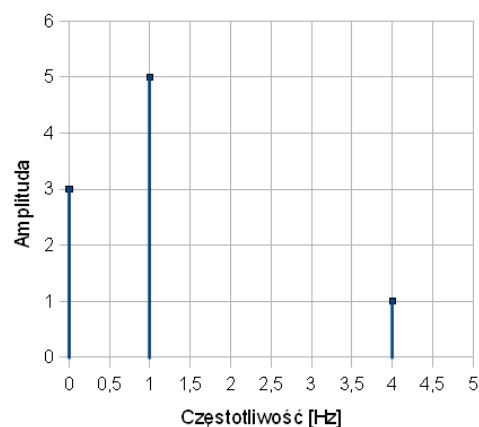
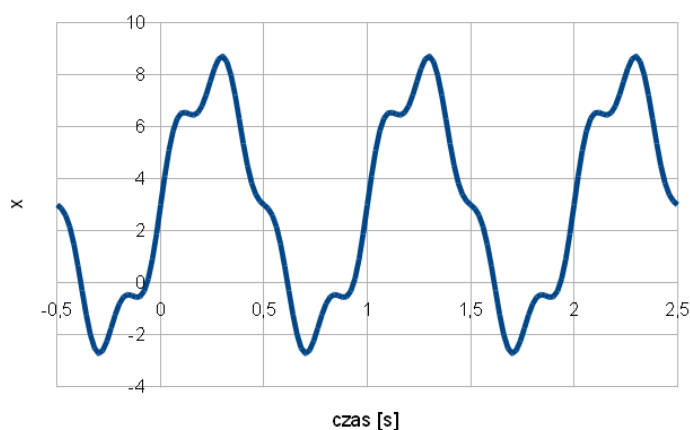
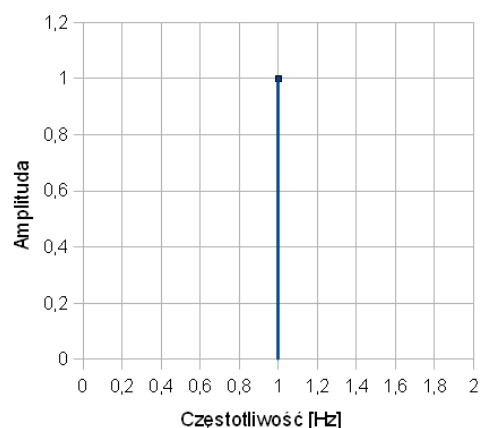
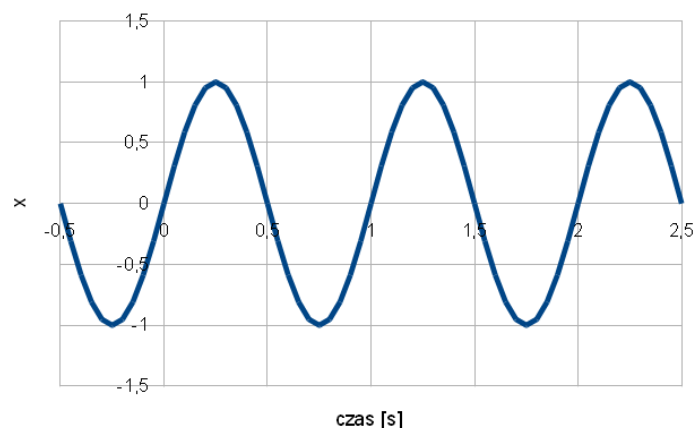
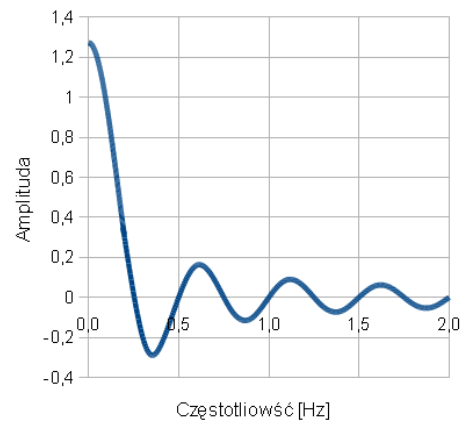
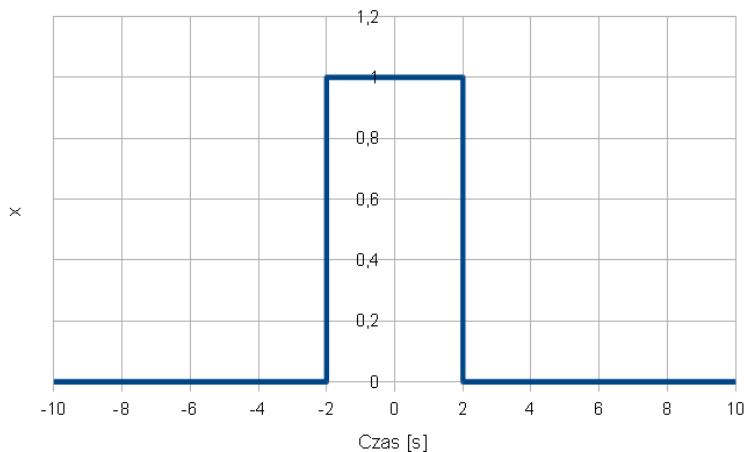


Analiza widmowa dźwięku (008; 22.07.2009; *processing*)

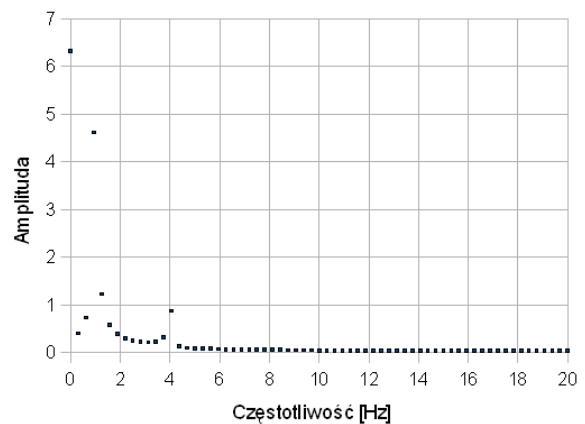
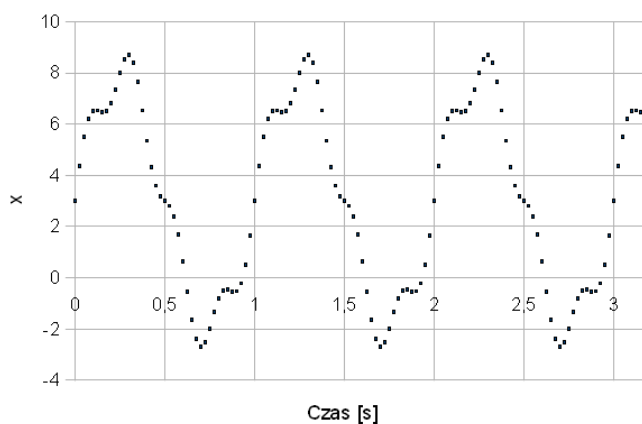
Wiele komputerowych odtwarzaczy muzyki rysuje zmieniające się w czasie wykresy słupkowe, które ewidentnie zależą od 'treści' tejże muzyki. Opiszę zatem poglądowo skąd biorą się owe wykresy, jak je uzyskać w *processingu* i do czego może się to przydać.

Transformata Fouriera (Fourier Transform, FT) jest przekształceniem matematycznym (całkowym), które dla sygnału będącego funkcją czasu generuje widmo tego sygnału, będące funkcją częstotliwości. Widmo to ma wartości zespolone, my jednak będziemy zajmować się tylko modułami tych liczb (tzw. widmem amplitudowym). Jeśli mamy do czynienia z sygnałem ciągłym opisanym matematycznie (dla czasu od minus do plus nieskończoności), to widmo tego sygnału jest widmem ciągłym lub dyskretnym (impulsowym). Przykładowe przebiegi i ich widma:





Dyskretna transformata Fouriera (Discreet Fourier Transform, DFT) jest przekształceniem analogicznym do transformaty furierowskiej, lecz realizowanym dla sygnału dyskretnego (spróbkowanego). Widmo otrzymane po DFT jest zawsze widmem dyskretnym. Przykładowy przebieg sygnału i jego widmo amplitudowe:



Szybka transformata Fouriera (Fast Fourier Transform, FFT) jest odmianą dyskretniej transformaty Fouriera, która wykorzystuje pewne jej właściwości do obliczania widma szybciej niż z definicji transformaty. Obliczenia FFT wykonuje się zazwyczaj przy zastosowaniu odpowiednio przygotowanych do tego bibliotek. Wymagane, aby liczba próbek sygnału była potęgą dwójki.

Czasowa transformata Fouriera (Time Fourier Transform, TFT) jest transformatą obliczaną dla kolejnych krótkich próbek czasowych sygnału, przez co otrzymujemy zmieniające się w czasie widmo sygnału.

Komputerowa karta dźwiękowa (nawet zintegrowana z płytą główną) ma ogromne możliwości jeśli chodzi o pomiar sygnałów. Według domyślnych ustawień, wszelkie sygnały dźwiękowe są próbkowane z częstotliwością 44100Hz i z rozdzielczością 16 bitów.

Podstawowe zależności dla widma dyskretnego obliczonego metodą FFT:

- dla częstotliwości próbkowania $f_p=44100\text{Hz}$ maksymalną częstotliwością na widmie sygnału będzie tzw. częstotliwość Nyquista równa zawsze połowie częstotliwości próbkowania $f_n=f_p/2=22050\text{Hz}$;
- dla liczby próbek sygnału N długość tego sygnału w sekundach wynosi $T=N/f_p$,
- kolejne częstotliwości (prążki widma), dla których obliczone zostało widmo są wielokrotnościami odwrotności długości próbki czasowej $df=1/T$

Pobieranie i kreślenie przebiegu sygnału.

Processing ma wbudowaną bibliotekę do zarządzania dźwiękiem o nazwie Minim. Daje ona ogromne możliwości z zakresu zarządzania i analizy dźwięku na komputerze. Przeanalizujmy następujący kod:

```
import ddf.minim.*; //import potrzebnych bibliotek
Minim minim; //utworzenie obiektu minim biblioteki Minim
AudioInput WE; //utworzenie obiektu wejścia sygnału

void setup()
{
  size(1024, 400);
  stroke(255);
  minim = new Minim(this); //utworzenie obiektu minim biblioteki Minim
  WE = minim.getLineIn(Minim.STEREO, 1024, 44100, 16);
  //obiekt WE przechowywać będzie próbki sygnału stereo w pakietach
  //po 1024 z częstotliwością próbkowania 44100Hz i rozdzielczości
  //16bitów
}

void draw()
{
  background(0);
  for(int i = 0; i < WE.bufferSize() - 1; i++)
  {
    line(i, 100 + WE.left.get(i)*100, i+1, 100 + WE.left.get(i+1)*100);
    line(i, 300 + WE.right.get(i)*100, i+1, 300 + WE.right.get(i+1)*100);
  }
}

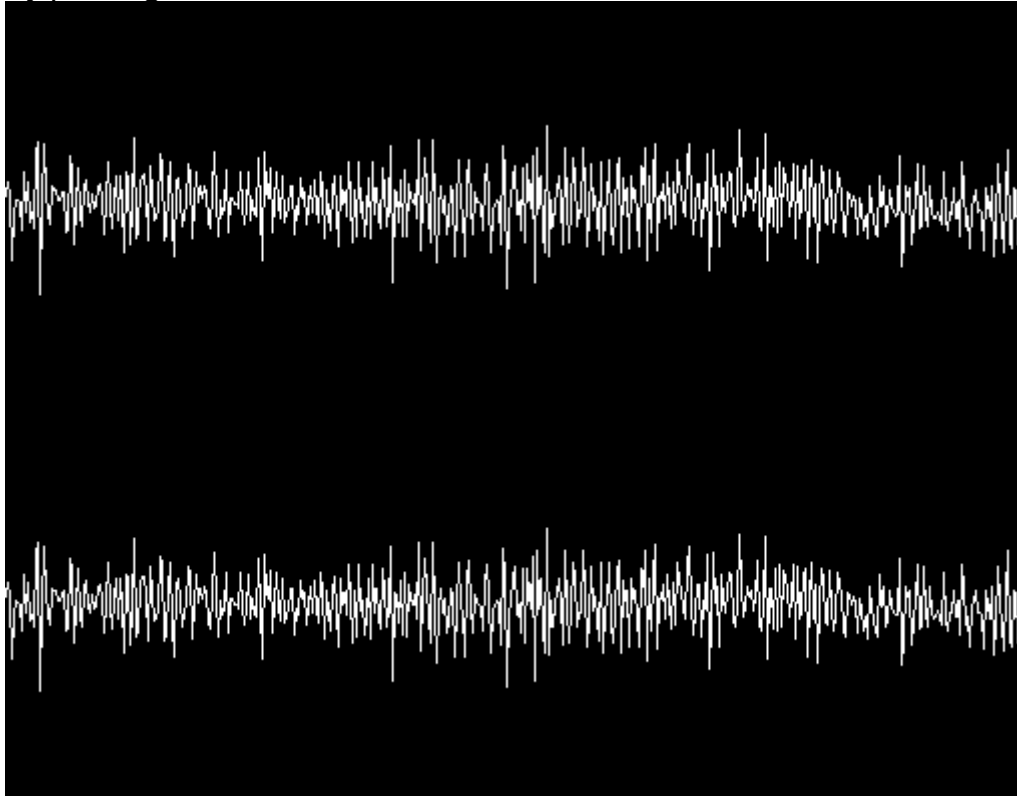
void stop() //blok poleceń niezbędny zawsze przy zastosowaniu
           //biblioteki minim
{
  WE.close();
  minim.stop();
  super.stop();
}
```

Polecenie `WE.bufferSize()` - zwraca wartość liczby próbek sygnału `WE`.

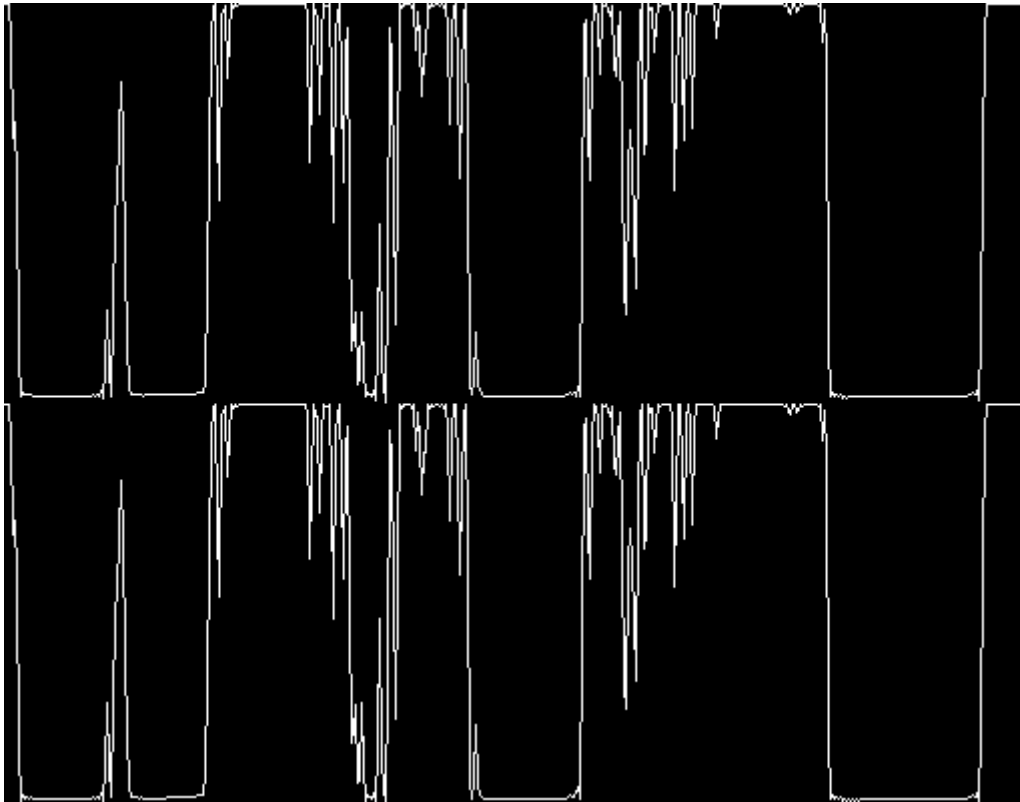
Polecenie `WE.left.get(i)` – zwraca wartość *i*-tej próbki lewego kanału sygnału `WE`.

Jeśli chcemy aby program odczytywał sygnał z mikrofonu, należy go włączyć w systemie operacyjnym (ew. dodatkowe wzmocnienie). Aby przez program odczytywany był dźwięk odtwarzany na komputerze, włączyć należy opcję Stereo Mix i ewentualnie wyłączyć mikrofon aby nie powodował zakłóceń.

Przykładowy przebieg:



Jeśli w czasie testowania programu zauważymy, że następuje przesterowanie sygnału (wartości sygnału są zbyt duże i następuje obcięcie wierzchołków) należy w systemie ściszyć głośność mikrofonu, wyłączyć dodatkowe wzmocnienie mikrofonu lub ściszyć Stereo Mix. Przykładowy przebieg przesterowany:



Kreślenie widma sygnału.

Ponownie stosując opisaną bibliotekę rozszerzamy wcześniejszy kod o możliwość kreślenia widma sygnału.

```
import ddf.minim.*;
import ddf.minim.analysis.*;    //import biblioteki analizy sygnałów
Minim minim;
AudioInput WE;
FFT widmo;    //utworzenie obiektu do przechowywania widma

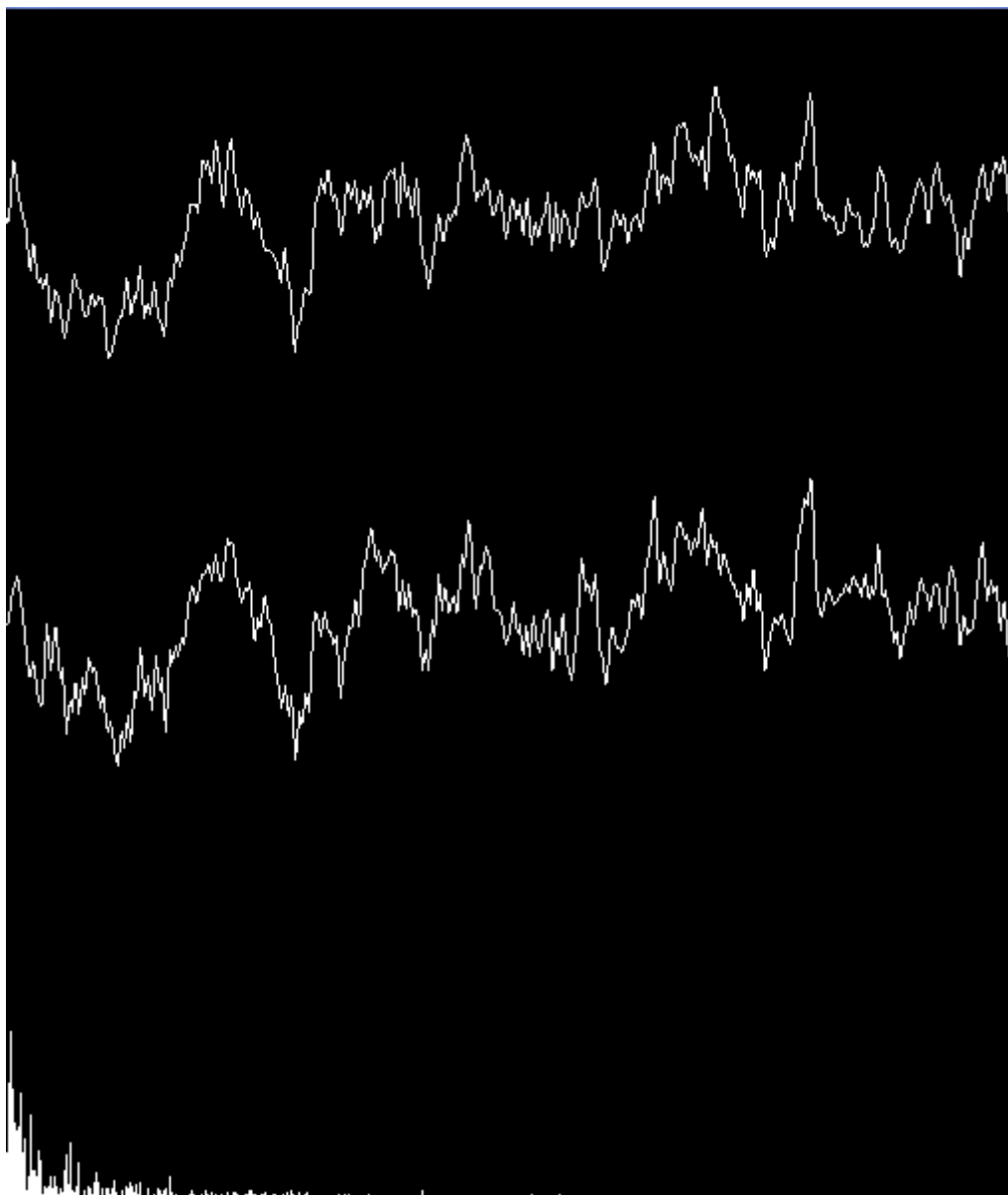
void setup()
{
  size(1024, 600);
  stroke(255);
  minim = new Minim(this);
  WE = minim.getLineIn(Minim.STEREO, 1024, 44100, 16);
  widmo = new FFT(WE.bufferSize(), WE.sampleRate()); //deklaracja widma
}

void draw()
{
  background(0);
  for(int i = 0; i < WE.bufferSize() - 1; i++)
  {
    line(i, 100 + WE.left.get(i)*100, i+1, 100 + WE.left.get(i+1)*100);
    line(i, 300 + WE.right.get(i)*100, i+1, 300 + WE.right.get(i+1)*100);
  }
  widmo.forward(WE.mix); //obliczenie widma z aktualnej próbki
  for(int i = 0; i < widmo.specSize(); i++)
  {
    line(i, height, i, height - widmo.getBand(i));
  }
}

void stop()
{
  WE.close();
  minim.stop();
  super.stop();
}
```

Polecenie `widmo.getBand(i)` zwraca wartość amplitudy (właściwie to modułu) widma dla i-tego prążka.

Przykładowy przebieg uzyskany za pomocą tego programu:

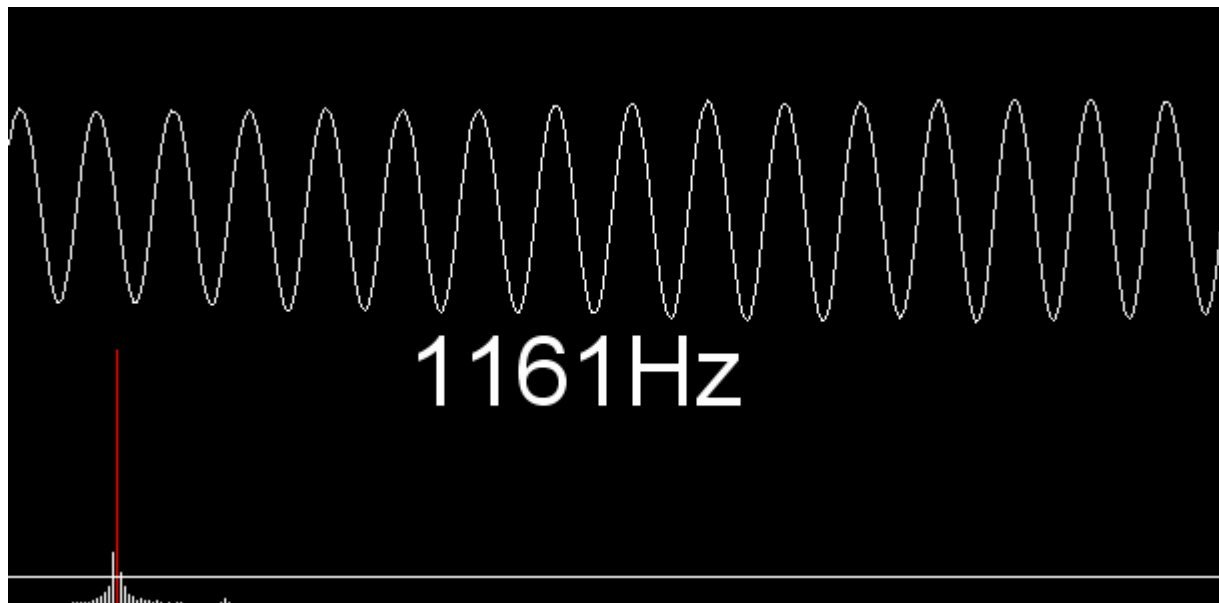


Zastosowanie.

W pierwszej kolejności polecam przetestować program dla różnych dźwięku z użyciem mikrofonu. Gwizdanie, dźwięki uzyskiwane za pomocą gitary, pianina.... wszystko to pozwala poznać istotę dźwięku, rozróżnić pojęcie wysokości i tonu. Bardzo przydatny jest do tego opis na stronie: http://www.fizykon.org/akustyka/spis_akustyka.htm.

Nasuwa się od razu proste zastosowanie opisanego programu – ustalając pewne progi na amplitudę dźwięków z zakresu pewnych częstotliwości możemy wykonać program reagujący na dźwięk (a reakcja może być dowolna). Prosto można odnaleźć na widmie prążek o największej amplitudzie i wyświetlić jego częstotliwość (to np. do strojenia instrumentów).

Tak wygląda rozszerzona wersja programu (pomiar mikrofonem podczas gwizdania, biała pozioma linia pokazuje wartość całkowitej energii sygnału – sumy iloczynów amplitud i kwadratów odpowiadających im częstotliwości):



Biblioteka Minim pozwala również na rejestrację dźwięków, generowanie ich, stosowanie filtrów, dodawanie efektów... pełna biblioteka z ogromną ilością przykładów:
<http://code.compartmental.net/tools/minim/>

Zastosowany w artykule program:

<http://myinventions.pl/008/AnalizaDzwieku.pde>