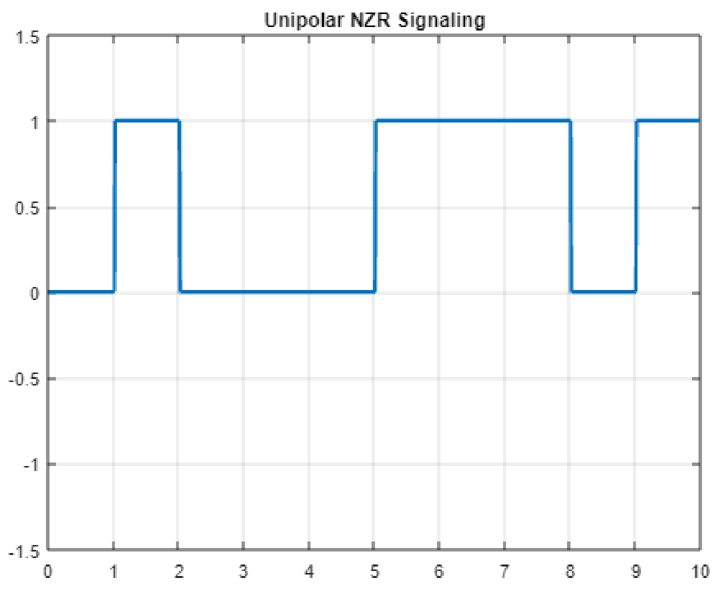
MATLAB code:

```
%Unipolar Non Return to Zero Line Coding
clc;
clear all;
close all;
N=10;
n=randi([0,1],1,N)
%Mapping Function
for m=1:N
    if n(m)==1
        nn(m)=1;
    else
        nn(m)=0;
    end
end
nn
%Signal Shaping
i=1;
t=0:0.01:length(n);
for j=1:length(t)
    if t(j)<=i
        y(j)=nn(i);
    else
        y(j)=nn(i);
        i=i+1;
    end
end
plot(t,y, 'linewidth',2);
axis([0,N,-1.5,1.5]); grid on;
title("Unipolar NZR Signaling");
```

Output:

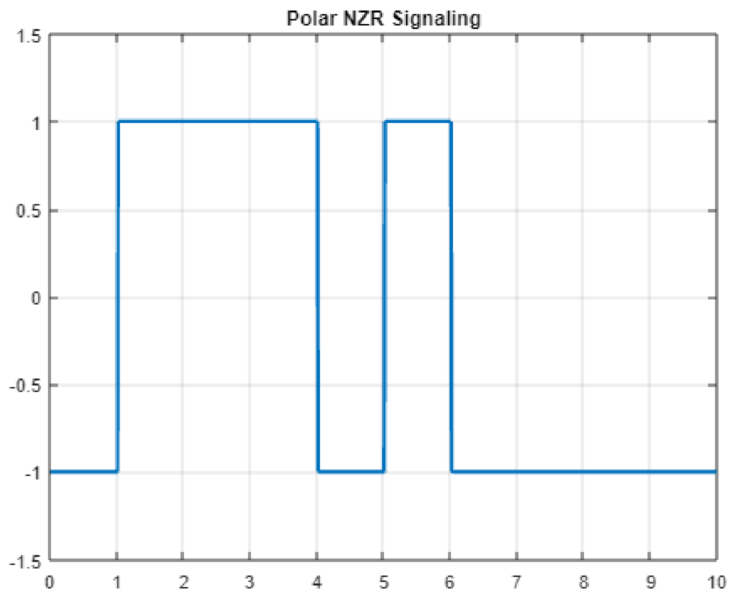


MATLAB Code:

```
%Polar Non Return to Zero Line Coding

clc;
clear all;
close all;
N=10; %Number of bits
n=randi([0,1],1,N)
%Mapping Function
for m=1:N
    if n(m)==1
        nn(m)=1;
    else
        nn(m)=-1;
    end
end
nn
%Signal Shaping
i=1;
t=0:0.01:length(n);
for j=1:length(t)
    if t(j)<=i
        y(j)=nn(i);
    else
        y(j)=nn(i);
        i=i+1;
    end
end
plot(t,y, 'linewidth',2);
axis([0,N,-1.5,1.5]); grid on;
title("Polar NZR Signaling");
```

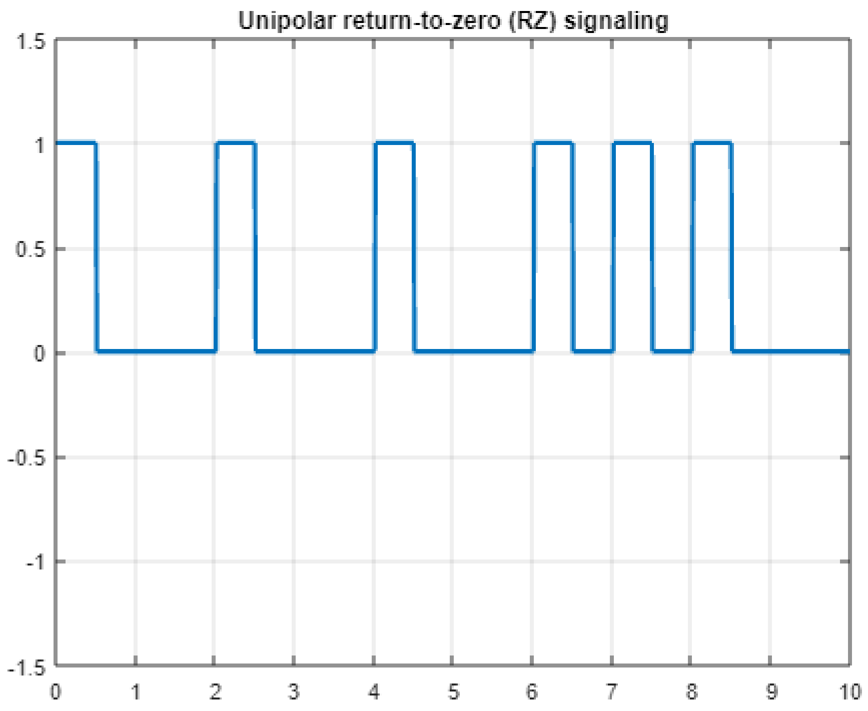
Output:



MATLAB Code:

```
%RZ Unipolar line coding
clc;
clear all;
close all;
N=10;
n=randi([0,1],1,N)
%RZ Pulse Shaping
i=1;
a=0;
b=0.5;
t=0:0.01:length(n);
for j=1:length(t)
    if t(j)>=a && t(j)<=b
        y(j)=n(i);
    elseif t(j)>b && t(j)<=i
        y(j)=0;
    else
        i=i+1;
        a=a+1;
        b=b+1;
    end
end
plot(t,y,'lineWidth', 2); %Linewidth 2 for clear visualization
axis([0,N,-1.5,1.5]); grid on;
title('Unipolar return-to-zero (RZ) signaling');
```

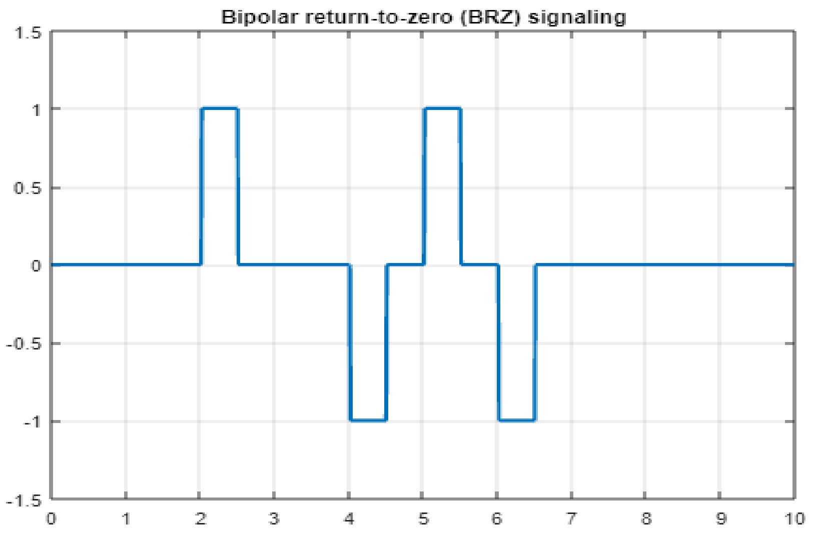
Output:



MATLAB Code:

```
N=10;
n=randi([0,1],1,N)
%Binary to Bipolar Conversion
f=1;
for m=1:N
    if n(m)==1
        if f==1
            nn(m)=1;
            f=-1;
        else
            nn(m)=-1;
            f=1;
        end
    else
        nn(m)=0;
    end
end
nn
%Bipolar RZ Pulse Shaping
i=1; a=0; b=0.5;
t=0:0.01:length(n);
for j=1:length(t)
    if t(j)>=a && t(j)<=b
        y(j)=nn(i);
    elseif t(j)>b && t(j)<=i
        y(j)=0;
    else
        i=i+1;
        a=a+1;
        b=b+1;
    end
end
plot(t,y,'lineWidth', 2); axis([0,N,-1.5,1.5]);grid on;
title('Bipolar return-to-zero (BRZ) signaling');
```

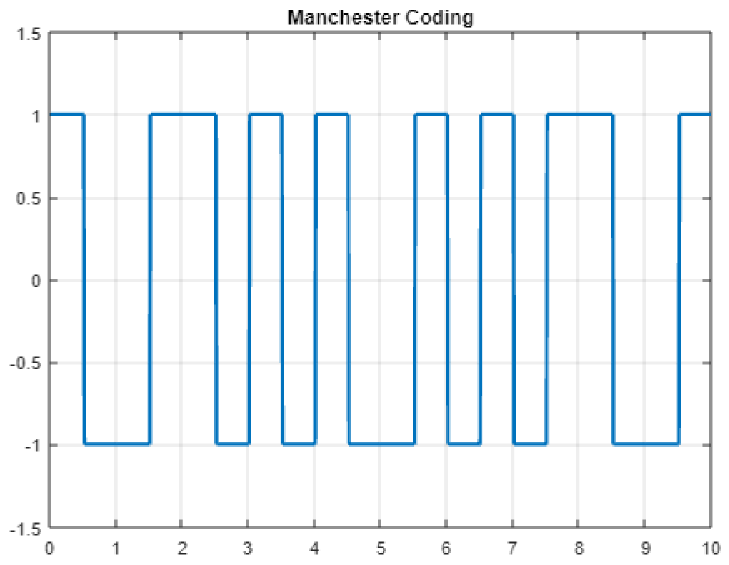
Output:



MATLAB Code:

```
%Split Phase-Manchester Coding
clc;
clear all;
close all;
N=10;
n=randi([0,1],1,N)
%Binary to Manchester Conversion
nnn=[];
for m=1:N
    if n(m)==1
        nn=[1 -1];
    else
        nn=[-1 1];
    end
    nnn=[nnn nn];
end
nnn
%Manchester Coding Pulse Shaping
i=1;
l=0.5;
t=0:0.01:length(n);
for j=1:length(t)
    if t(j)<=l
        y(j)=nnn(i);
    else
        y(j)=nnn(i);
        i=i+1;
        l=l+0.5;
    end
end
plot(t,y,'lineWidth', 2); %Linewidth 2 for clear visualization
axis([0,N,-1.5,1.5]); grid on;
title('Manchester Coding');
```

Output:



MATLAB Code:

```
clc;
clear all;
close all;
x= [1 0 1 0 1 0 1 0 1];
bp=0.000001;
disp('Binary Information at transmitter');
disp(x);
%Representation of Transmitting binary information as digital signal
bit=[];
for n=1:1:length(x)
    if x(n)==1
        se=ones (1,100);
    else
        x(n)==0;
        se=zeros (1,100);
    end
    bit=[bit se];
end

t1=bp/100:bp/100:100*length(x)*(bp/100);
subplot (3,1,1)
plot (t1, bit, 'linewidth', 2.5);
grid on;
axis([0 bp*length(x) -0.5 1.5]);
ylabel('Amplitude (volt)');
xlabel('Time (sec)');
title('Transmitting Information as Digital Signal');
%Binary ASK Modulation
A1=10;
A2=5;
br=1/bp;
f=br*10;
t2=bp/99:bp/99:bp;
ss=length(t2);
m=[];
for(i=1:1:length(x))
    if x(i)==1;
        y=A1*cos(2*pi*f*t2);
    else
        y=A2*cos(2*pi*f*t2);
    end
    m= [m y];
end
t3=bp/99:bp/99:bp*length(x);
subplot (3,1,2)
plot (t3,m);
grid on;
xlabel('Time (sec) ');

ylabel('Amplitude (volt)');
title('Waveform for binary ASK Modulation corresponding binary information');
%Binary ASK Demodulation
mn=[];
```

```
for n=ss:ss:length(m)
    t=bp/99:bp/99:bp;
    y=cos(2*pi*f*t);
    mm=y.*m((n-(ss-1)):n);
    t4=bp/99:bp/99:bp;
    z=trapz(t4, mm);
    zz=round((2*z/bp));
    if (zz>7.5)
        a=1;
    else
        a=0;
    end
    mn=[mn a];
end
disp('Binary Information at Receiver');
disp (mn);
%Represntation of Binary data into Digital signal
bit=[];
for n=1:length (mn)
    if mn(n)==1;
        se=ones(1,100);
    else
        se=zeros(1,100);
    end
    bit=[bit se];
end

t4=bp/100:bp/100:100*length(mn)*(bp/100);
subplot(3,1,3)
plot(t4, bit, 'linewidth', 2.5);
grid on;
axis([0 bp*length(mn) -0.5 1.5]);
xlabel('Time (sec) ');
ylabel('Amplitude (volt)');
title('Received information as Digital Signal');
```

MATLAB Code:

```
clc;
clear all;
close all;

% Binary information to be transmitted
x = [1 1 0 1 0 1];

bp = 0.000001; % Bit period

disp('Binary information at transmitter');
disp(x);

% Representation of transmitting binary information as a digital signal
bit = [];
for n = 1:length(x)
    if x(n) == 1
        se = ones(1, 100);
    else
        se = zeros(1, 100);
    end
    bit = [bit se];
end

tl = bp/100 : bp/100 : 100 * length(x) * (bp/100);

subplot(3, 1, 1);
grid on;
plot(tl, bit, 'linewidth', 2.5);
A = 5;
axis([0 bp * length(x) -0.5 1.5]);
ylabel('Amplitude (volt)');
xlabel('Time (sec)');
title('Transmitting information as a digital signal');

% Binary FSK Modulation
br = 1 / bp;
fl = br * 8;
f2 = br * 2;
t2 = bp / 99 : bp / 99 : bp;
ss = length(t2);
m = [];

for i = 1:length(x)
    if x(i) == 1
        y = A * cos(2 * pi * fl * t2);
    else
        y = A * cos(2 * pi * f2 * t2);
    end
    m = [m y];
end

t3 = bp / 99 : bp / 99 : bp * length(x);
```

```
subplot(3, 1, 2);
plot(t3, m);
xlabel('Time (sec)');
ylabel('Amplitude (volt)');
title('Waveform for binary FSK modulation corresponding to binary information');

% Binary FSK Demodulation
mn = [];
for n = ss : ss : length(m)
    t = bp / 99 : bp / 99 : bp;
    yl = cos(2 * pi * fl * t);
    y2 = cos(2 * pi * f2 * t);
    mn = [mn yl .* m((n - (ss - 1)) : n)];
    mn = [mn y2 .* m((n - (ss - 1)) : n)];
    t4 = bp / 99 : bp / 99 : bp;
    z1 = trapz(t4, mn(1:ss));
    z2 = trapz(t4, mn(ss + 1 : end));
    zz1 = round(2 * z1 / bp);
    zz2 = round(2 * z2 / bp);
    if zz1 > A / 2
        bit = [bit 1];
    elseif zz2 > A / 2
        bit = [bit 0];
    end
    mn = [];
end

disp('Binary information at Receiver');
disp(bit);

% Representation of binary information as a digital signal
bit_received = [];
for n = 1:length(bit)
    if bit(n) == 1
        se = ones(1, 100);
    else
        se = zeros(1, 100);
    end
    bit_received = [bit_received se];
end

t4 = bp / 100 : bp / 100 : 100 * length(bit) * (bp / 100);

subplot(3, 1, 3);
plot(t4, bit_received, 'Linewidth', 2.5);
grid on;
axis([0 bp * length(bit) -0.5 1.5]);
ylabel('Amplitude (volt)');
xlabel('Time (sec)');
title('Received information as a digital signal after binary FSK demodulation');
```

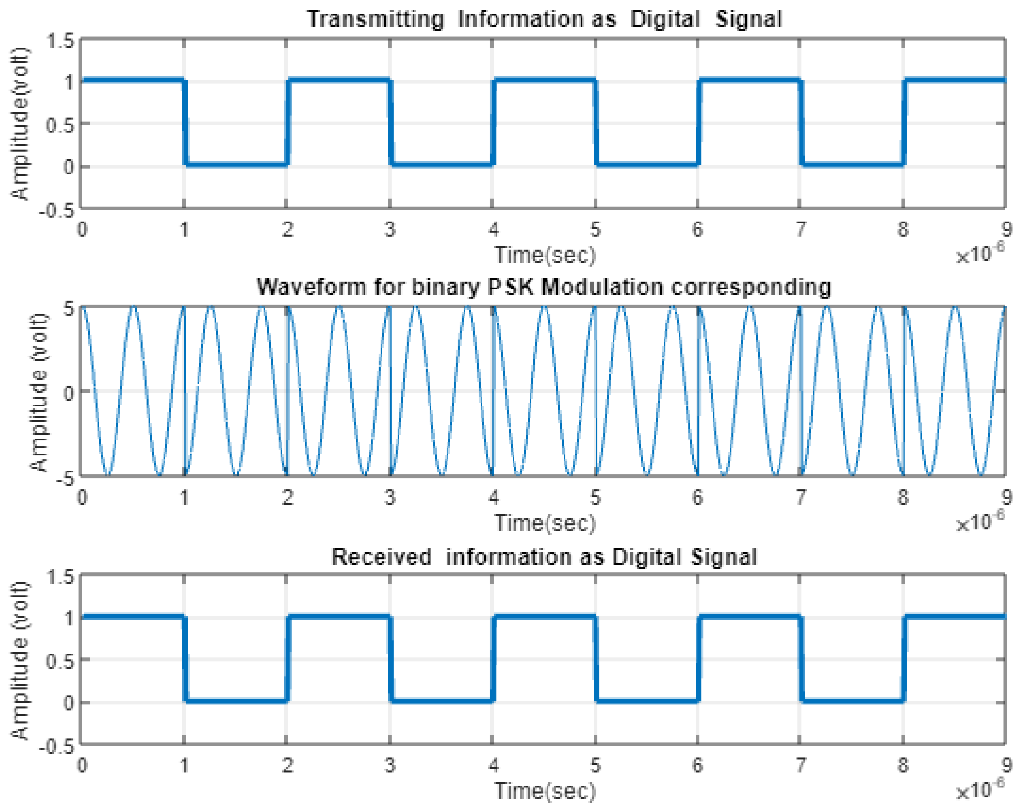
MATLAB Code:

```
clc; clear all; close all;
x=[1 0 1 0 1 0 1 0  1];
bp=0.000001;
disp('Binary  Information at transmitter');
disp(x);
%Representation of Transmitting binary information as digital signal
bit=[];
for  n=1:1:length(x)
    if  x(n)==1;
        se=ones(1,100);
    else
        x(n)==0;
        se=zeros(1,100);
    end
    bit=[bit  se];
end
tl=bp/100:bp/100:100*length(x)*(bp/100);
subplot(3,1,1)
plot(tl,bit,'linewidth',2.5); grid  on;
axis([0  bp*length(x) -0.51.5]);
ylabel('Amplitude(volt)');
xlabel('Time(sec)');
title('Transmitting  Information as  Digital  Signal');
%Binary  PSK Modulation
A=5;
br=1/bp;
f=br*2;
t2=bp/99:bp/99:bp;
ss=length(t2);
m=[];
for(i=1:1:length(x))
    if  x(i)==1;
        y=A*cos(2*pi*f*t2);
    else

        y=A*cos(2*pi*f*t2+pi);
    end
    m=[m y];
end

t3=bp/99:bp/99:bp*length(x) ;
subplot(3,1,2);plot(t3,m);grid  on;
xlabel('Time(sec)');
ylabel('Amplitude (volt)');
title('Waveform for binary PSK Modulation corresponding');
%Binary  FSK  Demodulation
mn=[];
for  n=ss:ss:length(m)
    t=bp/99:bp/99:bp;
    y=cos(2*pi*f*t);      %Carrier  signal
    mm=y.*m((n-(ss-1)):n);
    t4=bp/99:bp/99:bp;
    z=trapz(t4,mm);       %Intregation
    zz=round(2*z/bp);
    if(zz>0)
        a=1;
    else
        a=0;
    end
    mn=[mn a];
end
disp('Binary  Information at Receiver After PSK Demodulation');
disp(mn);
%Represntation of Binary data into Digital signal
bit=[];
for n=1:length(mn)
    if mn(n)==1;
        se=ones(1,100);
    else
        se=zeros(1,100);
    end
    bit=[bit  se];
end
t4=bp/100:bp/100:100*length(mn)*(bp/100);
subplot(3,1,3)
plot(t4,bit,'linewidth',2.5); grid on;
axis([0 bp*length(mn) -0.5 1.5]);
xlabel('Time(sec)') ;
ylabel('Amplitude  (volt)');
title('Received  information as Digital Signal');
```

<u>Output:</u>



MATLAB Code:

```
%QPSK waveform generation
clc;clear all; close all;
%x=[0 1 0 1];%input bits
x=randi([0,1],1,10)
%Bits to polar
for i=1:length(x)
    if x(i)==0
        p(i)=-1;
    else
        p(i)=1;
    end
end
%separation of even and odd sequence
even_seq= p(1:2:length(x));
odd_seq=p(2:2:length(x));

%NRZ polar line coder signal generation
i=1;
t=0:0.01:length(x);
m=2:2:length(x);
for j=1:length(t)
    if t(j)<=m(i)
        even_ps(j)=even_seq(i);
    else
        even_ps(j)=even_seq(i);
        i=i+1;
    end
end
i=1;
m=2:2:length(x);
for j=1:length(t)
    if t(j)<=m(i)
        odd_ps(j)=odd_seq(i);
    else
        odd_ps(j)=odd_seq(i);
        i=i+1;
    end
end
figure(1)
subplot(211);
plot(t,even_ps,'r');
subplot(212);
plot(t,odd_ps,'r');


%carrier signals generation
c1=cos(2*pi*1*t);
c2=sin(2*pi*1*t);

figure(2)
subplot(211);
plot(t,c1,'r');
subplot(212)
```

```
plot(t,c2,'b');

%QPSK waveform generation
r1=even_ps.*c1;
r2=odd_ps.*c2;
qpsk_sig=r1-r2;

figure(3)
subplot(311)
plot(t,r1,'r');
subplot(312)
plot(t,r2,'b');
subplot(313);
plot(t,qpsk_sig,'b');
```

MATLAB Code:

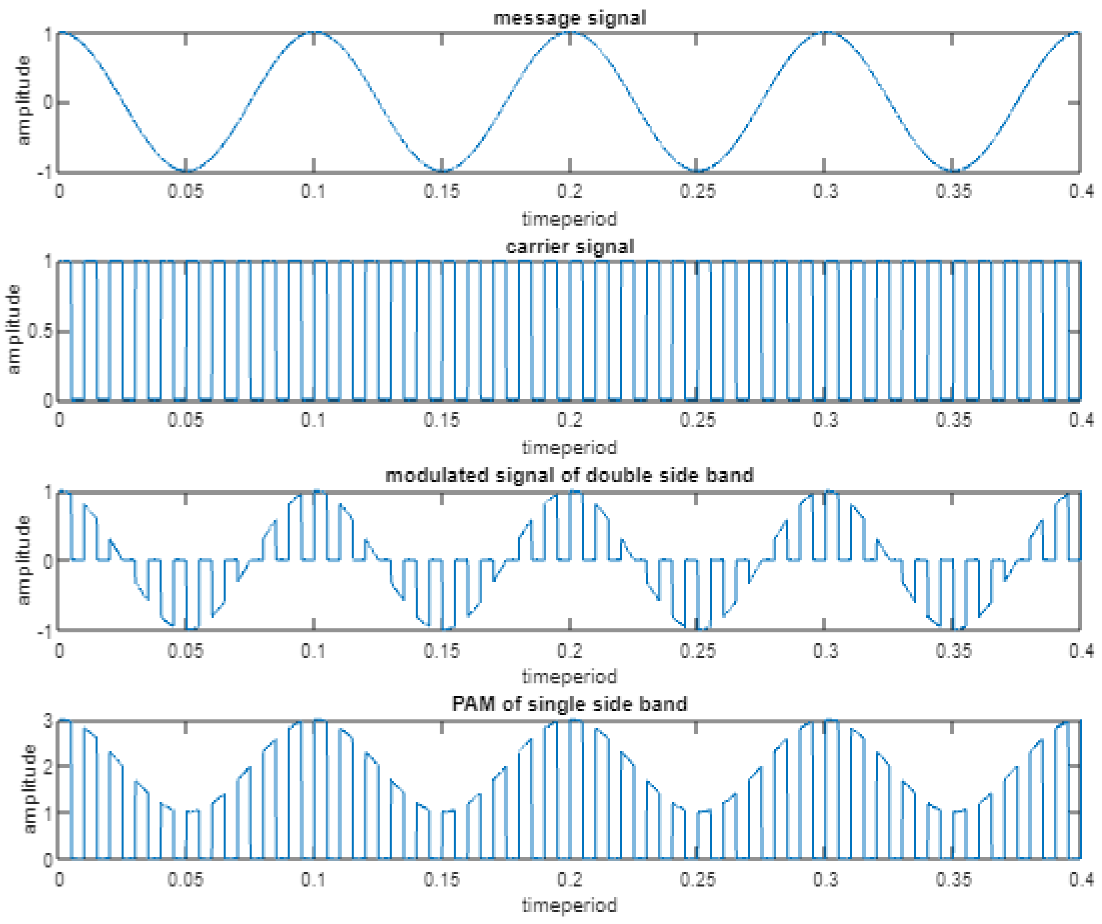
```
clc
close all
clear all
t = 0:0.0001:20; %sampling at niquist rate
c=input('Enter Bit Depth Of PCM Coding:');
part = -1:0.1:1;%A quantization partition defines several contiguous, nonoverlapping
ranges
%of values within the set of real numbers.
codebook = -1:0.1:1.1;%A codebook tells the quantizer which common value to assign to
inputs that
%fall into each range of the partition.
msg = cos(t);
[~,quants] = quantiz(msg,part,codebook);%returns a vector that tells which interval
each input is in
subplot(3,1,1);
plot(t,msg);
title('Message Signal');
subplot(3,1,2);
plot(t,quants);
title('Quantized Signal');
y = uencode(quants,c);
ybin=dec2bin(y,c); %converting it to final binary form to make it transmit ready
subplot(3,1,3);
plot(t,y);
title('PCM PLOT');

%then you can represent the partition as the three-element vector
partition = [0,1,3];
```

MATLAB Code:

```
clc;
clear all;
close all;
fc=100;
fm=fc/10;
fs=100*fc;
t=0:1/fs:4/fm;
mt=cos(2*pi*fm*t);
ct=0.5*square(2*pi*fc*t)+0.5;
st=mt.*ct;
tt=[ ];
%single sided PAM
for i=1:length(st);
    if st(i)==0;
        tt=[tt,st(i)];
    else
        tt=[tt,st(i)+2];
    end
end
figure(1)
subplot(4,1,1);
plot(t,mt);
title('message signal');
xlabel('timeperiod');
ylabel('amplitude');
subplot(4,1,2);
plot(t,ct);
title('carrier signal');
xlabel('timeperiod');
ylabel('amplitude');
subplot(4,1,3);
plot(t,st);
title('modulated signal of double side band');
xlabel('timeperiod');
ylabel('amplitude');
subplot(4,1,4);
plot(t,tt);
title('PAM of single side band');
xlabel('timeperiod');
ylabel('amplitude');
```

Output:


message signal
amplitude
timeperiod
carrier signal
amplitude
timeperiod
modulated signal of double side band
amplitude
timeperiod
PAM of single side band
amplitude
timeperiod