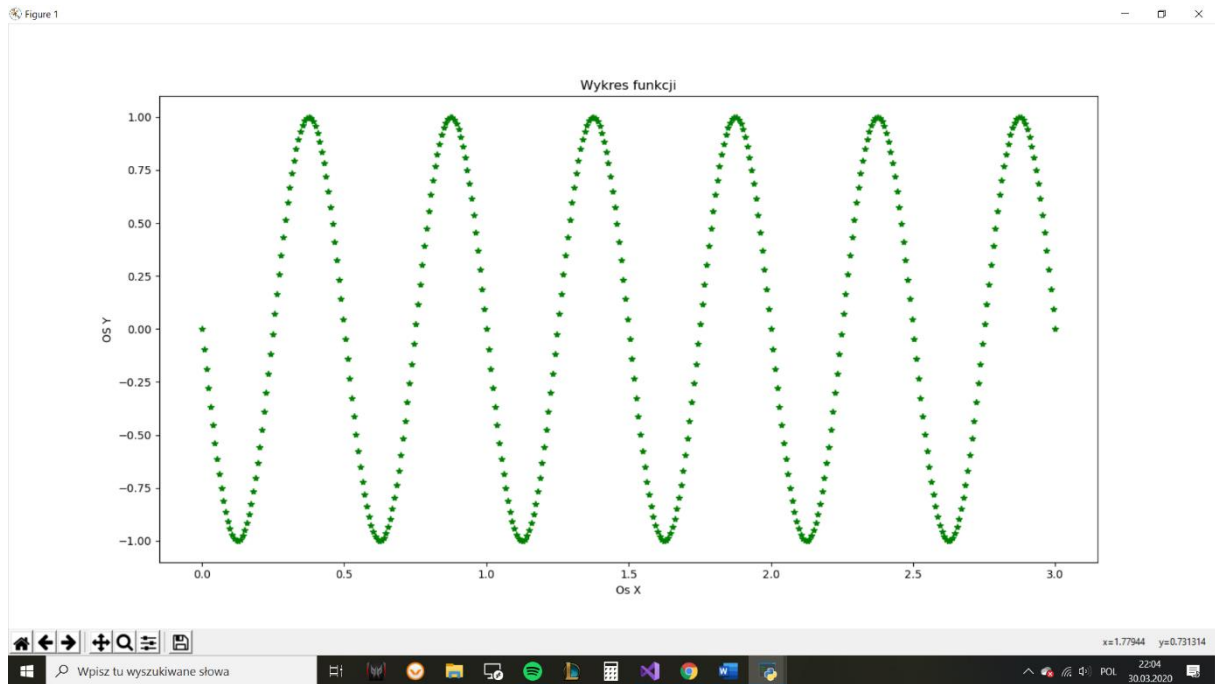


Sprawozdanie Lab02

Mateusz Proc 20C N1 nr 45123

Zadanie 1/.

Wykres tonu prostego



Treść kodu:

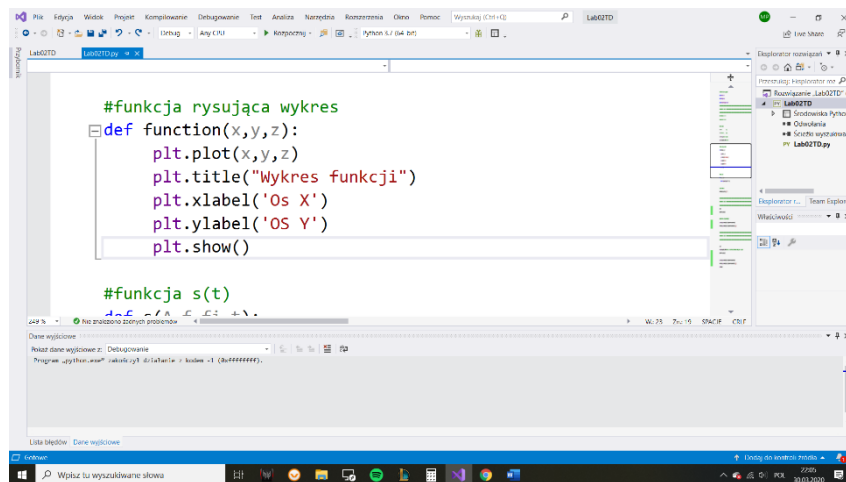
Podanie danych potrzebnych do zdania.

```
#####
##nr indeksu 4 5 1 2 3
##oznaczenia E D C B A

#Dane do zadań
A= 1.0 # [V]
f= 2 # dla A=2 , [HZ]
fi= 1 * np.pi #dla C =1 [rad]
t=np.linspace(0,3,400)# dla A=3

#funkcja wywołana w kodzie
```

Utworzenie funkcji rysującej wykresy.

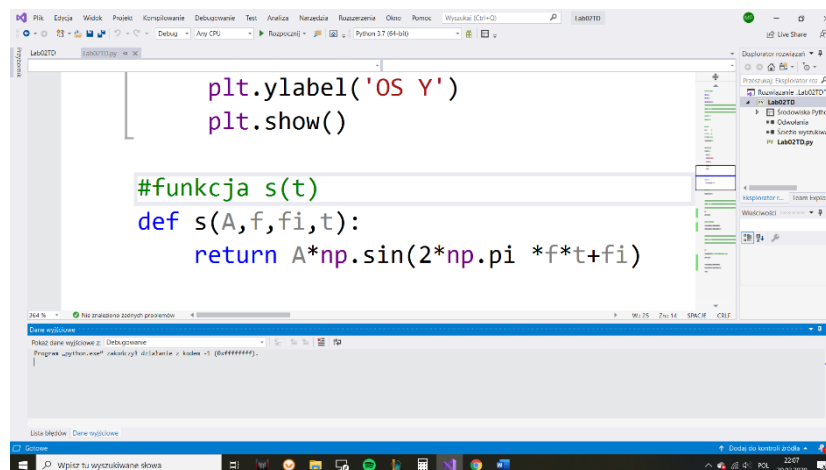


```
#funkcja rysująca wykres
def function(x,y,z):
    plt.plot(x,y,z)
    plt.title("Wykres funkcji")
    plt.xlabel('Os X')
    plt.ylabel('Os Y')
    plt.show()

#funkcja s(t)
```

The screenshot shows the Visual Studio Code interface with a Python file named 'Lab02TD.py'. The code defines a 'function' that plots a graph and a placeholder for 's(t)'. The interface includes a file explorer on the right showing the project structure and a terminal at the bottom.

Utworzenie funkcji s(t).

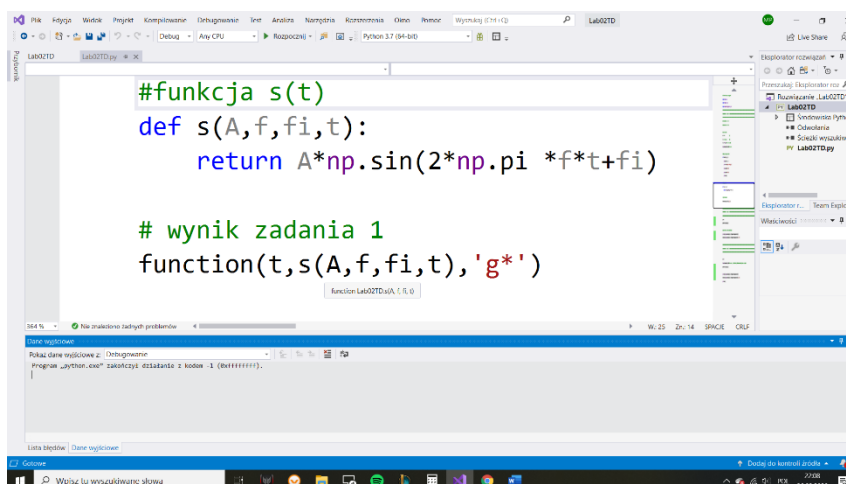


```
plt.ylabel('Os Y')
plt.show()

#funkcja s(t)
def s(A,f,fi,t):
    return A*np.sin(2*np.pi *f*t+fi)
```

This screenshot shows the implementation of the 's(t)' function. It defines a sine wave function with parameters A, f, fi, and t. The plot from the previous function is still visible in the background.

Wynik zadania.



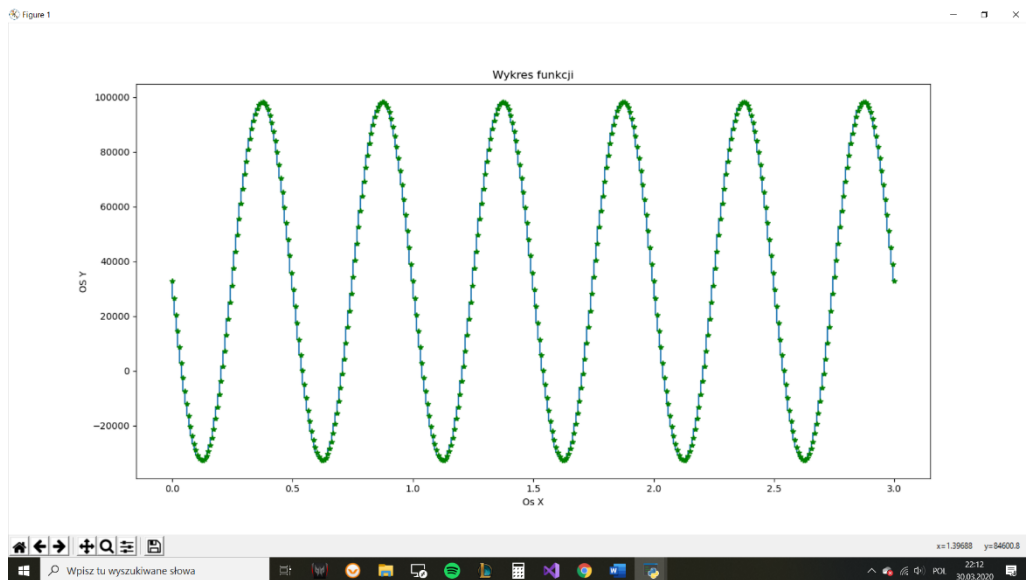
```
#funkcja s(t)
def s(A,f,fi,t):
    return A*np.sin(2*np.pi *f*t+fi)

# wynik zadania 1
function(t,s(A,f,fi,t),'g*')
```

The final screenshot shows the complete code for the assignment. It includes the 's(t)' function and a call to the 'function' to plot the result, labeled 'g*'. The terminal at the bottom shows the execution output.

Zadanie 2/.

Wykres skwantyzowanego sygnału dla $q=16$

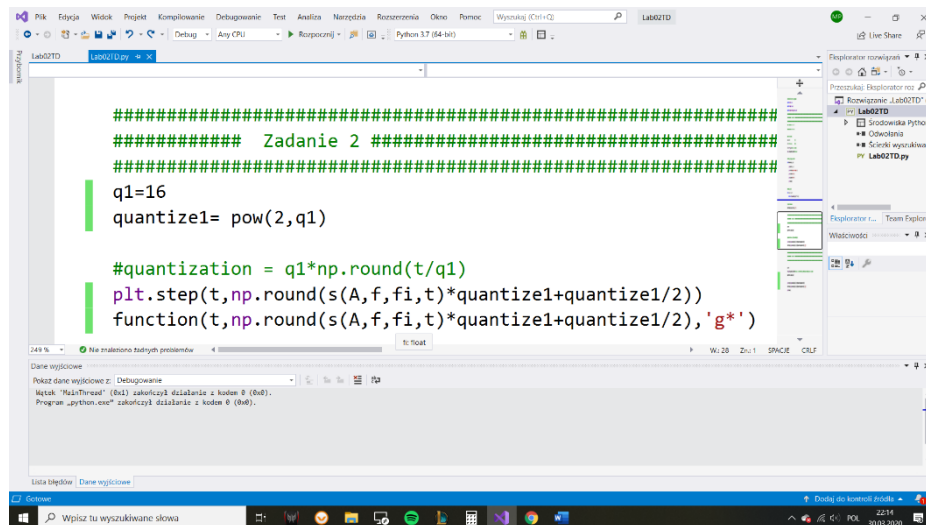


Treść kodu:

Podanie danych: (q) oraz spotęgowanie (2^q), t pobieram z zadania pierwszego.

```
#####  
##### Zadanie 2 #####  
#####  
q1=16  
quantize1= pow(2,q1)
```

Kwantyzacja sygnału z zadania pierwszego, funkcja `step` została użyta w celu pokazania „ścieżki” funkcji z zadania pierwszego.

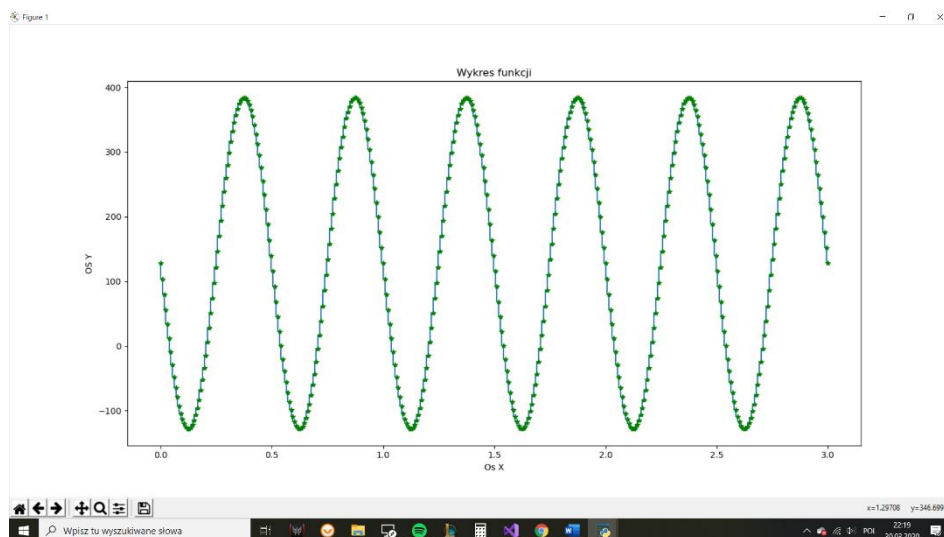


```
#####  
##### Zadanie 2 #####  
#####  
q1=16  
quantize1= pow(2,q1)  
  
#quantization = q1*np.round(t/q1)  
plt.step(t,np.round(s(A,f,fi,t)*quantize1+quantize1/2))  
function(t,np.round(s(A,f,fi,t)*quantize1+quantize1/2),'g*')
```

The screenshot shows a Python IDE with a file named `Lab02TD.py`. The code defines a quantization step size `q1=16` and uses `plt.step` to plot a quantized signal. The output window shows the execution of the `mainThread` function, which calls `quantize` with `q=8` and `fs=1000`.

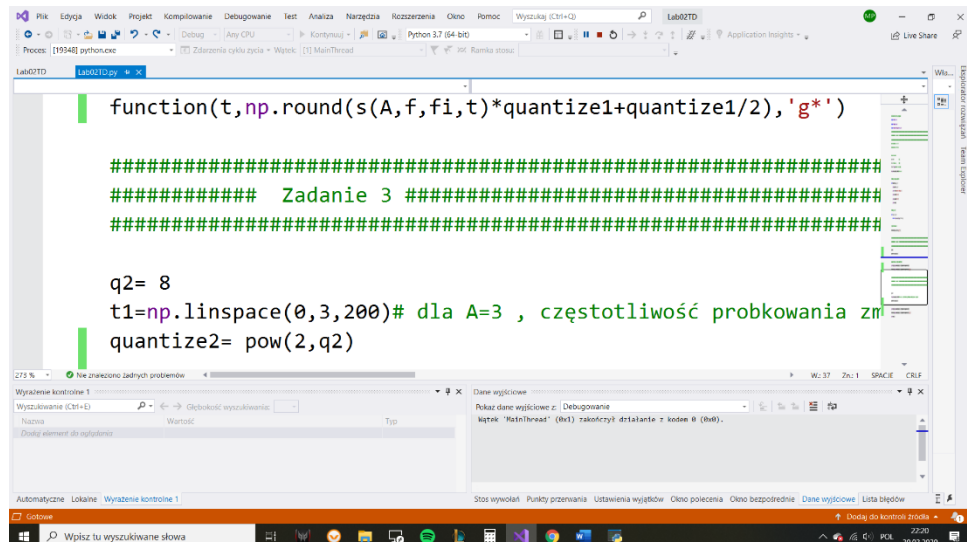
Zadanie 3/.

Wykres skwantyzowanego sygnału dla $q=8$ i połowy częstotliwość próbkowania



Treść kodu:

Podanie danych: nowe (q) oraz spotęgowanie (2^q), t podaje na nowo zmniejszając jego wartość o połowę względem poprzedniego zadania.

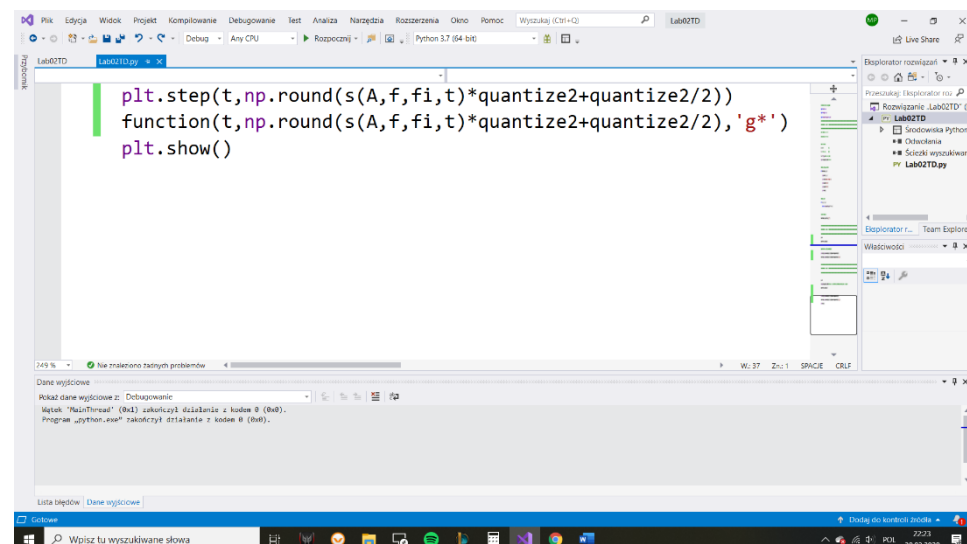


```
function(t,np.round(s(A,f,fi,t)*quantize1+quantize1/2),'g*')

#####
#####  Zadanie 3 #####
#####

q2= 8
t1=np.linspace(0,3,200)# dla A=3 , częstotliwość próbkowania zm
quantize2= pow(2,q2)
```

Kwantyzacja sygnału z zadania drugiego, funkcja `step` została użyta w celu pokazania „ścieżki” funkcji z zadania pierwszego.



```
plt.step(t,np.round(s(A,f,fi,t)*quantize2+quantize2/2))
function(t,np.round(s(A,f,fi,t)*quantize2+quantize2/2),'g*')
plt.show()
```