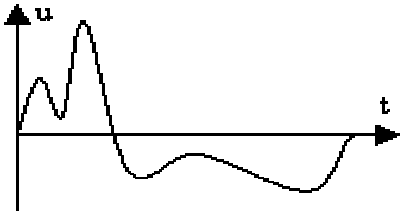
Hangfeldolgozás

Az alábbi alapfogalmak ismerete a mérés megértése és elvégzése szempontjából elengedhetetlen. Az ismertetés rövid, lexikonszerű. Erről a témáról bővebben jelfeldolgozással foglalkozó szakkönyvek írnak.

Idő- és frekvenciatartománnyal kapcsolatos fogalmak:

-analóg jel: időben változó feszültség- (áram-) érték. Értelmezési tartománya a pozitív időtengely, értéktartománya folytonos - feszültség (áram). Az értéktartomány általában előre ismert és korlátos.



-(frekvencia-) spektrum: egy (kör)frekvencia függvény, az analóg jel (időfüggvény), frekvencia-tartománybeli megfelelőjének (transzformáltjának) abszolút értéke ½ F(w)½ . Lásd Fourier transzformáció.

-spektrogram: kétváltozós (idő és frekvencia) függvény. Egy hosszabb időfüggvényt időtartományokra (szeletekre) bontunk úgy, hogy kis egymásutáni időfüggvények sorozata adja ki az eredetit. Ezután mindegyik kis időfüggvényből készítünk egy frekvenciaspektrumot. A frekvenciaspektrumok egymásutánisága jelenti a spektrogramot.

-Fourier transzformáció: egy eljárás, amelynek segítségével az idő- és frekvenciatartományok közötti átmenet elvégezhető. Ha egy T periódusú analóg jel transzformációja a cél, akkor *Fourier Sorbafejtésről (1°)* beszélünk. Ha analóg jelünk aperiódikus, akkor a *Fourier Integrál (2°)* módszert kell alkalmaznunk. Egy (analóg jel digitalizálásából eredő) számsor Fourier transzformációjára is van lehetőség, ekkor *Diszkrét Fourier transzformációnak (3°)* nevezett módszert (DFT) használunk, míg ennek gyorsított algoritmusa a *Gyors Fourier transzformáció (4°)* (FFT).

A transzformációk eredménye egy (w -tól függő) komplex függvény **F(w)**. Hangfeldolgozó programokban frekvenciaspektrum címén, általában ennek a függvénynek az abszolút értékét ábrázolják $= \sqrt{(RE(F(\omega)))^2 + (IM(F(\omega)))^2}$. A teljességhez hozzátartozna a komplex függvény fázisa is, azonban a fül csekély mértékű fázisérzékenysége miatt, ezt a legtöbb hangfeldolgozó eljárás elhanyagolja.

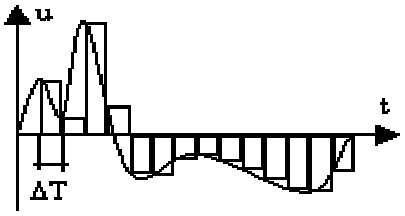
(1°) esetén csak $w_0 = 2p/T$ (alapfrekvencia) egész számú többszörösei (felharmonikusok) **w = k× w 0** esetén lehet az **F** függvénynek értéke (vonalas spektrum!). Ezzel szemben *(2°)* folytonos **F(w)**-t eredményez.
Ha **n** elemből álló számsort transzformálunk a *(3°)* vagy *(4°)* szerint (*(4°)* esetén n csak 2 egész számú hatványa lehet), akkor eredményül **2× n** darab valós számot kapunk. Az első **n** sorozat jelenti a transzformált **valós**, míg a második **n** darab, a transzformált **képzetes** részét. Az **n**-edik elemek (mindkét sorozatból) éppen az $\mathbf{f_{mv}}$ mintavételi frekvenciához (lásd alább) tartozó frekvenciakomponenst képviselik. Ebből

következik, hogy a frekvenciatengely felbontása $\mathbf{f_{mv}/n}$. Pl.: 44.1kHz-el mintavételezett hanganyagot 1024 mintánként FFT-zek. A spektrum frekvencia felbontása 44.1kHz/1024 » 43Hz.

-szűrők, szűrés fogalma: egy frekvenciaspektrum módosító eljárás. Analóg jelünk eredeti spektrumát szűrők segítségével torzítjuk. Bizonyos frekvenciakomponenseket (-tartományokat) kiemelünk (erősítünk), míg másokat elnyomunk. A szűrés célja a kellemesebb hanghatás elérésétől (szubjektív), az (objektív) mérési eredmények zajból való kihalászásáig igen sokrétű lehet. Szűrőzés eszközéül szintén több módszer közül válogathatunk. Szűrhetjük az analóg jelet **közvetlenül** analóg szűrők segítségével. Ezek lehetnek passzív, vagy aktív RLC szűrők. Az L,C elemek w -függő tulajdonságain alapulnak. **Közvetett** módon akkor szűrünk, amikor az analóg jelet előbb digitalizáljuk (számsorrá alakítjuk - lásd alább!), majd ezen a számsoron végzünk különböző műveleteket úgy, hogy a módosított számsorból visszalakított analóg jel olyan legyen, mintha az eredetit szűrtük volna közvetlen módszerekkel.

Analóg-digitál konverzióval kapcsolatos fogalmak:

-mintavételezés: során a folytonos időfüggvényből D **T** időközönként értékeket, "mintákat" veszünk. Ezek továbbra is analóg értékek, viszont fontos bemeneti jeléül szolgálnak egy analóg-digitál átalakítónak. A mintavételezést ún. mintavevő és tartó áramkörök végzik.



-mintavételi frekvencia (fmv): kifejezi, hogy másodpercenként hány mintavétel történik.

$$\mathbf{f_{mv}=1/D\ T}.$$

-mintavételi törvény: azt az időfüggvényt, amelynek frekvenciaspektrumában előforduló legmagasabb frekvenciájú komponens $\mathbf{f_{max}}$ értékű, legalább $\mathbf{f_{mv}} \geq \mathbf{2\ f_{max}}$ mintavételi frekvenciával kell mintavételeznünk. Pl. a zenei CD-kre rögzíteni kívánt legmagasabb frekvenciájú hang – az emberi fül felső hallhatósági határául elismert 20kHz – több mint kétszeresével (szabványban rögzített 44.1 kHz-el) mintavételezik a rögzítendő hanganyagot.

-analóg-digitál konverzió: egy feszültség értéket (előzőleg mintavételezett analóg jel mintáit) számmá alakít. Időigényes művelet. Mivel a következő minta beérkezése előtt az aktuális minta konverziójával végezni kell, a mintavételezés sebességének felső korlátját többnyire az analóg-digitál átalakítás ideje szabja meg. Vannak nagyon gyors, közepes sebességű és egészen lassú AD konverterek. A sebességen (konverziós időn) túlmenően az AD átalakítók másik fontos jellemzője a felbontás. Ez egy bitszámban kifejezett érték (**n**), és azt mutatja meg, hogy az átalakító által kezelhető bemeneti maximális feszültségtartományt $\mathbf{2^n}$ diszkrét értékre szabdalja szét.

-digitál-analóg konverzió: számot feszültséggé alakító eszköz. A konverzió gyakorlatilag azonnali. Jellemző paraméterei a bitszám, ill. a kimeneti feszültségtartomány. Bitszám darab 0 bemenet esetén a kimeneten a feszültségtartomány minimuma, míg bitszám darab 1-esre a feszültségtartomány maximumát kapjuk a kimeneten.

Jelfeldolgozás

Egy mérés (kísérlet) eredményeként kapott (nyers) analóg jel a legtöbb esetben további feldolgozást igényel. A feldolgozás célja a jel (mérési eredmények) megváltoztatása úgy, hogy a mérés szempontjából hasznosnak ítélt információkat kiemeljük, a haszontalanokat (zavaró információkat) pedig minél inkább

csökkentsük. A feldolgozás történhet időtartományban – pl. a jel amplitúdójának módosítása által – (lásd dinamika kompresszió a Cool Edit Pro program ismertetésénél), de végezhetünk frekvenciatartománybeli korrekciókat is – pl. szűrőzés.

Más szempontból vizsgálva a feldolgozás történhet közvetlen módon az analóg jelen végezve, de történhet közvetett módon is, amikor az analóg jelet (mintavevő és tartó, valamint AD átalakító áramkörökkel) számsorrá alakítják. Ezen a számsoron elvégzik a szükséges feldolgozó műveleteket, majd egy DA átalakító segítségével ismét analóg jellé (feszültséggé) alakítják.

Aszerint, hogy a feldolgozás a méréssel egyidőben, vagy éppen ellenkezőleg, a mérési eredmények rögzítése után történik (a feldolgozás időben késik), megkülönböztetünk “online” és “offline” feldolgozásokat.

Első pillanatban úgy tűnik, hogy a közvetlen (analóg) feldolgozás mindig “online”, a közvetett (digitális) pedig “offline”. Ez a legtöbb esetben így is van, de elvben analóg feldolgozás is történhet “offline” - pl. egy analóg magnetofon által rögzített jel későbbi feldolgozása -, és a közvetett (digitális) feldolgozás is történhet “online” módon, az egyre gyorsabb digitális elektronikai eszközök által.

Napjainkban a jelfeldolgozás egyre inkább közvetett (digitális) irányba tolódik. Köszönhető ez a digitális elektronika és a számítógépek rohamos fejlődésének, miközben áruk folyamatosan csökken. A számítógépek (általában a digitális elektronikai áramkörök) sebességének rohamos növekedtével az “online” digitális feldolgozás is teljesen hétköznapi és megszokott. Elég, ha csak arra gondolunk, hogy egy bő évtizeddel ezelőtt egy zenei hangstúdió létesítését és üzemeltetését csak nagyobb lemezkiadók, rádió és tv stúdiók engedhették meg maguknak. Manapság azonban gyakorlatilag bárki, saját otthonában berendezhet ilyet, csak egy multimédiás PC-re van szüksége.

Ez a mérés a jelfeldolgozás egy speciális esetével, a hangfeldolgozással foglalkozik. Tágabb értelemben vett hangfeldolgozásról akkor beszélünk, amikor a feldolgozandó analóg jel amplitúdója (n x 0.1V – n x 1V) tartományba, frekvenciaspektruma pedig a hallható (20 – 20kHz) tartományba esik.

Néhány szó a hangkártyákról

Az IBM PC-ket eredetileg csak igen szerény hangképzési lehetőségekkel látták el. Beépített hangszórója adott frekvenciájú és TTL szintű négyszögjelet tudott megszólaltatni. A PC-k képességeit azonban nagymértékben növelhetjük kiegészítő kártyák beépítésével. Ennek eredményeként jelentek meg a különböző típusú hangkártyák. Kezdetben pl. Adlib kártya esetén csak a hangkeltés volt a cél. A kártyán digitálisan vezérelhető analóg hangkeltő áramkörök (generátorok) találhatók. A hangkártyák fejlődésével a Sound Blaster a hangdigitalizálás és a hangminták lejátszása terén hozott újat. Kezdetben csak 8 bites hangfeldolgozási alapfeladatok ellátására volt képes, később a Sound Bluster teljes családdá fejlődött, aminek következtében lényegében a hangkártyák “szabványává” vált. A család újabb generációinak lehetőségei közül a fontosabbak:

-továbbfejlesztett DSP (Digital Sound Processor) egység, sztereó/mono felvétel/lejátszás 8 ill. 16 bites módban,

-programozható mintavételi frekvencia 4 - 45kHz-ig,

-bővített mixer chip az audioforrások keveréséhez,

-szoftverből állítható hangerő mindegyik bemenethez ill. a kimenethez,

-magas és mély hangkiemelési lehetőség,

-automatikus v. fix erősítés a mikrofonhoz,

-dinamikus szűrők a felvételhez és a visszajátszáshoz,

-beépített sztereó erősítő 4W/csatorna,

-bővített OPL-3 FM chip, amely sokgenerátoros hangkeltésre alkalmas,

-MIDI interfész,

-CDROM interfész,

-teljes kompatibilitás az előző változatokkal.

Zene- és hangfile formátumok

A hangkártyák fejlődésének útja meghatározta azokat a tárolási módokat (fileformátumokat), amelyek segítségével az előállított hangot (zenét) háttértárolókon rögzíteni lehet.

Mivel kezdetben a hangkártyák generátoraik segítségével csak hangkeltésre voltak alkalmasak, ezért olyan formátumok születtek, amelyek szorosan alkalmazkodtak a hardver adta lehetőségekhez. A generátorok keltette zene rögzítése nagyon egyszerű és tömör feladat. Az ilyen információkat tartalmazó fileokat **zenefileoknak** hívják. Csak az egyes generátorok megszólalásának paramétereit és időtartamát kell tárolni. Pl: az SBI (Sound Bluster Instruments) fileokat az FM hanggenerálással megszólaló hangszerek paraméterei alkotják. A file mérete 51byte!

Egy másik zenefile formátumot - a CMF-t (Creative Music File), a Creative Labs dolgozta ki, FM hangszerekkel lejátszott zene tárolására. A zenefilet három fontos részre lehet bontani: fejléc (header block), hangszerek (instrument block) és zene (music block). A fejléc a file fontosabb adatait, a hangszerblokk a felhasznált FM hangszerek leírását, a zeneblokk pedig a lejátszandó zenét leíró MIDI eseményeket tartalmazza.

További zenefileok: a MOD, amelyet Commodore Amiga gépekre fejlesztettek ki. Formátuma szorosan illeszkedik ezen gépek hardver lehetőségeihez. Különösen a játékok területén terjedt el olyannyira, hogy sok PC-s játék is ezt a formátumot használja.

A MIDI elektronikus hangszerek és a hozzájuk csatlakoztatható egyéb eszközök szabványává vált. Eredetileg a MIDI csupán néhány ésszerű ajánlás volt az elektronikus hangszerek közötti kommunikációhoz, amelyet a hangszergyártók vagy használnak vagy nem. Mivel azonban az egész rendszer ésszerű és könnyen megvalósítható, egyre többen alkalmazták. Az utóbbi években gyakorlatilag mindenki elfogadta, s ma már nem létezik olyan szintetizátor, amelynek ne lenne MIDI csatlakozója. A MIDI napjainkban sokkal többet jelent, mint egyszerű kommunikációs szabványt. A MID fileok MIDI üzeneteket (szekvenciákat) tárolnak. Egy MIDI üzenet az esemény leírásából (MIDIevent), és bekövetkezésének idejéből áll. Az elv hasonló, mint a kezdeti formátumok esetén, azzal a különbséggel, hogy azok egyedi típusú hangkártyákhoz tartoztak, a MIDI viszont a számítógépes zeneszerkesztők egységes formátumává vált.

A hangminták felvételének és lejátszásának lehetősége maga után vonta olyan fileformátumok megjelenését, amelyek a felvett minták számértékeit tárolják valamilyen formában. Az ilyen fileokat **hangfileoknak** hívják. Lényeges különbség a két formátum fajta között az, hogy a digitálisan tárolt hangmintákat tartalmazó fileok sokkal nagyobb terjedelműek, mint az FM chip "programozását" leíró zenefileok.

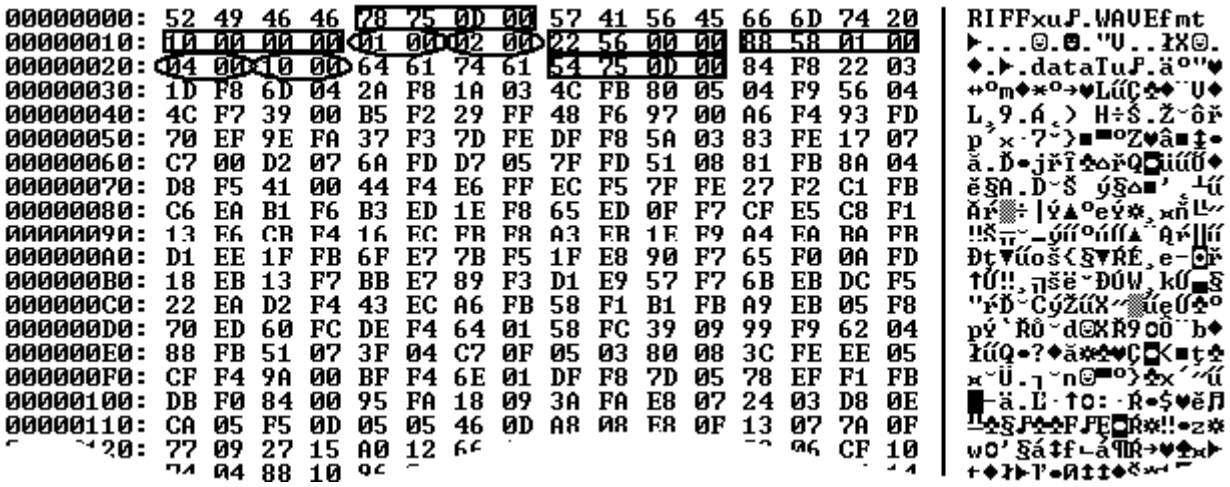
A VOC fileok rövid fejléccel kezdődnek, majd különálló blokkokkal folytatódnak. A blokkok lehetnek: hangminta-, hangminta folytatás-, üres- és lezáró blokk. A blokkok első byteja határozza meg a blokk típusát, majd a következő három byte a blokk hosszát. A további adatokat blokkspecifikusan kell értelmezni.

A WAV fileokat a Microsoft definiálta, tulajdonképpen a RIFF - mint általános multimédiás formátum - speciális esete. A WAV fileok annyira elterjedtek (és egyszerűek), hogy szerkezetük ismertetését az alábbiakban megkíséreljük:

Eltolás:	Mező:	Hossz:	Jelentés:
00h	"RIFF"	4 byte	azonosítja a Riff formátumot
04h	rDataLength	4 byte	Riff adatok hossza
08h	"WAVEfmtÈ "	8 byte	ez a Riff egy Wave
10h	wHeaderLength	4 byte	Wav fejléc hossza

14h	wFormat	2 byte	hullámtárolási mód = 1 tömörítetlen,
16h	nCh	2 byte	csatornák száma = 1 mono, =2 sztereó
18h	fmv	4 byte	mintavételi frekvencia Hz-ben
1Ch	BytePerSec	4 byte	átviteli sebesség Byte/s-ban = fmv*BytePerBlock
20h	BytePerBlock	2 byte	BytePerBlock csoportokban helyezkednek el a minták az adattömbön belül =nCh*(BitPerCh / 8)
22h	BitPerCh	2 byte	8 ill. 10h (8 v. 16 bites minták)
24h	"data"	4 byte	azonosítja az adatblokk kezdetét
28h	wDataLength	4 byte	minták adattömbjének hossza
2Ch	wDataArray	wDataLength byte	minták adattömbje, wDataLength hosszú

Példaként egy wav file kezdete látható az alábbiakban. A file mérete: 882048 byte.



000D7578h = 882040 byte a Riff adathossz 00000010h = 16 byte wav fejléc hossza

0001h = 1, azaz tömörítetlen wav file 0002h = 2, azaz sztereó

00005622h = 22050 Hz a mintavételi frekvencia 00015888h = 88200 byte/s átv.seb. (=22050*2*2)

0004h = 4 byte/blokk 0010h = 16 bit/minta

000D7554h = 882004 byte hosszú adattömb köv.

Ezután következnek a bal ill. jobb csatornák 16 bites mintái:

L csatorna: F884h, F81Dh, F82Ah, F84Ch, F904h, F74Ch ¼

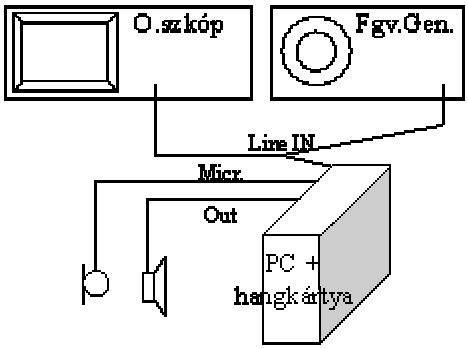
R csatorna: 0322h, 046Dh, 031Ah, 0580h, 0456h, 0039h ¼

Az eddigiekben ismertett hangfileok egy az egyben tartalmazták a minták értékeit bináris formában (természetesen a megfelelő fejléccel). Ha utána számolunk, hogy egy 22050 Hz-el mintavételezett, 16 bites mono csatorna percenként 22050Hz*2Byte*60s = 2646000/(1024*1024)MByte = 2.52MByte adatot termel, felmerül a kérdés: nem kellene-e valamilyen tömörítő eljárást alkalmazni. Az általánosan használt

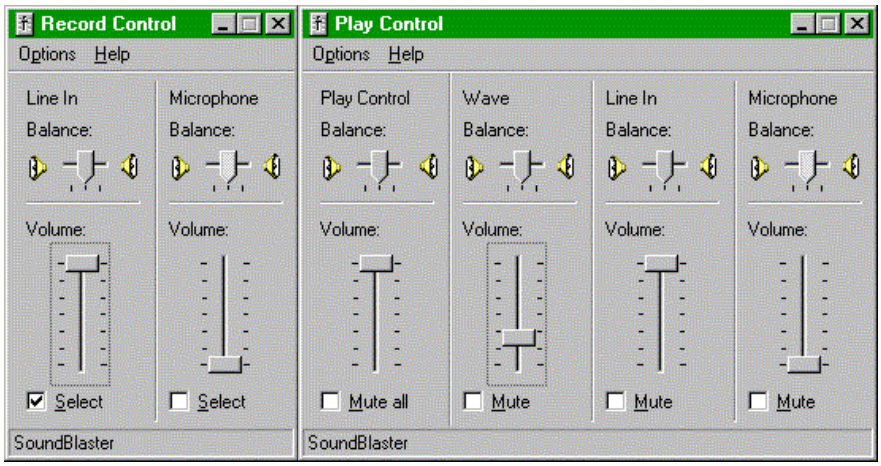
filetömörítő eljárások (zip, arj, lha, rar) gyakorlatilag semmit sem tudnak tömöríteni a hangfileokon. Ezért speciális - a hangminták sajátosságait figyelembe vevő - tömörítő algoritmusokat alkalmaznak. A legegyszerűbb ilyen eljárás abból a meggondolásból indul ki, hogy két egymásután következő minta értéke nem különbözhet radikálisan egymástól. A minták nem független egyedi "események", hanem igen is "függnek", hasonlítanak szomszédaikhoz. Ezért felesleges eltárolni minden egyes minta értékét például 16 biten, elegendő lehet a különbség tárolása, amely esetleg 4 biten is elfér. Máris 4:1-hez tömörítést végeztünk. Ez egy nagyon primitív tömörítő eljárás, napjainban sokkal bonyolultabb algoritmusokon nyugvó tömörítők terjednek (a szerzői jogvédők nem kis "örömére"), ezek közül legfontosabb az MP3-nak nevezett eljárás. Eredetileg mozgóképek tömörítésére dolgozták ki az MPEG eljárást. Fejlődési állomásai MPEG Layer 1, ...2, majd MPEG Layer 3. Ez utóbbit hívják röviden mp3-nak, mely egy veszteséges tömörítő eljárás. Lényege: figyelembe veszi az emberi fül sajátosságait mind amplitúdó-, mind pedig frekvenciatartományban. Amplitúdó tekintetében a digitalizálás nem egyenletes, csak a halkabb hangok esetén szükséges a finomabb felbontás, a hangosabb mintákat durvábban is quantálhatjuk. Frekvencia tekintetében az emberi fül érzékenysége 1kHz környékén a legnagyobb, a hallható tartomány két végén sokkal gyengébb. Ezért a rögzíteni kívánt hanganyagot frekvenciasávokra osztják és külön-külön rögzítik. Azok a sávok érdemelnek precízebb rögzítést, amelyekre a fül is érzékenyebb. A többi sávot elegendő kisebb pontossággal (durvább felbontással) rögzíteni.

Mérési összeállítás

Az összeállítás lényege egy PC számítógép és a benne elhelyezett hangkártya. A hangkártya két bemenettel és egy kimenettel rendelkezik. A két bemenet egyike a “Line IN”, amelyre egy függvénygenerátor jele kapcsolódik, a másik “Micr.” névre hallgat és egy mikrofon csatlakozik rá. A kimenetet “Out”-nak hívják és egy hangszórót hajt meg.



A bemenetek ill. a kimenet érzékenysége szoftveresen állítható. Ehhez kattintsunk kétszer a Win95 tálcáján található hangszóróikonon. Ekkor egy *Play Controlnak* nevezett ablak jelenik meg. Indítsuk el ugyanezt még egy példányban. A két példány közül az egyiket *Option/Properties/Recording* rádiógombbal állítsuk át *Record Control*lá. A *Record Control*on választhatjuk ki a kívánt bemenetet (ahonnan a felvétel készül) (LineIn, Micr) és annak érzékenységét.



A *Play Control* panel használható egyrészt a kimeneti hangerő állítására (bal oldali potméter), másrészt meghatározhatjuk, hogy az egyes csatornák milyen súllyal vegyenek részt a kimeneti jelben (mixer funkció, jobb oldali három potméter).

“Cool Edit Pro” program ismertetése

A teljesség igénye nélkül, csak a mérés szempontjából lényeges funkciók kerülnek ismertetésre. A program kezeli a számítógépben található hangkártyát, ennek segítségével hanganyagok felvételére ill. lejátszására képes. A felvett anyagok fileba menthetők, hogy azokat egy későbbi feldolgozás, szerkesztés vagy netán egy egyszerű lejátszás céljából később ismét visszatölthessük.

A program fontosabb feldolgozó funkciói: hanganyag visszafordítása, amplitúdó torzítások, késleltetések (visszhang), különböző típusú szűrőzések, konvolúció, frekvencia spektrum analízis, valamint időbeni gyorsabb/lassabb torzítások.

A program elindítása: Win95 bootolása után kattintsunk a *Start /Programs/Hangfeld/Cool Edit Pro* ikonra.

Kilépés a programból *File/Exit* parancs.

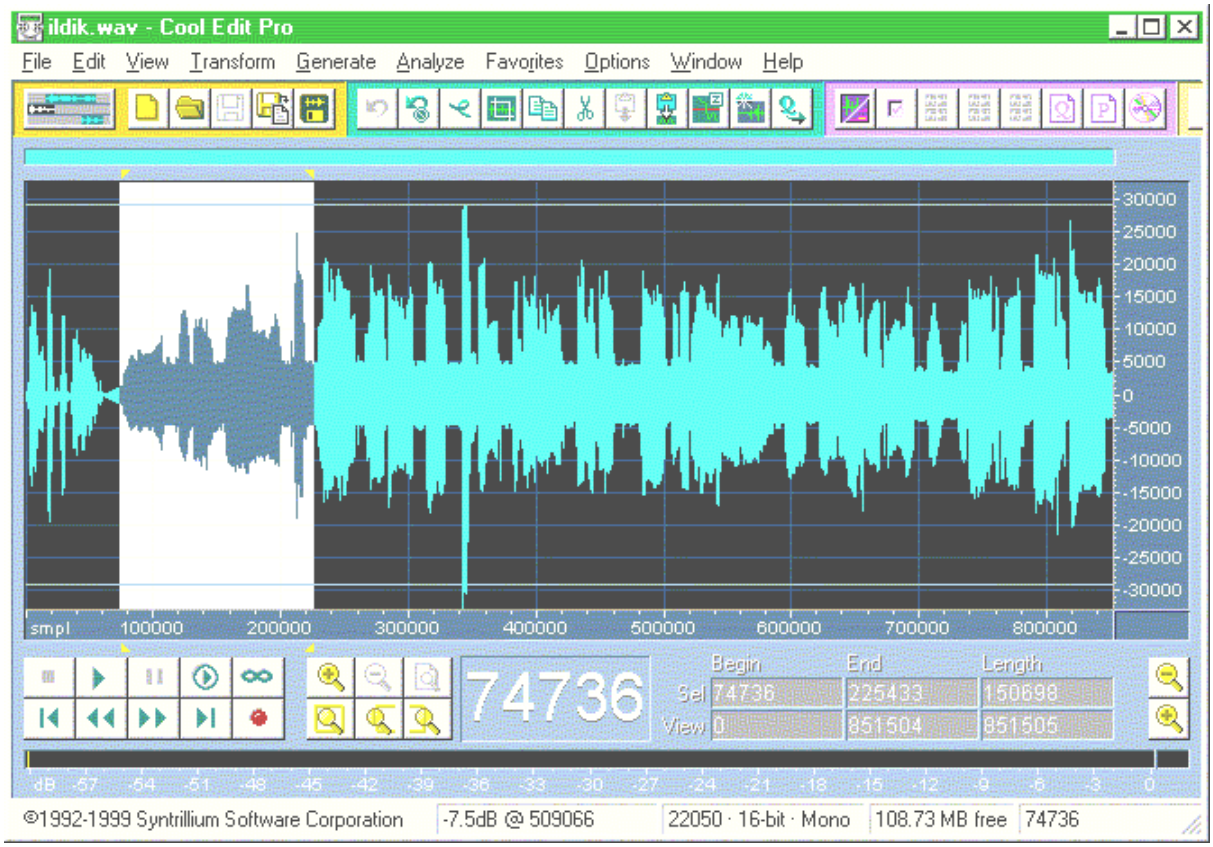
Korábban tárolt hangfile betöltése feldolgozás (szerkesztés) céljából: *File/Open* parancs.

Új hanganyag rögzítése előtt létre kell hozni egy üres filet (*File/New* parancs), melynek során meg kell adni a mintavételi frekvenciát, a csatorna számot mono/sztereo, ill. a digitalizálás felbontását 8/16/32 bit.

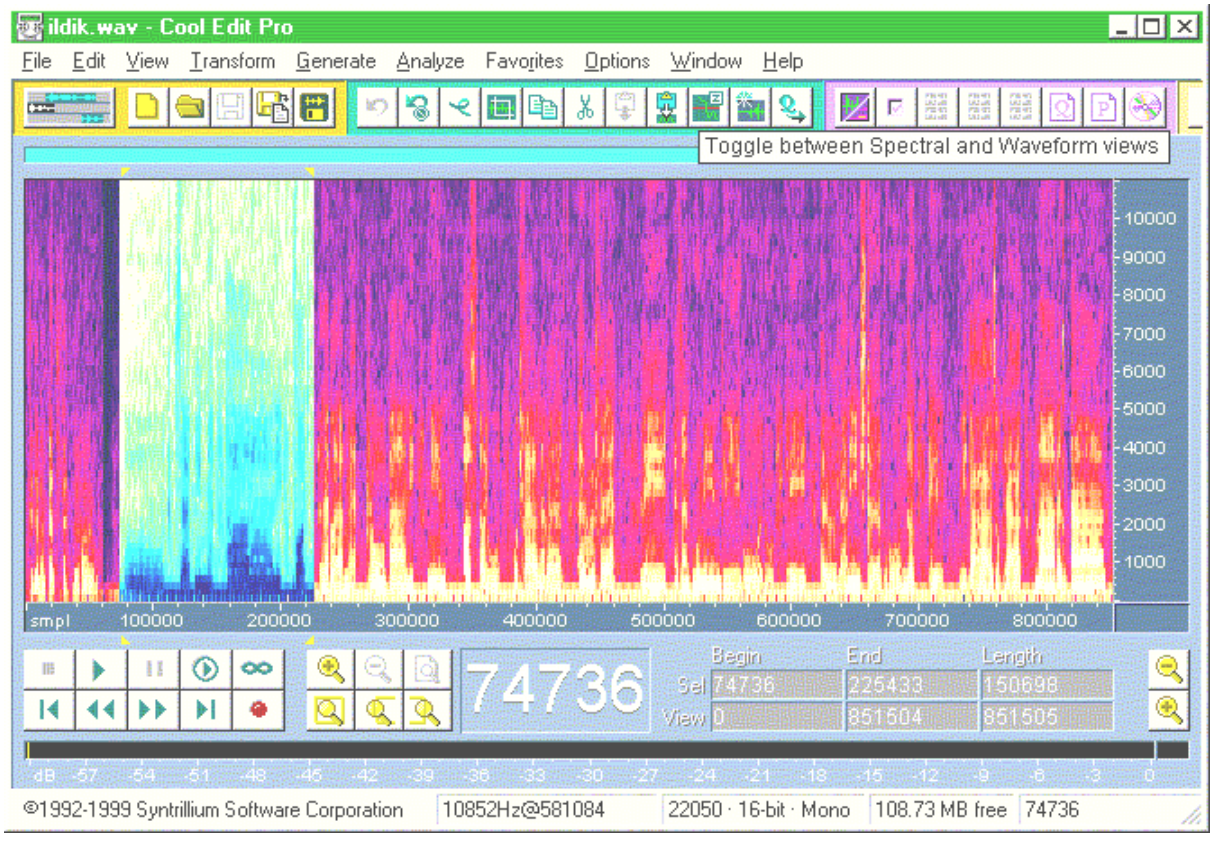
Hanganyag rögzítése: magnóknál megszokott gomb segítségével. A program bal alsó sarkában található ez a gombcsoport, egérkurzort föléjük helyezve sárga súgó tájékoztat funkciójukról.

A program főablakában a betöltött vagy újonnan felvett hanganyag jelalakját láthatjuk (*View/Waveform view*). Ez egy időtartománybeli nézet, az egyes minták értékeit egyenes szakaszokkal köti össze a program. A vízszintes tengelyen a minták sorszáma látható *View/Display Time Format/Samples*, vagy a korábban megadott mintavételi frekvencia segítségével a program idő értékekké skálázza át a tengelyt *View/Display Time Format/Decimal*. A +, – nagyítók segítségével az időskála tetszőleges mértékben zoomolható, egészen addig a mértékig, hogy az egyes minták jól elkülönülve látszanak. Ekkor a minták egérrel megfoghatók és értékük húzogatással változtatható. Kétszeri egérkattintásra a minta értékét dialógusablakban leolvashatjuk, vagy állíthatjuk.

A jobboldali függőleges tengelyen a minták értékei láthatók (például: egy előjeles 16 bites szám –32768 és +32767 között vehet fel értéket). Ezt a skálát is zoomolhatjuk a jobboldali + és – nagyítók segítségével.



A hanganyag spektrogramját tanulmányozhatjuk a főablakban, ha átkapcsolunk a *View/Spectral View* nézetre. Figyelem: a spektrogram egy kétváltozós függvény, és nem tévesztendő össze a spektrummal, amely egyváltozós. A vízszintes tengely skálája továbbra is mintaszám (vagy idő), a függőlegesé frekvencia, míg a spektrogram értékét a főablak adott pontjának színmélysége adja.



A betöltött vagy újonnan felvett hanganyag tetszőleges tartománya egér segítségével kijelölhető. Az ezután ismertetésre kerülő feldolgozó funkciók erre a kijelölt tartományra vonatkoznak.

Transform/Invert – Funkció, –1-el szorozza a kijelölt minták értékeit.

Transform/Reverse – A kijelölt tartomány mintáit időben megfordítja.

Transform/Amplitude/Amplify/Fade – Időben lineárisan (logaritmikusan) növekvő/csökkenő módon állítja az amplitúdót. Zeneszámok bevezető ill. záró részeit szokták ezzel a funkcióval kezelni.

Transform/Amplitude/Dynamics Processing/Graphic – A hanganyag dinamikája állítható ezzel a módszerrel, amely tulajdonképpen az amplitúdó-átvitel transzfer függvényét módosítja. Elérhetjük ezáltal pl. hogy a kisebb amplitúdójú (halkabb) hangokat jobban erősítsük, mint a hangosabbakat. Ezáltal dinamika kompressziót érünk el. Az átviteltechnikában gyakran használt módszer, hogy az adóoldalon dinamika kompressziót, míg a vevőoldalon dinamika expanziót alkalmaznak úgy, hogy a két transzfer karakterisztika szorzata 1 legyen. Ezáltal a jelátvitel torzításmentes, viszont az átviteli csatornában a halkabb hangok is magasabb amplitúdóval utaznak, ezáltal sokkal érzéketlenebbek a zajokkal szemben.

Transform/Amplitude/Envelope – Egy burkológörbe karakterisztikát állíthatunk be, amellyel a kijelölt hanganyagot szorozza.

Transform/Amplitude/Normalize – A kijelölt mintákat konstans értékkel szorozza, ezenkívül alkalmas pl. DC szint kiküszöbölésére (vagy netán hozzáadására), valamint sztereó felvétel esetén azonos csatorna szintek beállítására.

Transform/Amplitude/Pan Expand – Csak sztereó esetén működik ez a funkció, a balansz értéke időben állítható. Pl. mono vonatzakatolást másolunk egy üres sztereó file mindkét csatornájára, majd alkalmazzuk ezt a funkciót úgy, hogy a vonat “átmenjen” egyik oldalról a másikra.

Transform/Delay Effects/Delay – A kijelölt hanganyagot adott idővel késlelteti, és meghatározott súlyfaktorral összegzi az eredetivel. Kiválóan alkalmas mesterséges visszhangosítás előállítására.

Transform/Delay Effects/Multitap Delay – Sokszorosan összetett visszhangeffektusok generálására alkalmas.

Transform/Filters/FFT Filter – Grafikusan megadott frekvenciakarakterisztikával torzítja a hanganyag spektrumát.

Transform/Filters/Scientific Filters – Bessel, Butterworth és Csebisev típusú szűrők közül válogathatunk. Mindhárom típus lehet alul-, felül-, sáváteresztő ill. sávzáró. Megadható a szűrők alsó/felső vágófrekvenciáinak értéke, ill. a digitális szűrők fokszáma.

Transform/Special/Convolution – Előzőleg kijelölt hanganyag megadható a konvolúciós műveletnek, mint konvolváló “impulzus”. Ezután kell kijelölni a konvolválandó hanganyagot és elvégezni magát a konvolúciót.

Analyze/Frequency Analysis – A kijelölt tartomány frekvenciaspektrumát számolja. Állítható az FFT méret, ezáltal a frekvenciatengely felbontása.

Analyze/Statistics – Pl. hisztogramot számol az egyes amplitúdók előfordulási gyakoriságáról, de leolvashatjuk a min. ill. max. minta értékét, a felvétel offsetjét (DC szintjét), valamint teljesítményét.

A fenti ismertetésből kitűnik, hogy bár a CoolEdit Pro program igen sokrétű, hasznos feldolgozó funkciókkal rendelkezik, hiányzik az a lehetőség, hogy a felhasználó – pl. valamilyen matematikai függvény szerint – tetszőleges hullámformát generáljon. Ezt a hiányosságot pótolta a mérés szerzője egy saját programmal, ennek neve: WaveGen.exe.

A “WaveGen” program ismertetése

Tetszőleges hullámforma előállításán (elsődleges ok, amiért készült) túl, a program alkalmas meglévő hanganyag mintáit valamilyen – a felhasználó által definiált – szabály szerint módosítani. Sőt, a kezdeti sikereken felbuzdulva jelenleg hanganyag felvételére és lejátszására is alkalmas a program.

A felhasználó mindezen szabályokat egy Delphi-Pascal nyelven írt “programocska” (script) lefuttatásával érvényesíttetheti. Ezek a scriptek helyben szerkeszthetők, fileba menthetők, ill. onnan visszatölthetők.

A scriptek szerkesztését nagymértékben megkönnyíti, ha az **egér jobb gombjának** hatására lebomló helyi menüből választunk utasítást vagy eljárást (amelyet a szövegkurzor pozíciójába illeszt be).

A scriptek írásának szabályai:

Tulajdonképpen egy Pascal programocskáról van szó, amelynek

Szerkezete:

var *helyi változók deklarálása* pl: **var** valt1:tip1; valt2,valt3:tip2;

begin begin

programtörzs

end; end;

Felhasználhatók egyszerű és összetett változó típusok (a teljesség igénye nélkül):

- 8, 16, 32 bites egészek: **byte, word, dword** és ezek előjeles megfelelői: **shortint, smallint, longint,**

- valósak: **real, single, double, extended**,

- tömb (pl: **array[0..9] of byte;**), és

- rekord (pl: **record**

a:byte;

b:word;

c:single;

end;)

Műveleti jelek:

+, -, *, /, mod, div, shr, shl, not, or, and, xor, :=, =, <>, <, >, <=, >=

(pl: osztásnál / eredménye valós, míg div-é egész, mod az osztási maradékot adja. Not, or, and, xor mind egészekre, mind pedig logikai változókra alkalmazható. **Figyelem** az **=** és **:=** használatát illetően. Az első logikai egyenlőség vizsgálatot, míg az utóbbi értékadást jelent.)

Utasítások:

feltételes: **if** log_felt **then** ezt_hajtja_végre_ha_igaz;

if log_felt **then** ezt_hajtja_végre_ha_igaz **else** és_ezt_ha_hamis;

Ha a **then**, ill. **else** után több utasítást kívánunk használni azokat **begin end;** közé kell zárni. Figyelem az **else** előtt nem lehet **;**

ciklus-utasítások: **for** valt:=ert1 **to** (vagy **downto**) ert2 **do begin**

ciklusmag; // rendre lefut, miközben valt ert1-től ert2-ig egyesével nő (csökken)

end;

while log_felt **do begin**

ciklusmag; // ismétlődik: amíg log_felt igaz

end;

repeat

ciklusmag; // ismétlődik: amíg log_felt igazzá nem válik

until log_felt**;**

Különösen az utolsó kettőnél fokozottan figyeljünk a végtelen ciklusok elkerülése érdekében!

Előre definiált változók és eljárások (ezek a változók és eljárások deklaráció nélkül szabadon felhasználhatók)!

pi = 3.141592654...

fmv:dword, **ch**:byte, **bpch**:byte, **Nmax**:dword, csak olvasható változók. fmv a mintavételi frekvencia Hz-ben, ch =1 mono felvétel esetén, =2 sztereónál, bpch =1 ha a minták 8 bitesek, és =2 ha 16 bitesek. Nmax azt fejezi ki, hogy csatornánként hány mintával van dolgunk. Mivel ezek **csak olvashatók, értékeik megváltoztatására csak közvetett módon** nyílik lehetőség (*NewWave, PasteFromClipboard és LoadFromFile* eljárások által). A változók értékeit minden egyes futás után láthatjuk a program státus sorában.

L[i], R[i] tömbökkel a bal ill. jobb csatorna mintáinak értéke érhető el egyenként. **i 0 és Nmax-1 között** vehet fel értéket. **L[i], R[i] :byte** (előjel nélküli) típusúak 8 bites, és **:smallint** (előjeles) 16 bites felvétel esetén. **Mono** felvételnél csak az **L[i]** tömb használatának van értelme.

NewWave(*mintaveteli_freki***,** *csat_szam***,** *byte_per_csat***,** *hany_minta hosszu_legyen***);** eljárás egy új üres hullámforma helyét foglalja le a memóriában, miközben felépít egy szabványos **wav** fejlécet, és írja az **fmv, ch, bpch, Nmax** változókat. Az eljárás ismételt meghívása felülírja az előzőleg lefoglalt memóriaterületet.

CopyToClipboard; A memóriában található hullámforma tartalmát a Windows vágólapjára helyezi, így az más – wav formátumot – kezelni tudó programok számára is elérhetővé válik.

PasteFromClipboard; Ha a Windows vágólapján wav formátumú adat szerepel (ellenkező esetben nem történik semmi), azt bemásolja a memória-beli hullámforma helyére. Írja az **fmv, ch, bpch, Nmax** változókat.

SaveToFile('*file_nev***');** A memóriában található hullámforma tartalmát fileba írja.

LoadFromFile('*file_nev***');** Fileból tölti fel a memória-beli hullámforma helyét. **Figyelem:** a filenak **valóban wav** filenak kell lenni. Írja az **fmv, ch, bpch, Nmax** változókat.

Play; A memóriában található hullámforma tartalmát elejétől végig lejátsza. Idő előtti leállítása a Stop Playing menüvel lehetséges.

Rec; A memória-beli hullámforma helyét hangfelvétel útján tölti fel. Ez az eljárás helyet nem foglal, ezt tehát nekünk kell megtennünk egy korábbi NewWave, PasteFromClipboard vagy LoadFromFile valamelyikével. A felvétel addig tart, amíg az előzőleg lefoglalt hely be nem töltődik, vagy a Stop Recording menüvel azt erőszakosan meg nem szakítjuk.

További előre definiált függvények: (funkciójuk annyira egyértelmű, hogy csak felsorolás szintjén közöljük)

Round(x: Extended**)**: Longint**; Sqrt(**x: Extended**)**: Extended**;**

Sqr(x: Extended**)**: Extended**; Power(**base, exponent: Extended**)**: Extended**;**

Ln(x: Real**)**: Real**; LogN(**base, X: extended**)**: Extended**;**

Exp(x: Real**)**: Real**;**

Sin(x: extended**)**: extended**; Cos(**x: Extended**)**: Extended**;**

Tan(x: Extended**)**: Extended**; Cotan(**x: Extended**)**: Extended**;**

ArcSin(x: extended**)**: extended**; ArcCos(**x: extended**)**: extended**;**

DegToRad(degrees: Extended**)**: Extended**; RadToDeg(**radians: Extended**)**: Extended**;**

Magyarázó szövegek (comment-ek) elhelyezése a scriptben: *//* (két slash) **jeltől sorvégig** található szöveget figyelmen kivűl hagyja a fordító. Több sorra kiterjedő magyarázó szöveget **{}** (kapcsoszárójelek) **között** kell elhelyeznünk.

Végezetül lássunk egy példa scriptet:

```
var i :integer;

a, b :smallint;

t, m, fmod, fvivo :single;

begin

m:=0.8; fmod:=0.5; fvivo:=600;

PasteFromClipboard;

for i:=0 to Nmax-1 do begin

t:=i/fmv;

a:=L[i]; b:=R[i];

L[i]:=Round(a*(1+0.5*m*sin(2*pi*fmod*t)));

R[i]:=Round((1+0.5*m*b)*sin(2*pi*fvivo*t));

end;

Play;
```

end;

A futtatás előtt vágólapra kell másolni a módosítandó hanganyagot. Ezt a hanganyagot fogja a program a PasteFromClipboard paranccsal levenni és memóriába másolni. Eközben a hanganyagnak megfelelően beállítja az fmv, ch, bpch, Nmax változókat. Ezután egy for ciklus következik, ahol a futó változó 0-tól Nmax-1-ig vesz fel értéket (vagyis a teljes hanganyag mintáit egyenként elérhetővé teszi a ciklusmagban). Az i változót 32 bites egészként deklaráltuk, hiszen Nmax is ilyen. Az a, b változókban ideiglenesen tároljuk a bal ill. jobb csatorna aktuális mintáinak értékét. Mivel az L[i], R[i] értékek smallint-ek, ezért az a, b változóknak is ilyeneknek kell lenniük. t, m, fmod, fvivo változókat csak kényelmi ill. stilisztikai szempontból deklaráltuk, így lényegesen átláthatóbb az L[i], R[i] függvény. Mivel az i mintánként fut végig, és két minta közti időkülönbség 1/fmv Þ t (= i / fmv) a futó idő. A bal csatorna új értékét úgy képezzük, hogy a régi értéket (a) moduláljuk fmod frekvenciájú szinuszos hullámmal, m modulációs mélységben. A jobb csatorna esetén, magát a csatorna jelét (b) tekintjük moduláló jelnek, amely fvivo frekvenciájú szinuszos vivőt modulál. Említést érdemel a Round függvény, amely valósról egészre konvertál. Erre azért van szükség, mert a sin függvény valós értékkel tér vissza, L[i] pedig egészet kíván. Miután az összes mintát kezeltük (for ciklus end-je után), a Play paranccsal lejátszuk művünket. A programocska funkciójának hasznossága vitatható, viszont maga a megoldás jó kiindulópont lehet a scriptek írásával most ismerkedő hallgatónak. A WaveGen könyvtárában még további példa scriptek találhatók.