МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт искусственного интеллекта
Кафедра автоматических систем

Система менеджмента качества обучения
СМКО МИРЭА 7.3/04.3Дпр.5КУБ/О/220400.62-16

КУРСОВАЯ РАБОТА
по дисциплине «Сети и системы передачи информации»

ТЕМА: « Код NRZ »
(название темы)

Руководитель работы _____ Чернышев Н. Н.
(подпись) (Ф.И.О.)
отлично

Студент _____ Матяш А.А.
(подпись) (Ф.И.О.)

Группа ККСО-01-19 шифр 19К0002

МОСКВА 2022г.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА-Российский технологический университет»
РТУ МИРЭА

Институт искусственного интеллекта
Кафедра автоматических систем

УТВЕРЖДАЮ:
Заведующий кафедрой автоматических систем
Лютов А.Г.
(Ф.И.О.)
«11» 02 2022г.

ЗАДАНИЕ
на выполнение курсовой работы по дисциплине
«Сети и системы передачи информации»

Студент: Матяш Андрей Александрович
(Ф.И.О.)
Группа ККСО-01-19 , шифр 19К0002

Тема: « Код NRZ »
(название темы)

Перечень вопросов, подлежащих разработке:
- анализ принципов кодирования/декодирования;
- выбор и анализ алгоритмов работы кодера/декодера;
- разработка и описание аппаратной (программной) реализации кодера;
- разработка и описание аппаратной (программной) реализации декодера;
- разработка и описание примера кодирования/декодирования;
- оценка эффективности метода;
- анализ преимуществ и недостатков метода.

Срок представления к защите курсовой работы до «31» 05 2022г.

Задание на курсовую работу выдал _____ Чернышев Н. Н.
(подпись) (Ф.И.О.)
«11» 02 2022г.

Задание на курсовую работу получил _____ Матяш А.А.
(подпись) (Ф.И.О.)
«11» 02 2022г.

Содержание

1 Введение

Передача информации – это физическии процесс, посредством которого осуществляется перемещение знаков (сведении, способных предоставлять информацию) в пространстве или осуществляется физическии доступ субъектов к знакам. В общем случае возможно распространение самых различных видов сообщении: текстов, изображении, музыки, видео и др. Совокупность средств, служащих для передачи информации, называется системои передачи информации.

В современных сетях связи используются аналоговые и цифровые системы передачи информации с тенденцией постепенного перехода к применению только цифровых систем. Однако в ближайшем будущем большое число соединений будет устанавливаться с использованием обеих технологий. Для обеспечения в этих условиях заданных характеристик каналов и трактов, гарантирующих высокое качество передачи информации, принципы проектирования цифровых и аналоговых систем передачи должны быть совместимы.

Для повышения достоверности передачи данных, представления их в удобнои форме, обеспечения минимума среднего числа символов на одно сообщение информация преобразуется с помощью кодирующего устроиства. Сообщения представляют собой либо последовательность импульсов, означающую линейный код (в полосе пропускания), либо ограничивается набором непрерывно меняющейся формы волны, используя метод цифровой модуляции. Модуляция и соответствующая ей демодуляция осуществляются модемным оборудованием.

Информация в кабельных сетях передается в закодированном виде, то есть каждому биту передаваемой информации соответствует свой набор уровней электрических сигналов в сетевом кабеле. Передача происходит без модуляции или, как еще говорят, в основной полосе частот. Модуляция высокочастотных сигналов применяется в основном в бескабельных сетях, в радиоканалах.

Целями работы являются изучение и анализ линеиного кода NRZ, разработка и описание программы для кодирования и декодирования данных, обзор преимуществ и недостатков метода. Также будет проведено тестирование программнои реализации с целью проверки корректности ее работы.

2 Теоретическая часть

2.1 Общие понятия о кодировании

Кодированием называется процесс преобразования сообщения в комбинации из дискретных сигналов. Оно применяется для минимизации объема передаваемых данных, повышения ее достоверности, увеличения скорости передачи, а также повы- шения помехоустоичивости путем внесения дополнительнои избыточности. Правильный выбор кода позволяет снизить требования к выбору кабеля. Например, при разных кодах предельная скорость передачи по одному и тому же кабелю может отличаться в два раза. Код должен в идеале обеспечивать хорошую синхронизацию приема, низкий уровень ошибок, работу с любой длиной передаваемых информационных последовательностей. Код также влияет на сложность сетевои аппаратуры, а именно на узлы кодирования и декодирования.

Линейным кодированием называется процесс преобразования информации, представленной в цифровом виде, то есть хранящейся в формате серии из нулей и единиц, в цифровой сигнал.

При линейном кодировании передача большего количества битов выполняется в одном сигнале, благодаря чему используемая ширина полосы значительно уменьшается. Мощность пропускания данной полосы используется наиболее эффективно, плотность мощности очень благоприятная. Содержание времени является адекватным. При линейном кодировании избегают длинных строк 1 и 0, что позволяет сохранить прозрачность. Вероятность ошибки значительно снижается.

Различают 3 типа линейного кодирования: однополярный, полярный и биполярный Однополярная сигнализация также называется включением-выключением, или просто OOK. Наличие импульса представляет собой 1, а отсутствие импульса представляет собой 0.

В полярном типе передачи сигналов максимум данных представлен положительным импульсом, а минимум данных представлен отрицательным импульсом.

Метод биполярного кодирования предполагает наличие трех уровней напряжения, а именно +, – и 0. Такой сигнал называется двойным двоичным сигналом.
Данные типы сигнализация реализуются за счет применения кода NRZ («Невозврат к нолю») или RZ («Возврат к нулю»).

В зависимости от количества уровней напряжения, используемых для формирования сигналов чаще всего встречаются следующие виды систем линейного кодирования:

- Двухуровневое кодирование
- Трёхуровневое кодирование
- Четырёхуровневое кодирование
- Многоуровневое кодирование

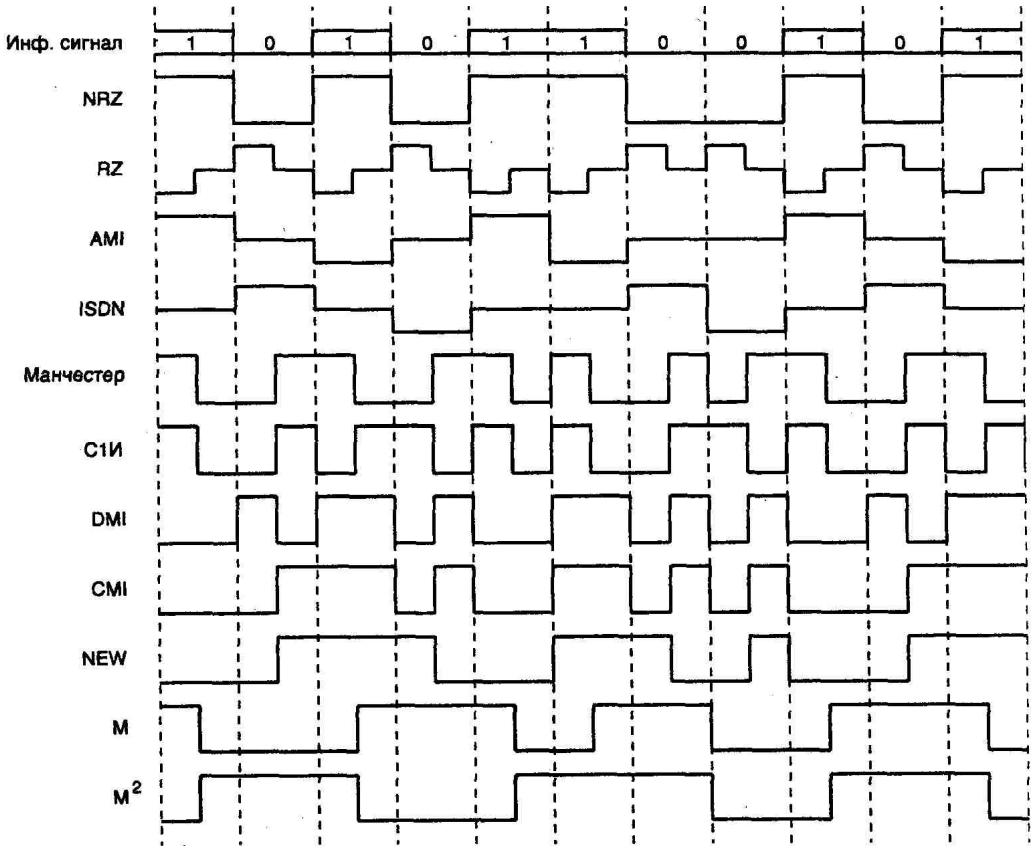

Рис. 1: Примеры линейного кодирования

Одним из видов линейного кодирования является код NRZ.

2.2 Описание кода NRZ

Код NRZ (non return to zero) — один из способов линейного кодирования, используется при передаче дискретных сообщений в канале связи, формируя сигнал, передаваемый на расстояние.

При передаче информации на расстояние информация представляется в цифровом виде и в канал связи формируется сигнал в соответствии с кодом: логическому нулю соответствует нижний уровень сигнала, логической единице соответствует верхний уровень сигнала.

NRZ-код не является самосинхронизирующимся, поэтому в устройствах передачи данных для синхронизации сигнала применяют скремблирование — в последовательность специально вводят детерминированный процесс, по которому происходит синхронизация тактовой частоты приёмника с передатчиком.

В спектре сигнала присутствует низкочастотная составляющая, которая приближается к постоянному сигналу при передаче серии передаваемых последовательностей из логических «единиц» или «нулей».

2.3 Варианты представления кода NRZ

Различают несколько вариантов представления кода:

- Униполярный код — логическая единица представлена верхним потенциалом, логический нуль представлен нулевым потенциалом;
- Биполярный код — логическая единица представлена положительным потенциалом, логический нуль представлен отрицательным потенциалом.

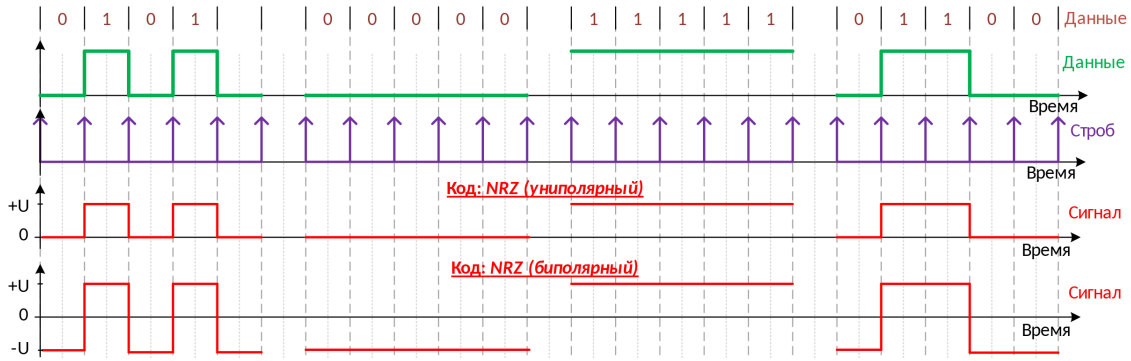

Рис. 2: Пример NRZ кодирования

2.4 Достоинства

- Простота реализации кода — код полностью соответствует поступающей на вход передатчика битовой последовательности и никаких дополнительных преобразований выполнять не нужно;
- Минимальная необходимая полоса пропускания линии связи.

2.5 Недостатки

- Потеря синхронизации во время приема слишком длинных блоков (пакетов) информации. В связи с этим код NRZ используется только для передачи короткими пакетами (обычно до 1 Кбита);
- Код может обеспечить обмен сообщениями (последовательностями, пакетами) только фиксированной, заранее обговоренной длины, поскольку по принимаемой информации приемник не может определить, идет ли еще передача или уже закончилась;
- Необходимость передачи старт-стопового бита для синхронизации приёмника с передатчиком;
- Наличие постоянной составляющей (ёмкостное сопротивление), из-за чего невозможно обеспечить гальваническую развязку с помощью трансформатора;
- Наличие ёмкостного сопротивления (в униполярном коде) — нарастание в проводном канале связи постоянной составляющей (паразитной ёмкости), которое препятствует функциональности электрооборудования (проблема решается за счет использования биполярного кода);
- Нарушение плотности следования единичных импульсов (плохая синхронизация приёмника и передатчика) — при передаче последовательности логических нулей или единиц происходит рассинхронизация передатчика и приемника;
- Для синхронизации передатчика с приемником применяется избыточность передачи данных (вводятся детерминированные последовательности, по которым производится синхронизация) или скремблирование, что усложняет реализацию и уменьшает скорость передачи данных.

2.6 Сравнение кодов NRZ и RZ

Код RZ (return to zero – с возвратом к нулю) – это трехуровневый код, который принципиально отличаются от NRZ тем, что сигнал имеет дополнительные переходы, используемые для синхронизации. В центре битового интервала всегда есть переход сигнала (положительный или отрицательный), следовательно, из этого кода приемник легко может выделить синхроимпульс (строб). Возможна временная привязка не только к началу пакета, как в случае кода NRZ, но и к каждому отдельному биту.

Еще одно важное достоинство кода RZ – простая временная привязка приема, как к началу последовательности, так и к ее концу. Приемник просто должен анализировать наличие изменения уровня сигнала в течение битового интервала. Первый битовый интервал без изменения уровня сигнала соответствует окончанию принимаемой последовательности бит. Поэтому в коде RZ можно использовать передачу последовательностями переменной длины.

Значимый недостаток кода RZ состоит в том, что для него требуется вдвое большая полоса пропускания канала при той же скорости передачи по сравнению с NRZ (так как здесь на один битовый интервал приходится два изменения уровня сигнала). Другой важный недостаток – наличие трех уровней, что всегда усложняет аппаратуру как передатчика, так и приемника.

Код RZ применяется не только в сетях на основе электрического кабеля, но и в оптоволоконных сетях.

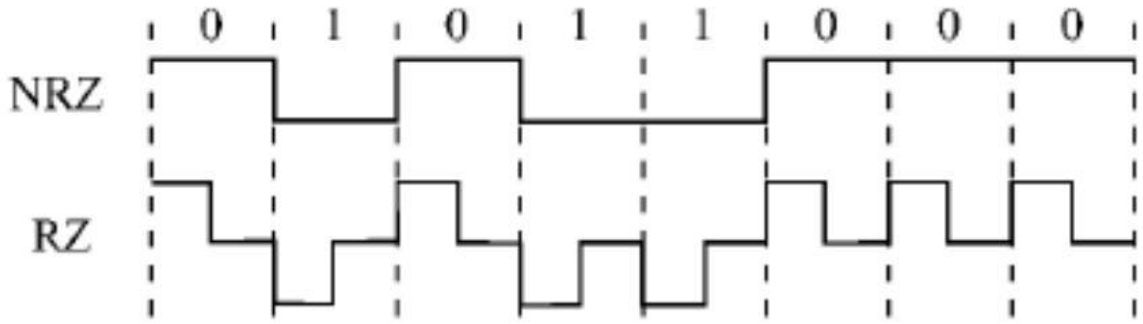

Рис. 3: Сравнение кодов NRZ и RZ

2.7 Применение

Наиболее известное применение кода NRZ – это стандарт RS232-C, последовательный порт персонального компьютера. Передача информации в нем ведется байтами (8 бит), сопровождаемыми стартовым и стоповым битами.

3 Аналитическая часть

Цели исследования:

- Выбор и анализ алгоритмов кодера и декодера;
- Разработка и описание программной реализации кодера и декодера;
- Проверка корректности работы разработанного варианта программной реализации системы передачи информации.

Предполагается, что кодер на основе кода NRZ преобразует входные данные в закодированную последовательность. Декодер выполняет обратную операцию. Пользователь сможет получать результат в формате закодированных и декодированных комбинаций в виде списка, а также осциллограмм.

В качестве среды для разработки выбрано окружение ОС Ubuntu 10.3.0 x64, язык программирования Python3 с модулем графики matplotlib.

4 Практическая часть

4.1 Структура программы

Программная реализация кодера и декодера состоит из модуля rz.py, в Приложении 1 приведен листинг. Данный модуль содержит необходимые функции для кодирования и декодирования данных.

Функции, реализованные в кодере, позволяют преобразовывать комбинацию из нулей и единиц в закодированную последовательность, а также позволяют выводить результат на экран в виде осциллограммы.

Функции декодера формируют декодированную последовательность из закодированных данных и позволяют выводить результат на экран в виде осциллограммы.

Далее опишем функции, которые используются как в кодере, так и в декодере.

4.2 Программная реализация кодера

Программа принимает в качестве параметра последовательность из нулей и единиц и, используя правила преобразования в коде NRZ, возвращает закодированные данные в виде списка.

Перед кодированием информации проверяется, инициализирован ли переданный функции список. Если последовательность содержит None, то вызывается исключение.

Если список инициализирован, то дальше вычисляется его длина. В случае, когда длина последовательности равна нулю, то есть он не содержит каких-либо данных, срабатывает исключение.

Создаётся новый список, в который мы будем записывать закодированные данные. После этого в цикле происходит последовательное преобразование исходной комбинации. Передаваемый в качестве аргумента список должен содержать только нули или единицы. При обнаружении каких-либо других значений вызывается исключение.

Если очередной элемент списка равен единице, то в список записывается 1, что соответствует отрицательному уровню напряжения. Иначе в закодированный список помещается 0.

На рисунке изображен участок сигнала, закодированной последовательности кодом NRZ.

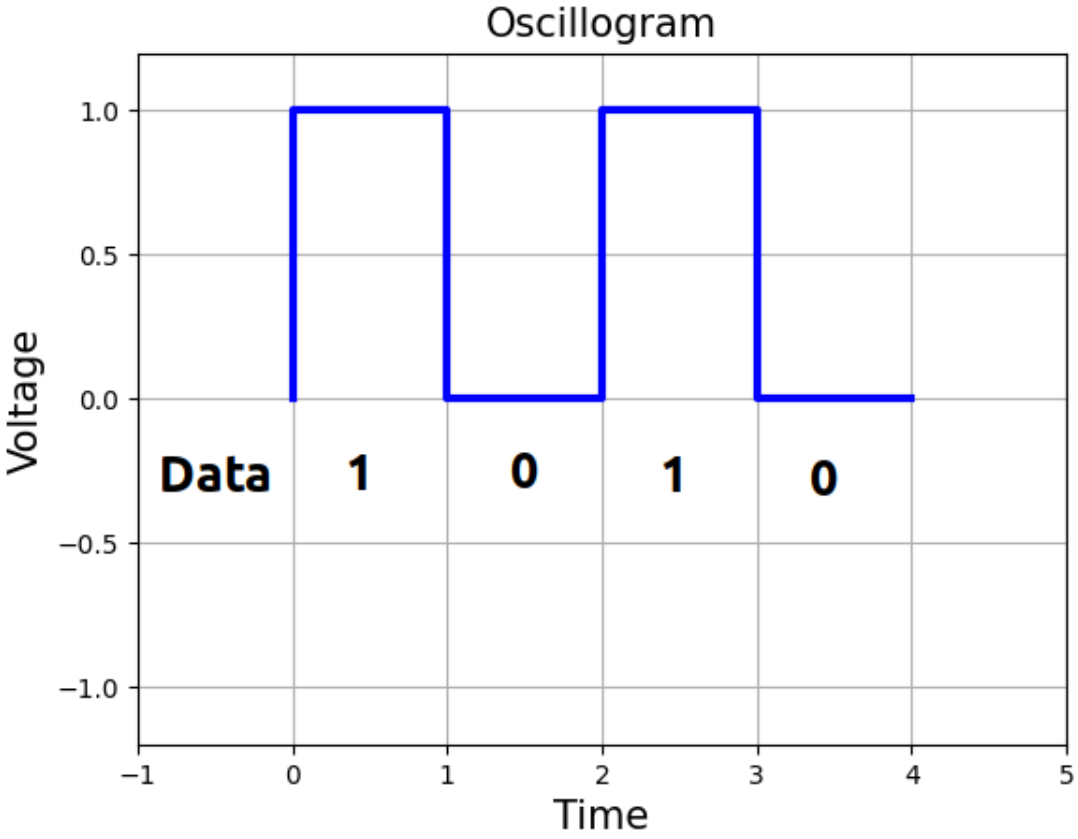

Рис. 4: Изображение участка закодированного сигнала

4.3 Программная реализация декодера

Программа принимает в качестве параметра последовательность из 0 и 1 и, согласно правилам преобразования в коде NRZ, возвращает декодированную серию из нулей и единиц в виде списка. Перед декодированием информации проверяется, инициализирован ли переданный функции список. Если последовательность содержит None, то вызывается исключение.

Если список создан, то далее вычисляется его длина. В случае, когда длина списка равна нулю, то есть он не содержит каких-либо данных, срабатывает исключение.

Создаётся новый список декодированной последовательности в который мы будем добавлять декодированные данные. После этого в цикле происходит последовательное преобразование закодированной последовательности. Передаваемый в качестве аргумента список должен содержать только 0 и 1. В случае обнаружения каких-либо других значений вызывается исключение.

Функция возвращает список с декодированными данными.

На рисунке изображен участок сигнала, декодированной последовательности кода NRZ.

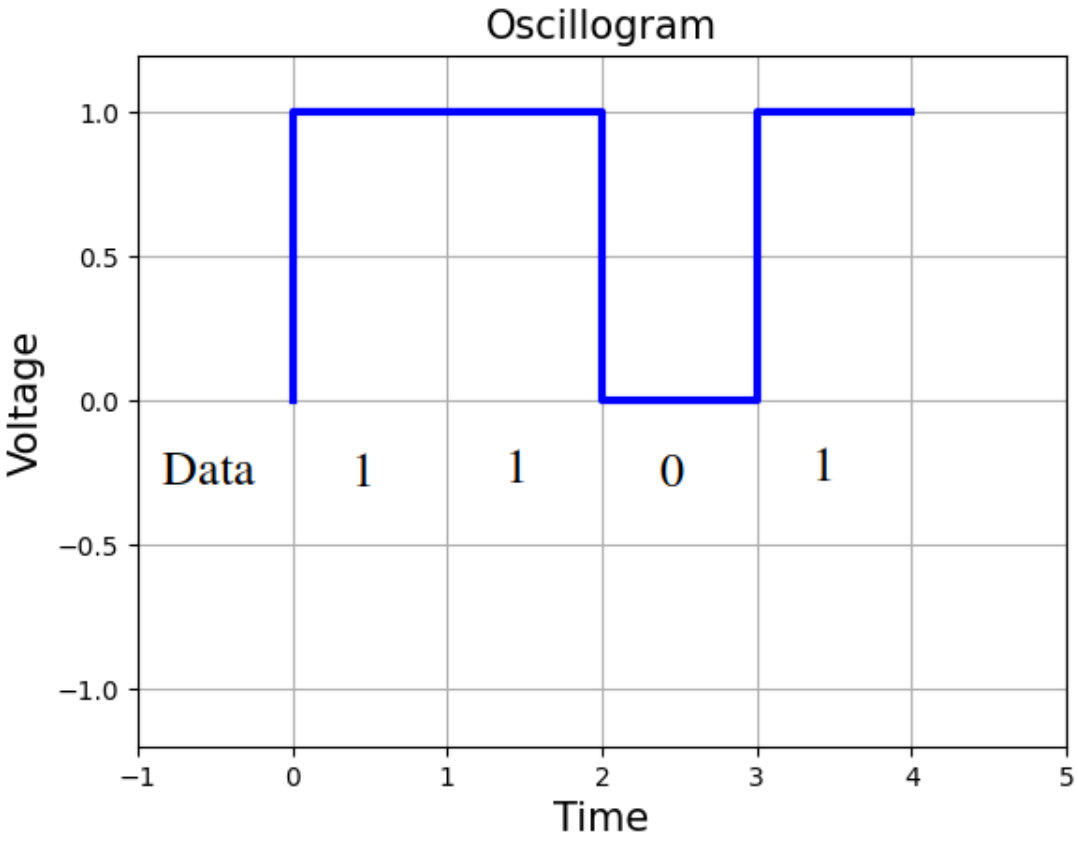

Рис. 5: Изображение участка декодированного сигнала

4.4 Пример кодирования

Продемонстрируем пример работы кодера. Сначала получим закодированную последовательность в виде списка и выведем результат в консоль.

Исходные данные:

[1, 0, 1, 0, 1, 1, 0, 0].

Воспользуемся программой coder_example1.py.

```python
#!/usr/bin/env python3
import nrz

def main():
    data = [1, 0, 1, 0, 1, 1, 0, 0]
    print(nrz.get_encoded_sequence(data))

if __name__ == "__main__":
    main()
```

Рис. 6: Код программы coder_example1.py

При запуске программы получим результат:

```
└─[0] <> python3 coder_example1.py
[1, 0, 1, 0, 1, 1, 0, 0]
```

Рис. 7: Результат выполнения программы coder_example1.py

Теперь получим осциллограмму для той же последовательности при помощи программы coder_example2.py.

```python
#!/usr/bin/env python3
import nrz

def main():
    data = [1, 0, 1, 0, 1, 1, 0, 0]
    nrz.print_encoded_oscillogram(data)

if __name__ == "__main__":
    main()
```

Рис. 8: Код программы coder_example2.py

Результат программы:

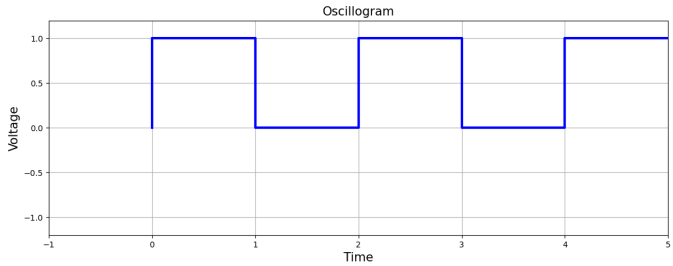

Рис. 9: Результат выполнения программы coder_example2.py

4.5 Пример декодирования

Продемонстрируем пример работы декодера. Сначала получим декодированную последовательность в виде списка и выведем результат в консоль.

Исходные данные:

[1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]

Воспользуемся программой coder_example1.py.

```python
#!/usr/bin/env python3
import nrz

def main():
    data = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
    print(nrz.get_decoded_sequence(data))

if __name__ == "__main__":
    main()
```

Рис. 10: Код программы decoder_example1.py

При запуске программы получим результат:

```
└─[130] <> python3 decoder_example1.py
[1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
```

Рис. 11: Результат выполнения программы decoder_example1.py

Теперь получим осциллограмму для той же последовательности при помощи программы decoder_example2.py.

```python
#!/usr/bin/env python3
import nrz

def main():
    data = [1,1,0,1]
    nrz.print_decoded_oscillogram(data)

if __name__ == "__main__":
    main()
```

Рис. 12: Код программы decoder_example2.py

Результат программы:

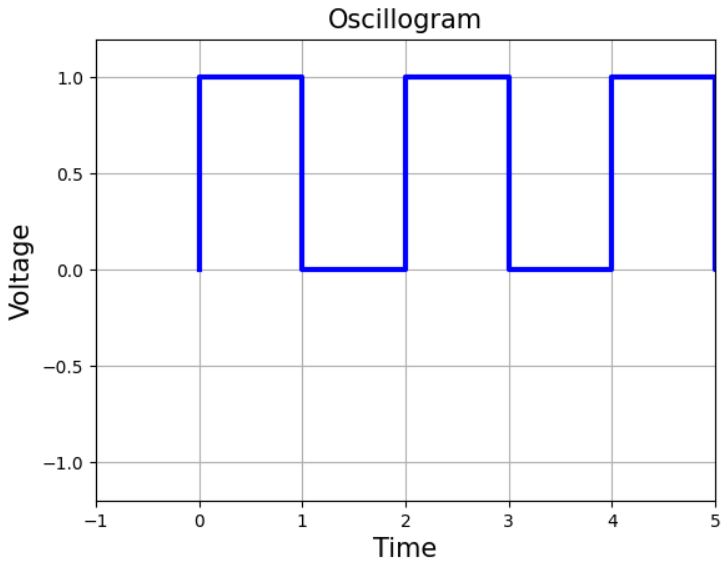

Рис. 13: Результат выполнения программы decoder_example2.py

5 Заключение

В данной работе был изучен и проанализирован код NRZ, рассмотрены его особенности, преимущества и недостатки. Реализована программа для кодирования и декодирования информации.

Разработанная программная реализация выполняет функции кодирования и декодирования корректно, что проверено тестированием, результаты которого также приведены.

Приложение 1

Файл nrz.py

```
#!/usr/bin/env python3
import matplotlib.pyplot as plt

def set_settings():
    plt.xlim([-1, 5])
    plt.ylim([-1.2, 1.2])
    plt.xlabel('Time', fontsize=15, color='black')
    plt.ylabel('Voltage', fontsize=15, color='black')
    plt.grid(True)
    plt.title("Oscillogram", fontsize=15)

def print_oscillogram(x_coordinates, y_coordinates):
    set_settings()
    plt.plot(x_coordinates, y_coordinates, 'b', linewidth=3)
    plt.show()

def print_encoded_oscillogram(sequence):
    y_coordinates = get_y_coordinates_for_encoding(get_encoded_sequence(sequence))
    x_coordinates = get_x_coordinates_for_encoding(len(y_coordinates))
    print_oscillogram(x_coordinates, y_coordinates)

def get_encoded_sequence(sequence):
    if sequence == None:
        raise Exception("Your sequence is None!")
    if len(sequence) == 0:
        raise Exception("The size of your sequence is zero!")

    encoded_sequence = []
    for i in range (len(sequence)):
        if sequence[i] != 0 and sequence[i] != 1:
            raise Exception("Incorrect data format!")
        if sequence[i] == 1:
            encoded_sequence.append(1)
        else:
            encoded_sequence.append(0)
    return encoded_sequence

def get_y_coordinates_for_encoding(sequence):
    y_coordinates = []
    y_coordinates.append(0)
    for i in range(0, len(sequence)):
        y_coordinates.append(sequence[i])
        y_coordinates.append(sequence[i])
    return y_coordinates

def get_x_coordinates_for_encoding(length):
    sequence = []
    step = 0
    sequence.append(step)
    for i in range(1, length, 4):
        sequence.append(step)
        step += 1
        sequence.append(step)
        sequence.append(step)
        step += 1
        sequence.append(step)
```

```
    return sequence

def print_decoded_oscillogram(sequence):
    y_coordinates = get_y_coordinates_for_decoding(get_decoded_sequence(sequence))
    x_coordinates = get_x_coordinates_for_decoding(len(y_coordinates))
    print_oscillogram(x_coordinates, y_coordinates)

def get_decoded_sequence(sequence):
    if sequence == None:
        raise Exception("Your sequence is None!")
    if len(sequence) == 0:
        raise Exception("The size of your sequence is zero!")
    decoded_sequence = []
    for i in range (len(sequence)):
        if sequence[i] != 0 and sequence[i] != 1:
            raise Exception("Incorrect data format!")
        if sequence[i] == 1:
            decoded_sequence.append(1)
        else:
            decoded_sequence.append(0)
    return decoded_sequence

def get_y_coordinates_for_decoding(sequence):
    y_coordinates = []
    y_coordinates.append(0)
    for i in range(0, len(sequence)):
        y_coordinates.append(sequence[i])
        y_coordinates.append(sequence[i])
    return y_coordinates

def get_x_coordinates_for_decoding(length):
    sequence = []
    step = 0
    sequence.append(step)
    for i in range(1, length, 2):
        sequence.append(step)
        step += 1
        sequence.append(step)
    return sequence
```

Вопросы к защите:

1. Назовите разновидности линейного кодирования?

Можно выделить следующие виды линейного кодирования: бинарное (двухуровневое), тринарное (трёхуровневое), тетрарное (четырёхуровневое) и кодирование с большим числом уровней.

2. Как реализуется синхронизация передатчика и приемника?

Для синхронизации сигнала применяют скремблирование. В последовательность специально вводят детерминированный процесс, по которому происходит синхронизация тактовой частоты приёмника с передатчиком.