

# Transmisja danych dźwiękiem w JavaScript od podstaw

## Część 1: Dyskretna Transformata Fouriera

Przez lata JavaScript stał się językiem o ogromnym potencjale. W nowoczesnych przeglądarkach znajdziemy API do wielu różnych zastosowań. Możemy np. rysować grafikę 2D/3D przy użyciu WebGL czy też użyć WebWorkerów dla zwiększenia wydajności bardziej złożonych obliczeń na procesorach wielordzeniowych. Możemy także obsługiwać sprzęt, taki jak mikrofon czy głośniki poprzez Web Audio API. Lista wszystkich interfejsów oferowanych przez przeglądarki z biegiem lat staje się coraz dłuższa. Sprawia to, że JS staje się coraz bardziej popularny, a w przypadku urządzeń mobilnych aplikacje często konkurują z rozwiązaniami natywnymi.

Celem tego artykułu jest wykorzystanie Web Audio API do powrotu do starych czasów. Czy pamiętamy jeszcze dźwięk modemu inicjującego wdzwaniane połączenie? Był on dość hałaśliwy i trwał kilka sekund. Po poprawnej inicjalizacji dane były transmitowane i odbierane poprzez zwykłe połączenie telefoniczne. W zasadzie można było „posłuchać” tych danych, podnosząc słuchawkę telefonu podłączonego do tej samej linii. Dlaczego dźwięk? Dlatego że w tamtych czasach wielu ludzi posiadało już linię telefoniczną. Użycie modemu oraz istniejącej infrastruktury było po prostu tańsze i prostsze.

Nie ma jednak róży bez kolców. Systemy telefoniczne w tamtych czasach były projektowane tylko do przesyłania ludzkiej mowy w bardzo ograniczonym zakresie. Cytując Wikipedię: „W telefonii użytkowy zakres częstotliwości leży między 300 Hz a 3400 Hz”. Innymi słowy, modemy musiały pracować w bardzo ograniczonym paśmie (około 3 kHz) na analogowym kanale z dość dużymi zakłóceniami. Pierwszy komercyjny modem wypuczonego do sprzedaży pod koniec lat 50. ubiegłego wieku. Pozwalał on na transmisję danych na poziomie 100 bitów na sekundę. Przez lata prędkość ta stopniowo zwiększała się. Na końcu ery modemów (w późnych latach 90.) urządzenia oferowały już prędkości rzędu 56 kbit/s.

Web Audio API pozwala na generowanie i przetwarzanie dźwięku w JavaScriptcie. Możemy zatem użyć go do stworzenia aplikacji działającej jak modem. Powietrze w naszym pokoju spełniałoby rolę analogowej linii telefonicznej. Ważną różnicą jest fakt, że obecny sprzęt ma dużo lepszą jakość. Pokrywa on w całości zakres dźwięków słyszalnych przez człowieka (w teorii częstotliwości pomiędzy 20 Hz a 20 kHz). Oznacza to, że mamy dużo większe pasmo do dyspozycji. Naszym celem jest przesłanie między dwoma komputerami danych binarnych nawet przy udziale niepożądanych zakłóceń, takich jak biały szum, muzyka czy rozmowa. Dodatkowo wszystkie operacje na próbkach audio zaimplementujemy samodzielnie. Pozwoli nam to zrozumieć od podstaw cały proces filtrowania sygnału. Jakich prędkości oczekujemy? Tak naprawdę nie chodzi tu o prędkość. Mogłoby być to nawet 8 bitów na sekundę. Prędkości na poziomie 64 bit/s pozwoliłyby przesyłać w kilka sekund np. numery telefonów czy adresy stron bez użycia maila czy komunikatora. Możliwe byłoby stworzenie prostego czata, który nie wymaga żadnego połączenia Internetowego.

By stworzyć aplikację tego typu od podstaw, musimy zaprzęgnąć techniki cyfrowego przetwarzania sygnałów (ang. *Digital Signal Processing*, DSP). Pierwszą rzeczą, od jakiej warto zacząć, jest Dyskretna Transformata Fouriera (ang. *Discrete Fourier Transform*, DFT).

### WPROWADZENIE DO DTF

Czy kiedykolwiek zastanawialiśmy się, jak to możliwe, że tak wiele bezprzewodowych urządzeń jest w stanie komunikować się jednocześnie bez żadnych zakłóceń? Na przykład możemy używać sieci Wi-Fi (2.4 GHz), telefonu komórkowego LTE (2600 MHz) oraz oglądać programy telewizyjne DVB-T (około 600 MHz) w tym samym czasie. Jedną z odpowiedzi jest fakt, że wszystkie te urządzenia pracują na różnych częstotliwościach fal elektromagnetycznych. Kiedy musimy radzić sobie z częstotliwościami, z pomocą przychodzi Transformata Fouriera. Dla sygnałów cyfrowych będzie to Dyskretna Transformata Fouriera. Co nam ona daje? Otóż transformata ta potrafi zamienić sygnał reprezentowany w dziedzinie czasu na dziedzinę częstotliwości. Innymi słowy, możemy wyciągnąć częstotliwości, które tworzą sygnał zmieniający się w czasie. Pozwala to np. skupić się tylko na wąskim zakresie, pomijając całą resztę spektrum. Transformata daje nam zatem możliwość filtrowania tylko tego, co nas interesuje.

Chwileczkę... czy nie mieliśmy czasem skupić się na dźwięku? Oczywiście tak, jednak generalnie w obydwu przypadkach chodzi o częstotliwość. Nie ma znaczenia, czy są to fale radiowe (3 kHz – 300 GHz) czy fale akustyczne słyszalne przez człowieka (20 Hz – 20 kHz). W obydwu przypadkach sygnał może być zdigitalizowany przez próbkowanie go w miarę upływu czasu, zatem algorytm jest ten sam.

W dzisiejszych czasach dźwięk nie jest już powszechnie używany do przesyłania informacji, jednak ciągle możemy znaleźć inne zastosowania DFT. Na przykład odtwarzacze muzyczne bardzo często pokazują na swoich wyświetlaczach wynik działania transformaty w postaci pasek equalizera. Z tych ładnie wyglądających „skaczących paseczków” możemy odczytać, jak głośny jest każdy z zakresów częstotliwości słuchanej piosenki. Na przykład możemy odczytać, jak szybki jest rytm basu, nawet bez zakładania słuchawek (Rysunek 1).