

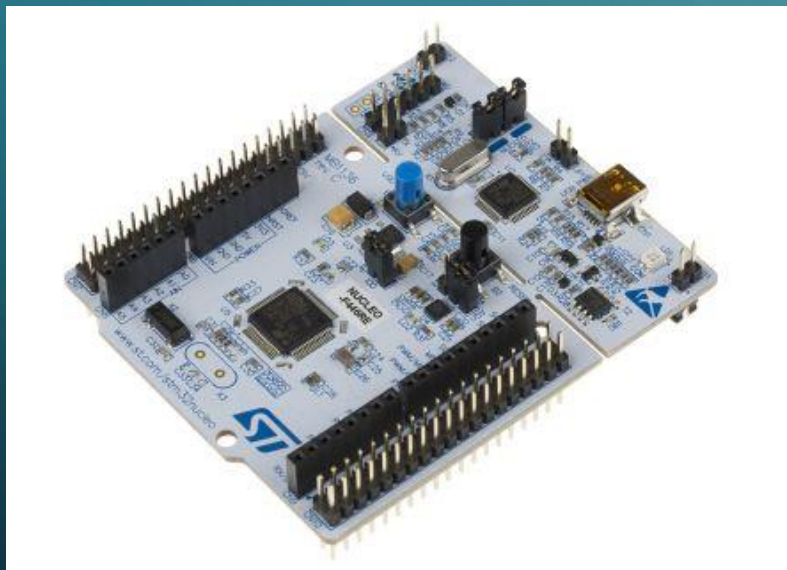
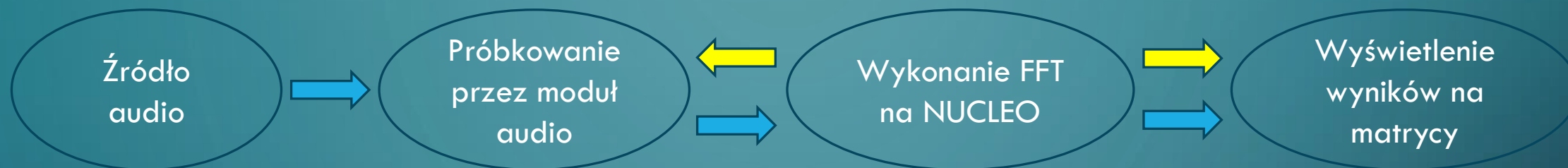
IMPLEMENTACJA ALGORYTMU FFT DLA UKŁADU NUCELO-F446RE

ZAAWANSOWANE ZASTOSOWANIA
UKŁADÓW MIKROPROCESOROWYCH

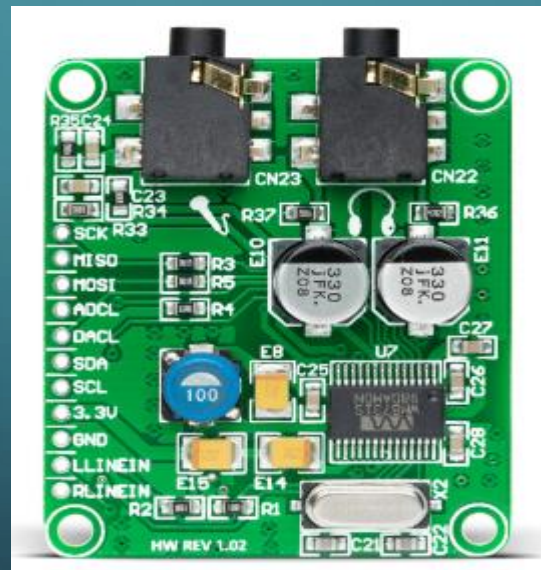
JAKUB PYZNAR
JAKUB SŁOTA

PROJEKT ZAKŁADAŁ NASTĘPUJĄCE ETAPY:

- Dane
- Konfiguracja



Rys 1. NUCLEO-F446RE



Rys 2. Proto Codec 508



Rys 3. Matryca LED 8x8
ze sterownikiem MAX7219

ETAPY REALIZACJI PROJEKTU

Założenia wstępne

- Określenie celów projektu, dobór elementów wykonawczych, wykonanie one-pagera, wybór sterownika.

Oprogramowanie podzespołów

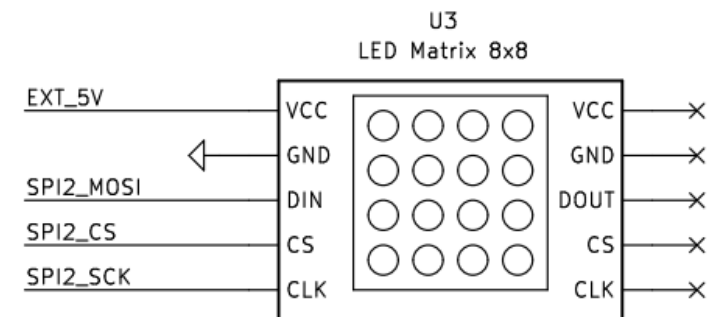
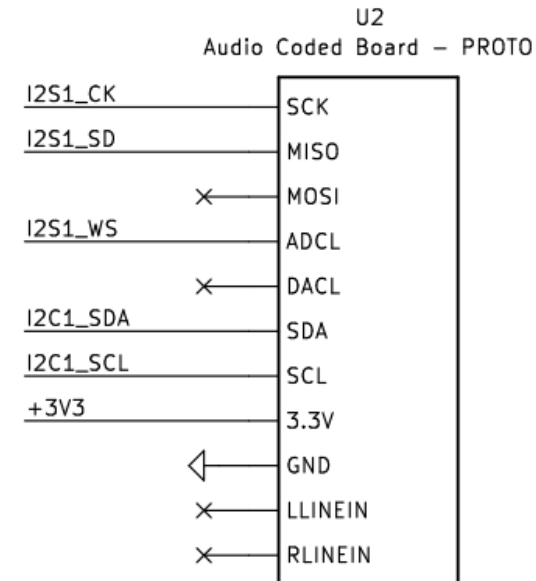
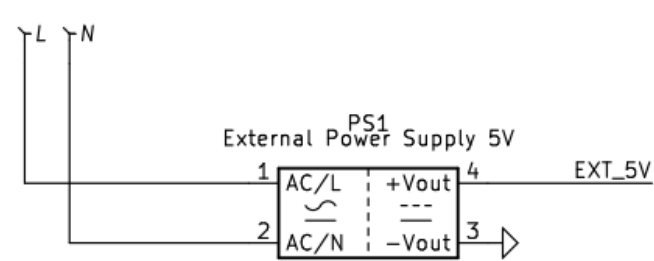
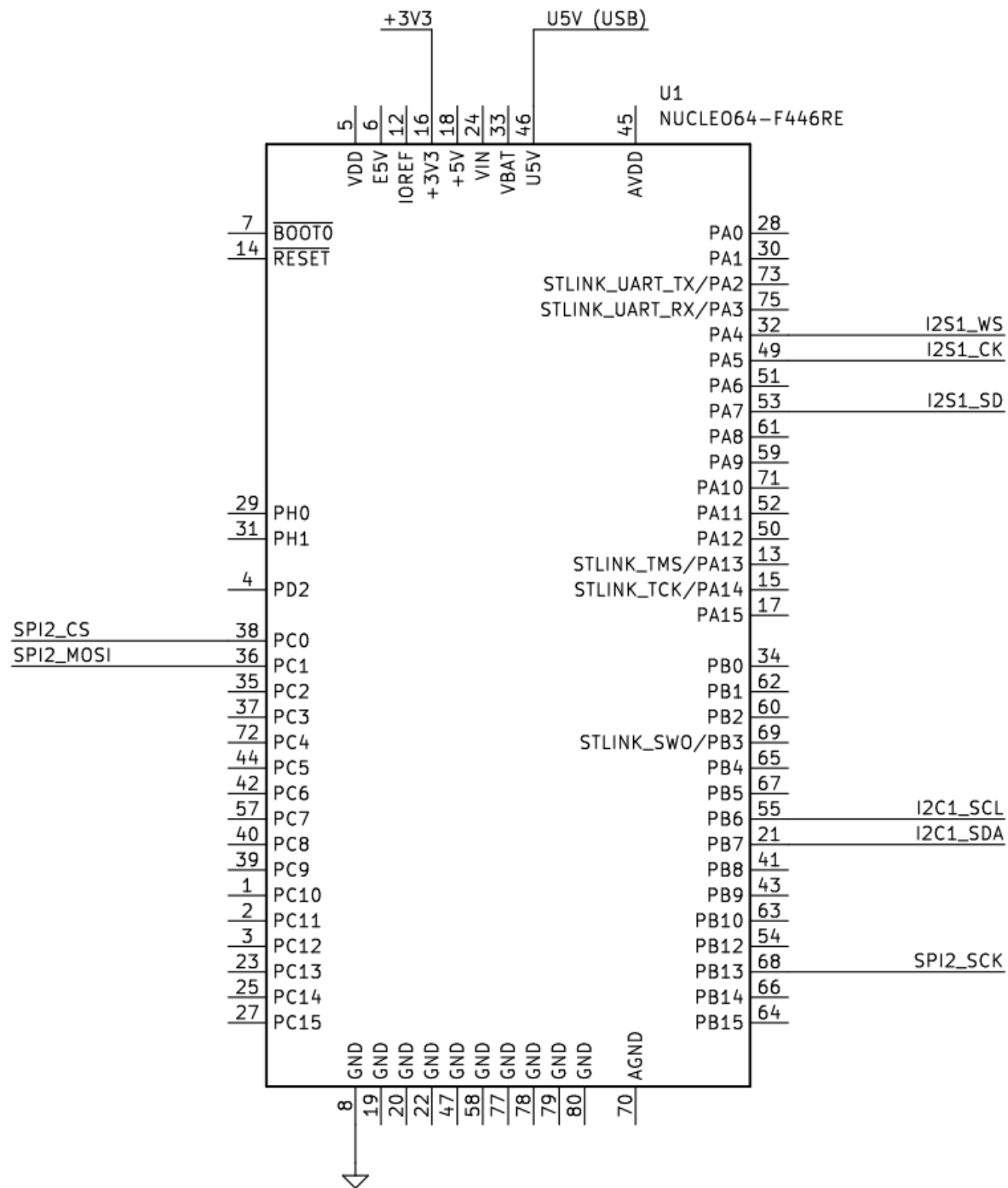
- Przygotowanie bibliotek do obsługi matrycy LED, kontrolera audio oraz obliczeń FFT.

Testy pojedynczych elementów

- Wykonanie serii testów dla matrycy LED, wygenerowanie sygnałów do przebadania FFT, przetestowanie różnych kombinacji ustawień kontrolera.

Walidacja wyników i usprawnienia

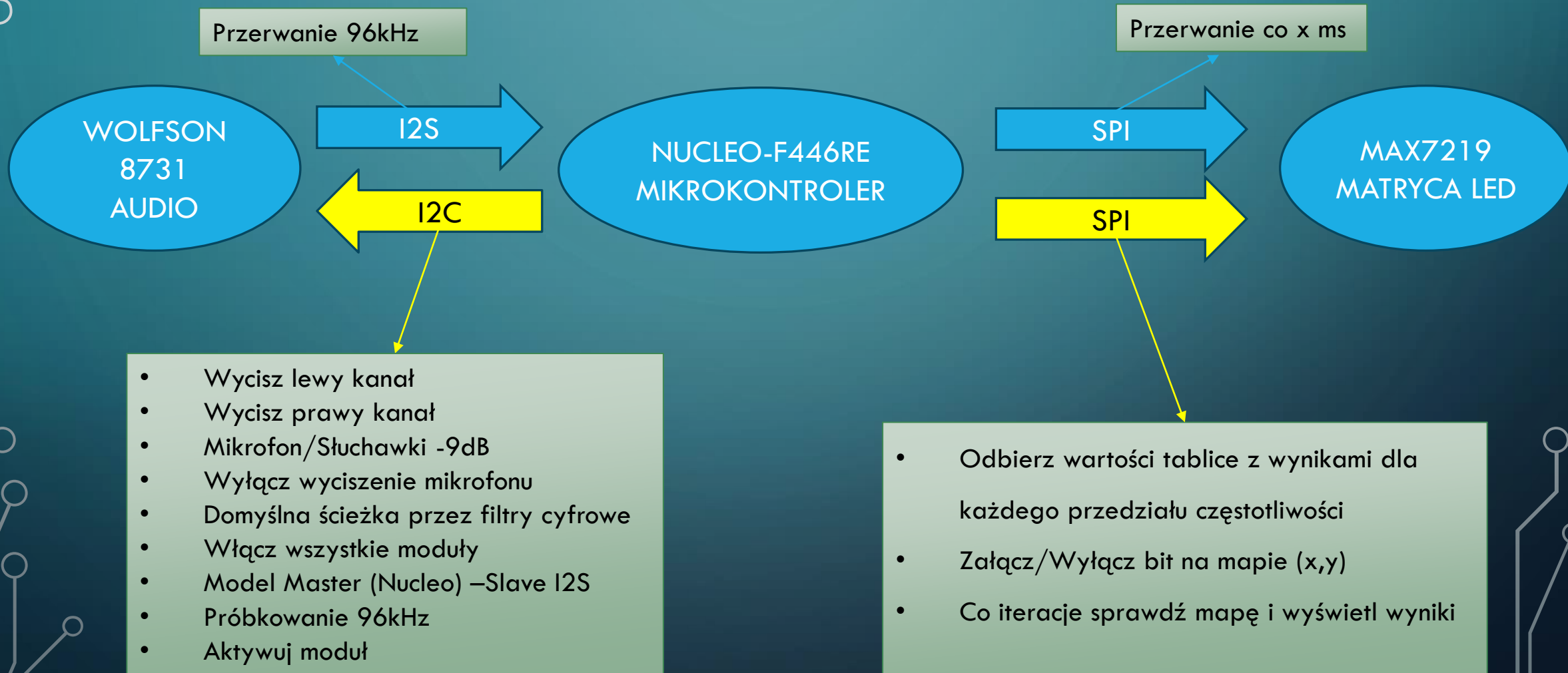
- Połączenie komponentów, rozwiązywanie problemów z komunikacją STM32 – Codec, zwalczanie wpływu szumów na wyniki FFT, implementacja filtrów, eksperymenty z wizualizacją.



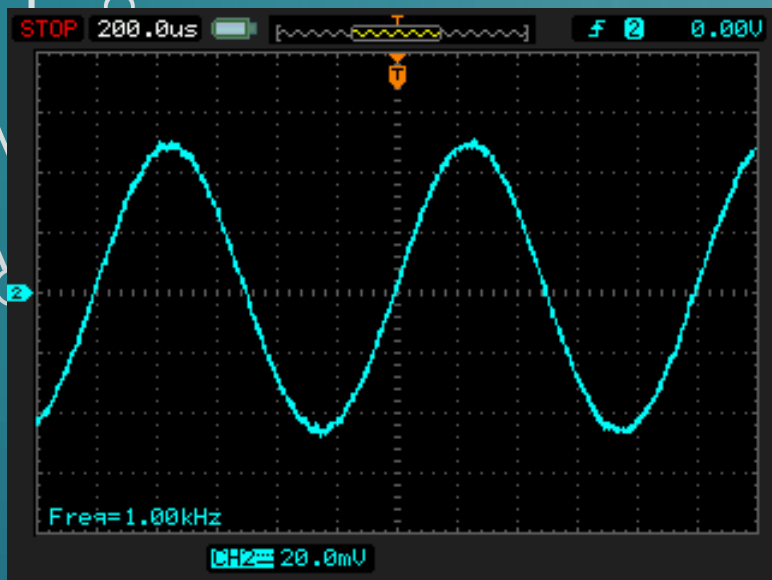
Schemat
połączeń układu

ZDJĘCIE UKŁADU

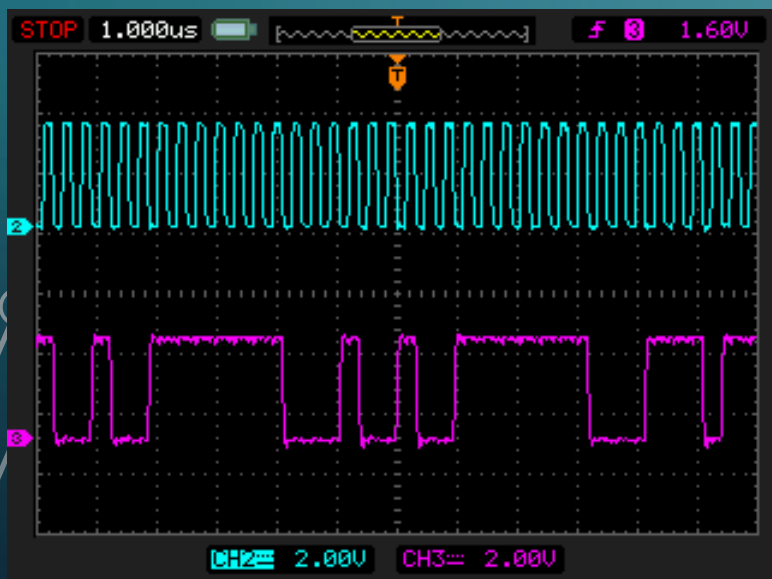
KOMUNIKACJA MIĘDZY STEROWNIKAMI



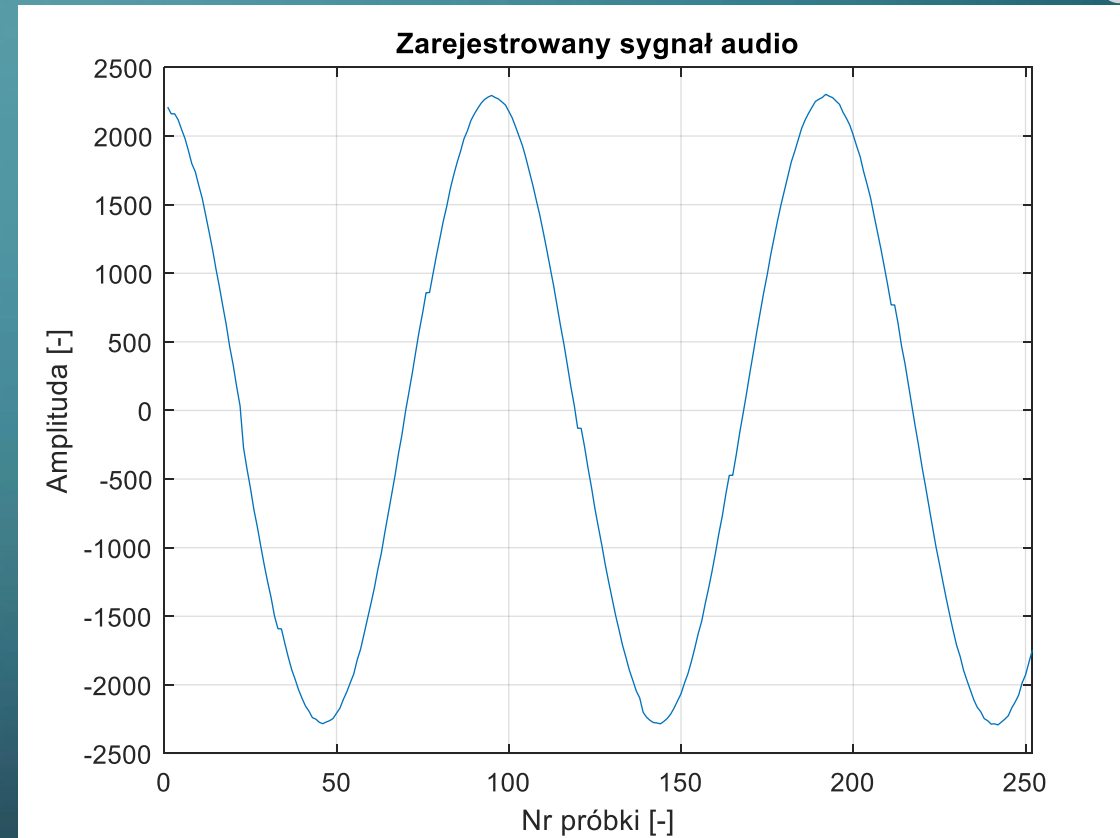
PRZYKŁADOWY ZAREJESTROWANY SYGNAŁ AUDIO



Pomiar sygnału
na wtyku
mini jack

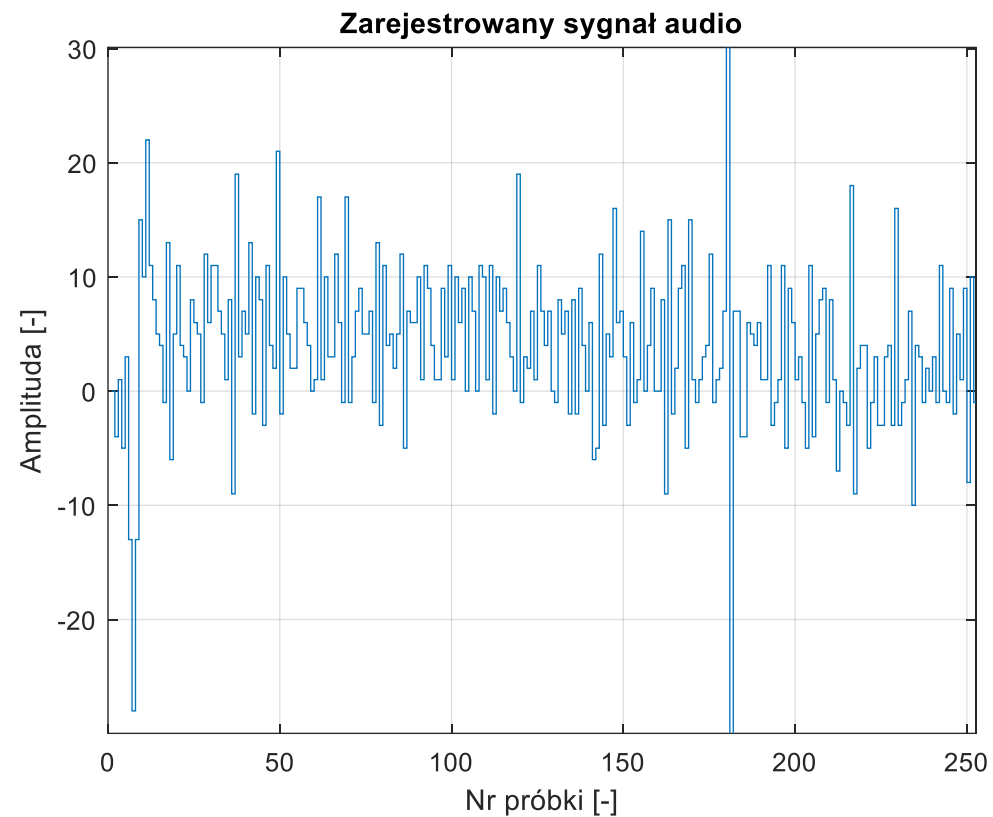
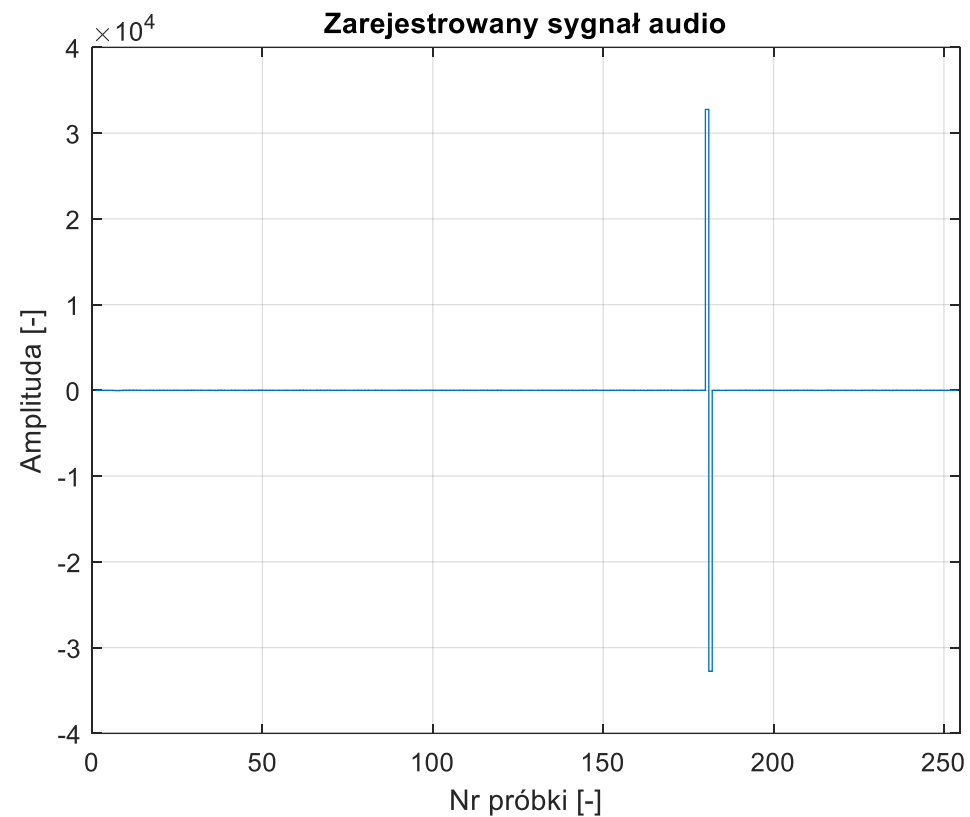


Pomiar sygnału
na liniach
zegarowej (CH2)
oraz danych (CH3)
protokołu I2S



Spróbkowany sygnał audio zapisany w pamięci mikrokontrolera

OBSERWACJA ZAKŁÓCEŃ NA „PUSTEJ LINII”



SPOSOBY OGRANICZENIA SZUMÓW

Osobny zasilacz dla modułu audio i LED

- Stabilizator napięcia na płytce NUCLO może nie być w stanie zapewniać stabilnego napięcia referencyjnego 3.3V dla pomiarów kodeka, ze względu na skokowe zmiany prądu zasilającego matrycę LED

Implementacja filtrów

- Wykorzystanie filtrów po stronie software' prowadzi do otrzymania bardziej wygładzonych przebiegów sygnału audio, w projekcie przetestowano dwa algorytmy: filtr medianowy oraz średniej ruchomej

Zmiana karty dźwiękowej

- Podczas testów okazało się, że część szumów pojawiała się przez wady karty dźwiękowej, co objawiało się występowaniem szumów na nie aktywnej linii. Po podłączeniu słuchawek objawiało się to cichym szarpanym dźwiękiem

ANALIZA CZĘSTOTLIWOŚCIOWA

Algorytm cyfrowego przetwarzania
sygnału audio

Odczyt sygnału z kodeka
i zastosowanie filtracji

Obliczenie transformaty
Fouriera

Wyliczenie modułu
i obliczenie średniej wartości
sygnału w założonych
przedziałach częstotliwości

Wersja klasyczna DFT

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}$$

Wersja rekurencyjna FFT o
zmniejszonej złożoności obliczeniowej

$$X_k = E_k + e^{-\frac{2\pi i}{N}k} O_k$$
$$X_{k+\frac{N}{2}} = E_k - e^{-\frac{2\pi i}{N}k} O_k$$

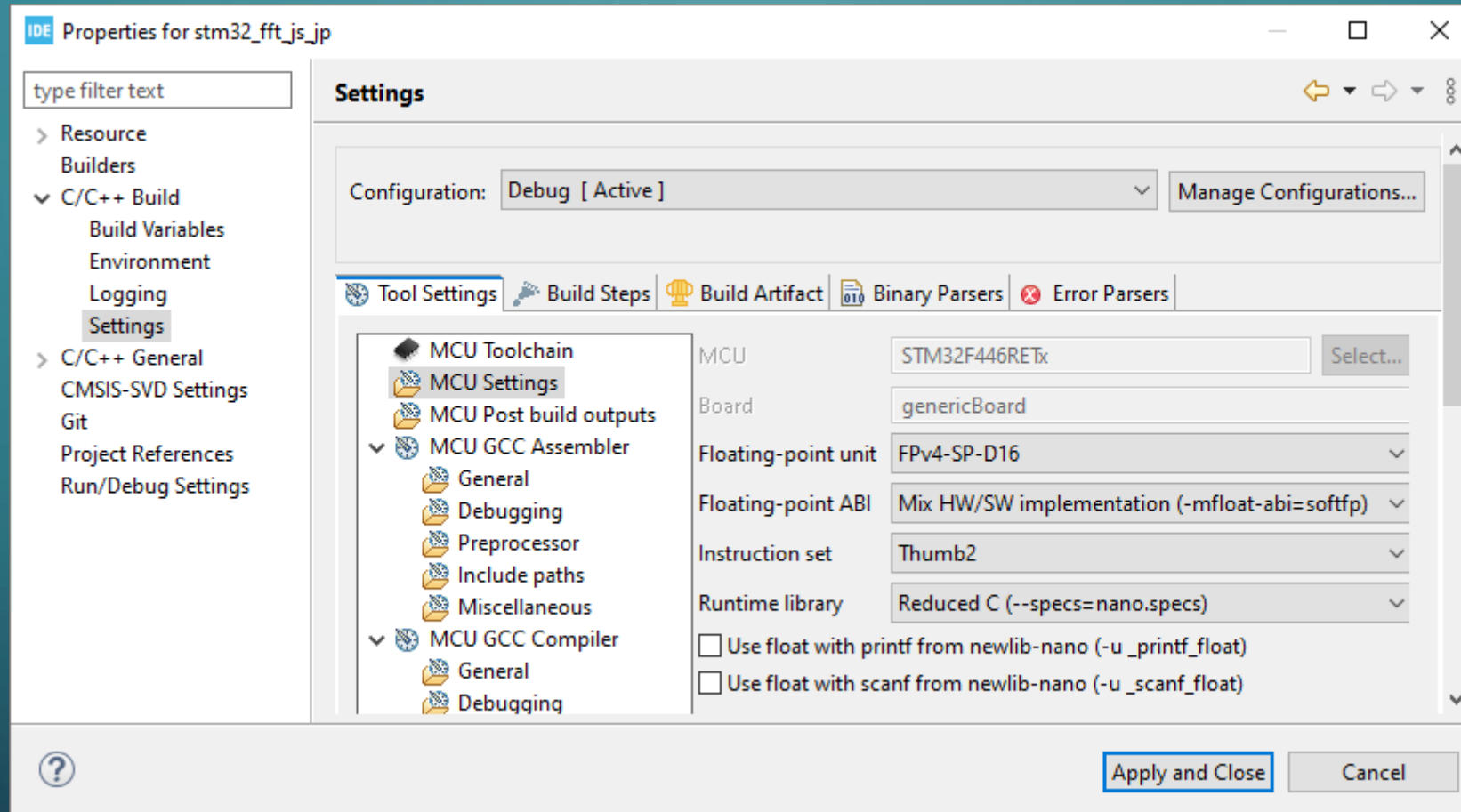
```
static inline void calc_fft(float complex * x, uint16_t N)
{
    if (N <= 1) return;
    else
    {
        float complex even[N/2];
        float complex odd[N/2];

        for (uint16_t i = 0; i < N/2; i++)
        {
            even[i] = x[2*i];
            odd[i] = x[2*i+1];
        }

        calc_fft(even, N/2);
        calc_fft(odd, N/2);

        for (uint16_t k = 0; k < N/2; k++)
        {
            float complex exp_term = cexp(-2.0 * I * M_PI * k / N);
            x[k] = even[k] + exp_term*odd[k];
            x[k+N/2] = even[k] - exp_term*odd[k];
        }
    }
}
```

SPRZĘTOWA OBSŁUGA LICZB ZMIENNOPRZECINKOWYCH



Zamiast programowej emulacji posłużono się sprzętową obsługą notacji zmiennoprzecinkowej przy pomocy modułu FPU dla CORTEX-M4

Testy FFT dla sygnałów sinusoidalnych z danego zakresu:

0 – 2 kHz	2 – 4 kHz	4 – 6 kHz	6 – 8 kHz
$f_s = 1\text{kHz}$	$f_s = 3\text{kHz}$	$f_s = 5\text{kHz}$	$f_s = 7\text{kHz}$
<pre>▼ result_intervals_buffer [0]: 8 [1]: 0 [2]: 0 [3]: 0 [4]: 0 [5]: 0 [6]: 0 [7]: 0 ▼ temp_val: [8] [0]: 11.7349768 [1]: 2.23904276 [2]: 1.32175279 [3]: 1.11696243 [4]: 0.722071767 [5]: 0.602069676 [6]: 0.616483867 [7]: 0.440129459 max_val: 11.7349768</pre>	<pre>▼ result_intervals_buffer [0]: 0 [1]: 8 [2]: 0 [3]: 0 [4]: 0 [5]: 0 [6]: 0 [7]: 0 ▼ temp_val: [8] [0]: 0 [1]: 6.37485409 [2]: 0 [3]: 0 [4]: 0.0223116837 [5]: 0 [6]: 0 [7]: 0.000453354849 max_val: 6.37485409</pre>	<pre>▼ result_intervals_buffer [0]: 1 [1]: 1 [2]: 8 [3]: 1 [4]: 0 [5]: 0 [6]: 0 [7]: 0 ▼ temp_val: [8] [0]: 0.602315843 [1]: 1.91624463 [2]: 10.9698811 [3]: 2.17565894 [4]: 0.99097091 [5]: 0.739819348 [6]: 0.700135112 [7]: 0.476375729 max_val: 10.9698811</pre>	<pre>▼ result_intervals_buffer [0]: 0 [1]: 0 [2]: 1 [3]: 8 [4]: 0 [5]: 0 [6]: 0 [7]: 0 ▼ temp_val: [8] [0]: 0.402157634 [1]: 0.656852901 [2]: 1.84141243 [3]: 11.6184444 [4]: 1.64097667 [5]: 0.963554859 [6]: 0.825703025 [7]: 0.54686594 max_val: 11.6184444</pre>

8 – 10 kHz

$f_s = 9\text{kHz}$

▼ result_intervals_buffer

[0]: 0
[1]: 0
[2]: 0
[3]: 0
[4]: 8
[5]: 0
[6]: 0
[7]: 0

▼ temp_val: [8]

[0]: 0
[1]: 0.0133516071
[2]: 0
[3]: 0
[4]: 6.37485361
[5]: 0
[6]: 0
[7]: 0.00230152695

max_val: 6.37485361

10 – 12 kHz

$f_s = 11\text{kHz}$

▼ result_intervals_buffer

[0]: 0
[1]: 0
[2]: 0
[3]: 0
[4]: 1
[5]: 8
[6]: 1
[7]: 0

▼ temp_val: [8]

[0]: 0.316879928
[1]: 0.301747799
[2]: 0.439862192
[3]: 0.949495971
[4]: 2.164294
[5]: 10.9435644
[6]: 1.99908257
[7]: 0.881401122

max_val: 10.9435644

12 – 14kHz

$f_s = 13\text{kHz}$

▼ result_intervals_buffer

[0]: 0
[1]: 0
[2]: 0
[3]: 0
[4]: 0
[5]: 1
[6]: 8
[7]: 0

▼ temp_val: [8]

[0]: 0.311269224
[1]: 0.269047171
[2]: 0.336930037
[3]: 0.589497209
[4]: 0.834981382
[5]: 1.98812509
[6]: 11.5782309
[7]: 1.55217481

max_val: 11.5782309

14 – 16 kHz

$f_s = 15\text{kHz}$

▼ result_intervals_buffer

[0]: 0
[1]: 0
[2]: 0
[3]: 0
[4]: 0
[5]: 0
[6]: 0
[7]: 8

▼ temp_val: [8]

[0]: 0
[1]: 0.00304831704
[2]: 0
[3]: 0
[4]: 0.000976331707
[5]: 0
[6]: 0
[7]: 6.37485361

max_val: 6.37485361

Sygnał będący sumą sinusów z każdego zakresu:

▼ result_abs_buffer

[0]:	86
[1]:	1129.81189
[2]:	4467.02344
[3]:	11212.6045
[4]:	3256.03345
[5]:	2094.95068
[6]:	1635.25208
[7]:	1408.92664
[8]:	11473.1074
[9]:	1297.24731
[10]:	1405.52417
[11]:	1733.45056
[12]:	2740.12769
[13]:	10528.9355
[14]:	5496.37988
[15]:	2560.24268
[16]:	2149.96777
[17]:	2598.07178
[18]:	5572.4292
[19]:	10410.6357
[20]:	2580.802
[21]:	1529.16931
[22]:	1152.03064

1125 Hz

3000 Hz

4875 Hz

7125 Hz

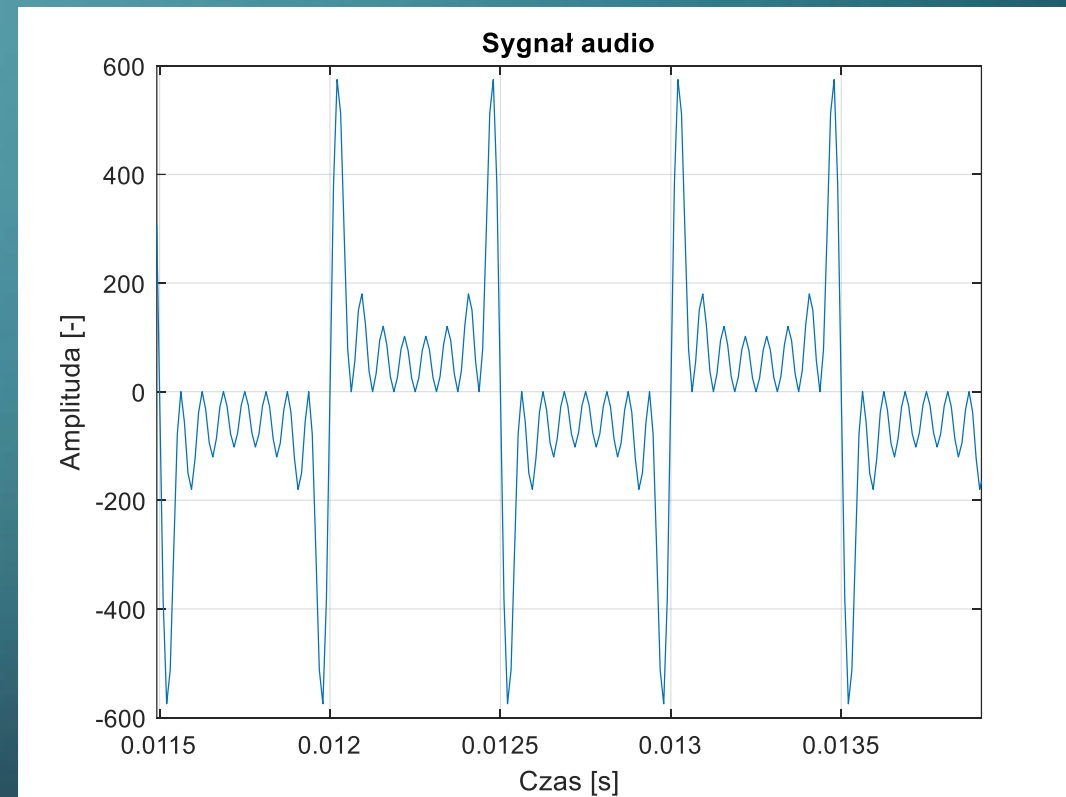
[23]:	988.798035
[24]:	11865.3867
[25]:	964.427063
[26]:	1102.96094
[27]:	1454.75879
[28]:	2480.11475
[29]:	10282.4863
[30]:	5729.86816
[31]:	2786.77368
[32]:	2372.87109
[33]:	2821.4209
[34]:	5803.02637
[35]:	10191.9922
[36]:	2339.21899
[37]:	1265.96375
[38]:	858.031372
[39]:	649.241821
[40]:	12271.8496
[41]:	443.3927
[42]:	386.614166
[43]:	345.42334

9000 Hz

10875 Hz

13125 Hz

15000 Hz



$$\frac{i_{dx}}{okno_FFT} * \text{częstotliwość próbkowania}$$

IDEA WIZUALIZACJI WYNIKÓW FFT NA MATRYCY LED

[0] 0 – 2.5k [4] 10k – 12.5k
[1] 2.5k – 5k [5] 12.5k – 15k
[2] 5k – 7.5k [6] 15k – 17.5k
[3] 7.5k – 10k [7] 17.5k – 20k [Hz]

Zakresy częstotliwości

0	1	2	3	4	5	6	7

Sygnał 5kHz
(o czym początkowo nie wiemy)

Moduł Audio Codec

NUCLEO-F446RE

Wektor przynależności sygnału o nieznannej częstotliwości do danego przedziału

0	0	8	1	0	0	0	0
---	---	---	---	---	---	---	---

[0] 0 – 2.5k	[4] 10k – 12.5k
[1] 2.5k – 5k	[5] 12.5k – 15k
[2] 5k – 7.5k	[6] 15k – 17.5k
[3] 7.5k – 10k	[7] 17.5k – 20k [Hz]

Zakresy częstotliwości

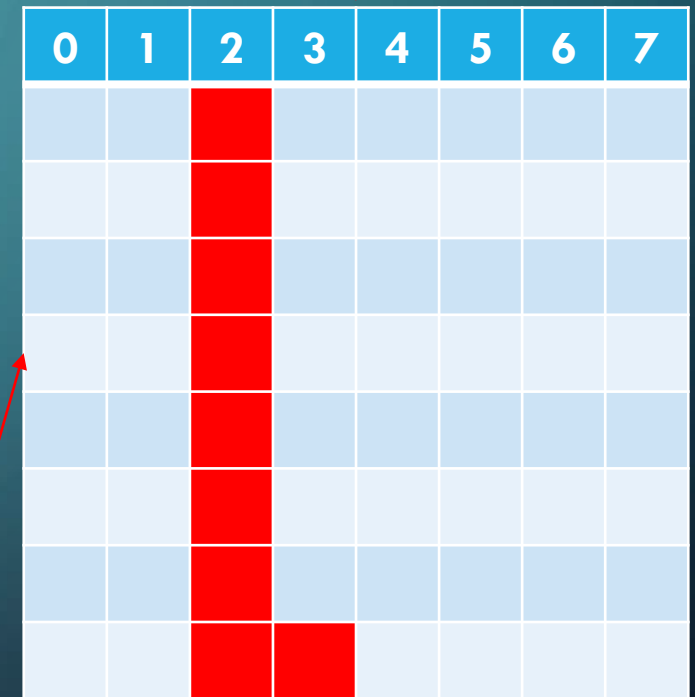
Sygnał 5kHz
(o czym początkowo
nie wiemy)

Moduł Audio Codec

NUCLEO-
F446RE

Wektor przynależności sygnału
o nieznannej częstotliwości do
danego przedziału

0	0	8	1	0	0	0	0
---	---	---	---	---	---	---	---



MATRYCA LED

Fota

```
MAX7219_STATUS MAX7219_SetSpectrum(int* spectrumTab, int numOfRanges)
{
    for(int i = 0; i < numOfRanges; i++)
    {
        for(int j = 0; j < spectrumTab[i]; j++)
        {
            if(MAX7219_SetPixel(i, j, MAX7219_ON) != MAX7219_OK)
                return MAX7219_ERROR;
        }
    }
    return MAX7219_OK;
}
```

Wywołanie FCN