RGui

# Biblioteka: „sound”

*Podstawowe funkcje do działań na plikach .wav i próbkach dźwięku.*

Autor biblioteki:
Matthias Heymann

Opracowała:
Magdalena Wanat

# PLIKI .wav

Format plików dźwiękowych stworzonych przez Microsoft oraz IBM.

Pliki .wav zawierają informacje o:

- strumieniu audio,
- częstotliwości próbkowania,
- ilości kanałów (mono, stereo).

Pliki te wymagają dużo miejsca ok. 172kB/s, mają ograniczoną wielkość pliku do 4 GB ze względu na 32 bitowe zmienne.

Zastosowanie:

- edycja dźwięku (nie należą do grupy plików kompresji stratnej),
- przenośne urządzenia audio (cyfrowe dyktafony i odtwarzacze).

# TWORZENIE WŁASNYCH PRÓBEK DŹWIĘKU

**Generowane funkcje:**

***Sine***(*freq, dur, rate, bits, channels*) - **sinusoidalna**

***Sawtooth***(*freq, dur, rate, bits, channels, reverse*) - **piłokształtna**

***Square***(*freq, dur, rate, bits, channels, upPerc*) - **kwadratowa**

***Silence***(*dur, rate, bits, channels*) - **cisza**

***Noise***(*dur, rate, bits, channels*) - **zakłócenia (szum)**

freq – częstotliwość sygnału,
dur – czas trwania sygnału w sek.
rate – próbkowanie w liczbach całkowitych pomiędzy 1000 a 48000 *(...,rate=44100,...)**,
bits – jakość próbkowania 8 lub 16(w niektórych przypadkach 24) bit/próbkę *(...,bits=16,...)**,
channels – 1 dla mono, 2 dla stereo *(...,channels=1,...)**,
reverse – jeśli TRUE przebieg zostanie odwrócony *(...,reverse=FALSE)**,
upPerc – numer pomiędzy 0 a 100 dający wartość procentową przebiegu *(...,upPerc=50)*.*
*(*)- wartości domyślne*

# FUNKCJA SOUND()

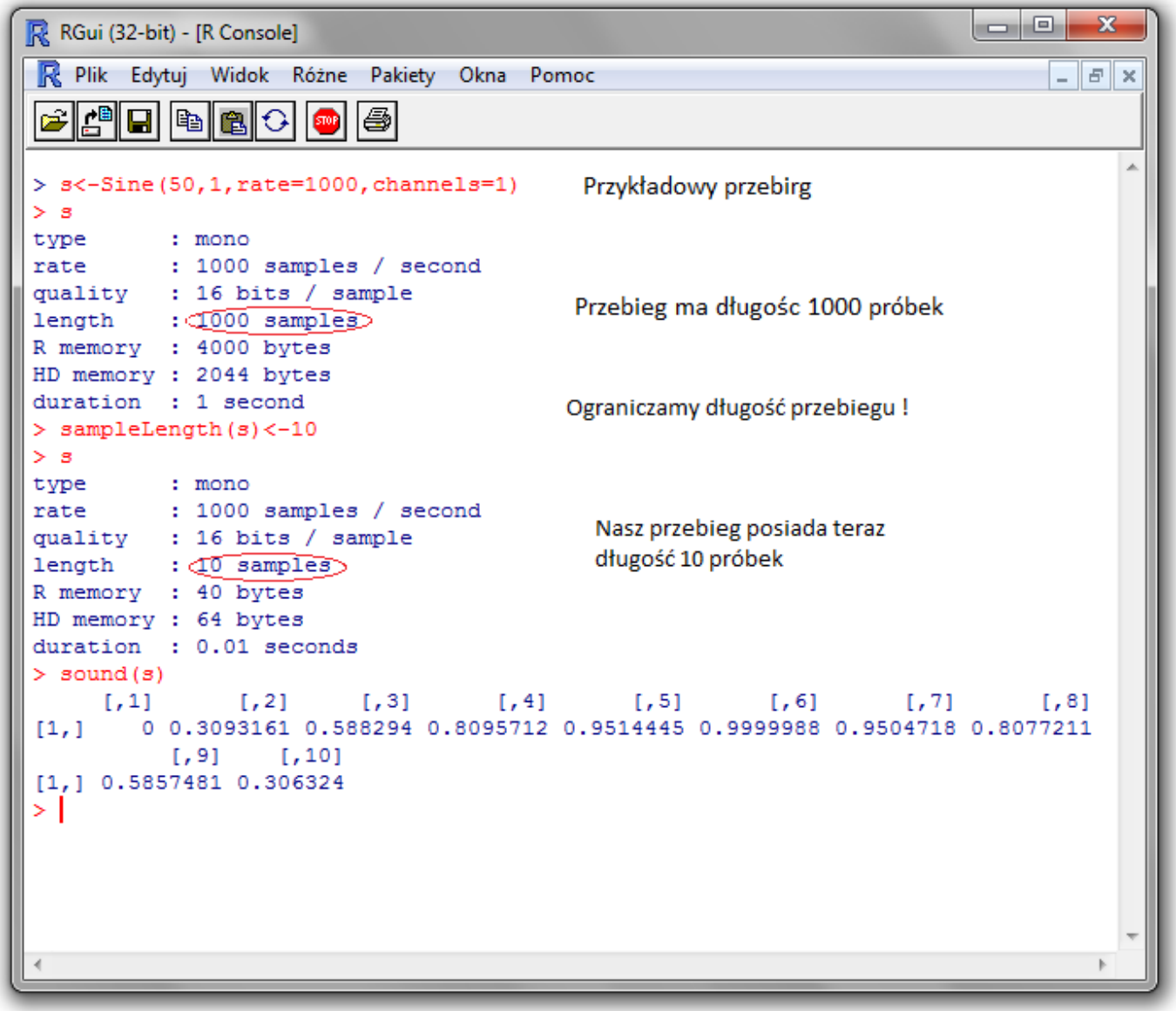
Funkcja ta pozwala na podgląd wartości, które przyjmuje dany sygnał dźwiękowy. Rozpisuje macierz w zależności od liczby kanałów, dla mono macierz [1,liczba próbek] lub dla stereo macierz [2,liczba próbek].

Niestety nie jesteśmy w stanie obserwować w pełni naszej macierzy, ze względu na domyślną liczbę bądź minimalną próbek macierz ta zawiera minimum 1000 próbek. Macierz najzwyczajniej się nie mieści w konsoli programu R. Musimy ograniczać liczbę wyświetlanych próbek.

Przykład:

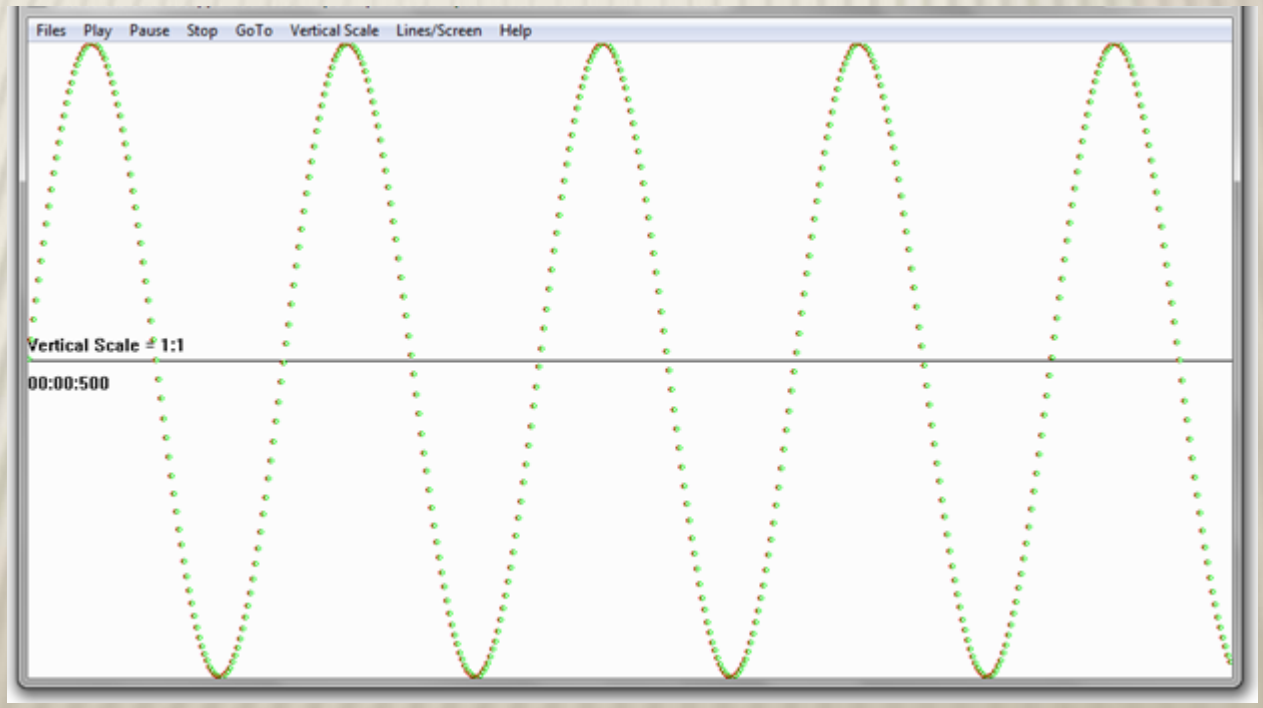
```
> sound(s)
     [,1]      [,2]     [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
[1,]    0 0.3093161 0.588294 0.8095712 0.9514445 0.9999988 0.9504718 0.8077211
          [,9]    [,10]
[1,] 0.5857481 0.306324
```

# OGRANICZANIE DŁUGOŚCI PRZEBIEGÓW



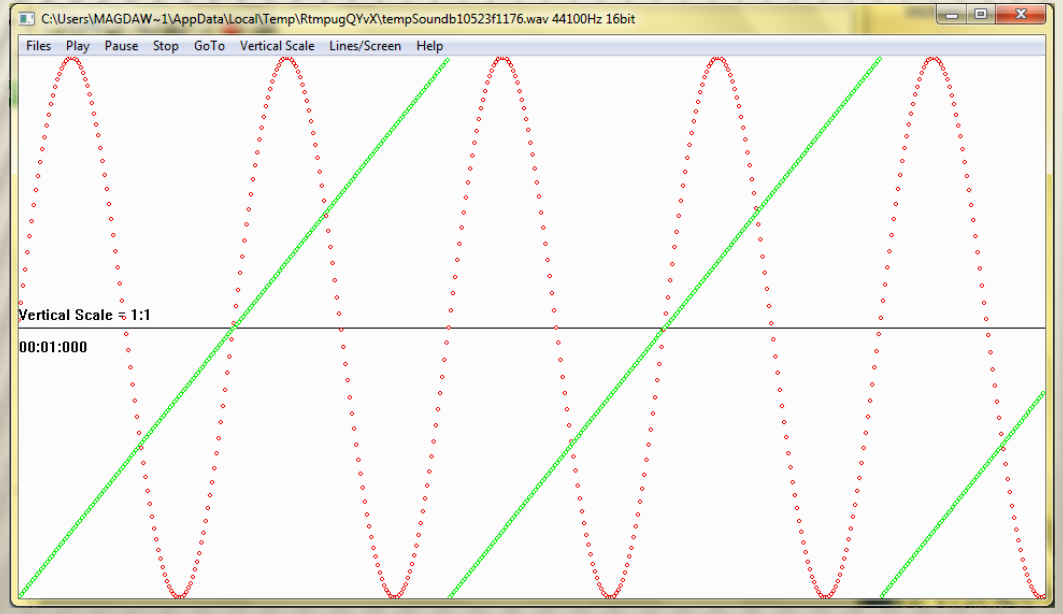
# POWIĄZANIE RGui Z INNYM ODTWARZACZEM .wav „playwave”

By powiązać odtwarzacz PlayWave z programem należy umieścić w katalogu roboczym plik .exe programu. Następnie należy ustawić odtwarzacz przez konsole funkcją *„setWavPlayer(‘playwave’)”*.

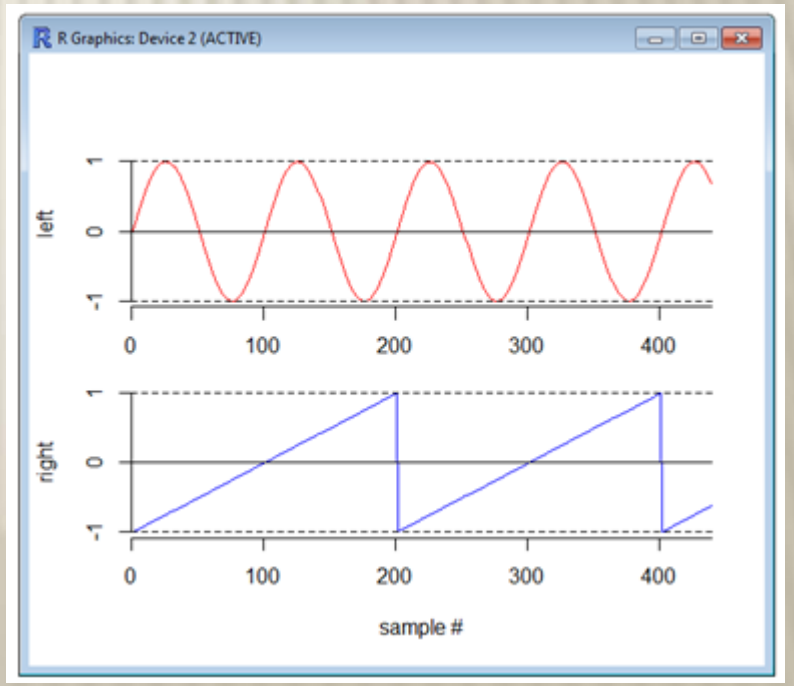


# USTAWIANIE RÓŻNYCH SYGNAŁÓW NA DWA RÓŻNE KANAŁY

```
> sLeft<-Sine(440,1)
> sRight<-Sawtooth(220,1)
> s<-stereo(sLeft,sRight)
> play(s)
> plot(s[0:440])
```



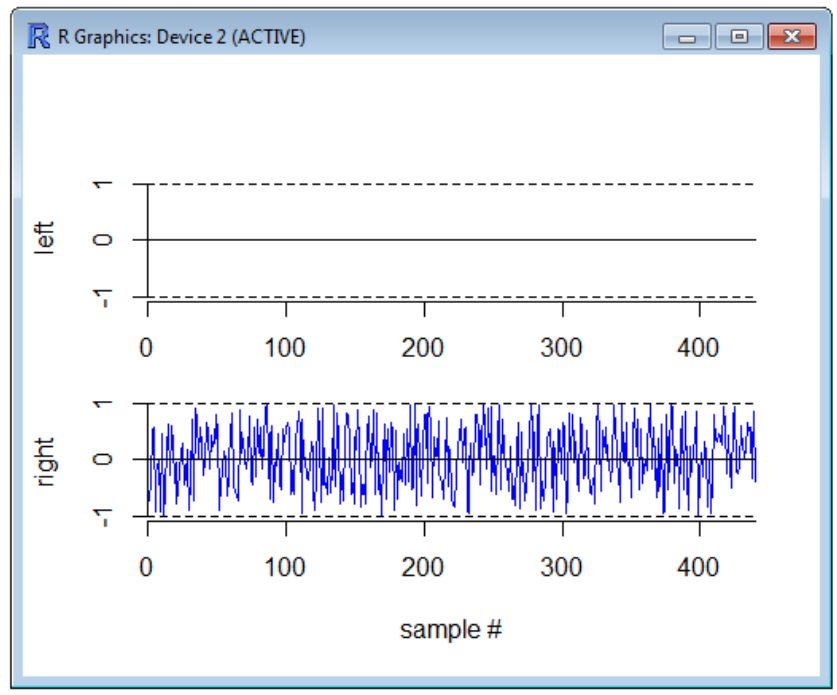

Przebiegi kanałów lewego i prawego:

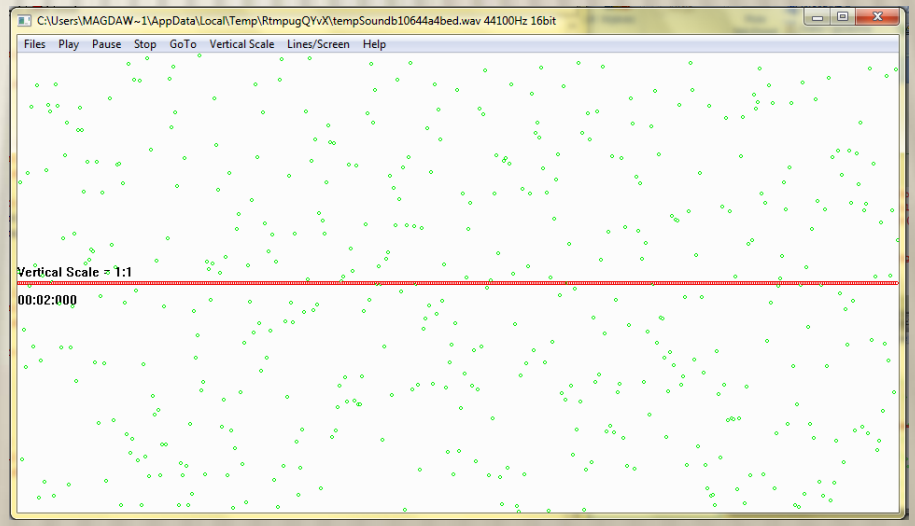


# WPROWADZANIE SZUMU I CISZY

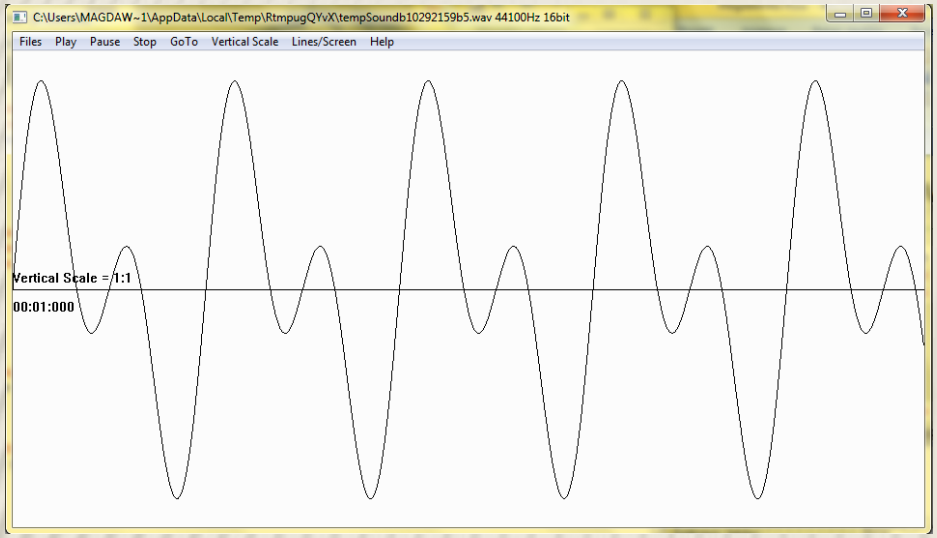
Wprowadzenie szumu to wprowadzenie sygnału o losowych wartościach amplitudy, sygnał ten symuluje nam zakłócenia. Wprowadzając ciszę możemy wyciszać jeden kanał co może ułatwić obserwację interesującego nas kanału.

```
> sRight<-Noise(2)
> sLeft<-Silence(1)
> s<-stereo(sLeft,sRight)
> plot(s)
> play(s)
> sampleLength(s)<-440
> plot(s)
>
```



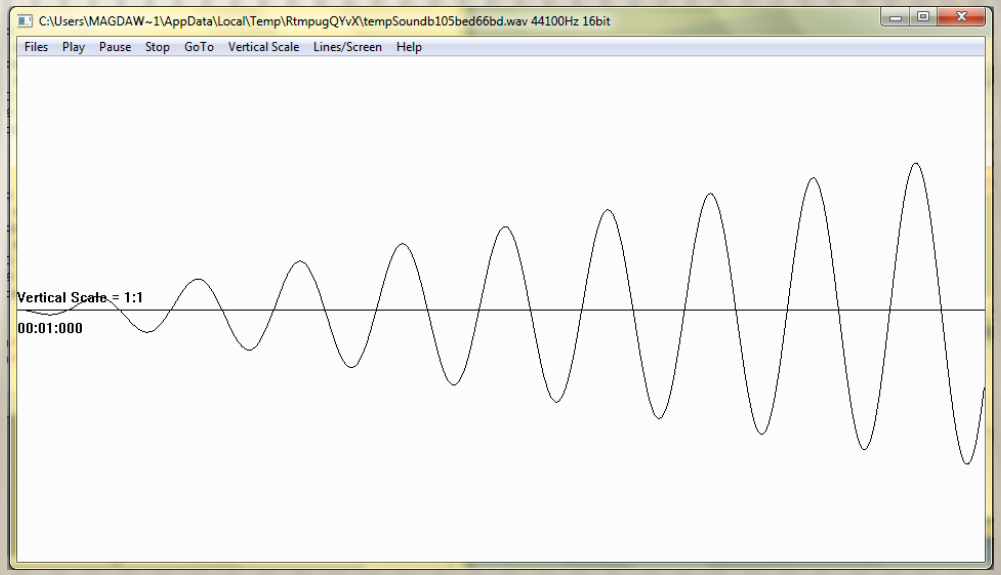



# MODYFIKACJA PRZEBIEGÓW SYGNAŁU




```
> e1<-Sine(440,1)
> e2<-Sine(220,1)
> play((e1+e2)/2)
```

```
> play(Sine(440,1)*Sine(5,1))
```




Mamy możliwość tworzenia własnych przebiegów sygnału sklejonych z wielu funkcji. Możemy stworzyć dowolny sygnał dźwiękowy, który możemy modyfikować i badać.

# OPERACJE NA GOTOWYCH PROGRAMACH I WPROWADZANIE ZMIAN MAŁO INWAZYJNYCH W KOD PROGRAMU

**Operacje na wartościach bitowych:**

- bits(s)<- wartość
- setBits(s, wartość)

**Operacje na częstotliwości próbkowania:**

- rate(s)<- częstotliwość próbkowania
- setRate(s, częstotliwość próbkowania)

**Zmiana kanału odtwarzania:**

- channels(s)<- kanał
- setChannels(s, kanał)

**Zmiana wartości czasu trwania sygnału:**

- duration(s)<- czas
- setDuration(s, czas)

**Zmiana długości przebiegu:**

- sampleLength(s)<- długość przebiegu
- setSampleLength(s, długość przebiegu)

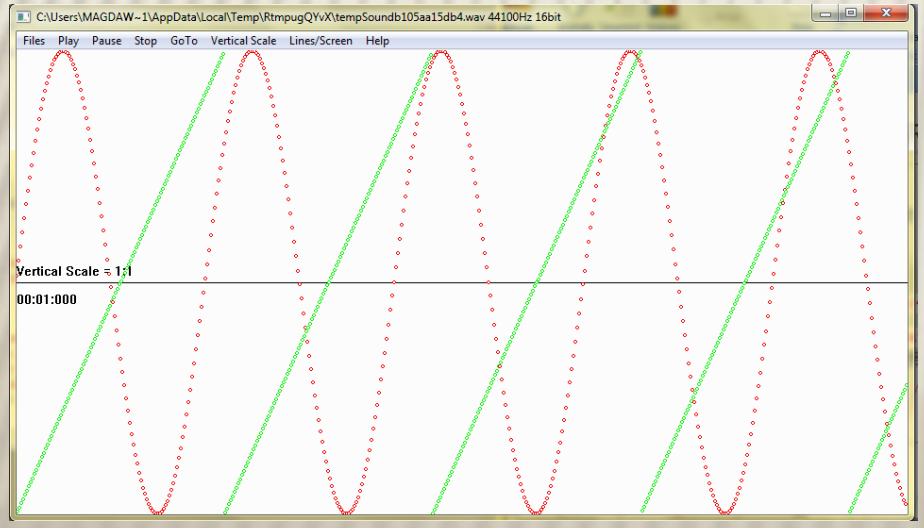
*s - generowany sygnał dźwiękowy*

# ZAMIENIANIE KANAŁÓW ODTWARZANIA

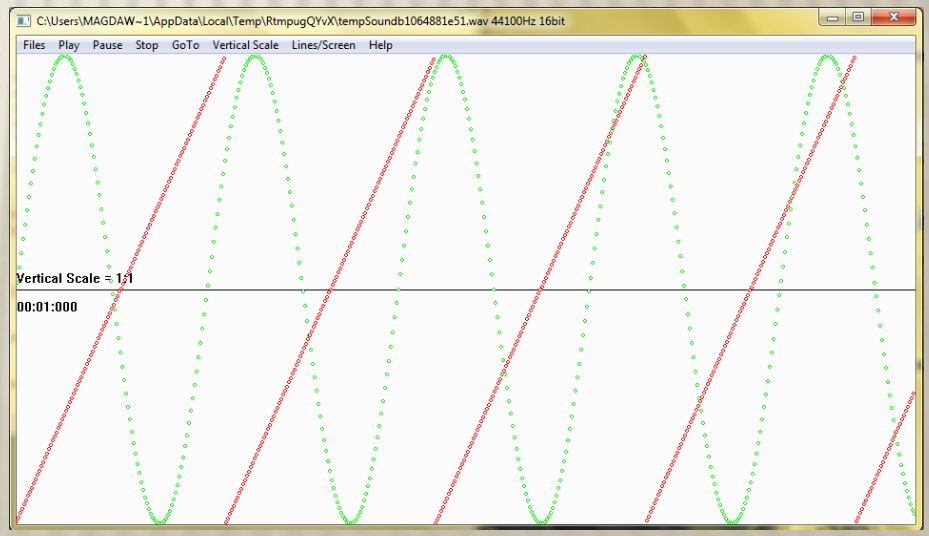
Poprzez dodanie do funkcji *play()* funkcję *mirror()* możemy zamienić kanały odtwarzania.

```
> s<-stereo(Sine(440,1),Sawtooth(440,1))
```

```
> play(s)
```




```
> play(mirror(s))
```



# WCZYTYWANIE PLIKÓW FORMATU .wav

By otworzyć w programie plik *wav* lub próbkę dźwięku stworzoną wcześniej przez nas należy skorzystać z funkcji *loadSample().*

Struktura funkcji:

*s<-loadSample("nazwa.wav",filecheck=TRUE)*

***s*** - sygnał otworzony,

***nazwa*** - nazwa pliku pod jaką występuje dany sygnał dźwiękowy,

***filcheck*** – domyślnie „TRUE" program sprawdza zgodność pliku, jeśli ustawimy „FALSE" to program nie sprawdzi czy plik zawiera jakieś informacje i czy jest zgodny z programem.

# ZAPISYWANIE STWORZONYCH PRÓBEK DŹWIĘKU DO PLIKÓW FORMATU .wav

By zapisać stworzoną próbkę dźwięku należy skorzystać z funkcji *saveSample().*

Struktura funkcji:

*saveSample(s,nazwa,overwrite=FALSE)*

***s*** - sygnał zapisywany,

***nazwa*** - nazwa pliku pod jaką zostanie zapisany,

***overwrite*** – jeśli „FALSE” to gdy mamy istniejący plik o takiej nazwie zostanie wyrzucony błąd, jeśli „TRUE” to istniejący plik zostanie zamieniony.

# ZASTOSOWANIE

- Modulacja i tworzenie dźwięków o różnej częstotliwości i amplitudzie.
- Badanie i wyciszanie zakłóceń dźwiękowych.
- Pomiary i analiza hałasu.
- Obróbka dźwięku.

# DZIĘKUJĘ ZA UWAGĘ