****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Semester: (Spring, Year:2024), B.Sc. in CSE (Day)**

**Lab Report NO #05**

**Course Title: Data Communication Lab**

**Course Code: CSE 308 Section: 221 D20**

**Lab Experiment Name:** Implementation of Encoding and Decoding Scheme Using AMI, Pseudo Ternary, and Manchester.

**Student Details**

| **Name** | | **ID** |
| --- | --- | --- |
| **1.** | Tanvir Ahmed | 221002461 |

**Lab Date : 21/5/2024**

**Submission Date : 30/5/2024**

**Course Teacher’s Name : Md. Romazan Alom**

| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |
| --- |

**Introduction:**

We have the task of implementing encoding and decoding schema using AMI, Pseudo Ternary, Manchester. At first, in the AMI I check if the bit is 1 or not, and then if it is 1 we alternate the signal based on the last bit. Else, we do nothing cause we made the x zero using the zeros function. So, if the bit is zero then it will stay in zero. In the decoding part, I simply check x and y axis and assign corresponding bits to an array. And I use a counter to access every signal. I did similar in the pseudo-ternary, just changed the logic. In Manchester, we have 2 signal elements for 1 bit. For this, we need to divide the x-axis while assigning the values. For example, if we have 1 in bit then we assign 1 in (i-1)\*n+1 to (i-1)\*n+n/2. It is the half of the 1 signal element. And in the other half, we assign -1. The other half is (i-1)*n+n/2 to i\*n.*

Source code:

* AMI:

bits = input("Enter Bit String:", 's');

bits = str2num(bits(:));

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T;

x = zeros(1,length(t));

lastbit=1;

**for** i=1:length(bits)

**if** bits(i)==1

x((i-1)\*n+1:i\*n)=-lastbit;

lastbit=-lastbit;

**end**

**end**

plot(t, x, 'Linewidth', 3);

result = [0];

counter=0;

**for** i=1:length(t)

**if** t(i)>counter

counter=counter+1;

**if** x(i)==1

result(counter)=1;

**elseif** x(i)==-1

result(counter)=1;

**else** result(counter)=0;

**end**

**end**

**end**

disp('AMI, Decoded Binary String:');

disp(result);

* Pseudo Ternary:

bits = input("Enter Bit String:", 's');

bits = str2num(bits(:));

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T;

x = zeros(1,length(t));

lastbit=1;

**for** i=1:length(bits)

**if** bits(i)==0

x((i-1)\*n+1:i\*n)=-lastbit;

lastbit=-lastbit;

**end**

**end**

plot(t, x, 'Linewidth', 3);

result = [0];

counter=0;

**for** i=1:length(t)

**if** t(i)>counter

counter=counter+1;

**if** x(i)==1

result(counter)=0;

**elseif** x(i)==-1

result(counter)=0;

**else** result(counter)=1;

**end**

**end**

**end**

disp('Pseudo ternary, Decoded Binary String:');

disp(result);

* Manchester:

bits = input("Enter Bit String:", 's');

bits = str2num(bits(:));

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T;

x = zeros(1,length(t));

**for** i=1:length(bits)

**if** bits(i)==1

x((i-1)\*n+1:(i-1)\*n+n/2) = 1;

x((i-1)\*n+n/2:i\*n) = -1;

**else**

x((i-1)\*n+1:(i-1)\*n+n/2) = -1;

x((i-1)\*n+n/2:i\*n) = 1;

**end**

**end**

plot(t, x, 'Linewidth', 3);

result = [0];

counter = 0;

**for** i = 1:length(t)

**if** t(i)>counter

counter = counter + 1;

**if** x(i)>0

result(counter) = x(i);

**else** result(counter) = 0;

**end**

**end**

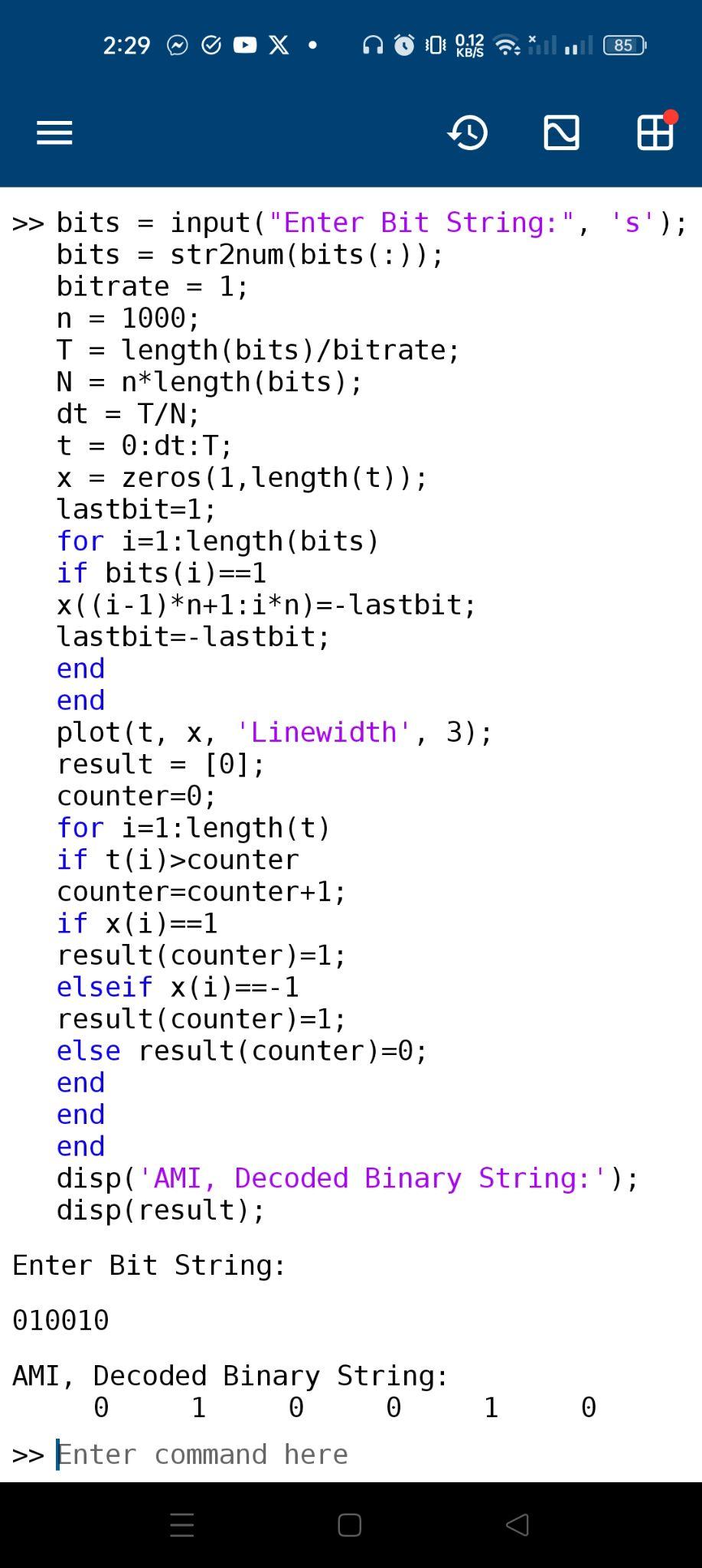
**end**

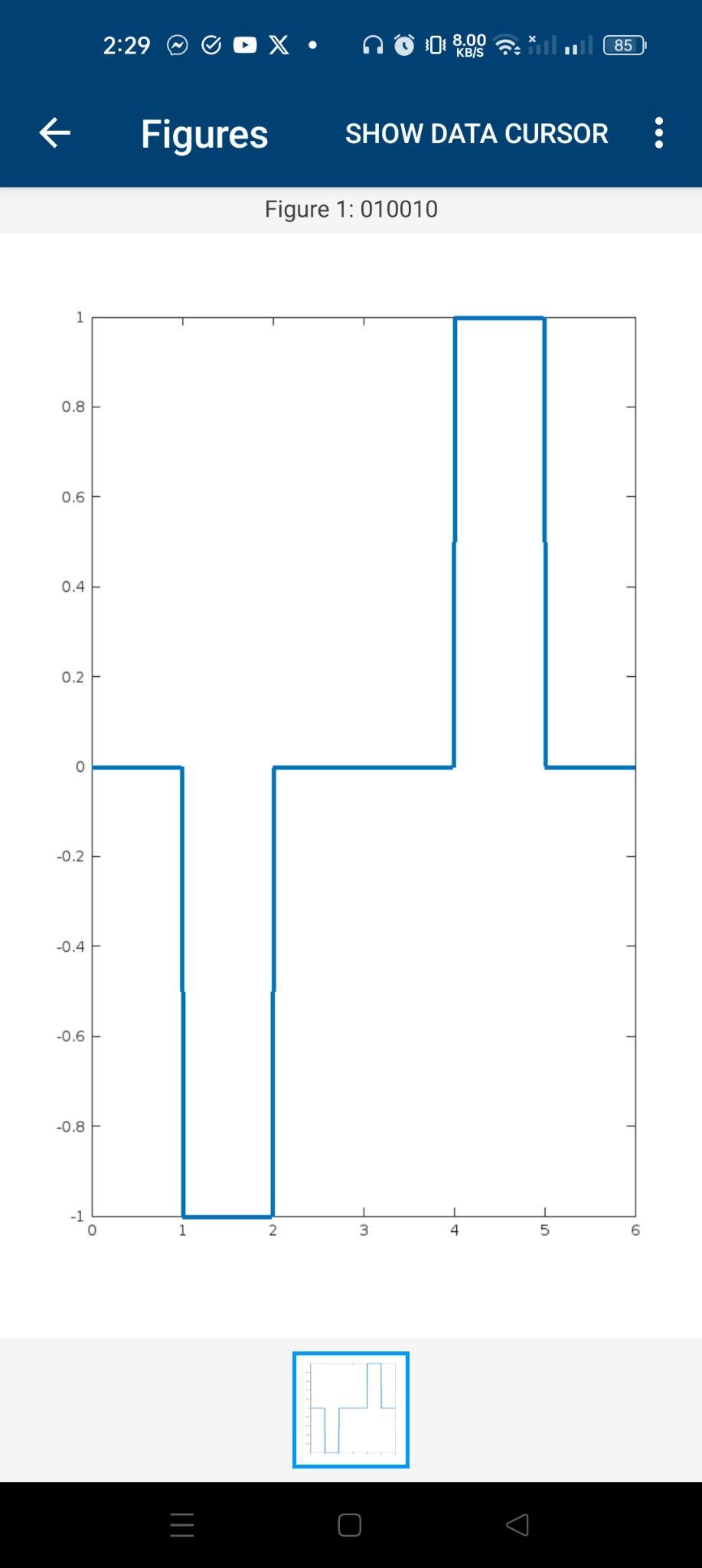
disp('Manchester Decoded Bit String:');

disp(result);

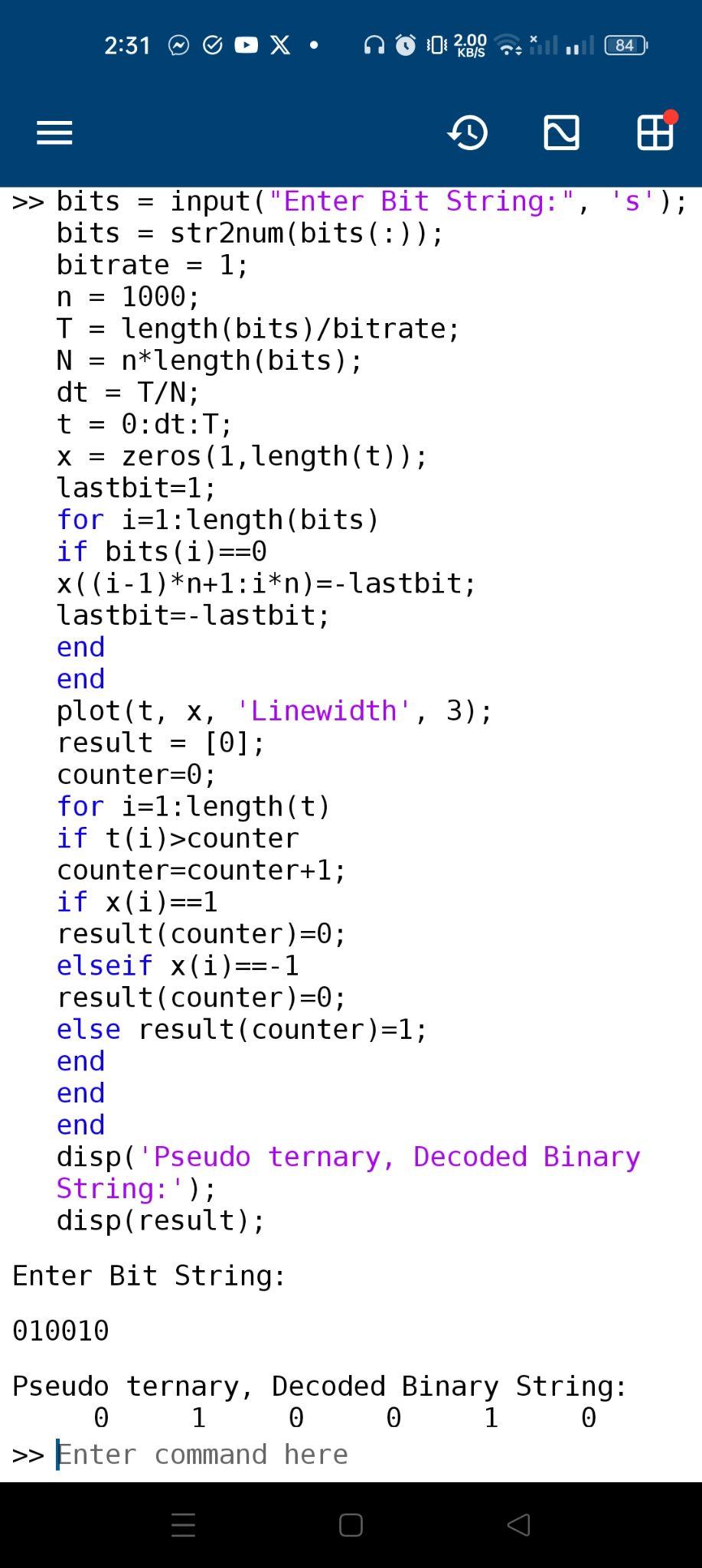
Output:

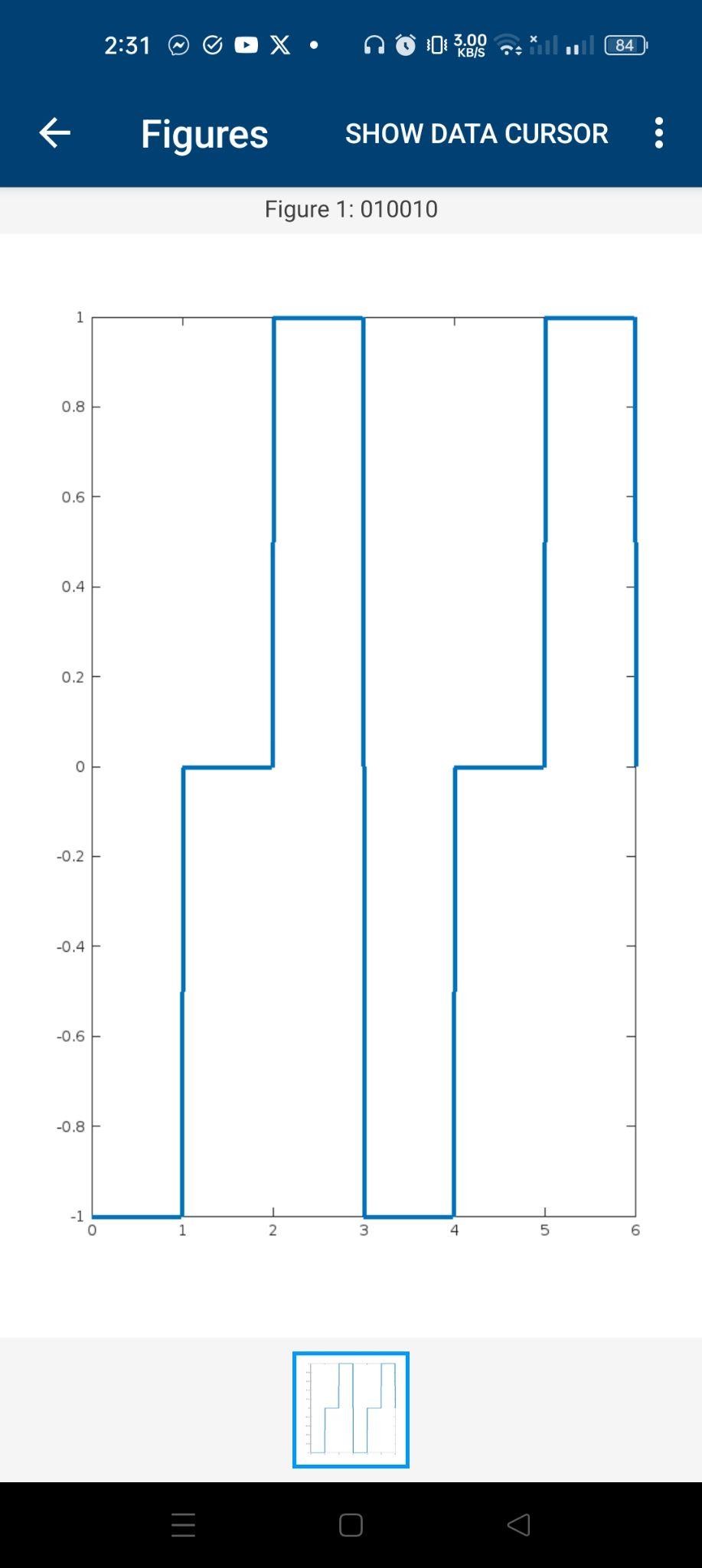
* AMI:



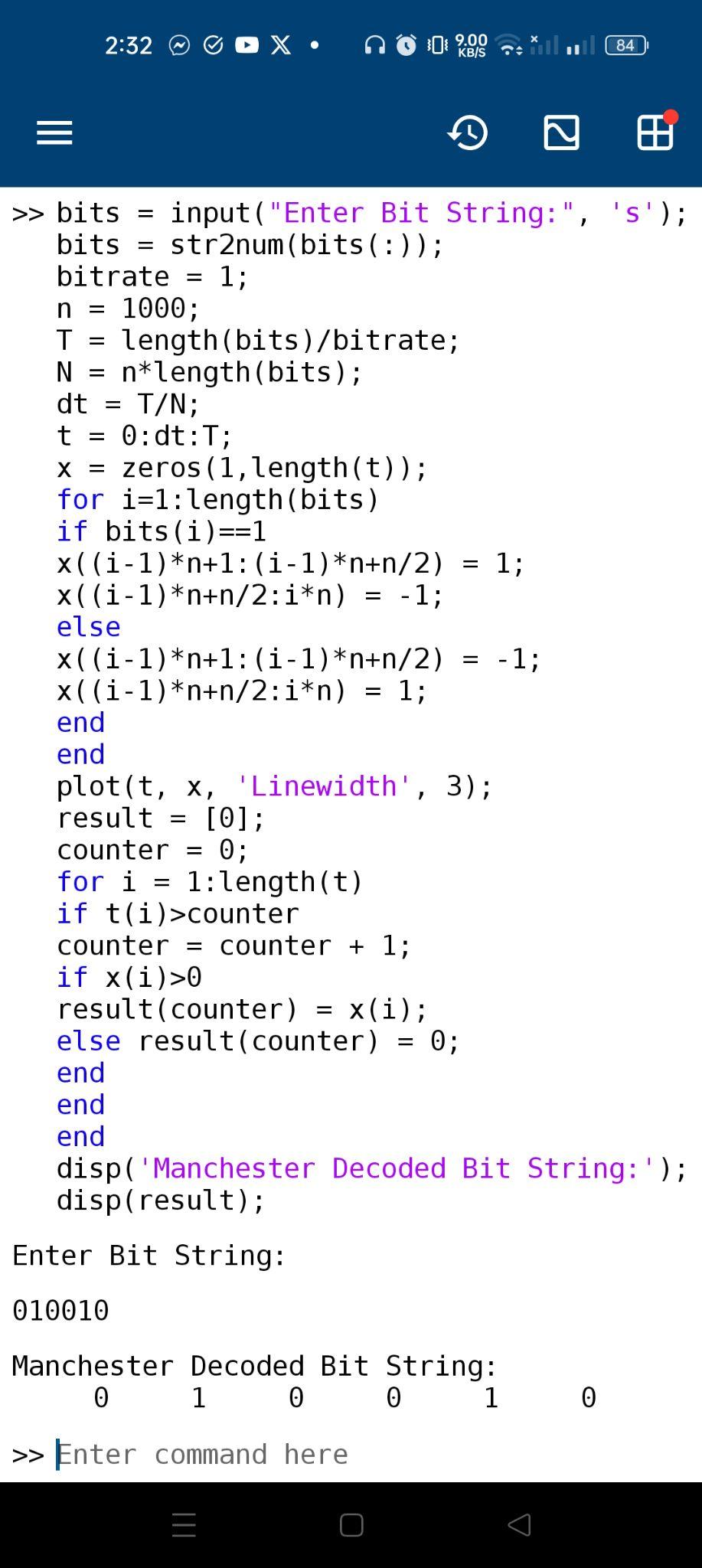


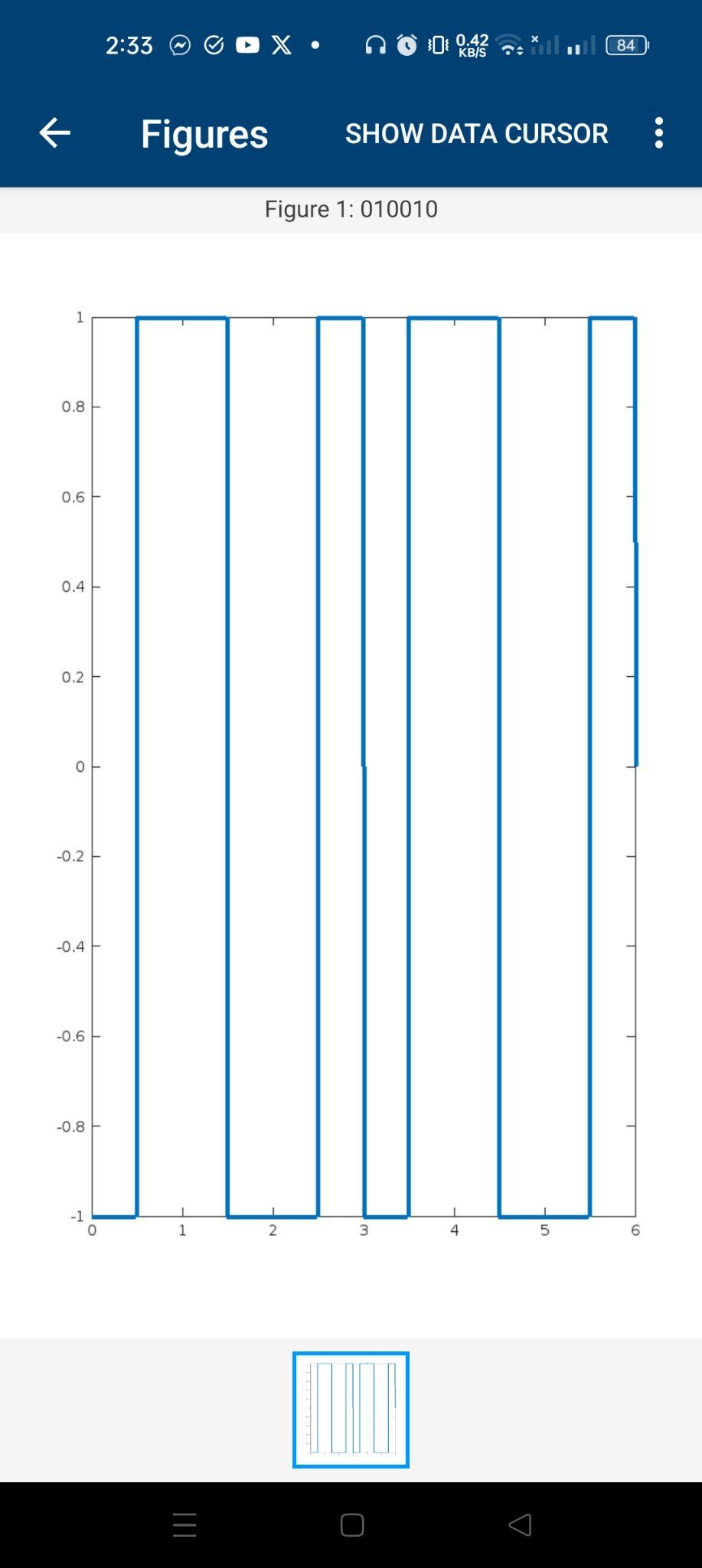
* Pseudo Ternary:





* Manchester:





**Discussion:**

I have completed the task properly, and the output showed correctly. I learned the logic of these schemes in the previous class. I just implemented those codes. I needed help with implementing the Manchester logic from the lab menial. Also, I learned how to implement decoding from the lab manual and I understand them all. I faced difficulty in taking input from users, To overcome this problem I took help from chatgpt. In the Manchester implementation, I faced more difficulties. I took help from the lab manual to overcome the issues. Finally, the output came properly.