

Группа М32051

К работе допущен \_\_\_\_\_

Студент Пантелеев, Артамонов

Работа выполнена \_\_\_\_\_

Преподаватель Ефремова Е.А.

Отчет принят \_\_\_\_\_

## **Рабочий протокол и отчет по задаче от лектора №1**

## Задание

Сформировать импульс, близкий к прямоугольному, из спектра плоских гармонических волн с несущей длиной волны  $\lambda_0 = 1.5$  мкм и длительностью  $\Delta t \approx 10$  нс. Какая спектральная ширина пакета? Про моделировать его прохождение через среду с заданной дисперсией фазовой скорости  $v_{\text{фаз}} = \sqrt{c^2 + b^2 \lambda^2}$  для  $b = \{1, 10, 100\}$ . Определить характерное время расплывания пакета.

Закодировать сообщение (короткое слово) такими импульсами и про моделировать передачу такого сообщения до времен порядка характерного времени расплывания пакета.

## Решение

Для создания прямоугольного импульса с длительностью  $\Delta t = 10 \text{ нс}$  из плоских гармонических волн с несущей длиной волны  $\lambda_0 = 1.5 \text{ мкм}$ , воспользуемся преобразованием Фурье. Преобразование Фурье представляет сигнал в виде непрерывной суммы гармонических волн различных частот.

Спектральная ширина пакета связана с временной длительностью импульса через соотношение неопределенности Гейзенберга:

$$\Delta \lambda * \Delta t \geq 1 \Rightarrow \Delta \lambda \approx \frac{1}{\Delta t} \approx \frac{1}{10 * 10^{-9} \text{ с}} \approx 10^8 \text{ Гц}$$

$\Delta \lambda$  – спектральная ширина пакета

$\Delta t$  – длительность импульса

Теперь рассмотрим прохождение импульса через среду с заданной дисперсией фазовой скорости:

$$v_{\text{фаз}} = \sqrt{c^2 + b^2 \lambda^2}$$

$v_{\text{фаз}}$  – фазовая скорость

$\lambda$  – длина волны

$b$  – параметр дисперсии

Для каждого значения  $b$ , нам необходимо определить характерное время расплывания пакета. Оно связано с дисперсией групповой скорости и расстоянием, на котором происходит расплывание. Характерное время расплывания пакета можно определить, используя формулу для дисперсионного сдвига:

$$\Delta t_{\text{disp}} = \frac{D * L * \Delta \lambda}{\lambda_0}$$

$\Delta t_{\text{disp}}$  – характерное время расплывания волнового пакета

$D$  – дисперсионный коэффициент среды

$L$  – длина среды, через которую проходит импульс

$\Delta \lambda$  – спектральная ширина пакета

$\lambda_0$  – центральная длина волны импульса

Так как  $v_{\text{фаз}} = \sqrt{c^2 + b^2 \lambda^2}$ , то  $D = \frac{d(v_{\text{фаз}})}{d\lambda} = \frac{b^2 \lambda}{\sqrt{c^2 + b^2 \lambda^2}}$

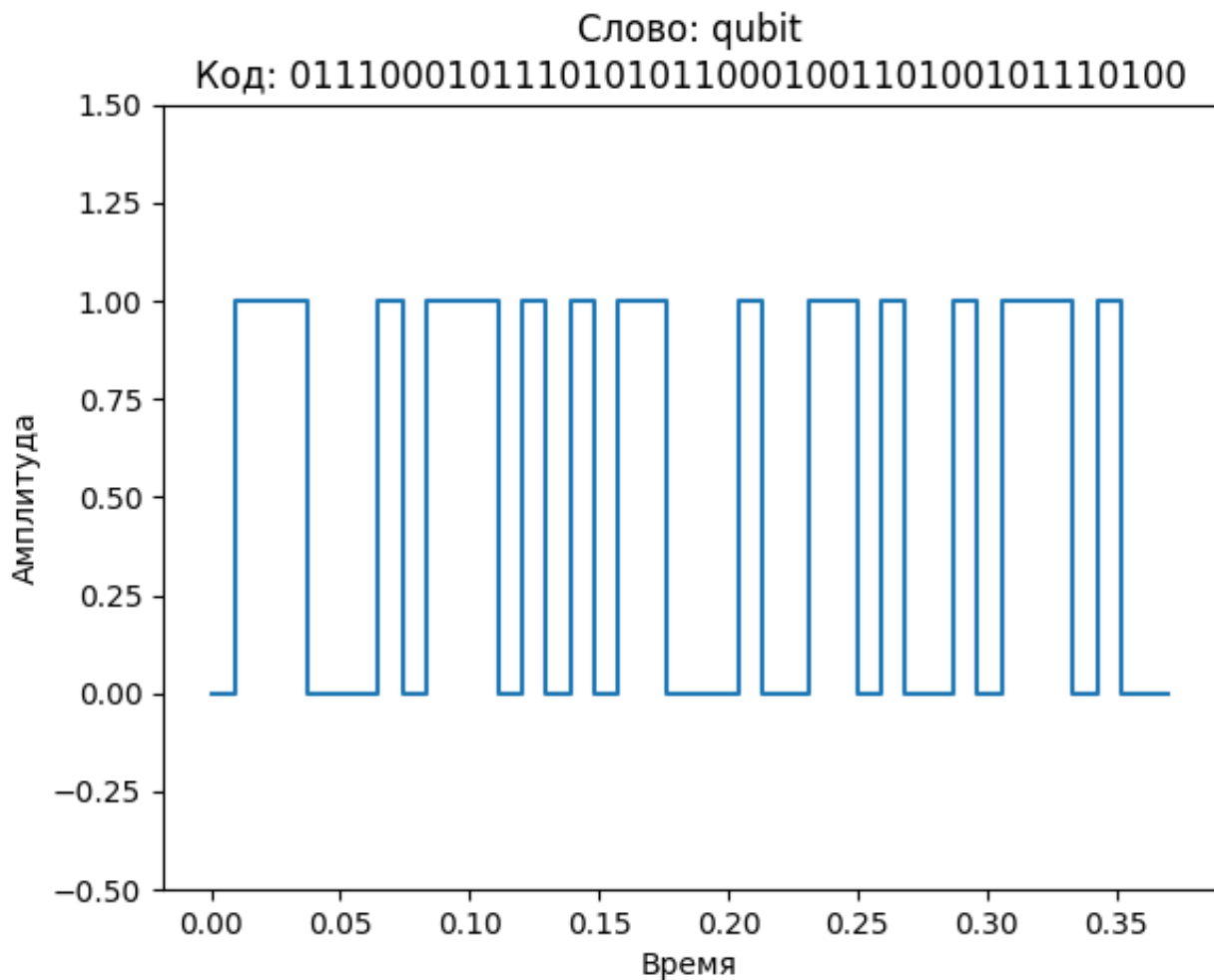
## Результат

```
/usr/bin/python3 /Users/smartoven/PycharmProjects/Python/Physics/LecTask1/main.py
Характерное время распыления пакета для b = 1: 3.7037037037037026e-07 с
Характерное время распыления пакета для b = 10: 3.703703703703703e-05 с
Характерное время распыления пакета для b = 100: 0.0037037037037037025 с
```

Для кодировки короткого сообщения с использованием импульсов будем использовать метод кодирования с использованием амплитуды импульсов.

Слово: qubit

Двоичный код: 01110001 01110101 01100010 01101001 01110100



## Исходный код

```
import matplotlib.pyplot as plt

def to_binary_str(s):
    return " ".join(f"{ord(i):08b}" for i in s)

def calculate_delta_t_dispersion(c, b, lambda_0, L,
delta_lambda):
    D = b ** 2 * lambda_0 / (c ** 2 + b ** 2 * lambda_0 ** 2) *
0.5
    delta_t_dispersion = (D * L * delta_lambda) / lambda_0 ** 2
    return delta_t_dispersion

c = 3e8
L = 1 # предполагаемая длина среды в метрах
lambda_0 = 1.5e-6
delta_t = 1e-5
delta_lambda = 1 / delta_t
for b in [1, 10, 100]:
    delta_t_dispersion = calculate_delta_t_dispersion(c, b,
lambda_0, L, delta_lambda)
    print(f"Характерное время расплывания пакета для b = {b}:
{delta_t_dispersion} c")

word = "qubit"
binary_code_str = to_binary_str(word)
print(f"Слово: {word}")
print(f"Двоичный код: {binary_code_str}")

binary_code = binary_code_str.replace(" ", "")

time_points = []
current_time = 0

L = 1
b = 1000
delta_t_dispersion = calculate_delta_t_dispersion(c, b,
lambda_0, L, delta_lambda)
bit_time = delta_t_dispersion / len(binary_code) # время для
прохождения одного бита
for bit in binary_code:
    time_points.append(current_time)
    time_points.append(current_time + bit_time)
    current_time += bit_time

values = []
```

```
for bit in binary_code:
    if bit == "0":
        values.extend([0, 0])
    elif bit == "1":
        values.extend([1, 1])

plt.plot(time_points, values)
plt.title(f"Слово: {word}\nКод: {binary_code}")
plt.xlabel("Время")
plt.ylabel("Амплитуда")
plt.ylim(-0.5, 1.5)
plt.show()
```

Исходный код опубликован на GitHub и может быть скачан и запущен:  
<https://github.com/SmartOven/Python/blob/master/Physics/LecTask1/main.py>