```
%Implementation of NRZ Unipolar in MATLAB

N=10; %Number of bits we want to represent in unipolar
n=randi([0,1],1,N); %generates a random input vector of N bits with
each bit as 0/1

%Unipolar Mapping
for m=1:N %runs the loop from first bit to last bit
    if n(m)==0 %if the current bit is 0 then it sets value of that
bit in nn(m) as 0
        nn(m)=0;
    else
        nn(m)=1; %if the current bit is 1 then it sets value of
that bit in nn(m) as 1
    end
end

i=1; %variable to keep a track on index of input data bits

time=0:.01:length(n); %generates a regularly spaced vector time
with intervals of 0.01 from values 0 to length of n

for j=1:length(time) %variable to keep a track on index of time
vector
    if time(j)<=i
        y(j)=nn(i); %if the value at current index of time vector
is less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
                    %data bit to vector y (Here vector y stores the
values
                    %corresponding to each value of time vector)
    else
        y(j)=nn(i); %if the value at current index of time vector
is less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
        i=i+1;      %then it increments the index of input data
vector
    end
end

plot(time,y); %plots graph between time axis and y axis
title('NRZ Unipolar'); %sets the title of the plot
xlabel('Time'); %sets the label of X axis
ylabel('Amplitude'); %sets the label of Y axis
axis([0 N -2 2]) %defines the range of x axis and y axis. Here x
axis is set from 0 to N and Y axis is set from -2 to 2
```

```
%Implementation of NRZ-L Polar in MATLAB

N=10; %Number of bits we want to represent in unipolar
n=randi([0,1],1,N); %generates a random input vector of N bits with
each bit as 0/1

%NRZ-L Polar Mapping
for m=1:N %runs the loop from first bit to last bit
    if n(m)==0 %if the current bit is 0 then it sets value of that
bit in nn(m) as 0
        nn(m)=-1;
    else
        nn(m)=1; %if the current bit is 1 then it sets value of
that bit in nn(m) as 1
    end
end

i=1; %variable to keep a track on index of input data bits

time=0:.01:length(n); %generates a regularly spaced vector time
with intervals of 0.01 from values 0 to length of n

for j=1:length(time) %variable to keep a track on index of time
vector
    if time(j)<=i
        y(j)=nn(i); %if the value at current index of time vector
is less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
                    %data bit to vector y (Here vector y stores the
values
                    %corresponding to each value of time vector)
    else
        y(j)=nn(i); %if the value at current index of time vector
is less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
        i=i+1;      %then it increments the index of input data
vector
    end
end

plot(time,y); %plots graph between time axis and y axis
title('NRZ-L Polar'); %sets the title of the plot
xlabel('Time'); %sets the label of X axis
ylabel('Amplitude'); %sets the label of Y axis
axis([0 N -2 2]) %defines the range of x axis and y axis. Here x
axis is set from 0 to N and Y axis is set from -2 to 2
```

```
%Implementation of NRZ-I Polar in MATLAB

N=10; %Number of bits we want to represent in unipolar
n=randi([0,1],1,N); %generates a random input vector of N bits with
each bit as 0/1

%NRZ-I Polar Mapping

nn(1)=1; %assuming that previous signal before starting of data set
was positive
for m=2:N %runs the loop from first bit to last bit
    if n(m)==0 %if the current bit is 0 then it sets value of that bit
same as previous one
        nn(m)=nn(m-1);
    else
        nn(m)=-nn(m-1); %if the current bit is 1 then it sets value of
that bit as inverted value to previous one i.e. opposite of previous
one
    end
end

i=1; %variable to keep a track on index of input data bits

time=0:.01:length(n); %generates a regularly spaced vector time with
intervals of 0.01 from values 0 to length of n

for j=1:length(time) %variable to keep a track on index of time vector
    if time(j)<=i
        y(j)=nn(i); %if the value at current index of time vector is
less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
                    %data bit to vector y (Here vector y stores the
values
                    %corresponding to each value of time vector)
    else
        y(j)=nn(i); %if the value at current index of time vector is
less than or equal to the current index of input data bit then,
                    %it assigns value present at current index of
input
        i=i+1;      %then it increments the index of input data vector
    end
end

plot(time,y); %plots graph between time axis and y axis
title('NRZ-I Polar'); %sets the title of the plot
xlabel('Time'); %sets the label of X axis
ylabel('Amplitude'); %sets the label of Y axis
axis([0 N -2 2]) %defines the range of x axis and y axis. Here x axis
is set from 0 to N and Y axis is set from -2 to 2
```