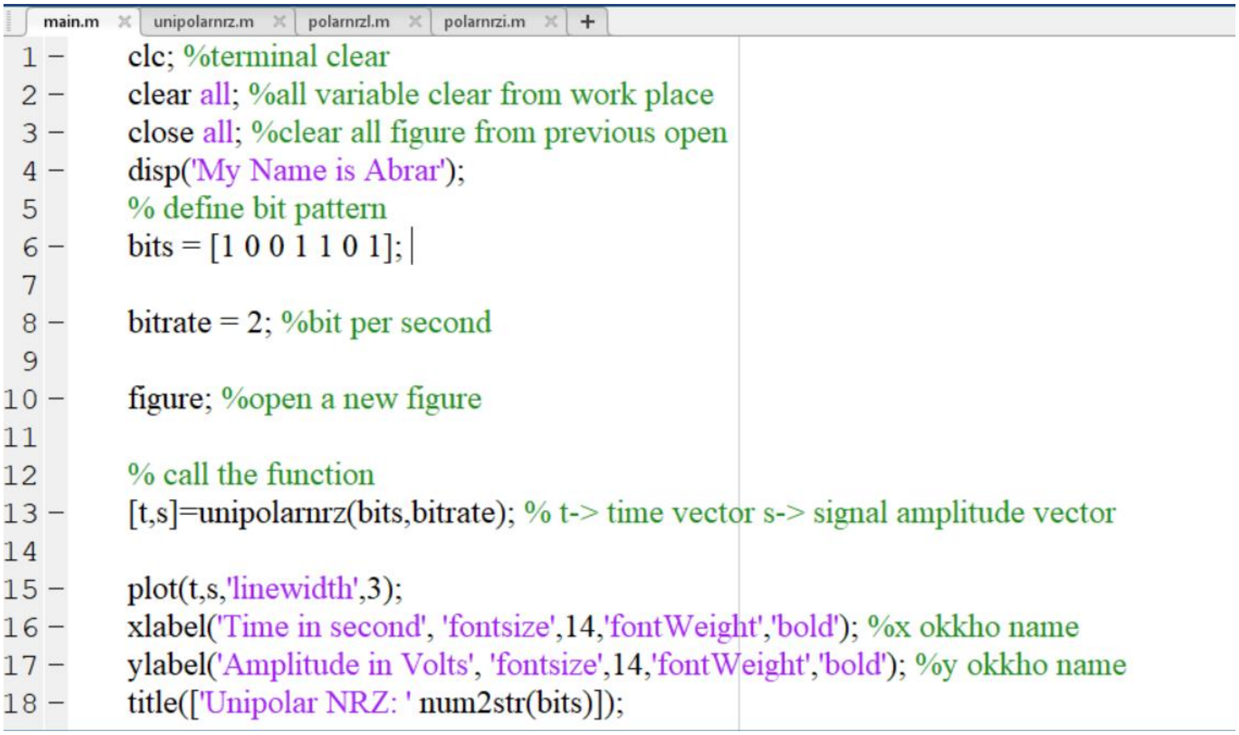
Main Function:

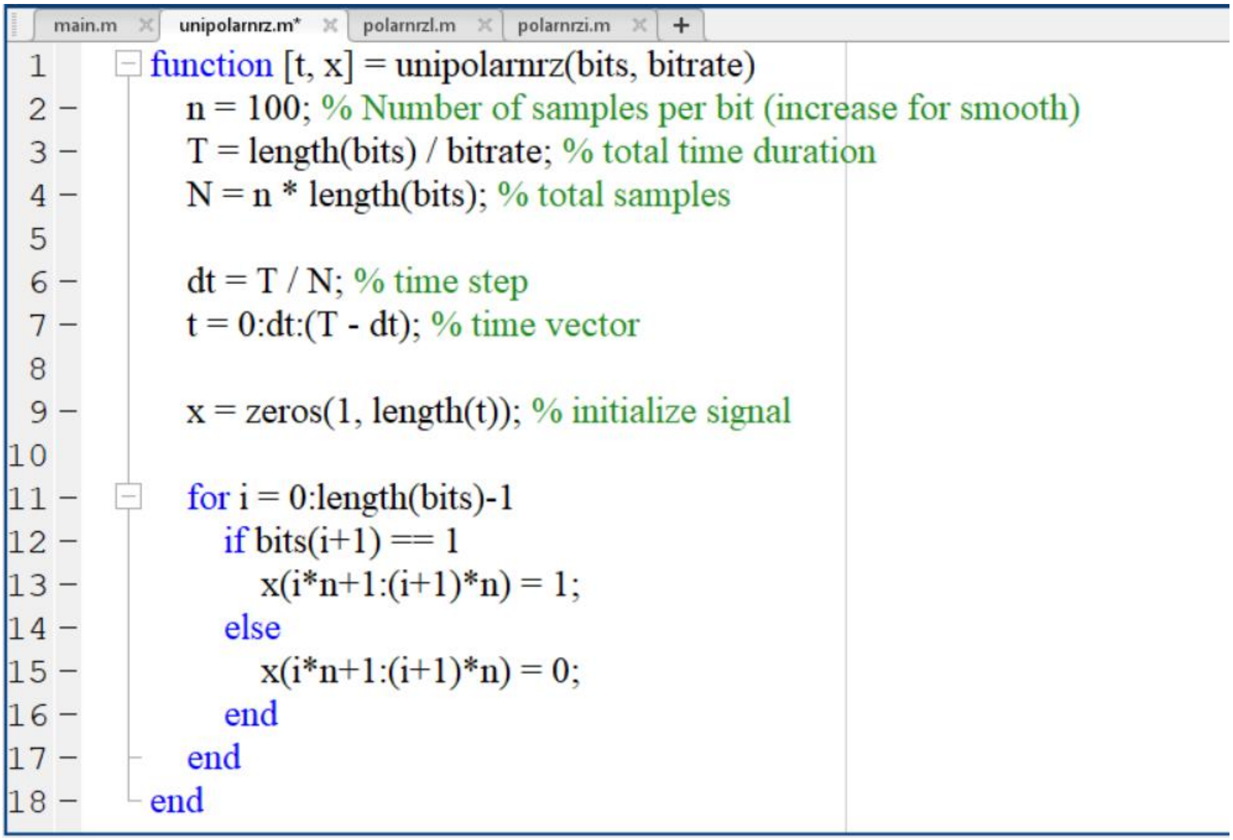
Source Code:



```
clc; %terminal clear
clear all; %all variable clear from work place
close all; %clear all figure from previous open
disp('My Name is Abrar');
% define bit pattern
bits = [1 0 0 1 1 0 1];

bitrate = 2; %bit per second

figure; %open a new figure

% call the function
[t,s]=unipolarnrz(bits,bitrate); % t-> time vector s-> signal amplitude vector

plot(t,s,'linewidth',3);
xlabel('Time in second', 'fontsize',14,'fontWeight','bold'); %x okkho name
ylabel('Amplitude in Volts', 'fontsize',14,'fontWeight','bold'); %y okkho name
title(['Unipolar NRZ: ' num2str(bits)]);
```

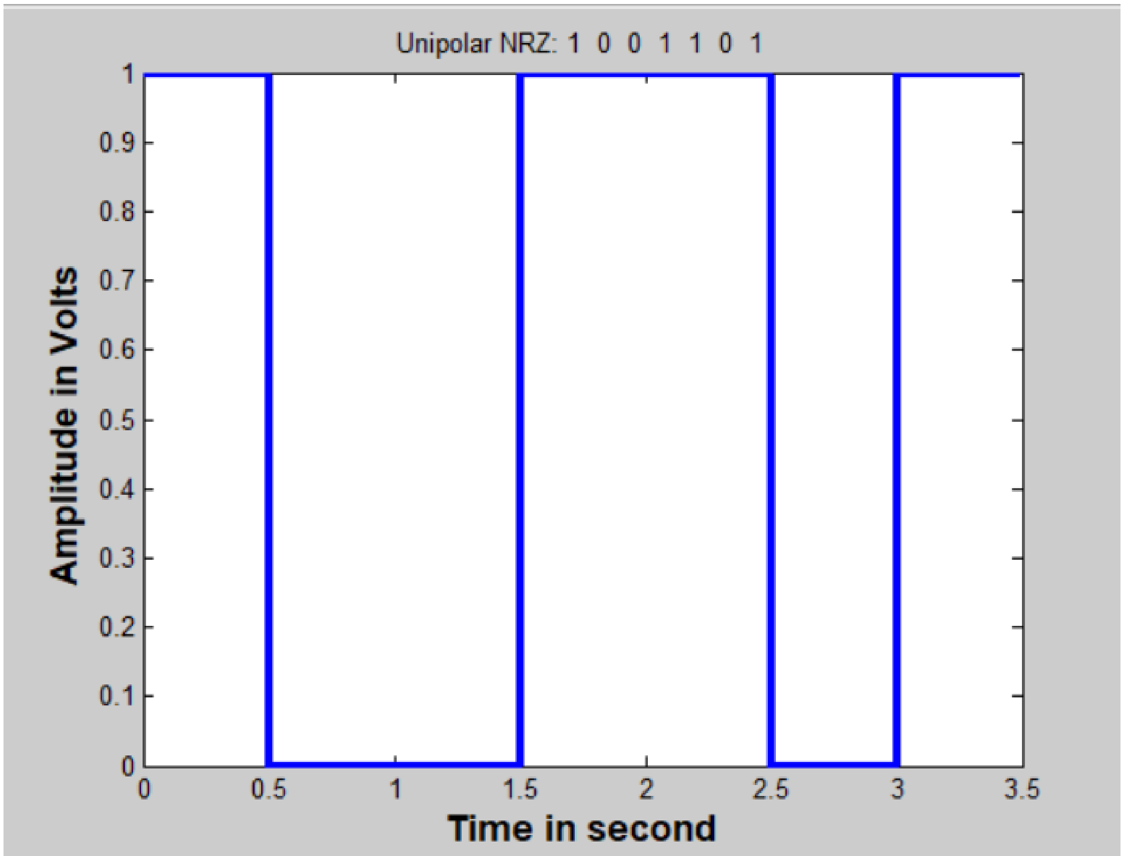
Problem 1: unipolarnrz signal verification

Source Code:



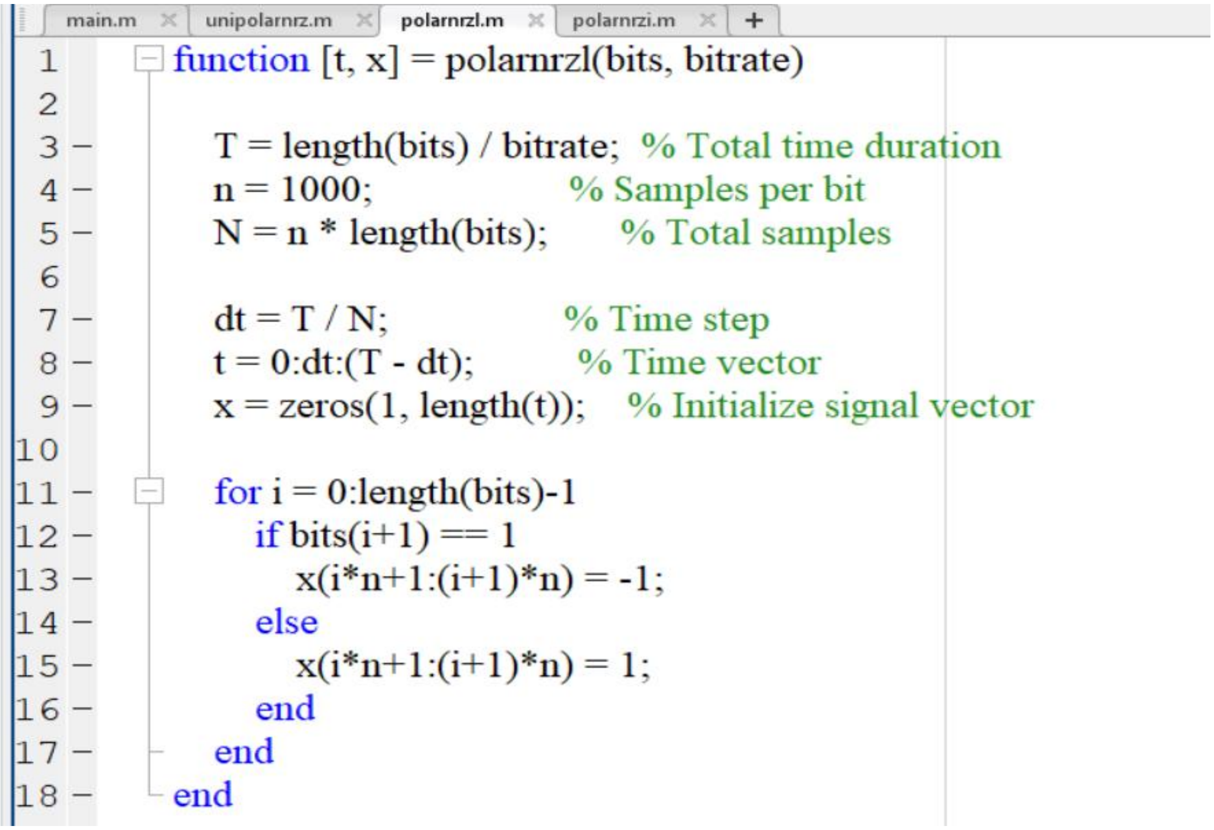
```
function [t, x] = unipolarnrz(bits, bitrate)
    n = 100; % Number of samples per bit (increase for smooth)
    T = length(bits) / bitrate; % total time duration
    N = n * length(bits); % total samples

    dt = T / N; % time step
    t = 0:dt:(T - dt); % time vector

    x = zeros(1, length(t)); % initialize signal

    for i = 0:length(bits)-1
        if bits(i+1) == 1
            x(i*n+1:(i+1)*n) = 1;
        else
            x(i*n+1:(i+1)*n) = 0;
        end
    end
end
```

Output:



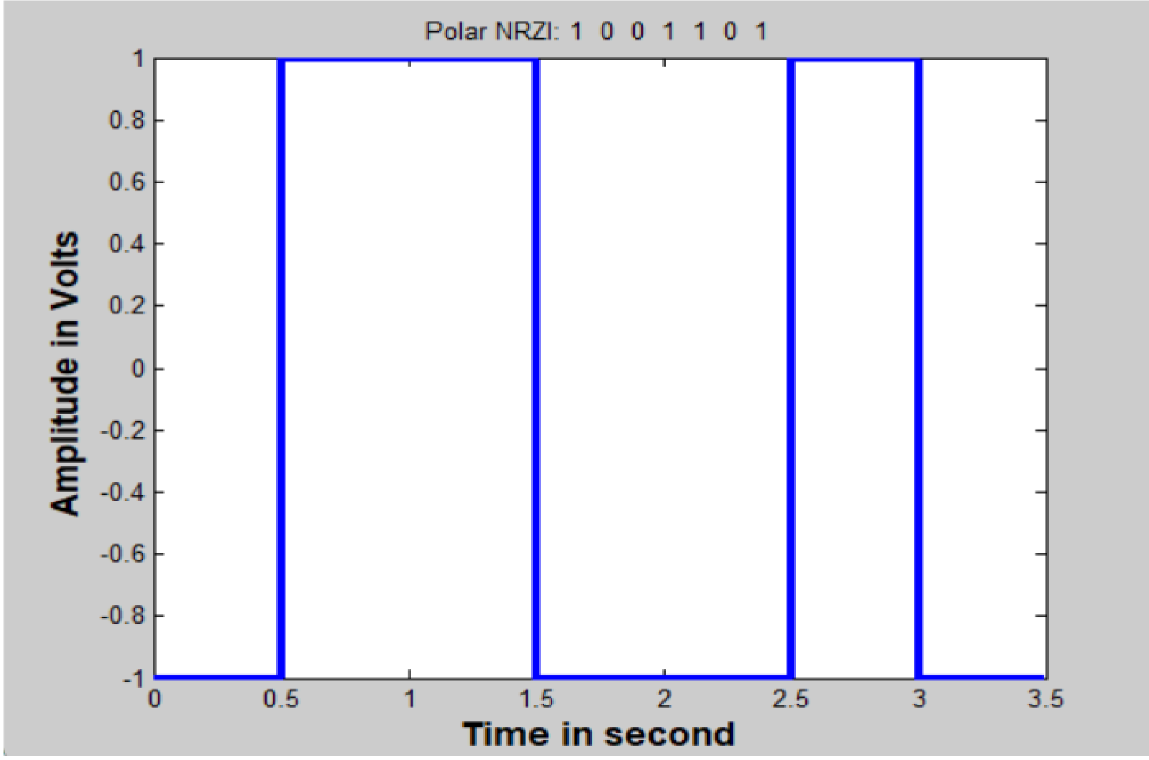
Problem 2: polarnrzl signal verification

Source Code:



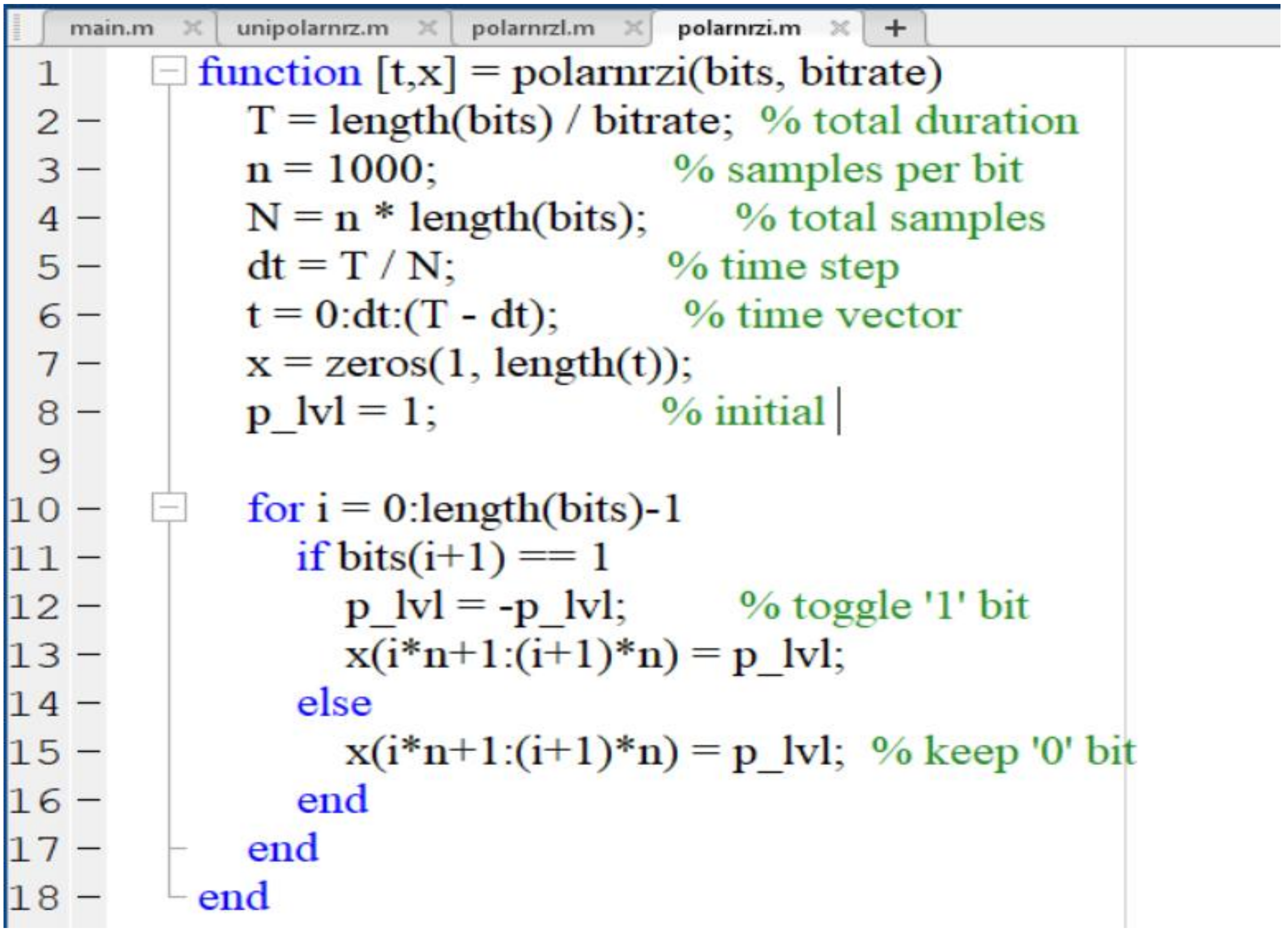
```
function [t, x] = polarnrzl(bits, bitrate)

    T = length(bits) / bitrate;  % Total time duration
    n = 1000;                   % Samples per bit
    N = n * length(bits);        % Total samples

    dt = T / N;                 % Time step
    t = 0:dt:(T - dt);           % Time vector
    x = zeros(1, length(t));     % Initialize signal vector

    for i = 0:length(bits)-1
        if bits(i+1) == 1
            x(i*n+1:(i+1)*n) = -1;
        else
            x(i*n+1:(i+1)*n) = 1;
        end
    end
end
```

Output:



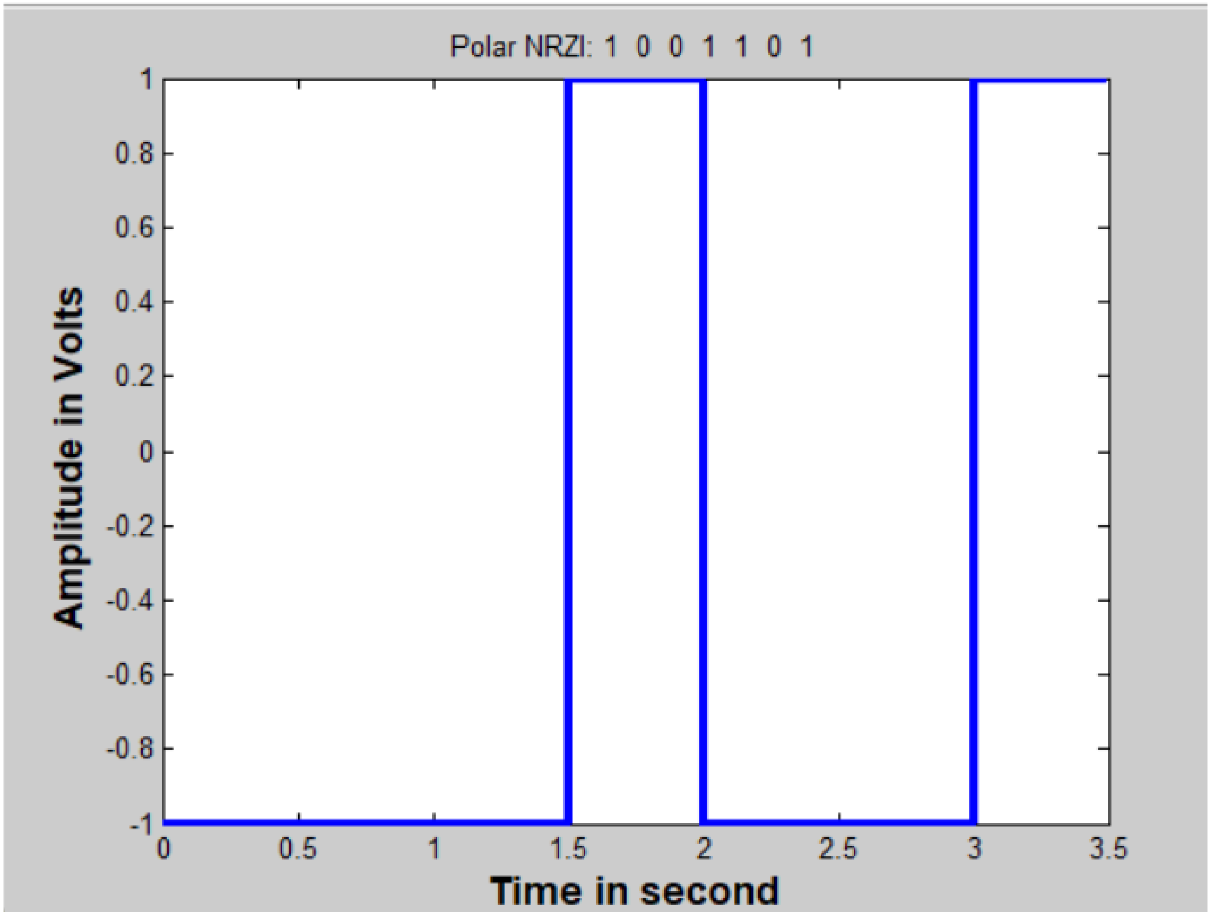
Problem 3: polarnrzi signal verification

Source Code:



```
function [t,x] = polarnrzi(bits, bitrate)
    T = length(bits) / bitrate;  % total duration
    n = 1000;                    % samples per bit
    N = n * length(bits);        % total samples
    dt = T / N;                  % time step
    t = 0:dt:(T - dt);           % time vector
    x = zeros(1, length(t));
    p_lvl = 1;                   % initial

    for i = 0:length(bits)-1
        if bits(i+1) == 1
            p_lvl = -p_lvl;        % toggle '1' bit
            x(i*n+1:(i+1)*n) = p_lvl;
        else
            x(i*n+1:(i+1)*n) = p_lvl;  % keep '0' bit
        end
    end
end
```

Output:


Polar NRZI: 1 0 0 1 1 0 1
Amplitude in Volts
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.5
1
1.5
2
2.5
3
3.5
Time in second