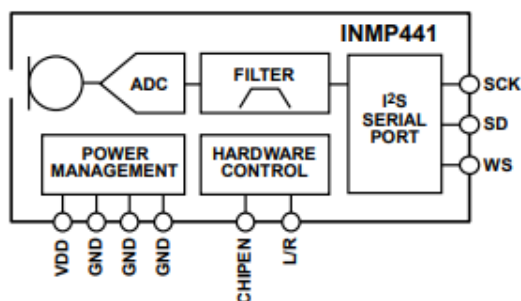


DSP

▼ Etap 1 - Dźwięk Cyfrowy

Moduł z Mikrofonem MEMS INMP 441- wyposażony w przetwornik Analogowo-Cyfrowy, filtry antyaliasingowe, I2S Port - sygnał gotowy do obróbki



Etap 2 Filtracja DSP

filtr pasmowoprzepustowy do odfiltrowania dźwięku piły -

25 Hz - 10,5 kHz, najczęściej 110 Hz - 2,5 kHz

<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.775173/full>

„the spectral composition of the noise emitted by the chainsaws with the dominance of components in frequency bands between **500 Hz and 8 kHz**”

```
from scipy.signal import butter, lfilter
```

```
def bandpass_filter(data, lowcut, highcut, fs, order=5):  
    nyq = 0.5 * fs  
    low = lowcut / nyq  
    high = highcut / nyq  
    b, a = butter(order, [low, high], btype='band')  
    y = lfilter(b, a, data)  
    return y
```

Etap 3 Analiza Widmowa - dopasowanie do wzorca

```

import numpy as np
from scipy.signal import correlate
import librosa

ref, sr = librosa.load('chainsaw_ref.wav', sr=None)
test, _ = librosa.load('recorded_sound.wav', sr=sr)

ref_f = bandpass_filter(ref, 500, 8000, sr)
test_f = bandpass_filter(test, 500, 8000, sr)

corr = correlate(test_f, ref_f, mode='valid')
corr_norm = corr / (np.linalg.norm(test_f) * np.linalg.norm(ref_f))

if np.max(corr_norm) > 0.3:
    print("Wykryto dźwięk piły!")
else:
    print("Brak piły.")

```

Implementacja:

- Python:
 - sounddevice, audiopy, librosa → dostęp do dźwięku
 - scipy.signal
 - numpy
 - noisereduce