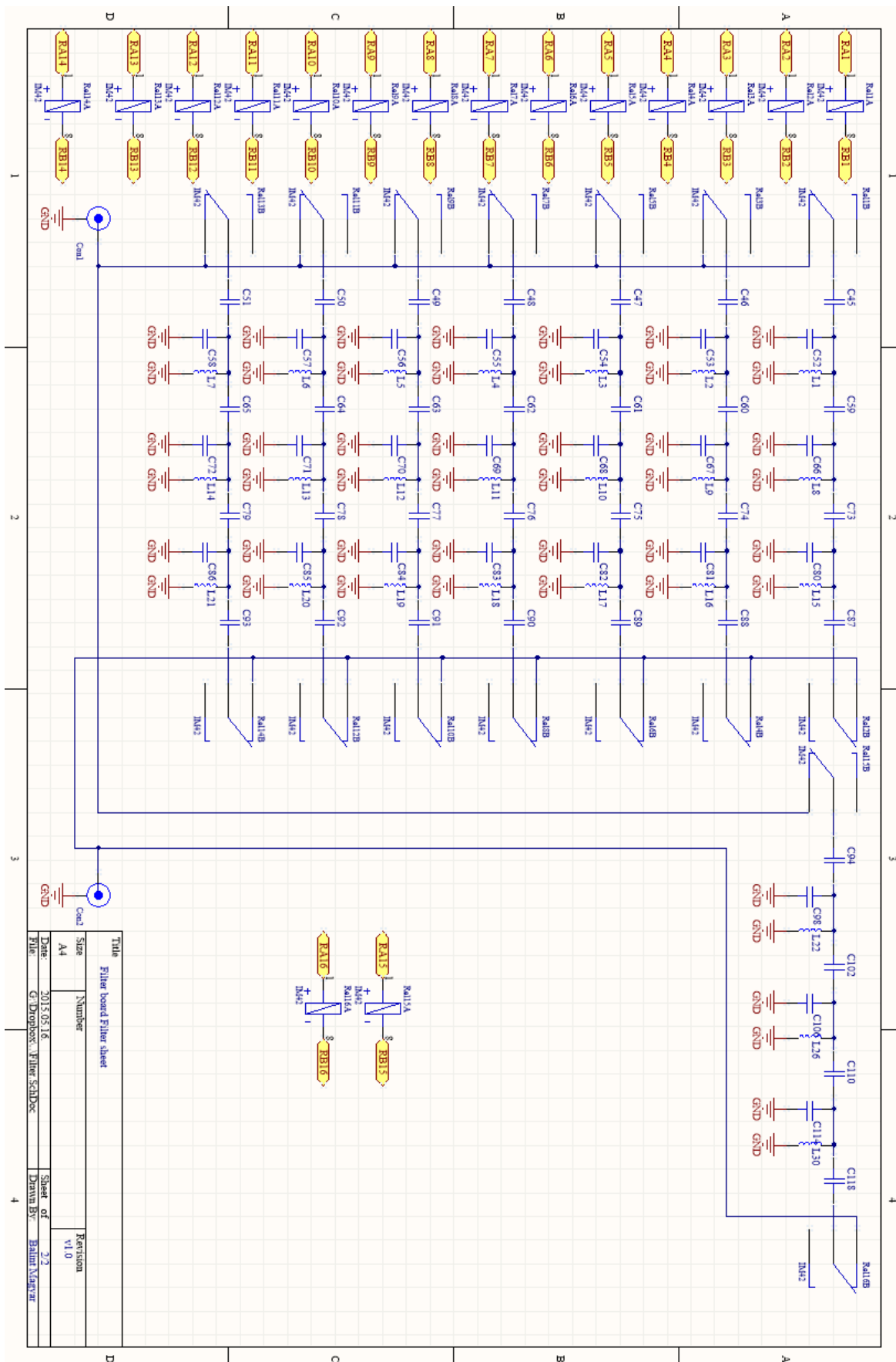
60. ábra Keverő kapcsolási rajza



61. ábra Relé panel kapcsolási rajza



62. ábra Szűrő kapcsológrelék kapcsolási rajza



63. ábra Szűrők kapcsolási rajza