Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

ОТЧЁТ

По лабораторной работе № 2

На тему

**ПАКЕТНАЯ ПЕРЕДАЧА ДАННЫХ**

Выполнил: Проверил:

Студент группы 050503 Преподаватель

Григорик И. А. Марцинкевич В. А.

Минск 2022

1. **Цель работы:**

Дополнить модуль первой лабораторной работы на основе последовательных интерфейсов RS-232 и RS-485. Ознакомиться с структурированием и созданием пакетов для передачи. Создать свой алгоритм байтстаффинга. Модуль должен быть оформлен в виде библиотеки функций или класса.

1. **Задание к лабораторной работе:**
2. Разработать программный модуль реализации процедуры байтстаффинга и кодирования пакета для передачи (приема) информации через последовательный интерфейс.
3. В программах выводить закодированный и незакодированный пакет. Проверить функционирование звена приемопередачи.

**3. Теоретическая часть**

В общем случае подавляющее большинство пакетов содержат следующую структуру: 1. Флаг начала, сигнализирующий о начале пакета; 2 – Адрес назначения; 3 – Адрес источника; 4 – Другие поля; 5 – сами данные (полезная нагрузка); 6 – Контрольная сумма (проверка целостности пакета). В данной лабораторной работе следует реализовать флаг начала пакета и кодирование данных.

Пакеты кодируются таким образом из-за непрерывности соединения. Т.е. в любой момент соединения устройство приёмника может подключиться к каналу и начать извлекать информацию. Для борьбы с ошибками чтения пакетов реализуется данный алгоритм. Данные кодируются и декодируются таким образом, чтобы флаг начала пакета был уникален. Пример представлен на рисунке 1.



Рис 1 – пример реализации байтстаффинга.

В моём случае байтстаффинг будет реализован как отдельные функции, которые будут преобразовывать данные пакета и возвращать их.

1. **Практическая часть**

Функция кодирования пакета:

def encode\_package(message: str) -> str:  
 # Byte staffing  
 start\_byte = 'a'  
 encode\_message = ''  
 encode\_message += start\_byte  
 for char in message:  
 if char == start\_byte:  
 encode\_message += '\*'  
 encode\_message += char  
  
 return encode\_message

Функция декодирования пакета:

def decode\_package(message: str) -> str:  
 # Byte staffing  
 staffing\_byte = message[0]  
 decoded\_message = ''  
 for char in message[1:]:  
 # If symbol is equal as start byte - just remove last string character  
 if char == staffing\_byte:  
 decoded\_message = decoded\_message[:-1]  
 decoded\_message += char  
  
 return decoded\_message