



Line Coding Encoder and Decoder

Submission Details

Name: Tushar Verma
Enroll No.: 2021BITE065
Submitted to: Dr. Iqra Altaf Gillani

Overview

This project implements a Line Coding Encoder and Scrambler with a digital data generator. The assignment involves generating a random data sequence, encoding it using various line coding schemes (NRZ-L, NRZ-I, Manchester, Differential Manchester, AMI), and optionally scrambling it using B8ZS or HDB3. The code also identifies the longest palindromic sequence in the generated data.

Line Encoder and Decoder GUI

This Line Encoder GUI is implemented using PySimpleGUI, Matplotlib, and NumPy. It provides a user-friendly interface for encoding and decoding various line encoding schemes.

Features:

- Initialize random input data.
- Enter custom input data.
- Encode and decode data using various line encoding schemes.
- Display the encoded and decoded data.
- Plot graphs for input, encoded, and decoded data.

Line Encoding Schemes:

- Polar NRZ-L

- Polar NRZ-I
- Manchester
- Polar RZ
- AMI
- Scrambling AMI B8ZS (Bipolar with 8 zero substitution)
- Differential Manchester
- Scrambling AMI HBD3 (High-Density Bipolar 3 Zeros)
- PCM (Pulse Code Modulation)
- Delta Modulation

How to Use:

1. Launch the GUI.
2. Initialize random input or enter custom input.
3. Encode the data using a selected encoding scheme.
4. Optionally, decode the encoded data.
5. Visualize the input, encoded, and decoded data using the "Show Graph" button.

How to Run the Code:

1. Ensure you have Python installed on your machine.
2. Clone the repository:

```
git clone https://github.com/tusharv01/LINECODER.git
```

3. Navigate to the project directory:

```
cd LINECODER
```

4. Run the main program:

```
python LineEncoder.py
```

or using the LineEncoder.ipynb file:

```
jupyter notebook
```

5. Navigate to the .ipynb file and run the cells.
6. Follow the on-screen prompts to input preferences for data generation, encoding, and scrambling.

Assumptions:

- All schemes are implemented using positive logic.
- Manchester encoding uses IEEE (802.3) convention.

- The format used is Manchester, and differential Manchester is G.E Thomas format.
- It is assumed that the user will give binary data to the encoder.

Connect with Me:

- **LinkedIn:** Tushar Verma
- **GitHub:** tusharv01

References:

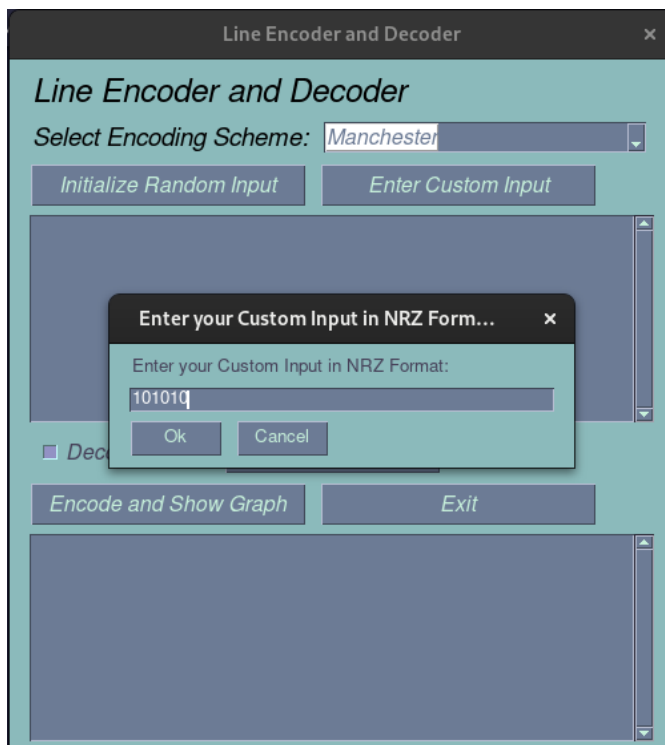
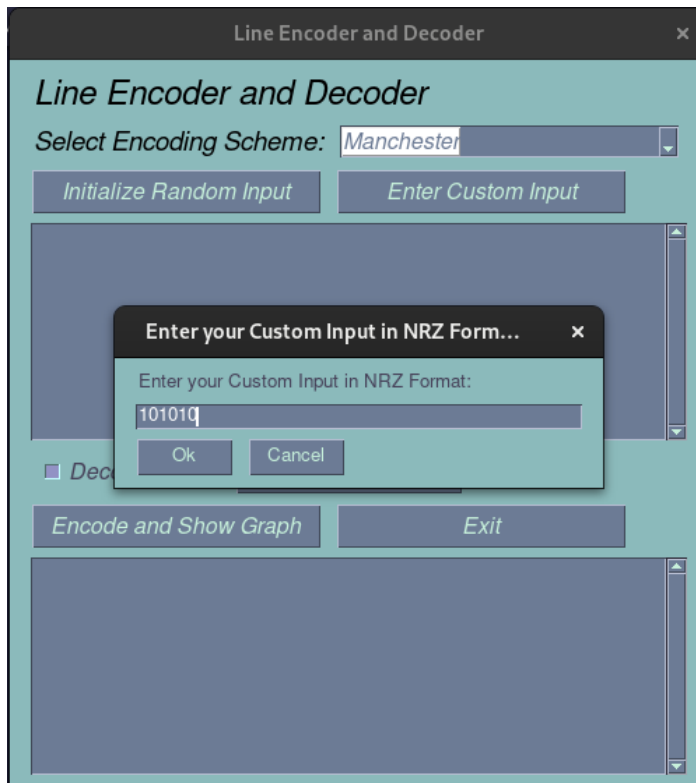
- **ChatGPT:** .
- **GeeksforGeeks (GFG):**
- **YouTube Videos:**

Copyright:

© 2023 Tushar Verma. All rights reserved.

LINE ENCODER AND DECODER GUI 👍😊:

- SELECT ENCODING SCHEME AND INITIALIZE THE INPUT:



- ENCODING AND DECODEING:

Line Encoder and Decoder

×

Line Encoder and Decoder

Select Encoding Scheme: Manchester

Initialize Random InputEnter Custom Input

Input Data
[1, 0, 1, 0, 1, 0]

☒ Decode SignalDecode

Encode and Show GraphExit

Encoded Data for Manchester
[-1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1]
Longest Palindrome in Encoded Data: [-1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1]

Decoded Data for Manchester
[0, 1, 0, 1, 0, 1]
Longest Palindrome in Decoded Data: [0, 1, 0, 1, 0]

GRAPH PLOTTING:

