



جامعة القاهرة



كلية الهندسة

# Advanced Microprocessor Architecture

## UART & USB Emulator

Represented to

**Dr. Ahmad Khattab**

**&**

**TA. Hassan EL-Menier**

name	sec	B.N
اية رضا عثمان عبداللطيف	1	43
عبدالعزيز محمد جمال الدين	2	51
عبدالله سعيد ربيع	2	53

## **Table of contents:**

<b>1-Introduction.....</b>	<b>2</b>
<b>2-Procedure.....</b>	<b>2</b>
<b>3-Code.....</b>	<b>4</b>
<b>4-Results.....</b>	<b>5</b>
<b>5-References.....</b>	<b>8</b>

## 1- INTRODUCTION

These days, there are a lot of interconnection protocols and buses used to connect between external devices at large scale and even connect internal peripherals inside PCs, or embedded systems at small scale.

In this report we will discuss an important Communication protocol which is UART and one of the most commonly used Serial buses which is USB and we will see how both of them transfer data and we will compare between the transmission time needed for each of them, overhead percentage and efficiency for each of them using MATLAB.

## 2- Procedure

The following flow chart showing how was our code built and showing the behavior of the code in every case

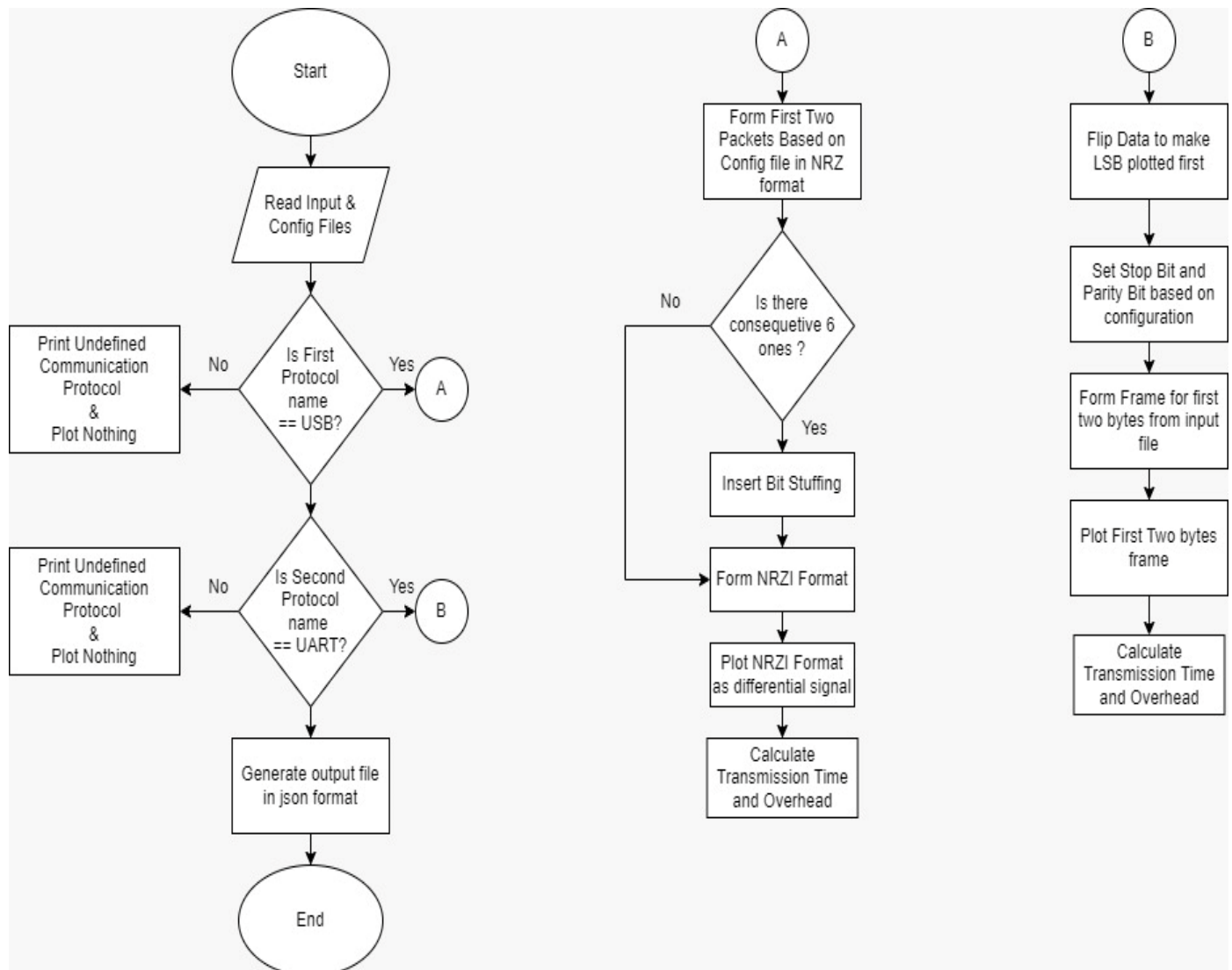


Figure 1: flow chart of the code

This app is made using GUI tool in MATLAB which is "App Designer".

This app is meant to be user friendly and easy to use by putting button "Read Files" which is used to read the input text file and configuration file according to the specifications written in the configuration file it plots the first two bytes transferred by UART and the first two packets transferred by USB. If protocols name in the configuration file were UART and USB otherwise it prints an error message to the user informing to write the right protocol names.

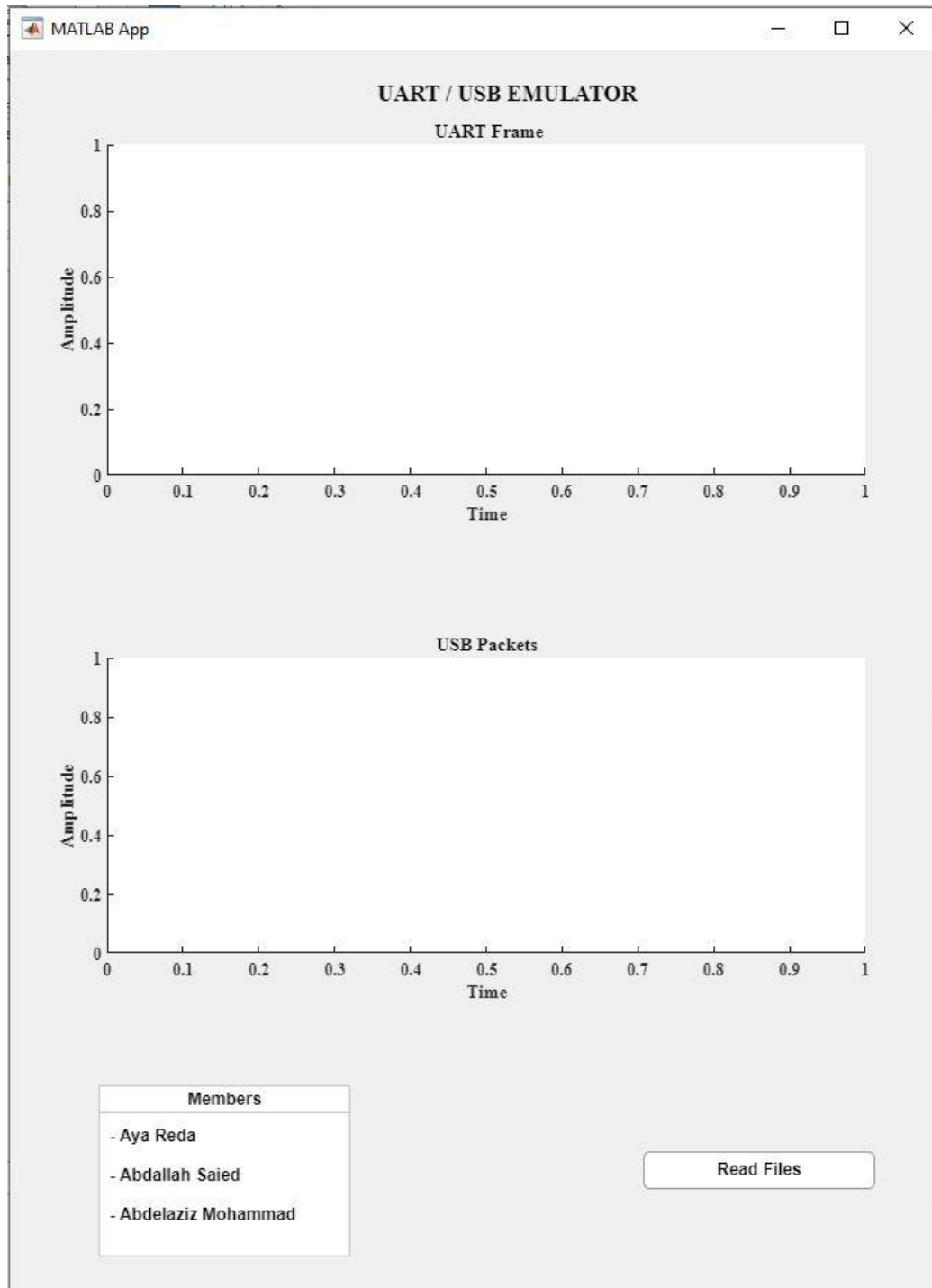


Figure 2: Application UI

### 3- Code

In this part we will show the most important blocks of the code.

```
%Open config.json
json_filename = 'conf.json';
json_config = jsondecode(fileread(json_filename));
% Open input.txt file
filename = 'inputdata.txt';
fileID = fopen(filename);
formatspec = '%c';
Array = fscanf(fileID,formatspec);
fclose(fileID);
```

Figure 3: block of reading the JSON file

```
%Generate output.json
% Convert to JSON text
jsonText2 = jsonencode(OUTPUT,'PrettyPrint',true)
% Write to a json file
fid = fopen('ELC3030_8.json', 'w');
fprintf(fid, '%s', jsonText2);
fclose(fid);
```

Figure 4: block of generating the JSON file

```
%Draw USB Packets
stairs(app.UIAxes2,[USB.output,USB.output(end)]+1.25);
hold(app.UIAxes2,'on');
stairs(app.UIAxes2,[USB.outputdiff,USB.outputdiff(end)]);
hold(app.UIAxes2,'off');
xlim(app.UIAxes2,[1,100]);

%Plot Transmition time & Overhead
tiledlayout(2,2);
nexttile;
plot(file_size,OUTPUT_USB_PLOT_TotalTxTime);
title('USB Transmit time');
xlabel('File Size');
nexttile;
plot(file_size,OUTPUT_USB_PLOT_OVERHEAD);
title('USB Overhead');
xlabel('File Size');
ylabel('Overhead');
```

Figure 5: block of USP graphs

```
%Draw UART Frame
stairs (app.UIAxes,[frame,frame(end)]);

% Plot Requirments
nexttile;
plot(file_size,OUTPUT_UART_PLOT_TotalTxTime);
title('UART Transmit time');
xlabel('File Size');
nexttile;
plot(file_size,OUTPUT_UART_PLOT_OVERHEAD);
title('UART Overhead');
xlabel('File Size');
ylabel('Overhead');
```

Figure 6: block of UART graphs

## 4- Results

### Case (1):

In case protocol names in the configuration file weren't UART and USB the output on the app interface will be as follows and no plots will be shown for overhead and total transmission time.

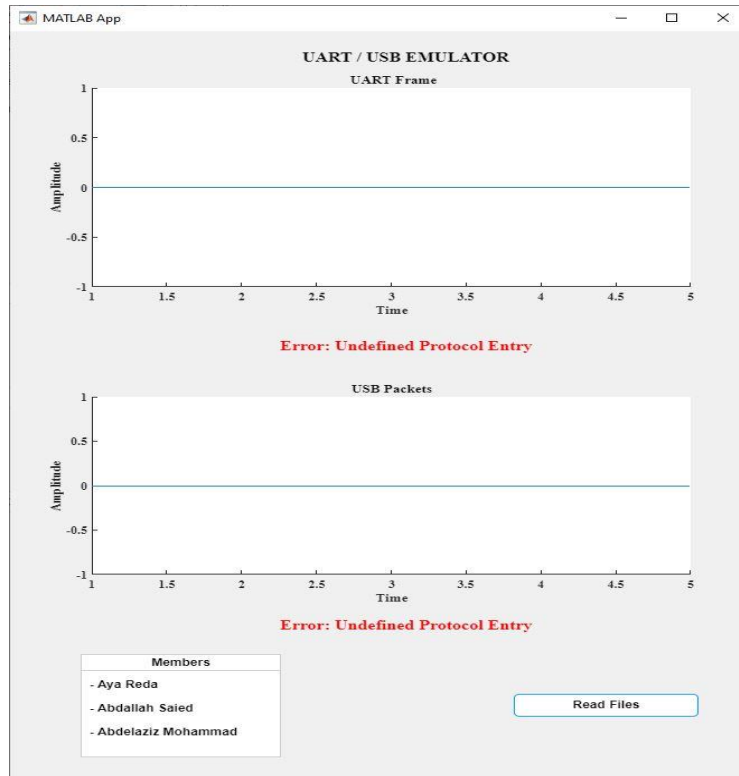


Figure 7: output of the APP in case 1

### Case (2):

Only one of the protocol names in the configuration file is correct and then the plotting of the correct protocol will appear and an error message will appear for the other protocol.

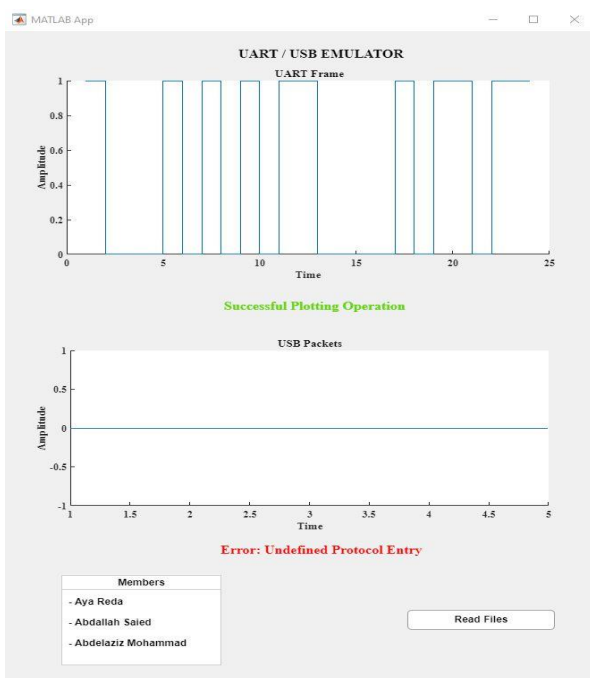


Figure 8: if UART is the only correct

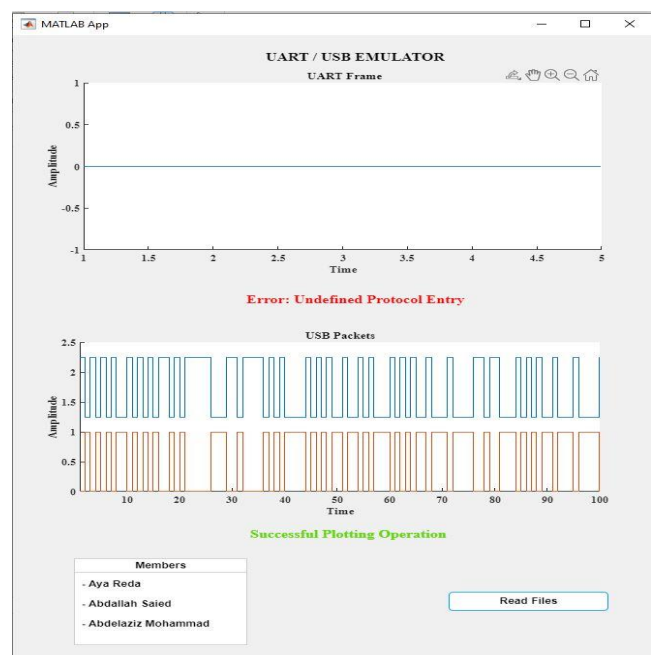


Figure 9: if USB is the only correct

### Case (3):

In case protocol names in the configuration file were UART and USB the output on the app interface will be frame contains two bytes transferred by UART and the other frame contains the two packets transferred by USB, And Overhead and Total Transmission Time Plots will be shown for both protocols as follows:

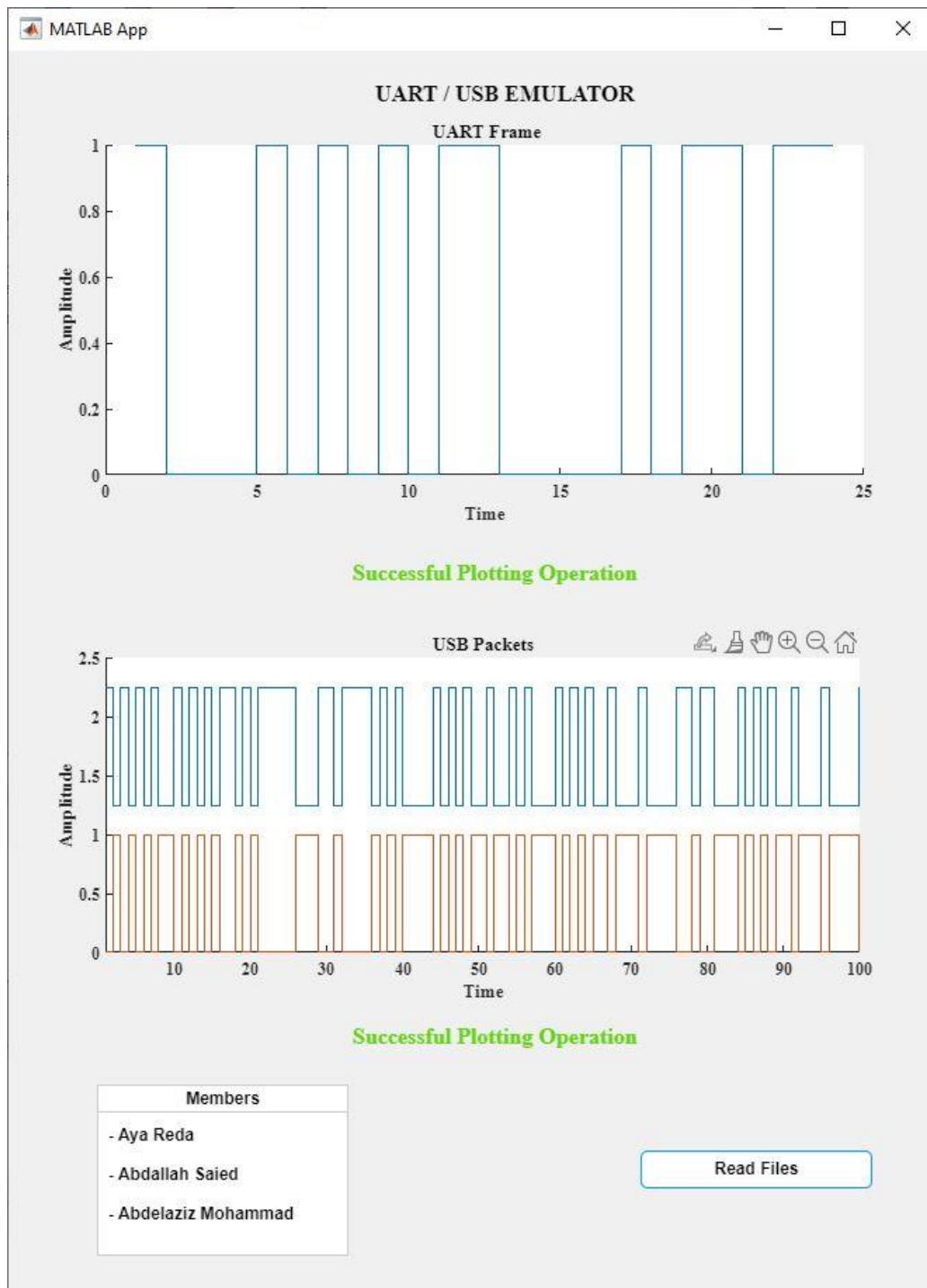


Figure 10: if both protocol are correct



The output file contains the total transmission time, efficiency percentage and overhead percentage for both UART and USB for bit duration = 0.0001 and file size = 1280 bytes is shown in figure 7.

Values in this file are dependent on the configuration in “conf.json” file so they change with the change of configuration file.

Transmission Time in both UART and USB increases linearly with file size with taking in consider that it will be larger in USB than UART

```

1  [
2  {
3    "protocol_name": "UART",
4    "outputs": {
5      "total_tx_time": 1.28,
6      "overhead": 20,
7      "efficiency": 80
8    }
9  },
10 {
11   "protocol_name": "USB",
12   "outputs": {
13     "total_tx_time": 1.054,
14     "overhead": 2.8462998102466792,
15     "efficiency": 97.153700189753323
16   }
17 }
18 ]

```

Figure 11: output JSON file

Overhead in UART is constant because whatever the size is, the equation values would be the same every time.

Overhead in USB is not constant because if the file size isn't divisible by Payload there would be a packet which doesn't contain the maximum payload so the overhead would be change

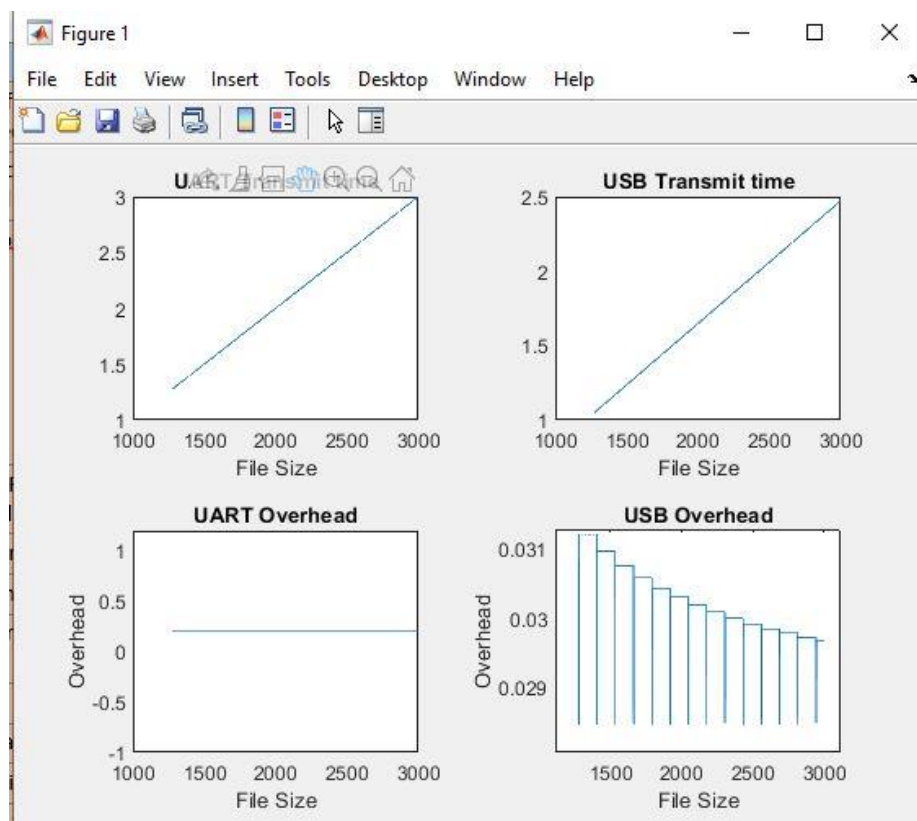


Figure 12: Overhead & Transmit time for both protocol



## 5- References

[1] Lecture Notes.

[2] Tutorial Notes.

[3] Matlab Documentations. [Online], Available: [mathworks.com](https://www.mathworks.com)

[4] UsefulLink.[online], Available: <https://beyondlogic.org/usbnutshell/usb3.shtml>