Digital Signal Encoding and Scrambling Techniques

Khushboo Dar(2022bite002) & Tavaheed Tariq(2022bite008)

13 November 2024

Professor Incharge: Dr. Iqra Altaf Gillani

Introduction

This assignment explores key digital and analog signal processing techniques, including encoding methods (NRZ, Manchester, etc.) and conversion techniques (PCM, DM) for analog-to-digital transformation. Additionally, it involves implementing scrambling methods (B8ZS, HDB3) to maintain signal integrity in data transmission. Also the decoding of the digital signal has also been done.

Software Implementation

Language: Python
Libraries Used: numpy , matplotlib, os
The `Matplotlib` library was used for graphical outputs, `numpy` for generating bit patterns and `os` for directory management.
The program prompts the user to select an input type: digital or analog. For digital input, the user can choose an encoding scheme (NRZ-L, NRZ-I, Manchester, Differential Manchester, or AMI); if AMI is selected, optional scrambling (B8ZS or HDB3) can be applied. For analog input, the user selects PCM or DM to convert the signal to digital, which can then be encoded. The program displays a graphical output and ends when the display window is closed. Moreover , the decoding of digital signals has also been included in the project.

Assumptions

In differential manchester we assume that the level initially is taken as 1.

How to implement the code

Follow the instructions below to set up and run the project.

Clone the Project Repository

First, clone the GitHub repository by running the following command in your terminal (or command prompt for Windows):

```
git clone https://github.com/Khushboo-Dar/Digital-signal-generator.git
```

Then, change your directory to the project directory:

```
cd Digital-signal-generator
```

Set Up the Virtual Environment

For Linux

To create a virtual environment and install required packages, first run:

```
virtualenv venv
```

Activate the virtual environment:

```
source venv/bin/activate
```

For Windows

For Windows, create the virtual environment by running:

```
python -m venv venv
```

Activate the virtual environment:

```
venv\Scripts\activate
```

Install Required Packages

Once the virtual environment is activated, install the necessary packages by running:

```
pip3 install numpy matplotlib
```

Run the Project

To run the project, use the following command:

```
python3 main.py
```

References

- https://github.com/shariethernet/Implementation-of-Line-encoding-schemes-in-python
- https://github.com/arasgungore/PCM-and-DM-modulators/blob/main/main.ipynb
- https://in.mathworks.com/matlabcentral/fileexchange/13553-data-encoding-ami-nrz-rz-
- https://www.gaussianwaves.com/2021/01/line-code-demonstration-in-matlab-and-python/
- https://github.com/Nikeshbajaj/Pulse-Code-Modulation/blob/master/pcm.m