

## Aplicații Labview pentru prelucrarea sunetelor prin placa de sunet

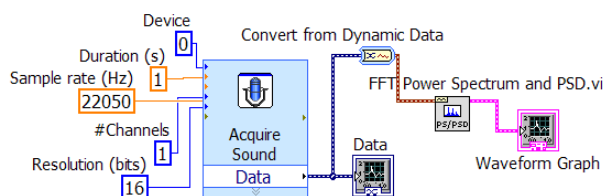
L12, Programare I, dec. 2020, Prof. I. Lupea

### 1. Achiziție de sunet pe durată finită

Este apelată funcția **Acquire Sound** plasată în paleta Sound. Prin această aplicație este achiziționat semnal de la microfon fiind folosită placa de sunet a calculatorului. Sunt stabiliți parametri de achiziție prin controale: dispozitivul (placa de sunet), rata de eșantionare (22050 eșantioane/ secundă), numărul de canale (un canal sau două canale), rezoluția unui eșantion (16 biți) și durata în secunde a achiziției. Durata achiziției este stabilită/ cunoscută de la apelul funcției Acquire Sound.

Funcția returnează semnalul achiziționat în format dinamic.

Prin funcția Convert from Dynamic Data (paleta *Express/ Signal manipulation*) se realizează conversia de la Dynamic Data la tablou de forme de undă. O funcție utilă este de asemenea *Get Dynamic Data Attributes*.

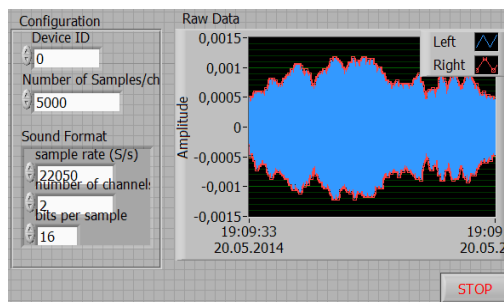


### 2. Achiziție continuă de sunet și puterea spectrală (pe durată nedefinită)

2.1. Sunt apelate trei funcții prin care este gestionată achiziția de sunet.

Funcția *Sound Input Configure.vi* realizează configurarea unei sarcini de achiziție sunet. Sound Format este o structură cu trei câmpuri. Câmpul *sample rate* (S/s) stabilește numărul de eșantioane achiziționate pe secundă; acestea se depun în bufferul intermediar de unde sunt citite de *SI Read.vi*. Se alege achiziție continuă selectând Continuous Samples. Controlul Number of Samples/ch stabilește mărimea bufferului intermediar de memorie unde se depun temporar eșantioanele provenite de la convertorul analogic digital.

Funcția *Sound Input Read.vi* realizează citirea repetată din bufferul de memorie, în cadrul ciclului While, a unui număr de eșantioane (stabilit prin Number of Samples/ch). Funcția returnează un tablou de două forme de undă (1D array of waveform), câte o formă de undă pentru fiecare canal de achiziție.

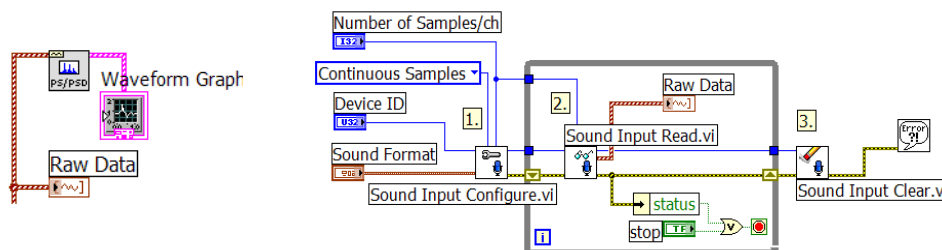


Funcția *Sound Input Clear.vi* dealocă memoria folosită pentru sarcina de achiziție configurată prin *SI Configure.vi*.

2.2. La **o ciclare** sunt achiziționate și afișate 5000 eșantioane. Spațierea între eșantioane este  $1/\text{Sample rate}$  (secunde) iar durata formei de undă este  $5000 \cdot 1/\text{Sample rate}$  (secunde). Semnalul de la fiecare canal este trasat în indicatorul grafic Waveform Graph (Raw Data) cu altă culoare (albastru, roșu).

2.3. Ciclul se **oprește în două situații**: a) apare eroare la execuția unei funcții (semnalată prin câmpul  $\text{status}=\text{T}$ ) sau b) se apasă controlul STOP.

2.4. Se adaugă **calculul puterii spectrale PS** a semnalului prin funcția FFT Power Spectrum and PSD.vi. Asociat funcției selectăm fereastra Hanning pentru pregătirea semnalului înainte de aplicarea transformatei Fourier (FFT.vi) și controlul *Averaging parameters*.

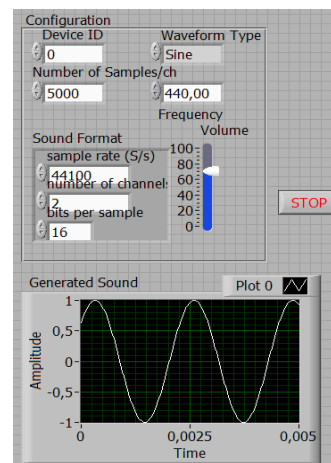


Se vor măsura pe indicatorul grafic (folosind un cursor) frecvențele unor vârfuli ale puterii spectrale obținute la lovirea diapazonului, a unui pahar de sticlă, bătutul din palme, ciupirea unor coarde ale instrumentelor muzicale, lovirea unor lamele ale xilofonului. Pentru reținerea vârfurilor puterii spectrale (în vederea măsurării frecvenței cu cursorul) se va selecta Peak hold pentru Averaging mode în cadrul controlului *Averaging parameters* al funcției FFT Power Spectrum and PSD.vi.

### 3. Generare sunet (ton) în mod continuu

3.1. Sunt apelate patru funcții prin care este gestionată generarea de sunet prin placa de sunet a calculatorului.

*Funcția SO Configure.vi* (configurarea plăcii de sunet) stabilește prin controlul Sound Format rata de trimitere a eșantioanelor la placa de sunet (44100eșantioane/ secundă), numărul de eșantioane trimise pe fiecare canal într-o tranșă (valoare care stabilește și mărimea bufferului intermediar de memorie), modul continuu de generare sunet (Continuous Samples). Este returnat un număr de identificare (task ID) a sarcinii de generare de sunet, număr care este trimis și la următoarele funcții care gestionează sarcina. Funcția *SO Write.vi* apelată repetitiv trimite efectiv tranșa de eșantioane (5000 în exemplu), date de tip tablou de forme de undă, la convertorul digital analogic al plăcii de sunet.

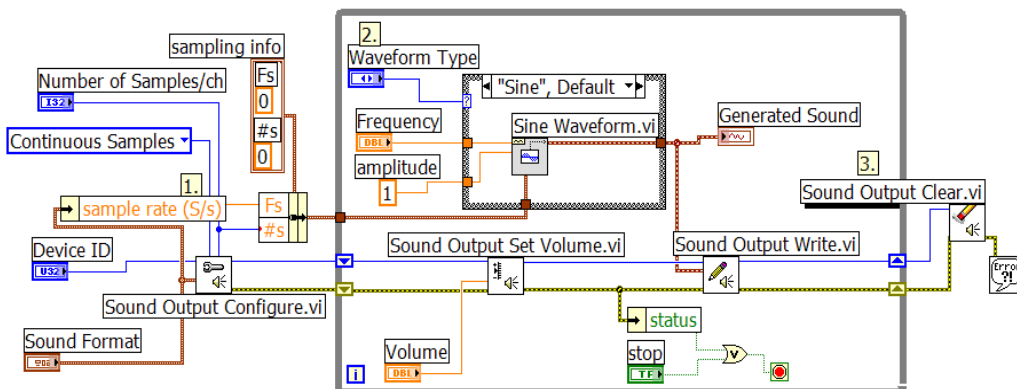
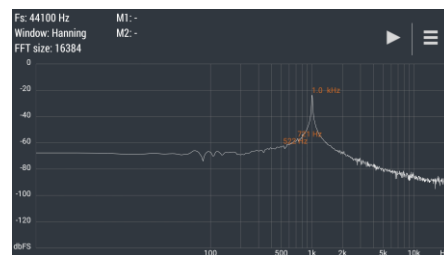


Funcția *SO Set Volume* este apelată repetitiv permite modificarea volumului (amplitudinea) sunetului generat.

Funcția *SO Clear.vi* este apelată după oprirea ciclului While realizând ștergerea sarcinii de generare de sunet.

3.2. Forma de undă primită de *SO Write.vi* este generată de funcția *Sine Waveform.vi*. Aceasta se va apela repetitiv astfel încât cele 44100 eșantioane/secundă (Sample rate) să fie asigurate. Se corelează astfel controlul *sampling info* cu Sound Format asigurând frecvența dorită a formei de undă sinusoidale și astfel frecvența tonului generat de placa de sunet. Structura *sampling info* a funcției *Sine Waveform* inițial (0,0) și se modifică la (44100, 5000). Reset signal se pune pe False pentru ca secvențele succesive de sinus să fie în continuare și iar tonul generat să nu fie cu întreruperi.

3.3. În timpul rulării aplicației modificăți frecvența semnalului sinusoidal. Măsurăți frecvența sunetului generat prin placa de sunet cu aplicația *Advanced Spectrum Analyzer PRO* de pe mobil.

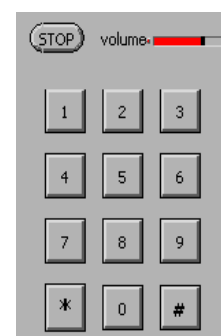


3.4. Instrucțiunea CASE selectează tipul semnalului de ieșire la frecvența=440Hz, amplitudinea=1. Se vor putea selecta și alte forme de undă prin instrucțiunea Case.

În graficul *Generated Sound* din panoul frontal se observă semnalul de trimis la *SO Write.vi*, durata și frecvența 'purată' de sinusoidală.

#### 4. Simularea sunetului asociat tastelor telefonului

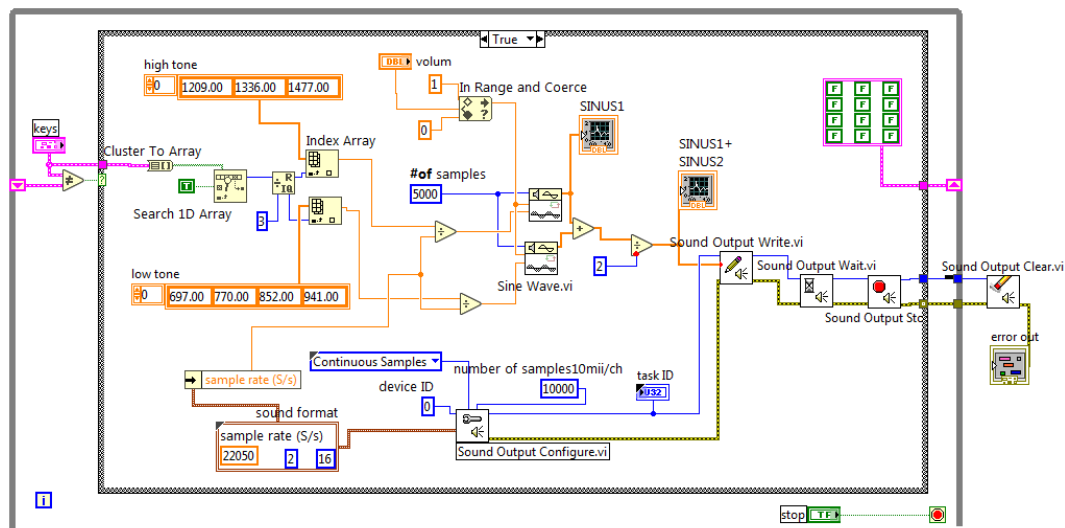
Aplicația generează câte un sunet la apăsarea unei taste din matricea de 4x3 butoane. Se rămâne în buclă While până la apăsarea butonului de "STOP".



La fiecare iterație, dacă se intră pe cazul True, registrul de transfer este inițializat cu structura de valori logice 3x4 False. La apăsarea unei taste structura keys va conține un câmp True; rezultă diferență între control și constanta structură (cluster) de 12 valori False. Se execută prin urmare cazul True. *Search ID Array.vi* identifică numărul de ordine a tastei apăsate (T) în tabloul boolean.

Împărțirea cu rest determină linia și coloana tastei apăsate. Prin *Index Array.vi* se extrage o valoare din constanta **sir frecvențe înalte** și altă valoare din **sir frecvențe joase**. Cu aceste valori se pregătesc intrările 'frequency' ale semnalelor sinusoidale generate de funcții *Sine Wave.vi*. Semnalele adunate vor genera sunetul asociat fiecărei taste.

**Sound Output Write.vi** generează sunetul dat de suma celor două tonuri, sunet specific tastei. Slide-ul **volume** modifică amplitudinea ambelor sinusoidale. Funcția *In range and Coerce* limitează volumul între 0 și 1.



Operatorul *diferit* este setat să compare cele două structuri ca ansamblu (rezultă o singură valoare T sau F) și nu pe componente. O variabilă locală furnizează *task ID* ul din cazul False pentru *SO Clear.vi*. Cazul False este traversat de firul structurii (cluster) cu 12 taste.

