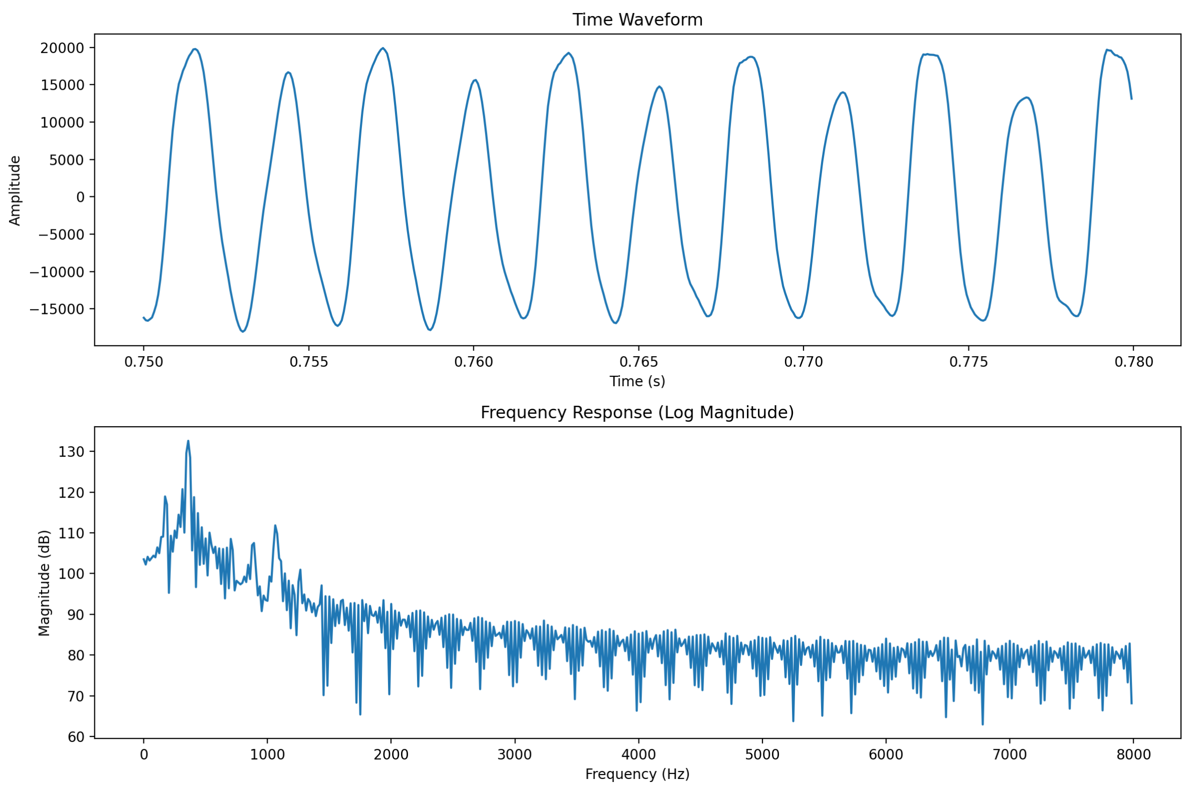
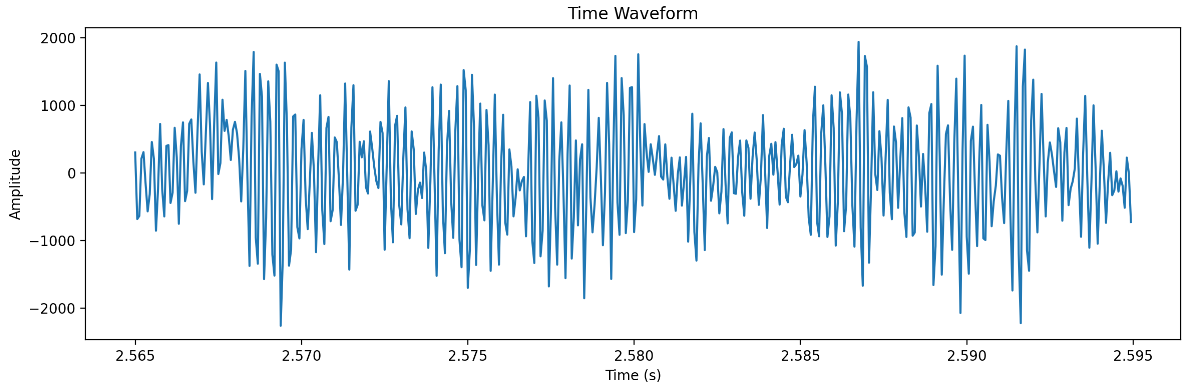
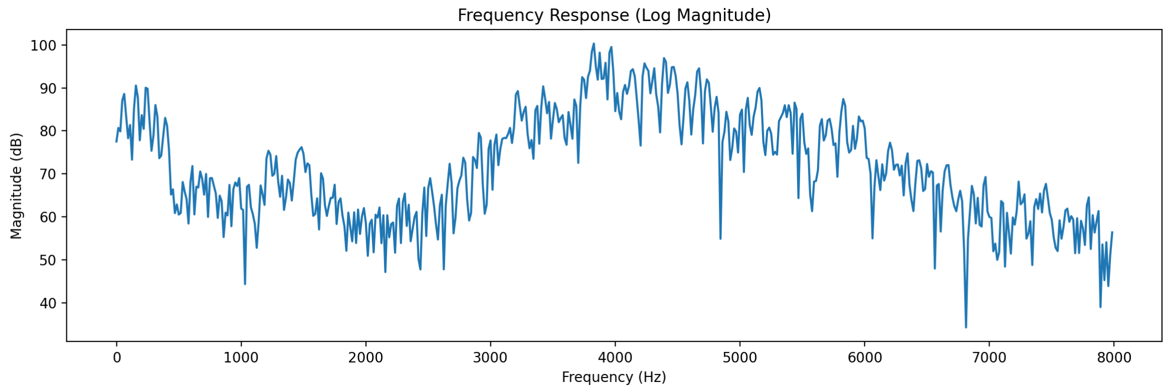
In [19]:
```
# Import necessary libraries
import requests
from bs4 import BeautifulSoup

# Function to get the note frequencies from the given URL
def get_note_frequencies(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    table = soup.find('center')
    rows = table.find_all('tr')
    note_freqs = {}
    for row in rows[1:]:
        cols = row.find_all('td')
        note = cols[0].text
        freq = float(cols[1].text)
        note_freqs[note] = freq
    return note_freqs

# Function to find the closest note for a given frequency
def find_closest_note(freq, note_freqs):
    closest_note = min(note_freqs.keys(), key=lambda note: abs(freq - note_freq
    return closest_note

# Get the note frequencies
note_freqs = get_note_frequencies('https://pages.mtu.edu/~suits/notefreqs.html'

# Find the fundamental frequency and corresponding note for signal_b
fund_freq_b = freq_b[np.argmax(np.abs(xhf_b))]
closest_note_b = find_closest_note(fund_freq_b, note_freqs)
print(f'The fundamental frequency for signal_b is {fund_freq_b} Hz, which corre

# Find the fundamental frequency and corresponding note for signal_c
fund_freq_c = freq_c[np.argmax(np.abs(xhf_c))]
closest_note_c = find_closest_note(fund_freq_c, note_freqs)
print(f'The fundamental frequency for signal_c is {fund_freq_c} Hz, which corre

# Plot the log magnitude of the frequency response for signal_b
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.plot(np.arange(len(signal_b)) / fs + start_time_b, signal_b)
plt.title('Time Waveform')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.subplot(2, 1, 2)
plt.plot(freq_b[:nfft//2], 20*np.log10(np.abs(xhf_b[:nfft//2])))
plt.title('Frequency Response (Log Magnitude)')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude (dB)')
plt.tight_layout()
plt.show()

# Plot the log magnitude of the frequency response for signal_c
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.plot(np.arange(len(signal_c)) / fs + start_time_c, signal_c)
plt.title('Time Waveform')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.subplot(2, 1, 2)
```

```
plt.plot(freq_c[:nfft//2], 20*np.log10(np.abs(xhf_c[:nfft//2])))
plt.title('Frequency Response (Log Magnitude)')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude (dB)')
plt.tight_layout()
plt.show()
```

```
The fundamental frequency for signal_b is -359.375 Hz, which corresponds to th
e note C0.
The fundamental frequency for signal_c is -3828.125 Hz, which corresponds to t
he note C0.
```




Time Waveform
Amplitude
Time (s)
2000
1000
0
−1000
−2000
2.565
2.570
2.575
2.580
2.585
2.590
2.595


Frequency Response (Log Magnitude)
Magnitude (dB)
Frequency (Hz)
100
90
80
70
60
50
40
0
1000
2000
3000
4000
5000
6000
7000
8000

In [ ]: