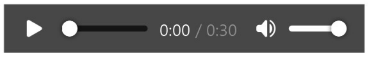
In [1]:
```
import librosa
import librosa.display
import IPython.display as ipd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
debussy_file = "audio/debussy.wav"
redhot_file = "audio/redhot.wav"
duke_file = "audio/duke.wav"
```

In [3]:
```
ipd.Audio(debussy_file)
```

Out[3]:



In [4]:
```
ipd.Audio(redhot_file)
```

Out[4]:

In [5]:
```
ipd.Audio(duke_file)
```

Out[5]:

In [6]:
```
debussy, sr = librosa.load(debussy_file)
redhot, _ = librosa.load(redhot_file)
duke, _ = librosa.load(duke_file)
```

In [7]:
```
debussy
```

Out[7]:
```
array([-0.01742554, -0.03567505, -0.04995728, ...,  0.00912476,
        0.00866699,  0.00964355], dtype=float32)
```

In [8]:
```
debussy.size
```

Out[8]:
```
661500
```

In [9]:
```
sample_duration = 1/sr
print(f"duration of one sample is {sample_duration:.6f}")
```

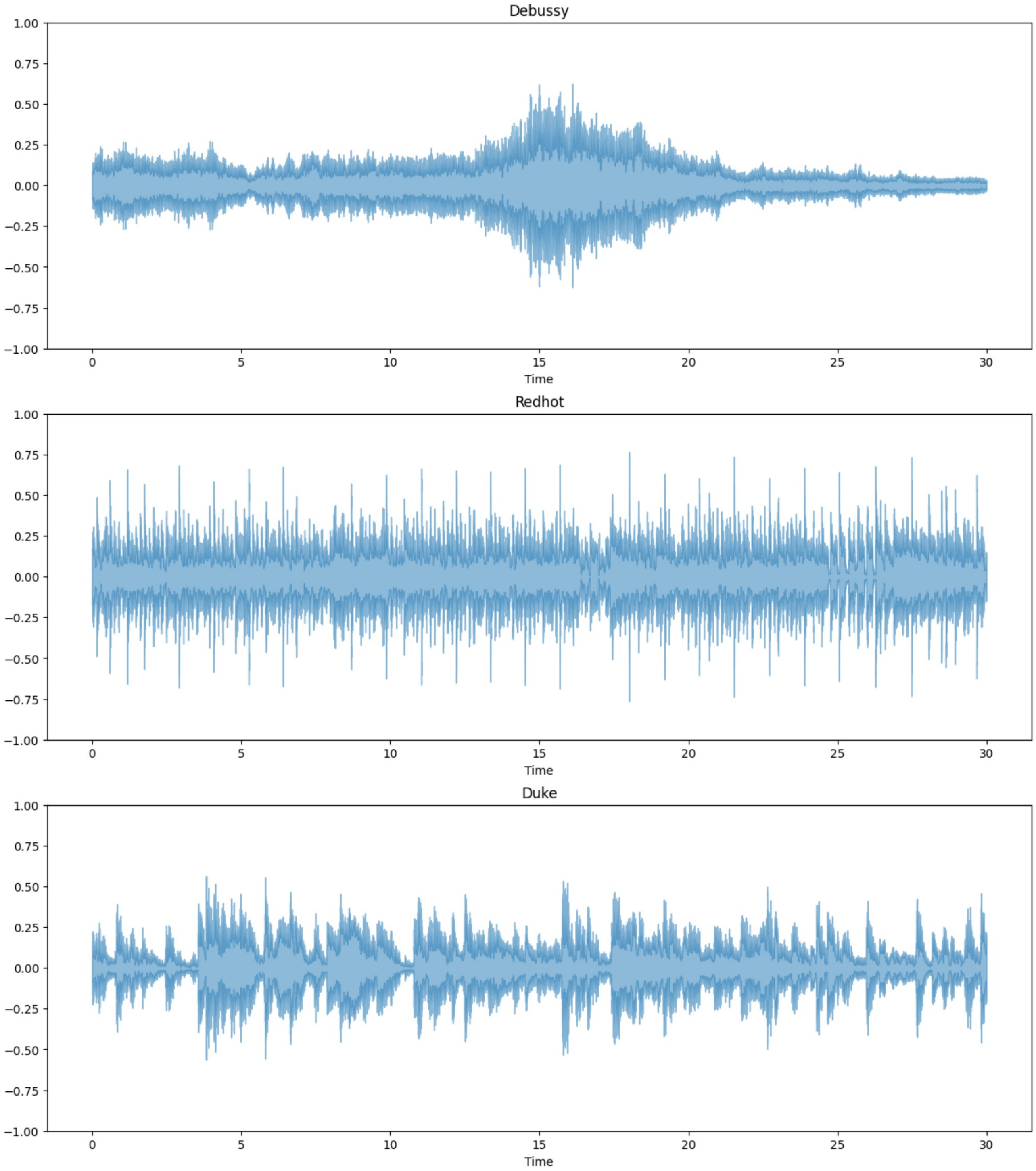
```
duration of one sample is 0.000045
```

In [10]:
```
duration = sample_duration * len(debussy)
print(f"duration of audio is {duration:.6f}")
```

```
duration of audio is 30.000000
```

In [11]:
```
plt.figure(figsize=(15,17))

plt.subplot(3,1,1)
librosa.display.waveshow(debussy, alpha=0.5)
plt.title("Debussy")
plt.ylim(-1,1)

plt.subplot(3,1,2)
librosa.display.waveshow(redhot, alpha=0.5)
plt.title("Redhot")
plt.ylim(-1,1)

plt.subplot(3,1,3)
librosa.display.waveshow(duke, alpha=0.5)
plt.title("Duke")
plt.ylim(-1,1)

# , alpha=0.5 for transparemcy
plt.show
```

Out[11]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [32]:

```
FRAME_SIZE = 1024
HOP_LENGTH = 512

def amplitude_envelope(signal, frame_size, hop_length):
    amplitude_envelope = []

    for i in range(0,len(signal),hop_length):
        current_frame_amplitude_envelope = max(signal[i:i+frame_size])
        amplitude_envelope.append(current_frame_amplitude_envelope)
    return np.array(amplitude_envelope)
```

In [33]:

```
def fancy_amplitude_envelope(signal, frame_size, hop_length):
    """Fancier Python code to calculate the amplitude envelope of a signal with a given frame size."""
    return np.array([max(signal[i:i+frame_size]) for i in range(0, len(signal), hop_length)])
```

In [34]:

```
ae_debussy = amplitude_envelope(debussy,FRAME_SIZE, HOP_LENGTH)
len(ae_debussy)
```

Out[34]:

```
1292
```

In [35]:

```
fancy_ae_debussy = fancy_amplitude_envelope(debussy,FRAME_SIZE, HOP_LENGTH)
len(ae_debussy)
```

Out[35]:

```
1292
```

In [36]:

```
(ae_debussy == fancy_ae_debussy).all()
```

Out[36]:

```
True
```

In [37]:

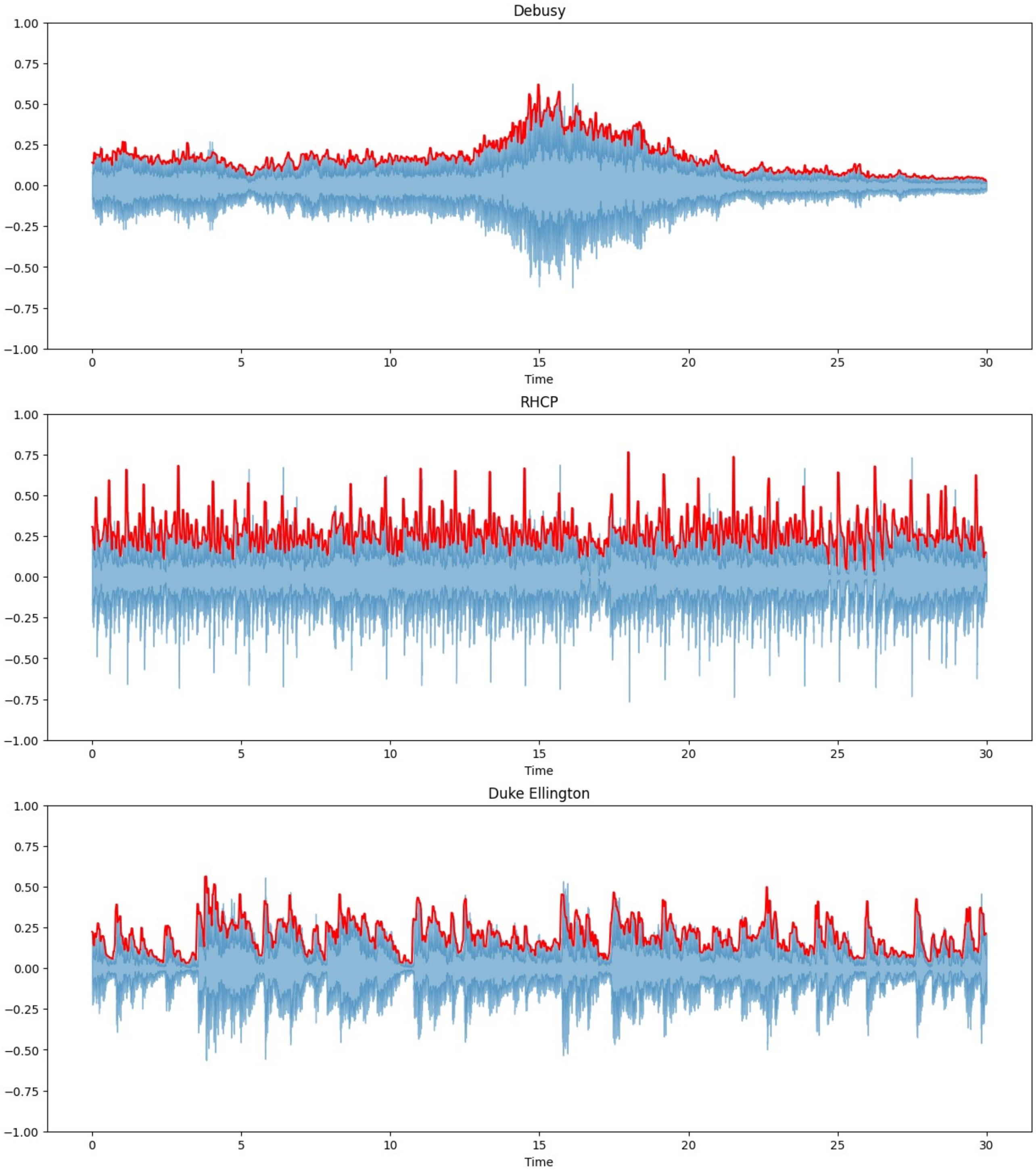
```
# calculate amplitude envelope for RHCP and Duke Ellington
ae_redhot = amplitude_envelope(redhot, FRAME_SIZE, HOP_LENGTH)
ae_duke = amplitude_envelope(duke, FRAME_SIZE, HOP_LENGTH)
```

In [38]:

```
frames = range(len(ae_debussy))
t = librosa.frames_to_time(frames, hop_length=HOP_LENGTH)
```

In [39]:

```
# amplitude envelope is graphed in red

plt.figure(figsize=(15, 17))

ax = plt.subplot(3, 1, 1)
librosa.display.waveshow(debussy, alpha=0.5)
plt.plot(t, ae_debussy, color="r")
plt.ylim((-1, 1))
plt.title("Debusy")

plt.subplot(3, 1, 2)
librosa.display.waveshow(redhot, alpha=0.5)
plt.plot(t, ae_redhot, color="r")
plt.ylim((-1, 1))
plt.title("RHCP")

plt.subplot(3, 1, 3)
librosa.display.waveshow(duke, alpha=0.5)
plt.plot(t, ae_duke, color="r")
plt.ylim((-1, 1))
plt.title("Duke Ellington")

plt.show()
```



Debusy

RHCP

Duke Ellington