

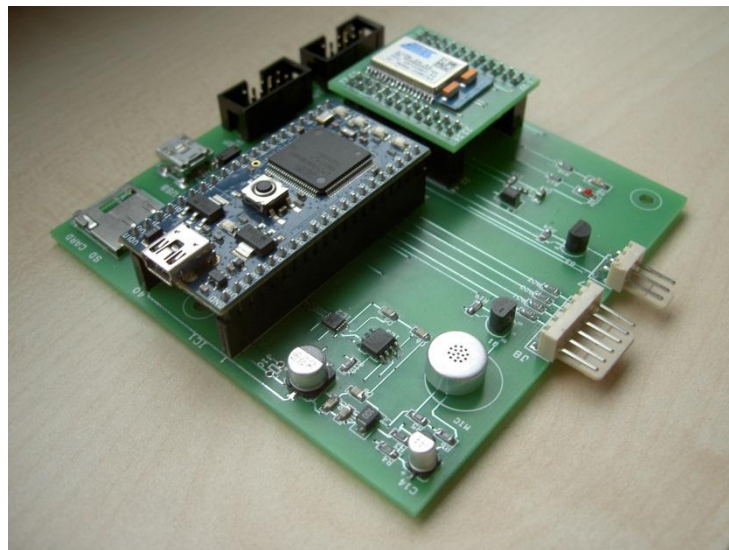


BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Tudományos Diákköri Konferencia dolgozat

Horkolás detektálás és elemzés

Beágyazott rendszerek és irányítástechnika szekció



Farkas Gergő

6. évf. Vill. szakos (MSc) hallgató

2011.

Konzulens: **Dr. Pataki Béla** docens

Méréstechnika és Információs Rendszerek Tanszék (MIT)

Tartalomjegyzék

Tartalomjegyzék	2
1. fejezet Bevezetés	4
1.1. A feladat bemutatása.....	4
1.2. Irodalomkutatás	5
1.2.1. Horkolás detektálás a mellkas mozgása és hang alapján	6
1.2.2. Horkolás detektálás a mellkasra rögzített szenzorokkal	8
1.2.3. Apnoe szindróma detektálása a horkolás spektruma alapján.....	8
1.2.4. A horkolás detektálása, annak matematikai modellje alapján	10
1.2.5. Az érzelmi állapot hatása a beszédhang alap frekvenciájára	13
2. fejezet A hardver bemutatása	15
2.1. A megvalósítás elvei.....	15
2.2. Az analóg áramkör.....	16
2.3. Az alkalmazott mikrovezérlő bemutatása.....	18
2.3.1. Az ARM Cortex-M3 architektúra	18
2.3.2. Az LPC1768 mikrovezérlő	20
2.3.3. A jelfeldolgozási műveletek a mikrovezérlőn	21
2.4. A megvalósított eszköz.....	24
2.5. Kommunikáció	26
2.5.1. Vezetékes kommunikáció	26
2.5.2. Zigbee alapú kommunikáció.....	26
3. fejezet Horkolás detektálására alkalmas algoritmusok.....	28
3.1. A hang tulajdonságai	28
3.2. Horkolás detektálása	30
3.2.1. Előfeldolgozás.....	31
3.2.1.1. Szűrés	32
3.2.1.2. Középkivágás:	36
3.2.2. Alapfrekvencia meghatározása	37
3.2.2.1. Autokorrelációs függvény (ACF).....	38
3.2.2.2. Átlagos magnitúdó különbség függvény (AMDF).....	46
3.2.2.3. Harmonikus Egyezés Módszere	47
3.2.2.4. Diszkrét kepsztrum analízis	52

3.2.2.5. Az alapharmonikus detektálási módszerek összefoglalása	53
4. fejezet.....	54
Az autokorrelációs függvény kiszámításán alapuló algoritmus a beágyazott környezetben....	54
4.1. A beágyazott szoftver feladata.....	54
4.2. A perifériák konfigurációja.....	55
4.3. A horkolás detektáló algoritmus implementációja	58
5. fejezet Eredmények	67
6. fejezet Összefoglalás	71
6.1. Áttekintés	71
6.2. Jövőbeni célkitűzések	72
Ábrajegyzék.....	73
Irodalomjegyzék	74
Melléklet.....	77
1. melléklet: A mérő panel kapcsolási rajza (első oldal)	77
2. melléklet: A mérő panel kapcsolási rajza (második oldal)	78
3. melléklet: A mérő panel NYÁK terve	78
4. melléklet: A Zigbee adapter NYÁK terve	79

1. fejezet

Bevezetés

1.1. A feladat bemutatása

Az orvostudomány fejlődése napjainkban elképzelhetetlen a különböző tudományok és szakterületek összehangolt, egymást segítő munkája nélkül. Ezen összekapcsolódás egyik megnyilvánulása a mérnöki tevékenység kibontakoztatása az orvosi technológia területén.

Ezen együttműködés jegyében a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karának Villamosmérnöki (MSc) Szakán írt szakdolgozatom témája egy horkolás detektálásra alkalmas eszköz fejlesztése.

Alapvető célunk az idősek, betegek állapotának otthonukban történő megfigyelése olcsó, egyszerű eszközökkel úgy, hogy őket ez a legkevésbé zavarja. A Méréstechnika és Információs Rendszerek Tanszéken ennek keretében kialakításra kerül egy úgynevezett intelligens szoba, amely egy átlagos ember mindennapjaihoz tartozó használati eszközökkel lesz berendezve (ágy, szék, fotel, televízió stb.), amelyek különféle érzékelőkkel vannak felszerelve. Ezek segítségével megfigyelhetjük, autonóm módon detektálhatjuk az ott tartózkodó ember különböző tevékenységeit, egészségügyi állapotát stb. Ebbe az intelligens szobába készül az általam fejlesztett eszköz, amely a rendszerbe integrálódva, alkalmas a szobában alvó egyén horkolásának detektálására.

A horkolás sajnos nagyon gyakori probléma, és kevesen gondolnák, pedig valószínűsíthető, hogy alvási zavarok állnak a gyermekkori hiperaktivitás, tanulási zavarok, viselkedési problémák, nappali álmoság és a figyelemzavarok hátterében is. Sokunkat érint a fáradékonyság, a fejfájások, táplálkozási zavarok, melynek egyik oka szintén alvászavar lehet. Továbbá a kutatók összefüggést mutattak ki az elhízás és az alvászavarok között is.

"A hangos, lélegzésszünetekkel járó horkolás kapcsolatba hozható a kardiovaszkuláris betegségekkel és megnövelheti az egészségügyi és kórházi kezelések számát." - derül ki a

SLEEP című lapban megjelent cikkből [1]. A tanulmányt a Semmelweis Ignác Orvostudományi Egyetem kutatói: dr. Novák Márta, dr. Mucsi István, dr. Dunai Andrea írták és dr. Kopp Mária professzor asszony által vezetett „Hungarostudy 2002” vizsgálatsorozat keretében, több mint 12 000 személyen elvégzett vizsgálaton alapul. A szakemberek elemzéséből kiderült: a hangosan horkolóknak 40 százalékkal nagyobb az esélyük a hipertenzió, 34 százalékkal a szívinfarktus és 67 százalékkal a stroke kialakulására, mint a nem horkolóknak. A csendes horkolás miatt a nők esetében megnövekedhet a magas vérnyomás kialakulásának kockázata. A megvizsgált nők 40, a férfiaknak 70 százaléka horkolt. A horkolást erősítheti az izomlazítók használata, a dohányzás, valamint az alkohol- és a drogfogyasztás. Fontos tehát, hogy felismerjük, detektáljuk a horkolás tényét, és szükség esetén orvoshoz forduljunk.

A TDK munka során egy olyan hardvert fejlesztettem ki, amely egy mikrofon segítségével felvételt készít az alvás során keletkező hangokról. Ezeket helyben, autonóm módon képes feldolgozni és felismerni a horkolás tényét. Fontos, hogy a felvett hangokat nem továbbítja más egységeknek, és nem menti el, elkerülve az ezzel járó esetleges személyiségi jogi problémákat. Egy másik fontos követelmény, hogy a megfigyelt egyént a lehető legkevésbé zavarjuk alvása közben. Az úgynevezett polyszomnográfias vizsgálatok során az egyénnek egy alváslaborban kell töltenie egy éjszakát, ahol nem éppen kényelmes körülmények között, mindenféle szenzorokkal és megfigyelőkkel a testén kell aludnia. Az én célom ezzel ellentétben éppen az, hogy a páciens a saját otthonában, testére erősített szenzorok nélkül tudjam megfigyelni. A következőben bemutatok néhány kapcsolódó fejlesztést.

1.2. Irodalomkutatás

A horkolás egy fontos jele lehet az alvás közben fellépő légzési rendellenességeknek, amilyen az obstruktív alvási apnoe szindróma (OSAS) is. A horkolás elemzése tehát, az alvászavarok diagnózisa és kezelése szempontjából alapvető fontosságú. Ennek megfelelően a szakirodalomban számos kutatás olvasható, amelyeknek a célja horkolás detektálása, illetve elemzése.

Számos eszközt fejlesztettek ki ebből a célból. Például *Satoshi Noda és társai* [2] egy olyan eszközt, amely a mellkas mozgása és a hang alapján képes detektálni a horkolás tényét.

P. Alexander Derchak és társai [3] mellkasra erősített szenzorokkal, *Robert B. Crossley* [4] nyak köré rögzíthető érzékelőkkel figyelik meg a páciens alvása közben. *Penzel és társai* [5,6] egy MESAM IV nevű digitális megfigyelő eszközt fejlesztettek az oxigén szaturáció, szívverés, horkolás, és test pozíció monitorozására. *Yeh-Liang Hsu és társai* [7] egy otthoni betegmonitorzásra alkalmas hordozható egységet fejlesztettek. *Rymut és társai* [8] a páciens nyakán elhelyezett piezoelektromos érzékelőkkel, míg *Schechter és társai* [9] gyorsulásmérővel figyelik meg a páciens torkának akkusztikus rezgéseit.

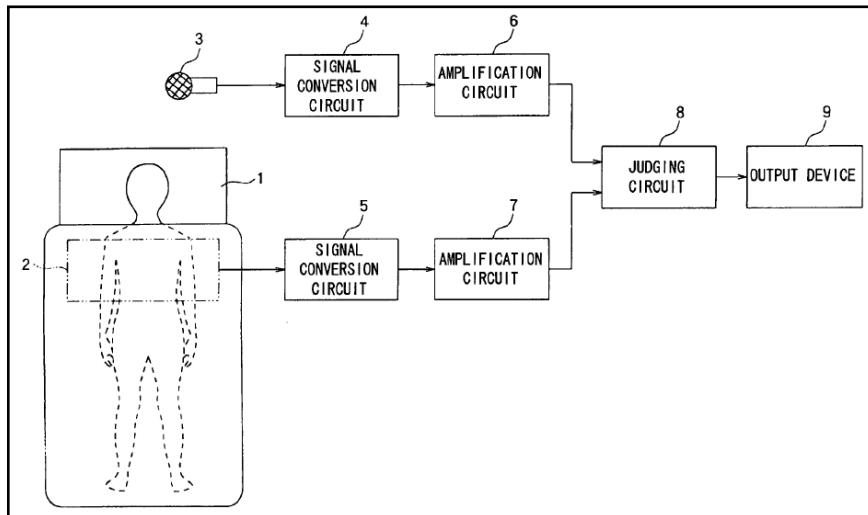
A sokféle hardver fejlesztés mellett az irodalomban különböző jelfeldolgozási módszerek, algoritmusok is találhatóak a horkoláshang elemzésére. Több kutatás [10,11,12] során is a horkoláshang spektrumát vizsgálták obstruktív alvási apnoe szindrómás betegek és egyszerű horkolók esetében. *U.R. Abeyratne és társai* [13] a horkolásról egy matematikai modellt alkottak, amely segítségével a mért regisztrátum alapján a beszéd és a horkolás megkülönböztethető.

A beszéd és a horkoláshang sok közös tulajdonságot mutat, ezért érdemes a beszédhang elemzési módszerekkel is megismerkednünk, amelyek irodalma szintén igen sokrétű. *Marius Vasile és társai* [14] az érzelmi állapotnak a beszédhang frekvenciájának ingadozásra gyakorolt hatását vizsgálták. *T. Kinnunen és társai* [15] tanulmányukban egy olyan rendszert mutatnak be, amely valós időben képes azonosítani a beszélő személyt annak hangja alapján.

A következőkben ezen kutatások közül a legfontosabbakat részletesebben is bemutatom.

1.2.1. Horkolás detektálás a mellkas mozgása és hang alapján

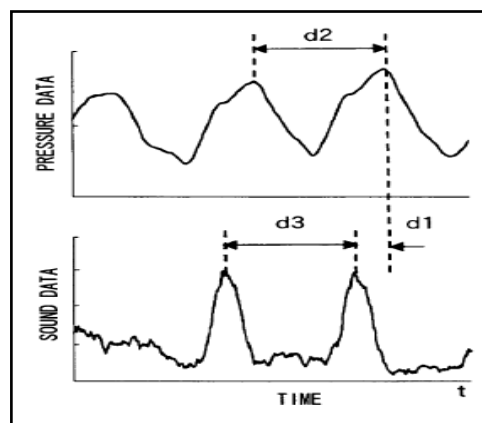
Satoshi Noda, Takeshi Minamiura és Hidetaka Sakai japán tudósok egy olyan horkolás detektálásra alkalmas eszközt fejlesztettek, amely egy nyomásérzékelő szenzorral méri az emberi test által a fekvőhelyre kifejtett nyomást be- illetve kilégzéskor, továbbá egy mikrofonnal detektálja a kiadott hangokat alvás közben [2]. A két érzékelő kimenő jelei alapján egy döntő áramkör ítéli meg, hogy a páciens horkol-e.



1. ábra - A horkolás detektáló rendszer blokkvázlata

A döntő áramkör érzékeli a csúcsokat a nyomásérzékelő által érzékelt jelben, és a mikrofon jelében. Ezek egyidejűsége alapján dönt a horkolás tényéről.

Tehát az eszköz nem vesz fel (tárol el) hangokat és csupán csak a horkolás tényét állapítja meg.



2. ábra - A nyomásérzékelő és a mikrofon jele

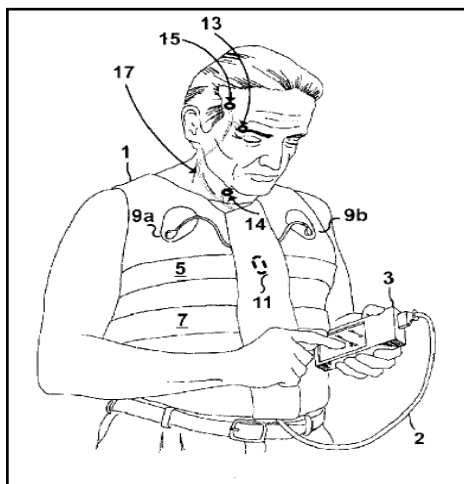
A nyomásérzékelő egy kapacitív elven működő érzékelő, a páciens háta illetve mellkasa alá helyezve. Két párhuzamos vékony vezető síklap között egy elasztikus anyag szolgál dielektrikumként. Az eszköz ezen érzékelőnek a nyomás hatására bekövetkező kapacitás változását méri.

A fejlesztés számos általam támasztott követelményt teljesít, hiszen a lehető legkevésbé zavarja a páciens alvás közben, és a felvett hangot nem továbbítja, hanem helyben feldolgozza a mért adatokat, és dönt a horkolás tényéről. A kapacitív nyomásérzékelő ötlete különösen tetszett nekem, hiszen így a vizsgált személy alvás közben való zavarása nélkül információt gyűjthetünk az éjszakai mozgásáról. Amint azt majd később látni fogjuk, a saját

eszközömet is alkalmassá tettem arra, hogy további fejlesztések során, akár egy kapacitív nyomásérzékelővel, akár nyúlásmérő bélyeges szenzorokkal bővíthető legyen.

1.2.2. Horkolás detektálás a mellkasra rögzített szenzorokkal

P. Alexander Derchak, Lance Myers és Gary Michael Lucia tanulmányukban egy olyan rendszert mutatnak be, amely a mellkasra rögzített szenzorokkal méri a mellkas mozgását ki illetve belégzéskor, illetve egy mikrofon segítségével rögzíti a hangot [3].



3. ábra - Horkolás detektálás a mellkasra rögzített szenzorokkal

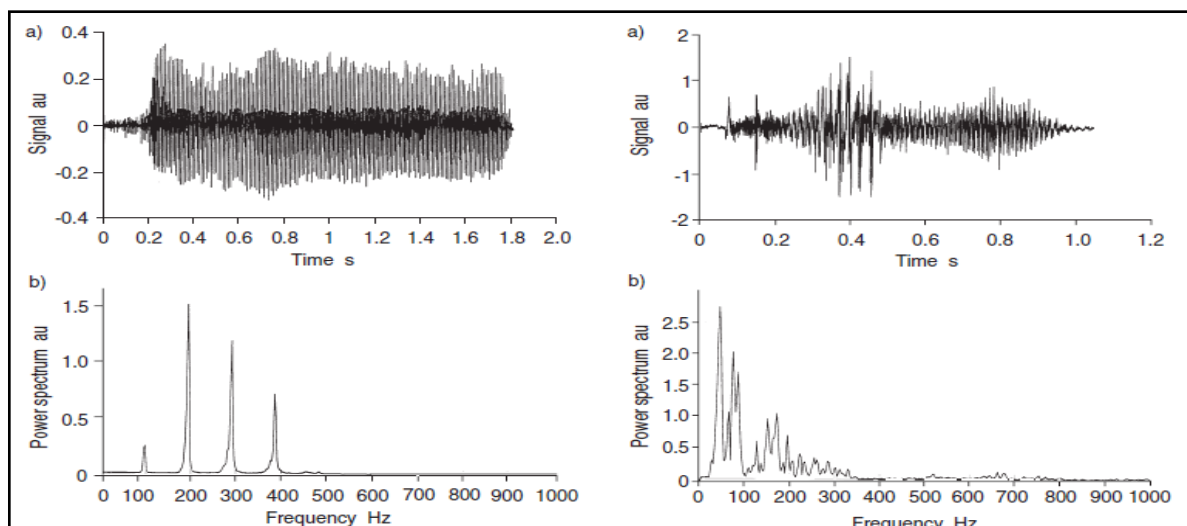
A mérési adatok feldolgozása az előző módszerhez hasonló módon történik: ha a mellkas mozgásában és a mikrofon által felvett hangban lévő csúcs időben egybeesik, akkor horkolást detektálnak.

Az emberi testre erősített szenzorokat a saját fejlesztésemben szeretném elkerülni, hiszen a vizsgálandó személytől nem várhatjuk el, hogy nyugodtan tudjon aludni, miközben testét teleraktuk különböző elektronikus eszközökkel.

1.2.3. Apnoe szindróma detektálása a horkolás spektruma alapján

J.A. Fiz, J. Abad, R. Jané, M. Riera, M.A. Mañanas, P. Caminal, D. Rodenstein, J. Morera 17 férfit vizsgáltak, amelyek közül 10-en alvási apnoe szindrómában (alvás közbeni légzéskimaradás) szenvedtek, 7-en pedig egyszerű horkolók voltak [10]. Egész éjszaka poliszomnografiát végeztek rajtuk és a horkoláshang spektrumát elemezték.

A horkolás elemzése két alapvetően különböző spektrum minta létezésére derített fényt. Az egyikben az alapharmonikus frekvencia jelenléte mellett megjelennek a felharmonikus frekvenciák. A másikban egy alacsony frekvenciás csúcs jelenik meg, ami körül egy keskeny frekvencia sávban szétszóródik a spektrum világosan azonosítható felhangok nélkül.



4. ábra - Két alapvetően különböző horkolás spektrum

Az eredmények biztatóak a kutatás szempontjából: A 7 egyszerűen csak horkoló férfi és ketten az alvási apnoe szindrómában szenvedők közül az első mintának megfelelő spektrummal rendelkeznek, míg a maradék 8 alvási apnoe szindrómás férfi horkoláshangjának spektruma a második mintának felelt meg. A frekvenciacsúcs az alvási apnoe szindrómásoknál jóval alacsonyabb frekvenciákon jelentkezett, mint az egyszerűen csak horkoló páciensek esetén.

Tehát az eredmények jelentős különbségeket mutattak az egészséges és az alvási apnoe szindrómás emberek horkoláshang spektrumában. Az én elsődleges célom ugyan a horkolás detektálása, de a jövőre nézve biztató az eredmény, hogy csak a horkolás spektruma önmagában már mennyi információt hordoz. Hozzá kell tennem azonban, hogy a tapasztalataim nem támasztják alá ezen kutatás eredményét, ugyanis az általam vizsgált horkoláshang spektrumok egyikében sem lehetett annyira egyértelműen felfedezni az alapharmonikust és a hozzá tartozó felharmonikusokat, mint azt az első mintának megfelelő spektrumban láthatjuk.

1.2.4. A horkolás detektálása, annak matematikai modellje alapján

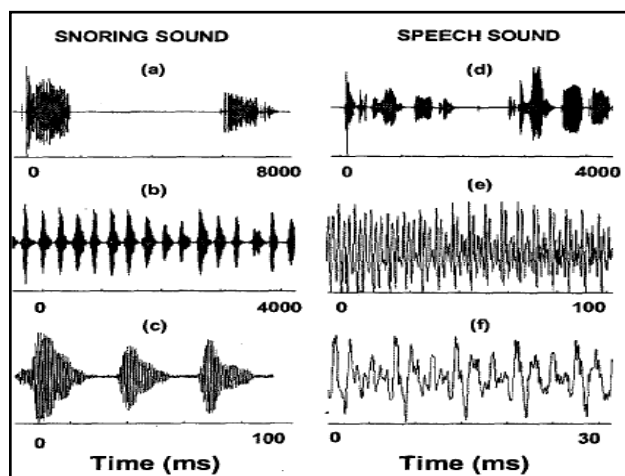
U.R. Abeyratne, C.K.K. Patabandi és K. Puvanendran a szingapúri *Nanyang Műszaki Egyetem* kutatói tanulmányukban [13] leírják a horkolás matematikai modelljét, és ennek segítségével bemutatják, hogy milyen módon lehet a horkolást és az alvási apnoe szindrómát detektálni a hangfelvétel elemzése alapján. Bemutatják a hasonlóságokat és a különbségeket a horkoláshoz és a beszédhez tartozó jelekben, amelyeket kihasználva lehetőség nyílik ezen kétfajta komponens külön választására. A horkoláshang és a beszédhang megkülönböztetése az általam fejlesztett egységnek is fontos feladata, ezért a tanulmányukról itt egy részletesebb leírást adok.

A horkoláshang diszkrét (mintavételezett) $y[n]$ jelének jó matematikai modellje a következő:

$$y[n] = s_s[n] + s_p[n] + b[n] = s[n] + b[n],$$

ahol $s_s[n]$ a horkoláshoz tartozó jel, $s_p[n]$ az álomban való esetleges beszédhez tartozó jel, míg $b[n]$ az elektronikus és akusztikus háttérzaj. Az $s[n]$ jel együttesen tartalmazza beszédhez és a horkoláshoz tartozó hangokat. Az elemzéshez az egyik legfontosabb feladat a zaj leválasztása a jelről. Ez azonban az esetek nagy többségében nem tehető meg teljes mértékben. Mégis, megfelelő digitális szűrőkkel, illetve a zajforrásoknál lévő mikrofonok jelének felhasználásával és a zavarjellel való kompenzációval kielégítően jó jel-zaj viszonyt lehet elérni.

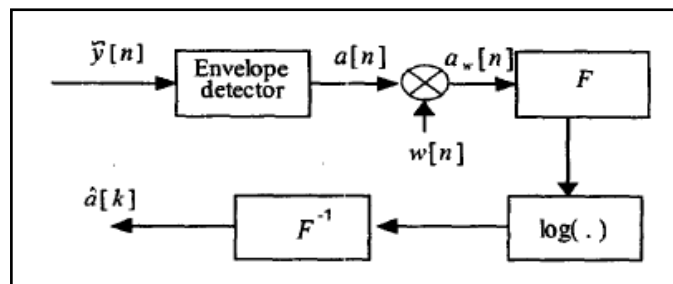
A beszédhez és a horkoláshoz tartozó jelek meglehetősen hasonlóak, ami nem csoda, hiszen ugyanazon a helyen keletkeznek, mindkettőt modulálják a beszédszervek, és a felső légutak akusztikus tulajdonságai. Ezt láthatjuk az 5. ábrán:



5. ábra - A horkolás és beszédhang összehasonlítása

Látható, hogy a horkoláshang és a beszédhang jele is nullaközepű pszeudo-periodikus jel. Mégis, ez az a tulajdonság, ami alapján különbséget tudunk tenni a két jel között. A horkolás jeléhez tartozó pszeudo-periódusidő (továbbiakban T_s) nagyobb és sokkal kevésbé ingadozik, mint a beszédhanghoz tartozó jel pszeudo-periódusideje (T_p). Tehát, a mért regisztrátumból valamilyen módon kiszámítva T_s és T_p nagyságát és ingadozásának mértékét lehetőségünk van a kétféle, amúgy nagyon hasonló jelalak megkülönböztetésére. Sőt, látni fogjuk, hogy az alvási apnoe szindrómás horkolás detektálására is jó ez a módszer.

A pszeudo-periódusidő meghatározása a tanulmányban a következő blokkvázlat alapján történik:



6. ábra - Pszeudo-periódusidő meghatározása

Az ábrán $\tilde{Y}[n]$ zajtól a lehető legjobb mértékben elválasztott horkolás vagy beszédhang egy nem túl hosszú (néhányszor 10 ms) mintasorozata. Azért csak egy rövid mintasorozatot dolgozunk fel egy lépésben, mert ebben a periódusidő viszonylag még állandónak tekinthető. Az $\tilde{Y}[n]$ jel burkoló görbéje $a[n]$, amely már tehát nem egy nulla közepű jel, hanem csak pozitív értékeket tartalmaz és periódusideje megegyezik $\tilde{Y}[n]$ pszeudo-periódusidejével. $w[n]$ egy olyan hosszú Hamming ablak, amilyen hosszú a vizsgált regisztrátum. Ennek szorzata $a[n]$ -nel $a_w[n]$. A blokkvázlaton F és F^{-1} a diszkrét Fourier-transzformációt és annak inverzét, míg $\log(\cdot)$ értelemszerűen a logaritmus képzést jelöli. Az így előálló időtartománybeli $\hat{a}[k]$ jel tehát a következő:

$$\hat{a}[k] = F^{-1}\{ \log(F\{a_w[n]\}) \}$$

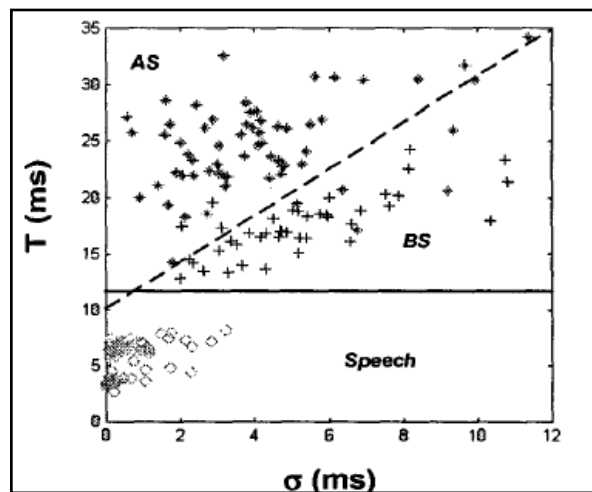
Az $\hat{a}[k]$ jel maximuma pont annál az időindexnél van, amely az adott regisztrátumhoz tartozó pszeudo-periódusidő. Ezek alapján a horkolás vagy beszédhang pszeudo-periódusideje nagyságának és ingadozásának meghatározása a következő lépésekből áll:

1. A zaj eltávolítása a horkoláshang diszkrét (mintavételezett) $y[n]$ jeléből, és így $\tilde{Y}[n]$ előállítása.
2. $\tilde{Y}[n]$ -ből mozgóablakszerűen, 50%-os átlapolással 80 ms hosszúságú regisztrátumok kivágása. A 80 ms már elegendően hosszú idő ahhoz, hogy az

már a nagyobb periódusidejű horkoláshangból is tartalmaz minimum 2-3 periódust, ugyanakkor még elegendően rövid ahhoz, hogy ennyi idő alatt a periódusidő állandónak tekinthető.

3. Minden regisztrátumra meghatározzuk a fent említett módon a pszeudo-periódusidőt.
4. A horkolás vagy beszédhang pszeudo-periódusidejének nagysága az előző pontban kapott eredmények átlaga (T), ingadozása pedig ugyanezen eredmények szórása (σ)

Sok mérést végezve és az eredményeket ábrázolva egy szórás-periódusidő diagramon, az alábbi biztató eredmény született:



7. ábra - A beszédhang, az egyszerű horkoláshang és az alvási apnoe szindrómás horkoláshang paramétereinek eloszlása

Az alvási apnoe szindrómás horkoláshoz (AS) tartozó eredményeket '*', az egyszerű horkoláshoz (BS) tartozókat '+', míg a beszédhez (Speech) tartozókat 'o' jelöli az ábrán. Az eredmények magukért beszélnek. Látható, hogy viszonylag jól elkülöníthető tartományokba esnek az egyes csoportokhoz tartozó eredmények, lehetőséget nyújtva ezáltal azok elkülönítésére.

A saját fejlesztésem kezdetekor a mért adatokat soros porton továbbítottam a PC-nek, és MATLAB-ban megvalósítottam ugyan ezt az algoritmust. Azonban azt tapasztaltam, hogy $\hat{a}[k]$ jelben az adott pszeudo-periódusidőnél lévő értéknél több nagyobb érték is szerepelt (többnyire kisebb időindexeknél). Kétségtelenül egy lokális maximum van a jelben a keresett pszeudo-periódusidőnek megfelelő helyen, de az esetek nagy többségében nem ez volt a globális maximum. Ennek oka az lehet, hogy ellentétben velük én nem alkalmaztam külön

mikrofonokat a lehetséges zajforrásoknál, ezáltal nem is tudtam olyan kiváló zajsűrűséget megvalósítani.

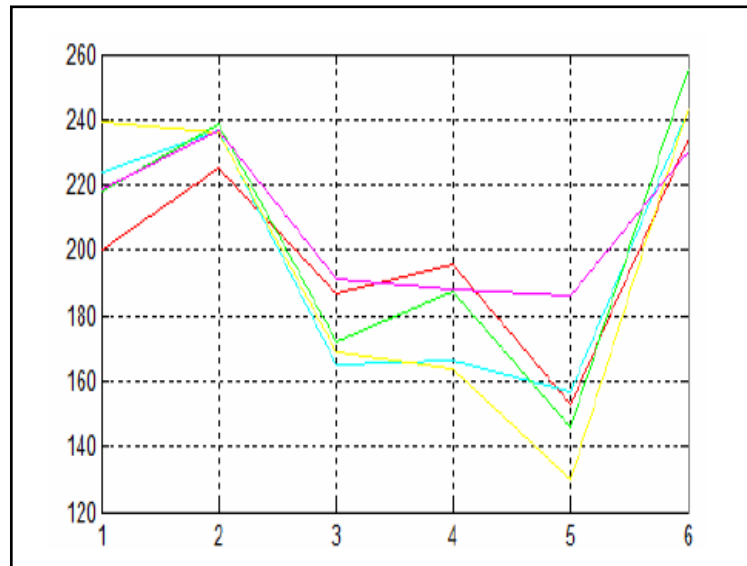
1.2.5. Az érzelmi állapot hatása a beszédhang alap frekvenciájára

A hang alapján a horkoláson túl egyéb jellemzők is felismerhetőek, detektálhatóak. Egy jó példa erre a Kolozsvári Műszaki Egyetem kutatói, Marius Vasile Ghiurcau, Adrian Lodin és Corneliu Rusu tanulmánya [14], amelyből kiderül, hogy lehetőség nyílik az izgalmi állapot hang alapján történő detektálására.

"Az európai emberek körében a beszédhang alapfrekvenciájának átlagos értéke a férfiak esetében 120 Hz, a nők esetében 220 Hz, míg a gyerekek esetében 330 Hz. Az alapfrekvencia értéke nem csak a beszélő személyétől függ, hanem sok egyéb hatás is befolyásolja, mint például az érzelmi állapot, az aktuális napszak és az egészségi állapot." Az érzelmi állapot tehát hatással van a beszéd alaphang frekvenciájának ingadozására. Ennek bizonyítására 10 ember, 5 férfi és 5 nő beszédét vizsgálták különböző (boldog, mérges, félelmes, unatkozó, szomorú és közömbös) érzelmi állapotokban. Mindegyik vizsgált személy, mind a 6 vizsgált érzelmi állapota során megmérték a beszédhang átlagos alaphang frekvenciáját és annak szórását.

A néhány másodperces beszédmintákat 16 kHz-es frekvenciával mintavételezték. A beszédmintákat 10 ms hosszúságú, 50%-os átlapolású részekre osztották. Ezen részek alapfrekvenciáját külön-külön megbecsülték. A becsült értékek átlaga és szórása adja az adott beszédminta átlagos alapfrekvenciáját, és annak szórását.

A 8. ábrán láthatjuk külön színnel jelölve a megvizsgált 5 nő hangjának átlagos alapfrekvenciáját a különböző érzelmi állapotokban. Az Y tengelyen a frekvencia értékek olvashatóak Hz-ben, míg az X tengelyen az egyes érzelmi állapotokat ábrázolták: (1) félelem, (2) boldogság, (3) unalom, (4) közömbösség, (5) szomorúság, (6) dühösség. Az 5 vizsgált személy eredményeit különböző színű törött vonalak mutatják az ábrán. Megállapítható, hogy a vizsgált alanyok hangjának alapfrekvenciája az érzelmi állapotoktól függően hasonló módon változott. A frekvencia értéke szomorú (5) állapotban a legalacsonyabb, míg dühös (6) állapotban a legmagasabb. A vizsgált férfiak eredményei is hasonló diagramot alkotnak.



8. ábra - Beszédhang alapfrekvenciája a különböző érzelmi állapotokban

A további elemzésekből kiderül, hogy az érzelmi állapotok közül az unalom (3) a legkevésbé kimutatható a hang alapján. Ebben az esetben, az alapfrekvencia a közömbös (4) állapotbelihez közeli. A legnagyobb eltérés a semlegestől a szomorúság állapota során tapasztalható, ekkor az alapfrekvencia a közömbös állapotbelihez képest akár 25%-al alacsonyabb is lehet. Szintén nagy eltérést mutat a mérgesség, a boldogság és a félelem állapota, amelyek során a frekvencia megemelkedik.

2. fejezet

A hardver bemutatása

2.1. A megvalósítás elvei

Az eszköznek a felvett (mintavételezett) hangok alapján autonóm módon el kell tudni dönteni a horkolás tényét. Az emberi jogi korlátok miatt a hanganyagot nem továbbíthatja más egységnek és nem tárolhatja el, kizárólag a horkolás tényét, illetve a megfigyelt személy állapotára vonatkozó információkat: mikor, mennyit horkolt, hogyan változott a horkolás módja az előző időszakhoz képest, stb. Ebből is következik, hogy a fejlesztés során egy nagyobb teljesítményű mikrovezérlőre van szükség, amely képes ellátni nagyobb számításigényű jelfeldolgozási feladatokat is. Az olcsóság is fontos szempont, hogy adott esetben egy átlagember is meg tudja vásárolni magának az eszközt, ezért a kimondottan jelfeldolgozási feladatokra szánt, viszont nagyon drága digitális jelfeldolgozó processzor (DSP) ötletét elvettem.

Manapság, architektúrájukat tekintve egyre inkább konvergencia mutatkozik a 32 bites ARM mikrovezérlők és a DSP-k között. Az ARM Cortex szériáját a nagy teljesítményhez képest alacsony ár jellemzi. Például, egy Cortex-M3 architektúrájú mikrovezérlő ma már körülbelül 2-3 ezer Ft-os áron beszerezhető. A Méréstechnika és Információs Rendszerek Tanszéken az *mbed NXP LPC1768* nevű fejlesztőkártya [16] segítségével lehetőség van a fent említett architektúrájú mikrovezérlővel való megismerkedéshez. Ez a processzor elegendően olcsó, és elegendően nagy teljesítményű a nagyobb számításigényű feladatok ellátására, így alkalmas arra, hogy az általam fejlesztett rendszer központi egységként működjön. A mikrovezérlő és a fejlesztőkártya a következőkben részletesen bemutatásra kerül.

2.2. Az analóg áramkör

Egy áramkörben, amelyben valamilyen analóg jelet mérünk alapvető fontosságú, hogy a tápfeszültség a lehető legnagyobb mértékben zajmentes legyen. Az általam tervezett kapcsolásban (9.ábra), ennek megfelelően az analóg tápfeszültséget a digitális tápfeszültségnek egy LC taggal való szűrésével állítom elő ($L=10\mu H$, $C=100nF$). Esetünkben különösen fontos az analóg tápfeszültség zajmentessége, hiszen a használt kondenzátor mikrofont közvetlenül az analóg tápfeszültséggel feszítem elő, így az ezen megjelenő zavarfeszültség közvetlenül megjelenik a hasznos jelben. Ezért, az előfeszítést biztosító felhúzó ellenállások közös pontja és a föld közé további kondenzátorokat helyeztem el, hogy az így keletkező RC tag az esetlegesen még jelen levő zavarokat kiszűrje.

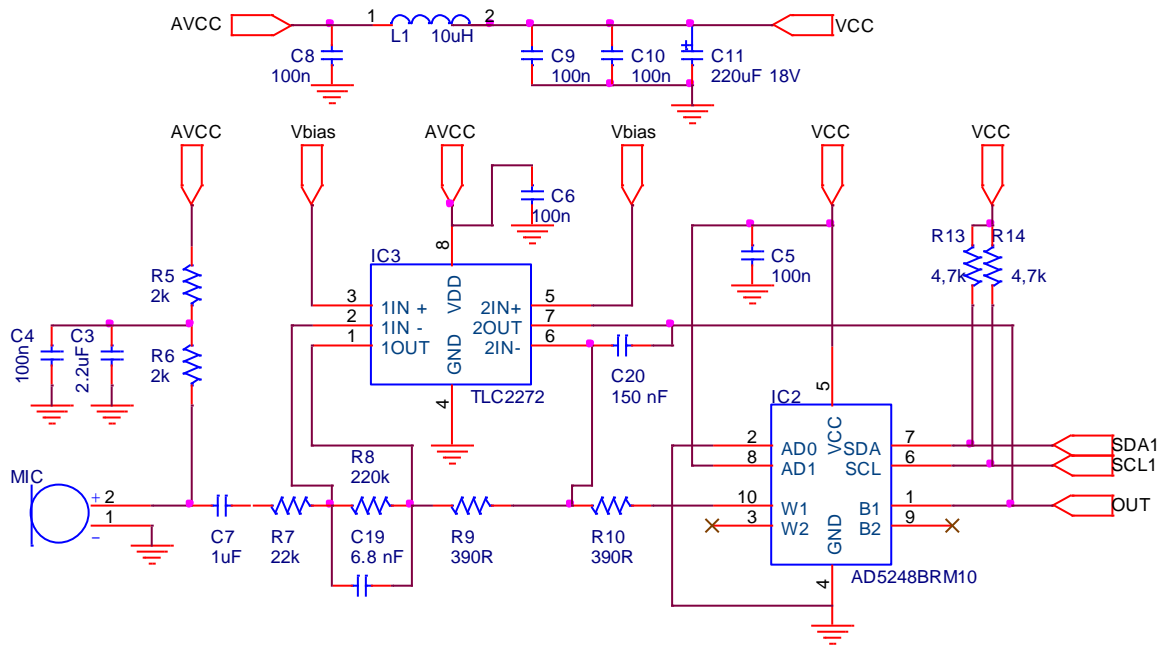
Az analóg jel erősítésére a *Texas Instruments*[®] által gyártott TLC2272 típusú műveleti erősítőt [17] alkalmazom, amely számunka ideális abból a szempontból, hogy egy tápfeszültség elegendő a számára (nem kell pozitív és negatív táp is). További előnyös tulajdonsága, hogy olcsó és kimenete kivezélhető a föld és a tápfeszültség közötti teljes tartományon. Kezdetben a szintén olcsón beszerezhető, LM324 típusú műveleti erősítő [18] alkalmazásával próbálkoztam. Ennek maximális kimeneti feszültsége (Output Voltage High Limit) az adatlap, és a méréseim alapján is 3.3 V-os táplálás esetén 2.1 V, 5 V-os táplálás esetén 3,8 V. Az erősítő kimeneti feszültsége, gerjesztés (hang) nélküli esetben a tápfeszültség felénél van (1,65 V illetve 2, 5V), ezért gerjesztés esetén a műveleti erősítő kimenete felfele eléri a kivezélhetőség határát. Ezzel szemben a TLC2272 képes megközelíteni a tápfeszültséget, amely az egész rendszerben egységesen 3,3V.

A 9. ábrán látható, hogy az erősítés két fokozatban történik. Mindkét fokozat gyakorlatilag egy fázisfordító alkapcsolás, amelynek földpontja el van tolva V_{bias} feszültséggel. V_{bias} -t az analóg tápfeszültségből egy feszültségosztóval állítom elő, úgy, hogy értéke éppen a tápfeszültség fele (1,65 V) legyen. Az első fokozat 10 szerez erősítést valósít meg a következő képlet szerint:

$$\left| -\frac{R_8}{R_7} \right| = \frac{220 \text{ k}\Omega}{22 \text{ k}\Omega} = 10$$

Az erősítő visszacsatoló ágában az R_8 (220 k Ω) ellenállással párhuzamosan egy 6,8 nF értékű kondenzátor kapcsolódik. Ezen párhuzamos RC tag egy elsőfokú aluláteresztő szűrőt valósít meg, amelynek törésponti frekvenciája:

$$f_1 = \frac{1}{2 \cdot \pi \cdot R_8 \cdot C_{19}} = \frac{1}{2 \cdot \pi \cdot 220 \text{ k}\Omega \cdot 6,8 \text{ nF}} = 106,4 \text{ Hz}$$



9. ábra - Az analóg rész kapcsolási rajza

A második fokozat szintén egy fázisfordító alapkiosztást valósít meg. Ennek visszacsatoló ágában egy, az *Analog Devices*[®] által gyártott, AD5248 típusú digitális potenciométer [19] helyeztem el, amellyel a fokozat erősítése hangolható. Az AD5248 egy kétsatornás, I²C interfészen keresztül programozható digitális potenciométer, amelynek értéke 0 Ω és 10 kΩ között változtatható. A digitális programozhatóságnak köszönhetően, lehetőségünk nyílik az erősítés értékét a mikrovezérlő által hangolni. Ennek köszönhetően az eszköz működés közben képes önmagát kalibrálni.

A második fokozat által beállítható maximális erősítés értéke:

$$\left| -\frac{R_{10} + R_{potméter}}{R_9} \right| = \frac{390 \Omega + 10 \text{ k}\Omega}{390 \Omega} \approx 26,6$$

Tehát, a két fokozat együttes erősítését figyelembe véve, a mikrofon jelének erősítése 10 és 266 között hangolható a digitális potenciométer értékének megfelelő beállításával.

Az elsőhöz hasonlóan a második fokozat visszacsatoló ágába is az ott lévő két soros ellenállással párhuzamosan kapcsolva egy kondenzátort helyeztem el. Ennek értékét (150 nF) úgy választottam meg, hogy az általa megvalósított aluláteresztő szűrő törésponti frekvenciája jó közelítéssel az első fokozatban lévő aluláteresztő szűrő töréspontjával essen egybe. A második fokozat maximális erősítése esetén e törésponti frekvencia értéke tehát:

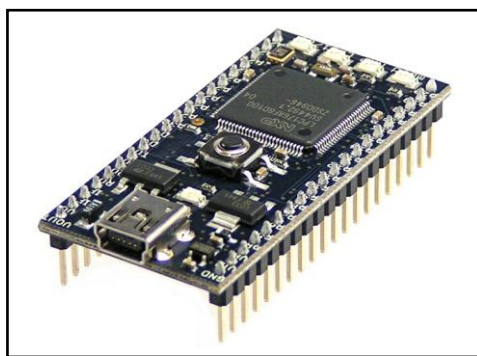
$$f_2 = \frac{1}{2 \cdot \pi \cdot (R_{10} + R_{potméter}) \cdot C_{20}} = \frac{1}{2 \cdot \pi \cdot (390 \Omega + 10 \text{ k}\Omega) \cdot 150 \text{ nF}} = 102,1 \text{ Hz}$$

A két fokozat sorosan kapcsolódik ezért a két elsőfokú aluláteresztő szűrő egy másodfokú aluláteresztő szűrőt alkot, amelynek törésponti frekvenciája $\approx 105 \text{ Hz}$. Ezen frekvencia felett az együttes analóg szűrő elnyomása 40 dB/dekád . Tekintve, hogy a mintavételi frekvencia értéke 2500 Hz , az analóg szűrő elnyomása a mintavételi frekvencia felénél (1250 Hz) több mint 40 dB . Tehát, ezen analóg szűrő átlapolódás gátló szűrőként biztosítja, hogy a mintavételezés után a spektrumok ne lapolódjanak át.

Az erősítők földpontja el van tolva V_{bias} értékével, ezért a kimeneten (OUT) egy $1,65 \text{ V}$ közepű jel keletkezik. A kapcsolásban ezt közvetlenül a mikrovezérlő beépített A/D átalakítójának bemenetére vezetem, amely 2500 Hz -es mintavételi frekvenciával mintavételezi azt.

2.3. Az alkalmazott mikrovezérlő bemutatása

A *2.1. fejezetben* említettek szerint egy Cortex-M3 architektúrájú mikrovezérlőt alkalmazok. Ehhez a Méréstechnika és Információs Rendszerek Tanszék rendelkezésemre bocsátott egy *mbed NXP LPC1768* fejlesztőkártyát (*10. ábra*), amelynek központi eleme a fent említett architektúrájú, *LPC1768* típusú mikrovezérlő [20].



10. ábra - mbed NXP fejlesztőkártya

2.3.1. Az ARM Cortex-M3 architektúra

Az ARM architektúra (Advanced RISC Machine) egy 32 bites, az ARM Limited cég által fejlesztett RISC CPU architektúra, amely ma igen népszerű a beágyazott rendszerek körében. Energiatakarékosságuk miatt az ARM architektúrájú CPU-k a vezetők a hordozható

elektronikai piacon - ahol az alacsony energiafogyasztás fontos tervezési szempont. Az ARM architektúra minden tagja egységesen a következő RISC tulajdonságokkal rendelkezik:

- Load/store architektúra, azaz csak a load/store utasítások fordulhatnak a memóriához.
- Egyszerű utasításokkal rendelkezik, amelyek végrehajtása egy gépi ciklust igényel.
- Utasítás készlete ortogonális (bármely utasításban bármely címzési mód használható).
- Általános felépítésű, 32 bites utasítások (egyszerűbb dekódolás és pipelineing).
- Adatcsatornás (pipeline) adatfeldolgozás.
- Huzalozott utasítás végrehajtás.
- Nagy méretű regisztertár, amelyek regiszterbankokra vannak felosztva.
- Harvard-architektúrájú (külön program-, és adatbusszal).

Az egyszerűbb felépítés kompenzálására, néhány egyedi jellemzőt is hozzáadtak:

- A legtöbb utasítás feltételesen végrehajtható utasítás, így csökkentve az elágazás idejét és kompenzálva az elágazás előrejelző (branch prediction) hiányát.
- 32-bites léptető, amellyel teljesítmény-veszteség nélkül lehet aritmetikai és cím-számolásokat végezni (barrel shifter)
- Erőteljes indexelt címzési módok
- Egyszerű, de gyors, két prioritási szintű megszakítás-alrendszer cserélt regiszter-készlettel

Az újabb ARM processzorok rendelkeznek egy tömörített utasítás-készlettel, a Thumb-bal, amely 16 bites utasításokat használ (de továbbra is 32 bites adatokkal dolgozik). A Thumb rövidebb utasítás kódjaival kevesebb funkció érhető el. Például csak az elágazások köthetők feltételhez és sok utasítás kód nem érheti el a CPU regisztereit. A rövidebb utasítás kódok használatával növekszik a kódsűrűség, annak ellenére, hogy néhány művelethez több utasítás szükséges. Különösen azokban a helyzetekben, amelyekben a memória, port vagy busz kevesebb mint 32 bit-re van korlátozva, a rövidebb Thumb utasítás kódok jobb teljesítményt nyújtanak, mint a 32 bites, a limitált memória-sávszélesség jobb kihasználása miatt.

Az ARM Cortex-M3 [21] egy szabványosított mikroprocesszor mag, amely egy 32 bites processzort, egy speciális busz architektúrát, fejlett megszakításkezelő egységet, debug rendszert és szabványos memória layoutot tartalmaz. A Cortex-M3 Harvard architektúrájú, tehát külön program-, és adatbusszal rendelkezik, és tipikus RISC processzor, tehát egy utasítást ~ 1 gépi ciklus alatt hajt végre pipeline műveletvégzés segítségével.

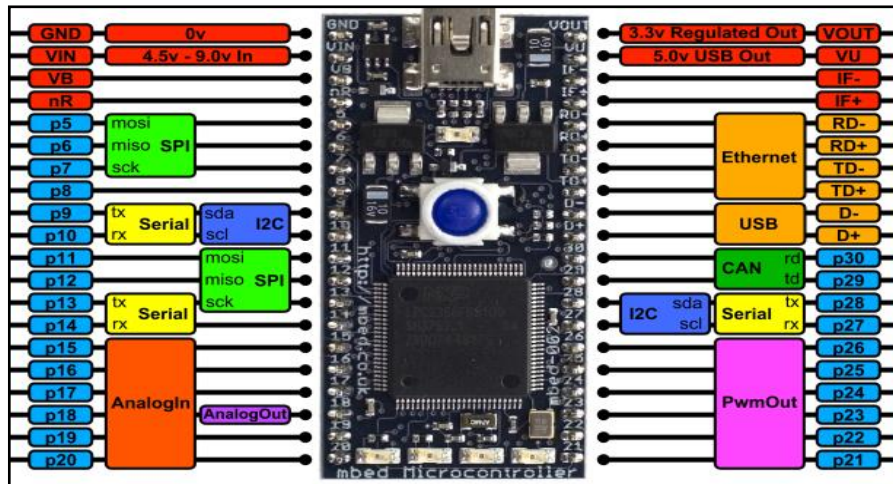
2.3.2. Az LPC1768 mikrovezérlő

Az LPC1768 tehát az előző pontban bemutatott architektúrájú mikrokontroller, amelynek legfőbb jellemzői a következők:

Órajelének frekvenciája akár 130 MHz is lehet. A fejlesztő kártyán ez az érték 96 MHz, amely egy 12 MHz-es külső oszcillátor frekvenciájának 8 szorozásával áll elő. A frekvencia többszörözését a mikrovezérlő belső PLL egysége végzi. Ez az érték (szemben a tanszéki *mitmót* rendszerben használt 8 bites AVR ATmega128 típusú mikrovezérlő maximális 8MHz-es frekvenciájával) elegendően nagy ahhoz, hogy akár nagyobb számításigényű feladatokat is valós időben el tudjon végezni a processzor.

512 Kbájt program memóriával (flash), és 64 Kbájt belső adatmemóriával rendelkezik, amely szintén többszöröse az előbb említett AVR mikrovezérlő 128 Kbájt program illetve 4Kbájt adatmemóriájának. Ekkora belső memória készlet mellett lehetőségünk nyílik nagyobb mennyiségű adat tárolására és azon való műveletvégzésekre, továbbá szintén nagyobb memóriaigényű SD kártya vezérlésre is.

Az LPC1768 egy beépített, 8 csatornás, 12 bites analóg-digitális konverterrel (ADC), illetve egy 10 bites digitális-analóg konverterrel (DAC) rendelkezik. Ezek közül az én alkalmazásomban az A/D átalakító a fontosabb, hiszen ez végzi a mikrofon erősített analóg jelének mintavételezését, és digitalizálását. A fejlesztő kártyán az analóg digitális konverter referencia feszültsége és az analóg tápfeszültség is a 3,3 V-os digitális tápfeszültség megfelelő LC taggal ($C=1\mu\text{F}$, $L=100\mu\text{H}$) való szűrésével áll elő, tehát a teljes A/D átalakítási tartomány 3,3 V. Az A/D átalakítás fokozatos közelítés (szukcesszív approximáció) elvén történik, amelynek gyakorisága 12 bites konverzió esetén is 200 kHz lehet. Alacsonyabb (például 8) bitszámú konverzió esetén ez az érték akár 1 MHz is lehet, ami szintén jóval nagyobb, mint az ATmega128 mikrovezérlő beépített 10 bites analóg-digitális átalakítójának maximális 125 kHz-es maximális gyakorisága. Az LPC1768 nyolc darab bemenő analóg csatornával rendelkezik, amelyek közül az mbed fejlesztőkártyán csak hat darabot vezettek ki (*p15-p20*), ahogyan az a *11. ábrán* látható.



11. ábra - Az mbed NXP fejlesztőkártya lábkiosztása

A mikrovezérlő a következő beépített soros kommunikációs interfészekkel rendelkezik:

- I²C busz, amelyek közül kettőt vezettek ki a fejlesztő kártyán.
- SPI periféria, amelyek közül kettőt vezettek ki a fejlesztő kártyán.
- Négy darab UART periféria belső FIFO, DMA és RS-485 támogatással. Ezek közül 3-at kivezettek az mbed fejlesztőkártya lábaira.
- CAN vezérlő
- USB host és USB device vezérlő
- Ethernet vezérlő

Ezekon kívül beépített, 8 csatornás, általános célú DMA controllerrel, 70 általános célú I/O lábbal, 4 darab általános célú időzítő/számláló egységgel, ultra alacsony fogyasztású Real Time Clock perifériával, Watchdog Timer (WDT) perifériával, 6 általános célú PWM kimenettel, PWM motorvezérlővel, illetve kvadratúra enkóderrel is rendelkezik.

2.3.3. A jelfeldolgozási műveletek a mikrovezérlőn

Az eszköznek a horkoláshang alapján kell eldöntenie a horkolás tényét, tehát a benne lévő processzornak a hangfelvételen valamilyen digitális jelfeldolgozást kell végeznie. Érdekes ezért megvizsgálunk, hogy az *LPC1768* mikrovezérlő milyen módon támogatja a jelfeldolgozási lépések végrehajtását. Az egyes műveletek időzítési paramétereit a Cortex-M3 maghoz tartozó technikai referencia leírásban [22] (Cortex-M3 Technical Reference Manual) olvashatjuk. A következőkben sorra veszem jelfeldolgozás során előforduló főbb műveleteket.

Összeadás, kivonás:

A mikrovezérlő hardveresen támogatja az összeadás, kivonás műveleteket, így a valós, 16, és 32 bites, valamint előjeles és előjel nélküli számok összeadását 1 órajelciklus alatt végzi. Az LPC1768 mikrovezérlő alapvetően nem jelfeldolgozásra szánt processzor (DSP), ezért a komplex számokkal való műveletvégzést nem támogatja hardveresen. Két komplex szám összeadása, illetve kivonása során a műveletet külön a valós és a képzetes részekre is el kell végeznünk. Tehát, egy komplex összeadás, illetve kivonás 2 órajelciklust igényel.

Szorzás:

A processzor beépített hardveres szorzó egységgel rendelkezik, így a valós, 16, illetve 32 bites, előjeles, és előjel nélküli számok szorzását is 1 órajelciklus alatt végzi. Két komplex szám szorzása a következő módon történik:

$$(a + j \cdot b) \cdot (c + j \cdot d) = (a \cdot c - b \cdot d) + j \cdot (b \cdot c + a \cdot d)$$

Külön kezelve a valós és képzetes részeket egy komplex szorzáshoz 4 db. valós szorzás és 1-1 db. összeadás, illetve kivonás szükséges, tehát a mikrovezérlőn 6 órajelciklust igényel. Komplex szám valóssal való szorzásához ezzel ellentétben, csak két szorzás művelet szükséges, így végrehajtása 2 órajelciklus ideig tart.

Osztás:

Általában, még a DSP processzorok sem támogatják az osztás műveletét, mert csak olyan rekurzív osztó algoritmus áll rendelkezésünkre, amely úgy végzi el több lépésben az osztás műveletét, hogy az egyes fokozatok bemeneteit az előző fokozatok kimenetei alkotják. Ez azt jelenti, hogy még az ún. párhuzamos osztóművek is úgy működnek, hogy az egyes fokozatok jelterjedés szempontjából sorba kapcsolódnak.

A Cortex-M3 magú processzorok esetében az osztás művelete a 32 bites előjeles, illetve előjel nélküli számok esetén is 2-12 órajelciklust igényel. Az osztás ideje függ az osztandótól és az osztótól, amelyek, ha közel vannak egymáshoz, akkor hamarabb végrehajtható a művelet.

MAC (Multiply And Accumulate):

A szorzás-akkumulálás annyira fontos és gyakori művelet, hogy a jelfeldolgozó processzorok külön utasítást használnak hozzá. Olyan hardver konfigurációra van szükség, amely képes ezt a műveletsort egy lépésben elvégezni. Ezért minden fixpontos és lebegőpontos DSP processzor tartalmaz speciális gyors szorzó-akkumuláló részegységet. A

fixpontos processzorok szorzói általában két 16 bites, kettes komplementes kódban ábrázolt számot fogadnak és szoroznak össze, az eredményt 32 biten ábrázolják. A teljes szorzás-akkumulálás (MAC) művelet végrehajtásához egy gépi ciklus szükséges.

A Cortex-M3 magú processzorok nem tartalmaznak speciális szorzó-akkumuláló részegységet, de egy szorzás és egy összeadás egymásutánjaként 2 órajelciklus alatt képesek elvégezni a MAC műveletet.

Konvolúció:

A digitális jelfeldolgozás egyik gyakori művelete a konvolúció, hiszen egy hálózat választás időtartományban a gerjesztés és az impulzusválasz konvolúciójával kapjuk. Diszkrét idejű rendszerben a konvolúciós integrál konvolúciós összeggé alakul. DSP feladatok során nincs szükségünk az időfüggvényre, csak a pillanatnyi kimeneti mintára. Egy FIR szűrő esetén ezt az alábbi összefüggésekkel számíthatjuk ki:

$$y(n) = \sum_{k=0}^{N-1} w(k) \cdot x(n-k), \quad n = 0 \dots N-1$$

Egy N hosszúságú FIR szűrő esetén a konvolúció művelete ennek megfelelően N^2 db. MAC műveletet jelent, amely egy Cortex-M3 magú mikrovezérlő esetében $2 \cdot N^2$ órajelciklust igényel.

DFT (diszkrét Fourier-transzformáció):

A digitális feldolgozó egységek elterjedése a Fourier-transzformáción alapuló eljárások virágzását eredményezte. Ha egy folyamatos, sávkorlátozott jelből N számú mintát veszünk, akkor e minták alapján végrehajtott Fourier-transzformáció (az általános képletnek időben diszkrét értékekre való átalakításával) az alábbi lesz:

$$X_k = \sum_{i=0}^{N-1} x_i \cdot e^{-j \cdot \frac{2 \cdot \pi}{N} \cdot k \cdot i}, \quad k = 0 \dots N-1$$

A DFT tehát N darab rendelkezésre álló minta felhasználásával állítja elő a spektrumot N darab pontban. A frekvenciatartományban 0-tól f_s -ig (mintavételezési frekvencia) N darab pontban egyenletesen állítja elő a spektrumot. Az X_k , a k-adik előállított Fourier-transzformált érték tehát $f = \frac{f_s}{N} \cdot k$ valós frekvenciának felel meg. Ez alapján számítható a DFT frekvenciafelbontása, amely $\Delta f = \frac{f_s}{N}$.

A diszkrét Fourier-transzformáció komplex adatok esetén N^2 komplex szorzást és $N \cdot (N - 1)$ komplex összeadást jelent. A műveletigény tehát igen nagy. Némileg javít a helyzeten, hogy a digitális jelfeldolgozás során a DFT bemenete valós számsorozat, így a komplex szorzás helyett komplex szám valóssal való szorzását végezzük, amely "csak" 2 órajelciklust igényel a 6 helyett. Így, egy N hosszúságú adatvektor diszkrét Fourier-transzformációja $2 \cdot N^2 + 2 \cdot N \cdot (N - 1)$ órajelciklust igényel egy Cortex-M3 magú mikrovezérlő esetében. Ez meglehetősen nagy, ezért dolgoztak ki a DFT szimmetriatulajdonságain alapuló kevésbé műveletigényes algoritmusokat.

FFT (gyors Fourier-transzformáció):

Az FFT algoritmus nagy gyorsítási lehetőséget jelent a diszkrét Fourier-transzformációhoz képest, ha a bemeneti adatvektor hossza 2 egész kitevőjű hatványa ($N = 2^m$). Ebben az esetben a diszkrét spektrum számítása $\frac{1}{2} \cdot N \cdot \log_2(N)$ komplex szorzást, és $N \cdot \log_2(N)$ komplex összeadást igényel [23]. A komplex összeadás 2, míg valós bemeneti adatok esetén a komplex szorzás szintén 2 órajelciklus ideig tart. Ennek megfelelően egy N hosszúságú, valós bemeneti adatvektor FFT-je

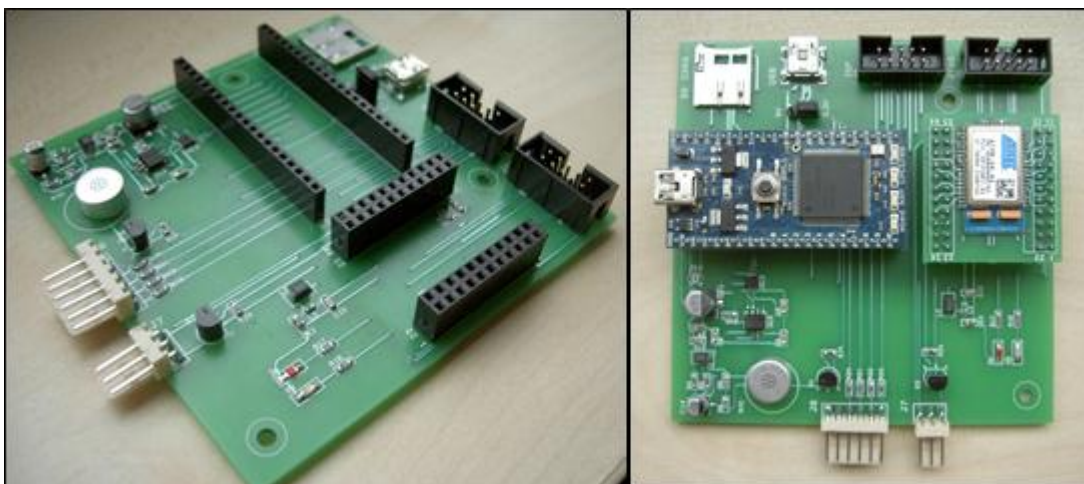
$$N \cdot \log_2(N) + 2 \cdot N \cdot \log_2(N) = 3 \cdot N \cdot \log_2(N)$$

órajelciklust igényel a Cortex-M3 magú mikrovezérlők esetében.

2.4. A megvalósított eszköz

A kész hardvert láthatjuk a *12.-13. ábrákon*, amelynek nyomtatott áramkörén megvalósítottam az előbbieken bemutatott analóg áramkört. A panelen egy-egy hüvelysort helyeztem el az mbed fejlesztőkártya, illetve a következőkben bemutatásra kerülő Zigbit modul részére. Az eszköz a tápellátását az mbed fejlesztőkártyán keresztül a PC USB portjáról kapja (5V). Vezérlő PC hiányában lehetőség van elemes vagy akkumulátoros tápellátásra is.

A panelre helyezett MicroSD kártya foglalat lehetővé teszi, hogy az egység egy MicroSD kártyára mentse a naplózási adatokat, például, hogy a vizsgált személy mikor és mennyit horkolt.



12. ábra - Az elkészült hardver a vezérlők
nélkül

13. ábra - A teljes eszköz

A hardvert úgy terveztem meg, hogy az alvászvizsgálathoz tartozó egyéb méréseket is tudjon végezni. Az előző félévekben többek között nyúlásmérő bélyeges terhelésmérő szenzorokkal is foglalkoztam. Az ágy négy lába alatt mérve a terhelést lehetőség adódik a páciens éjszakai mozgásainak megfigyelésére is. A panelt ennek megfelelően alkalmassá tettem arra, hogy négy darab ilyen nyúlásmérő bélyeges terhelésmérő szenzort csatlakoztathassunk rá.

Az *1.2.1. fejezetben* bemutatott irodalomban láttuk, hogy kapacitív elven működő nyomásérzékelőt helyezve a lepedő alá, megfigyelhetjük a mellkas mozgását is. Évfolyamtársam Erdős Csanád Gergely előző félévben egy ilyen kapacitív elven működő terhelés érzékelőt fejlesztett, amely a soros I²C interfészen keresztül csatlakoztatható az én alvászvizsgáló eszközömrre is.

Ezekén kívül mozgásérzékelő szenzort is csatlakoztathatunk az egységre, amelynek segítségével például érzékelni tudjuk, hogy a megfigyelt személy felkelt az ágyból. Ugyancsak a későbbi bővítési lehetőségekre gondolva kivezettem a panelen a soros I2C, SPI, RS232 és USB kommunikációs vonalakat.

Az egység teljes kapcsolási rajza a *1.* illetve a *2. mellékletben*, míg a nyomtatott áramköri terve a *3. mellékletben* látható.

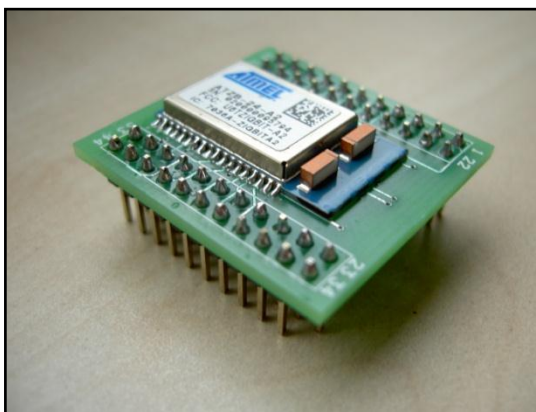
2.5. Kommunikáció

2.5.1. Vezetékes kommunikáció

Az mbed NXP fejlesztőpanel programozó kábele egyben egy soros USART kommunikációs vonalat is megvalósít az eszköz és a PC között. Rendeltetésszerű használat esetén erre nincs szükség, hiszen az egység önállóan, autonóm módon dolgozza fel a hangfelvételt, és azt nem tárolhatja el és nem is továbbíthatja a PC-nek. Azonban a jelfeldolgozási algoritmusokat kezdetben MATLAB környezetben fejlesztettem, illetve valósítottam meg. Tehát, az első verzióban az egység a mért adatokat soros porton keresztül továbbította a PC-nek.

3.5.2. Zigbee alapú kommunikáció

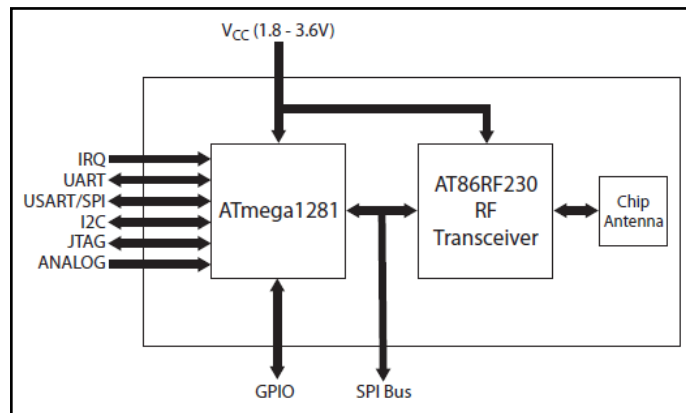
Ahogy az a bevezetőben említettem, a horkolás detektáló egység egy, a tanszéken kialakítandó intelligens szobába készül. Az itt elhelyezésre kerülő vezérlők, egységesen, zigbee-n keresztül kommunikálnak egymással, és a vezérlő PC-vel. Ennek megfelelően, a hardvert alkalmassá tettem a zigbee kommunikációra is, amelyet egy Atmel[®] által gyártott ATZB-24-A2 típusú Zigbit modul (14. ábra) rendszerbe való illesztésével értem el.



14. ábra - ATZB-24-A2 Zigbit modul az adapteren

A Zigbit modul sajnos meglehetősen drága (kb. 7000Ft), ezért terveztem hozzá egy adapter nyomtatott áramkört, aminek segítségével egy tűsoron keresztül tudom csatlakoztatni az egységemhez, így adott esetben más alkalmazásomban is fel tudom majd használni. Az adapter nyomtatott áramköri terve a 4. mellékletben látható.

A Zigbit modul önállóan is alkalmas lehet egy kisebb rendszer vezérlésére, hiszen található benne egy ATmega1281 típusú kisebb teljesítményű mikrovezérlő, ahogyan a modul blokkvázlatát ábrázoló 15. ábrán láthatjuk.



15. ábra - A Zigbit modul blokkvázlata

A modulban található még egy AT86RF230 típusú zigbee adóvevő egység, és az ehhez szükséges antennák felületszerelt kivitelben (chip antenna). A zigbee adóvevő SPI periférián keresztül csatlakozik a mikrovezérlőhöz, amely annak teljes vezérlését ellátja. Így a modulban lévő mikrokontrollerrel USART periférián keresztül kommunikálva lebonyolítható a teljes zigbee kommunikáció.

A rádió adóvevő egység a 2.4-2.4835 GHz-es tartományban működik, amelyben 16 különböző csatorna használható. A csatornák 5 MHz-re vannak egymástól. A kimeneti teljesítmény -17...+3 dBm tartományban 16 lépésben állítható, a vevő érzékenység -101 dBm. Az adatátviteli sebesség 250 kbps.

3. fejezet

Horkolás detektálására alkalmas algoritmusok

3.1. A hang tulajdonságai

"Hangnak a szubjektív hangérzetet keltő, levegőben terjedő longitudinális hullámokat nevezzük. Általánosabb értelemben a bármilyen rugalmas közegben terjedő longitudinális hullám a hanghullám." Olvashatjuk Holics László "Fizika 1" [24] című könyvének 401. oldalán.

Amit mi hangként érzékelünk az a mozgási energia egy fajtája, az akusztikai energia. Az akusztikai energia valamilyen fizikai közegben (pl. levegő) fellépő nyomásváltozás, illetőleg annak továbbterjedése (hullám). Egy teljes periódus két részből áll: a periódus első felében a levegő molekulák összesűrűsödnek (magasabb nyomás), majd ezt követi egy ritkulási szakasz (alacsonyabb nyomás). Minél nagyobb mértékű a sűrűsödés, illetve ritkulás, annál nagyobb a hangnyomás és ebből kifolyólag az amplitúdó.

Az egységnyi idő alatti levegő nyomásváltozások számát a hullám frekvenciájának hívjuk, ennek mértékegysége a Hz (Hertz). Azt az időt, amely alatt egy teljes hullám lejátszódik, periódusidőnek hívjuk, ami a frekvencia reciproka.

A hanghullám a levegőben kb. 340 m/s sebességgel terjed. Ez többek közt függ a páratartalomtól és a légköri nyomástól is, viszont nem függ a frekvenciától. Az a távolság, amelyet egy adott frekvenciájú hanghullám egy periódus alatt tesz meg, hullámhossznak nevezzük. (hullámhossz = hang terjedési sebessége / frekvencia).

"Azt, hogy egy hangot milyen magasnak hallunk, a hang frekvenciája szabja meg. A magasság a frekvenciával nő. A szinuszos hanghullám az ún. *tiszta hang*. A hangvilla és a megfelelő elektromos generátorral működtetett hangszóró hangja tiszta hang. A zenei hang kvázi periodikus hullám, de nem szinuszos. A magasságérzetet itt is elsősorban a periodikus rezgés frekvenciája szabja meg. A rezgés felbontható szinuszos hullámok összegére (Fourier-

felbontás). A *hangszín* érzetét a zenei hangban előforduló *felharmonikusok* adják. Az ún. *alaphang* a hang frekvenciájával megegyező szinuszos összetevő. A zenei hang Fourier-felbontásában szereplő, az alaphang frekvenciájánál nagyobb frekvenciájú hullámokat *felharmonikusoknak* nevezzük. Szigorúan *periodikus hang* felharmonikusainak frekvenciája az alaphang frekvenciájának egész számszorosa. A felharmonikusok amplitúdóinak az alaphang amplitúdójához viszonyított arányai szabják meg a *hangszín* érzetét." [24]

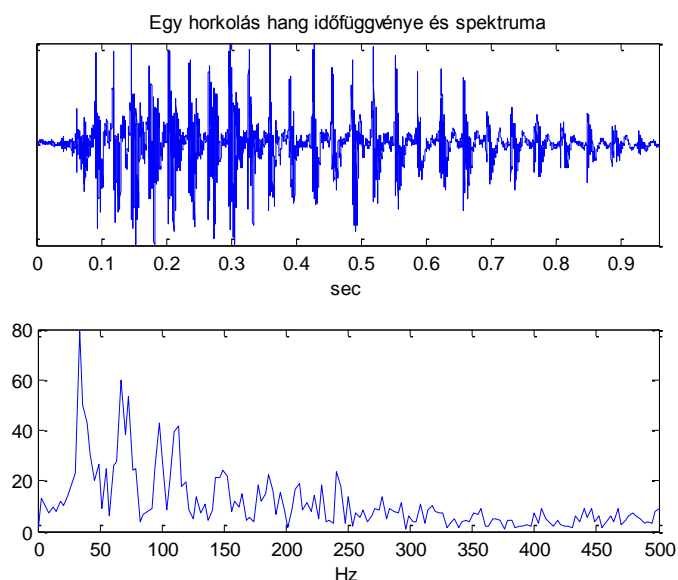
A horkoláshang az emberi beszédhanghoz hasonlóan ugyan nem szigorúan periodikus hang, de a rövid időtartamú vizsgálat során közel periodikusnak tekinthető (pszeudo-periodikus). Így, a horkolás, illetve beszédhangokhoz hangmagasságot rendelhetünk, amelyet az adott hang pszeudo-periódusidejének megfelelő alaphang frekvenciája határoz meg.

Az ember által hallható frekvenciatartomány: 20 Hz - 20 kHz. Ahogyan azt az *1.2.5. fejezetben* bemutatott irodalomban [14] olvashattuk: "Az európai emberek körében a beszédhang alapfrekvenciájának átlagos értéke a férfiak esetében 120 Hz, a nők esetében 220 Hz, míg a gyerekek esetében 330 Hz." Ezt igazolja az *1.2.4. fejezetben* tárgyalt kutatás [13], amelynek eredménye a *7. ábrán* látható. Az ábráról leolvasható, hogy a bemutatott algoritmus segítségével megmért beszédhangok pszeudo-periódusidejére 3 ms és 8 ms közötti értékek adódtak, amelyeknek pont a 100 - 400 Hz frekvenciatartomány felel meg. Szintén a *7. ábrán* látható, hogy a horkoláshangok pszeudo-periódusideje ennél nagyobb (15 - 35 ms), ennek megfelelően alapfrekvenciájuk kisebb, a 30 - 60 Hz értéktartományban helyezkedik el.

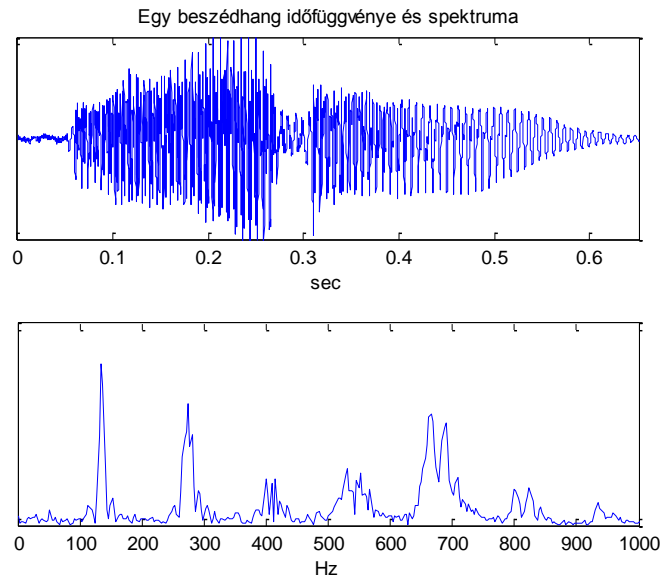
A hanghullám elektronikai reprezentációja egy, a hang ütemének megfelelően változó feszültség. Ezt mintavételezve lehetőségünk nyílik a hangok digitális feldolgozására. A mintavételi tétel szerint egy jel mintavételezése során, akkor nem veszítünk információt, ha a mintavételi frekvencia a jel sávkorlátjának kétszeresénél nagyobb. A minél kevesebb információ elvesztéséhez biztosítani szeretnénk, hogy az alapfrekvenciához tartozó felharmonikus frekvenciák se lapolódjanak át. Mindezekkel együtt, a horkoláshang elemzéséhez tehát elegendő lenne 2-3 kHz-es mintavételi frekvencia. A megvalósításban, mint ahogyan majd látni is fogjuk, 2500 Hz-es mintavételi frekvenciát alkalmazok.

3.2. Horkolás detektálása

Alapvető célom az alvás során keletkező, illetve jelen lévő hangok alapján a megfigyelt személy horkolásának detektálása. A horkoláshang pszeudo-periodikus volta miatt, jól elkülönül az éjszaka során esetlegesen jelen levő egyéb zajoktól (utcaról besűrűdő járművek zaja, ajtó nyitása/zárása, egyéb zörejek), amelyek jelalakjában a nevükből is adódóan (zaj, zörej) nincsen periodikusság, nincsen kitüntetett f_0 alaphang frekvencia sem. Sajnos az alvás közben esetlegesen bekövetkező beszédhez tartozó hangtól való megkülönböztetés nehezebb, hiszen a beszédhang is pszeudo-periodikus. Azonban, az 1.2.4. fejezetben bemutatott irodalomban [13] láttuk, hogy a horkolás és a beszédhang a hangmagasság alapján jól elkülönül egymástól: a horkoláshang a 30 - 60 Hz, a beszédhang a 100 - 400 Hz tartományban helyezkedik el. A 16.-17. ábrákon az általam felvett egy-egy horkolás, és beszédhang időfüggvényét, illetve spektrumát láthatjuk. Jól megfigyelhető az ábrákon, hogy a horkoláshang spektrumában 40 Hz, a beszédhang spektrumában 140 Hz körüli az alapharmonikus frekvencia értéke.



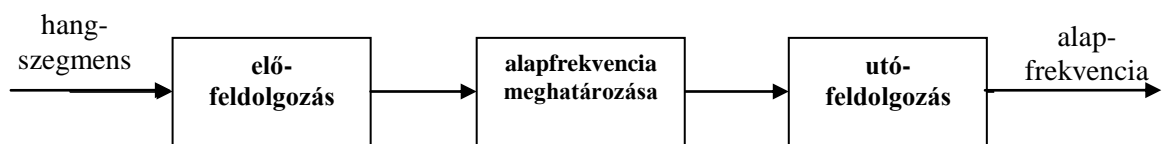
16. ábra - Egy horkoláshang időfüggvénye és spektruma



17. ábra - A "három" szóhoz tartozó beszédhang időfüggvénye és spektruma

Tehát, a horkolás hang detektálását visszavezethetjük a 30-60 Hz tartományban elhelyezkedő alaphanggal rendelkező pszeudo-periodikus hangok detektálására. A hangmagasság detektálás egyszólamú környezetben a beszéd analízisben, illetve a zenei tartalom kinyerésében is az egyik legjelentősebb eszköz. Ennek megfelelően bő irodalommal rendelkezik.

A hangmagasság detektálás célja tehát, a pszeudo-periodikus hang f_0 alaphangfrekvenciájának meghatározása. Ezen eljárások 3 fő lépésre oszthatóak: Előfeldolgozás, alaphangfrekvencia meghatározás, utófeldolgozás.



3.2.1. Előfeldolgozás

Az előfeldolgozó egység feladata, hogy csökkentse a bemeneti adat mennyiségét ezzel megkönnyítse az alaphangfrekvencia meghatározó egység munkáját. Az előfeldolgozás elsősorban a zajok elnyomására és azon tulajdonságok kiemelésére szolgál, amik pontosabbá teszik a detektálást.

3.2.1.1. Szűrés

A digitális szűrők két típusra oszthatók: véges impulzusválaszú FIR (Finite Impulse Response) és végtelen impulzusválaszú IIR (Infinite Impulse Response) szűrőkre. A két típus kimenőjelét meghatározó konvolúciós összeg az alábbi összefüggésekkel adható meg:

$$FIR: y_n = \sum_{k=0}^{N-1} h_k \cdot x_{n-k}$$

$$IIR: y_n = \sum_{k=0}^{\infty} h_k \cdot x_{n-k}$$

A bemenőjelet x , a kimenőjelet y , az impulzusválaszt h , jelöli. A FIR szűrő esetében az impulzus válasz mintái maguk a szűrőegyütthatók, a szűrő fokszáma ezen együtthatók száma (N). Az IIR szűrő fenti formulája a gyakorlati számításokra alkalmatlan. Ezért általában a következő rekurzív alakban fejezik ki:

$$y_n = \sum_{k=0}^N a_k \cdot x_{n-k} - \sum_{k=1}^M b_k \cdot y_{n-k}$$

Az a_k , illetve b_k szűrőegyütthatók a szűrő átviteli függvényében a számláló, illetve a nevező együtthatóit alkotják.

Fontos kérdés, hogy FIR, vagy IIR szűrőt alkalmazzunk. Mindkét szűrő típusnak vannak előnyei és hátrányai is, ezek a következők:

- A FIR szűrők tökéletesen lineáris fázismenettel rendelkeznek. Azokban az alkalmazásokban, ahol a fázistorzítás nem megengedett (pl. alakhű jelátvitel igénye), csak FIR szűrőket használhatunk. Az IIR szűrők fázismenete nemlineáris, különösen a sávhatárok környezetében.
- A FIR szűrők nem rekurzív kialakításúak, ezért mindig stabilak. Az IIR szűrők stabilitása nem minden esetben garantálható.
- A kerekítési zaj és az együtthatók kvantálási hibája, amit az ábrázolás véges szóhossza okoz, FIR szűrők esetén jóval kisebb problémát jelent, mint az IIR szűrők esetében.
- Azonos amplitúdómenetet megvalósító, meredek levágású szűrők esetén a FIR kialakítás lényegesen több együtthatót igényel, mint az IIR.
- A FIR szűrők megvalósítása egyszerű (kis számításigényű implementáció), míg az IIR szűrők megvalósítására speciális struktúrákat alkalmaznak, mint a *lattice*, *hullámdigitális*, vagy a *rezonátoros struktúra*, amelyek nagy számításigényűek.

- A hagyományos analóg szűrőformulák (Butterworth, Csebisev, inverz Csebisev, elliptikus szűrők) egyértelműen transzformálhatók IIR digitális szűrők specifikációjává. A FIR szűrőknek nincs analóg megfelelőjük, ezért szintézisük zárt formula helyett a frekvenciamenet pontonkénti megadásával lehetséges. Ezért a FIR szűrők tervezése algebrailag bonyolultabb, mint az IIR szűrőké.
- Az IIR szűrők színes, míg a FIR szűrők fehér kvantálási zajjal rendelkeznek.
- Az IIR szűrőknek nagy, a FIR szűrőknek kicsi a paraméterérzékenysége.

Számunkra a valós idejű jelfeldolgozás miatt fontos követelmény a kis számításigényű megvalósíthatóság a mikrovezérlőn, amely a véges impulzus válaszu (FIR) szűrő mellett szól. A mikrovezérlő a véges számábrázolási képessége miatt a megtervezett szűrőegyütthetők kvantált értékeivel tud csak számolni. Ez a tény egy IIR szűrő esetében instabilitáshoz is vezethetne. A FIR szűrők használatát indokolja az is, hogy az újabb fejlesztésű nagyteljesítményű mikrovezérlők architektúráját és utasításkészletét elsősorban a FIR szűrők minél könnyebb megvalósításához illesztették. Mindezeket figyelembe véve, FIR szűrő alkalmazása mellett döntöttem.

Az f_0 alapharmonikus gyakran nem a legerősebb frekvenciakomponens a spektrumban. Ennek oka, hogy a hangszálakban keletkező hangot a gégefő feletti üregek (garat, orrüreg és szájüreg) rezonátor-rendszerként viselkedve módosítják: a felharmonikusok egyes csoportjait felerősítik (rezonanciahelyek), másokat gyengítenek. Ezeket a rezonanciahelyeket és környéküket a beszéd analízisben *formánsnak* nevezik. A rezonancia frekvenciája a *formánsfrekvencia*. A formánst jellemzi még *sáv szélessége* és *amplitúdója*. Az előbbi alatt szemléletesen a rezonanciahely frekvenciabeli kiterjedését, az utóbbi alatt az erősítés mértékét értjük. [25] A formáns tartomány a beszédelemzés magánhangzó felismerésének fontos tényezője. A formáns tartományra jellemző az állandóság. Például, a gyermek magasan hangzó "á" hangjában a formáns tartomány ugyanott van, mint a felnőtt mélyebben hangzó "á" hangjában. Minden magánhangzóhoz 2 vagy 3 formáns tartomány tartozik.

Tehát, az f_0 alapfrekvencia meghatározása során nem a legerősebb frekvencia komponens keressük, hanem a legkisebb frekvenciájú komponens a spektrumban. Láttuk, hogy az alapfrekvencia a horkolás illetve beszédhang esetén a 30 - 400 Hz tartományban helyezkedik el. Az első formáns tartomány rendszerint 400 Hz és 1000 Hz között található, ezért a 2.-8. harmonikus komponensek gyakran erősebbek az alapharmonikusnál. Az erős harmonikusok a hangmagasság meghatározását jelentősen megnehezítik, és harmonikus illetve szubharmonikus hiba forrásai lehetnek, amelyek során az alapharmonikus helyett

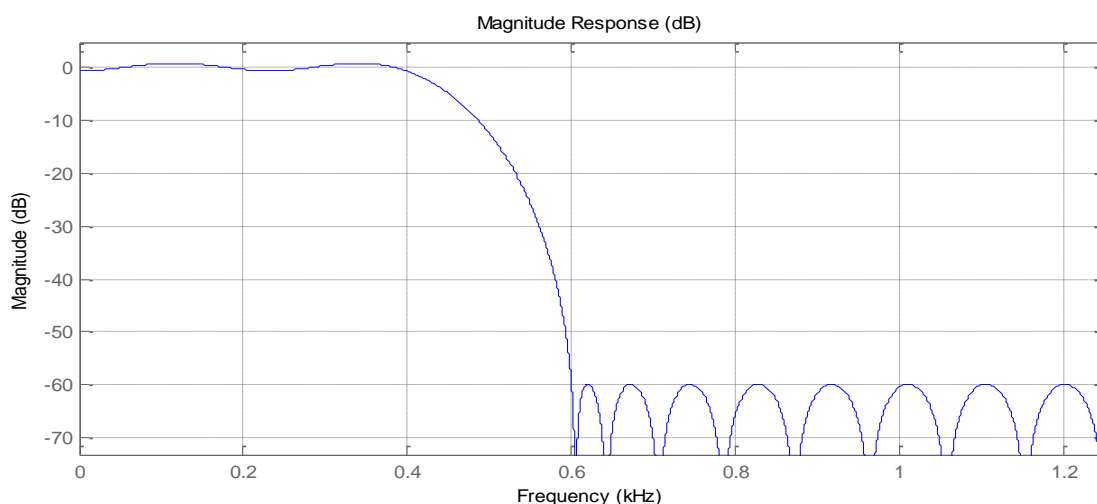
valamelyik harmonikus, illetve szubharmonikus detektáljuk. Ezen harmonikusokat célszerű kiszűrni a jelből az alaphfrekvencia meghatározását végző jelfeldolgozási algoritmus futtatása előtt.

Tekintve, hogy az alapharmonikus frekvenciák a 30-400 Hz tartományban helyezkednek el, a 400 Hz feletti frekvencia komponensek nem hasznosak a hangmagasság detektálása szempontjából. Ezért, egy olyan aluláteresztő szűrőt kell alkalmaznunk, amelynek sávszélessége ~400 Hz. Céлом az, hogy a kívánt hatást minél kisebb foksámú FIR szűrő alkalmazásával érjem el. Ehhez a következő specifikációt szabtam meg a szűrőnek:

áteresztőtartomány	0 ... 400 Hz	0,5 dB ingadozás
átmenet tartomány	400 ... 600 Hz	-
zárótartomány	600 ... 1250 Hz	60 dB elnyomás

1. táblázat - Az alkalmazott szűrő specifikációja

A 0.5 dB az áteresztőtartományban ($10^{-(0,5/20)} = 1,059$) kb. 6%-os ingadozásnak míg, a 60 dB a zárótartományban ($10^{-(60/20)} = 0.001$) 1000-ed részre való elnyomásnak felel meg. Ezen nem túl magas követelmények indokoltnak tekinthetők. Tekintve, hogy az áteresztő- és átmenet tartomány szélessége viszonylag nagy, ezért a specifikáció teljesítéséhez nem szükséges túl nagy foksámú FIR szűrő. A MATLAB-ban a *Remez algoritmust* használó *firpm()* függvénnyel tervezett szűrő 25 együttható segítségével teljesítette a specifikációt (18. ábra).



18. ábra - 400 Hz sávszélességű aluláteresztő szűrő

A mérés során alacsony frekvenciás zajok lehetnek jelen, gondoljunk például egy, a szobában zúgó ventilátorra. Ezek a jel időtartománybeli alakjában alapvonal vándorlást

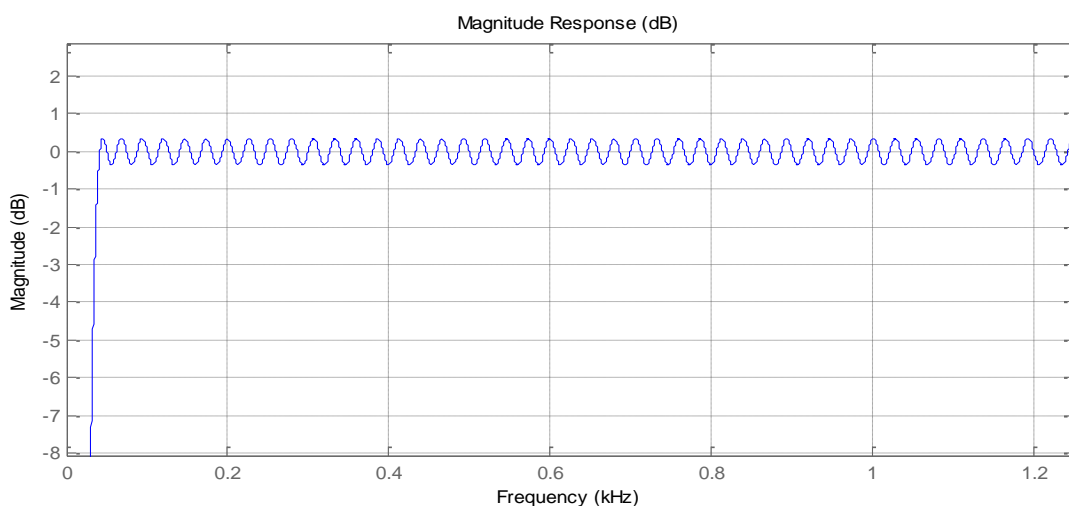
eredményeznek. Később látni fogjuk, hogy az alapprofrekvencia meghatározása során a jelet 100 ms hosszúságú szegmensekre bontom, és ezen szegmensekhez határozom meg a hozzájuk tartozó pszeudo-periódusidőt. A 100 ms elegendően rövid idő ahhoz, hogy a beszéd illetve horkoláshang ez alatt közel periodikusnak tekinthető, továbbá elegendően hosszú ahhoz, hogy a legalacsonyabb frekvenciájú horkolás hangokból (30 Hz) is legalább 3 periódust tartalmazzon. Az alacsony frekvenciás zajokból azonban kevesebb periódust tartalmaz (10 Hz alattiakból egy teljes periódust sem), ami nagyban meghamisítja a 30-400 Hz tartományban az alapharmonikus detektálását. Ezért, a megvalósításomban elengedhetetlen egy felüláteresztő FIR szűrő alkalmazása, amellyel szemben a következő követelményeket támasztottam:

zárótartomány	0 ... 20 Hz	60 dB elnyomás
átmenet tartomány	20 ... 30 Hz	-
áteresztőtartomány	30 ... 1250 Hz	0,5 dB ingadozás

2. táblázat - Felüláteresztő szűrő specifikációja

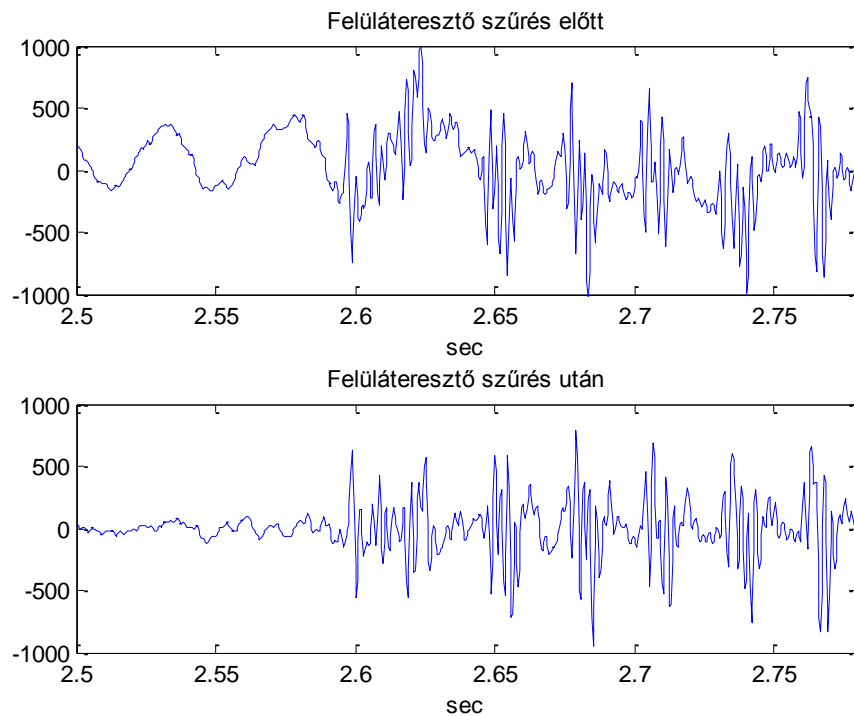
A zárótartomány, és az átmenet tartomány szélessége nagyon kicsi az áteresztőtartományhoz képest ezért ezen specifikációt csak viszonylag nagy fókuszú FIR szűrő teljesíti. Ez hátrány, hiszen a sok szűrőegyüttható nagy számításigényt jelent a mikrovezérlőnek, azonban látni fogjuk, hogy az alkalmazott LPC1768 típusú mikrovezérlő ezzel a feladattal még bőven megbirkózik a valós idejű megvalósításban is.

A MATLAB-ban a *Remez algoritmust* használó *firpm()* függvénnyel tervezett szűrő 189 együttható segítségével teljesítette a fenti specifikációt.



19. ábra - Az alkalmazott felüláteresztő szűrő

A felüláteresztő szűrés hatását láthatjuk a következő ábrán. Megfigyelhetjük, hogy a jel időtartománybeli alakjában a szűrés előtt alapvonal vándorlás tapasztalható, amely a szűrés után megszűnik.



20. ábra - Felüláteresztő szűrés hatása

3.2.1.2. Középkivágás:

A középkivágás a különböző alappfrekvencia meghatározási algoritmusok jeleket előkészítő módszere [26]. Egy nemlineáris torzítást végez a jelen, amelyet a következő formula segítségével írhatunk le:

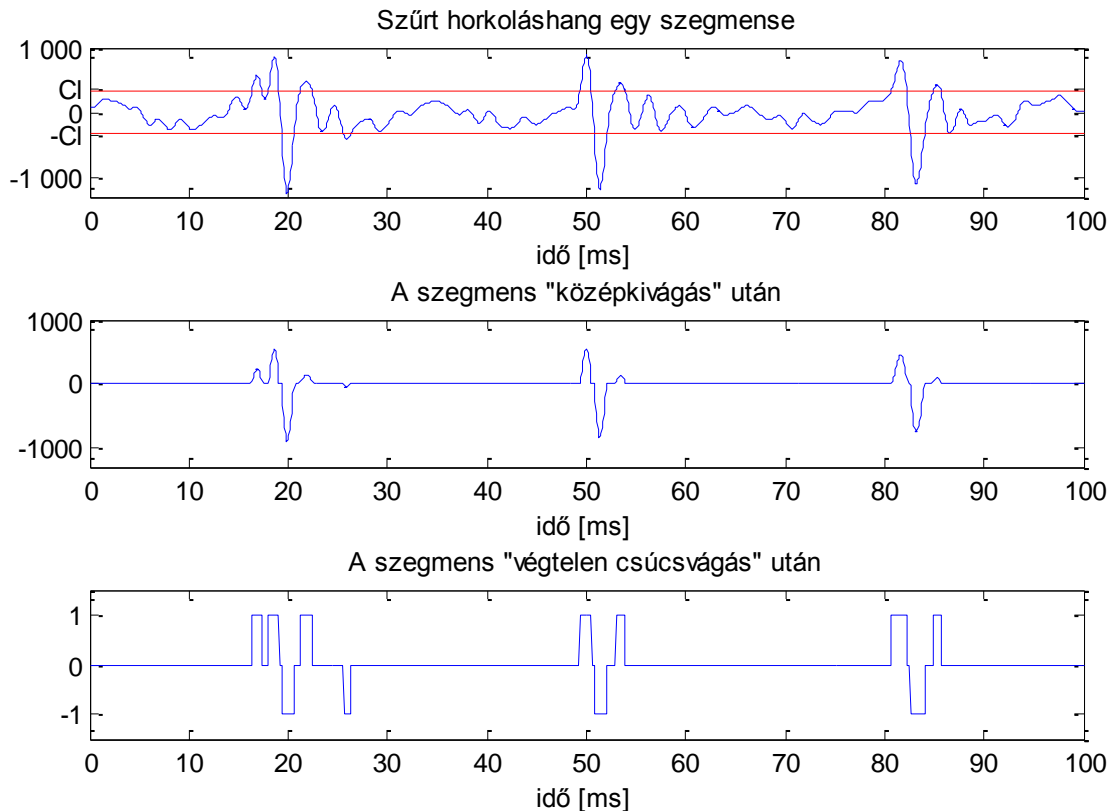
$$y(n) = f[x(n)] = \begin{cases} x(n) - C_l, & \text{ha } x > C_l \\ 0, & \text{ha } |x| \leq C_l \\ x(n) + C_l, & \text{ha } x < -C_l \end{cases}$$

A képletben $x(n)$ a feldolgozandó jel, $y(n)$ középkivágott jel, C_l pedig a vágási szint (clipping level).

Egy másik hasonló nemlineáris előfeldolgozási eljárás az ún. végtelen csúcsvágás, amely során nemcsak a jel közepét vágjuk ki, hanem a vágási szint felett lévő csúcsokat 1 értékűnek állítjuk be:

$$y(n) = \text{sgn}[x(n)] = \begin{cases} 1, & \text{ha } x > C_l \\ 0, & \text{ha } |x| \leq C_l \\ -1, & \text{ha } x < -C_l \end{cases}$$

A C_l vágási szint értéke általában, és mint majd látni fogjuk az én alkalmazásomban is a jel csúcserkének 30%-a [26]. A 21. ábrán egy horkolóshangból kivágott 100 ms hosszúságú szegmensen mutatom be az előző két módszer hatását.



21. ábra - Középkivágás és végtelen csúcsvágás alkalmazása

A középkivágási technikák segítségével megszabadulunk a két horkolás, illetve beszédhang között jelen lévő kis amplitúdójú zajoktól is.

Egy nulla középértékű jel alapharmonikus frekvenciájának meghatározása szempontjából érdektelen a jel amplitúdója. Így, a végtelen csúcskivágás alkalmazásával nem veszítünk a hangmagasságra vonatkozó információból, sőt a számításigény szempontjából kifejezetten előnyös ez a módszer, hiszen a nagy amplitúdók helyett csak két különböző értékkel (+1, -1) kell számolnunk.

3.2.2. Alapfrekvencia meghatározása

Az f_0 alapfrekvencia meghatározására az irodalomban több módszer található. Ezek közül az öt legjelentősebbet mutatom be, és hasonlítom össze. Az időtartománybeli módszerek közül a rövid idejű autokorrelációs függvényen (ACF), illetve az átlagos

magnitúdó különbség függvényen (AMDF) alapuló algoritmusokat, a frekvencia-tartománybeli módszerek közül a rövididejű Fourier analízist, a harmonikus egyezés módszerét, és a diszkrét kepsztrum analízist mutatom be.

3.2.2.1. Autokorrelációs függvény (ACF)

Az autokorrelációs függvény kifejezése a következő:

$$R_{xx}(n) = \sum_{i=-\infty}^{\infty} x(i) \cdot x(i+n)$$

Tehát, a függvény n -edik mintájának kiszámításához a jel mintáiból képzett vektort skalárisan össze kell szoroznunk önmaga n mintával való eltoltjából képzett vektorral. Ebből következően az $R_{xx}(0)$ érték a jel energiájával egyezik meg, és ez az érték az autokorrelációs függvény abszolút maximum értéke. Adott n esetén tehát az $R_{xx}(n)$ érték annál nagyobb lesz, minél jobban hasonlít a jel n mintával való eltoltja saját magához. A kváziperiodikus jelek esetén tehát az ACF-nek az alapperiódus egész számú többszöröseinél karakterisztikus lokális maximumai vannak. Alkalmas tehát arra, hogy egy adott hanghoz tartozó jel időtartománybeli alakjából detektálni tudjuk hangmagasságot.

A rövid idejű autokorrelációs függvény a jelnek csak egy véges hosszúságú szegmensére számítja ki a fenti összeget. Tehát egy N hosszúságú jel esetén a rövid idejű autokorrelációs függvénye a következő alakú:

$$R_{xx}(n) = \sum_{i=0}^{N-1} x'(i) \cdot x'(i+n), \quad \text{ahol } \begin{cases} x'(i) = x(i), & \text{ha } i \in [0, N-1] \\ x'(i) = 0, & \text{ha } i \geq N \end{cases}$$

Értelemszerűen, ebben az esetben $R_{xx}(0)$ értéke az adott jelszegmens energiájával egyezik meg, továbbá az adott jelszegmens pszeudo-periódusidejének egész számú többszöröseinél az autokorrelációs függvénynek lokális maximum helyei vannak. Nagy különbség azonban az előbbihez képest, hogy a 0-nál kisebb és, az $(N-1)$ -nél nagyobb indexű minták értéke 0, hiszen az N hosszúságú jelszegmensnek csak $0, 1, \dots, N-1$ indexű elemei vannak. Tehát, a fenti képletben szereplő összeg azon tagjai, amelyekre már $(i+n) \geq N$, nullák lesznek. Így, a rövid idejű autokorrelációs függvény számításakor, minél nagyobb n értéke, annál kevesebb nem nulla tag szerepel az összegben, tehát átlagosan annál kisebb lesz $R_{xx}(n)$ értéke. Másképpen megfogalmazva: A véges jelhosszúság miatt, az ACF szorzódik egy háromszög ablakkal. Ennek eredménye, hogy az autokorrelációs függvényben a

jelszegmens pszeudo-periódusidejének egész számú többszöröseinél lévő lokális maximumok értékei is egyre kisebbek. Ezért, ($R_{xx}(0)$ után) a második legnagyobb értékű lokális maximum helye tartozik a meghatározni kívánt pszeudo-periódusidőhöz.

Sajnos az autokorrelációs függvény számítása az időtartományban nagyon számításigényes. $R_{xx}(0)$ kiszámításához N db. szorzás és $N-1$ db. összeadás elvégzésére van szükség. Az ACF számítása, de a diszkrét Fourier-transzformáció, vagy a konvolúció művelete is nem más, mint MAC műveletek sorozata. A 2.3.3. fejezetben láttuk, hogy, az LPC1768 mikrovezérlő hardveresen ugyan nem támogatja a szorzás-akkumulálás (MAC) művelet elvégzését, de magát a szorzást igen. Így, egy MAC művelet elvégzése két gépi ciklust igényel.

$R_{xx}(0)$ kiszámításához N darab MAC műveletre van szükség. Az n értékének növekedésével egyre csökken azon mintapárok száma, amelyek közül egyik sem nulla, ezért az autokorrelációs függvény n . mintájának kiszámításához $N-n$ darab MAC műveletet kell elvégeznünk. Tehát, az ACF összes (N db.) mintájának számítása során a processzor $\frac{N \cdot (N+1)}{2}$ darab MAC műveletet végez, amelyet összesen $N \cdot (N+1)$ gépi ciklus alatt képes elvégezni a mikrovezérlő.

Az autokorrelációs függvény azonban, megegyezik a jel teljesítménysűrűség-függvényének inverz Fourier-transzformáltjával. Így, az ACF hatékonyabban számolható a frekvencia tartományban, a gyors Fourier-transzformáció (Fast Fourier Transform, FFT) segítségével. Először frekvencia tartományba transzformáljuk a jelet, majd az amplitúdó-spektrumot megszorozzuk komplex konjugáltjával, végül visszatranszformáljuk újra időtartományba, a következő képletek szerint:

$$X_k = FFT(x(n))$$

$$R_{xx}(n) = IFFT(|X_k|^2)$$

A diszkrét Fourier-transzformáció (DFT) N mintából álló bemeneti adatvektor estén N^2 komplex szorzást és $N \cdot (N-1)$ komplex összeadást jelent, amelyet az LPC1768 típusú mikrovezérlő $2 \cdot N^2 + 2 \cdot N \cdot (N-1)$ órajelciklus alatt végez el (lásd: 2.3.3. fejezet). Azonban, ha a bemenő vektor hosszúsága kettőhatvány, akkor az FFT algoritmus nagy gyorsítási lehetőséget jelent. A 2.3.3. fejezetben láttuk, hogy a használt processzor az FFT algoritmust $3 \cdot N \cdot \log_2(N)$ órajelciklus alatt végzi el. Az inverz diszkrét Fourier-transzformáció (IDFT) és az inverz gyors Fourier-transzformáció (IFFT) algoritmusok lépésszáma az előbbiekkal analóg módon alakul.

Egy komplex szám abszolút érték négyzetét úgy kapjuk, ha a valós rész önmagával vett szorzatát hozzáadjuk a képzetes rész önmagával vett szorzatához:

$$|a + j \cdot b|^2 = a^2 + b^2$$

Tehát, kettő szorzás és egy összeadás műveletre van szükség. Mindkét utóbbi művelet elvégzését hardveresen támogatja a mikrovezérlő, ezért egy komplex szám abszolút érték négyzetének számítása 3 órajelciklus ideig tart. Az N komponensből álló spektrum abszolút érték négyzetének kiszámításához ennek megfelelően $3 \cdot N$ gépi ciklusra lenne szükség. Azonban, kihasználhatjuk azt, hogy valós időfüggvényű bemenő jel esetén a diszkrét Fourier-transzformáció eredménye komplex konjugált szimmetrikus, továbbá egy komplex számnak és konjugáltjának abszolút érték négyzete megegyezik. Így, elegendő az első $\frac{N}{2}$ spektrumkomponensre kiszámítani $|X_k|^2$ -et, amely $\frac{3 \cdot N}{2}$ gépi ciklust igényel a mikrovezérlőn. Tehát, az ACF számítása során az FFT elvégzése $3 \cdot N \cdot \log_2(N)$, majd az amplitúdó spektrum megszorítása a komplex konjugáltjával $\frac{3}{2} \cdot N$, majd az IFFT elvégzése további $3 \cdot N \cdot \log_2(N)$ darab gépi ciklust jelent. Összesen $6 \cdot N \cdot \log_2(N) + \frac{3}{2} \cdot N$ órajelciklus idő kell az autokorrelációs függvény ilyen módon történő kiszámításához.

A következő táblázat tartalmazza néhány kettőhatvány értékű vektorhossz esetére, hogy miként alakul az autokorrelációs függvény kiszámításához szükséges gépi ciklusok száma az időtartománybeli, illetve a frekvenciatartománybeli számítási módszer esetén.

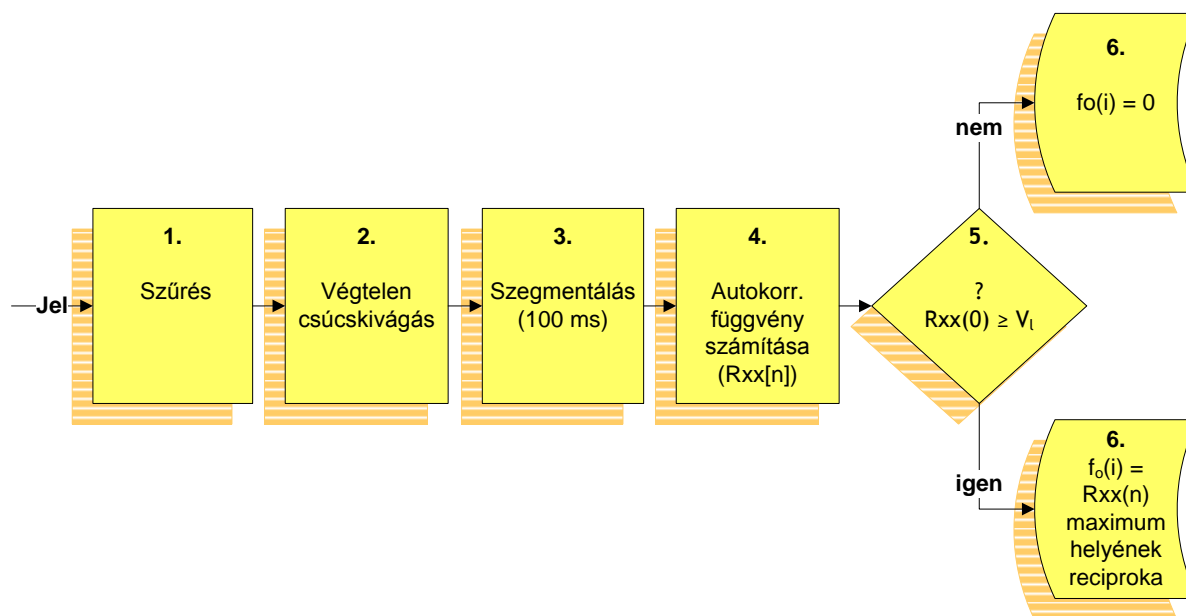
vektorhossz N	időtartomány $N \cdot (N+1)$	frekvenciatartomány $6 \cdot N \cdot \log_2(N) + 3/2 \cdot N$
4	20	54
16	272	408
64	4160	2400
256	65792	12672
512	262656	28416
1024	1049600	62976
2048	4196352	138240

3. táblázat - Az autokorrelációs függvény számításához szükséges lépésszámok

A későbbiekben látni fogjuk, hogy az autokorrelációs függvényt a jelnek közel 100 ms időtartamú szegmenseire számítom ki a hangmagasság detektáló algoritmus során. A 100 ms-nak a 2500 Hz mintavételezési frekvencia mellett, a kettőhatványok közül az 256 vektorhossz felel meg a leginkább. A táblázatban látható, hogy ekkora mintaszámú szegmens rövid idejű autokorrelációs függvényének kiszámítása több mint 5-ször annyi műveletvégzést jelent az

időtartománybeli módszer esetében, mint a frekvenciatartománybeli módszerrel. Ennek megfelelően az ACF-t a frekvenciatartománybeli módszerrel számítom ki a következőkben.

Az autokorrelációs függvényt alkalmazó hangmagasság detektáló algoritmusokról bővebben a [26, 27] irodalmakban olvashatunk. Az algoritmust az én alkalmazásomnak megfelelően optimalizáltam, az általam használt változat lépéseit láthatjuk a következő ábrán.



22. ábra - Az autokorrelációs függvény kiszámításán alapuló algoritmus

A számítás első két lépése a 3.2.1 fejezetben bemutatott alul-, illetve felüláteresztő szűrés, illetve végtelen csúcskivágás. A harmadik lépésben a jelet mozgóablakszerűen, 50%-os átlapolással ≈ 100 ms hosszúságú szegmensekre bontom fel. Ez az idő már elegendően hosszú ahhoz, hogy az már a nagyobb periódusidejű horkoláshangból is tartalmaz minimum 2-3 periódust, ugyanakkor még elegendően rövid ahhoz, hogy ennyi idő alatt a periódusidő állandónak tekinthető. Az autokorrelációs függvény számításánál alkalmazott FFT és IFFT műveletek hatékony elvégezhetősége érdekében egy szegmens pontosan 256 mintavett értékből áll, tehát egy szegmens pontos hossza 102,4 ms.

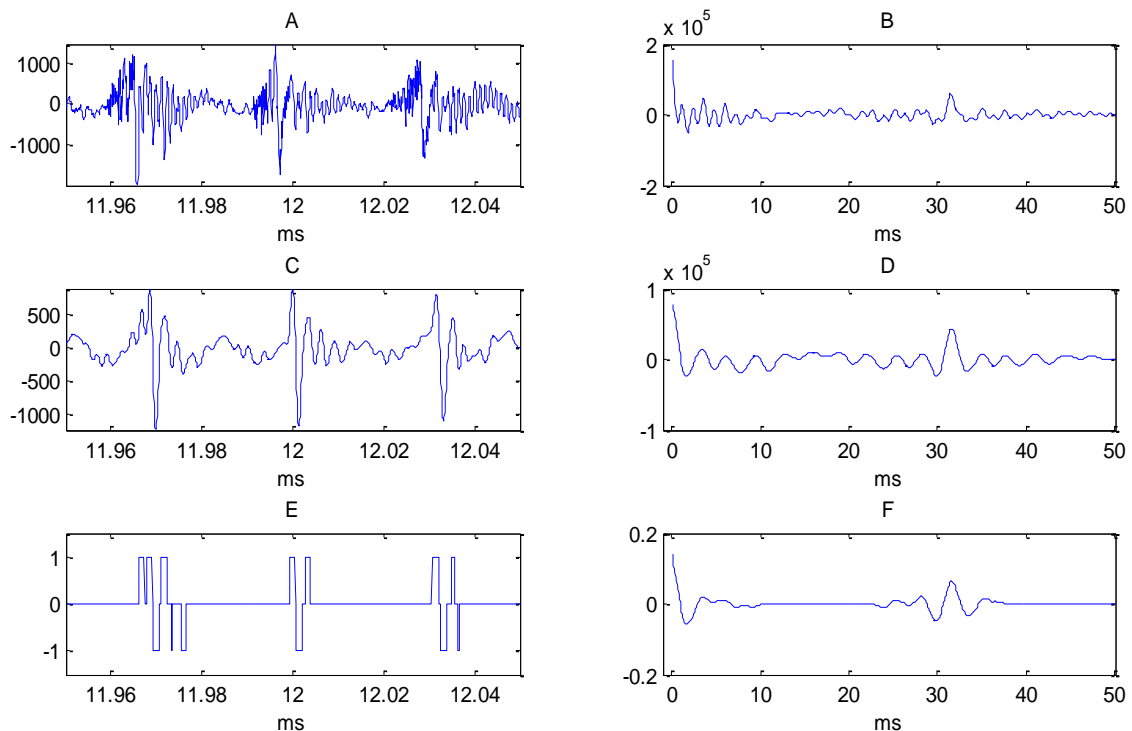
A 4., 5. és 6. lépésekben minden szegmensre külön meghatározom az adott szegmenshez tartozó alaphang frekvenciát. Először (4. lépés) kiszámolom az adott szegmenshez tartozó rövid idejű autokorrelációs függvényt.

A horkoláshang alaphangfrekvenciája 30-60 Hz tartományban helyezkedik el, aminek a 15-35 ms késleltetés felel meg. Tehát, horkoláshoz tartozó hangszegmens esetén biztosan ebbe a tartományba fog esni az autokorrelációs függvény második legnagyobb értékű lokális maximuma. Tekintve, hogy alapvető célunk a horkolás detektálása, felmerülhet a kérdés, hogy miért nem elegendő ebben a tartományban kiszámolnunk az autokorrelációs függvényt? A válasz erre a kérdésre az, hogy az esetlegesen felmerülő alvás közbeni beszéd pszeudo-

periódusideje kisebb ugyan, mint 15 ms, de annak egész számú többszöröseihez tartozó lokális maximum helyek az autokorrelációs függvényben, beleeshetnek ebbe a tartományba. Ezért, ha csak a 15-35 ms késleltetésnek megfelelő helyeken számítanánk ki az autokorrelációs függvényt, és azt néznék, hogy esik-e bele lokális maximum, akkor a beszédhangokat is horkolásnak detektálnánk. Tehát ahhoz, hogy azt tudjuk mondani, hogy az adott hangszegmens horkoláshoz tartozik, meg kell győződnünk róla, hogy az autokorrelációs függvénynek ebbe a tartományba eső lokális maximuma a második legnagyobb értékű lokális maximum.

Az 5. lépésben történik annak eldöntése, hogy az adott szegmens egyáltalán egy hanghoz (horkolás, beszéd) tartozik-e. Ebben a döntésben segítségünkre van $R_{xx}(0)$ értéke, amely a jelszegmens energiájával arányos. Ha, az adott jelszegmens rövid idejű autokorrelációs függvényének nulladik indexű eleme kisebb, mint egy V_l küszöbérték (voice level), akkor a szegmens nem hasznos hanghoz tartozik, a hozzá tartozó alaphfrekvencia érték nulla ($f_o(i) = 0$). Ellenkező esetben, a szegmens valamilyen hanghoz tartozik, és az autokorrelációs függvény második legnagyobb értékű lokális maximumának helye az adott szegmens pseudo-periódusideje. Tehát az alaphfrekvencia értéke annak reciproka.

A következő ábrán bemutatom, hogy miként hat a szűrés, illetve a végtelen csúcskivágás egy adott hangszegmens időfüggvényére, illetve az autokorrelációs függvényére.



23. ábra - A szűrés és a végtelen csúcskivágás hatása az ACF-re

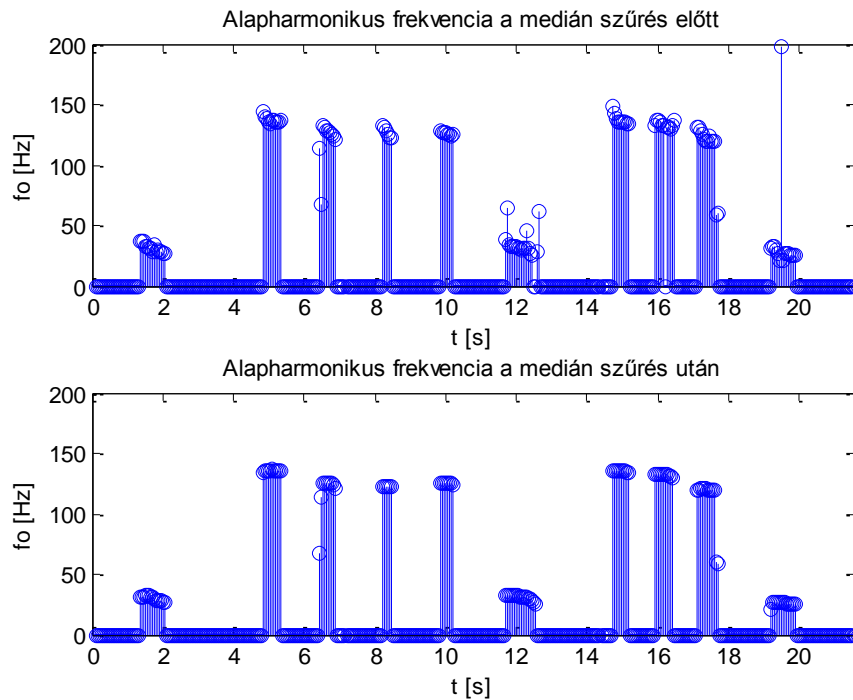
Az A , illetve B jelű grafikonon egy előfeldolgozás nélküli hang, egy 100 ms-os szegmensének időfüggvényét és autokorrelációs függvényét láthatjuk. A hangszegmens egy horkoláshanghoz tartozik. Látható, hogy ennek megfelelően az autokorrelációs függvényében a második legnagyobb értékű lokális maximum helye kb. 32 ms késleltetésnél van.

A C , illetve D jelű grafikonokon azt ábrázoltam, hogy miként alakul ugyanezen hangszegmens idő- és autokorrelációs függvénye a szűrés után. Megfigyelhetjük, hogy mindkét függvény kevésbé zajos, illetve az autokorrelációs függvény második legnagyobb maximumának helye nem változott.

Az E , illetve F jelű grafikonon a végtelen csúcskivágás hatását láthatjuk. Az időfüggvény (E) a végtelen csúcskivágásnak megfelelően csak -1 , 0 és 1 értékeket vesz fel. Az autokorrelációs függvény "kisimult" az előző állapotokhoz képest, mindemellett második legnagyobb maximumának helye továbbra sem változott. Tehát az előfeldolgozás után a számunkra érdekes információ továbbra is benne van a jelben.

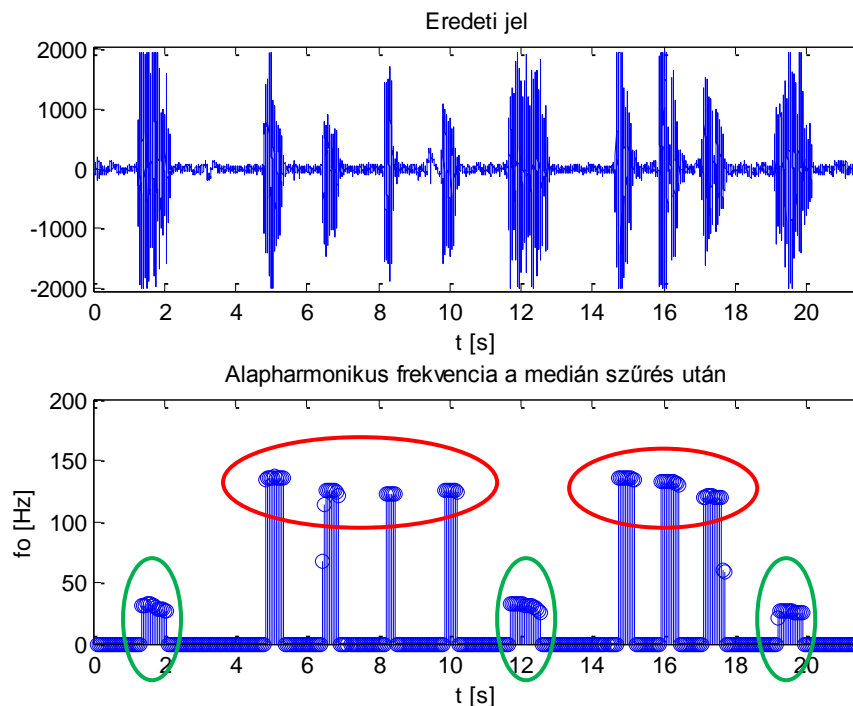
A bemutatott algoritmus, tehát minden ~ 100 ms hosszúságú szegmens esetén meghatározza az adott szegmenshez tartozó hang hangmagasságát. A szegmenseket 50%-os átlapolással vettem fel, ezért 50 ms időközönként kapunk egy új becsült alapfrekvencia értéket. Azonban, e hangmagasság detektálás még tartalmazhat hibákat. Például, előfordulhat téves hasznos/nem hasznos hangszegmens döntés, illetve előfordulhat olyan eset is, hogy a rövid idejű autokorrelációs függvény második legnagyobb maximum helyének pontos meghatározása nem lehetséges. Továbbá, az ACF számításakor akár a felharmonikusok, akár az f_0 felénél levő frekvencia komponens szerepe is megnőhet, túlsúlyozódhat, ezzel hamis eredményt adva. Érdeemes tehát, az algoritmus eredményeként kapott alapfrekvencia sorozaton egy olyan utófeldolgozást elvégezni, amely a ritkán előforduló kiugró, illetve nem megfelelő értékeket "kisimítja". A nemlineáris simítási technikák közül a medián szűrést alkalmaztam, amely során a jelet környezetének a mediánjával, azaz a nagyság szerinti középső elemmel helyettesítjük. A mediániszűrt jel n -edik mintájának értékét az eredeti jel $(n - L)$ -től $(n + L)$ -ig terjedő indexű mintái határozzák meg. Ezen $2L + 1$ db. minta mediánja lesz a szűrt jel n -edik mintája. Az algoritmus MATLAB környezetben történő tesztelése során az adódott, hogy a kívánt hatás eléréséhez L optimális értéke 4.

A 24. ábrán egy 22 másodperces hangfelvételen a lefuttatott algoritmus eredményét láthatjuk a medián szűrés előtt és után. Megfigyelhető, hogy a szűrés után eltűntek a környezetükből kiugró értékek az alapfrekvencia sorozatból, és a jel jól elkülöníthető részekre vált szét.



24. ábra - A medián szűrés hatása

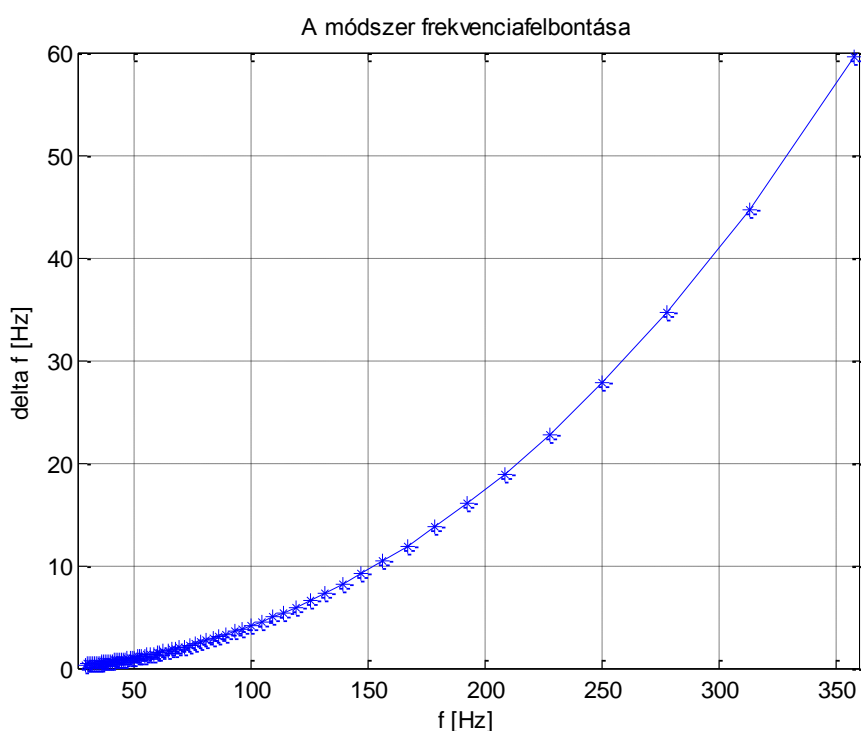
A 25. ábrán az eredeti jel időfüggvénye, és a rajta lefuttatott hangmagasság detektáló algoritmus eredménye, azaz a számított alaphfrekvencia sorozat látható. A tesztelésre használt hangfelvétel elején, közepén és végén egy-egy horkoláshang, közöttük pedig beszédhangok szerepelnek.



25. ábra - A hangmagasság detektálás eredménye az ACF kiszámításán alapuló módszer esetében

Az ábrán zöld színnel bekarikáztam a horkoláshangokhoz tartozó, piros színnel a beszédhangokhoz tartozó számított alapfrekvencia értékeket. Megfigyelhető, hogy az előbbiek hangmagassága 30 Hz körül, az utóbbiaké 120-130 Hz körül helyezkednek el. Tehát, csak az algoritmus eredményét ismerve is megállapítható lenne, hogy mikor horkolt a megfigyelt beteg.

Érdekes kiszámolnunk, hogy a számunkra érdekes frekvenciatartományban (30-400 Hz) mekkora a módszer frekvenciafelbontása. Az alapfrekvenciát a mért periódusidőből számítjuk reciprok képzéssel. A periódusidőt a mintavételi frekvencia (2500 Hz) által meghatározott 0.4 ms-os felbontással tudjuk meghatározni, hiszen az autokorrelációs függvény második legnagyobb értékű lokális maximuma gyakorlatilag bármelyik mintánál lehet. A 30 Hz - 400 Hz frekvenciatartománynak a 2.5 ms - 34 ms periódusidő tartomány felel meg, amelyhez az autokorrelációs függvénynek a 6. - 85. mintái tartoznak. A frekvenciafelbontást úgy kapjuk, hogy ezen tartományban lévő szomszédos mintákhoz tartozó frekvencia értékek különbségét vesszük. Ezt láthatjuk a következő ábrán, amelyen a 30-400 Hz tartományban detektálható frekvencia értékek függvényében ábrázoltam a hozzájuk tartozó megfelelő frekvenciafelbontás értékeket.



26. ábra - Az ACF kiszámításán alapuló módszer frekvenciafelbontása

Az ábrán csillaggal jelöltem azokat a frekvencia értékeket, amelyeket ezzel a módszerrel pontosan detektálni tudunk. Megfigyelhető, hogy a horkolás tartományában (30-

60 Hz) a frekvenciafelbontás értéke megfelelően nagy, ez esetben az ACF kiszámításán alapuló módszer 1 Hz-nél kisebb hibával szolgáltatja a hangmagasság értékét. Sőt, a 150 Hz-nél kisebb alapfrekvenciájú beszédhangok esetén (a legtöbb beszédhang ilyen) a felbontás értéke kisebb, mint 10 Hz.

A módszer előnye, hogy nemcsak a horkolás tényét detektálja, hanem a horkoláshang, sőt a beszédhang hangmagassága is kiadódik, ezért alkalmas lehet a módszer további horkolás, illetve beszédelemzésekre is. Az ACF-t az időtartományban számolva nagyon nagy lenne a számításigénye a módszernek. Azonban a frekvenciatartománybeli kiszámítási technikával, az FFT, illetve IFFT műveletek hatékonyságát kihasználva a számításigény nagymértékben lecsökkent. Sőt, később látni fogjuk, hogy a nagyobb pontszámú FFT-t igénylő, spektrum elemzésén alapuló módszereknél kisebb a számításigénye. Nagy előnye a frekvenciatartománybeli technikákhoz képest, hogy a számunkra érdekes kismagasságú tartományban (30-400 Hz) nagyon pontos a felbontása.

3.2.2.2. Átlagos magnitúdó különbség függvény (AMDF)

Átlagos magnitúdó különbség függvény:

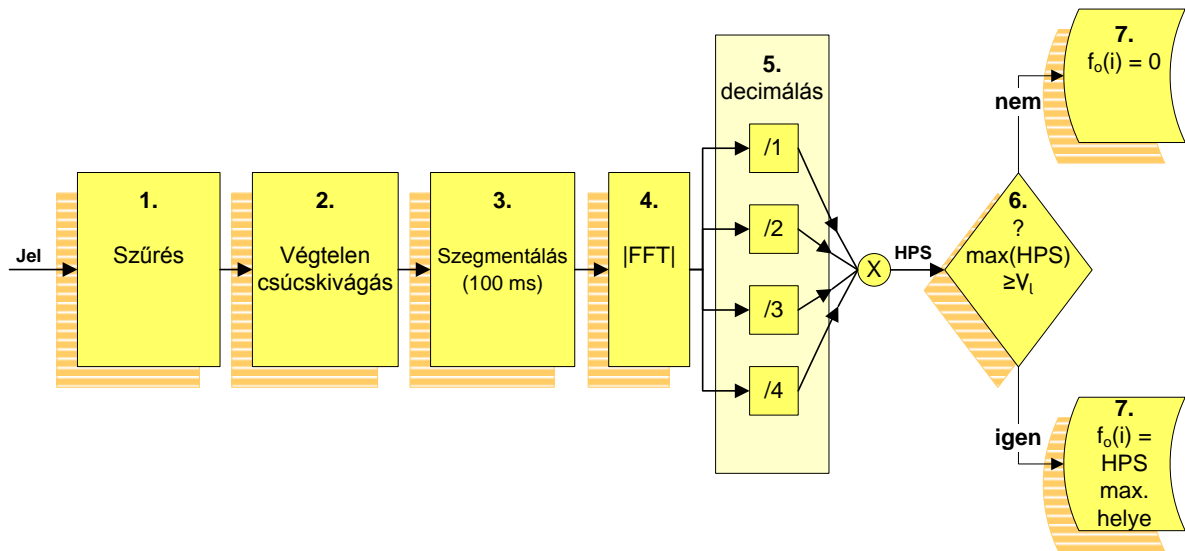
$$AMDF_n(j) = \frac{1}{N} \sum_{i=1}^N |x_n(i) - x_n(i+j)|, 1 \leq j \leq MAXLAG$$

Kváziperiodikus jelek esetén a függvénynek az alapperiódus egész számú többszöröseinél karakterisztikus lokális minimumai vannak. Természetesen, az AMDF globális minimuma a nulladik indexű tagjánál van, hiszen ez felel meg a nulla késleltetésnek, amikor a jelet önmagából vonjuk ki. Az AMDF előnye, az autokorrelációs függvényhez képest, hogy csak összeadás műveletet kell végeznünk (nem kell szorozni), hátránya hogy jóval nagyobb a művelet igénye. Ugyanis, ellentétben az autokorrelációs függvénnyel, az AMDF számítását mindenképpen időtartományban kell végeznünk.

A horkolás detektálását végző algoritmus megegyezik az előző fejezetben bemutatottal, azzal a különbséggel, hogy most az adott hangszegmens pseudo-periódusidejét az AMDF segítségével határozzuk meg. MATLAB környezetben ezt az algoritmust is megvalósítottam, és megállapítottam, hogy alkalmas a hangmagasság detektálására, azonban nagy számításigénye miatt valós idejű alkalmazásokban nem célszerű a használata.

3.2.2.3. Harmonikus Egyezés Módszere

M.R. Schroeder tanulmányában [29] egy olyan módszert mutat be, amely a hangmagasságot a frekvenciatartománybeli jelalak alapján határozza meg. Sajnos ez nem egyértelmű feladat, hiszen az alapharmonikus általában nem a legnagyobb amplitúdójú komponens a spektrumban. Azonban, az alapharmonikus frekvencia egész számú többszöröseinél megjelennek a felharmonikusok. A módszer ezen utóbbi állításon alapul. Lényege a következő: Ha, összenyomjuk a jel spektrumát a felére, harmadára, negyedére, stb., akkor az alapharmonikus és a hozzá tartozó felharmonikusok pont ugyan arra a helyre kerülnek a spektrumban. Az összenyomás az eredeti spektrum megfelelő mértékű alul-mintavételezésével történhet. Ezeket összeszorozva egymással és az eredeti spektrummal, kapjuk az úgynevezett harmonikus szorzat spektrumot, amelyet ezentúl HPS-sel rövidítünk (Harmonic Product Spectrum). A maximuma pont az alapharmonikus frekvenciánál lesz, hiszen az egyes összenyomott spektrumokban pont az alapharmonikus helyére kerülnek a megfelelő felharmonikusok. A hangmagasság detektálásához tehát, csupán a HPS maximum helyét kell megkeresnünk. Ennek megfelelően, a Harmonikus Egyezés Módszere az alapharmonikus frekvencia detektálására a következő lépésekből áll:



27. ábra - A harmonikus egyezés módszere

Az első három lépés megegyezik az első módszernél bemutatott lépésekkel, amelyekben a jel előfeldolgozása, illetve szegmentálása történik. A szegmensek ebben az esetben is ~100 ms (256 db. minta) hosszúságúak, és 50%-os átlapolódással vettem fel őket.

A *negyedik lépésben* kiszámolom az egyes hangszegmensek spektrumát a gyors Fourier-transzformáció segítségével. Tekintve, hogy a mintavételi frekvencia 2500 Hz, egy 100 ms hosszúságú szegmens 256 mintából áll. Azonban a spektrum számításánál lehetőségünk van ennél nagyon pontszámú FFT-t alkalmaznunk. Ez a gyakorlatban úgy valósul meg, hogy az adott szegmenst az időtartományban reprezentáló 256 minta mögé annyi darab nullát szúrunk be, amennyivel nagyobb az FFT pontszáma, mint 256, majd az FFT-t az így keletkezett mintasorozatra végzem el. Ennek segítségével a szegmensek spektrumát sűrűbben számíthatjuk ki. A Harmonikus Egyezés Módszere során a spektrum negyed rendű alulmintavételezésére is sor kerül, ezért a megfelelő pontosságú frekvenciafelbontás eléréséhez viszonylag nagy, legalább 512 pontszámú FFT-t kell alkalmaznunk. A kettőhatvány pontszámra, az FFT algoritmus gyorsítási lehetőségének kihasználásához van szükség. Az 512 pontszám $\Delta f = \frac{F_s}{N} = \frac{2500}{512} \approx 4.88 \text{ Hz}$, a 256 pontszám $\Delta f = \frac{F_s}{N} = \frac{2500}{256} \approx 9,77 \text{ Hz}$ frekvenciafelbontást jelent az eredeti spektrumban. Minél nagyobb az FFT pontszáma, annál jobb a frekvenciafelbontás, de annál nagyobb a számításigény is. Láttuk, hogy az alapharmonikus frekvencia a horkolóshang esetén a 30 - 60 Hz, a beszédhang esetén a 100 - 400 Hz tartományban helyezkedik el. Ezért, ha a célunk csupán a horkolóshang detektálása, akkor elegendő számunkra ez az utóbbi kb. 10 Hz-es frekvenciafelbontás is. Ha azonban egyéb elemzéseket is szeretnénk végezni a hangokon, akkor nagyobb pontszámú, és ezért nagyobb számításigényű FFT-t kell alkalmaznunk.

Az *ötödik lépésben* az alulmintavételezéseket gyakorlatilag az eredeti spektrum decimálásával valósítottam meg. Tehát, a felére, harmadára és negyedére összenyomott spektrumot úgy kapom, hogy az eredeti spektrumban csak minden második, harmadik, illetve negyedik spektrum komponenst veszek figyelembe. Ezen spektrumokat összeszorozva kapom a harmonikus szorzat spektrumot (HPS).

A *hatodik lépés* annak eldöntése, hogy az adott szegmens valamilyen hanghoz tartozik-e. A döntést a HPS maximális értéke alapján hozza meg az algoritmus, ha ugyanis ez egy beállított küszöbértéknél nagyobb, akkor az adott szegmens egy hanghoz tartozik, amelynek alapfrekvenciája a HPS maximum helye, ellenkező esetben a szegmens nem hanghoz tartozik, így a hozzá tartozó alaphang frekvencia érték nulla.

Az előző módszerekhez hasonlóan tehát, az algoritmus szolgáltat egy alaphang frekvencia sorozatot, amelynek minden eleme az előtte lévő 100 ms hosszúságú hangszegmens detektált hangmagsága. Ezen a sorozaton a kiugró hibás értékek megszüntetése érdekében ebben az esetben is egy medián szűrést végzek el az utófeldolgozás keretében.

Vizsgáljuk meg az algoritmust, a processzorral szemben támasztott számítási igény szempontjából. Az N pontszámú FFT algoritmus a teljes spektrumot kiszámítja $\frac{Fs}{N}$ frekvenciaközönként, és a 2.3.3. fejezetben láttuk, hogy mindez $3 \cdot N \cdot \log_2(N)$ órajelciklust igényel az alkalmazott mikrovezérlőben. Az algoritmus során a spektrum abszolút értékét kell kiszámolnunk. Egy komplex szám algebrai alakjából (valós és képzetes rész) az abszolút értékét a következő képpen kapjuk:

$$|a + j \cdot b| = \sqrt{a^2 + b^2}$$

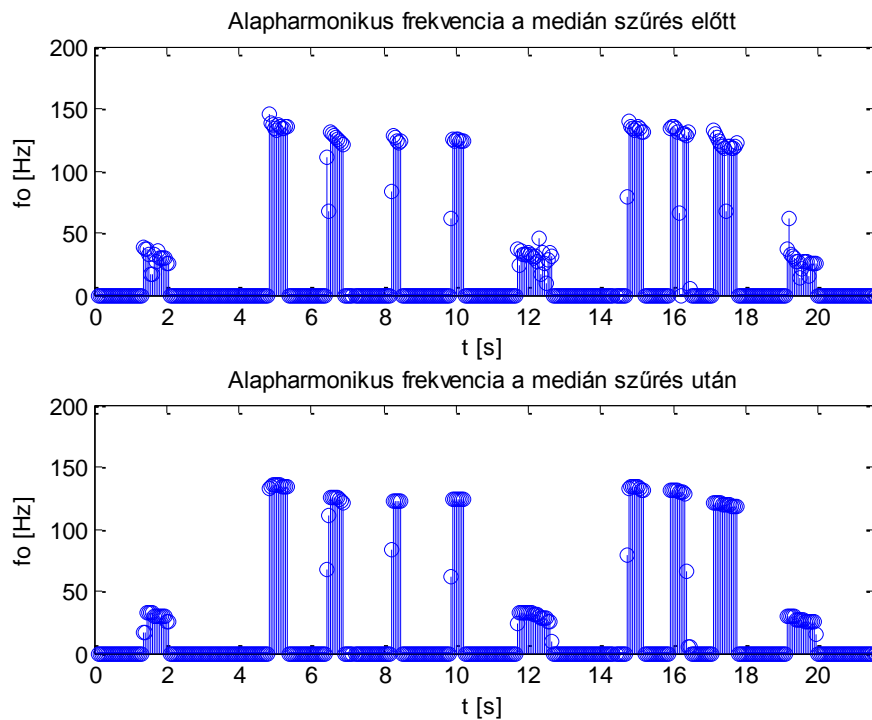
Egy fixpontos processzoron a gyökvonás megvalósítása azonban igen körülményes, és pontatlan, ezért célszerű a spektrum abszolút értéke helyett az abszolút érték négyzetével számolni. Ezt megtehetjük, mert a harmonikus szorzat spektrumon maximumhely keresést fogunk végezni, és a négyzetre emelés művelete monoton. Így a maximumhely szempontjából irreleváns, hogy HPS-t a spektrum abszolút értékéből, vagy annak négyzetéből számítjuk. A 3.2.2.1. fejezetben láttuk, hogy az FFT eredményéből a spektrum abszolút érték négyzetének kiszámításához $\frac{3 \cdot N}{2}$ gépi ciklus elegendő.

A szegmensek időfüggvénye valós, ezért gyors Fourier-transzformáció eredménye komplex konjugált szimmetrikus lesz. Tehát, számunkra érdekes információt csak az első $\frac{N}{2}$ spektrumkomponens hordoz, amelyek a $\left[0, \frac{Fs}{2}\right]$ frekvenciatartományt fedik le. A spektrum alul-mintavételezésekor is csak ezen komponensekkel kell számolnunk. Ebből következően a felére összenyomott spektrum $\frac{N}{2 \cdot 2} = \frac{N}{4}$, a harmadára összenyomott spektrum $\frac{N}{2 \cdot 3} = \frac{N}{6}$, míg a negyedére összenyomott spektrum $\frac{N}{2 \cdot 4} = \frac{N}{8}$ db. nullától különböző spektrumkomponenst tartalmaz. Ezen négy spektrum összeszorzásánál tehát, mindegyik spektrumnak csak az első $\frac{N}{8}$ darab komponense számít a szorzat eredménye szempontjából, hiszen a többi komponensük a negyedére összenyomott spektrummal való szorzás miatt biztosan nullával is szorzódik. A spektrumok összeszorzásánál tehát, elegendő csak az első $\frac{N}{8}$ darab spektrumkomponenseket szorozni egymással.

A négy spektrum k -adik komponensének összeszorzása 3 komplex szorzást igényel, így összesen $3 \cdot \frac{N}{8}$ darab szorzás műveletet kell elvégeznünk, amely pontosan ennyi további gépi ciklust jelent az alkalmazott mikrovezérlőnek. Tehát, a harmonikus szorzat spektrum számítása összesen $3 \cdot N \cdot \log_2(N) + \frac{3}{2} \cdot N + \frac{3}{8} \cdot N$ órajelciklust igényel, amely $N=256$ pontos

FFT esetén 6624, míg $N=512$ pontos FFT esetén 14784 gépi ciklust jelent. Ezen utóbbi már nagyobb, mint az ACF kiszámításán alapuló módszer műveletigénye. (3.2.2.1 fejezet)

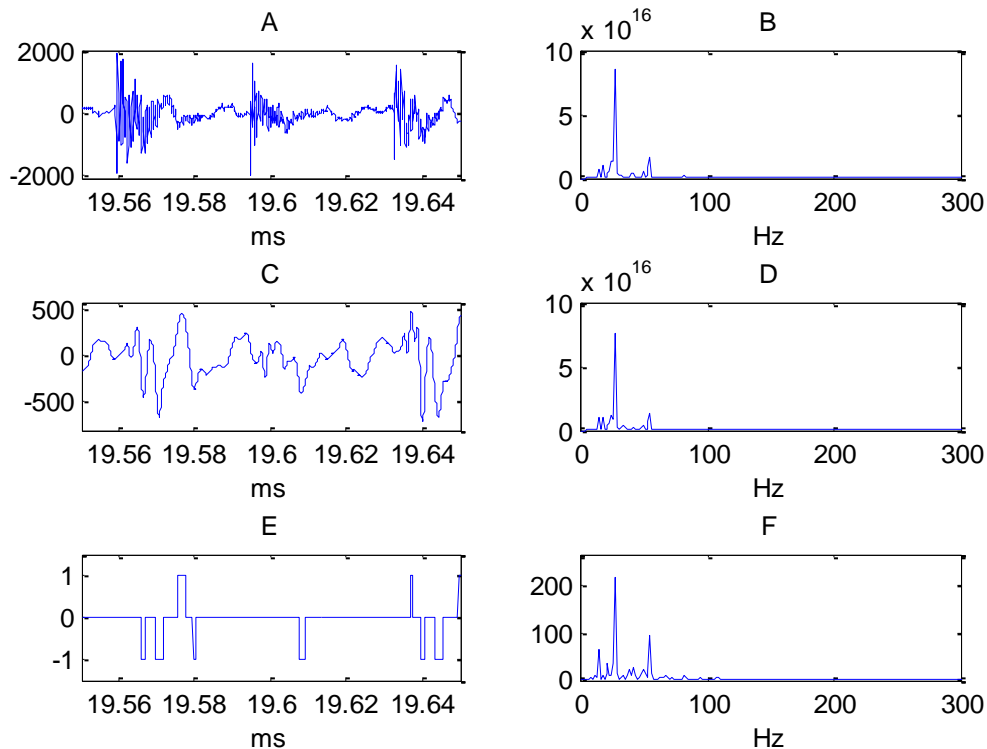
A következőkben bemutatásra kerülő eredmények számítása során 512 pontos FFT-t használtam. Az egyes algoritmusok tesztelésére használt hangfelvételen lefuttatott, Harmonikus Egyezés Módszerét alkalmazó algoritmus eredményét láthatjuk a következő ábrán az utófeldolgozás (medián szűrés) előtt és után.



28. ábra - A Harmonikus Egyezés Módszerén alapuló algoritmus eredménye

Megállapíthatjuk, hogy az eredmények nagyon hasonlóak az előző módszerekéhez, tehát ez a módszer is alkalmas a hangmagasság meghatározására, így a horkolás detektálására is. Az eredmények alapján egyértelműen meghatározható, hogy a tesztelésre használt hangminta mely részei tartoznak horkoláshoz (30-60 Hz), és melyek beszédhez (100-400 Hz).

Érdeemes megvizsgálni most is, hogy az előfeldolgozás lépései (aluláteresztő szűrés és végtelen csúcskivágás) mennyiben befolyásolják egy szegmenshez tartozó HPS-t. A következő ábrán egy horkoláshanghoz tartozó 100 ms-os szegmens eredeti (A), szűrt (B) és végtelen csúcskivágott (E) változatait és a belőlük számolt harmonikus szorzat spektrumokat (B, D és F) láthatjuk.



29. ábra - A szűrés és a végtelen csúcskivágás hatása a HPS-re

Megfigyelhető, hogy a HPS maximum helye ($\approx 35 \text{ Hz}$) nem változott meg az előfeldolgozás hatására. A szűrés és a végtelen csúcskivágás egyszerűsíti a számításokat, de a jel számunkra érdekes információtartalmát nem változtatja meg.

Összefoglalásul elmondható, hogy a Harmonikus Egyezés Módszere alkalmas a hangmagasság meghatározására. Előnye, hogy kisebb FFT pontszámok esetén a számításigénye kisebb, mint az előzőleg bemutatott technikák esetén láttuk. Hátránya, hogy a megfelelő pontosságú frekvenciafelbontáshoz nagy pontszámú FFT-t kell alkalmaznunk, amely esetén a műveletigény már nagyobb, mint az ACF kiszámításán alapuló módszeré. Ha, célunk csupán a horkolás detektálása, akkor a Harmonikus Egyezés Módszere $N=256$ pontszámú FFT használatával a viszonylag kevés műveletvégzéssel célhoz vezet. Láttuk, hogy ekkor azonban a frekvencia felbontása $\approx 10 \text{ Hz}$, amely egyéb elemzésekre már nem alkalmas. Ha, a horkolás detektálásán kívül egyéb elemzéseket is szeretnénk végezni, akkor célszerűbb az autokorrelációs függvény kiszámításán alapuló módszert alkalmaznunk. Ezen utóbbi módszer műveletigénye még megfelelően alacsony, ugyanakkor nagyon pontos, különösen a számunkra érdekes tartományban (30-400 Hz).

3.2.2.4. Diszkrét kepsztrum analízis

Az $x(t)$ jel kepsztruma:

$$c(q) = \mathcal{F}^{-1}\{ \log(| \mathcal{F}\{ a(t) \} |^2) \}$$

Ez egy idő argumentumú függvény, de az argumentumot idő helyett kefrenciának nevezik. Kváziperiodikus jelszakaszok széleinek - felfutások, lecsengések, átmenetek - elemzésére szolgál: az alapperiódussal megegyező kefrencia helyen impulzusszerű kiugrás jelenik meg. Ezt az eljárást főleg beszédfelismerésnél használják. A kepsztrum a spektrum spektrumaként értelmezhető, ahol az x tengelyen az időegység (a kefrencia) helyezkedik el, a hozzájuk tartozó értékek a spektrum változását jelentik. Eredetileg a földrengések, illetve bombarobbanások okozta szeizmikus rezgések elemzésére találták ki a módszert. Később az emberi beszédfelismerés, és a visszatérő radarjelek elemzésének egyik legfőbb eszköze lett a kepsztrum analízis [30].

Az emberi hangképzés során a hangszálakban keletkező hang a gégefő feletti üregeken (garat, orrüreg és szájüreg) átjutva módosul ezen üregek, mint rezonátor rendszerek átviteli függvénye szerint. Ha $x(t)$ a hangszálakban keletkező hang, $y(t)$ pedig az egyéb hangképző szerveink (vokális traktus) együttes súlyfüggvénye, akkor a képletben szereplő $a(t)$ hang ezek konvolúciójaként áll elő.

$$a(t) = x(t) * y(t)$$

A Fourier-transzformáció a konvolúciót szorzásba viszi, így $a(t)$ spektrumának abszolút érték négyzete a következő képpen írható fel:

$$| \mathcal{F}\{ a(t) \} |^2 = |A(f)|^2 = |X(f)|^2 \cdot |Y(f)|^2$$

A logaritmus képzés nagy előnye, hogy a szorzást összeadásba viszi át, így:

$$\log|A(f)|^2 = \log|X(f)|^2 + \log|Y(f)|^2$$

A hangszálak által keltett hang és a vokális traktus energia spektruma a logaritmikus skálán összeadódik, ezáltal sok esetben könnyen szétválaszthatóak. A beszédfelismerésben a kepsztrum analízis nagy előnye az, hogy segítségével képesek vagyunk különválasztani a vokális traktus és a beszéd súlyfüggvényét, ezáltal a beszéd főbb jellemzőit is különválaszthatjuk.

A horkolás detektálás és elemzés területén a módszer egy megvalósításáról részletes leírást adtam az *1.2.4. fejezetben* az ott bemutatott [13] irodalom kapcsán.

A kepsztrum abban különbözik az autokorrelációs függvényétől, hogy kiszámításakor az inverz diszkrét Fourier-transzformáció elvégzése előtt a spektrum abszolút érték négyzetének

vesszük a természetes alapú logaritmusát is. A logaritmusképzés azonban az alkalmazott processzoron csak a memóriában tárolt logaritmus táblázatból való visszakereséssel végezhető el. Ez jóval több számítási időt, illetve memóriaigényt jelent az ACF számításához képest. Ezen kívül viszont, a hangmagasság detektálása ugyanazon az elven történik, amit a 3.2.2.1. fejezetben az autokorrelációs függvény kiszámításán alapuló módszer esetében bemutatam. Ezen utóbbi módszer a kisebb számításigénye miatt a beágyazott, kevés erőforrással rendelkező alkalmazásomban szükséges valós idejű megvalósítás szempontjából előnyösebb, mint a diszkrét kepsztrum analízis.

3.2.2.5. Az alapharmonikus detektálási módszerek összefoglalása

Az előző fejezetekben láttuk, hogy négy különböző algoritmust vizsgáltam meg MATLAB környezetben, amelyek mindegyike alkalmasnak bizonyult az alapharmonikus frekvencia detektálására. Az alkalmazásomban az eszköznek valós időben kell elvégeznie a horkolás detektálást, ezért számomra az egyik legfontosabb követelmény az algoritmusokkal szemben az alacsony számításigény.

Láttuk, hogy a legalacsonyabb számításigénye a Harmonikus Egyezés Módszerének van, abban az esetben, ha szegmensenként 256 pontszámú FFT-t alkalmazunk. Hátrányként azonban felmerül, hogy ebben az esetben a hangmagasság detektálás frekvenciafelbontása ≈ 10 Hz, amely a horkolás detektálására elegendő, de további elemzésekre nem alkalmas.

Ha, célunk a horkolás detektálásán kívül az is, hogy egyéb elemzéseket is végezhesünk, akkor célszerűbb az autokorrelációs függvény kiszámításán alapuló módszert alkalmaznunk. Ennek műveletigénye még megfelelően alacsony a valós idejű megvalósításhoz az alkalmazott mikrovezérlőben, ugyanakkor nagy felbontású, különösen a számunkra érdekes tartományban (30-400 Hz).

4. fejezet

Az autokorrelációs függvény kiszámításán alapuló algoritmus a beágyazott környezetben

4.1. A beágyazott szoftver feladata

A 2. fejezetben leírtaknak megfelelően, a horkolás detektálására kifejlesztett eszköz központi egysége a 2.3.2. fejezetben részletesen bemutatott LPC1768 típusú nagyteljesítményű mikrovezérlő. Pontosabban ezen mikrovezérlő köré épített *mbed NXP LPC1768* nevű fejlesztőkártya [16]. A gyártó (NXP) ezen gyors prototípus készítésre optimalizált kártyához széleskörű támogatást nyújt. Ezek egyike egy internetes fejlesztőkörnyezet [31], amely online lehetővé teszi C++ nyelven a programok írását, fordítását és letöltését az *mbed NXP LPC1768* típusú mikrokontrollerre. Ezen kívül tartalmaz egy *mbed* könyvtárat, amelyben a mikrovezérlő összes perifériájához található API (*Application Programming Interface*), amelyek az előre megírt függvények segítségével az alkalmazások fejlesztését nagyban segítik. Ezen internetes fejlesztőkörnyezet az ARM RealView fordítóját használja, amelynek köszönhetően a C++ nyelven írt programból hatékony gépi kódot generál. Mindezen előnyök tudatában döntöttem úgy, hogy a beágyazott szoftver fejlesztését ezen internetes fejlesztőkörnyezetben végzem.

Célom tehát a horkolás detektálása, és ezt az előző fejezetben leírtak szerint az autokorrelációs függvény kiszámításán alapuló algoritmus implementálásával érem el. Ennek megfelelően a beágyazott szoftver feladata a következő:

Elvégzi a mikrofon analóg jelének 2500 Hz-es mintavételi frekvenciával való mintavételezését és analóg-digitális átalakítását. Az így kapott mintasorozatra elvégzi a 22. ábrán látható, az autokorrelációs függvény kiszámításán alapuló algoritmust, majd az azt követő utófeldolgozást (medián szűrést). A megvalósításban az esetlegesen bekövetkező téves

horkoláshang detektálás kiküszöbölésére kihasználom a horkolás periodikusságát. Ehhez egy horkoláshang bekövetkezése után az egység megvárja, hogy a következő 2-5 másodpercben jön-e ismét horkoláshang. Majd ezt megismétli még egyszer, és ilyen módon periodikusan 3-szor egymás után detektált horkoláshang után dönt úgy az egység, hogy horkolás történik.

Az eszközt úgy terveztem meg, hogy közvetlen vezetékes kapcsolatra, vezeték nélküli zigbee kapcsolatra és vezeték nélküli flash tárolós mérésekre is fel legyen készülőve. Ezért a horkolás tényét képes elküldeni a zigbee hálózaton keresztül, vagy soros porton a PC-nek, vagy naplózni egy fájlba a microSD kártyán.

4.2. A perifériák konfigurációja

Az LPC1768 mikrokontroller 4 db általános célú, 32 bites **időzítő/számláló egységgel** rendelkezik, amelyek közül kettőt használok fel.

Az egyik (**Timer0**) felel a mintavételi frekvencia ütemezésért. A 2500 Hz frekvenciának 400 μ s mintavételi időköz felel meg. Ennek eléréséhez a mikrovezérlő APB (*Advanced Peripheral Bus*) órajeléből ($f_{CPU} = 96 \text{ MHz}$) a Timer órajelét egy 96-os előosztással ($d_{presc.} = 96$) állítom elő (1 MHz). A megszakítást egy úgynevezett *Output Compare Match* esemény hatására következik be, vagyis amikor a számláló értéke megegyezik a Match Control Register (MCR) beprogramozott értékével, akkor generálódik egy megszakítás. A számláló Match Control Register-ét (MCR0) 399-re állítva ($d_{MCR} = 400$), és a számlálót CTC (Clear Timer on Compare match) üzemmódban üzemeltetve az ütemezést jelentő megszakítási szubrutin érvényre jutásának frekvenciája a következő:

$$f_{ISR} = \frac{f_{CPU}}{d_{presc.} \cdot d_{MCR}} = \frac{96 \text{ MHz}}{96 \cdot 400} = 2500 \text{ Hz}$$

Minden compare match esemény megszakítást generál, amelynek hatására történik a mintavételezés.

Egy másik időzítő/számláló egység (**Timer2**) szolgál egy horkoláshang bekövetkezése után a következő horkoláshang bekövetkezéséig eltelt idő másodperc pontosságú mérésére. Ehhez a Timer órajelét a CPU órajeléből a 32 bites előosztó segítségével 96000000 értékű előosztással 1 Hz-re állítom. A beágyazott szoftver futása során egy horkoláshang beérkezése után elindítja a Timert, majd segítségével ellenőrzi, hogy a következő 2-6 másodpercen belül ismét történik-e horkoláshang detektálása.

Az LPC1768 mikrovezérlő egy beépített, 8 csatornás, 12 bites analóg-digitális konverterrel (**A/D átalakító**) rendelkezik. A fejlesztő kártyán az analóg-digitális konverter referencia feszültsége a 3,3 V-os digitális tápfeszültség megfelelő LC taggal ($C=1\mu\text{F}$, $L=100\mu\text{H}$) való szűrésével áll elő, tehát a teljes A/D átalakítási tartomány 3,3 V. Az A/D átalakító órajelét a mikrovezérlő APB (*Advanced Peripheral Bus*) buszának órajelének a leosztásával állíthatjuk elő. Ezt az alkalmazásban az adatlap [20] szerinti maximális értékre ($f_{ADC} = 13 \text{ MHz}$) állítottam. Egy konverzió 65 órajel ciklus ideig tart, tehát:

$$T_{conv.} = N \cdot \frac{1}{f_{ADC}} = 65 \cdot \frac{1}{13 \text{ MHz}} = 5 \mu\text{s}$$

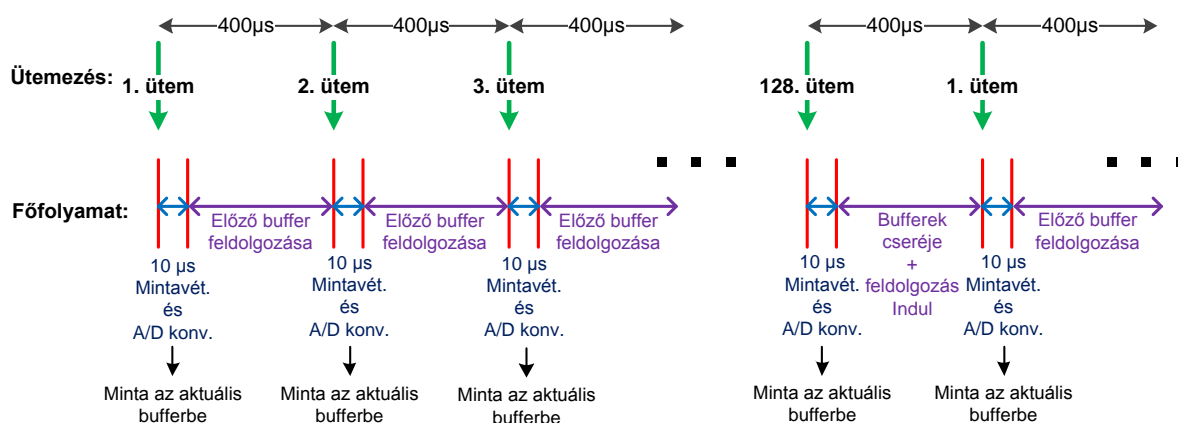
Az eszköz egyetlen mikrofont tartalmaz, ezért az AD átalakítónak egyetlen csatornáját kell csak alkalmaznunk, így a csatornaváltással és azzal járó csatornaváltási idő kiváráásával nem kell foglalkoznunk. Az általam fejlesztett beágyazott szoftverben a mintavételi frekvencia ütemét jelentő *Timer0* megszakítási szubrutinban történik az analóg-digitális átalakítás. A megszakítást nem lehet megszakítani, ezért az analóg-digitális konverzió végét jelző megszakítási szubrutin csak az éppen futó *Timer0* megszakítási szubrutin után juthatna érvényre. Így, az A/D átalakítóban a megszakítás érvényre jutását letiltva, ellenben a megszakításkérést jelző bitet figyelve, egyetlen függvény segítségével végzem el az átalakítást. Ezt megtehetem, hiszen az $5 \mu\text{s}$ átalakítási idő megfelelően kicsi ahhoz, hogy mellette még legyen idő a számítások elvégzésére. Gondoljunk csak arra, hogy a mintavételi időköz $400 \mu\text{s}$ és csak $51,2 \text{ ms}$ időközönként (minden 128. mintavett érték után) végez a processzor számítást. Így, ilyen rövid konverziós idő mellett nincs szükség arra, hogy az A/D átalakítással párhuzamosan a processzor más műveleteket végezzen, majd a konverzió végét egy megszakítással jelezze az analóg-digitális átalakító. A mintavételezés ütemezését megtekinthetjük a *31. ábrán*.

Az A/D átalakító 12 bites, tehát a működése során 0 és 4096 közötti számokat szolgáltat. Értelemszerűen a 0 az analóg földnek (0 V), a 4096 pedig az analóg tápfeszültségnek (3,3 V) felel meg. A mikrofon az analóg tápfeszültség felével van előfeszítve, így gerjesztetlen állapotban az A/D átalakító kimenetén 2048 jelenik meg ideális esetben. Azonban, a mintavételezés után minden mintából kivonok 2048-at, így a periodikus hangoknak egy 0 középső periodikus digitális számsorozatot feleltetek meg.

A mikrokontroller négy **UART** (Universal Asynchronous serial Receiver and Transmitter) perifériával rendelkezik. Ezek közül egyet (*UART1*) használok az alkalmazásban, amelyet úgy konfiguráltam fel, hogy a kiküldött adat 8N1 formátumú (8 adatbit, nincs paritásbit, 1 stop bit), a BaudRate pedig 115200 bps legyen. Az

4.3. A horkolás detektáló algoritmus implementációja

Az implementált, autokorrelációs függvény kiszámításán alapuló algoritmus blokkvázlata a 22. ábrán látható. Az algoritmus első lépéseként a mikrofon mintavételezett és digitalizált jelén a **FIR szűrést** kell megvalósítanunk. A 3.2.1.1. fejezetben leírtaknak megfelelően egy alul-, és egy felüláteresztő szűrést is el kell végeznünk a jelen. Az ehhez szükséges FIR szűrőket MATLAB-ban terveztem meg. Egy FIR szűrő együtthatóinak abszolút értéke kisebb, mint 1. Az LPC1768 azonban egy fixpontos processzor, ezért a szűrés implementálásához fel kell skáláznunk a szűrőegyütthatókat, majd a szűrés végén az eredményt megfelelő módon leskálázni. A megfelelő pontosság eléréséhez az eredeti szűrőegyütthatókat 2^{15} -el szoroztam meg, és az így nyert számokat a mikrovezérlő programmemóriájában 16 bites előjeles egész típusként tároltam (*const int16_t*). Azért érdemes kettő egész kitevőjű hatványával végezni a skálázást, mert a szűrés elvégzése után a leskálázáskor ugyanennyivel kell osztani, amely osztást már a mikrovezérlőnek kell elvégeznie. Ahogyan a DSP processzorok általában, úgy az LPC1768 típusú mikrovezérlő is az osztás műveletét csak körülményesen, több órajelciklus (2-12) alatt tudja elvégezni. Azonban, a kettő hatványával való osztás helyettesíthető egyszerűen a bitek léptetésével. Így, a szűrés által eredményezett 32 biten ábrázolt számokban a biteket 15 helyi értékkel jobbra tolva a leskálázás 1 órajelciklus alatt elvégezhető.



31. ábra - A beágyazott szoftver ütemezése

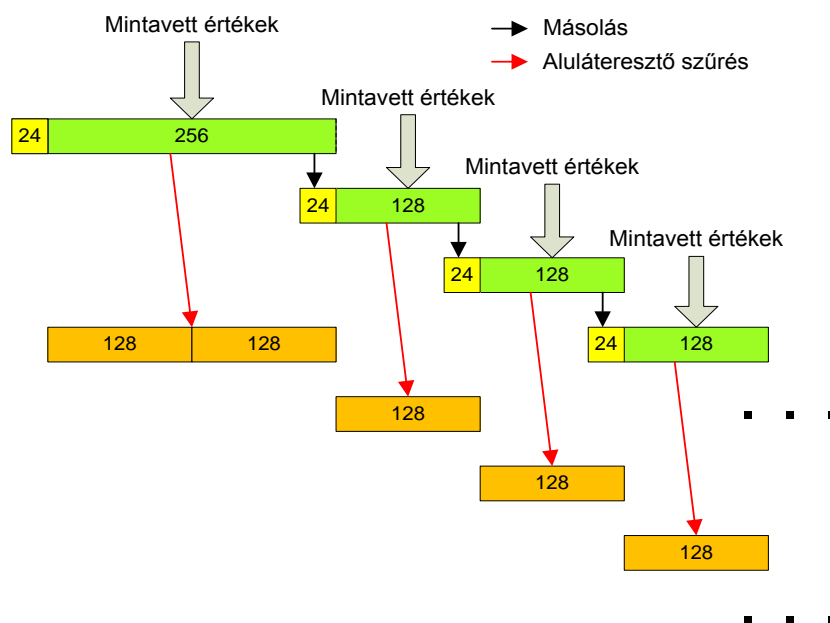
A 31. ábra az eszköz beágyazott szoftverének ütemezését mutatja. A horkolás detektáló algoritmus során a jelet ~ 100 ms-os szegmensekre bontom 50%-os átlapolással. 2500 Hz mintavételi frekvencia mellett egy szegmensnek 256 minta felel meg. Az 50%-os átlapolás azt eredményezi, hogy a mikrovezérlőben minden 128. mintavett érték után új számolás kezdődik. Az A/D átalakítás a mintavételezés ütemét biztosító *Timer0* megszakítási

szubrutinban történik. Kettő 128 méretű bufferbe gyűjtöm a mintákat úgy, hogy amíg az egyikbe történik a minták gyűjtése, addig a másikban lévő adatokon a mikrovezérlő elvégzi a számításokat. Minden 128. minta beérkezése után a bufferek szerepet cserélnek. Később látni fogjuk, hogy egy buffer (128 minta) feldolgozását a mikrovezérlő $7,2\text{ ms}$ idő alatt végzi el. Ez azt jelenti, hogy a processzor az aktuális buffer feltöltésekor az első $\frac{7,2\text{ ms}}{0,4\text{ ms}} = 18$ ütem alatt elvégzi az előző buffer feldolgozását. Így, a maradék 110 ütem alatt van elegendő idő az esetlegesen szükséges SD kártyára való írásra, soros porti, vagy zigbee kommunikációra.

A mintákon először az alul-, majd a felüláteresztő szűrést végzem el. A MATLAB-ban megtervezett aluláteresztő FIR szűrő 25, a felüláteresztő pedig 189 együtthatóból áll. A FIR szűrés során a szűrt jel n . mintájának (y_n) kiszámításához a következő konvolúciót kell elvégeznünk:

$$y_n = \sum_{k=0}^{N-1} h_k \cdot x_{n-k}$$

N a szűrőegyütthatók számát ($N=25$ az alul-, $N=189$ a felüláteresztő szűrő esetén), h a szűrőegyütthatókat, x pedig a szürendő jel mintáit jelöli. Látható, hogy a szűrt jel 0. mintájának kiszámításához a szürendő jel 0. mintáján kívül az az előtti $N-1$ darab mintára is szükség van. Ezért, véges hosszúságú vektorokon elvégezve a szűrést a szürendő jelet tartalmazó vektor elejét ki kell egészítenünk $N-1$ darab mintával, hogy a szűrt jel szinkronban legyen a szürendő jellel.

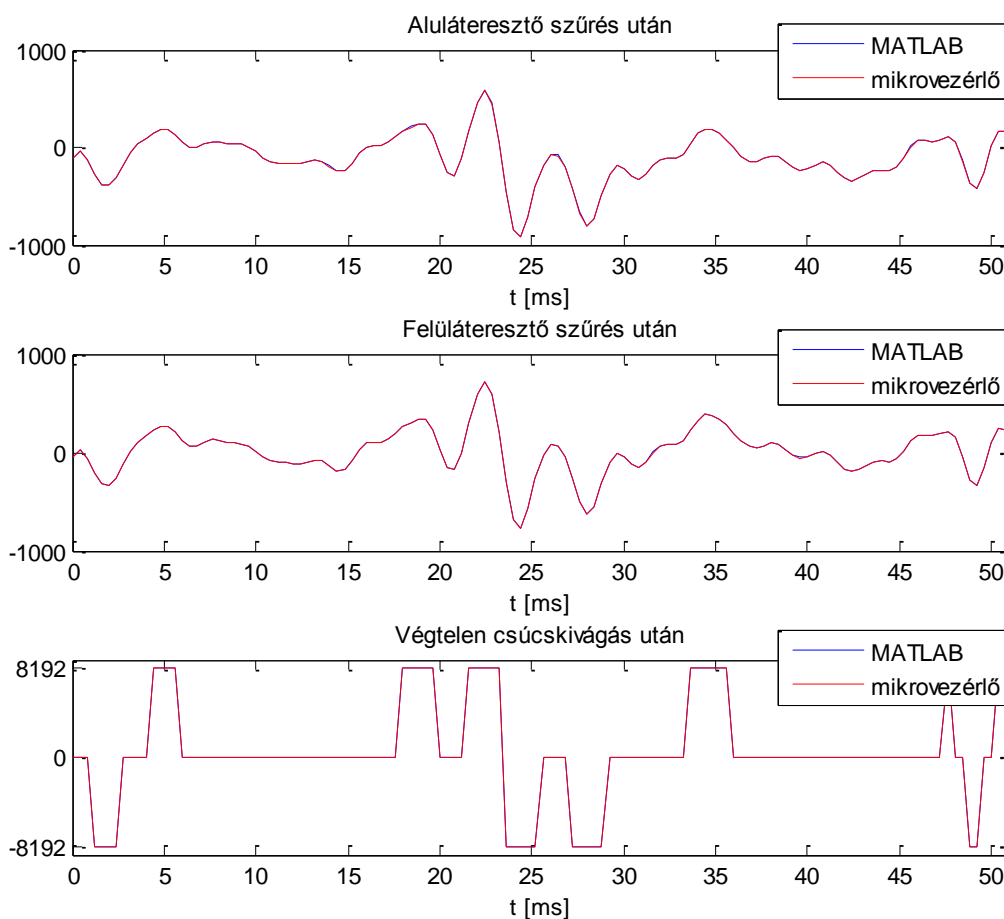


32. ábra - Aluláteresztő szűrés implementációja

Az algoritmus következő lépése a 3.2.1.2. fejezetben bemutatott végtelen csúcskivágás, amelynél szintén figyelembe kell vennünk, hogy egy fixpontos mikrovezérlőn történik az implementáció. Célunk a számítások minél pontosabb elvégzése, azonban ennek a mikrovezérlő véges számábrázolási pontossága, és véges pontosságú műveletvégző képessége határt szab. A számolás közben elkövetett relatív hibák akkor lesznek a legkisebbek, ha a számolások során a jelet úgy skálázzuk, hogy minél jobban kihasználjuk az ábrázolható számtartományt (minél nagyobb legyen az amplitúdó), ugyanakkor a legrosszabb esetben sem történjen túlsordulás. A használt processzor 32 bites, tehát az általa ábrázolható legnagyobb előjeles szám 2^{31} . Az algoritmus következő lépése a végtelen csúcskivágás után szegmensek autokorrelációs függvényének kiszámítása lesz, amelyet a szegmensek teljesítménysűrűség-függvényéből inverz Fourier transzformációval számítunk ki a 3.2.2.1. fejezetben megbeszéltek szerint. A teljesítménysűrűség-függvény a jel spektrumának abszolút érték négyzete. Mindezeket figyelembe véve kísérleti alapon arra jutottam, hogy biztosan nem lesz túlsordulás a processzorban, ha a **végtelen csúcskivágás** során a jel amplitúdóját $2^{13} = 8192$ -re állítom be:

$$y(n) = 8192 \cdot \text{sgn}[x(n)] = \begin{cases} 8192, & \text{ha } x > C_l \\ 0, & \text{ha } |x| \leq C_l \\ -8192, & \text{ha } x < -C_l \end{cases}$$

A beágyazott szoftver fejlesztése során a számítások helyességét és pontosságát, egy minta hangfelvételen ellenőriztem. A pontosság szempontjából a referenciát a MATLAB jelentette. A minta hangfelvételen a számításokat elvégeztem a MATLAB-bal, és a mikrovezérlővel is, és az eredményeket összevetettem egymással. A következő ábrán ezen hangfelvételnek egy 128 mintából álló ~50 ms-os részét láthatjuk az aluáteresztő szűrés, a felüláteresztő szűrés, illetve a végtelen csúcskivágás után:



34. ábra - Előfeldolgozás pontossága a mikrovezérlőn

Látható, hogy a mikrovezérlő által számolt adatok szinte tökéletesen lefedik a MATLAB által számoltakat. A felüláteresztő szűrés után az általam tapasztalt maximális eltérés a számítások között 0,9 értékű volt, amely a ~ 2000 értékű maximális jelamplitúdóhoz képest elhanyagolható hibának tekinthető. A végtelen csúcskivágás után azonban két mintasorozat tökéletesen megegyezik, hiszen itt az adott mintának csak a vágási szinthez való elhelyezkedése számít. Eltérést az okozhat, hogy a mikrovezérlő által kiszámolt, és a referencia minta a vágási szinthez képest ellentétesen helyezkednek el. Tekintve, hogy számítás során elkövetett relatív hiba nagyon kis értékű, ez az eset nagyon ritkán következik be. Ha azonban bekövetkezik, akkor sem okoz nagy hibát, hiszen a néha előforduló hibákat az algoritmus utolsó lépéseként alkalmazott medián szűrés kisimítja.

Az algoritmus következő lépése a szegmensek **autokorrelációs függvényeinek a kiszámítása**. Ezt a korábbiakban megbeszéltek szerint a frekvenciatartományban végzem, amelyhez a processzornak FFT (gyors Fourier-transzformáció), és inverz FFT műveleteket kell végeznie. Ezek hatékony implementálását már sokan elvégezték, ennek megfelelően az interneten számos ingyenesen letölthető FFT-t illetve inverz FFT-t végző programkód létezik.

Ezek közül az *Ivan Mellen* [32] által írt, kifejezetten Cortex-M3 magú mikrovezérlőkre optimalizált, assembly nyelven írt RADIX-4 típusú FFT és inverz FFT műveleteket megvalósító programkódot használtam fel. A RADIX-4 típusú FFT további optimalizálhatóságot jelent a szokványos (RADIX-2 típusú) FFT-hez képest, ellenben a feldolgozható blokk mérete csak 4 valamelyik hatványa lehet. A mi esetünkben a szegmensek mérete 256, amely 4-nek a 4. hatványa. Az mbed NXP LPC1768 fejlesztő kártya internetes fejlesztőkörnyezete támogatja az assembly betétek és külső assembly modulok alkalmazását, így az FFT-t és inverz FFT-t végző függvényeket könnyedén beilleszthettem a saját szoftverembe.

Az autokorrelációs függvény kiszámításához a mikrovezérlő a következő műveleteket végzi el minden 100 ms-os szegmensre:

Az első lépés az FFT elvégzése, majd az így kapott spektrumból a teljesítménysűrűség-függvény ($|X_k|^2$) kiszámítása.

$$X_k = FFT(x(n)).$$

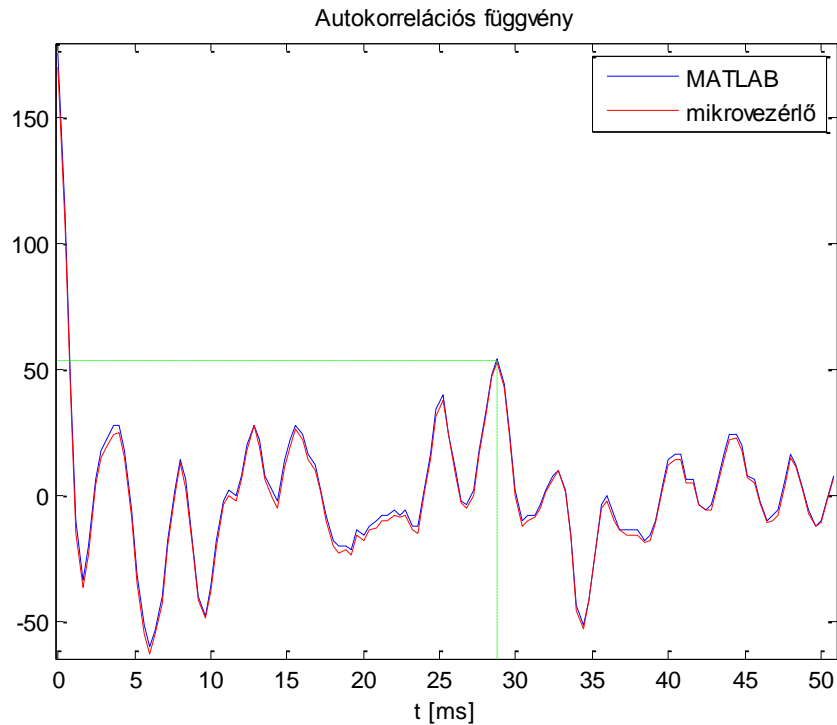
Az FFT-t végző függvény a spektrumkomponenseket valós rész és képzetes rész alakban szolgáltatja, így a teljesítménysűrűség-függvény komponenseit a következő képpen számítom ki:

$$|X_k|^2 = Re\{X_k\}^2 + Im\{X_k\}^2.$$

Ezek után az autokorrelációs függvény kiszámításához már csak az inverz FFT-t végző függvényt kell meghívunk:

$$R_{xx}(n) = IFFT(|X_k|^2).$$

A műveletvégzés pontossága szempontjából a Fourier-transzformáció elvégzése már nagyobb gondot jelent egy fixpontos mikrovezérlőnek, ugyanis ezen művelet során számos komplex exponenciálissal való szorzást kell elvégeznünk. Akár milyen jól használjuk is ki a processzor által ábrázolható számtartományt az operandusoknak a műveletvégzés idejére való felskálázásával, a processzor a végtelen tizedes törtekkel (pl. a komplex exponenciális számokkal) való szorzást csak véges pontossággal tudja elvégezni. A következő ábrán a korábban említett minta hangfelvétel egy szegmensének a MATLAB, illetve a mikrovezérlő által számított autokorrelációs függvényét láthatjuk.



35. ábra - Az autokorrelációs függvény számításának pontossága

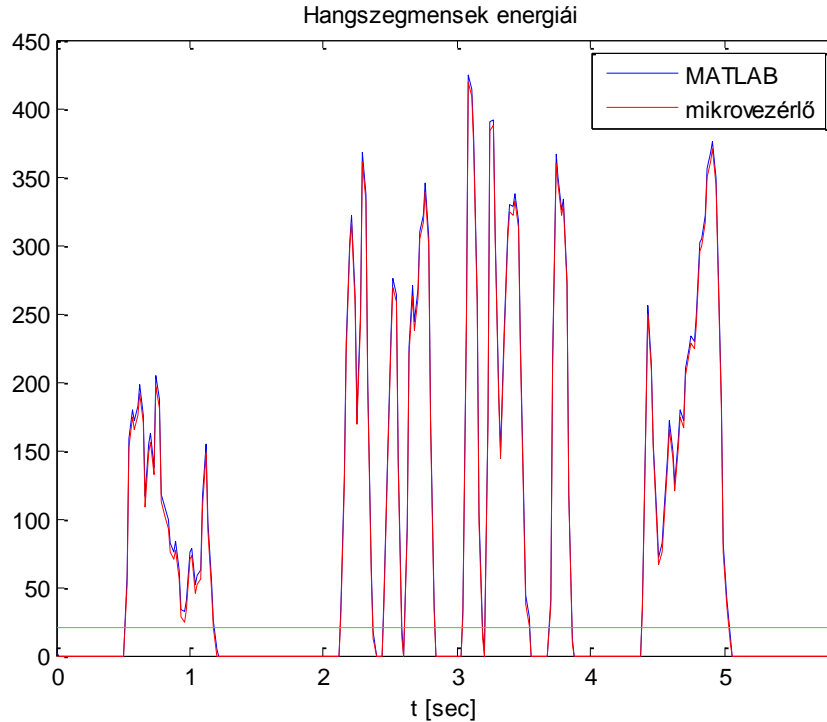
Látható, hogy a mikrovezérlő által számított adatok és a referenciaként használt MATLAB által számítottak már nem tökéletesen egyeznek meg egymással, a mikrovezérlő véges pontosságú számítási képessége miatt. Azonban, a számítási hiba még mindig megfelelően kis értékű, hiszen az ábrán látható piros és kék görbék jó közelítéssel azonosnak mondhatóak, továbbá mi csak a maximumra vagyunk kíváncsiak.

A 3.2.2.1. fejezetben megbeszéltek szerint az autokorrelációs függvény a következő alakú:

$$R_{xx}(n) = \sum_{i=0}^{N-1} x(i) \cdot x(i+n)$$

$R_{xx}(0)$ értéke (a jel energiája) alkalmas annak eldöntésére, hogy a jelszegmens ténylegesen hanghoz tartozik-e, vagy sem. Ha ez nagyobb egy bizonyos V_l küszöbértéknél (voice level), akkor az autokorrelációs függvény második legnagyobb értékű lokális maximumának helye a jelszegmens pseudo-periódusideje. Ezt láthatjuk bejelölve a 35. ábrán, amelyről így leolvasható, hogy a szegmenshez tartozó pseudo-periódusidő értéke 29 ms. Az alapharmonikus frekvencia értéke ennek reciproka: $\frac{1}{29 \text{ ms}} \approx 35 \text{ Hz}$.

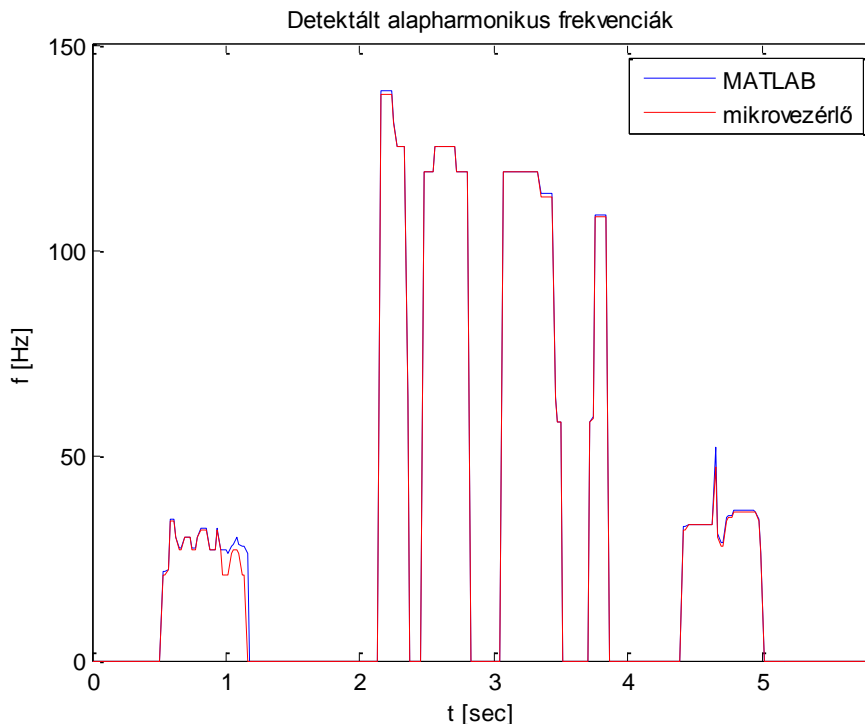
A 36. ábrán a minta hangfelvételre a mikrovezérlő, és a MATLAB által kiszámolt energia ($R_{xx}(0)$) értékeket láthatjuk.



36. ábra - A hangszegmensek energiáinak számítása

Az ábrán a zöld vonal a V_l döntési küszöböt (voice level) mutatja, amely feletti energia érték esetén az eszköz az adott szegmenst valóban egy hanghoz tartozónak ítéli. Továbbá, az ábrán megfigyelhető, hogy a mikrovezérlő által számított értékek jó közelítéssel megegyeznek a MATLAB által számítottakkal.

A döntési küszöböt meghaladó energiájú hangszegmensek detektált alapharmonikus frekvenciája az autokorrelációs függvény alapján számolt pszeudo-periódusidő reciproka. A közel 100 ms hosszúságú hangszegmensek 50%-os átlapoltsága következtében ~50 ms időközönként kapunk egy új becsült alapharmonikus frekvencia értéket. A 3.2.2.1. fejezetben leírtaknak megfelelően ezen alapharmonikus sorozaton egy utófeldolgozást végzünk: egy adott időpontban a számított alapharmonikus frekvencia helyett az utolsó 5 érték mediánja lesz a detektált alapharmonikus frekvencia. A minta hangfelvételen a teljes algoritmus eredménye a 37. ábrán látható.



37. ábra - A minta hangfelvételen detektált alapharmonikus frekvenciák

A számítás pontossága szempontjából referenciaként számító MATLAB és a mikrovezérlő jó közelítéssel ugyanazt az eredményt szolgáltatották. Az ábrából már egyértelműen megállapítható, hogy a minta hangfelvétel elején és végén lévő hangok horkoláshoz, míg a közepén lévők beszédhez tartoznak.

Összességében elmondhatjuk, hogy a mikrovezérlő a fixpontos számábrázolása ellenére alkalmas arra, hogy az algoritmushoz szükséges számításokat megfelelő pontossággal elvégezze. A valós idejű megvalósításhoz alapvető fontosságú a számítások megfelelő sebességű elvégzése. A hangszegmensek 50 %-os átlapolódása azt jelenti, hogy a mikrovezérlőnek 51,2 ms időközönként kell egy újabb számítást elvégeznie. Tehát, a számítások elvégzésére pontosan ennyi ideje van a mikrovezérlőnek. Ennek megvizsgálásához a számítások elejét és végét egy LED felvillantásával jeleztem a panelen, majd oszcilloszkóp segítségével lemértem a kettő között eltelt időt. Azt az eredményt kaptam, hogy egy szegmensen az algoritmust 7,2 ms alatt elvégzi a processzor, tehát a számítás sebessége is megfelelőnek bizonyult.

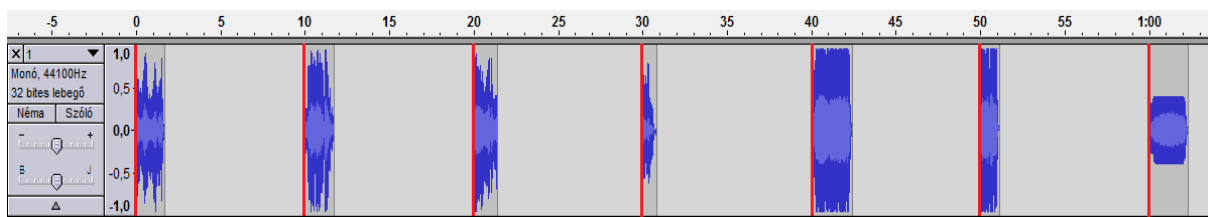
5. fejezet

Eredmények

A tesztelés elvégzéséhez az eszköz beágyazott szoftverét úgy konfiguráltam fel, hogy egy hangszegmenst akkor tekintsen horkoláshanghoz tartozónak, ha az adott, és a közvetlenül előtte lévő 6 darab hangszegmens mindegyike esetén a medián szűrés után kapott alapharmonikus frekvencia értéke 30 és 60 Hz között van. Az 50 %-os átlapolódással felvett 100 ms hosszúságú hangszegmensek miatt ez azt jelenti, hogy legalább 400 ms ideig 30 és 60 Hz között kell lennie a számított hangmagasságnak egy horkoláshang észleléséhez. Ha ez bekövetkezik, akkor az eszköz az SD kártyára egy .txt fájlba elmenti az észlelés pontos idejét év:hónap:nap óra:perc:másodperc formátumban. Egy horkoláshang észlelése után az eszköz 2 másodpercig vár, mielőtt újra elkezdené az algoritmust, biztosítva ezzel, hogy egy horkolás csak egyszer kerüljön rögzítésre az SD kártyán lévő naplózási adatok közé.

A teszteléshez 17 darab különböző horkoláshangot töltöttem le az internetről. Ezeket az *Audacity* nevű ingyenes program segítségével egy hangfájlba helyeztem el úgy, hogy egymástól pontosan 10 másodpercre helyezkedjenek el. Az eszközt úgy programoztam fel, hogy az indítása után az első észlelt horkoláshang bekövetkeztekor nullázza a valós idejű órájának (*Real Time Clock*) perc illetve másodperc számlálóját. Ezáltal az SD kártyára mentett horkoláshang észlelési időpontok szinkronba kerültek a hangfájl idejével, hiszen a tesztelésre készített felvétel rögtön egy horkoláshanggal kezdődik.

Az eszköz bekapcsolása után tehát, ezt a hangfájlt játszottam le. A tesztelés után a próba hangfelvételt az *Audacity*-ben ábrázoló képbe pirossal bejelöltem, hogy az egység mely időpontokban észlelt horkolást. Ennek az első 1 perces időtartamát ábrázolja a 38. *ábra*. Megfigyelhető, hogy a felvételen szereplő 17 különböző horkoláshang mindegyikét felismerte az eszköz.



38. ábra – Az első teszt hangfelvétel és az észlelt horkoláshangok

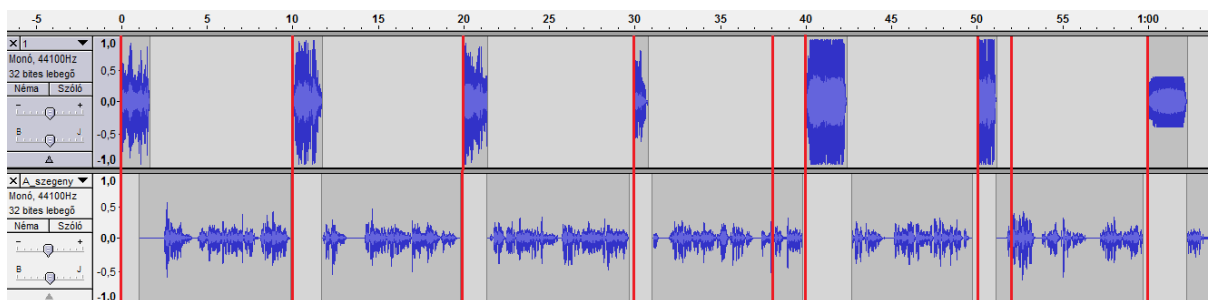
A következő lépésben az *Audacity* programmal ugyanezen horkoláshangok közé beszédhangokat vágtam. A második teszt hangfelvételen egy női hang beszél a horkoláshangok között. Ennek eredményét láthatjuk a 39. ábrán.



39. ábra – A második teszt hangfelvétel és az észlelt horkoláshangok

Ahogy az ábrán is látható az eszköz e teszt során már követett el hibát. Azonban, a közel 3 perces felvétel során mindösszesen 3 alkalommal észlelt horkoláshangot ott, ahol valójában beszédhang szerepelt. Mindemellett a valódi horkoláshangokat továbbra is jól felismerte.

Ezután a horkoláshangok közé férfi beszédhangot vágtam:

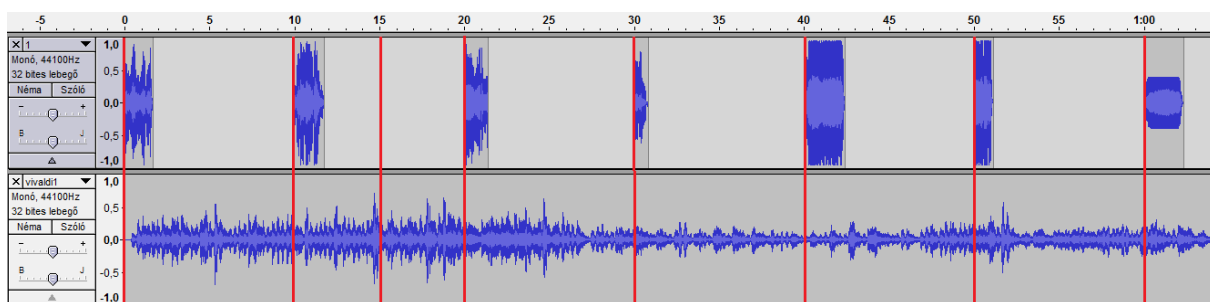


40. ábra – A harmadik teszt hangfelvétel és az észlelt horkoláshangok

A 3 perces felvétel során a valódi horkoláshangok felismerése mellett, 5 alkalommal észlelt az eszköz tévesen horkoláshangot.

A következő teszt során az előbbieknél alkalmazott horkoláshang sorozattal párhuzamosan *Vivaldi: A nyár* című versenyművének egy részét játszottam le. A szinte csupa

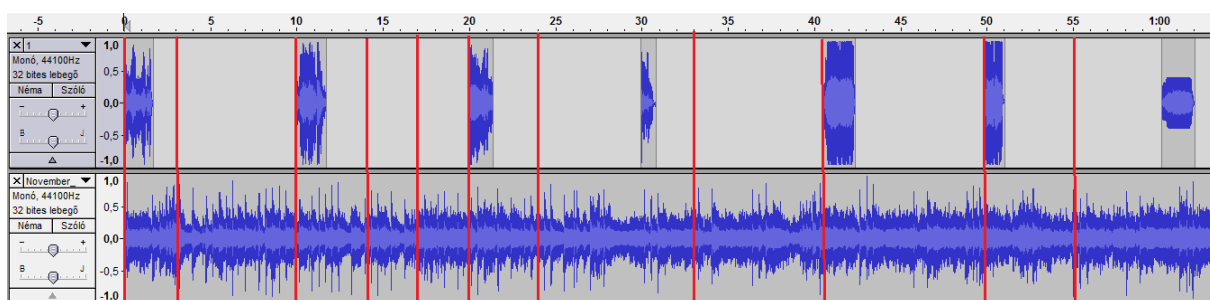
magas hangokból álló zenemű alatt mindösszesen kétszer észlelt az eszköz tévesen horkoláshangot, míg az igazi horkoláshangokat most is felismerte. Ennek az első egy perces részét láthatjuk a következő ábrán:



41. ábra – A negyedik teszt hangfelvétel és az észlelt horkoláshangok

Az utolsó teszt során a horkoláshang sorozat mellett egy rock számot játszottam le, amely tele volt mély basszus hangokkal továbbá a dob és cintányér okozta zörejekkel, zajokkal. A horkoláshangoknál helyenként hangosabb zene több alkalommal is elnyomta azokat, így a 17 közül csak 13-at ismert fel az eszköz. Továbbá a közel 3 perces teszt hangfelvétel alatt 16 alkalommal észlelt tévesen horkolás hangot. Ennek első percét mutatja a

42. ábra:



42. ábra - Az ötödik teszt hangfelvétel és az észlelt horkoláshangok

Összességében elmondható, hogy az eszköz jól ismerte fel az igazi horkoláshangokat mindaddig, amíg nem nyomta el azokat a hangos zene. Azonban feltételezhetjük, hogy a páciens, aki a horkolásáról szeretne információkat gyűjteni, nem hangos zeneszó mellett alszik, hanem lehetőleg csendes körülmények között.

A horkoláshangokat az eszköz viszonylag jól megkülönböztette a beszédhangoktól, azonban kétségtelen, hogy néha előfordultak téves horkoláshang észlelések.

A célom az volt, hogy az egyes horkolásnak tekintett események alapján el tudja dönteni az eszköz, hogy ténylegesen az alvás horkolási fázisa zajlik-e. Egy horkantást (bár megfelel a jelfeldolgozásban keresett jelnek) önmagában még nem tekintek horkolásnak, csak akkor, ha több ilyen hang érkezik adott időn belül. Ezt a következő módon valósítottam meg:

A mikrovezérlő *Timer2* belső időzítő/számláló egysége segítségével mérem egy horkoláshang észlelése után eltelt másodpercek számát. Az eszköz figyel, hogy egy észlelés után 2-6 másodpercen belül történik-e újabb horkolás. Ennek segítségével az eszköz a horkolás periodikusságára „rászinkronizál”, és ha ilyen módon egymás után 3 alkalommal történt horkoláshang észlelése, abban az esetben tekinti valóban horkolásból származónak a hangokat.

Mindezeket összevetve megállapítható, hogy az eszköz alkalmas arra, hogy egy csendes körülmények között alvó (értsd: nem hangos zene mellett) ember horkolását detektálja. Képes megkülönböztetni a horkoláshangot a szintén pszeudo-periodikus beszédhangoktól, így az alvás közben esetlegesen bekövetkező beszéd hangját nem tekinti horkoláshangnak. Az eszköz figyelembe veszi a horkolás periodikusságát, és csak a periodikusan 3-szor egymás után észlelt horkoláshang után dönt a horkolás tényéről.

6. fejezet

Összefoglalás

6.1. Áttekintés

A kutató-fejlesztő munka során kialakítottam egy eszközt, amely alkalmas az alvó egyén horkolásának detektálására. Az eszközzel szemben támasztott követelmény volt, hogy egyszerű és olcsó legyen, alkalmas legyen a vizsgált személy otthonában való megfigyelésére, továbbá, hogy őt az alvásában a lehető legkevésbé zavarja. Mindezeket figyelembe véve kialakítottam egy hardver egységet, amely egy mikrofon segítségével figyeli a környezetéből jövő hangokat. Ezeket tárolás nélkül helyben feldolgoza, és csak a horkolásra vonatkozó információt menti el a benne lévő microSD kártyára. Ezzel nem sérti meg a személyiségi jogokat, ugyanis a felvett hangokat nem tárolja el, illetve nem küldi el más egységnek. A vizsgált személynek tehát nem kell attól félnie, hogy az álmában esetlegesen elárult bizalmas dolgokat az eszköz rögzíti. Szemben a poliszomnográfias vizsgálatokkal, az általam fejlesztett készülék a páciens saját otthonában, mindeféle testére rögzített szenzorok nélkül képes megfigyelni.

Az egységet egy *ARM Cortex M3* magú *LPC1768* típusú nagyteljesítményű mikrovezérlő irányítja, amely olcsón (2-3 ezer Ft) beszerezhető, és alkalmas a nagyobb számításigényű jelfeldolgozási feladatok ellátására.

A kutatás során áttekintettem a szakirodalomban közölt, horkolás detektálásra is alkalmazható módszereket. Megvizsgáltam a talált algoritmusok megbízhatóságát, majd értékeltem azokat a szükséges erőforrás, és a valós idejű futás lehetőségének szempontjából. A vizsgálat után a legígéretesebb algoritmust megvalósítottam a mikrokontrolleres beágyazott környezetben, és laboratóriumi körülmények között teszteltem az elkészült készüléket.

6.2. Jövőbeni célkitűzések

Az *5. fejezetben* olvashattuk, hogy a készülék tesztelése a kifejezetten erre a célra készített teszthangfelvételek segítségével megtörtént. Az egyik legelső jövőbeni feladat az eszköznek a nem laboratóriumi körülmények között (pl. a megfigyelt személy otthonában) történő tesztelése.

A bevezetőben említettem, hogy a készülék a Méréstechnika és Információs Rendszerek Tanszéken kialakításra kerülő intelligens szoba egyik eszköze lesz. Feladata a szobában alvó ember horkolásának detektálása. A szobában lévő egységek zigbee-n keresztül kommunikálnak egymással, ennek megfelelően a hardver kialakításánál az eszközt alkalmassá tettem a zigbee alapú kommunikációra. A jövőbeni célkitűzések között szerepel tehát az eszköznek az intelligens szobában lévő zigbee hálózatba való integrációja.

A hardver kialakításánál céltom az volt, hogy az eszköz a horkolás detektálásán túl komplexebb alvászvizsgálatra is alkalmas legyen. Az előző félévekben többek között nyúlásmérő bélyeges terhelésmérő szenzorokkal is foglalkoztam. Az ágy négy lába alatt mérve a terhelést lehetőség adódik a páciens éjszakai mozgásainak megfigyelésére is. A panelt ennek megfelelően alkalmassá tettem arra, hogy négy darab ilyen nyúlásmérő bélyeges terhelésmérő szenzort csatlakoztathassunk rá. Az egyik jövőbeni feladat az eszköznek ilyen irányú továbbfejlesztése

Ahogy az az *1.2.5. fejezetben* bemutatott szakirodalomban [14] is láttuk, a horkoláson túl egyéb jellemzők (pl. izgalmi állapot, elesés stb.) hang alapján történő detektálására is lehetőség nyílik. Tekintve, hogy hardver alkalmas ezen feladatok ellátására is, a jövőben érdemes lehet ezen kutatások irányába is elindulni.

Ábrajegyzék

1. ábra - A horkolás detektáló rendszer blokkvázlata.....	7
2. ábra - A nyomásérzékelő és a mikrofon jele	7
3. ábra - Horkolás detektálás a mellkasra rögzített szenzorokkal.....	8
4. ábra - Két alapvetően különböző horkolás spektrum	9
5. ábra - A horkolás és beszédhang összehasonlítása.....	10
6. ábra - Pseudo-periódusidő meghatározása.....	11
7. ábra - A beszédhang, az egyszerű horkoláshang és az alvási apnoe szindrómás horkoláshang paramétereinek eloszlása.....	12
8. ábra - Beszédhang alapfrekvenciája a különböző érzelmi állapotokban.....	14
9. ábra - Az analóg rész kapcsolási rajza.....	17
10. ábra - mbed NXP fejlesztőkártya.....	18
11. ábra - Az mbed NXP fejlesztőkártya lábkiosztása	21
Az elkészült hardver a vezérlők nélkül.....	24
12. ábra - Az elkészült hardver a vezérlők nélkül	25
13. ábra - A teljes eszköz.....	25
14. ábra - ATZB-24-A2 Zigbit modul az adapteren	26
15. ábra - A Zigbit modul blokkvázlata.....	27
16. ábra - Egy horkoláshang időfüggvénye és spektruma	30
17. ábra - A "három" szóhoz tartozó beszédhang időfüggvénye és spektruma.....	31
18. ábra - 400 Hz sávszélességű aluláteresztő szűrő	34
19. ábra - Az alkalmazott feluláteresztő szűrő	35
20. ábra - Feluláteresztő szűrés hatása.....	36
21. ábra - Középkivágás és végtelen csúcskivágás alkalmazása	37
23. ábra - A szűrés és a végtelen csúcskivágás hatása az ACF-re.....	42
24. ábra - A medián szűrés hatása	44
25. ábra - A hangmagasság detektálás eredménye az ACF kiszámításán alapuló módszer esetében	44
26. ábra - Az ACF kiszámításán alapuló módszer frekvenciafelbontása	45
27. ábra - A harmonikus egyezés módszere	47
28. ábra - A Harmonikus Egyezés Módszerén alapuló algoritmus eredménye.....	50
29. ábra - A szűrés és a végtelen csúcskivágás hatása a HPS-re	51
30. ábra - SPI kommunikáció	57
31. ábra - A beágyazott szoftver ütemezése	58
32. ábra - Aluláteresztő szűrés implementációja	59
33. ábra - Alul- és feluláteresztő szűrés implementációja	60
34. ábra - Előfeldolgozás pontossága a mikrovezérlőn	62
35. ábra - Az autokorrelációs függvény számításának pontossága	64
36. ábra - A hangszegmensek energiáinak számítása.....	65
37. ábra - A minta hangfelvételen detektált alapharmonikus frekvenciák	66
38. ábra – Az első teszt hangfelvétel és az észlelt horkoláshangok	68
39. ábra – A második teszt hangfelvétel és az észlelt horkoláshangok	68
40. ábra – A harmadik teszt hangfelvétel és az észlelt horkoláshangok.....	68
41. ábra – A negyedik teszt hangfelvétel és az észlelt horkoláshangok	69
42. ábra - Az ötödik teszt hangfelvétel és az észlelt horkoláshangok	69

Irodalomjegyzék

- [1] A Dunai, A Keszei, MS Kopp, CM Shapiro, I Mucsi, M Novak, "*Cardiovascular disease and health care utilization in snorers: a population survey*" SLEEP, Vol 31/Issue 3 March 1, 2008
- [2] Satoshi Noda, Takeshi Minamiura and Hidetaka Sakai, "*Method of and device for snore detection*" Sanyo Electric Co.,Ltd., U. S. Patent, Jun.17,2008
- [3] P. Alexander Derchak, Lance Myers, Gary Michael Lucia, "*System and method for snore detection and confirmation*", Patent Application Publication, dec.13,2007
- [4] Robert B. Crossley, "*Multifunctional behavioral modification device for snoring, bruxism, and apnea*", U.S. Patent Documents Dec.29,1987, 4,715,367
- [5] Penzel, T., Amend, G., Meinzer, K., Peter, J. H., von Wichert, P., "*MESAM A heart-rate and snoring recorder for detection of obstructive sleep apnea*", Sleep, 13:175-182, 1990.
- [6] Penzel, T., Althaus, W., Meinzer, K., Peter, J. H., von Wichert, P., "*A device for ambulatory heart rate, oxygen saturation and snoring recording*", Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 13, :1616-1617, 1991.
- [7] Yeh-Liang Hsu, Ming-Chou Chen, Chih-Ming Cheng, Chang-Huei Wu, "*Development of a portable device for home monitoring of snoring*", Biomed Eng Appl Basis Comm, 2005(August); 17: 176-180.
- [8] Rymut et al., "*Method and apparatus for monitoring respiration*" US Patent application publication 2002/0072685 A1, 2002.
- [9] Schechter et al., "*Graphical readout of laryngotracheal spectra and airway monitor*" US Patent. Patent number 5.058.600, 1991.
- [10] J.A. Fiz, J. Abad, R. Jané, M. Riera, M.A. Mañanas, P. Caminal, D. Rodenstein, J. Morera, "*Acoustic analysis of snoring sound in patients with simple snoring and obstructive sleep apnoea*" Eur Respir J., 1996, 9, 2365–2370.
- [11] Pérez-Padilla JR, Slawinski E, Difrancesco LM, Feige RR, Remmers JE, Whitelaw WA "*Characteristics of the snoring noise in patients with and without sleep apnea*". Am Rev Respir Dis 1993, 147, 635-644.

[12] Dalmasso F, Prota R, "Snoring: analysis, measurement clinical implications and applications". Eur Respir J, 1996. 9, 146-59.

[13] U.R. Abeyratne, C.K.K. Patabandi and K. Puvanendran, "Pitch-jitter analysis of snoring sounds for the diagnosis of sleep apnea", 2001 Proceedings of the 23rd Annual EMBS International Conference, October 25-28, Istanbul, 2072-2075

[14] Marius Vasile Ghiurcau, Adrian Lodin, Corneliu Rusu, "A Study of the Effect of Emotional State upon the Variation of the Fundamental Frequency of a Speaker", Journal of Applied Computer Science & Mathematics, no.7 – Special Issue

[15] T. Kinnunen, E. Karpov, P. Franti, "Real-time speaker identification and verification", in IEEE Transactions on Audio, Speech and Language Processing, vol. 14. no. 1, pp. 277-288, Jan. 2006.

[16] Az mbed NXP LPC1768 fejlesztőkártya honlapja, leírása

<http://mbed.org/nxp/lpc1768/>

[17] A Texas Instruments® által gyártott TLC2272 típusú műveleti erősítő adatlapja

<http://focus.ti.com/docs/prod/folders/print/tlc2272.html>

[18] A Texas Instruments® által gyártott LM324-es típusú műveleti erősítő adatlapja

<http://focus.ti.com/docs/prod/folders/print/lm324.html>

[19] Az Analog Devices® által gyártott, AD5248 típusú digitális potenciométer adatlapja

<http://www.alldatasheet.com/view.jsp?Searchword=AD5248>

[20] Az LPC1768 típusú mikrovezérlő adatlapja

<http://www.alldatasheet.com/datasheet-pdf/pdf/275723/PHILIPS/LPC1768.html>

[21] A Cortex-M3 processzor mag leírása az ARM honlapján

<http://www.arm.com/products/processors/cortex-m/cortex-m3.php>

[22] Cortex-M3 Technical Reference Manual

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337e/index.html>

[23] Dr .Schnell László: "Jelek és rendszerek mérés technikája III." Tankönyvkiadó, Budapest, 1990

[24] Holics László, Abonyi Iván, Beleznyai Ferenc, Csákány Antal, Flórik György, Juhász András, Tasnádi Péter, Sükösd Csaba: "Fizika 1.-2." Műszaki Könyvkiadó, Budapest, 1992, ISBN 963 10 9452 9

[25] Böhm Tamás Mihály: "Formánskövető és módosító algoritmus felhasználási lehetőségeinek vizsgálata" Diplomatervezés, Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest, 2003.

[26] Li Tan and Montri Karnjanadecha, "Pitch detection algorithm: Autocorrelation Method and AMDF" Proceedings of the 3rd International Symposium on Communications and Information Technology, 2: 551-556, September 2003.

[27]. M. M. Sondhi, "New methods of pitch extraction" IEEE Trans. Audio Electroacoust., vol. AU-16, pp. 262-266, June 1968.

[28] G. S. Ying, L. H. Jamieson, and C. D. Mitchell, "A probabilistic approach to AMDF pitch detection" in Proc. Int. Conf. Spoken Language Processing, Philadelphia, PA, Oct. 1996, pp. 1201–1204.

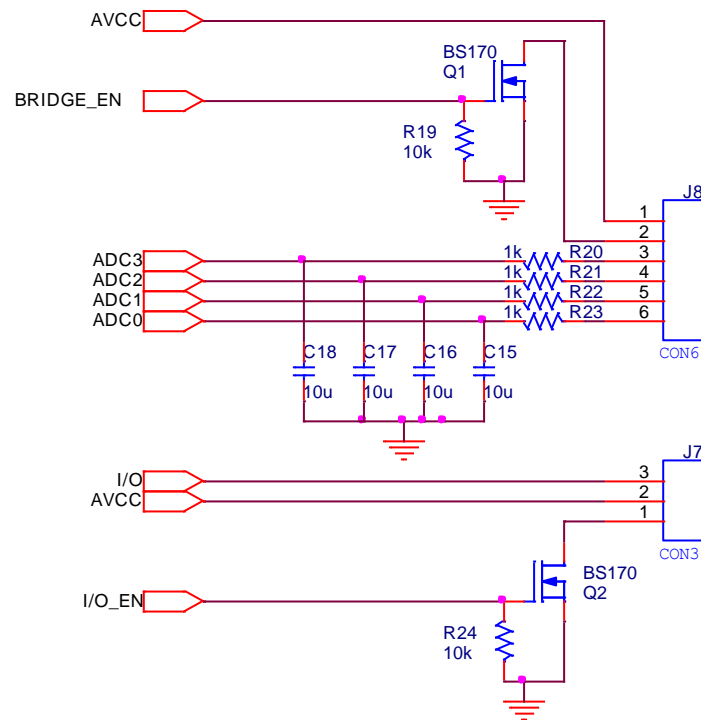
[29] M.R. Schroeder, "Period histogram and product spectrum: New methods for fundamental frequency measurement" JASA, 43(4):829-834, 1968.

[30] Jeong J., Moir T.J., „Two-Microphone Cepstrum Approach to Real-Time Speech Enhancement Methods” 2006 IEEE International Conference, Digital Object Identifier: 10.1109/ICEIS.2006.1703199

[31] Az mbed NXP LPC1768 fejlesztőkártya internetes fejlesztőkörnyezete
<https://mbed.org/compiler/>

[32] Ivan Mellen "Complex 16 bit Radix 4 FFT and Inverse FFT for Cortex-M3" Ingyenesen elérhető FFT és inverz FFT műveleteket végző assembly programkód Cortex-M3 magú mikrovezérlőkre.
<http://www.embeddedsignals.com/FFTCM3.s>

2. melléklet: A mérő panel kapcsolási rajza (második oldal)



3. melléklet: A mérő panel NYÁK terve

