ლექცია #12

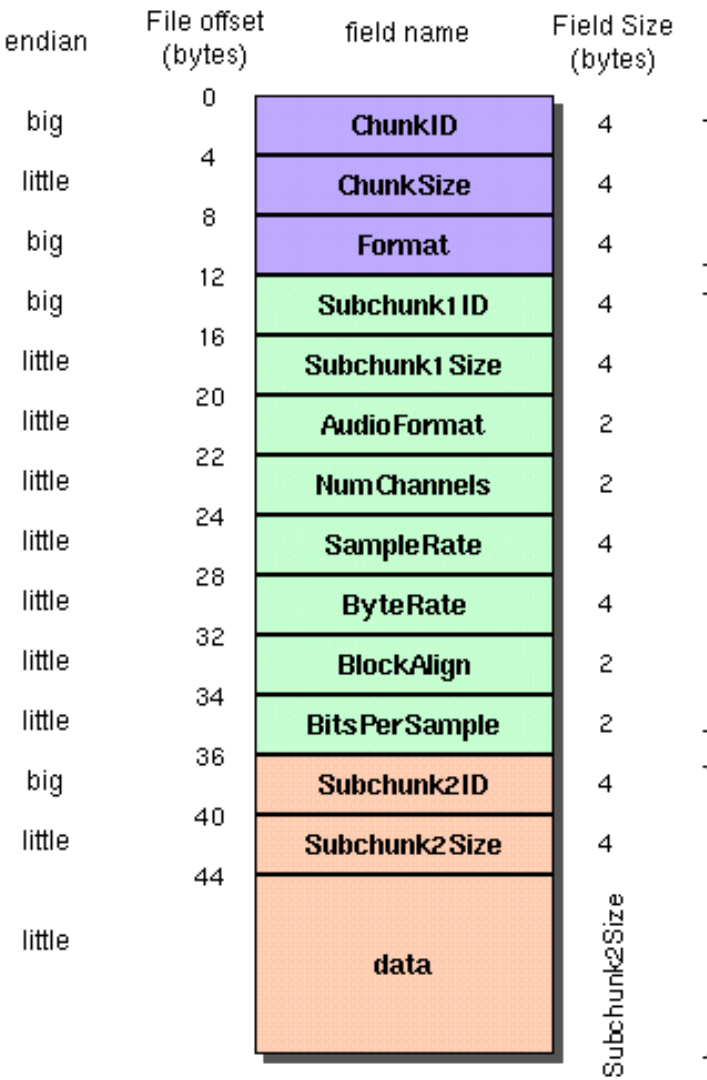
I/O ოპერაციები .wav ფაილის მაგალითზე

.wav ფაილი

Microsoft WAVE (გამოიყენება 1991 წლიდან):

- კონტეინერის ტიპი - RIFF (Resource Interchange File Format)
- RIFF კონტეინერი შედგება ძირითადი ნაჭერის (chunk) და ქვენაჭრებისაგან (sub-chunks)
- აუდიო მონაცემები შეიძლება იყოს შეკუმშულიც და შეუკუმშავიც
- ყველაზე გავრცელებულია შეუკუმშავი აუდიო LPCM (Linear Pulse Code Modulation) ფორმატში
- იგივე LPCM ფორმატი გამოიყენება აუდიო კომპაქტ. დისკებშიც
- .wav ფაილების გამოყენება ხდება არა აუდიო მიზნებითაც (მაგ. LTspice-ში ძაბვის და დენის გრაფიკების შესანახად)

.wav ფაილის სტრუქტურა

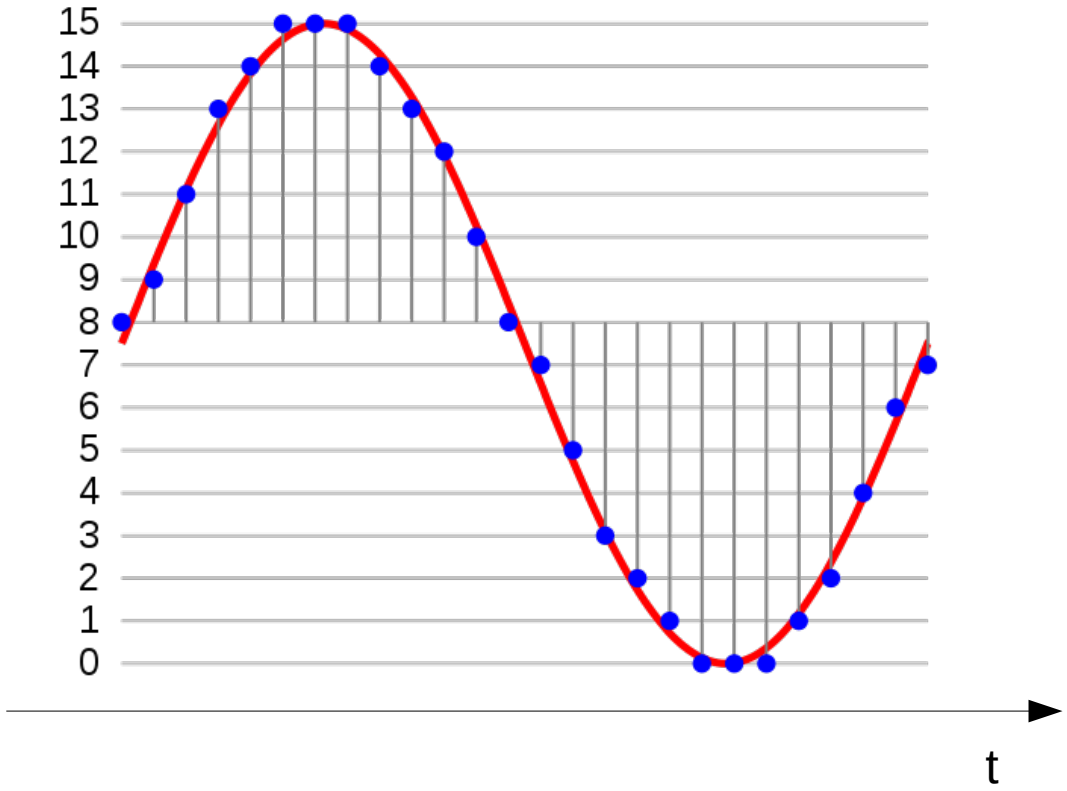



```
0000000000 52 49 46 46 RIFF
0000000004 24 B4 2D 00 $´-.
0000000008 57 41 56 45 WAVE
0000000012 66 6D 74 20 fmt
0000000016 10 00 00 00 ....
0000000020 01 00 02 00 ....
0000000024 44 AC 00 00 D¬..
0000000028 10 B1 02 00 .±..
0000000032 04 00 10 00 ....
0000000036 64 61 74 61 data
0000000040 00 B4 2D 00 .´-.
0000000044 00 00 00 00 ....
0000000048 00 00 00 00 ....
0000000052 00 00 00 00 ....
0000000056 00 00 00 00 ....
0000000060 00 00 00 00 ....
0000000064 00 00 00 00 ....
```

.wav ფაილის ფორმატი

LPCM ფორმატი:

- დაყოფის (sampling) სიხშირე: 8, 11.025, 22.050, 44.1, 48, 96, 192, 384 კჰც
- ვერტიკალური გარჩევისუნარიანობა: 8, 16, 20, 24, 32 ბიტი

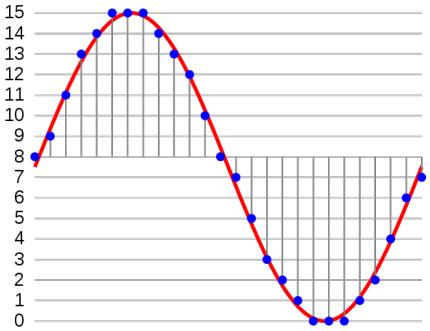


.wav ფაილის ფორმატი

LPCM ფორმატი:

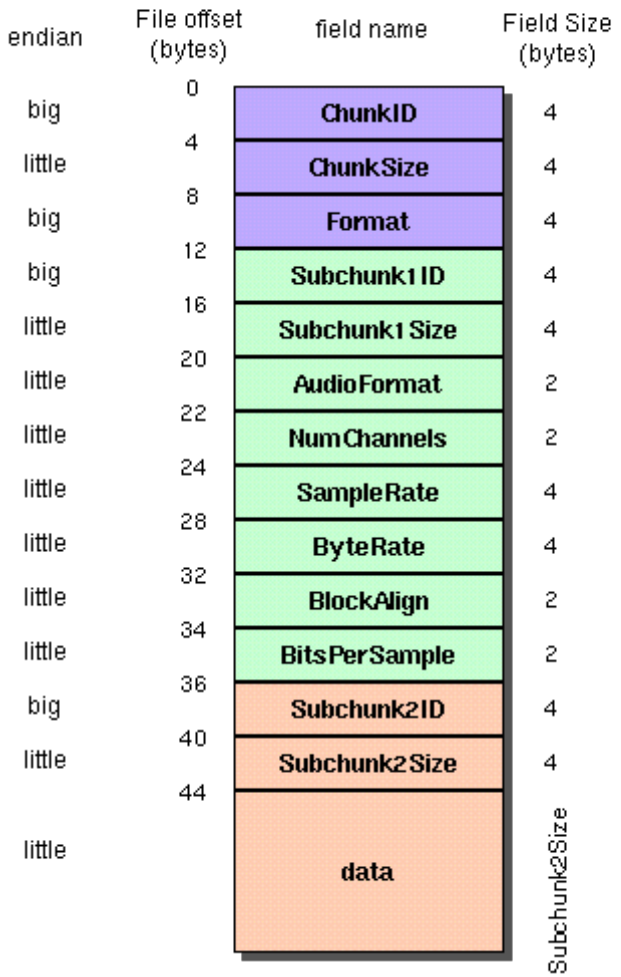
- დაყოფის (sampling) სიხშირე: 8, 11.025, 22.050, 44.1, 48, 96, 192 კჰც
- ვერტიკალური გარჩევისუნარიანობა: 8, 16, 20, 24, 32 ბიტი

ვერტ. ბიტები N	კოდი (მინ. მაქს.) 2^N	1 ბიტის შესაბამისი შეცდომა (-1ვ...+1 ვ)	მაქს. დინამიური ინტერვალი (დბ)
8	(-128, 127)	8 მვ	42
16	(-32768, 32767)	30 მკვ	90
24	(-8388608, 8388607)	0.1 მკვ	138



.wav ფაილის წაკითხვა

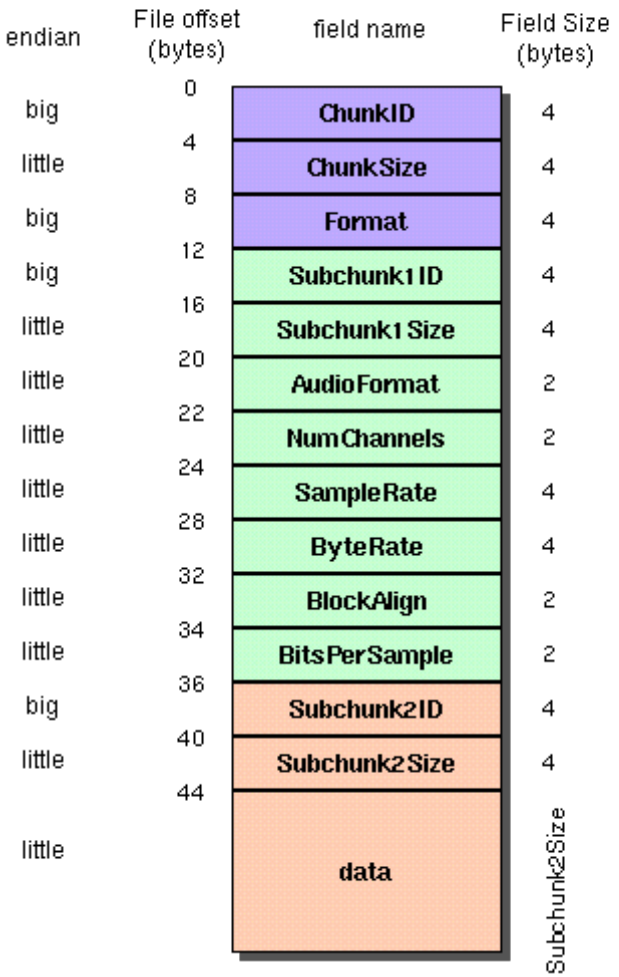
სათაურის წაკითხვა




```
ifstream inFile("rec.wav", ios::binary);

char name[4];
inFile.read(name, 4);
if (!strstr(name,"RIFF")) return -1;

int ChunkSize;
inFile.read((char*)&ChunkSize, 4);

char Format[4];
inFile.read(Format, 4);
if (!strstr(Format,"WAVE")) return -2;

char Subchunk1ID[4];
inFile.read((char*)&Subchunk1ID, 4);
if (!strstr(Subchunk1ID,"fmt ")) return -3;

int Subchunk1Size;
inFile.read((char*)&Subchunk1Size, 4);

short AudioFormat;
inFile.read((char*)&AudioFormat, 2);

short NumOfChannels;
inFile.read((char*)&NumOfChannels, 2);

Int_t SampleRate;
inFile.read((char*)&SampleRate, 4);

int ByteRate;
inFile.read((char*)&ByteRate, 4);
```

.wav ფაილის წაკითხვა

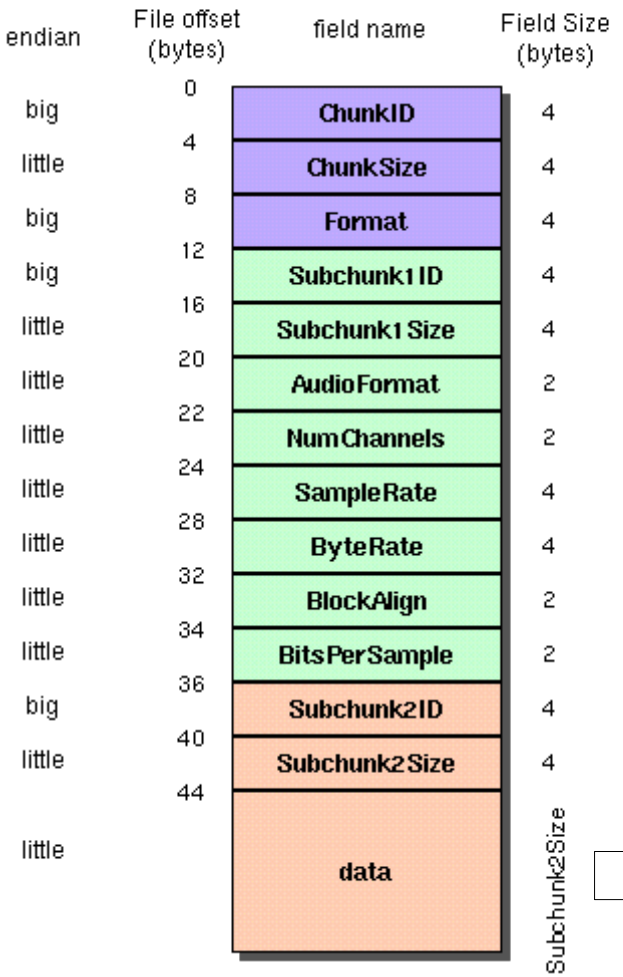
სათაურის წაკითხვა




```
short BlockAlign;
inFile.read((char*)&BlockAlign, 2);

short BitsPerSample;
inFile.read((char*)&BitsPerSample, 2);

char Subchunk2ID[4];
inFile.read(Subchunk2ID, 4);
if (!strstr(Subchunk2ID,"data")) return -4;

int Subchunk2Size;
inFile.read((char*)&Subchunk2Size, 4);
int NumOfSamples = Subchunk2Size*8/BitsPerSample/NumOfChannels;

cout << "NumOfChannels = " << NumOfChannels << endl;
cout << "SampleRate = " << SampleRate << " Hz" << endl;
cout << "BitsPerSample = " << BitsPerSample << endl;
cout << "Number of Samples = " << NumOfSamples << endl;

float **Channel;
float *Time;

try {
  Time   = new float [NumOfSamples];
  Channel = new float* [NumOfChannels];
  for (int k=0; k<NumOfChannels; k++) Channel[k] = new float[NumOfSamples];
}
catch (bad_alloc&) {
  cout << "Error allocating memory: No enough free memory!" << endl;
  return -5;
}
```

.wav ფაილის წაკითხვა

სიგნალის წაკითხვა




```
int data;
int MaxRange = pow(2,BitsPerSample);
int MaxPositive = MaxRange/2 - 1;
int SSize = BitsPerSample/8;
bool Is2sComplement = BitsPerSample>=16;

inFile.seekg(44, ios::beg);

for (int k=0; k<NumOfSamples; k++) {
  for (int i=0; i<NumOfChannels; i++) {
    data = 0;
    inFile.read((char*)&data, SSize);
    if (Is2sComplement) {
      if (data>MaxPositive) Channel[i][k] = data - MaxRange;
      else Channel[i][k] = data;
    }
    else Channel[i][k] = data - MaxRange/2;
  }

  Time[k] = 1./SampleRate*k;
}
```

Sample 1		Sample 2		Sample 3		...
L	R	L	R	L	R	

პრაქტიკული გამოყენების მაგალითი

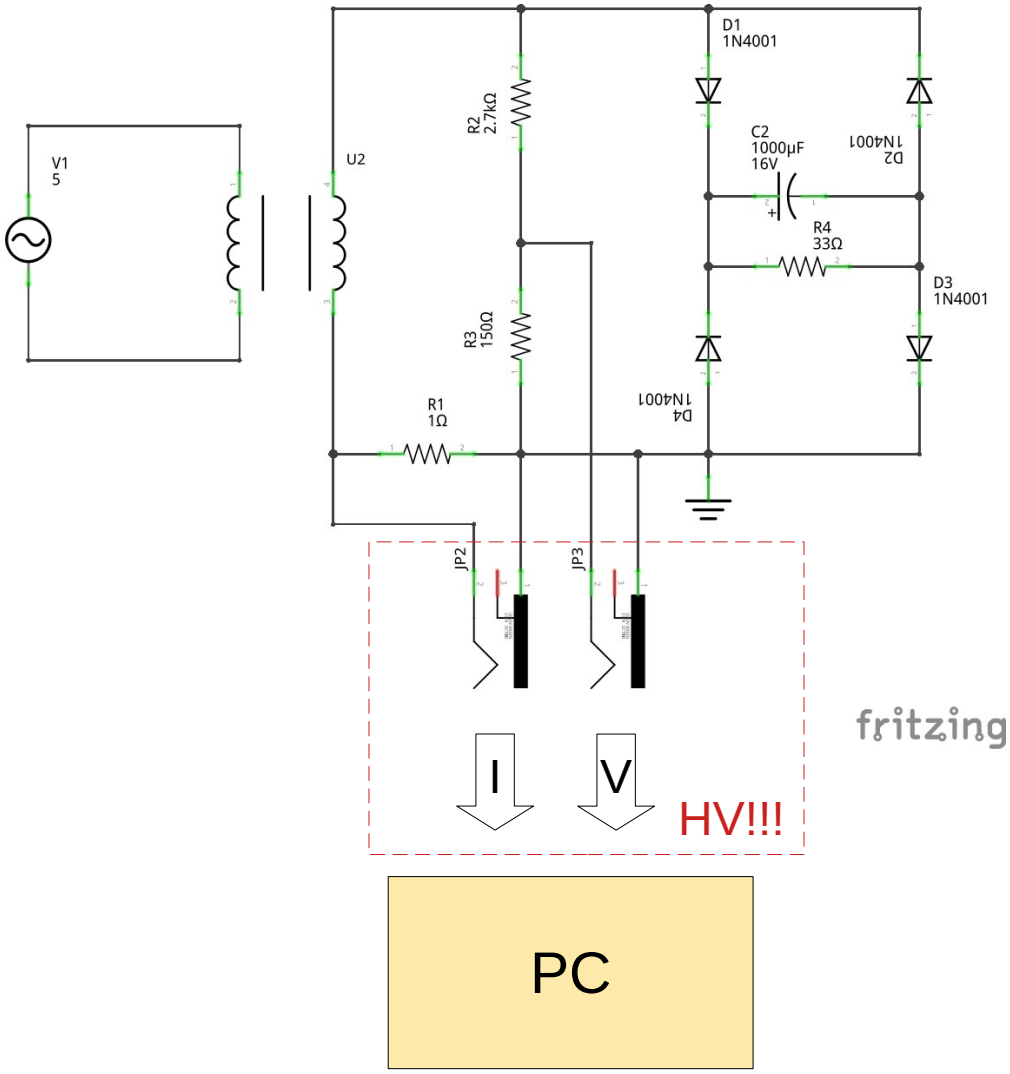
გავზომოთ ტრანსფორმატორის სიმძლავრის კოეფიციენტი სრულ პერიოდიანი გამმართველით დატვირთვისას.

$$\cos(\varphi) = \frac{P_{Real}}{V_{RMS} I_{RMS}}$$

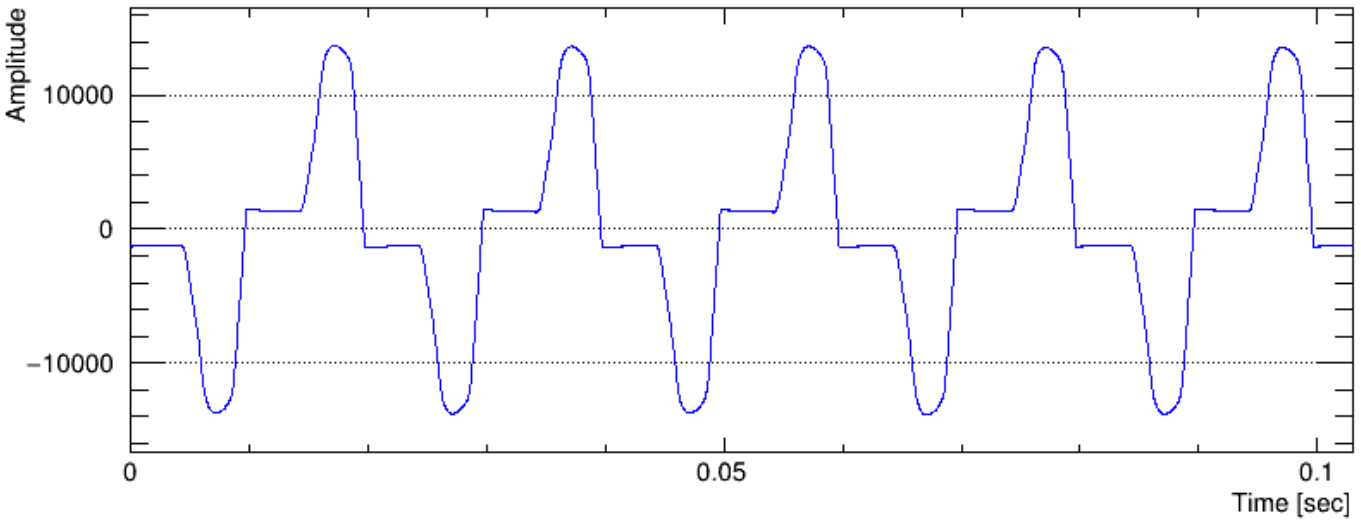
$$P_{Real} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} V(t) I(t) dt \approx \frac{\Delta t}{T_2 - T_1} \sum_i V_i I_i$$

$$V_{RMS} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} V^2(t) dt} \approx \sqrt{\frac{\Delta t}{T_2 - T_1} \sum_i V_i^2}$$

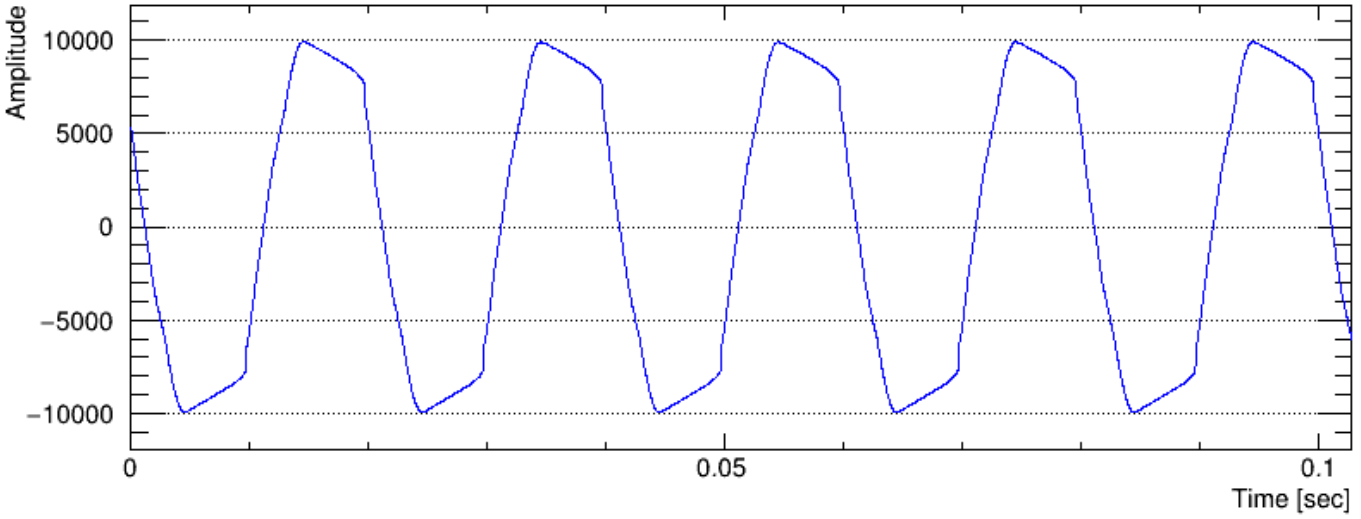
$$I_{RMS} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} I^2(t) dt} \approx \sqrt{\frac{\Delta t}{T_2 - T_1} \sum_i I_i^2}$$



ძაბვისა და დენის დროზე დამოკიდებულება



დენი



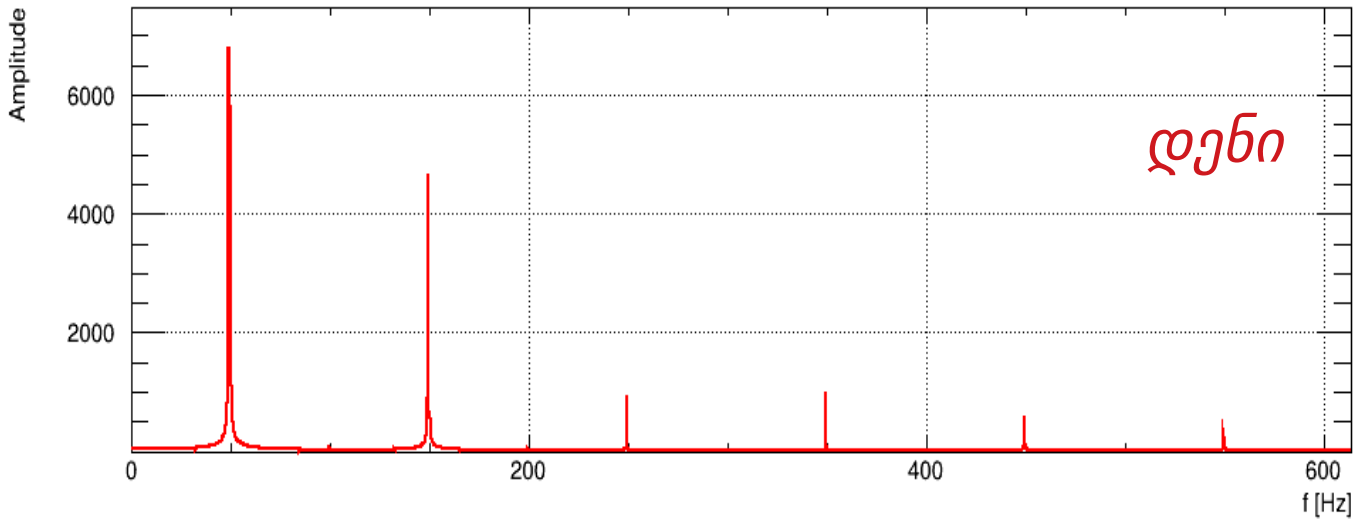
ძაბვა

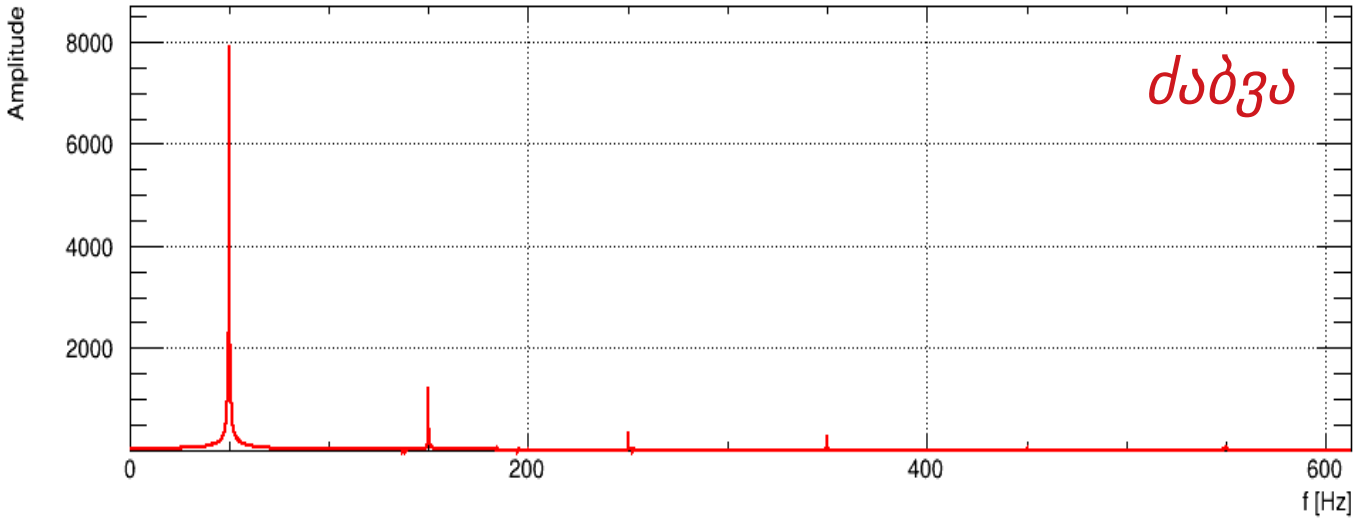
საბოლოო რეზულტატი:

$\cos(\varphi) \approx 0.8$

ძაბვისა და დენის სიხშირული სპექტრები

დისკრეტული ფურიე-გარდაქმნის (FFT) სპექტრი





დამატებითი ამოცანები

ფილტრების და აუდიო გამაძლიერებლების მახასიათებლების შესწავლა:

- გატარების ზოლი
- წრფივი დამახინჯების კოეფიციენტი
- ფაზური მახასიათებლები

მონაცემთა შეგროვების სხვა სისტემები

…..

ინტერნეტ-რესურსები

https://wavefilegem.com/how_wave_files_work.html

https://en.wikipedia.org/wiki/WAV