304144 Maciej Dąbkowski PPJG 24Z

Koncepcja projektu: Equalizer do mikrofonu

1. Opis funkcjonalności

Została stworzona aplikacja **LabVIEW**, która działa jako **equalizer** dla sygnału audio pochodzącego z mikrofonu. Program umożliwi użytkownikowi regulację poszczególnych pasm częstotliwości w czasie rzeczywistym, a także wizualizację sygnału wejściowego i przetworzonego, dodany został także „pitch shifter".

Główne funkcjonalności:

1. **Odbiór sygnału audio z mikrofonu**
    - Aplikacja odbiera sygnał dźwiękowy z mikrofonu podłączonego do komputera.
    - Sygnał jest przetwarzany w czasie rzeczywistym.

    Na początku jest następuje konfiguracja mikrofonu **Sound Input Configure.v**i oraz inicjowana jest kolejka. Następnie w pętli konsumenta następuje akwizycja danych w czasie rzeczywistym z mikrofonu za pomocą **Sound Input Read.vi** oraz potem dane dodawane są do kolejki.

2. **Equalizer (korekcja częstotliwości)**
    - Możliwość modyfikacji sygnału przy pomocy **filtrów pasmowych** (3 pasm częstotliwości).
    - Użytkownik może regulować wzmocnienie lub tłumienie dla każdego pasma za pomocą suwaków (od 0 do 1).
    - Możliwość zmiany barwy głosu za pomocą „pitch shiftera".

    Regulacja wzmocnienia następuje za pomocą **choice_band.vi**. „Pitch shifting" odbywa się za pomocą **Pitch Shift in Real Time.vi**.

3. **Wizualizacja sygnału audio**
    - Wyświetlanie sygnału w czasie rzeczywistym w formie:
        - Wykresu czasowego (oscyloskop) – sygnał przed oraz po modyfikacji
        - Widma częstotliwościowego (FFT).

4. **Odtwarzanie i zapis sygnału**
    - Opcja odtwarzania przetworzonego sygnału przez głośniki.

- Zapis przetworzonego sygnału audio do pliku WAV.

  Zapisywanie do pliku .wav odbywa się za pomocą **Save To Wav.vi**. Zostały skonfigurowane również Vi do odtwarzania sygnału audio w czasie rzeczywistym (**Sound Output Configure.v**i, **Sound Output Read.vi).**

5. **Kontrola parametrów aplikacji**
   - Interfejs graficzny z:
     - Suwakami do regulacji poszczególnych pasm equalizera oraz to zmiany modulacji głosu „pitch shifting" (Możemy cały czas zmieniać parametry).
     - Guzikiem „Start Recording" do wystartowania nagrywania głosu.
     - Guzikiem do zatrzymania nagrywania i zapisania sygnału (Tylko po kliknięciu „Start Recording".
     - Regulacją poziomu głośności sygnału wyjściowego(Możemy cały czas zmieniać parametry).

2. Opis architektury

Architektura aplikacji bazowana jest na modelu **producent-konsument**:

1. **Producent**:
   - Odpowiada za akwizycję danych (odbieranie sygnału z mikrofonu).
   - Dane są umieszczane w buforze (kolejce) FIFO (First In, First Out).
2. **Konsument**:
   - Odpowiada za pobieranie danych z kolejki i ich przetwarzanie (zapis do pliku, wizualizacja systemu audio, korekcja częstotliwości oraz odtwarzanie sygnału).

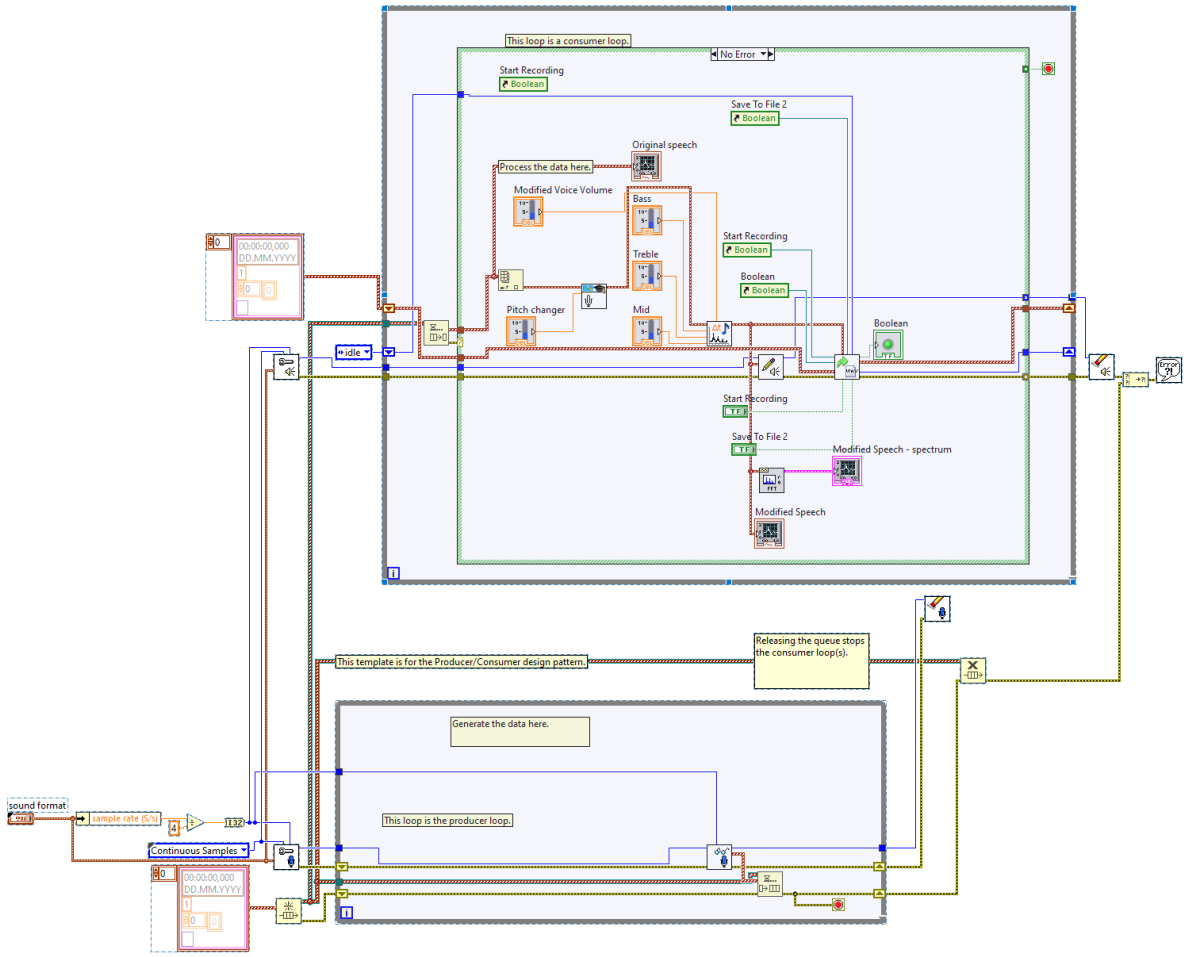
Korekcja częstotliwości oraz „pitch shifting" – odbywa się od razu po wybraniu z kolejki sygnału. Użytkownik jest w stanie za pomocą kontrolek regulować niskie, średnie oraz wysokie częstotliwości oraz modulację częstotliwości.

Zapis do pliku został zrobiony jako maszyna stanów (oddzielny Vi). Stan „idle" działa gdy użytkownik nic nie klika. Po kliknięciu „Start Recording" przechodzi w stan „write", który zapisuje sygnał do tablicy. Po kliknięciu „Save Sound" włącza się stan „save", który otwiera lub tworzy plik i do niego zapisuje sygnał. Następnie przechodzi do stanu „idle".

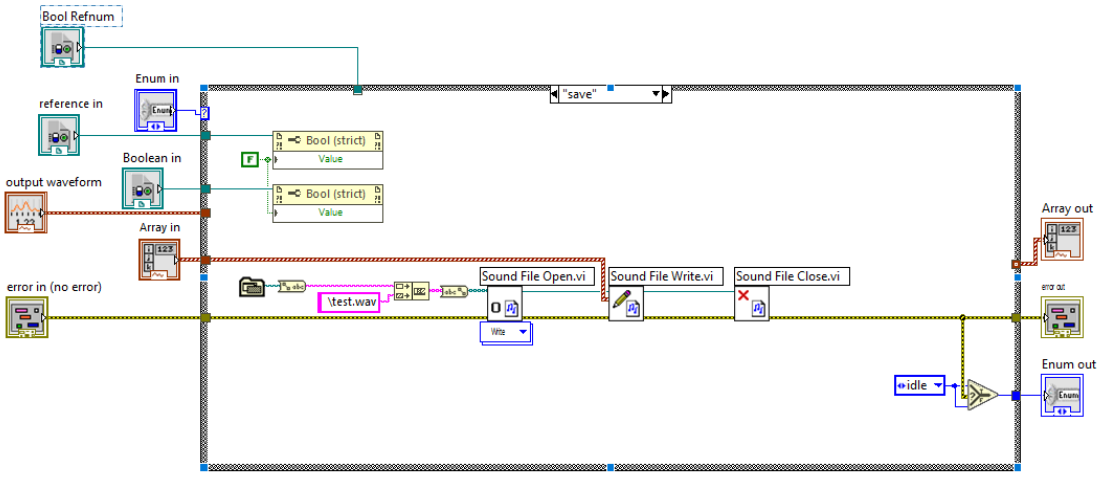
Wizualizacja sygnału to „Waveform graph" podłączone do różnych sygnałów.

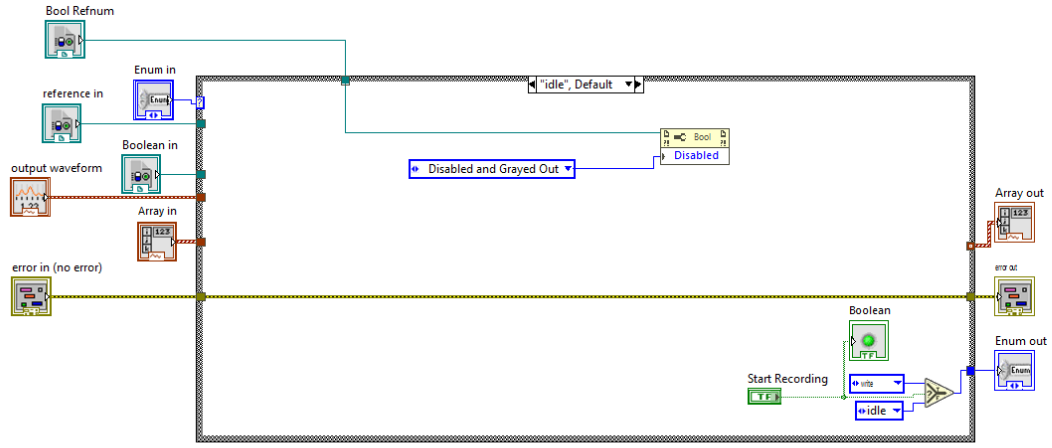
Problem odświeżania GUI oraz problem zmiany wartości parametru w trakcie konkretnego działania został rozwiązany za pomocą referencji. Wewnętrzne Vi korzystają z nich do zaktualizowania paramentów głównych wewnątrz nich.

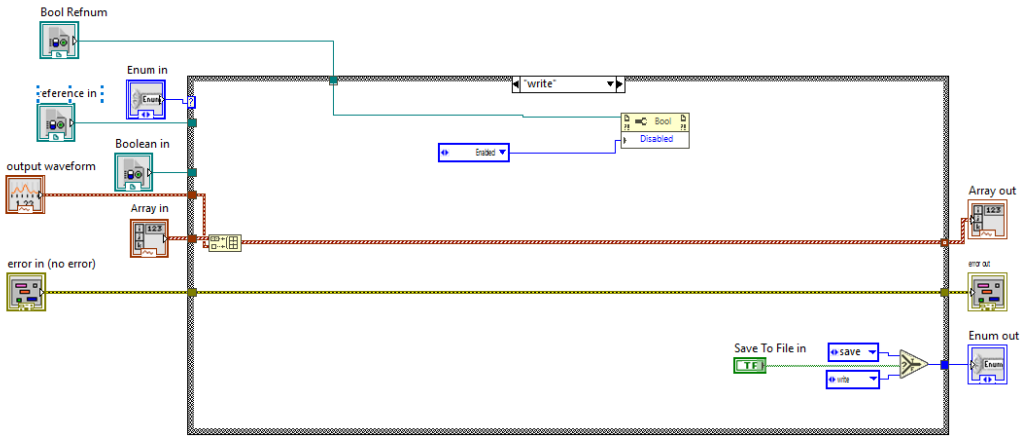
Block diagram głównego Vi.



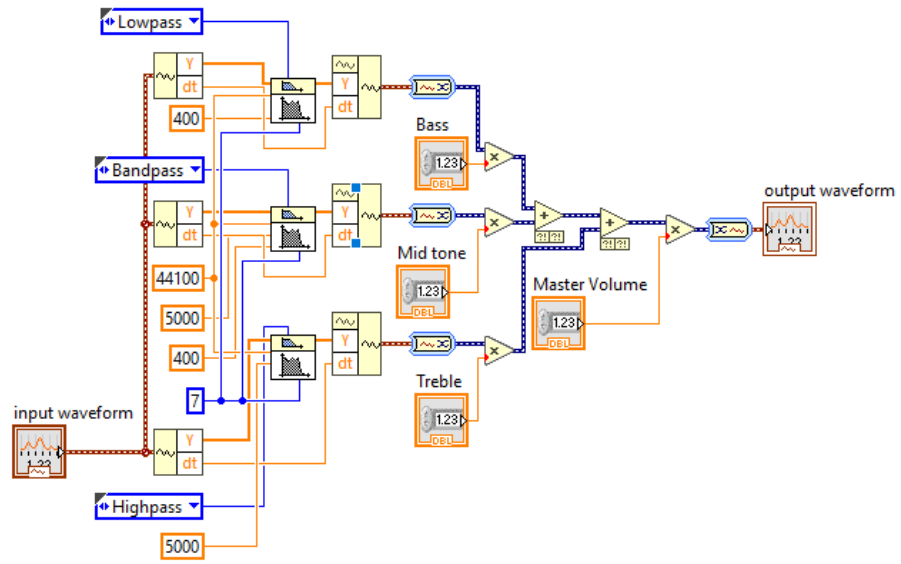
Block diagram Save to Wav.vi



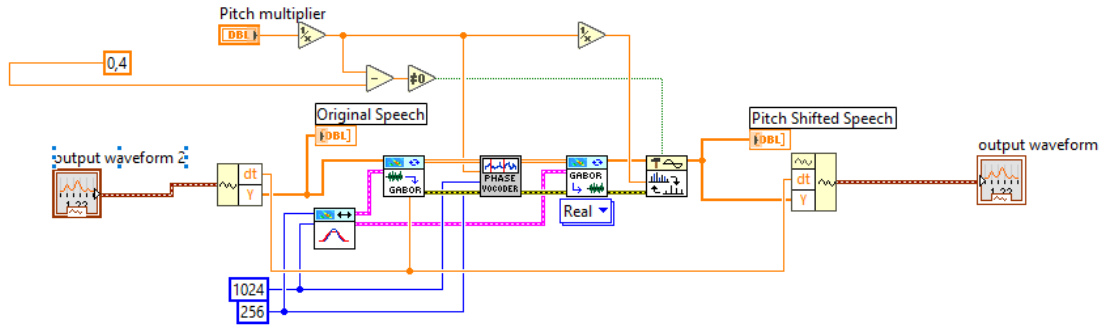




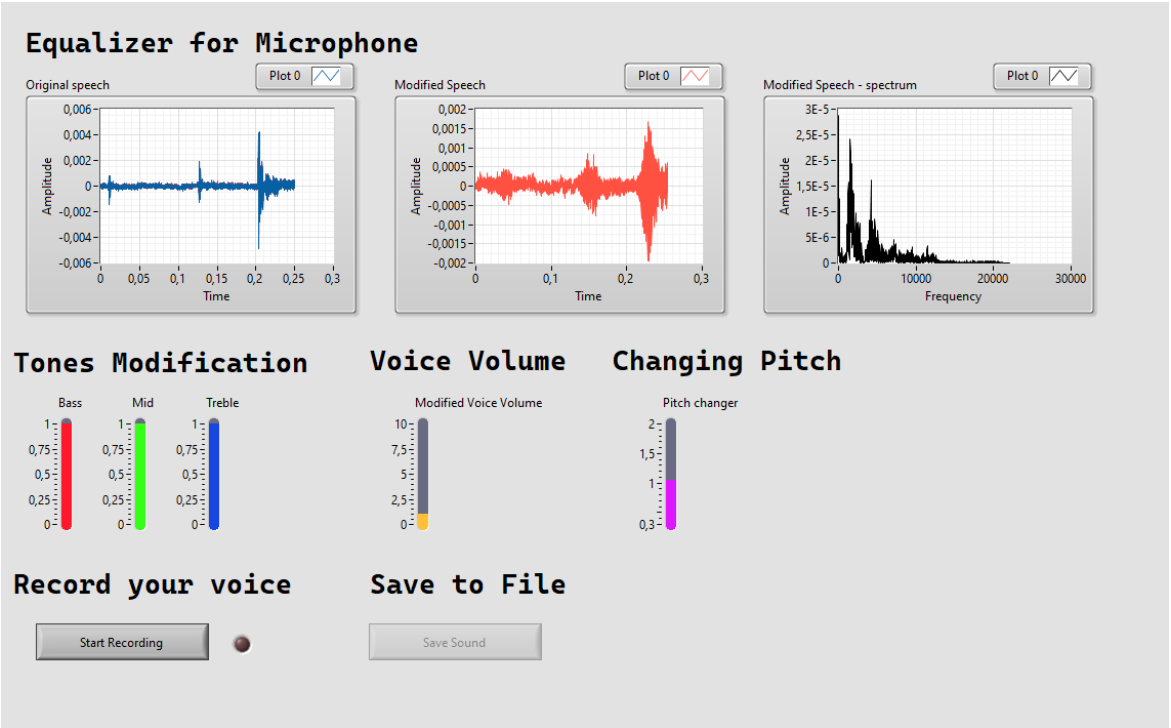
Block diagram choice band.vi



Block diagram Pitch Shifter in Real Time.vi



GUI programu



Zbudowany projekt do pliku exe