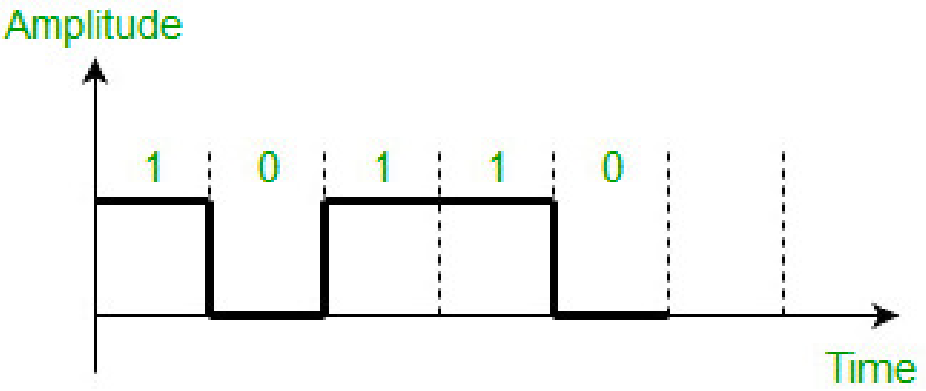
भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
भारतीय प्रौद्योगिकी संस्थान तिरुपति
Renigunta Road, Settipalli Post, Chittoor District, Tirupati – 517506

EE3107 Communication systems

Project codes



Team 19:

EE19B045 Abhimenyu
EE19B047 M. Suchithra

Line coding techniques

- **Introduction**

  Line coding (also called *digital baseband modulation* or *digital baseband transmission*) is a process carried out by a transmitter that converts data, in the form of binary digits, into a baseband digital signal that will represent the data on a transmission line.

  Unipolar Line coding

  There are two variations in Unipolar line coding−

  - Non Return to Zero (NRZ)
  - Return to Zero(RZ)

  Unipolar NRZ

```
% Button pushed function: UniNRZButton
function UniNRZButtonPushed(app, event)

    A = 1;  %---Amplitude
    b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    y = ones(1,N);
    for k = 1:N
        y = [y A*ac(k)*ones(1,length(x))];
    end
    plot(app.UIAxes,y);

end
```

Unipolar RZ

```
% Button pushed function: UniRZButton
function UniRZButtonPushed(app, event)
    A = 1;  %---Amplitude
     b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    z = ones(1,N);
    for k = 1:N
      z = [z A*ac(k)*ones(1,round(length(x)/2)) 0*ac(k)*ones(1,round(length(x)/2))];
    end

    plot(app.UIAxes,z);
end
```

Polar Line coding

Polar line coding uses both positive and negative voltage levels to represent binary values.

There are two variations in polar line coding−

- Non Return to Zero (NRZ)
- Return to Zero(RZ)

Polar NRZ

```
% Button pushed function: PolarNRZButton
function PolarNRZButtonPushed(app, event)

    A = 1;  %---Amplitude
    b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));%---Bit stream
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    e = ones(1,N);
    for k = 1:N
        e = [e ((-1)^(ac(k) + 2))*A*ones(1,round(length(x)))];
    end



    plot(app.UIAxes,e);
end
```

Polar RZ

```
% Button pushed function: PolarRZButton
function PolarRZButtonPushed(app, event)

    A = 1;  %---Amplitude
    b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));%---Bit stream
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    d = ones(1,N);
     for k = 1:N
        c = ones(1,round(length(x)/2));
        b = zeros(1,round(length(x)/2));
        p = [c b];
        d = [d ((-1)^(ac(k)+2))*(A)*p];
    end


    plot(app.UIAxes,d);
end
```

Bipolar Line coding

The bipolar line coding scheme uses all the three voltage levels. We have explored the Alternate Mark Inversion (AMI) type of bipolar encoding.

AMI

```
% Button pushed function: AMIButton
function AMIButtonPushed(app, event)
    A = 1;  %---Amplitude
    b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    g = ones(1,N);
    i=0;
    for k = 1:N
        g = [g (-1)^(i)*A*ac(k)*ones(1,length(x))];
        if ac(k)==1
            i=i+1;
        end
    end
    plot(app.UIAxes,g);
end
```

Manchester Line coding

Manchester encoding is a widely used line coding scheme that embeds timing information in the transmitted signal. It does this by ensuring that there is a transition (high-to-low or low-to-high) in the middle of every bit time, making it easy for the receiver to retrieve a clock signal from the incoming bit stream and maintain synchronisation with the transmitted signal.

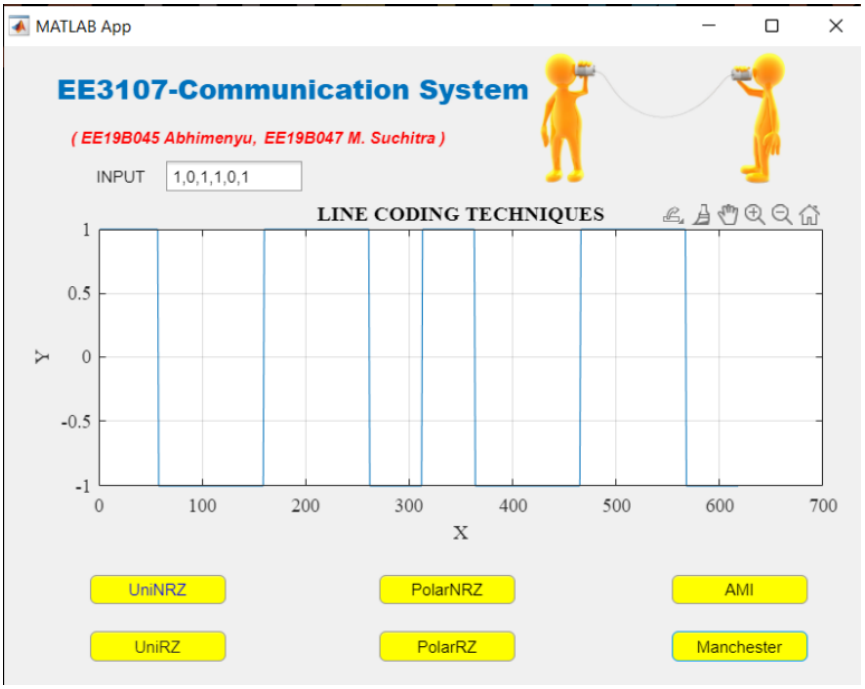
A logic high (binary one) is represented by a positive pulse with a period of half a bit time followed by a negative pulse of the same duration. Similarly, A logic low (binary zero) consists of a negative pulse followed by a positive pulse, each with a period of half a bit time.

```
% Button pushed function: ManchesterButton
function ManchesterButtonPushed(app, event)

    A = 1;  %---Amplitude
    b=app.INPUTEditField.Value;%your input is a=[1,2,3,4];
    ac=str2double(strsplit(b,','));%---Bit stream
    N = length(ac);   %---Bit Length of input bits
    x=0:1:100; %---interval of bit time period
    f = ones(1,N);
     for k = 1:N
        c = ones(1,round(length(x)/2));
        b = -1*ones(1,round(length(x)/2));
        p = [c b];
        f = [f ((-1)^(ac(k)+1))*A*p];
    end

    plot(app.UIAxes,f);
end
```

- **Conclusion:**

  Line coding techniques are an essential part of modern day digital communication systems. It doesn't require modulation and prevents distortion etc so that the signal is received without much difficulty. We came across 6 different types of line coding techniques and developed an interactive MATLAB app that takes binary inputs and produces the encoded wave to transmit over the channel to the receiver.

  eg- 1,0,1,1,0 is the input and I want Manchester type of coding to occur on the input. I press the manchester button and get my waveform on to the screen as shown below.



By and large, this project helped us to appreciate the major options to encode my digital signal so that we can develop more technology that plays with the properties of the line code and provides necessary information of the coded signal.