```
clc;
clear all;
%RZ+I

bit_stream = [1 1 0 0 1 1];
no_bits = length(bit_stream); %no_bits=6
bit_rate = 1000; % 1 kbps
pulse_per_bit = 2; % for differential manchester
pulse_duration = 1/((pulse_per_bit)*(bit_rate));
%pulse_duration=1/2*1000=1/2000=0.0005=0.5ms
no_pulses = no_bits*pulse_per_bit; % no_pulse=6*2=12

samples_per_pulse = 500;
fs = (samples_per_pulse)/(pulse_duration); %sampling frequency
% including pulse duration in sampling frequency
% ensures having enough samples in each pulse
t = 0:1/fs:(no_pulses)*(pulse_duration); % sampling interval
% total duration = (no_pulse)*(pulse_duration)
no_samples = length(t); % total number of samples
dig_sig = zeros(1,no_samples);

max_voltage = +2;
min_voltage = -2;
inv_bit = 0; % inverting bit (in the lab sheet it is 1, but it should be 0)
last_state = max_voltage; %+v
inv_last_state = min_voltage; % inverse of last state, -v

for i = 1:no_bits
    j = (i-1)*2;
    if bit_stream(i) == inv_bit
        dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
inv_last_state*ones(1,samples_per_pulse);
        dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
last_state*ones(1,samples_per_pulse);
    else
        dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
last_state*ones(1,samples_per_pulse);
        dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
inv_last_state*ones(1,samples_per_pulse);
        temp_cons = last_state; % temporary constant
        last_state = inv_last_state;
        inv_last_state = temp_cons;
    end
end

figure
plot(t,dig_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('Voltage')
ylim([(min_voltage - (max_voltage)*0.2) (max_voltage+max_voltage*0.2)])
title(['Differential Manchester for ',num2str(bit_stream),', last state =
',num2str(last_state),', inverting bit is ',num2str(inv_bit),''])
```

```
clc;
clear all;

bit_stream = [1 0 0 0 1 1 0 1];
no_bits = length(bit_stream); %no_bit=8
bit_rate = 1000; % 1 kbps
pulse_per_bit = 1; % for unipolar nrz
pulse_duration = 1/((pulse_per_bit)*(bit_rate));
%pulse_duration=1/1*1000=1x10-3 (second)=1 ms
no_pulses = no_bits*pulse_per_bit; %no_pulses=8*1=8


samples_per_pulse = 500;
fs = (samples_per_pulse)/(pulse_duration); %sampling frequency
t = 0:1/fs:(no_pulses)*(pulse_duration); % sampling interval, total
duration = (no_pulse)*(pulse_duration)
no_samples = length(t); % no_samples=4001
dig_sig = zeros(1,no_samples); % intial digital signal (line coded
signal according to unipolor nrz)
max_voltage = 5;
min_voltage = 0;

for i = 1:8
    if bit_stream(i) == 1
        dig_sig(((i-1)*(samples_per_pulse)+1):i*(samples_per_pulse)) =
max_voltage*ones(1,samples_per_pulse);
    else
        dig_sig(((i-1)*(samples_per_pulse)+1):i*(samples_per_pulse)) =
min_voltage*ones(1,samples_per_pulse);
    end
end

plot(t,dig_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('Voltage')
ylim([(min_voltage - (max_voltage)*0.2)
(max_voltage+max_voltage*0.2)])
title(['Unipolar NRZ for ',num2str(bit_stream),''])
```

```
clc;
clear all;

bit_stream = [1 0 1 0 1 1 0 1];
no_bits = length(bit_stream); % no_bits=8
bit_rate = 1000; % 1 kbps
pulse_per_bit = 2; % for unipolar rz
pulse_duration = 1/((pulse_per_bit)*(bit_rate));
%pulse_duration=1/(2*1000)=1/2000=0.0005=0.5ms
no_pulses = no_bits*pulse_per_bit; %no_pulses=8*2=16

samples_per_pulse = 500;
fs = (samples_per_pulse)/(pulse_duration); %sampling frequency
% including pulse duration in sampling frequency
% ensures having enough samples in each pulse
t = 0:1/fs:(no_pulses)*(pulse_duration); % sampling interval
% total duration = (no_pulse)*(pulse_duration)
no_samples = length(t); % total number of samples

dig_sig = zeros(1,no_samples);
max_voltage = 5;
min_voltage = 0;

for i = 1:no_bits
    j = (i-1)*2;
    if bit_stream(i) == 1
        dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
max_voltage*ones(1,samples_per_pulse);

dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
zeros(1,samples_per_pulse);
    else
        dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
min_voltage*ones(1,samples_per_pulse);

dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
zeros(1,samples_per_pulse);
    end
end

plot(t,dig_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('Voltage')
ylim([(min_voltage - (max_voltage)*0.2)
(max_voltage+max_voltage*0.2)])
title(['Unipolar RZ for ',num2str(bit_stream),''])
```