In [26]:

```
import numpy as np
import scipy as sp
import scipy.signal as sg
import librosa as lib
import matplotlib.pyplot as plt
import random
```

In [27]:

```
# Part 5
```

In [28]:

```
# Part 5.1
```

In [29]:

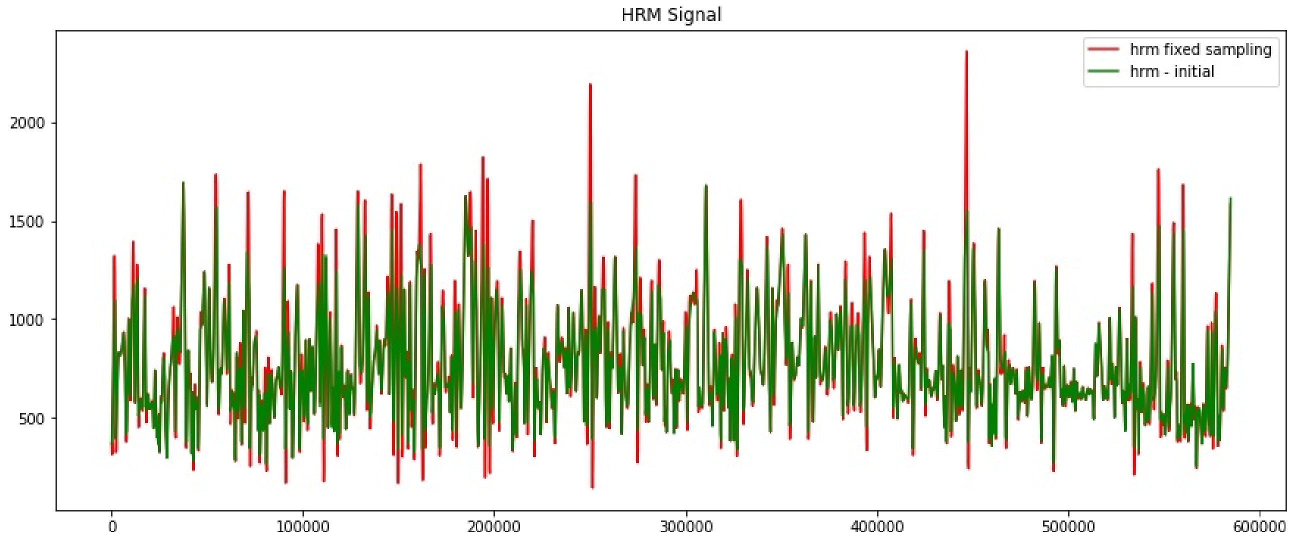
```
# loading of signal
hrm_orig = np.load('hrm_orig.npy')
# sampling times
time_samples = np.cumsum(hrm_orig)
# desired sampling rate
fs1 = 5

freq1 = sp.interpolate.interp1d(time_samples, hrm_orig, kind='cubic')
# multiplication beacuse time_samples is in msecs
tnew = np.arange(min(time_samples), max(time_samples), (1/fs1)*1e3)
hrm = freq1(tnew)
```

In [30]:

```
# Signal Depiction

plt.figure(figsize=(15,6))
plt.plot(tnew, hrm, color = 'r')
plt.plot(t_orig, hrm_orig, color = 'g')
plt.title('HRM Signal')
plt.legend(["hrm fixed sampling","hrm - initial"]);
```



In [31]:

```
# Part 5.2
```

In [32]:

```
# number of samples
N = 1000
# sampling time
tn = np.zeros(N)
# noise
noise = np.zeros(N)

for i in range(N):
    tn[i] = random.randint(1, 100)
    noise[i] = random.uniform(0,0.1)

tn = np.sort(tn)
x_t = np.sin(2*np.pi*tn) + np.cos(2*np.pi*tn/3) + noise

#interpolation
f2 = sp.interpolate.interp1d(tn, x_t, kind='linear')
fs2 = 10
tnew = np.arange(min(tn), max(tn), (1/fs2))
x = f2(tnew)
```

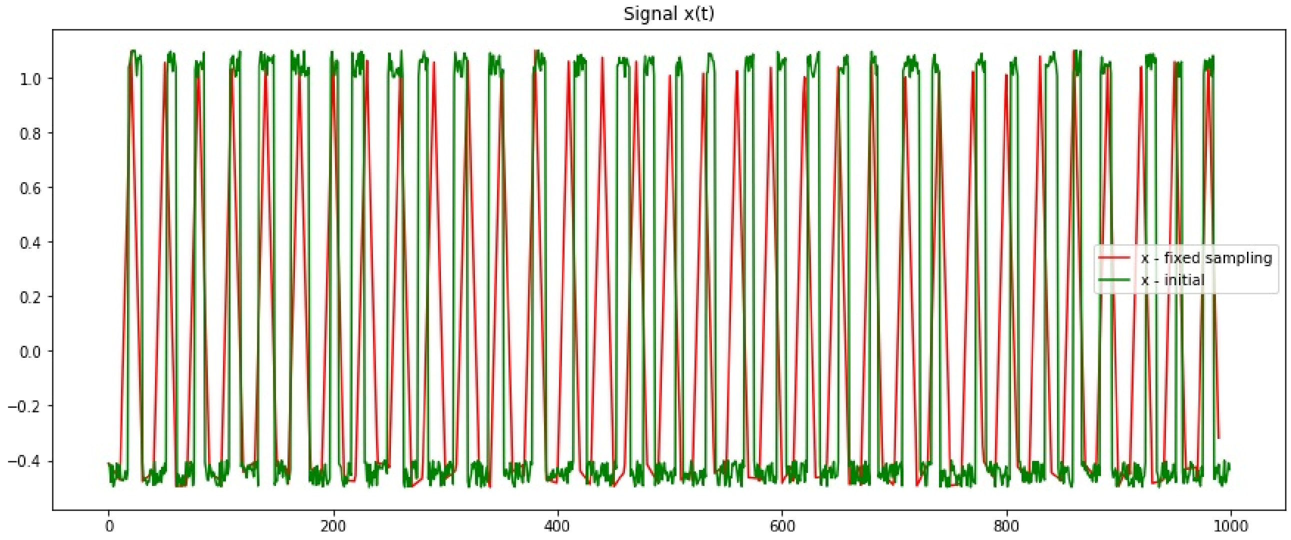
In [33]:

```
# Signal Depiction

plt.figure(figsize=(15,6))
plt.plot(x, color='r')
plt.plot(x_t, color='g')
plt.title('Signal x(t)')
plt.legend(["x - fixed sampling","x - initial"])
```

Out[33]:

```
<matplotlib.legend.Legend at 0x21f178a4948>
```



In [34]:

```
# Part 5.3
```

In [35]:

```
# Schuster Periodogram

def periodogram(signal):
    return (abs(np.fft.fft(signal))**2) / len(signal)

Pper = [periodogram(hrm), periodogram(x)]
```

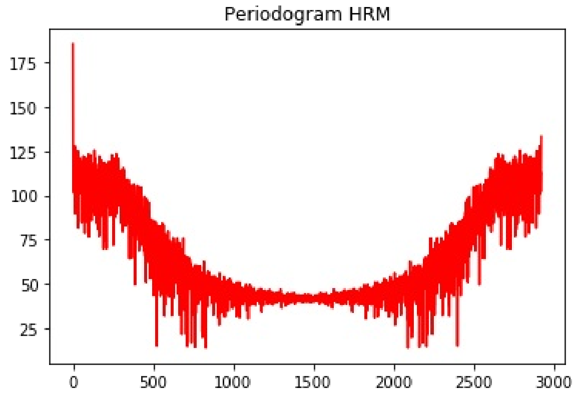
In [38]:

```
# Periodogram Depiction

plt.plot(20*np.log10(Pper[0]), color='r')
plt.title("Periodogram HRM")
```

Out[38]:

```
Text(0.5, 1.0, 'Periodogram HRM')
```

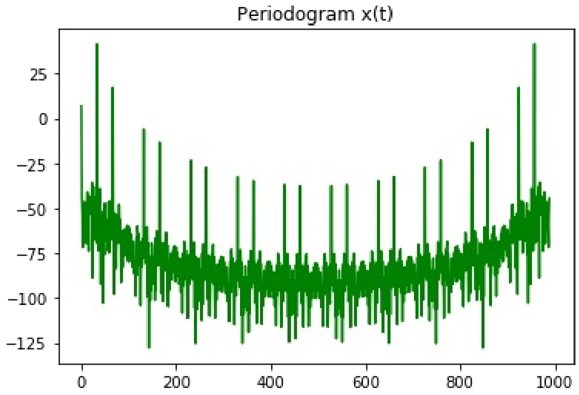


In [39]:

```
plt.plot(20*np.log10(Pper[1]), color='g')
plt.title("Periodogram x(t)")
```

Out[39]:

```
Text(0.5, 1.0, 'Periodogram x(t)')
```



In [40]:

```
# Part 5.4
```

In [44]:

```
# Welch Periodogram

D = 50
L = 100
fs = [fs1, fs2]
signals = [hrm, x]
Pper = []

for i in range(2):
    frame_samples = L*fs[i]
    frames = int((len(signals[i])/frame_samples)*L/D)
    pper = []

    # Bartlett window
    wb = np.bartlett(frame_samples)

    for j in range(frames-1):
        start = j * int(frame_samples/2)
        finish = start + frame_samples

        pper.append(periodogram(wb * signals[i][start:finish]))

    Pper.append(sum(pper)/frames)
```

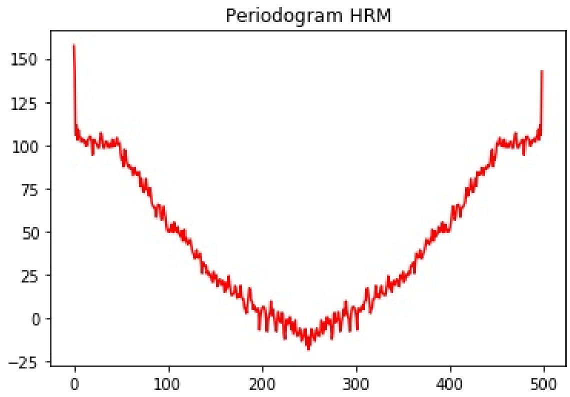
In [45]:

```
# Periodogram Depiction

plt.plot(1,2,1)
plt.plot(20*np.log10(Pper[0]), color='r')
plt.title("Periodogram HRM")
```

Out[45]:

```
Text(0.5, 1.0, 'Periodogram HRM')
```


Periodogram HRM