

# My Project

Generated by Doxygen 1.9.8



<b>1 Telecommunication software engineering - Digital transmitter design and signal modulation</b>	<b>1</b>
1.1 Project Goal	1
1.2 Project Description	1
1.2.1 Main Activities	1
1.3 Repository Structure	2
1.4 Modulation techniques	2
1.4.1 Amplitude Modulation (AM)	2
1.4.2 Amplitude Shift Keying (ASK)	2
1.4.3 Frequency Modulation (FM)	2
1.4.4 Frequency Shift Keying (FSK)	3
1.4.5 Phase Shift Keying (PSK)	3
1.4.6 Binary Phase Shift Keying (BPSK)	3
1.4.7 Quadrature Amplitude Modulation (QAM)	3
1.4.8 Quadrature Phase Shift Keying (QPSK)	3
1.5 Pytest	3
1.6 Technologies Used	3
<b>2 Hierarchical Index</b>	<b>5</b>
2.1 Class Hierarchy	5
<b>3 Class Index</b>	<b>7</b>
3.1 Class List	7
<b>4 Class Documentation</b>	<b>9</b>
4.1 BPSK_modulation_test.TestBPSKModulation Class Reference	9
4.1.1 Member Function Documentation	9
4.1.1.1 test_basic_case()	9
4.1.1.2 test_bit_duration()	9
4.1.1.3 test_empty_binary_data()	10
4.1.1.4 test_high_frequency()	10
4.1.1.5 test_invalid_binary_data()	10
4.1.1.6 test_large_binary_data()	10
4.1.1.7 test_negative_bit_duration()	10
4.1.1.8 test_negative_sample_rate()	10
4.1.1.9 test_single_bit()	11
4.1.1.10 test_zero_bit_duration()	11
4.2 FM_modulation_test.TestFMModulation Class Reference	11
4.2.1 Member Function Documentation	11
4.2.1.1 test_different_message_frequencies()	11
4.2.1.2 test_frequency_deviation_change()	12
4.2.1.3 test_high_sample_rate()	12
4.2.1.4 test_increasing_carrier_frequency()	12
4.2.1.5 test_invalid_input()	12

4.2.1.6 test_large_duration()	12
4.2.1.7 test_low_sample_rate()	12
4.2.1.8 test_negative_frequency_deviation()	13
4.2.1.9 test_short_signal_duration()	13
4.2.1.10 test_signal_amplitude_variation()	13
4.3 FSK_modulation_test.TestFSKModulation Class Reference	13
4.3.1 Member Function Documentation	14
4.3.1.1 test_basic_case()	14
4.3.1.2 test_bit_duration()	14
4.3.1.3 test_empty_binary_data()	14
4.3.1.4 test_high_frequencies()	14
4.3.1.5 test_invalid_binary_data()	14
4.3.1.6 test_large_binary_data()	14
4.3.1.7 test_negative_sample_rate()	15
4.3.1.8 test_single_bit()	15
4.3.1.9 test_zero_bit_duration()	15
4.3.1.10 test_zero_carrier_frequencies()	15
4.4 PSK_modulation_test.TestPSKModulation Class Reference	15
4.4.1 Member Function Documentation	16
4.4.1.1 test_basic_case()	16
4.4.1.2 test_bit_duration()	16
4.4.1.3 test_empty_binary_data()	16
4.4.1.4 test_high_frequency()	16
4.4.1.5 test_invalid_binary_data()	16
4.4.1.6 test_large_binary_data()	17
4.4.1.7 test_negative_bit_duration()	17
4.4.1.8 test_negative_sample_rate()	17
4.4.1.9 test_single_bit()	17
4.4.1.10 test_zero_bit_duration()	17
4.5 QAM_modulation_test.TestQAMModulation Class Reference	17
4.5.1 Member Function Documentation	18
4.5.1.1 test_basic_case()	18
4.5.1.2 test_bit_duration()	18
4.5.1.3 test_empty_binary_data()	18
4.5.1.4 test_high_frequency()	18
4.5.1.5 test_invalid_binary_data_length()	18
4.5.1.6 test_invalid_constellation_points()	19
4.5.1.7 test_invalid_constellation_points_number()	19
4.5.1.8 test_large_binary_data()	19
4.5.1.9 test_negative_sample_rate()	19
4.5.1.10 test_single_symbol()	19
4.6 QPSK_modulation_test.TestQPSKModulation Class Reference	19

---

4.6.1 Member Function Documentation . . . . .	20
4.6.1.1 test_basic_case() . . . . .	20
4.6.1.2 test_bit_duration() . . . . .	20
4.6.1.3 test_empty_binary_data() . . . . .	20
4.6.1.4 test_high_frequency() . . . . .	20
4.6.1.5 test_invalid_binary_data_length() . . . . .	20
4.6.1.6 test_large_binary_data() . . . . .	21
4.6.1.7 test_negative_sample_rate() . . . . .	21
4.6.1.8 test_single_symbol() . . . . .	21
4.6.1.9 test_uneven_sample_rate_bit_duration() . . . . .	21
4.6.1.10 test_zero_carrier_frequency() . . . . .	21
<b>Index</b>	<b>23</b>



# Chapter 1

## Telecommunication software engineering - Digital transmitter design and signal modulation

### 1.1 Project Goal

*The goal of this project is to design a basic digital transmitter, focusing on:*

- Signal generation,
  - Implementation of various modulation techniques,
  - Analysis of a communication channel with noise,
  - Evaluation of the efficiency and robustness of different modulation schemes.
- 

### 1.2 Project Description

This project explores the concept of a digital transmitter, how it operates, and how signals are generated and prepared for transmission through a communication channel. It delves into the practical aspects of signal modulation and the challenges of transmitting signals through noisy channels.

#### 1.2.1 Main Activities

1. *Designing a digital transmitter:*

- Developing a foundational digital transmitter using Python, focusing on modular design and clear signal processing steps.

2. *Implementation of modulation schemes:*

- Amplitude Modulation (AM),
  - Amplitude Shift Keying (ASK),
  - Frequency Modulation (FM),
  - Frequency Shift Keying (FSK),
-

- Phase Shift Keying (PSK),
- Binary Phase Shift Keying (BPSK),
- Quadrature Amplitude Modulation (QAM),
- Quadrature Phase Shift Keying (QPSK).

### 3. *Simulation of a communication channel with noise:*

- Implementing a channel with Additive White Gaussian Noise (AWGN).
- Analyzing the Signal-to-Noise Ratio (SNR).

### 4. *Evaluation of efficiency and robustness:*

- Assessing the energy efficiency of each modulation scheme.
- Analyzing noise resistance.

## 1.3 Repository Structure

Directory name	Description
AM-ASK+tests	This directory contains all code files necessary for AM and ASK modulations, as well as their respective test files.
FM-FSK+tests	This directory contains all code files necessary for FM and FSK modulations, as well as their respective test files.
PSK-BPSK+tests	This directory contains all code files necessary for PSK and BPSK modulations, as well as their respective test files.
QAM-QPSK+tests	This directory contains all code files necessary for QAM and QPSK modulations, as well as their respective test files.
README.md	Project details.

## 1.4 Modulation techniques

### 1.4.1 Amplitude Modulation (AM)

**Amplitude Modulation (AM)** is a modulation technique where the amplitude of the carrier signal is varied in proportion to the instantaneous amplitude of the message signal. It's a simple method but is susceptible to noise.

#### **Amplitude Modulation (AM) image**

Figure 1: Amplitude modulation

### 1.4.2 Amplitude Shift Keying (ASK)

**Amplitude Shift Keying (ASK)** is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In its simplest form, the presence of a carrier wave represents a binary 1, and the absence represents a binary 0.

#### **Amplitude Shift Keying (ASK) image**

Figure 2: Amplitude shift keying

### 1.4.3 Frequency Modulation (FM)

**Frequency Modulation (FM)** is a modulation technique where the frequency of the carrier signal is varied in proportion to the instantaneous amplitude of the message signal. FM is less susceptible to noise than AM.

#### **Frequency Modulation (FM) image**



Figure 3: Frequency modulation

#### 1.4.4 Frequency Shift Keying (FSK)

**Frequency Shift Keying (FSK)** is a form of frequency modulation that represents digital data as variations in the frequency of a carrier wave. Different frequencies are used to represent different binary values.

**Frequency Shift Keying (FSK) image**

Figure 4: Frequency shift keying

#### 1.4.5 Phase Shift Keying (PSK)

**Phase Shift Keying (PSK)** is a modulation technique where the phase of the carrier signal is varied to represent digital data. The amplitude and frequency of the carrier signal remain constant.

**Phase Shift Keying (PSK) image**

Figure 5: Phase shift keying

#### 1.4.6 Binary Phase Shift Keying (BPSK)

**Binary Phase Shift Keying (BPSK)** is a form of phase shift keying where the phase of the carrier signal is varied to represent binary data. It uses two phases to represent binary 0 and 1.

**Binary Phase Shift Keying (BPSK) image**

Figure 6: Binary phase shift keying

#### 1.4.7 Quadrature Amplitude Modulation (QAM)

**Quadrature Amplitude Modulation (QAM)** is a modulation technique that combines both amplitude and phase modulation to transmit more data per symbol. It uses multiple amplitude levels and phase shifts to encode data.

**Quadrature Amplitude Modulation (QAM) image**

Figure 6: Quadrature amplitude modulation

#### 1.4.8 Quadrature Phase Shift Keying (QPSK)

**Quadrature Phase Shift Keying (QPSK)** is a form of phase shift keying where the phase of the carrier signal is varied to represent digital data. It uses four phases to represent two bits of data.

**Quadrature Phase Shift Keying (QPSK) image**

Figure 7: Quadrature phase shift keying

---

## 1.5 Pytest

\*The pytest framework is used to run the tests for each modulation technique. The tests can be run with: `pytest -v --cov 'file.py'`.*`

*Figure 8 shows how pytest output looks when running the tests for all the modulation techniques:*

Figure 8: Code coverage

---

## 1.6 Technologies Used

- **Programming Language:** Python
- **Libraries:**

- `numpy` - Numerical data processing.
- `matplotlib` - Visualization of simulation results.
- `scipy` - Noise generation and SNR analysis. .
- `pytest` - Testing framework.
- `coverage` - Code coverage analysis.

## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

unittest.TestCase	
BPSK_modulation_test.TestBPSKModulation . . . . .	9
FM_modulation_test.TestFMModulation . . . . .	11
FSK_modulation_test.TestFSKModulation . . . . .	13
PSK_modulation_test.TestPSKModulation . . . . .	15
QAM_modulation_test.TestQAMModulation . . . . .	17
QPSK_modulation_test.TestQPSKModulation . . . . .	19



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">BPSK_modulation_test.TestBPSKModulation</a>	9
<a href="#">FM_modulation_test.TestFMModulation</a>	11
<a href="#">FSK_modulation_test.TestFSKModulation</a>	13
<a href="#">PSK_modulation_test.TestPSKModulation</a>	15
<a href="#">QAM_modulation_test.TestQAMModulation</a>	17
<a href="#">QPSK_modulation_test.TestQPSKModulation</a>	19

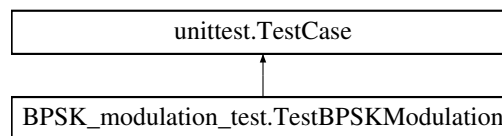


# Chapter 4

## Class Documentation

### 4.1 BPSK\_modulation\_test.TestBPSKModulation Class Reference

Inheritance diagram for BPSK\_modulation\_test.TestBPSKModulation:



#### Public Member Functions

- [test\\_basic\\_case](#) (self)
- [test\\_bit\\_duration](#) (self)
- [test\\_high\\_frequency](#) (self)
- [test\\_large\\_binary\\_data](#) (self)
- [test\\_single\\_bit](#) (self)
- [test\\_invalid\\_binary\\_data](#) (self)
- [test\\_zero\\_bit\\_duration](#) (self)
- [test\\_negative\\_sample\\_rate](#) (self)
- [test\\_negative\\_bit\\_duration](#) (self)
- [test\\_empty\\_binary\\_data](#) (self)

#### 4.1.1 Member Function Documentation

##### 4.1.1.1 test\_basic\_case()

```
BPSK_modulation_test.TestBPSKModulation.test_basic_case (
    self )
```

Test the basic functionality of the BPSK modulation function.

Verifies:

- The lengths of all generated signals are consistent.
- The modulating signal corresponds correctly to the binary data input.

##### 4.1.1.2 test\_bit\_duration()

```
BPSK_modulation_test.TestBPSKModulation.test_bit_duration (
    self )
```

Test the function with varying bit durations.

Verifies:

- The total duration of the generated signal matches the expected value.

#### 4.1.1.3 test\_empty\_binary\_data()

```
BPSK_modulation_test.TestBPSKModulation.test_empty_binary_data (
    self )
```

Test the function with empty binary data.

Verifies:

- The function returns empty arrays when the binary data is empty.

#### 4.1.1.4 test\_high\_frequency()

```
BPSK_modulation_test.TestBPSKModulation.test_high_frequency (
    self )
```

Test the function with a high carrier frequency.

Verifies:

- The carrier signal is a normalized sine wave with amplitude in the range [-1, 1].

#### 4.1.1.5 test\_invalid\_binary\_data()

```
BPSK_modulation_test.TestBPSKModulation.test_invalid_binary_data (
    self )
```

Test the function with invalid binary input data.

Verifies:

- The function raises a ValueError for non-binary input data.

#### 4.1.1.6 test\_large\_binary\_data()

```
BPSK_modulation_test.TestBPSKModulation.test_large_binary_data (
    self )
```

Test the BPSK modulation function with a large binary data input.

Verifies:

- The function scales appropriately without errors.

#### 4.1.1.7 test\_negative\_bit\_duration()

```
BPSK_modulation_test.TestBPSKModulation.test_negative_bit_duration (
    self )
```

Test the function with a negative bit duration.

Verifies:

- The function raises a ValueError for a negative bit duration.

#### 4.1.1.8 test\_negative\_sample\_rate()

```
BPSK_modulation_test.TestBPSKModulation.test_negative_sample_rate (
    self )
```

Test the function with a negative sample rate.

Verifies:

- The function raises a ValueError for a negative sample rate.



#### 4.1.1.9 test\_single\_bit()

```
BPSK_modulation_test.TestBPSKModulation.test_single_bit (
    self )
```

Test the BPSK modulation function with a single-bit binary data input.

Verifies:

- The generated signals correspond to the single bit.

#### 4.1.1.10 test\_zero\_bit\_duration()

```
BPSK_modulation_test.TestBPSKModulation.test_zero_bit_duration (
    self )
```

Test the BPSK modulation function with zero bit duration.

Verifies:

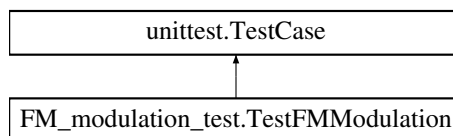
- The function returns empty arrays.

The documentation for this class was generated from the following file:

- PSK-BPSK+tests/BPSK\_modulation\_test.py

## 4.2 FM\_modulation\_test.TestFMModulation Class Reference

Inheritance diagram for FM\_modulation\_test.TestFMModulation:



### Public Member Functions

- [test\\_increasing\\_carrier\\_frequency](#) (self)
- [test\\_frequency\\_deviation\\_change](#) (self)
- [test\\_short\\_signal\\_duration](#) (self)
- [test\\_low\\_sample\\_rate](#) (self)
- [test\\_high\\_sample\\_rate](#) (self)
- [test\\_signal\\_amplitude\\_variation](#) (self)
- [test\\_large\\_duration](#) (self)
- [test\\_different\\_message\\_frequencies](#) (self)
- [test\\_invalid\\_input](#) (self)
- [test\\_negative\\_frequency\\_deviation](#) (self)

### 4.2.1 Member Function Documentation

#### 4.2.1.1 test\_different\_message\_frequencies()

```
FM_modulation_test.TestFMModulation.test_different_message_frequencies (
    self )
```

Test FM modulation with different frequencies in the message signal.

Verifies:

- The FM signal reflects the changing message signal frequency.

#### 4.2.1.2 test\_frequency\_deviation\_change()

```
FM_modulation_test.TestFMModulation.test_frequency_deviation_change (
    self )
```

Test FM modulation with varying frequency deviation.

Verifies:

- The maximum instantaneous frequency deviation matches or exceeds the set value.

#### 4.2.1.3 test\_high\_sample\_rate()

```
FM_modulation_test.TestFMModulation.test_high_sample_rate (
    self )
```

Test FM modulation with a very high sampling rate.

Verifies:

- The function handles high-resolution signals correctly.

#### 4.2.1.4 test\_increasing\_carrier\_frequency()

```
FM_modulation_test.TestFMModulation.test_increasing_carrier_frequency (
    self )
```

Test FM modulation with a standard carrier frequency and parameters.

Verifies:

- Signal lengths are consistent across all generated components.

#### 4.2.1.5 test\_invalid\_input()

```
FM_modulation_test.TestFMModulation.test_invalid_input (
    self )
```

Test FM modulation with invalid inputs.

Verifies:

- The function raises a ValueError for invalid parameter combinations.

#### 4.2.1.6 test\_large\_duration()

```
FM_modulation_test.TestFMModulation.test_large_duration (
    self )
```

Test FM modulation with a very large signal duration.

Verifies:

- The function handles long signals without errors.

#### 4.2.1.7 test\_low\_sample\_rate()

```
FM_modulation_test.TestFMModulation.test_low_sample_rate (
    self )
```

Test FM modulation with a lower sampling rate.

Verifies:

- Signal lengths are consistent despite the lower resolution.

#### 4.2.1.8 test\_negative\_frequency\_deviation()

```
FM_modulation_test.TestFMModulation.test_negative_frequency_deviation (
    self )
```

Test FM modulation with a negative frequency deviation.

Verifies:

- The function raises a ValueError.

#### 4.2.1.9 test\_short\_signal\_duration()

```
FM_modulation_test.TestFMModulation.test_short_signal_duration (
    self )
```

Test FM modulation with a shorter signal duration.

Verifies:

- The total duration of the generated signal is as expected.

#### 4.2.1.10 test\_signal\_amplitude\_variation()

```
FM_modulation_test.TestFMModulation.test_signal_amplitude_variation (
    self )
```

Test FM modulation when the message signal amplitude changes.

Verifies:

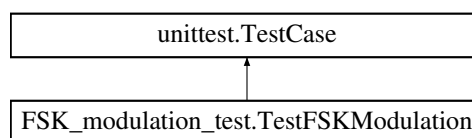
- The FM signal adapts correctly to varying amplitudes.

The documentation for this class was generated from the following file:

- FM-FSK+tests/FM\_modulation\_test.py

## 4.3 FSK\_modulation\_test.TestFSKModulation Class Reference

Inheritance diagram for FSK\_modulation\_test.TestFSKModulation:



### Public Member Functions

- [test\\_basic\\_case](#) (self)
- [test\\_bit\\_duration](#) (self)
- [test\\_high\\_frequencies](#) (self)
- [test\\_large\\_binary\\_data](#) (self)
- [test\\_single\\_bit](#) (self)
- [test\\_invalid\\_binary\\_data](#) (self)
- [test\\_zero\\_bit\\_duration](#) (self)
- [test\\_negative\\_sample\\_rate](#) (self)
- [test\\_zero\\_carrier\\_frequencies](#) (self)
- [test\\_empty\\_binary\\_data](#) (self)

### 4.3.1 Member Function Documentation

#### 4.3.1.1 test\_basic\_case()

```
FSK_modulation_test.TestFSKModulation.test_basic_case (  
    self )
```

Test the basic functionality of the FSK modulation function.

Verifies:

- The lengths of all generated signals are consistent.
- The modulating signal corresponds correctly to the binary data input.

#### 4.3.1.2 test\_bit\_duration()

```
FSK_modulation_test.TestFSKModulation.test_bit_duration (  
    self )
```

Test the function with varying bit durations.

Verifies:

- The total duration of the generated signal matches the expected value.

#### 4.3.1.3 test\_empty\_binary\_data()

```
FSK_modulation_test.TestFSKModulation.test_empty_binary_data (  
    self )
```

Test the function with empty binary data.

Verifies:

- The function returns empty arrays when the binary data is empty.

#### 4.3.1.4 test\_high\_frequencies()

```
FSK_modulation_test.TestFSKModulation.test_high_frequencies (  
    self )
```

Test the function with high carrier frequencies.

Verifies:

- The carrier signals are normalized sine waves with amplitudes in the range [-1, 1].

#### 4.3.1.5 test\_invalid\_binary\_data()

```
FSK_modulation_test.TestFSKModulation.test_invalid_binary_data (  
    self )
```

Test the function with invalid binary input data.

Verifies:

- The function raises a ValueError for non-binary input data.

#### 4.3.1.6 test\_large\_binary\_data()

```
FSK_modulation_test.TestFSKModulation.test_large_binary_data (  
    self )
```

Test the FSK modulation function with a large binary data input.

Verifies:

- The function scales appropriately without errors.

#### 4.3.1.7 test\_negative\_sample\_rate()

```
FSK_modulation_test.TestFSKModulation.test_negative_sample_rate (
    self )
```

Test the function with a negative sample rate.

Verifies:

- The function raises a ValueError for a negative sample rate.

#### 4.3.1.8 test\_single\_bit()

```
FSK_modulation_test.TestFSKModulation.test_single_bit (
    self )
```

Test the FSK modulation function with a single-bit binary data input.

Verifies:

- The generated signals correspond to the single bit.

#### 4.3.1.9 test\_zero\_bit\_duration()

```
FSK_modulation_test.TestFSKModulation.test_zero_bit_duration (
    self )
```

Test the FSK modulation function with zero bit duration.

Verifies:

- The function returns empty arrays.

#### 4.3.1.10 test\_zero\_carrier\_frequencies()

```
FSK_modulation_test.TestFSKModulation.test_zero_carrier_frequencies (
    self )
```

Test the function with zero carrier frequencies.

Verifies:

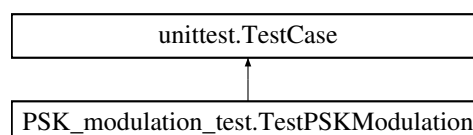
- The function does not raise an error, but the output signals are all zeros.

The documentation for this class was generated from the following file:

- FM-FSK+tests/FSK\_modulation\_test.py

## 4.4 PSK\_modulation\_test.TestPSKModulation Class Reference

Inheritance diagram for PSK\_modulation\_test.TestPSKModulation:



### Public Member Functions

- [test\\_basic\\_case](#) (self)
- [test\\_bit\\_duration](#) (self)
- [test\\_high\\_frequency](#) (self)
- [test\\_large\\_binary\\_data](#) (self)

- [test\\_single\\_bit](#) (self)
- [test\\_invalid\\_binary\\_data](#) (self)
- [test\\_zero\\_bit\\_duration](#) (self)
- [test\\_negative\\_sample\\_rate](#) (self)
- [test\\_negative\\_bit\\_duration](#) (self)
- [test\\_empty\\_binary\\_data](#) (self)

## 4.4.1 Member Function Documentation

### 4.4.1.1 test\_basic\_case()

```
PSK_modulation_test.TestPSKModulation.test_basic_case (  
    self )
```

Test the basic functionality of the PSK modulation function.

Verifies:

- The lengths of all generated signals are consistent.
- The modulating signal corresponds correctly to the binary data input.

### 4.4.1.2 test\_bit\_duration()

```
PSK_modulation_test.TestPSKModulation.test_bit_duration (  
    self )
```

Test the function with varying bit durations.

Verifies:

- The total duration of the generated signal matches the expected value.

### 4.4.1.3 test\_empty\_binary\_data()

```
PSK_modulation_test.TestPSKModulation.test_empty_binary_data (  
    self )
```

Test the function with empty binary data.

Verifies:

- The function returns empty arrays when the binary data is empty.

### 4.4.1.4 test\_high\_frequency()

```
PSK_modulation_test.TestPSKModulation.test_high_frequency (  
    self )
```

Test the function with a high carrier frequency.

Verifies:

- The carrier signal is a normalized sine wave with amplitude in the range [-1, 1].

### 4.4.1.5 test\_invalid\_binary\_data()

```
PSK_modulation_test.TestPSKModulation.test_invalid_binary_data (  
    self )
```

Test the function with invalid binary input data.

Verifies:

- The function raises a ValueError for non-binary input data.

**4.4.1.6 test\_large\_binary\_data()**

```
PSK_modulation_test.TestPSKModulation.test_large_binary_data (
    self )
```

Test the PSK modulation function with a large binary data input.

Verifies:

- The function scales appropriately without errors.

**4.4.1.7 test\_negative\_bit\_duration()**

```
PSK_modulation_test.TestPSKModulation.test_negative_bit_duration (
    self )
```

Test the function with a negative bit duration.

Verifies:

- The function raises a ValueError for a negative bit duration.

**4.4.1.8 test\_negative\_sample\_rate()**

```
PSK_modulation_test.TestPSKModulation.test_negative_sample_rate (
    self )
```

Test the function with a negative sample rate.

Verifies:

- The function raises a ValueError for a negative sample rate.

**4.4.1.9 test\_single\_bit()**

```
PSK_modulation_test.TestPSKModulation.test_single_bit (
    self )
```

Test the PSK modulation function with a single-bit binary data input.

Verifies:

- The generated signals correspond to the single bit.

**4.4.1.10 test\_zero\_bit\_duration()**

```
PSK_modulation_test.TestPSKModulation.test_zero_bit_duration (
    self )
```

Test the PSK modulation function with zero bit duration.

Verifies:

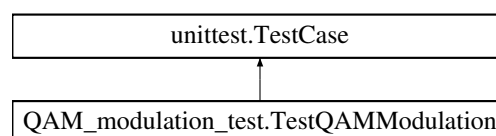
- The function returns empty arrays.

The documentation for this class was generated from the following file:

- PSK-BPSK+tests/PSK\_modulation\_test.py

**4.5 QAM\_modulation\_test.TestQAMModulation Class Reference**

Inheritance diagram for QAM\_modulation\_test.TestQAMModulation:



## Public Member Functions

- [test\\_basic\\_case](#) (self)
- [test\\_bit\\_duration](#) (self)
- [test\\_high\\_frequency](#) (self)
- [test\\_large\\_binary\\_data](#) (self)
- [test\\_single\\_symbol](#) (self)
- [test\\_invalid\\_binary\\_data\\_length](#) (self)
- [test\\_invalid\\_constellation\\_points](#) (self)
- [test\\_invalid\\_constellation\\_points\\_number](#) (self)
- [test\\_negative\\_sample\\_rate](#) (self)
- [test\\_empty\\_binary\\_data](#) (self)

## 4.5.1 Member Function Documentation

### 4.5.1.1 test\_basic\_case()

```
QAM_modulation_test.TestQAMModulation.test_basic_case (  
    self )
```

Test the basic functionality of the QAM modulation function.

Verifies:

- The lengths of all generated signals are consistent.
- The modulating signal has the correct length.

### 4.5.1.2 test\_bit\_duration()

```
QAM_modulation_test.TestQAMModulation.test_bit_duration (  
    self )
```

Test the function with varying bit durations.

Verifies:

- The total duration of the generated signal matches the expected value.

### 4.5.1.3 test\_empty\_binary\_data()

```
QAM_modulation_test.TestQAMModulation.test_empty_binary_data (  
    self )
```

Test the function with empty binary data.

Verifies:

- The function returns empty arrays when the binary data is empty.

### 4.5.1.4 test\_high\_frequency()

```
QAM_modulation_test.TestQAMModulation.test_high_frequency (  
    self )
```

Test the function with a high carrier frequency.

Verifies:

- The carrier signals are normalized sine and cosine waves with amplitude in the range [-1, 1].

### 4.5.1.5 test\_invalid\_binary\_data\_length()

```
QAM_modulation_test.TestQAMModulation.test_invalid_binary_data_length (  
    self )
```

Test the function with invalid binary input data length.

Verifies:

- The function raises a ValueError for binary data length not a multiple of bits per symbol.



**4.5.1.6 test\_invalid\_constellation\_points()**

```
QAM_modulation_test.TestQAMModulation.test_invalid_constellation_points (
    self )
```

Test the function with invalid constellation points.

Verifies:

- The function raises a ValueError for non-complex constellation points.

**4.5.1.7 test\_invalid\_constellation\_points\_number()**

```
QAM_modulation_test.TestQAMModulation.test_invalid_constellation_points_number (
    self )
```

Test the function with invalid number of constellation points.

Verifies:

- The function raises a ValueError for number of constellation points not a power of 2.

**4.5.1.8 test\_large\_binary\_data()**

```
QAM_modulation_test.TestQAMModulation.test_large_binary_data (
    self )
```

Test the QAM modulation function with a large binary data input.

Verifies:

- The function scales appropriately without errors.

**4.5.1.9 test\_negative\_sample\_rate()**

```
QAM_modulation_test.TestQAMModulation.test_negative_sample_rate (
    self )
```

Test the QAM modulation function with a negative sample rate.

Verifies:

- The function raises a ValueError when sample\_rate is negative.

**4.5.1.10 test\_single\_symbol()**

```
QAM_modulation_test.TestQAMModulation.test_single_symbol (
    self )
```

Test the QAM modulation function with a single symbol binary data input.

Verifies:

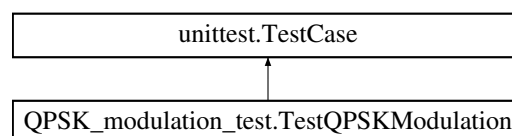
- The generated signals correspond to the single symbol.

The documentation for this class was generated from the following file:

- QAM-QPSK+tests/QAM\_modulation\_test.py

**4.6 QPSK\_modulation\_test.TestQPSKModulation Class Reference**

Inheritance diagram for QPSK\_modulation\_test.TestQPSKModulation:



## Public Member Functions

- [test\\_basic\\_case](#) (self)
- [test\\_bit\\_duration](#) (self)
- [test\\_high\\_frequency](#) (self)
- [test\\_large\\_binary\\_data](#) (self)
- [test\\_single\\_symbol](#) (self)
- [test\\_invalid\\_binary\\_data\\_length](#) (self)
- [test\\_negative\\_sample\\_rate](#) (self)
- [test\\_empty\\_binary\\_data](#) (self)
- [test\\_zero\\_carrier\\_frequency](#) (self)
- [test\\_uneven\\_sample\\_rate\\_bit\\_duration](#) (self)

## 4.6.1 Member Function Documentation

### 4.6.1.1 test\_basic\_case()

```
QPSK_modulation_test.TestQPSKModulation.test_basic_case (
    self )
```

Test the basic functionality of the QPSK modulation function.

Verifies:

- The lengths of all generated signals are consistent.
- The modulating signal has the correct length.

### 4.6.1.2 test\_bit\_duration()

```
QPSK_modulation_test.TestQPSKModulation.test_bit_duration (
    self )
```

Test the function with varying bit durations.

Verifies:

- The total duration of the generated signal matches the expected value.

### 4.6.1.3 test\_empty\_binary\_data()

```
QPSK_modulation_test.TestQPSKModulation.test_empty_binary_data (
    self )
```

Test the function with empty binary data.

Verifies:

- The function returns empty arrays when the binary data is empty.

### 4.6.1.4 test\_high\_frequency()

```
QPSK_modulation_test.TestQPSKModulation.test_high_frequency (
    self )
```

Test the function with a high carrier frequency.

Verifies:

- The carrier signals are normalized sine and cosine waves with amplitude in the range [-1, 1].

### 4.6.1.5 test\_invalid\_binary\_data\_length()

```
QPSK_modulation_test.TestQPSKModulation.test_invalid_binary_data_length (
    self )
```

Test the function with invalid binary input data length.

Verifies:

- The function raises a ValueError for binary data length not a multiple of 2.

#### 4.6.1.6 test\_large\_binary\_data()

```
QPSK_modulation_test.TestQPSKModulation.test_large_binary_data (
    self )
```

Test the QPSK modulation function with a large binary data input.

Verifies:

- The function scales appropriately without errors.

#### 4.6.1.7 test\_negative\_sample\_rate()

```
QPSK_modulation_test.TestQPSKModulation.test_negative_sample_rate (
    self )
```

Test the QPSK modulation function with a negative sample rate.

Verifies:

- The function raises a ValueError when sample\_rate is negative.

#### 4.6.1.8 test\_single\_symbol()

```
QPSK_modulation_test.TestQPSKModulation.test_single_symbol (
    self )
```

Test the QPSK modulation function with a single symbol binary data input.

Verifies:

- The generated signals correspond to the single symbol.

#### 4.6.1.9 test\_uneven\_sample\_rate\_bit\_duration()

```
QPSK_modulation_test.TestQPSKModulation.test_uneven_sample_rate_bit_duration (
    self )
```

Test the function with a sample rate that is not a multiple of the bit duration.

Verifies:

- The function still generates signals without errors.

#### 4.6.1.10 test\_zero\_carrier\_frequency()

```
QPSK_modulation_test.TestQPSKModulation.test_zero_carrier_frequency (
    self )
```

Test the function with a zero carrier frequency.

Verifies:

- The carrier signals are constant (all zeros).

The documentation for this class was generated from the following file:

- QAM-QPSK+tests/QPSK\_modulation\_test.py



# Index

BPSK\_modulation\_test.TestBPSKModulation, 9

- test\_basic\_case, 9
- test\_bit\_duration, 9
- test\_empty\_binary\_data, 9
- test\_high\_frequency, 10
- test\_invalid\_binary\_data, 10
- test\_large\_binary\_data, 10
- test\_negative\_bit\_duration, 10
- test\_negative\_sample\_rate, 10
- test\_single\_bit, 10
- test\_zero\_bit\_duration, 11

FM\_modulation\_test.TestFMModulation, 11

- test\_different\_message\_frequencies, 11
- test\_frequency\_deviation\_change, 11
- test\_high\_sample\_rate, 12
- test\_increasing\_carrier\_frequency, 12
- test\_invalid\_input, 12
- test\_large\_duration, 12
- test\_low\_sample\_rate, 12
- test\_negative\_frequency\_deviation, 12
- test\_short\_signal\_duration, 13
- test\_signal\_amplitude\_variation, 13

FSK\_modulation\_test.TestFSKModulation, 13

- test\_basic\_case, 14
- test\_bit\_duration, 14
- test\_empty\_binary\_data, 14
- test\_high\_frequencies, 14
- test\_invalid\_binary\_data, 14
- test\_large\_binary\_data, 14
- test\_negative\_sample\_rate, 14
- test\_single\_bit, 15
- test\_zero\_bit\_duration, 15
- test\_zero\_carrier\_frequencies, 15

PSK\_modulation\_test.TestPSKModulation, 15

- test\_basic\_case, 16
- test\_bit\_duration, 16
- test\_empty\_binary\_data, 16
- test\_high\_frequency, 16
- test\_invalid\_binary\_data, 16
- test\_large\_binary\_data, 16
- test\_negative\_bit\_duration, 17
- test\_negative\_sample\_rate, 17
- test\_single\_bit, 17
- test\_zero\_bit\_duration, 17

QAM\_modulation\_test.TestQAMModulation, 17

- test\_basic\_case, 18
- test\_bit\_duration, 18

test\_empty\_binary\_data, 18

- test\_high\_frequency, 18
- test\_invalid\_binary\_data\_length, 18
- test\_invalid\_constellation\_points, 18
- test\_invalid\_constellation\_points\_number, 19
- test\_large\_binary\_data, 19
- test\_negative\_sample\_rate, 19
- test\_single\_symbol, 19

QPSK\_modulation\_test.TestQPSKModulation, 19

- test\_basic\_case, 20
- test\_bit\_duration, 20
- test\_empty\_binary\_data, 20
- test\_high\_frequency, 20
- test\_invalid\_binary\_data\_length, 20
- test\_large\_binary\_data, 20
- test\_negative\_sample\_rate, 21
- test\_single\_symbol, 21
- test\_uneven\_sample\_rate\_bit\_duration, 21
- test\_zero\_carrier\_frequency, 21

Telecommunication software engineering - Digital transmitter design and signal modulation, 1

test\_basic\_case

- BPSK\_modulation\_test.TestBPSKModulation, 9
- FSK\_modulation\_test.TestFSKModulation, 14
- PSK\_modulation\_test.TestPSKModulation, 16
- QAM\_modulation\_test.TestQAMModulation, 18
- QPSK\_modulation\_test.TestQPSKModulation, 20

test\_bit\_duration

- BPSK\_modulation\_test.TestBPSKModulation, 9
- FSK\_modulation\_test.TestFSKModulation, 14
- PSK\_modulation\_test.TestPSKModulation, 16
- QAM\_modulation\_test.TestQAMModulation, 18
- QPSK\_modulation\_test.TestQPSKModulation, 20

test\_different\_message\_frequencies

- FM\_modulation\_test.TestFMModulation, 11

test\_empty\_binary\_data

- BPSK\_modulation\_test.TestBPSKModulation, 9
- FSK\_modulation\_test.TestFSKModulation, 14
- PSK\_modulation\_test.TestPSKModulation, 16
- QAM\_modulation\_test.TestQAMModulation, 18
- QPSK\_modulation\_test.TestQPSKModulation, 20

test\_frequency\_deviation\_change

- FM\_modulation\_test.TestFMModulation, 11

test\_high\_frequencies

- FSK\_modulation\_test.TestFSKModulation, 14

test\_high\_frequency

- BPSK\_modulation\_test.TestBPSKModulation, 10
- PSK\_modulation\_test.TestPSKModulation, 16
- QAM\_modulation\_test.TestQAMModulation, 18

QPSK\_modulation\_test.TestQPSKModulation, 20  
 test\_high\_sample\_rate  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_increasing\_carrier\_frequency  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_invalid\_binary\_data  
     BPSK\_modulation\_test.TestBPSKModulation, 10  
     FSK\_modulation\_test.TestFSKModulation, 14  
     PSK\_modulation\_test.TestPSKModulation, 16  
 test\_invalid\_binary\_data\_length  
     QAM\_modulation\_test.TestQAMModulation, 18  
     QPSK\_modulation\_test.TestQPSKModulation, 20  
 test\_invalid\_constellation\_points  
     QAM\_modulation\_test.TestQAMModulation, 18  
 test\_invalid\_constellation\_points\_number  
     QAM\_modulation\_test.TestQAMModulation, 19  
 test\_invalid\_input  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_large\_binary\_data  
     BPSK\_modulation\_test.TestBPSKModulation, 10  
     FSK\_modulation\_test.TestFSKModulation, 14  
     PSK\_modulation\_test.TestPSKModulation, 16  
     QAM\_modulation\_test.TestQAMModulation, 19  
     QPSK\_modulation\_test.TestQPSKModulation, 20  
 test\_large\_duration  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_low\_sample\_rate  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_negative\_bit\_duration  
     BPSK\_modulation\_test.TestBPSKModulation, 10  
     PSK\_modulation\_test.TestPSKModulation, 17  
 test\_negative\_frequency\_deviation  
     FM\_modulation\_test.TestFMModulation, 12  
 test\_negative\_sample\_rate  
     BPSK\_modulation\_test.TestBPSKModulation, 10  
     FSK\_modulation\_test.TestFSKModulation, 14  
     PSK\_modulation\_test.TestPSKModulation, 17  
     QAM\_modulation\_test.TestQAMModulation, 19  
     QPSK\_modulation\_test.TestQPSKModulation, 21  
 test\_short\_signal\_duration  
     FM\_modulation\_test.TestFMModulation, 13  
 test\_signal\_amplitude\_variation  
     FM\_modulation\_test.TestFMModulation, 13  
 test\_single\_bit  
     BPSK\_modulation\_test.TestBPSKModulation, 10  
     FSK\_modulation\_test.TestFSKModulation, 15  
     PSK\_modulation\_test.TestPSKModulation, 17  
 test\_single\_symbol  
     QAM\_modulation\_test.TestQAMModulation, 19  
     QPSK\_modulation\_test.TestQPSKModulation, 21  
 test\_uneven\_sample\_rate\_bit\_duration  
     QPSK\_modulation\_test.TestQPSKModulation, 21  
 test\_zero\_bit\_duration  
     BPSK\_modulation\_test.TestBPSKModulation, 11  
     FSK\_modulation\_test.TestFSKModulation, 15  
     PSK\_modulation\_test.TestPSKModulation, 17  
 test\_zero\_carrier\_frequencies  
     FSK\_modulation\_test.TestFSKModulation, 15  
     test\_zero\_carrier\_frequency  
         QPSK\_modulation\_test.TestQPSKModulation, 21