

M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Szélessávú Hírközlés és Villamosságtan Tanszék

Szabó Gábor

Beltéri helymeghatározás kódosztással nyalábolt látható fényes kommunikációval

Tudományos Diákkör dolgozat

Konzulens:

Fehér Gábor

Budapest, 2014.

Kivonat

A LED-es lámpákkal megvalósítható látható fénnel történő kommunikáció (visible light communication, VLC) egy nagyon ígéretes módszer, mely egy lehetséges alternatívájaként szolgálhat bizonyos egyirányú rádiós kapcsolatoknak az egyre inkább telítődő rádiós frekvenciatartomány tehermentesítése érdekében. A látható fényes kommunikáció előnyei beltéri alkalmazások esetén használhatók ki a legjobban. Egy lehetséges felhasználási terület a beltéri helyzetmeghatározás, melyben a több egyforma adótól lévő távolságot a vételi teljesítmények alapján határozhatjuk meg. Ehhez azonban az egyes adók jeleit valahogy el kell tudni különíteni egymástól, amire egy lehetséges megoldás a kódosztásos nyalábolás (code division multiplexing, CDM). E nyalábolási módszer lényege, hogy az azonos sávban lévő különböző forrásból származó jeleket különböző kvázi-ortogonális kódokkal megvalósított digitális modulációval tesszük szétválaszthatóvá. A dolgozatomban bemutatom egy VLC CDM rendszer megvalósításához szükséges lépéseket. Szimulációs modellek alapján megbecsülöm egy ilyen rendszer beltéri helyzetmeghatározáshoz szükséges paramétereit, valamint egy demonstrációs célra kifejlesztett áramkörrel verifikálom a CDM rendszer működőképességét.

Abstract

The visible light communication (VLC) with LED lamps is a very promising technique which can be a possible alternative to certain simplex radio connections aiming to relieve the quickly saturating radio frequency domain. The advantages of visible light communication can be exploited by indoor applications most efficiently. A possible application is indoor positioning, in which the distance of the receiver and a number of similar transmitters (light sources) can be approximated by the signal strengths. To do this the signals of each transmitter must be discriminable. A certain method that ensures the discriminability of signals is code division multiplexing (CDM). In this multiplexing method the channels share a common frequency band but the signals of different transmitters use digital modulation with different quasi-orthogonal codes. In my study I show step-by-step how to realize a VLC CDM system. I approximate the parameters of a positioning system of this type and verify the functionality of CDM scheme with demonstration circuits.

Tartalomjegyzék

Bevezetés.....	5
1 Többszörös csatorna-hozzáférés és VLC	5
1.1 Térosztásos nyalábolás	5
1.2 Időosztásos nyalábolás	6
1.3 Frekvenciaosztásos nyalábolás.....	7
1.4 Kódosztásos nyalábolás.....	7
1.5 Összefoglalás.....	9
2 CDM szimuláció	10
2.1 A kódolás menete	11
2.2 A dekódolás menete	11
2.3 A szimuláció eredményei	13
3 A VLC kapcsolat energiamérlegének meghatározása.....	15
3.1 A vételi sávszélesség meghatározása	16
3.2 A detektor kiválasztása, zajszámítás	16
3.3 A szükséges adóteljesítmény meghatározása	18
3.3.1 A Lux Aeterna fotometriai szimulátor program.....	18
3.3.2 A számítások	22
4 A szoftveres rész realizálása 1	23
4.1 Kódok, emelt koszinusz szűrő.....	23
4.2 Kódolás.....	23
4.3 Adatok kiküldése a hangkimeneten.....	24
4.4 Adatvétel a hangbemeneten.....	24
4.5 Dekódolás.....	25
5 A szoftveres rész realizálása 2	25
5.1 A szoftveres rész működésének ellenőrzése	26
6 A VLC áramkörök.....	28
6.1 Az adóáramkör	28
6.2 A vevőáramkör	30
6.3 Paneltervek	31
Összefoglalás.....	33
Hivatkozások.....	34

Bevezetés

A dolgozatom egy olyan látható fényes kommunikáción (VLC¹) alapuló beltéri helyzetmeghatározási rendszer megvalósítási lehetőségét vizsgálja, mely a csatornák szétválasztására kódosztásos nyalábolást (CDM²) használ. Először ismertetem a nyalábolási módokat a látható fényvel történő szabadtéri kommunikáció szempontjából, hogy kiderüljön miért van értelme egyáltalán a kódosztásos nyalábolással foglalkozni. Ezek után szimuláció segítségével tesztelem a kódosztásos rendszer működését szoftveres oldalról először *Matlab* programmal, majd a valós kommunikációra is alkalmas saját *Delphi*-ben írt könyvtáramat mutatom be és az ezt használó programokkal végzett próba kapcsolatot. A szoftveres részek tárgyalása után elméleti megfontolások és szimulációk alapján meghatározom a látható fényes kapcsolat energiamérlegét és ezt felhasználva a hardveres rész paramétereit is, és ismertetem a tervezett áramköröket. Végül a rendszer működését a tényleges megvalósítással fogom ellenőrizni.

1 Többszörös csatorna-hozzáférés és VLC

A szó szoros értelmében vett látható fényvel történő kommunikáció nem zárja ki azt a lehetőséget, hogy a látható optikai tartományban több különböző hullámhosszon különböző adatokat sugározzunk. A vevő oldalon hullámhossz-szelektív szűrőkkel lenne a jel szétválasztható, minden hullámhosszt külön detektálva. Ez a hullámhosszosztásos nyalábolás (WDM³) vezetett hullámú optikai összeköttetéseknel egy általános módszer [4]. Annak ellenére, hogy az IEEE 802.15.7 szabvány lehetőséget biztosít hasonlóra (CSK, lásd *PHY III* [5]), a VLC-ben ez mégse egy könnyen megvalósítható dolog, mivel a forrásként használt világítótestek széles optikai spektrumú fehér fényű lámpák. Ebből kifolyólag a látható fényes kommunikáció csatornája alatt a tetszőleges összetételű látható fény intenzitásmodulált spektrumát értjük.

Ez a csatorna elméletben az egyenfeszültségtől (0 Hz) szinte tetszőleges frekvenciákig használható. Szélessége gyakorlatilag a fényforrás modulálhatóságának felső határfrekvenciájától és a detektor sávzélességétől függ. A többszörös hozzáférési lehetőségek helyett valójában *nyalábolási* (multiplexálási) lehetőségeket vizsgálunk, amelyek megegyeznek a rádiós kommunikációban használhatókkal. A nyalábolási módok felhasználásával később megvalósítható akármilyen többszörös hozzáférési rendszer, de a magasabb szintű – inkább programozási – problémákkal, mint adatkeretek szervezése, címzés, nem fogok foglalkozni. Megjegyzendő, hogy a célként kitűzött beltéri helyzetmeghatározás esetén elegendő egy azonosító ismételt sugárzása, az említett magasabb szintű feladatokra nincs is szükség.

1.1 Térosztásos nyalábolás

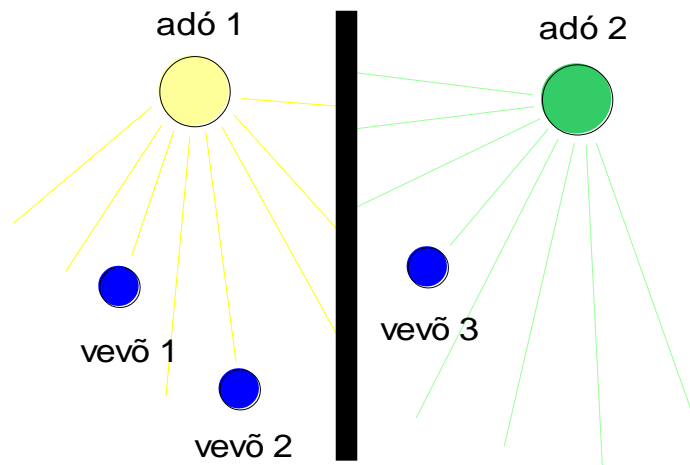
A térosztásos nyalábolás (SDM⁴) azt jelenti, hogy a különböző eszközök közötti kommunikáció egymástól térben elválasztva történik. Ez a látható fényes kommunikáció esetén úgy valósítható meg, ha a külön kommunikáló eszközcsoportok például külön szobában, vagy egy térben de egymástól a látótávolságnál távolabb helyezkednek el. A térosztásos nyalábolás tulajdonképpen nem is valódi nyalábolási módszer egy VLC rendszerben, mivel nem egy helyen valósít meg több párhuzamos kommunikációt.

¹ visible light communication

² code division multiplexing

³ wavelength division multiplexing

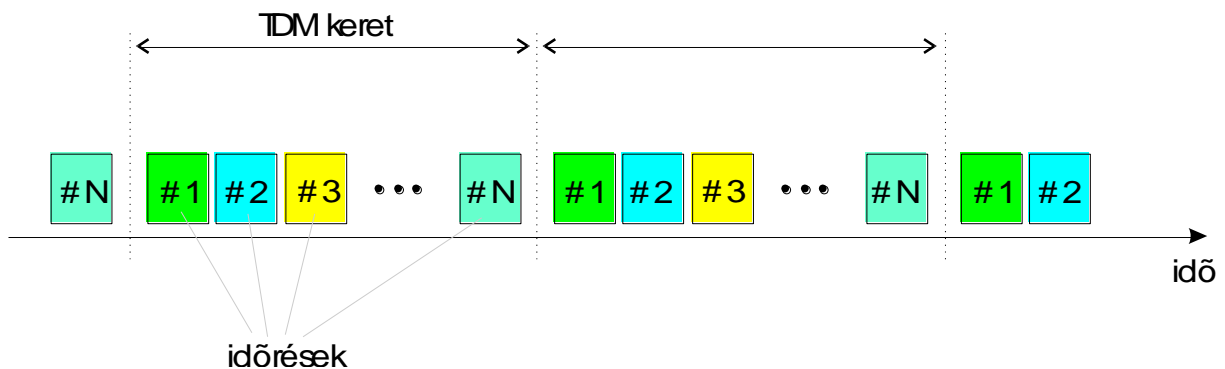
⁴ space division multiplexing



1. ábra: térosztásos nyalábolás

1.2 Időosztásos nyalábolás

Időosztásos nyalábolás (TDM⁵) esetén az egymást látó eszközök úgy képesek kommunikálni, hogy a csatornahasználatot időben felosztják egymás között. Egyszerre mindig csak egy eszköz ad a saját idejében, ilyenkor a többi csendben van és a megfelelő egység(ek) veszi(k) az adást. Minden eszközre sor kerül egymás után, majd az egész periodikusan ismétlődik.



2. ábra: időosztásos nyalábolás

A 2. ábrán látható a TDM adás szerkezete. Azt az időtartományt, mely alatt egy eszköz megszólalhat, időrésnek nevezzük. Az időrészeket a periodikusan ismétlődő TDM keret fogja össze. Minden keretben N darab időrés található, ennyi lesz a maximálisan kezelhető eszközök száma. Általában minden eszköz egy fix, a kereten belül valahányadik, időrésben ad, de előfordulhat olyan is, hogy egy eszköz több időrest is használ egyszerre.

Az időosztásos nyalábolás megvalósításához nagyon fontos a szinkronizáció: minden eszköznek elég pontosan tudnia kell, hogy mikor következik a saját időrése. Az ábrán nincs feltüntetve, de erre a célra szinkronizációs szekvenciákat kell elhelyezni általában a TDM keretek elején. Az időrészek között az átlapolódások elkerülése érdekében biztonsági időszávok vannak.

Az időosztásos nyalábolás olyan esetekben alkalmazható, ahol minden résztvevő eszköz vevő is, mivel a kommunikáció figyelése nélkül nem tudná meghatározni az időrest amiben adhat⁶. Látható fényes kommunikációnál könnyen előfordulhat, hogy több független adó van, amik esetleg egymásra még rálátással sem rendelkeznek, ezért nem előnyös ez a szisztéma

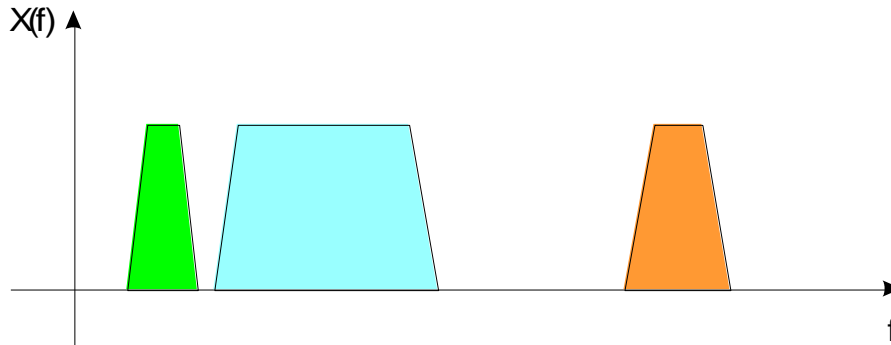
⁵ time division multiplexing

⁶ Az olyan elfajult esetektől eltekintek, amikor az időrészek szélsőségesen hosszúak és ehhez arányosan a biztonsági időszávok is (pl. órák nagyságrendje). Ilyenkor helyi időmérő egységekkel is megoldható a szinkronizáció, ezért nem kell minden eszköznek vevőnek is lenni.

számunkra. Természetesen külső szinkronizációs hálózattal is vezérelhető TDM adás, de az már nem tisztán látható fényes kommunikáció.

1.3 Frekvenciaosztásos nyalábolás

A frekvenciaosztásos nyalábolás (FDM⁷) jelentése, hogy a különböző adások különböző frekvenciasávokon továbbítódnak. Talán ez a legtriviálisabb nyalábolási módszer; ha több akármilyen vivőfrekvenciás modulációt használunk egy időben különböző frekvenciákon, akkor is valójában frekvenciaosztásos nyalábolást valósítunk meg.



3. ábra: FDM spektrum

FDM esetén tehát minden adó különböző frekvencián ad⁸, és a vevők az általuk venni kívánt adást frekvenciatartománybeli szűrővel választják ki. A változatos szélességű sávoknak elég távol kell lenniük egymástól ahhoz, hogy a vevőkben szomszédos sávok belógása nélkül ki lehessen őket választani.

Látható fényes kommunikáció esetén a frekvenciaosztásos nyalábolás a könnyű megvalósíthatóság miatt jól használható, és nem támaszt semmilyen extra követelményt az eszközökkel szemben.

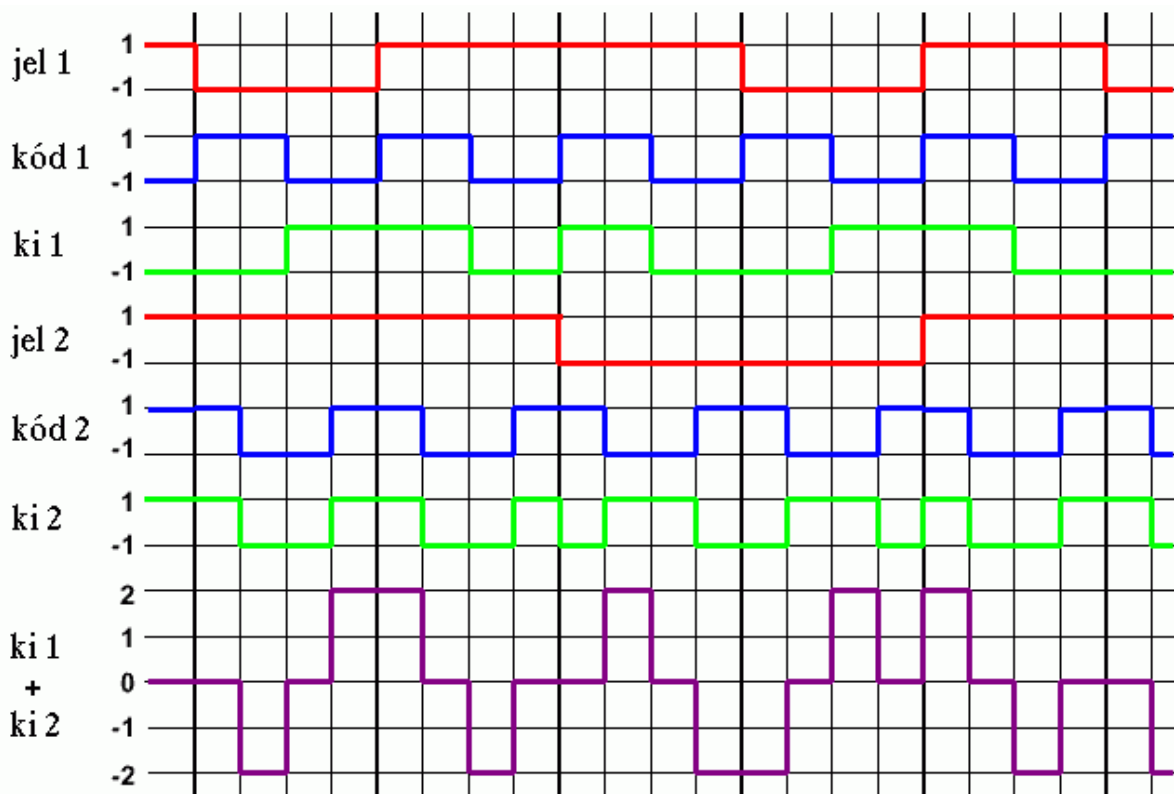
1.4 Kódosztásos nyalábolás

Létezik egy olyan módszer, melynek segítségével akkor is elkülöníthető több jel, ha azok egy helyen, egy időben, egy frekvencián kell hogy kiadásra kerüljenek. Ez a kódosztásos nyalábolás (CDM), ami egy digitális nyalábolási mód, analóg jelek átvitele esetén nem alkalmazható. Bár ez nem kritérium, a gyakorlatban egyelőre szinte kizárólagosan bináris CDM létezik, ezért mi is ezzel fogunk foglalkozni.

A kódosztásos nyalábolás működését talán legegyszerűbb a szinkronizációt igénylő teljesen ortogonális kódok példáján megérteni. Szükség van egy megfelelő (2-hatvány) hosszúságú ortogonális bináris kódkészletre, mely (az adatbitekkel együtt) 1 és -1 reprezentációban értelmezendő. A kódoknál a bit helyett a *chip* szót használjuk az elemi egység elnevezésére. Az ortogonalitás ebben az esetben azt jelenti, hogy bármelyik két kód esetén ha a megfelelő sorszámú chipeket összeszorozzuk és az eredményeket összeadjuk, akkor nullát kapunk (a két kódvektor skaláris szorzata nulla). Ha egy N hosszúságú kódkészletet használunk, akkor minden bit adatra N chip fog jutni a kódoló kimenetén, vagyis a kiterjesztési tényező N . A chipsebesség az eredeti bitsebességnek szintén az N -szerese lesz, ha a (hasznos) bitsebesség állandó. Maga a kódolás úgy történik, hogy a kódot minden adatbittel egymás után megszorozzuk. Ha ezt elvégezzük minden adatfolyamra a saját kódjával, a kimeneteket összeadjuk. A folyamatot a 4. ábra szemlélteti 4-es kiterjesztési tényező esetén.

⁷ frequency division multiplexing

⁸ Egymástól elég távol lévő adók adhatnak azonos frekvencián, mint például a mobiltelefon-hálózatok cellás rendszerében.

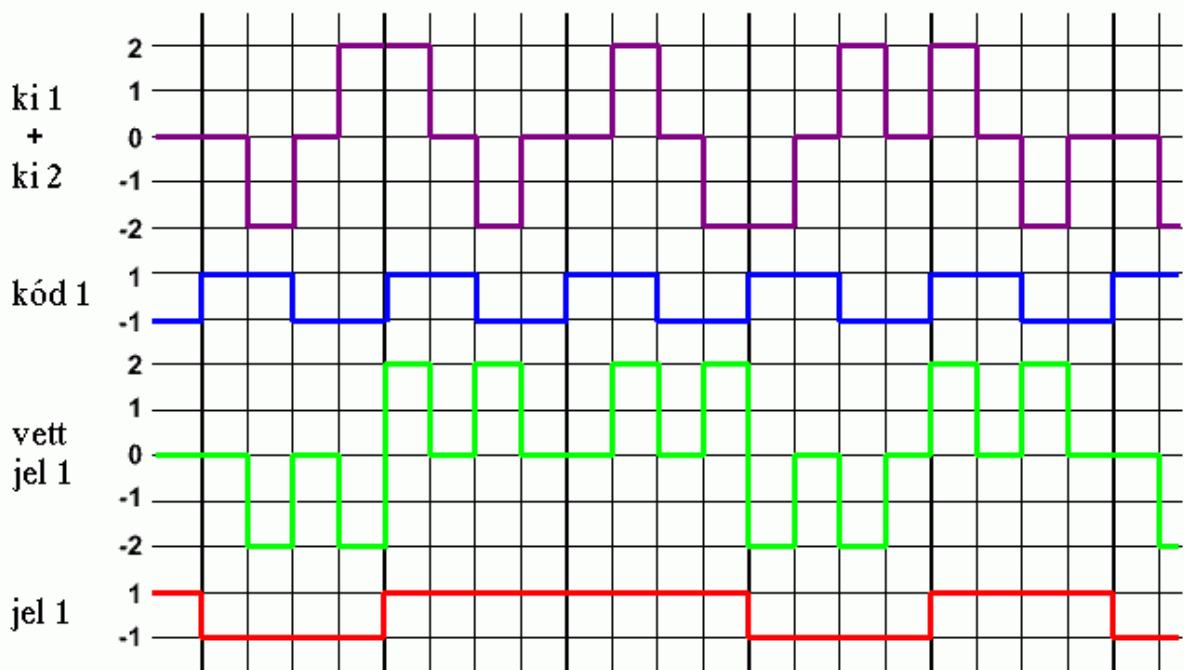


4. ábra: kódolási példa 4 bit hosszú kódokkal⁹

A két kód, $(1, 1, -1, -1)$ és $(1, -1, -1, 1)$, ortogonalitása ellenőrizhető: a skaláris szorzat: $1+(-1)+1+(-1)=0$. A kimenetek összegeként előállt jelből ezután úgy kaphatjuk vissza valamelyik eredeti jelet, hogy a hozzá tartozó kóddal megszorozzuk az összegjelet, és a kapott eredményt bitidőnként átlagoljuk (5. ábra).

Mint már korábban említettem, ortogonális kódkészlet alkalmazása esetén kötelező a szinkronizáció, mivel ortogonális kódok különböző eltoltjai már nem lesznek ortogonálisak. Így ilyen kódok főleg például egy adó által sugárzott különböző jelek nyálábolására alkalmasak. Viszont ortogonális kódok helyett kvázi-ortogonális kódkészletet is használhatunk (például Gold-kódok), melyekben a kódok egymás eltoltjaira is kvázi-ortogonálisak. Így nem lesz szükség szinkronizációra. Hátrányuk viszont az ilyen kódoknak, hogy a nem tökéletes ortogonalitás miatt az egyik jel dekódoláskor a többi jelet a dekóder kis zajként érzékeli.

⁹ A kép forrása: Csapaki Gyula, Németh Krisztián: *Hálózati technológiák és alkalmazások jegyzet*



5. ábra: a dekódolás szemléltetése¹⁰

Megjegyzendő, hogy a kódosztásos nyálábolás egyben spektrumkiterjesztést is végez. Ez könnyen belátható úgy, hogy ha egy bit helyett N chipet adunk ki, akkor az eredeti legfeljebb f frekvenciájú négyszögjelek helyett legfeljebb $N \cdot f$ frekvenciájú négyszögjelek fordulhatnak elő a kimeneten (ezért hívhatjuk az N kódszóhosszt kiterjesztési tényezőnek is). A spektrumkiterjesztés csökkenti a kommunikáció keskenysávú zavarokra való érzékenységét cserébe a nagyobb spektrális helyfoglalásért, viszont a közös sávon osztozó csatornák miatt egy zavar minden csatornát megzavar azonos mértékben.

A kódosztásos nyálábolás még egy nagy előnye, hogy az eszközök által használt kódok nagyon egyszerűen átírhatók, így a fizikai paraméterek (pl. frekvencia) változtatása és az ilyenhez szükséges bonyolultabb áramkörü elemek (pl. frekvenciaszintetizátor) nélkül is megvalósítható az eszközök konfigurálása.

Ha látható fényes kommunikációt szeretnénk CDM-mel megvalósítani, akkor a több adó adásának elkülönítésére kvázi-ortogonális kódkészletet kell választanunk. (Tökéletesen ortogonális kódkészlet esetén biztosítanunk kellene az adók közötti szinkront, ami azért nehézkes, mert VLC esetén az adók gyakran egymástól teljesen független eszközök.) A kódolt adatfolyamot aztán kiadhatjuk alapsávban OOK vagy impulzus pozíció modulációval¹¹, továbbá akármilyen vivőfrekvenciás modulációval. Utóbbi esetben hibrid rendszer is könnyen megvalósítható, melyben a CDM átvitel mellett külön kommunikáció folyhat FDM-ben.

1.5 Összefoglalás

Bár egyesek foglalkoznak külső szinkronizációval megvalósított VLC TDM kommunikációval is [1], a tiszta látható fényes kommunikáció szempontjából a triviális megoldáson (SDM) kívül szinte csak az FDM és a CDM jöhet szóba, vagy az ezekből alkotott hibrid rendszer. A két módszer összehasonlítása a következő táblázatban látható.

¹⁰ A kép forrása: Csopaki Gyula, Németh Krisztián: *Hálózati technológiák és alkalmazások jegyzet*

¹¹ Az ilyen rendszereket OCDMA-nak is nevezik.

FDM	CDM
digitális és analóg átvitel egyszerűbb	csak digitális átvitel bonyolultabb (kódolási/dekódolási programozási feladatok)
konfigurálási lehetőség bonyolultabb (pl. frekvenciaszintézis szükséges)	egyszerűen konfigurálható (kód átírása a programban)
arányos sáv szélesség (több csatorna → nagyobb), normál spektrumkihasználás	fix sáv szélesség (adatsebességtől és kódszóhossztól függ), gyakran rosszabb spektrumkihasználás (csatornák számától függ)
normál zavarérzékenység	kisebb zavarérzékenység, viszont egy interferencia minden csatornát megzavar

Mindkét rendszernek megvannak a maga előnyei és hátrányai, csak konkrét esetben dönthető el, hogy éppen melyiket célszerűbb alkalmazni. Beltéri helyzetmeghatározás esetén a kódosztásos rendszer talán legszembevetőbb előnye, hogy ugyanaz a viszonylag egyszerű hardver használható minden jelforrás esetén.

2 CDM szimuláció

Ebben a fejezetben a kódosztásos nyálábolás megvalósításának módját ismertetem. Mivel a nem szinkronizált kvázi-ortogonális kódkészleten alapuló adások dekódolásának gyakorlatával kapcsolatban nem találtam túl sok mindent, ezért az algoritmusokat saját magam fejlesztettem ki. Ebben segítségemre volt az egyik egyetemi jegyzet is [2].

A kódosztásos adás és vétel működőképességét először *Matlab* szimulációval ellenőriztem. A paraméterek definiálása előtt viszont szükségem volt a kommunikáció sáv szélességének meghatározására, hogy egy gyakorlatban is könnyen megvalósítható rendszert tudjak szimulálni. Arra az elhatározásra jutottam, hogy a számításigényes kódolás és dekódolás miatt az adás és a vétel szoftveres részét egy *személyi számítógép* fogja megvalósítani. Hogy ne kelljen semmilyen különleges eszköz, azt találtam ki, hogy az adatbemenet és -kimenet a *hangkártyán* keresztül fog megvalósulni. Mivel manapság akármilyen hangkártya képes a szabványos 44100 Hz-es mintavételezésre 16-bites felbontással, ezért az általam választott adatfolyam 44100 Hz-es 16-bites mono audió lett. Ebből adódóan a *maximális elméleti sáv szélesség 22050 Hz*, ezzel valósítottam meg a szimulációt.

Mivel az alapsávi sáv szélesség már amúgy is elég kicsi, ezért vivőfrekvenciás digitális moduláció nem igazán jöhet szóba. A moduláció típusának ezért egyszerű OOK-t választottam. Az OOK jelek sáv szélessége a négyszöges jelek felharmonikusai miatt elég nagy (elméletileg végtelen), de a sávkorlátozás csak a Nyquist-szűrővel valósítható meg ha el akarjuk kerülni a szimbólumközi áthallást (a csatorna átvitelét tekintsük frekvenciafüggetlennek). Erre megfelelő a távközlésben gyakran használt emelt koszinusz szűrő.

Másik kritikus dolog a megfelelő kódkészlet kiválasztása. A kódokat nem magam generáltam, mivel találtam egy olyan kódkészlet-adatbázist¹², ahol mindenféle lehetséges kódkészletből sokféle hosszúságú elérhető, így ezeket használtam fel. A szimulációk során kiderült, hogy az amúgy jó autokorrelációs és keresztkorrelációs tulajdonságokkal rendelkező kódkészlet típusa (maximum hosszúságú álvéletlen kódok, Gold-kódok, Kasami-kódok) nincs nagy hatással a dekódolhatóságra. Ezzel ellentétben a kódok hosszúsága nagyon fontos paraméter. Minél hosszabb a kód, annál kevesebb adat vihető át ugyanakkora chipsebesség mellett, viszont

¹² http://www.ee.oulu.fi/~kk/optim_codes_info.html

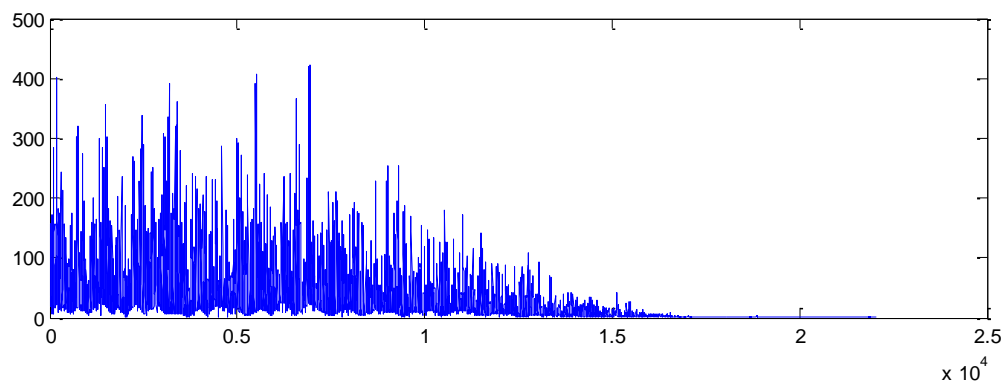
állandó számú egyidejű adás esetén annál jobb lesz az egyes adások kiválaszthatósága és dekódolhatósága.

2.1 A kódolás menete

A kódoláshoz és dekódoláshoz egyaránt szükséges, ha a bináris kizáró vagy műveletet számok összeszorzásával szeretnénk helyettesíteni (a bitek a könnyű kezelhetőség érdekében egész szám elemű tömbökben vannak), hogy a bináris adatok 1/0 reprezentáció helyett +1/-1 reprezentációban legyenek. Van egy ilyen formában adott valahány elemű kódkészlet, és egy legfeljebb ugyanennyi elemű tesztadatsor készlet. A végső mintavételi frekvencia – mivel a hangkimenet szimulálása a cél – 44100 Hz, ezért a chipsebességet a maximálisan lehetséges 22050 Hz-re kell választani. Ebben az esetben emelt koszinusz szűrővel a kimeneti jel sávszélessége biztosan legfeljebb 22050 Hz lesz.

Legyen a kódhossz C , és egy tesztadatsor hossza T . Egy tesztadatsor minden elemét összeszorozzuk egy kódvektorral, melynek eredményeképpen egy $C \cdot T$ hosszúságú kódolt adatvektort kapunk. Ezt az adatvektort kétszeresen túlmintavételezzük hogy megkapjuk a 44100 Hz-es mintarátát, majd az előzetesen előállított emelt koszinusz FIR szűrővel digitális szűrést hajtunk végre (konvolúció). Az emelt koszinusz szűrő lekerekítési tényezője a szimulációban $3/5$, ami 17 kHz körüli sávszélességet eredményez, és az impulzusválasza se túl hosszú.

Egy ilyen kimeneti jel spektruma a 6. ábrán látható.



6. ábra: egy véletlen kimeneti jel spektruma (kódhossz: 511), az X-tengelyen a frekvencia található Hz-ben

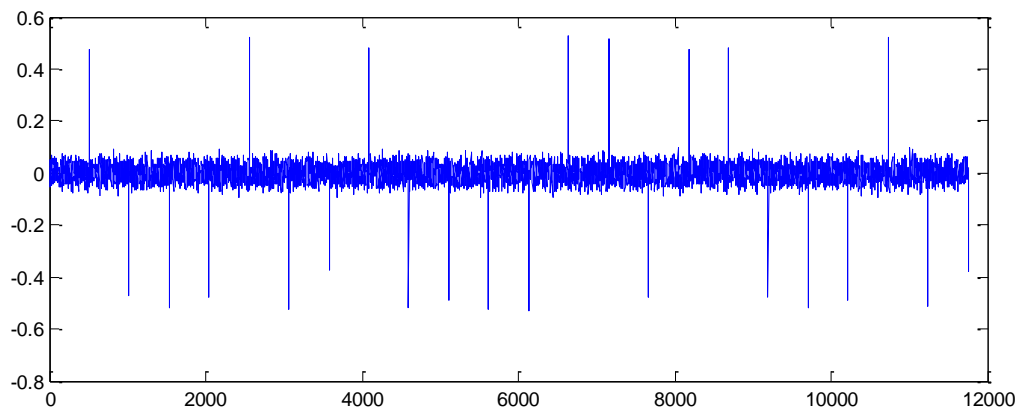
A kimeneti jelek generálása után az összes kimenetet véletlenszerű eltolással szuperponáljuk (összeadjuk) tetszőleges súlyokkal. Az alapértelmezett súlyozás egyenletes, vagyis minden jel egyenlő mértékben szerepel majd a kimenetben.

Az összegzett kimeneti jelekhez, hogy egy valóshoz hasonló bemenetet hozzunk létre a dekódoló számára, egyenletes additív zajt is adhatunk, melynek átlagértéke beállítható a jel átlagértékéhez képest. A zaj hozzáadása úgy történik, hogy a mintasor hosszával megegyező hosszúságú egyenletes eloszlású véletlen számokból álló vektort generálunk a megfelelő maximális értékkel, majd ezt hozzáadjuk a mintasorhoz.

2.2 A dekódolás menete

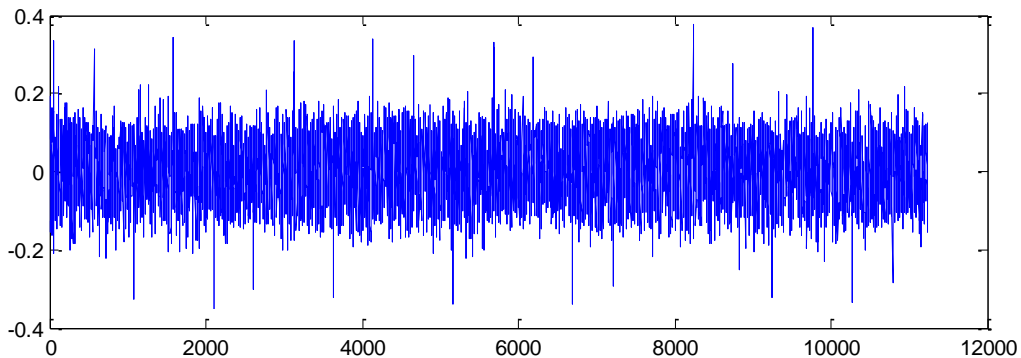
A dekódolás előtt a 44100 Hz-es jelet újramintavételezem 22050 Hz-re, hogy egy minta egy chipnek feleljen meg. Ezután következik a csúszó ablakos dekódolás: a bemeneti mintasoron egy kódszóhossz méretű ablakot csúsztatunk végig mintánként lépkedve. Minden lépésben kiszámoltatjuk a keresztkorrelációt az ablak tartalma és az összes szóba jöhető kód között. Az így létrejövő adatsorokban (minden kódhoz tartozik egy) korrelációs csúcsok keletkeznek azokon a helyeken, ahol az adott kóddal szorzott bitnek kezdő pontja van. A korrelációs

csúcsok jó időzítés esetén kódszóhossznyi távolságra vannak egymástól, előjelük pedig megadja az adott bit értékét. Ábrázolva egy ilyen adatsort a 7. ábrához hasonló grafikont kell kapnunk.



7. ábra: a dekódolás során keletkező adatsor egy „szép” esete

Ezek után már csak magukat a biteket kell visszaállítani, ami a 7. ábrához hasonló tiszta esetben nem tűnik bonyolult feladatnak, de kevésbé ideális helyzetben nem triviális dolog. Például a 8. ábrán látható adatsorban már nem emelkednek ki annyira a korrelációs csúcsok, bizonyos helyeken „álcúcsok” is láthatók, és semmi garancia nincs arra hogy a legnagyobb csúcsok lesznek a korrelációs csúcsok. Ettől függetlenül lehetőség szerint egy ehhez hasonló adatsorból is illene egy jó algoritmussal visszaállítani a biteket.



8. ábra: a dekódolás során keletkező adatsor egy lehetséges esete

Az általam kitalált bitvisszaállító algoritmus a következőképpen működik:

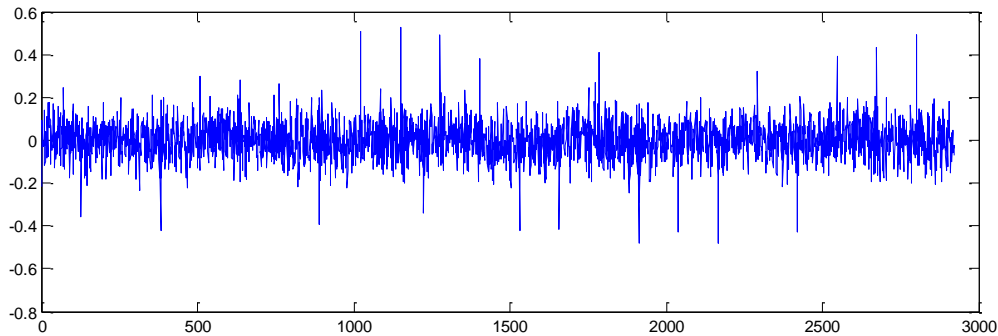
- Megkeresi a legnagyobb abszolút értékű csúcsot a vizsgált adatsorban, és feltételezi, hogy ez egy korrelációs csúcs.
- Az abszolút maximum helyéből mindkét irányban kijelöli a korrelációs csúcsok várható helyét, feltételezve hogy azok kódszóhossznyi távolságra vannak egymástól.
- A kijelölt csúcsok R sugarú környezetében megkeresi az abszolút maximumot, és azt feltételezi hogy az a keresett korrelációs csúcs. Erre azért van szükség, mert 1 mintányi elcsúszás még egész könnyen előfordulhat. Az R paraméter tipikus értéke 1-2.
- A feltételezett korrelációs csúcsok előjele alapján visszaadja a biteket.
- A feltételezett korrelációs csúcsok abszolút értékeinek és az egész mintasor abszolút középértékének arányait átlagolva egy minőség paramétert határoz meg, ami az adott kódhoz tartozó „csatornát” jellemzi.

A szimuláció során ehhez még egy olyan kiegészítő eljárást is készítettem, ami automatikusan leellenőrzi a visszakapott bitek helyességét (a szimulációban természetesen elérhetőek a tesztadatsor eredeti bitjei).

2.3 A szimuláció eredményei

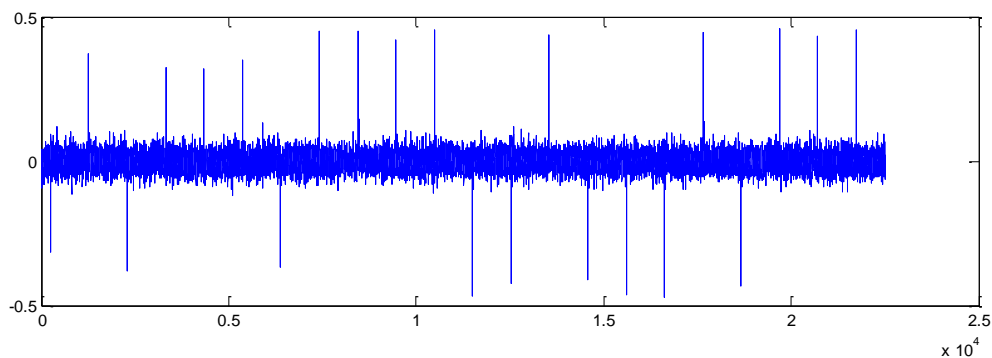
A szimulációk során – azt leszámítva ha kifejezetten ezen paraméterek változtatásának hatását akartam vizsgálni – végig 4 egyidejű egyenlő átlagteljesítményű adással és zajmentes közeggel számoltam. A vétel jóságát a dekódolt adatsor ábrázolásával és szemrevételezésével vizsgáltam.

Legelőször 127 hosszúságú kódkészlettel szimuláltam, mely kb. 160 bps bitrátát¹³ tesz lehetővé. Ebben az esetben mindenféle egyéb bonyolítás nélkül is eléggé zajba veszett a korrelációs csúcsok egy része (9. ábra), ezért megpróbálkoztam nagyobb kódszóhosszakkal.



9. ábra: egy tipikus kimenet 127 hosszúságú kódok esetén

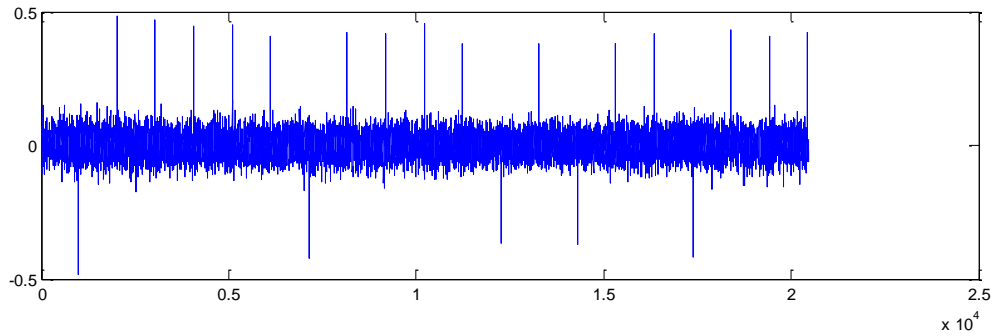
Az egyidejű adások számának változatlanul hagyásával a korrelációs csúcsok kiemelkedése a kódszóhossz növelésével a vártaknak megfelelően nőtt. 1023 hosszúságú kóddal a kimenet már kielégítően „szépnek” mondható (10. ábra), ezért a továbbiakban ezzel a kódkészlettel számoltam. Megjegyzendő viszont, hogy ebben az esetben a bitráta mindössze 20 bps. Ahogy azt korábban említettem, a kódkészlet típusának megváltoztatása nem okozott szemmel látható változásokat a kimenetben.



10. ábra: egy tipikus kimenet 1023 hosszúságú kódok esetén

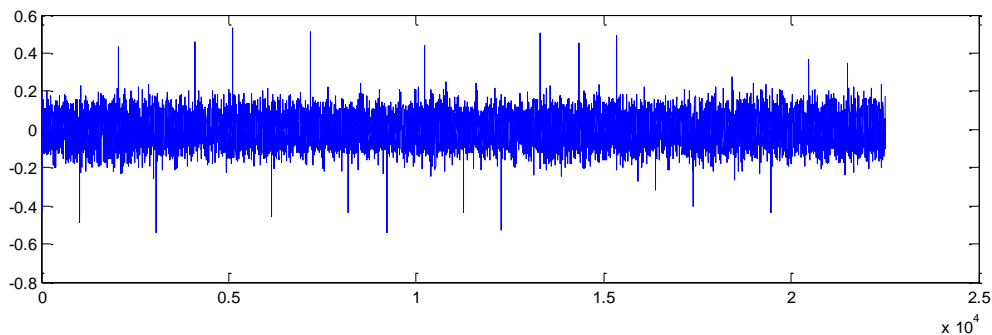
Megpróbálkoztam az adások számának növelésével is. Amint az a 11. ábrán látható, az egyidejű adások számának növelése nem rontja túlságosan a detektálhatóságot. Az adások számát kétszeresére (8-ra) növelve a középső zajos sáv körülbelül arányosan szélesedik ki, viszont a korrelációs csúcsok továbbra is kiugróak maradnak.

¹³ A számítás módja: bitsebesség = chipsebesség / kódszóhossz.



11. ábra: egy tipikus kimenet 1023 hosszúságú kódok és 8 egyidejű adás esetén

Ezzel szemben ha az adások nem egyenlő erősségűek, a minőségromlás a gyengébb adások vétele esetén sokkal látványosabb lesz. A következő példában a 4 adás közül az egyik erősségét 2-szeresére, egy másikat 4-szeresére növeltem¹⁴; az 12. ábra az egyik eredeti erősségű csatornát mutatja.

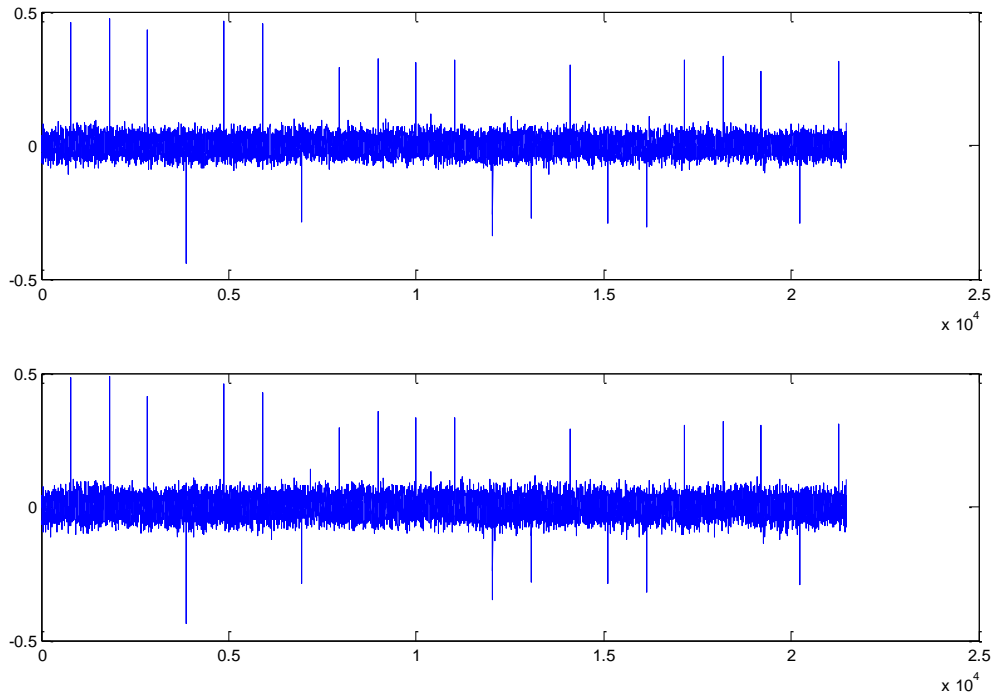


12. ábra: egy tipikus kimenet 1023 hosszúságú kódok esetén, ha másik adások erősségét néhányszorosára növeltük

A zavaró jelek növekedése átlagosan megegyező mértékű volt a 8 csatornás esethez képest (+4 adásnyi), de ebben az esetben láthatóan sokkal erőteljesebb romlás következett be. Ez valószínűleg azzal magyarázható, hogy a különböző adások számának növelésekor statisztikus jellegű összegződés következik be, míg az egyes csatornák erősségének növelésekor „legrosszabb eset” típusú addíció történik.

Ellenőriztem az additív zaj hatását is: a dekódoló bemeneti jeléhez a korábban ismertetett módon olyan zajt adtam, melynek várható értéke a jel átlagértékének fele volt. Ez körülbelül 6 dB jel-zaj viszonynak felel meg. A 13. ábrán látható, hogy ez a jelentős mértékű zaj a dekódolt adatsort csak alig észrevehetően rontotta. Ez azt jelenti, hogy a CDM átvitel zajtűrő képessége igen nagy.

¹⁴ amplitúdóban értve



13. ábra: ugyanaz a kimenet zaj nélkül (felső) és 6 dB-es jel-zaj viszonyal (alsó)

A szimulációk alapján a legnagyobb problémának az tűnik, hogy a nagyobb teljesítményű adások könnyen elnyomhatják a gyengébbeket, például ha az egyik adó jóval közelebb van a vevőhöz mint a többi. Mivel frekvenciatartományban közös sávban mennek a jelek, ezért automatikus erősítésszabályzás jellegű dolgok nem valósíthatók meg. Viszont a kis adatsebességnek köszönhetően szóba jöhet olyan megoldás, mellyel az elnyomó adások zavaró hatása valamelyest kiküszöbölhető. Mivel az elnyomó adás jele teljesen biztosan vehető, ezért elméletileg megvalósítható lenne az, hogy a vett biteket az adáshoz tartozó kóddal a vételi oldalon ismét kódoljuk, majd az így keletkező mintasort kivonjuk a bemeneti mintasorból a megfelelő eltolással. Ezután a dekódolást újra végrehajtva az elnyomott adások könnyebben detektálhatóak lesznek.

Próbálkoztam egy ilyen algoritmus realizálásával, de a dolog az általam kitalált formában problémákba ütközött. Voltak bizonyos esetek, amikor ez a bonyolultabb eljárás javított az eredményeken, viszont sok esetben romlás következett be. A probléma oka az időzítési hibákban találtatott meg. Egy 1 mintányi elcsúszást az egyszerű algoritmus még simán elvisel, viszont ha a regenerált adás kivonása véletlenül ennyivel elcsúszik, akkor tulajdonképpen nem kivonás történik, hanem egy plusz jel hozzáadása. Ez főleg azért kellemetlen, mert az ilyen minimális időzítési hiba gyakran a vizsgált mintasor közben is változhat, tehát előfordulhat olyan eset, amikor a mintasor egyik felénél a kívánt hatást váltja ki a művelet, a többi részénél viszont az eltolódás közbenső változása miatt romlás következik be. A jelenség miatt úgy döntöttem, hogy a végső változatba csak az egyszerű dekódoló eljárás fog bekerülni, de ez az algoritmus véleményem szerint a későbbiekben még kísérletezésre érdemes.

3 A VLC kapcsolat energiamérlegének meghatározása

Ahhoz, hogy a látható fényes kommunikáció megvalósításához megfelelő áramköröket lehessen tervezni, ismerni kell a kommunikáció „energiamérlegét”. Más szavakkal tudni kell, hogy mekkora teljesítménnyel kell sugározni ahhoz, hogy egy adott detektoron a zajból kielégítően kiemelkedő jelet kapjunk. Ez a forrásteljesítmény-igény nagyban függ a fotodetektor méretétől és annak iránybeállításától, iránykarakterisztikájától, továbbá a

fényforrás és a detektor távolságától. A megfelelő detektor jelszint meg függ a bemeneti zajtól, amit a fotodióda paraméterei és a vételi sávszélesség is befolyásolnak¹⁵. Látható, hogy sok minden sok mindentől függ, ami miatt fontos, hogy néhány dolgot a szükséges fénytjeljesítmény számítása előtt meghatározzunk.

3.1 A vételi sávszélesség meghatározása

A fotodetektor (és az áramkörök) zaja erősen és nyilvánvalóan függ a vételi sávszélességtől, ezért először ezt célszerű meghatározni. Ahogy a CDM szimulációs részben már kiderült, a számítógép a hangkártyán keresztül fog kommunikálni az alacsony szintű analóg áramkörökkel 44100 Hz-es mintavételi frekvencia mellett, ami azt jelenti, hogy az áramköröknek legfeljebb a 0..22050 Hz-es sávot kell tudni átvinni. Annak ellenére, hogy így maga a hangkártya is végez valamiféle aluláteresztő szűrést, a nagyobb frekvenciás zavarok ugyanúgy rontják a vételi minőséget. Arra viszont kicsi az esély, hogy egyszerűbb áramkörökkel éles kb. 25 kHz-es sávszűrést lehessen végrehajtani, úgyhogy érdemes inkább rosszabb esettel számolni. Vegyük a vételi sávszélességet kb. 200 kHz-nek, és akkor nagy valószínűséggel nem fogjuk a zajt alábecsülni.

3.2 A detektor kiválasztása, zajsámítás

Egy olyan detektorra van szükség, ami kellően érzékeny a látható fényes tartományban, és minél nagyobb látószöggel rendelkezik. Az se hátrány, ha érzékeny felülete is minél nagyobb, de csak addig a pontig, amíg az ebből adódó sávszélesség-csökkenés nem zavaró.

Megfelelő választásnak tűnik az Excelitas Technologies cég VTP1188SH fotodiódája. Ez egy átlátszó lencsés tokozású, viszonylag nagy érzékeny felületű, széles látószögű PIN fotodióda. Fényképe a 14. ábrán látható.



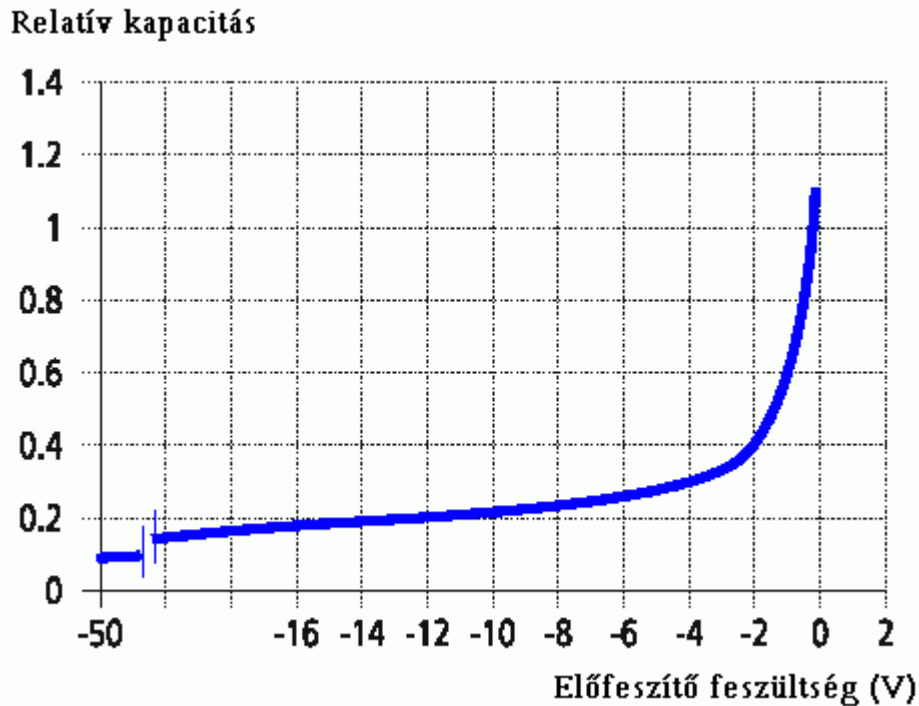
14. ábra: a VTP1188SH fotodióda fényképe

Elektromos paramétereit (25°C-on) a következő táblázat foglalja össze.

Érzékeny felület	11 mm ²
50%-os látószög (2φ)	120°
Rövidzárási áram (min.)	200 μA
Sötétáram (max.)	30 nA
Kapacitás előfeszítés nélkül (max.)	300 pF
Érzékenység λ _p -nél (tip.)	0,55 A/W
Működési hullámhossztartomány	400..1100 nm
Legnagyobb érz. hullámhossza (λ _p)	925 nm

¹⁵ Az esetleges áramköri zajoktól egyenlőre tekintsünk el.

Látható, hogy a fotodióda viszonylag nagy sötétárammal és kapacitással rendelkezik. A sötétáram egyenáram; a látható fényes kommunikáció során az egyen-fotoáramot létrehozó modulálatlan fény nem érdekes, ezért egyenáramot nem kell mérnie egy VLC detektornak. Ebből adódóan a sötétáram mint paraméter csak a zaj szempontjából érdekes, egyébként nem okoz problémát. A nagy chipméret miatti nagy kapacitás számottevően lejjebb csökkenthető záróirányú előfeszítéssel (15. ábra), de a csökkentett érték se lesz túlságosan kicsi.



15. ábra: a VTP1188SH fotodióda kapacitása az előfeszítés függvényében

Ahhoz, hogy a detektor sáv szélessége elég nagy legyen ekkora kapacitásérték mellett is, a fotodióda előerősítőjének bemeneti impedanciája nem lehet túlságosan nagy. Egy megfelelő transzimpedanciás erősítővel ez a probléma kiküszöbölhető.

Mivel a fotodióda áramgenerátoros forrás, ezért a zaj is áramként generálódik rajta. Egy fotodiódának (mint más eszközöknek) kétféle zaja van: a sörétzaj és a termikus (Johnson-) zaj.

A termikus zajáram képlete:

$$I_{\text{termikus}} = \sqrt{\frac{4kTB}{R_{\text{sh}}}}, \text{ ahol}$$

k a Boltzmann-állandó (kb. $1,38E-23$),
 T az eszköz hőmérséklete (kelvin),
 B a zajmérés sáv szélessége,
 R_{sh} a fotodióda párhuzamos ellenállása.

A sörétzajáram számításának módja:

$$I_{\text{sörét}} = \sqrt{2e(I_{\text{foto}} + I_{\text{sötét}})B}, \text{ ahol}$$

e az elemi töltés (kb. $1,602E-19$),
 I_{foto} a fotoáram,
 $I_{\text{sötét}}$ a sötétáram,
 B a zajmérés sáv szélessége.

A fotodióda párhuzamos ellenállása legrosszabb esetben $10 \text{ M}\Omega$ körül lehet, a hőmérsékletet vegyük 300 K -nek. A sötétáram ugyan csak 30 nA , de a fotoáram akár $100 \mu\text{A}$ is lehet egy erősebben megvilágított helyen, úgyhogy ez utóbbi lesz a domináns. A sáv szélességet a korábban meghatározott 200 kHz -nek vehetjük.

A megadott értékekkel a képletekbe helyettesítve a két zajáram effektív értéke:

$$I_{\text{sörét}} \approx 2,5 \text{ nA}$$

$$I_{\text{termikus}} \approx 18 \text{ pA}$$

A termikus zajáram két nagyságrenddel a sörétzajáram alatt van, úgyhogy elhanyagolásával nem okozunk nagy számítási hibát. Ezért a felvázolt körülmények között a fotodióda zajárama körülbelül $2,5 \text{ nA}_{\text{eff}}$. Ha a vételi sáv szélesség 2 MHz lenne (azaz a jelenlegi 10-szerese), a zajáram effektív értéke $\sqrt{10}$ -szeres, azaz körülbelül 7,9 nA lenne.

3.3 A szükséges adóteljesítmény meghatározása

Az adóteljesítményt úgy kell meghatározni, hogy a fotodióda kimenetén a jel-zaj viszony elég jó legyen. Mivel adott modulálható fényforrások állnak rendelkezésünkre (legfeljebb pár wattos teljesítmény-LED-ek), ezért nem is az adóteljesítményt határozzuk meg, hanem azt, hogy egy konkrét fényforrásból hány darab kell a szükséges fotoáram eléréséhez. Ebben lesz segítségünkre a saját fejlesztésű fotometriai szimulátor programom.

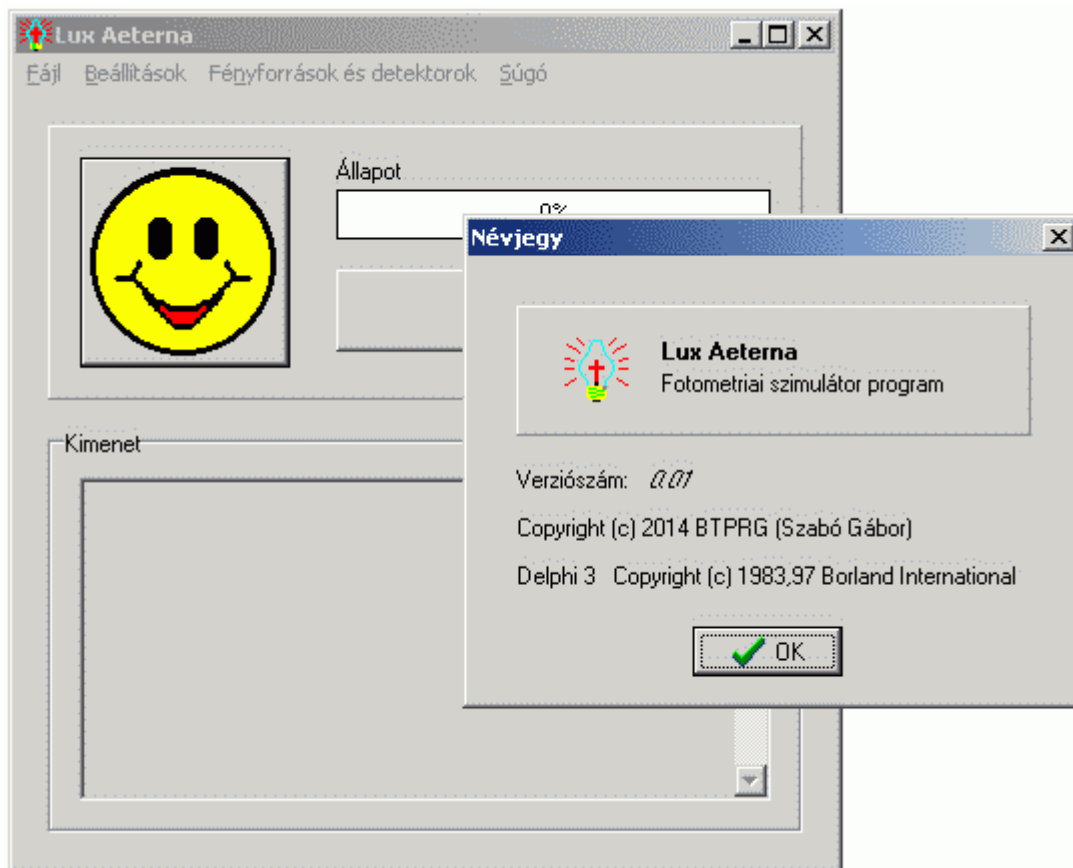
3.3.1 A Lux Aeterna fotometriai szimulátor program

A Lux Aeterna programmal egy tetszőleges méretű terep „plafon” részén tetszőleges elrendezésben helyezhetünk el fényforrásokat, és az egész terepen térben szabadon helyezhetünk el detektorokat. A fényforrásoknál a program fényáram, spektrális összetétel és iránykarakterisztika paramétereket kezel, a detektoroknál a paraméterek érzékeny felület, iránykarakterisztika és hullámhosszfüggő érzékenység. A karakterisztika jellegű paraméterek, hogy az integrálás egyszerű összegzéssel legyen helyettesíthető, tetszőleges felbontással megadott konstans tartományokként kerülhetnek bevitelre. Az egyszerűség kedvéért csak forgásszimmetrikus féltér iránykarakterisztikákat lehet megadni egy síkmetszet alapján.

A programban háromféle számítási mód lehetséges:

- A legfontosabb mód a detektorok¹⁶ fotoáramának számítása. A program itt minden detektorra kiszámolja, hogy a terepen lévő fényforrások összesen mekkora fotoáramot generálnak a fotodiódában. Ehhez figyelembe veszi az iránybeállításokon kívül a fényforrások spektrális összetételét és a detektorok hullámhosszfüggő érzékenységét.
- Lehetőség van a beállított terep „padlójának” megvilágítását kiszámolni. Ebben az esetben a padlót beállítható felbontással cellákra bontjuk, és ezek megvilágítását számolja a program. Ennek a számításnak a kimenete a gyakori nagy adatmennyiség miatt csak fájlba írható ki.
- Ha nincs szükség a terep pontos megvilágítására, akkor számoltathatjuk csak annak minimumát is. Ez egyszerűbb és gyorsabb, mintha a kimeneti fájlból kellene kikeresni a minimális értéket.

¹⁶ A program azt feltételezi, hogy a detektorok fotodiódák, de a számítási modell ugyanúgy jó akármilyen eszközre, ami (megközelítőleg) a bemenő fényteljesítménnyel arányos kimeneti jelet ad.



16. ábra: a Lux Aeterna fotometriai szimulátor program

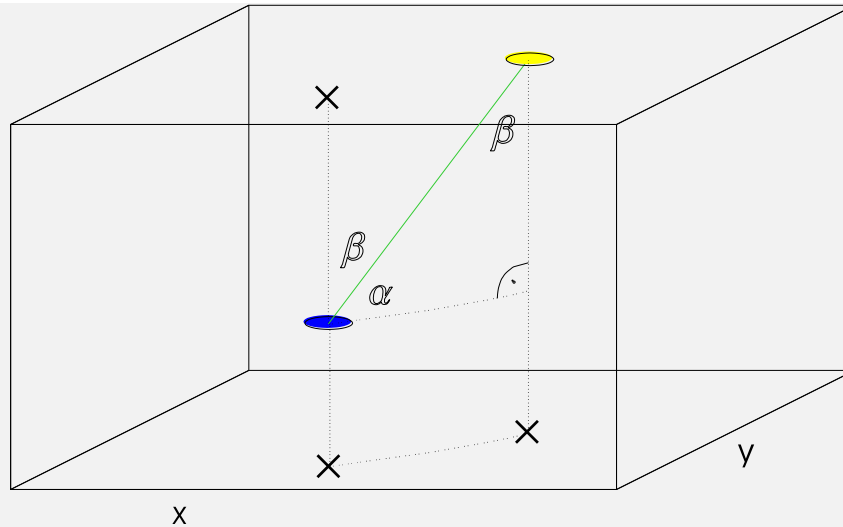
A program a bevitt objektumokat (fényforrás, detektor) automatikusan fájlba menti, és minden indításkor betölti azokat. Ezen kívül még a szimulációs elrendezések (terep méretei, objektumok elhelyezkedése) is menthetők, betölthetők.

A program által használt matematikai modelleket és a számítás menetének leírását a program dokumentációjából másoltam ki.

A program kétféle módon képes dolgozni: megvilágítottságot (illetve annak minimumát) lehet számolni a teljes terepre, vagy a detektor(ok)on keletkező fotoáramot lehet meghatározni. A számítás bizonyos pontig megegyezik, utána különböző.

Közös számítási lépések

Beesési szögek és távolság



A terep magasságának és a vizsgált rész „padlótól” mért függőleges helyzetének (megvilágítottság mérése esetén ez mindig nulla) különbsége megadja a derékszögű háromszög függőleges befogóját. A fényforrás és a vizsgált rész x és y koordinátáinak páronkénti különbségéből Pitagorasz-tétellel kiszámítható a derékszögű háromszög vízszintes befogója is. Így újabb Pitagorasz-tétellel meghatározható a fényforrás és a vizsgált rész távolsága. Váltószögek révén a (függőlegeshez viszonyított) irányszögek a fényforrás és a vizsgált rész esetén megegyeznek (β), a derékszögű háromszögből szögfüggvényekkel számíthatók.

A fényforrás irányfüggő fényerőssége

A fényforrás látható fénytéljesítményét jellemző paramétere a fényáram. A fényáram az iránykarakterisztika alapján oszlik el a féltér minden irányában. A számolásokhoz a térbeli iránykarakterisztika térszögtartományaira eső részeket kell meghatározni (\rightarrow fényerősség), amit a következő egyenletrendszer megoldásával tehetünk meg:

$$\left. \begin{aligned} I_1 \cdot \Omega_1 + I_2 \cdot \Omega_2 + I_3 \cdot \Omega_3 + \dots &= \Phi \\ I_1 : I_2 : I_3 : \dots &= x_1 : x_2 : x_3 : \dots \end{aligned} \right\}$$

ahol I_i az i -edik szögtartomány fényerőssége, Ω_i az i -edik szögtartomány térszöge, Φ a teljes fényáram, x_i az i -edik szögtartományhoz tartozó relatív intenzitás a síkbeli iránykarakterisztikában. Csak a fényerősség értékek az ismeretlenek, minden más érték ismert vagy számolható. Egy síkbeli szögtartományhoz tartozó (a 0° körüli részt leszámítva) gyűrű alakú térszög kiszámításának módja:

$$\Omega_i = 2\pi \cdot (1 - \cos \alpha_i) - 2\pi \cdot (1 - \cos \alpha_{i-1})$$

ahol α_i és α_{i-1} a szögtartomány felső és alsó határszögei. I_i ($i > 1$) kiszámítható I_1 -ből és a megfelelő relatív intenzitások arányából. Így a következő egyenlet marad:

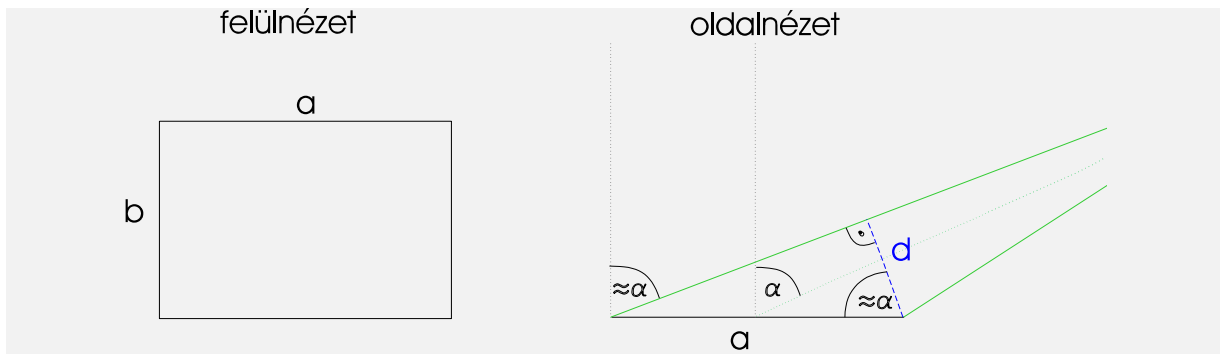
$$x_1 \cdot I_1 \cdot \Omega_1 + \frac{x_2}{x_1} \cdot I_1 \cdot \Omega_2 + \frac{x_3}{x_1} \cdot I_1 \cdot \Omega_3 + \dots = \Phi$$

Ebből kifejezhető I_1 , amiből kiszámolható az összes fényerősség érték. I_1 helyett – amennyiben az nulla – fixálható akármelyik nem nulla fényerősség érték.

Megvilágítás számítása

Az egy cellára eső fény térszöge

A „padló” egy cellájának hatásos területéből és annak fényforrástól lévő távolságából meghatározható egy térszög érték. A hatásos területet némi közelítéssel (a fényforrás nagyon távol van) viszonylag egyszerűen számolható az ábrán látható megfontolás alapján.



Így $A_h = b \cdot d = b \cdot a \cdot \cos \alpha$, ahol a és b a cella dimenziói, α pedig a cellát a fényforrással összekötő egyenes függőlegessel bezárt szöge.

A térszög számításánál szintén a fényforrás nagy távolságát felhasználó közelítéssel élek, és a gömbfelszín helyett egyszerűen a hatásos területet használom a képletben.

$$\Omega = \frac{A_h}{r^2}, \text{ ahol } r \text{ a cella távolsága a fényforrástól.}$$

A cella fényárama és megvilágítása

A fényforrás megfelelő irányú fényerőssége és a cellára eső fény térszöge alapján kiszámolható a cellára jutó fényáram. A fényáramot a cella területével elosztva kijön a cella megvilágítása:

$$\Phi = I \cdot \Omega, \quad E = \frac{\Phi}{A}$$

ahol Φ a fényáram, I a fényerősség, Ω a térszög, E a megvilágítás, A a cella (valódi) területe.

A detektor fotoáramának számítása

A detektorra eső fényáram

A detektorra eső fény térszögének számolásakor nem kell hatásos érzékeny felülettel számolni, mivel a detektor iránykarakterisztikája már magában foglalja ezt. Így a térszög egyszerűen az érzékeny felületből és a távolságból számítható:

$$\Omega = \frac{A_e}{r^2}.$$

A detektorra eső fényáram a fényforrás megfelelő szögű fényerősségének és ennek a térszögnek a szorzata lesz a detektor relatív iránykarakterisztikájának megfelelő szögű részével súlyozva.

A fotoáram kiszámítása

A detektor (fotodióda) fotoárama nem határozható meg kizárólag a detektorra jutó fényáramból. A fényforrás fényárama az emberi szem érzékenységből adódó láthatósági függvény alapján kerül kiszámításra a spektrális összetétel alapján, a detektor érzékenysége viszont általában egész máshogy hullámhosszfüggő. Viszont a fényforrás relatív spektrális összetétele, fényárama és a detektor érzékenysége hullámhosszfüggése alapján meg lehet határozni a generált fotoáramot. A fényáram kiszámításának módja¹⁷ lesz segítségünkre:

$$F = 6830021 \text{ lm/W} \cdot \int_0^{\infty} \bar{y}(\lambda) J(\lambda) d\lambda, \text{ ahol}$$

F a fényáram (lumen),

$J(\lambda)$ a sugárzás spektrális teljesítményeloszlása (W/m),

$\bar{y}(\lambda)$ (ami $V(\lambda)$ -ként is ismert) a standard láthatósági függvény (dimenzió nélküli),

λ a hullámhossz (m).

¹⁷ http://en.wikipedia.org/wiki/Luminosity_function#Details

A hullámhossz-tartományonként megadott láthatósági függvény ($V(\lambda)$) és relatív spektrális teljesítménysűrűség függvény (fényforrás spektrális összetétele) miatt integrálás helyett szorzással írhatunk fel egy egyenletet:

$$683 \cdot (c_1 x \cdot V(\Delta\lambda_1) \cdot \Delta\lambda_1 + c_2 x \cdot V(\Delta\lambda_2) \cdot \Delta\lambda_2 + \dots) = \Phi$$

ahol c_i az i -edik hullámhossztartományhoz tartozó relatív teljesítményérték, $V(\Delta\lambda_i)$ a láthatósági függvény értéke az i -edik hullámhossztartományban, $\Delta\lambda_i$ az i -edik hullámhossztartomány szélessége, Φ a detektorra jutó fényáram. Az egyenletben csak x arányossági tényező az ismeretlen, mely kifejezhető és kiszámolható. Ezután $c_i \cdot x$ lesz a (már nem relatív) teljesítménysűrűség az i -edik hullámhossztartományban W/nm-ben. Végül a fotoáramot a hullámhosszfüggő érzékenységgel való beszorzással kaphatjuk meg:

$$c_1 x \cdot \Delta\lambda_1 \cdot S(\Delta\lambda_1) + c_2 x \cdot \Delta\lambda_2 \cdot S(\Delta\lambda_2) + \dots = I_{\text{foto}}$$

ahol $S(\Delta\lambda_i)$ a detektor érzékenysége az i -edik hullámhossztartományban, I_{foto} a fotoáram.

3.3.2 A számítások

Legalább 20 dB jel-zaj viszony nem árt ahhoz, hogy a kommunikáció egyszerűbb vevővel is működőképes legyen, főleg hogy a vevő egyelőre ismeretlen áramkörei is adnak hozzá zajt a rendszerhez (ez a hozzáadott zaj egyszerűen nem meghatározható). Ez az érték azt jelenti, hogy a jeláram a zajáram 10-szerese kell legyen. A Lux Aeterna által számított fotoáram a fényforrások által generált teljes (modulálatlan) fényre vonatkozik, a modulált fény által generált változó fotoáram az intenzitásmoduláció modulációs mélységétől függ. A jeláram csúcserőtelje 100%-os modulációs mélység esetén egyezik meg a modulálatlan fotoárammal. Hogy ne támasszunk túl nagy követelményeket a vevővel szemben, az intenzitásmoduláció modulációs mélységét vegyük 30%-nak¹⁸. És mivel ezek csúcserőteljek, szinuszos jelet feltételezve még egy $\sqrt{2}$ -es csúcs tényezővel való leosztás is szükséges a számított zajjal összehasonlítható effektív érték számításához. Tehát ha a Lux Aeterna által adott érték x ,

akkor a számításokhoz az $y = \frac{0,3x}{\sqrt{2}}$ értéket kell felhasználnunk, ennek kell legalább a 2,5 nA

tízszerezésének lennie.

Fényforrásként első körben a laborunkban nagy mennyiségben elérhető OF-HPW1-1SL típusú 1W-os (80 lm) fehér teljesítmény-LED-et szeretném használni. Emlékeztetőül: a fotodióda típusa pedig VTP1188SH. Kétféle helyzetben is kiszámítom a szükséges LED-ek számát: amikor a detektor pontosan szemben áll a fényforrással, és amikor a detektor kb. 26°-os szögben¹⁹ látja a fényforrást. Felételezzük, hogy a fényforrások a „plafonon”, a detektorok a „padlón” vannak elhelyezve.

A plafon magassága	LED-ek szemben [db]	LED-ek 26°-ban [db]
2 m	1	1
3 m	1	2
4 m	2	3
5 m	3	5
6 m	4	7
7 m	6	9
8 m	7	11
9 m	9	14
10 m	11	17

¹⁸ Ehhez egy LED moduláló árama esetén jóval nagyobb modulációs mélységre is szükség lehet a nemlinearitás miatt.

¹⁹ Ha a fényforrás a plafonon, a detektor a padlón van, és a függőleges irányú távolságnak (plafon magasságának) fele a vízszintes irányú távolság.

Ha az amúgy igen nagy szögben világító (dómsugárzó) LED-ek fényét valamilyen lámpatest segítségével jobban egy irányba tereljük, akkor kevesebb LED is elég lehet ugyanolyan hatótávolság elérésére. Mivel egy adott pontban mérhető teljesítménysűrűség arányos a forrás teljesítményével, ezért például háromszor ekkora teljesítményű LED-ekből átlagosan harmadannyi kell ugyanahhoz a hatótávolsághoz.

4 A szoftveres rész realizálása 1

A kódosztásos átvitel szimulációjára megfelelő volt a *Matlab* környezet, viszont a valós kommunikációhoz ennél valamivel több kell. A fix blokkokon értelmezett műveletek helyett olyan programkörnyezet szükséges, ami valós időben képes adatfolyam kiküldésére és feldolgozására. Emellett kitűzött célom volt az is, hogy a Windows-ra szánt programok ne függjenek semmilyen futtató környezettől vagy hasonlótól, ezáltal a számításokat is elemi szinten kellett megírni. Ez a cél a *Matlab*-ot azonnal kizárta a végső program lehetséges alapjai közül. A választás végül a *Delphi*-re esett, mivel a Pascal nyelvben elég otthonosan mozogok, és a Delphi ablaképítő környezete az esetleges felhasználói felületek kialakítását is megkönnyíti.

A számítógépes CDM kommunikáció lelke végül is egy függvénykönyvtár lett, a *CDMU.DLL*, melynek segítségével végrehajtható a CDM kódolás, egyszerű keretszervezés, adatfolyam kiküldése a hangkártyán, adatfolyam bevitele a hangkártyáról, dekódolás és keretdetekció. Ezt a könyvtárat fogom ebben a részben – nem programkód szinten – ismertetni.

4.1 Kódok, emelt koszinusz szűrő

A könyvtár 511 hosszúságú 16 elemű kódkészletre van beállítva. A kódokat be lehet tölteni megfelelő formátumú külső fájlból is, de betölthető a beépített kódkészlet is, mely a DLL fájlban erőforrásként (resource) van eltárolva.

Mivel az emelt koszinusz szűrő impulzusválaszának generálását kicsit bonyolult lett volna alacsony szinten leprogramozni, ezért ez is külső megfelelő formátumú fájlból tölthető be, vagy a programkód formájában bevitt 0,6-os lekerekítési tényezőjű impulzusválasz választható ki.

4.2 Kódolás

A gyorsaság és az egyszerűség érdekében a kódoló rész statikus memóriefoglalású elemekkel van megoldva, ami miatt a kódolás csak kisebb (néhány bájtos) egységekben valósítható meg. Egy külön eljárással lehet a használni kívánt kódot kiválasztani, míg a kódolandó adatok a kódoló eljárás paramétereként adandók meg. A kimenet egy 44100 Hz-es mono 16-bites előjeles mintákat tartalmazó Windows memóriaobjektum lesz (*GlobalAlloc*), mely mindentől függetlenül jön létre és szabad felhasználású, a mintasor nem küldődik ki automatikusan a hangkimeneten. A chipsebesség 22050 Hz.

A kódolás automatikus keretszervezést is tartalmaz, mely igény szerint letiltható. Egy keret a következőképpen néz ki:

SYNC	start	adat	paritás	lezárás
6 bit	1 bit	36 bit	1 bit	1 bit
101010	0	xxxxx...xx	x	1

A *SYNC* minta előfordulása az *adat* részen bitbeszúrással történik, ezért a hasznos adatbitek száma néha kevesebb is lehet mint a megadott bitszám²⁰. A keret páros paritást használ. Az

²⁰ Emiatt lehetőség van kódolás előtt lekérdezni, hogy egy adatblokkból hány keret fog generálódni.

egyszerű keretszervezési lehetőség célja, hogy a vett adatok helyessége saját ellenőrző eljárások írása nélkül is könnyen megvalósítható legyen.

A kódolás során lehetőség van *ismételhető mintasor* generálására is. Ez azt jelenti, hogy az emelt koszinusz szűrő impulzusválaszával történő konvolúció során úgy járunk el, mintha egy végtelen periodikus jel egy periódusára vett konvolúciót számolnánk. Ebben az esetben a kimenet hossza meg is fog egyezni a bemeneti mintasor hosszával. Ha erre nincs igény, akkor a konvolúció felvezető része nullákkal lesz számolva (időtartományban a bemenet előtt nincs semmi), és lesz levezető rész is szintén nullákkal számolva, ahol a bemenetnek vége van.

4.3 Adatok kiküldése a hangkimeneten

Ha a kódoló által kiadott memóriaobjektumokban lévő mintasorokat ki akarjuk küldeni a hangkimeneten, akkor a memóriaobjektum azonosítóját a FIFO²¹ típusú hangkimenet várakozási sorba tölthetjük. A hangkimenetet külön eljárással indíthatjuk el, ami egy új feldolgozási szálat (thread) hoz létre, mely addig fut, amíg azt külön paranccsal le nem állítjuk, vagy elfogynak a kiküldendő adatok. Futás közben folyamatosan olvassa be a mintasorokat a várakozási sorból és adja ki a hangkártyán őket. A várakozási sorban lévő memóriaobjektumok feldolgozás után nem szűnnek meg mindenképpen, de beállítható automatikus felszabadítás is.

Van lehetőség egy alapértelmezett mintasor beállítására is. Ha ilyen beállításra kerül, akkor a várakozási sor kiürülése esetén automatikusan ez kerül kiküldésre egészen addig ismételve, míg új elem nem kerül a várakozási sorba. Ez a funkció használható periodikusan ismétlődő jelzés folyamatos sugárzására is.

A várakozási sor kezelésével körültekintően kell eljárni. Ha olyan memóriaobjektumot szeretnénk felszabadítani, ami még nem került 100%-os eséllyel kiküldésre, akkor célszerű az erre való eljárással kiüríteni a várakozási sort. A várakozási sorban lévő, de külsőleg felszabadított memóriablokkok futási hibához vezethetnek, mivel csak a várakozási sorba való elhelyezéskor van ellenőrizve a memóriaobjektumok valósága.

A hangkimenet eljárás a Windows-ban megadott alapértelmezett kimeneti eszközt használja a Windows Multimédia Rendszeren (MMSystem) keresztül. Ha valamilyen hiba történik a hangkimenet során (eszköz megnyitása sikertelen stb.), akkor a hangkimenet szál befejeződik, és a kilépési kódjából tudható meg a váratlan leállítás oka.

Amennyiben minél egyszerűbben szeretnénk eljárni egyetlen adatblokk kiküldése során, a korábbiakat automatikusan használó segédeljárással is kiküldhetjük azt. Ez esetben semmi más teendő nincs csak ennek az egy eljárásnak a meghívása.

A hangkimenet rész akár külső forrásból származó mintasor kiküldésére is használható. Ebben az esetben egy saját magunk által létrehozott memóriaobjektumba kell tölteni 44100 Hz-es 16-bites előjeles mono formátumú nyers mintasort, és ezt átadni a megfelelő eljárásnak.

4.4 Adatvétele a hangbemeneten

A hangbemenet a hangkimenethez hasonlóan szintén a Windows Multimédia Rendszert használja az alapértelmezett bemeneti eszközzel. A hangbemenet az erre való eljárással indítható el, és szintén külön feldolgozási szálon fut, melynek visszatérési értéke tájékoztat az esetleges hibaeseményről. 44100 Hz-es 16-bites előjeles mono mintasort olvas be a hangkártyáról folyamatosan egészen addig, míg a megfelelő eljárással le nem állítjuk a hangbemenetet. A beolvasást indító eljárásnak paraméterként kell megadni azt a segédeljárást,

²¹ „First In, First Out” adatkezelési stratégia

mely minden esetben meghívódik a minták feldolgozására, ha megtelik az egyik bemeneti puffer²².

Akár saját ilyen feldolgozó eljárást is írhatunk, de két ilyen beépített eljárás is létezik. Az egyik tesztelési célú, és a kapott mintákat egyszerűen a „*cdmu_be.raw*” nevű fájlba menti. A másik ilyen eljárás a mintasort újramintavételezi a chipsebességnek megfelelően 22050 Hz-re, ami tulajdonképpen azt jelenti, hogy minden két mintából átlagolással egyet csinál. Az újramintavételezett mintasor bekerül a szintén FIFO elvű bemeneti listába, ahol a további feldolgozásig várakozik társaival együtt. Az eljárás egy lekérhető azonosítójú Windows eseménnyel jelzi, amikor új elemet helyezett el a bemeneti listába.

A bemeneti listából a megfelelő eljárásokkal sajátmagunk is kiolvashatjuk a mintasorokat, melyek természetesen a hangbemenet leállítása után is elérhetőek maradnak. De ezt rábízhatjuk a beépített bitdekódoló eljárásra is, melyet a következőkben ismertetek. Ügyelni kell viszont arra, hogy a folyamatos hangbemenet melletti valós idejű feldolgozás tényleg valós idejű maradjon, vagyis a feldolgozási sebesség elég gyors legyen ahhoz, hogy minden új elemet a beérkezés pillanatában el lehessen kezdeni feldolgozni. Ellenkező esetben a bemeneti lista mérete folyamatosan növekedni fog, ami nagy memóriahasználathoz vezethet, de legalábbis a feldolgozás a hangbemenet leállítása után is jó darabig futhat a beérkezett adatmennyiség lekezeléséhez.

4.5 Dekódolás

Az előző részben említett bitdekódoló eljárás meghívás esetén automatikusan olvassa be a bemeneti lista elemeit, és dekódolja a különböző kódokhoz tartozó bitfolyamokat. Azt, hogy az első hány kóddal végezze a dekódolást, a kódkiválasztó eljárással lehet beállítani (mint a kódolásnál használt kódot). A bitdekódoló a paraméterként megadott megfelelő típusú eljárást hívja meg minden esetben, ha valamelyik kódhoz tartozó bitcsoport dekódolását befejezte; ennek az eljárásnak adja át a biteket és a számított minőség információt is.

A bitek dekódolása addig tart, amíg a bemeneti lista tartalmaz elemeket, vagy a feldolgozás el nem ért egy „a beolvasási ciklusban utolsónak” megjelölt elemhez (lásd: hangbemenet). Ha a bemeneti lista kiürült, akkor az eljárás egy bizonyos ideig várakozik újabb elemre, majd ennek híján visszatér.

A bitdekódoló eljárás *stdcall* hívási konvencióval definiált, így külön szálat is hozhatunk létre neki a WinAPI *CreateThread* eljárással. Ha megfelelő, akkor ettől függetlenül egyszerűen a főszálból is hívható, ha a programnak nincs egyéb feladata a dekódoláson kívül.

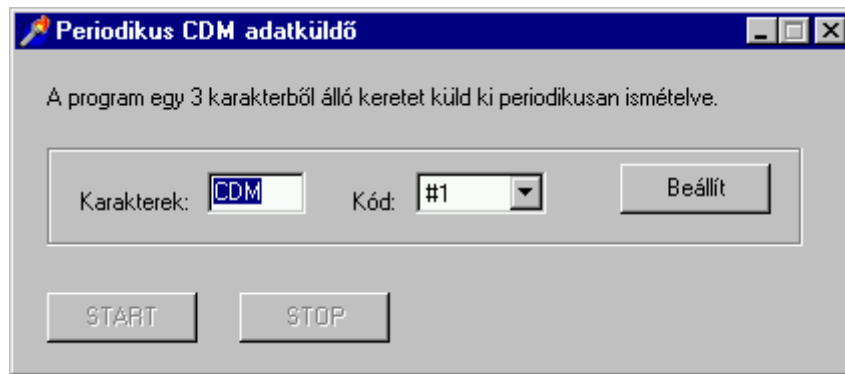
A könyvtárban külön függvények vannak keretek dekódolására is, de ezeket a visszkapott bitek alapján sajátmagunknak külön kell meghívogatni, ha ellenőrizni akarjuk a keret esetleges jelenlétét. Ezek az eljárások a keret keresésekor csak a SYNC biteket nézik, ezek megléte esetén mindenképpen dekódnak. Ha viszont hibát találnak, azt részletesen jelzik (pl. paritáshiba, stop bit hibája stb.).

5 A szoftveres rész realizálása 2

Az előzőkben ismertetett CDMU.DLL függvénykönyvtár elkészülte után maguknak a vezérlő programoknak a megírása mindössze a megfelelő eljárások helyes felhasználásán múlik. Két külön vezérlő programot csináltam.

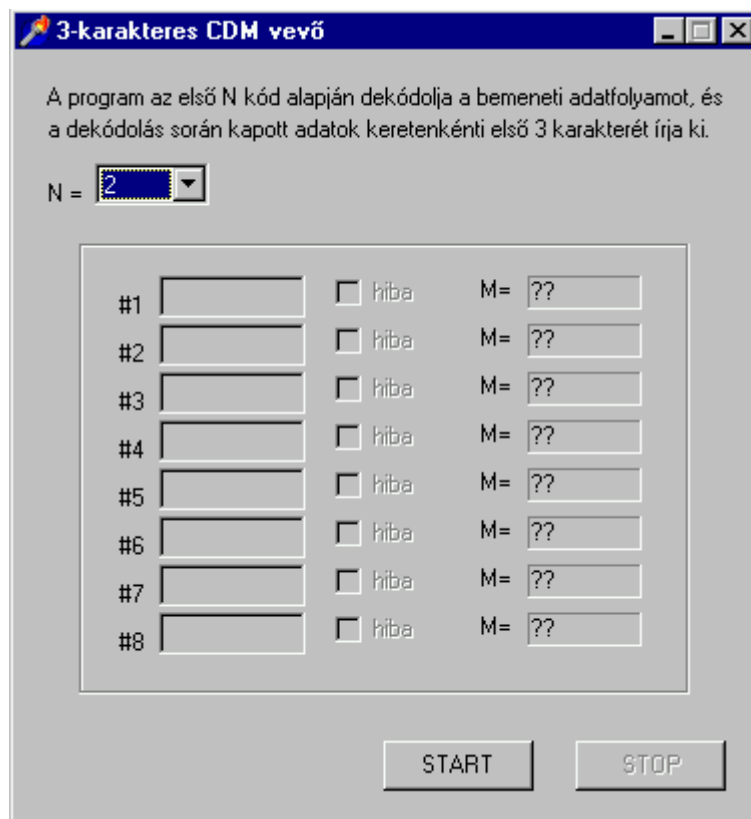
Az egyik program egy 3 karakterből álló tetszőleges kóddal kódolt keretet küld ki periodikusan a hangkimeneten. Ehhez a korábban ismertetett alapértelmezett kimeneti mintasort használja, melyet ismételhető típusúra generál a kiválasztott kóddal.

²² Ha leállítjuk a hangbemenetet, az utolsóként kiküldött puffertartalom mellé egy „utolsó a beolvasási ciklusban” jelzés is kerül.



17. ábra: az adó program ablaka

A másik a vevő, mely a hangbemenetet olvassa be, és valós időben dekódolja a kiválasztott első N csatorna kereteit. Sikeres dekódolás esetén a keretben található első három karaktert kiírja, és felvillantással jelzi a keret vételét. Jelzi továbbá a kerethibákat és detektált keret híján is folyamatosan mutatja a kódhoz tartozó csatorna minőségét.

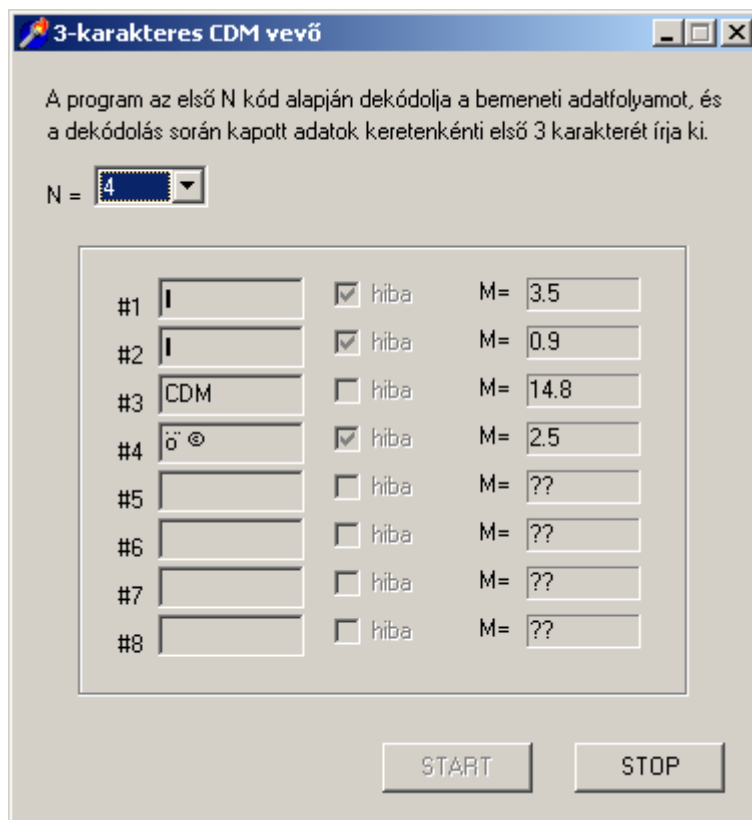


18. ábra: a vevő program ablaka

Ez a 3 karakteres adás-vétel elegendő a rendszer működésének tesztelésére.

5.1 A szoftveres rész működésének ellenőrzése

Az adó és a vevő program működését még a hardverek elkészülte előtt lehetséges tesztelni, ha az adott hangkártya kimenetet és bemenetet egyszerű vezetékkel kötjük össze. Ezt megvalósítva az adatátvitel elsőre működött. A vevő program ablaka működés közben a 19. ábrán látható.



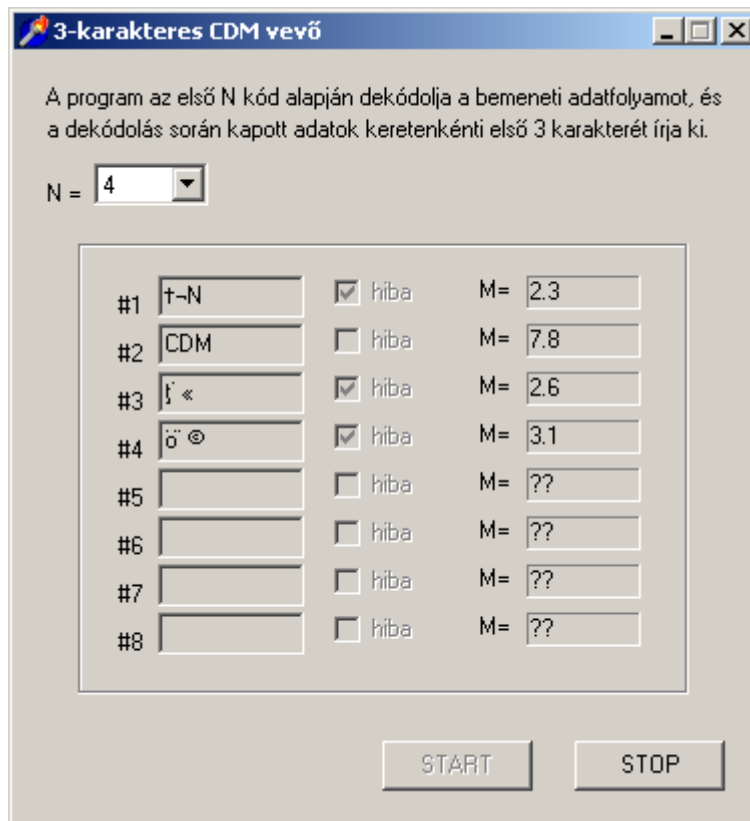
19. a vevő program ablaka, miközben a bemenetre a 3-as kóddal kódolt adás lett vezetve közvetlen vezetékkel

Az adóval a 3-as kóddal adtam egy másik gépen a „CDM” karaktereket, melynek vonalkimenetéről került a jel a laptop mikrofon bemenetére²³. Az adás minősége kb. 14 és 18 között mozgott²⁴, ezt nem túl nagy mértékben a beállított hangerő is befolyásolta. Érdekes viszont, hogy a hibák elég gyakoriak még így egy adás és közvetlen kapcsolat esetén is. Volt, hogy jó pár keret lement hiba nélkül, viszont néha előjön *egy-egy bithiba* (ez a kerethibából látszik, vagy abból, hogy az egyik karakter helyett más jelenik meg), sokkal ritkábban *az egész kerettartalom rossz*. Ezen jelenségek okát a későbbiekben még feltétlenül vizsgálni kell. Megjegyzendő, hogy elég sok a hamis keretdetekció azokon a csatornákon, melyek nincsenek is használatban. Ez nem túl nagy probléma, valószínűleg a mindössze 8-bites keretszinkron miatt van. Sokkal érdekesebb, hogy gyakran a többi (nem használt) csatornák némelyikében is megjelenik a használt csatornán küldött karakterekhez hasonló, sőt előfordult már hibátlan keret is a másik csatorna tartalmával (amellett, hogy a minőség paraméter ugyanúgy 5 alatt marad). Ez az áthallás jellegű dolog valószínűleg a kódok nem tökéletes keresztkorrelációs tulajdonságai miatt van, vizsgálatra érdemes hogy más kódkészlettel mennyire tapasztalható a jelenség.

Kíváncsisággéppen kipróbáltam azt is, hogy mi történik, ha a messze nem Hi-Fi minőségű számítógép-hangszórón adom ki a CDM jeleket, és mikrofonnal veszem azokat. Meglepő eredményt kaptam (vevő program: 20. ábra).

²³ Akár egy gépen is futhat az adó és a vevő program, ha a hangkártya illesztőprogramja illetve maga az eszköz támogatja a duplex üzemmódot. Amikor a lappal csináltam az adást is, akkor is ugyanilyen eredményt kaptam.

²⁴ A minőség paraméter számítása megegyezik a Matlab-szimulációnál megadottal. Értéke a (vélt) korrelációs csúcsok abszolút értékének és a jel abszolút középértékének hányadosának átlaga.



20. ábra: a vevő program ablaka, miközben a 2-es kóddal kódolt és hangszórón kiadott adást mikrofonnal vette a program

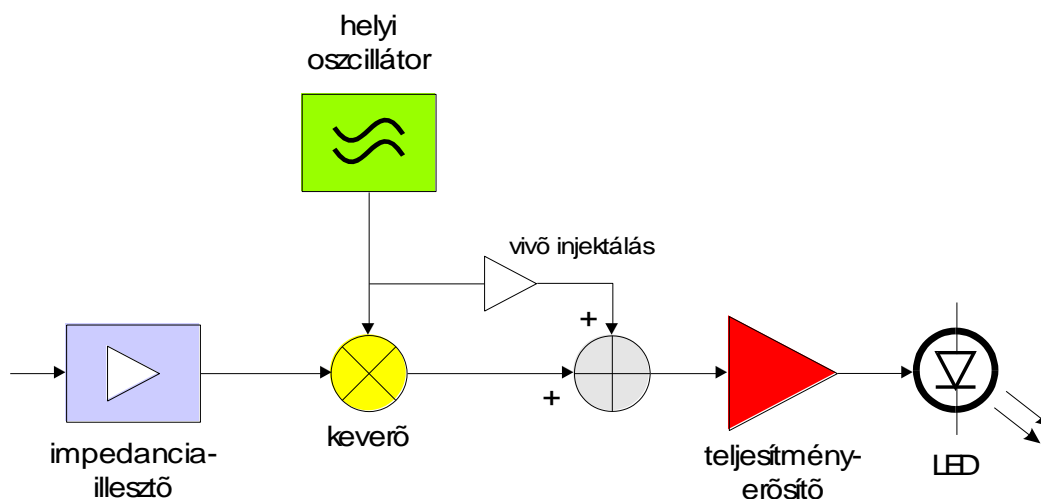
A mikrofont kb. 20 cm-re tartva a hangszórótól normál hangerő mellett a használt 2-es kódhoz tartozó csatorna minősége 7..8 körüli lett, és érvényes kereteket lehetett venni. Ugyan a hibák száma jóval gyakoribb volt, mint a vezetékes esetben, a számos hibátlanul vett keret nagyon jó eredmény. Ezek szerint e CDM rendszer zavartűrése nagyon jó, hiszen a bizonyosan borzasztó átviteli karakterisztikájú rossz minőségű hangszórón és a zajos akármilyen mikrofonon keresztül is valamilyen szinten működőképes maradt az átvitel. Bár még látszik hogy vannak megvizsgálni való dolgok, a számítógépes rendszer a tapasztalatok alapján készen áll a VLC-s hardverekkel történő éles bevetésre.

6 A VLC áramkörök

A szoftver ellenőrzése után el lehet kezdeni a látható fényes kommunikációt megvalósító hardver tervezéséhez és megvalósításához. A számítások és a használni kívánt optikai eszközök már megvannak, ezekhez kell hozzátervezni a megfelelő áramköröket.

6.1 Az adóáramkör

Mint már a látható fényvel történő kommunikációs kapcsolat energiamérlegének számításakor említettem, az adóban egy 1 W körüli teljesítményű LED-et szeretnék használni. Az átvitel során kb. 2 MHz-es vivőfrekvencián az egyszerűség kedvéért kétoldalsávú amplitúdómodulációt (AM-DSB) szeretnék megvalósítani. Ehhez a LED-et egy A-osztályú teljesítményerősítővel kell meghajtani, melynek bemenete az AM jel lesz.



21. ábra: a VLC adó blokkvázlata

Az általam tervezett áramkör blokkvázlata a 21. ábrán látható. A bemenet egy impedanciaillesztőn keresztül csatlakozik a keverőre, ahol szorozódik a helyi oszcillátor jelével. A jelen lévő vivős AM-DSB előállításához a keverő kimenetére a vivőt is injektálni kell (a vevőben burkolódetektort szeretnék alkalmazni). A nagyfrekvenciás jel végül a teljesítményerősítő bemenetére kerül, mely meghajtja a fénydiódát.

Az áramkör pontos kapcsolási rajza a függelékben található. A hangkártyáról jövő jel a *KI* csatlakozón érkezik az áramkörre. A program *mono* jelet ad ki, így lényegtelen, hogy a jobb vagy a bal csatornáról vesszük a jelet; az áramkörre a véletlenszerűen kiválasztott jobb csatorna jele kerül. Az *RV1* trimmerrel hardveresen is állítható a jelerősség, mellyel megakadályozható a keverő telítésbe vitele. A feszültségosztóként beállított trimmer kimenete a *Q1* tranzisztorra épülő emitterkövető bemenetére csatlakozik, mely nagy bemeneti impedanciát biztosít a hangkártya felé²⁵, illetve az *SA602A* kétszeresen kiegyenlített keverő áramkör (*UI*) körülbelül 1,5 k Ω -os bemenetét.

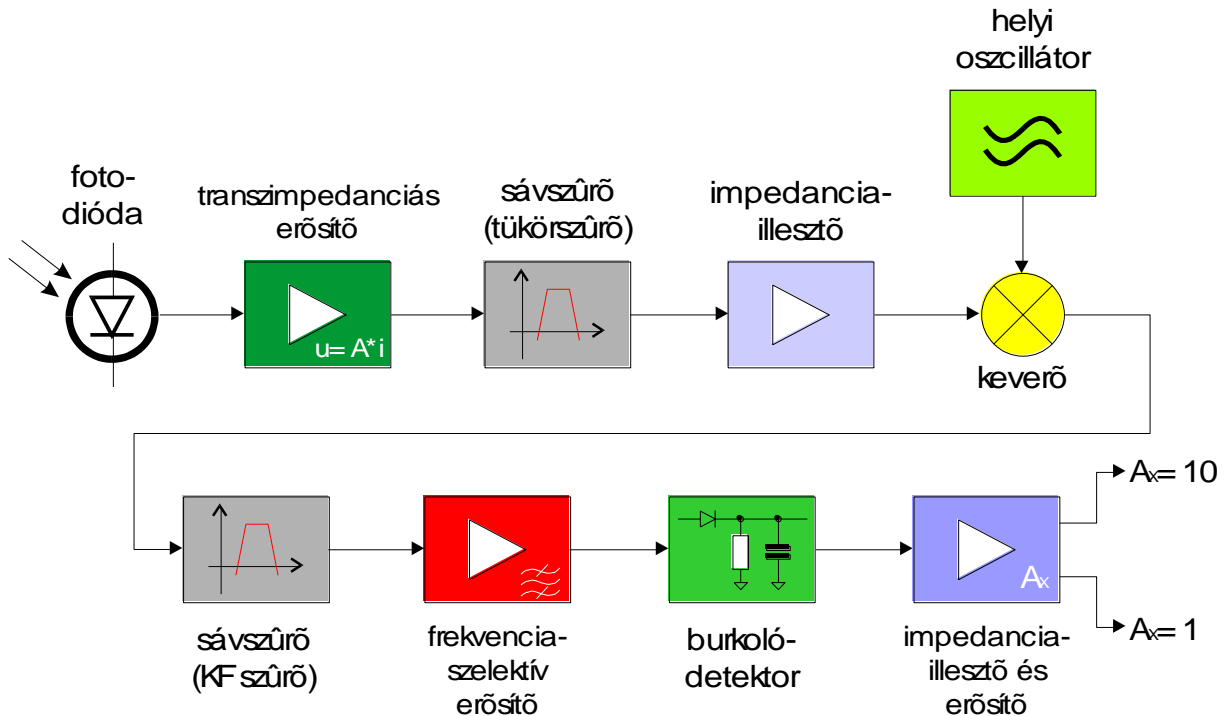
Az *UI* keverő áramkör egyben megvalósítja a helyi oszcillátort is, és az *XI* kvarckristály rezegtetésével kb. 1,8432 MHz-re keveri fel a CDM jeleket. A kétszeres kiegyenlítettség miatt a kimeneti jelben a vivő nem szerepelne (AM-DSB/SC), ezért a kiegyenlítettséget el kell rontanunk valamilyen mértékben, hogy vivőt injektálhassunk a kimenetre [3]. Erre szolgál az *R10* ellenállás és az *RV2* trimmer, utóbbival a vivőtartalmat lehet állítani. A nagyfrekvenciás jel a *C9* kondenzátoron át a *Q2* és *Q3* tranzisztorokból álló Darlington-kapcsolású végerősítőre kerül, mely a jellel a *D1* teljesítmény-LED áramát vezérli. Mivel A-osztályú végerősítőről van szó, a *Q3* tranzisztoron jelentős teljesítmény disszipálódik. A melegedés hatására a munkaponti áram is változik. Az *R4* és *R6* ellenállások visszacsatoló hatása már elég a hőmegfűtés megakadályozására, de a munkaponti áram még így is számottevően változhat a melegedés hatására. E változás tovább csökkentésére szolgál *Q4*, mely a munkaponti áram egy az *R6* által meghatározott bizonyos szint fölé emelkedése esetén nyitni kezd, csökkentve ezzel *Q3* bázisáramát. Az *R5-C10* szűrő szerepe, hogy *Q4* csak a munkaponti áramot stabilizálja, és ne próbálja meg kiszabályozni a jelet is.

Az áramkör 6 V-os tápfeszültségről megy, de az impedanciaillesztő és a keverő egy külön stabil 5 V-os tápról (*VCC*) megy a zavarok visszacsatolódási esélyének csökkentése érdekében, melyet az *U2* párhuzamos regulátor állít elő.

²⁵ A hangkártya vonalkimenete kb. 10 k Ω -os terhelő impedanciát vár.

6.2 A vevőáramkör

Az előbb ismertetett adóhoz készült fotodiódás vevőáramkör tulajdonképpen egy AM szupervevő²⁶. Hátrány a klasszikus középhullámú szupervevőkhöz képest, hogy a tervezés során nem használhattam transzformátorokat, és a középhullámú adásokhoz képest jóval nagyobb sáv szélesség miatt kerámiaszűrőt se a KF fokozatban.



22. ábra: a VLC vevő blokkvázlata

Az általam tervezett áramkör blokkvázlata a 22. ábrán látható. A fotodióda áramjelét egy transzimpedanciás erősítő konvertálja feszültséggé jó konverziós tényezővel. A keverő a tükrösűrőként funkcionáló sávszűrőn és egy impedanciaillesztőn keresztül csatlakozik a transzimpedanciás erősítőhöz. A helyi oszcillátor jelével szorozva a bemenőjelből előáll a középfrekvenciás jel, mely egy KF sávszűrőn keresztül jut a frekvenciaszelektív erősítő bemenetére, ahol a KF jel erősítésre kerül. A felerősített jelet a burkolódetektor demodulálja, és az utolsó erősítő fokozaton keresztül jut a hangkártýára, mely tetszés szerint funkcionálhat erősítőként vagy kizárólag impedanciaillesztőként.

Az áramkör pontos kapcsolási rajza a függelékben található. A *D1* fotodióda jelét az *U3* áramvisszacsatolású szélessávú műveleti erősítő mint transzimpedanciás erősítő veszi. A műveleti erősítő a referenciafeszültséget az *R4-R5* feszültségosztóról kapja, a fotodióda így kb. 5 V-os záróirányú előfeszítésben részesül. Az „egyenfény” által okozott egyenáramú rész a fotoáramban akár 100 μ A is lehet anélkül, hogy a műveleti erősítő kimenetét telítésbe vigye. A jel az *R1* ellenálláson keresztül jut az *L1-C13-C14* tagokból álló párhuzamos rezgőkörre, mely a hozzávetőlegesen a tükrösűrő szerepét tölti be az áramkörben. Bár egyszerű kéttárolós áramkör lévén a rezgőkör nem hajt végre meredek szűrést, hasznos a vivőfrekvenciától (1,8432 MHz) távolabb eső zavaró jelek kiszűrése szempontjából is. Hogy a rezgőkör ne legyen leterhelve (jósági tényező romolna), a szintén *SA602A* típusú *U4* keverő áramkör bemenete a *Q1* tranzisztorra épülő emitterkövetőn keresztül csatlakozik, mely elegendően nagy bemeneti impedanciájú.

²⁶ Szupervevő, mint supersonic heterodyne, mely frekvenciatranszponálást végez a gerjedés nélküli nagy erősítés elérése érdekében.

A keverő helyi oszcillátora *X1* alapján 2,4576 MHz-en rezeg, így az 1,8432 MHz-es vivőn ülő jel körülbelül 614,4 kHz-re kerül. Ez a jel aztán az *L3-C25-C26* rezgőkör érintésével jut a *Q3* tranzistorból álló KF erősítő bemenetére, mely a kollektorkörében lévő *L2-C17-C18* rezgőkör által szelektív erősítést végez. A két rezgőkör tölti be a KF szűrő szerepét, mely így nem olyan éles mint egy kerámia- vagy kristálysűrő esetén, de a célnak megfelel. A két rezgőkört a *C26* és *C18* trimmerkondenzátorokkal úgy kell behangolni, hogy rezonanciafrekvenciájuk közel azonos legyen.

A *C19* kondenzátoron keresztül csatlakozik az erősítőhöz a diódás burkolódetektor. Az amplitúdómodulált jelet a *D2* germánium tús dióda és a *C24-R12* elemek demodulálják. A demodulált jel a *Q2* által megvalósított erősítőn keresztül jut a hangkártya bemenetére²⁷. Ez az erősítő kétféle módban is használható *JP1* választó jumper állásától függően:

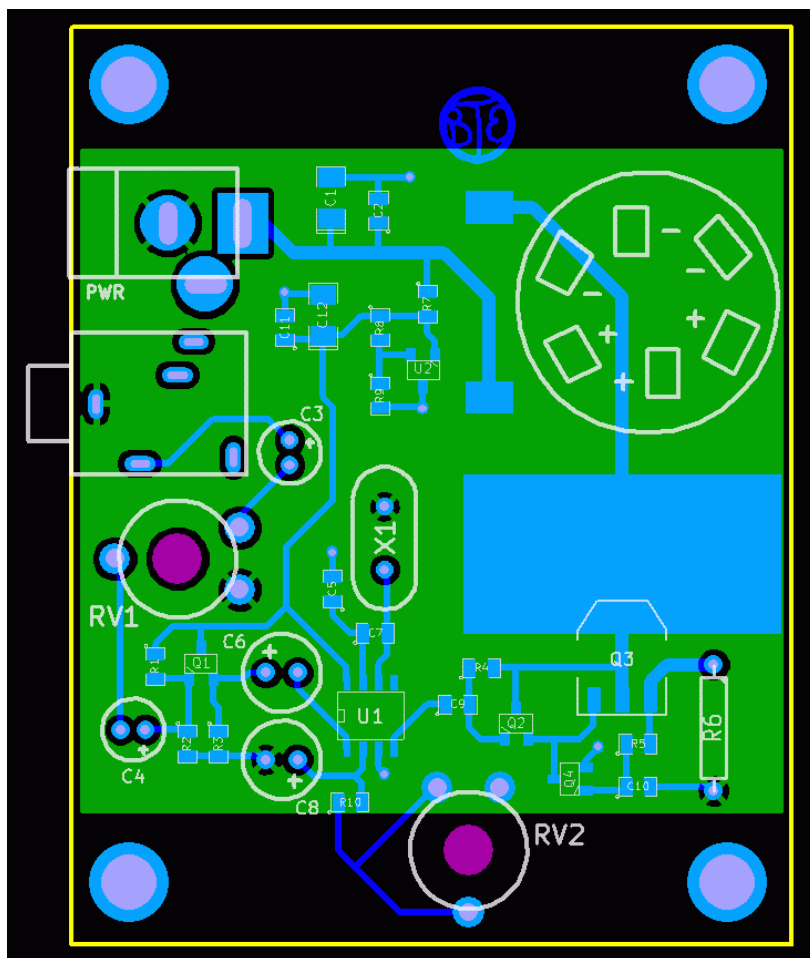
- 2-3 állásban az erősítő kb. 10-es feszültségerősítést csinál, ha a kimenő jel enélkül nem lenne elég nagy.
- 1-2 állásban az erősítő emitterkövetőként funkcionál, feszültségerősítés nincs.

Az áramkör 13..24 V-os tápfeszültségről működőképes, egy stabilizálatlan 12 V-os tápegység teljesen megfelelő. A bemeneti fokozat 10 V-os tápfeszültségről (*VCC*) megy, melyet az *U1* stabilizátor állít elő. A keverőtől kezdve a többi rész egy külön 6 V-os tápfeszültségről (*VDD*) működik, melyet *U2* állít elő *VCC*-ből. A többfokozatú tápellátás célja itt is a zavarvédelem fokozása.

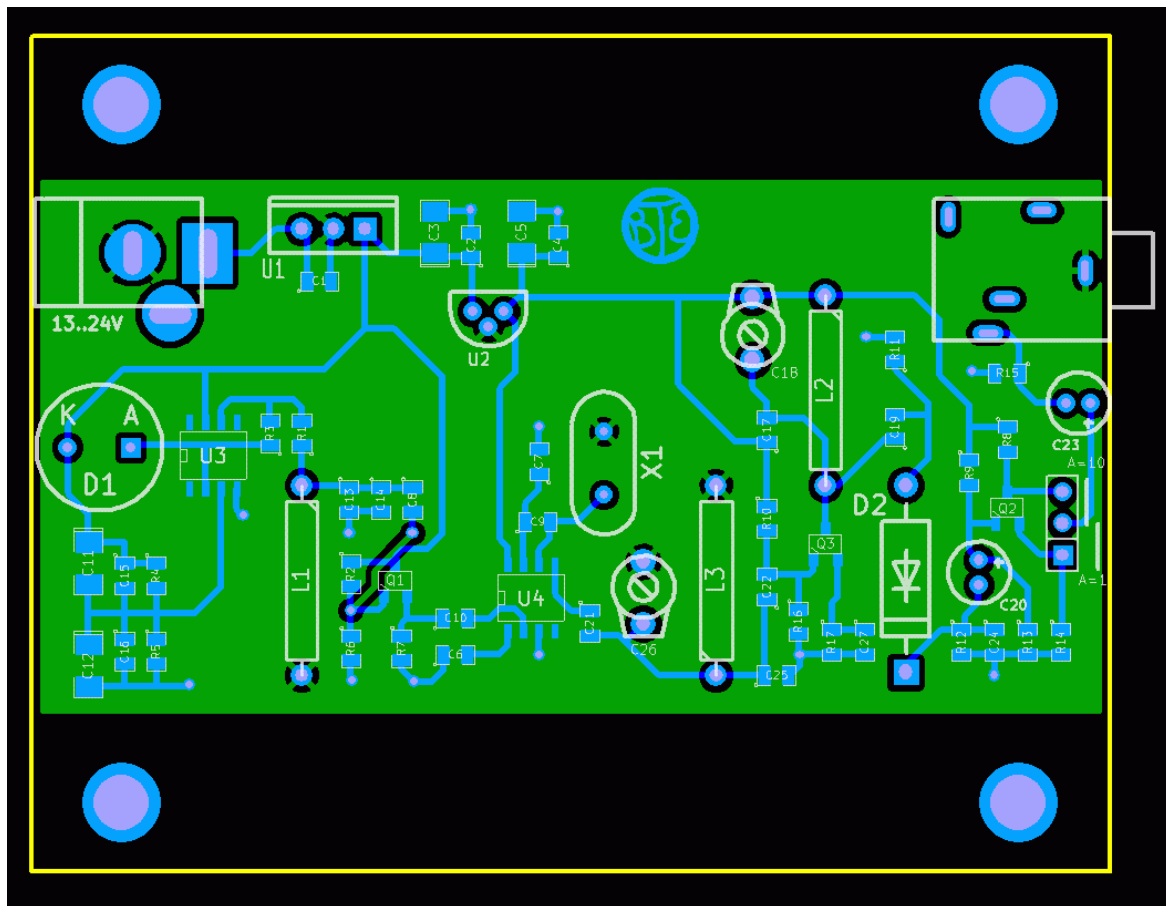
6.3 Paneltervek

A két áramkör nyomtatott huzalozású paneljának tervei kétoldalas elrendezésben, nagyrészt felületszerelt alkatrészek használatával készültek. A tervezésnél igyekeztem figyelembe venni elektromágneses kompatibilitási elveket, például hogy a kimeneti fokozat elemei lehetőség szerint ne kerüljenek közel a bemenethez. A kész tervek a 23. és a 24. ábrán láthatók.

²⁷ A csatlakozás úgy van kialakítva, hogy mikrofonbemenetre is ráköthető legyen az áramkör.



23. ábra: a VLC adó áramkör nyomtatott huzalozású lemezének terve



24. ábra: a VLC vevő áramkör nyomtatott huzalozású lemezének terve

Az áramkörökön a hátsó rézréteg vörös, az elülső rézréteg kék színnel szerepel (az egymással átfedésben lévő részeknél a szín is keveredik). Mindkét áramkör földkitöltést kapott a hátsó rézrétegen.

Összefoglalás

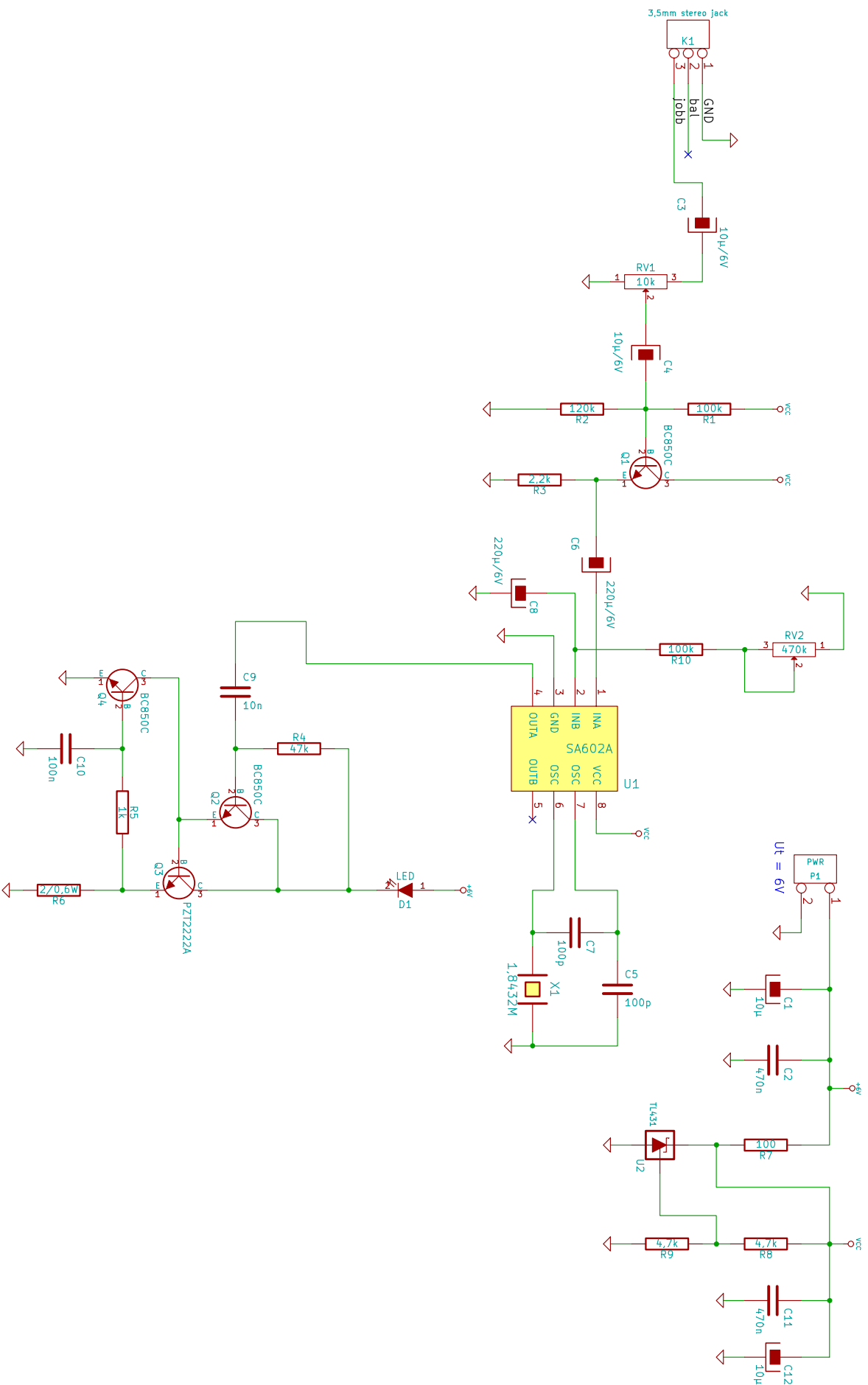
Sajnos a VLC áramkörök legyártására és ellenőrzésére a dolgozat befejezéséig nem jutott idő, ettől függetlenül az általam kidolgozott szoftveres részek működnek. Még több dolog van, ami további vizsgálatra szorul, de egyelőre úgy néz ki hogy a beltéri helyzetmeghatározás célú kis adatsebességű VLC CDM szisztéma működőképes.

Hivatkozások

- [1] F-L. Jenq, T-J. Liu and F-Y. Leu, «An AC LED Smart Lighting System with Visible Light Time-Division Multiplexing Free Space Optical Communication», 2011.
- [2] Csopaki Gyula, Németh Krisztián: «Mobiltelefon-hálózatok: UMTS», Hálózati technológiák és alkalmazások, BME-VIK egyetemi jegyzet.
- [3] Bill, W3DUQ & Tom, K1JJ «602 Balanced Modulator»,
<http://www.amwindow.org/tech/htm/602mod.htm>
- [4] Z. Mei, D. Zhao and J. Liu, «A Survey of Optical Networks and Its Visual Management and Control», 2011.
- [5] IEEE Standard 802.15.7, «Short-Range Wireless Optical Communication Using Visible Light», pp. 251-263, 2011.

Függelék

A következő két oldalon a VLC adó és vevő kapcsolási rajza található.



File: KC-VLCCDM_ado.sch

Sheet: /

Title: LED-es adó a VLC CDM-hez

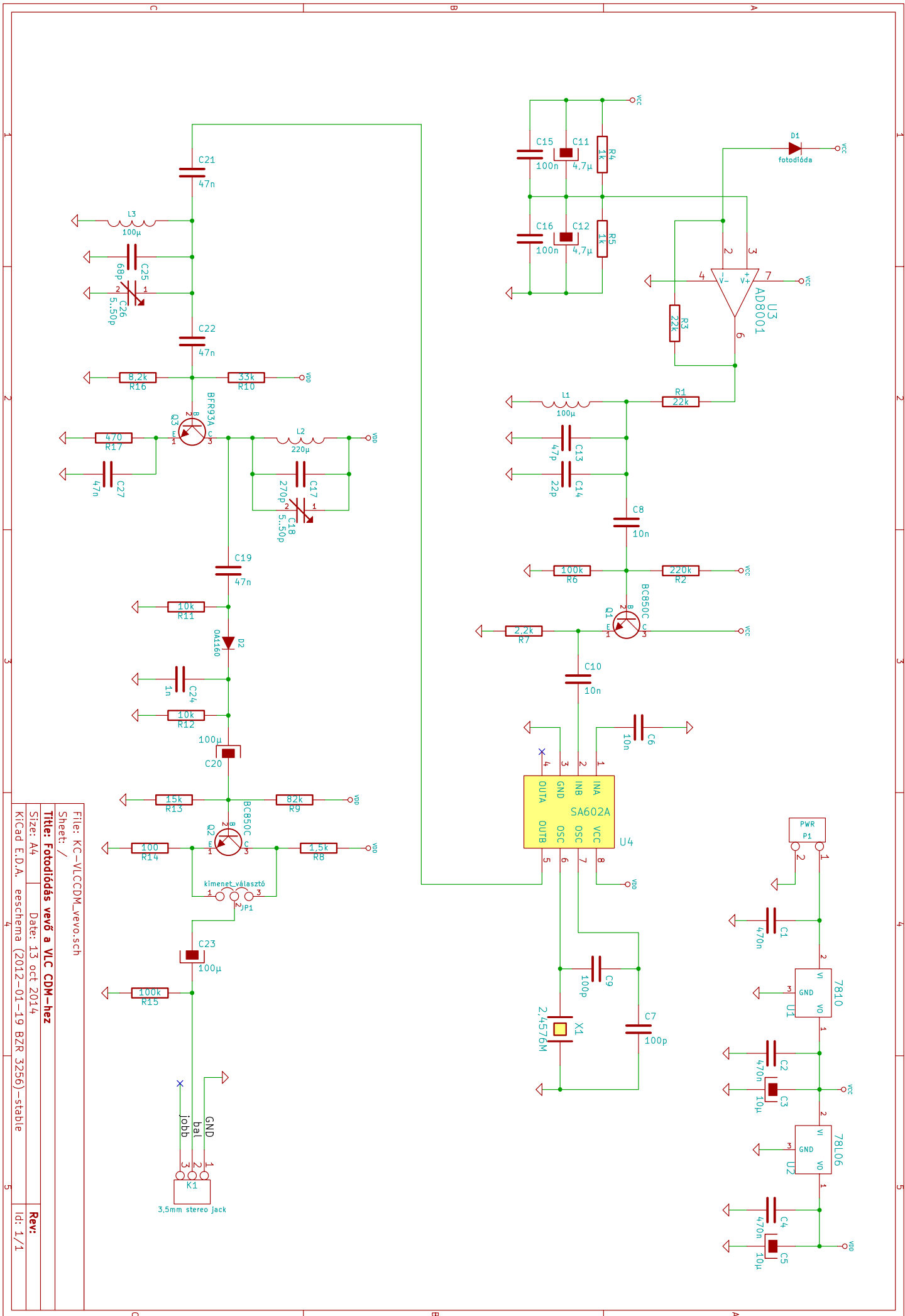
Size: A4

Date: 22 oct 2014

KiCad E.D.A. eeschema (2012-01-19 BZR 3256) - stable

Rev:

Id: 1/1



File: KC-VLCCDM_vevo.sch

Sheet: /

Title: Fotodiódás vevő a VLC CDM-hoz

Size: A4

Date: 13 oct 2014

KiCad E.D.A. eeschema (2012-01-19 BZR 3256) - stable

Rev:

Id: 1/1