23rd August, 2019

Interfacing Adxl345 with Arduino UNO over I2C

A Realtime Vibration time-history and spectrum monitoring and Logging System.

Meraj Ali Shah
DEPARTMENT OF MECHANCAL ENGINEERING, NEDUET KARACHI.
Sumaika Khan
DEPARTMENT OF MECHANCAL ENGINEERING, NEDUET KARACHI.

Contents

Hookup guide

Required components are

1- Arduino UNO Microcontroller.
2- Adxl345 Accelerometer.
3- Jumper wires.

Make connections as shown in Figure-1

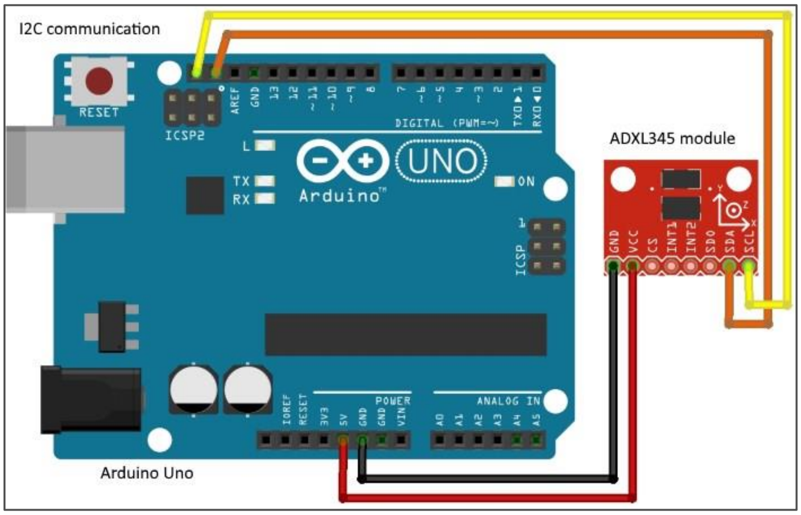


Figure 1 - Hookup diagram

Setting up the Arduino IDE

This section helps you to install Arduino IDE and required Adxl345 libraries.

Follow these steps in sequence to setup Arduino IDE

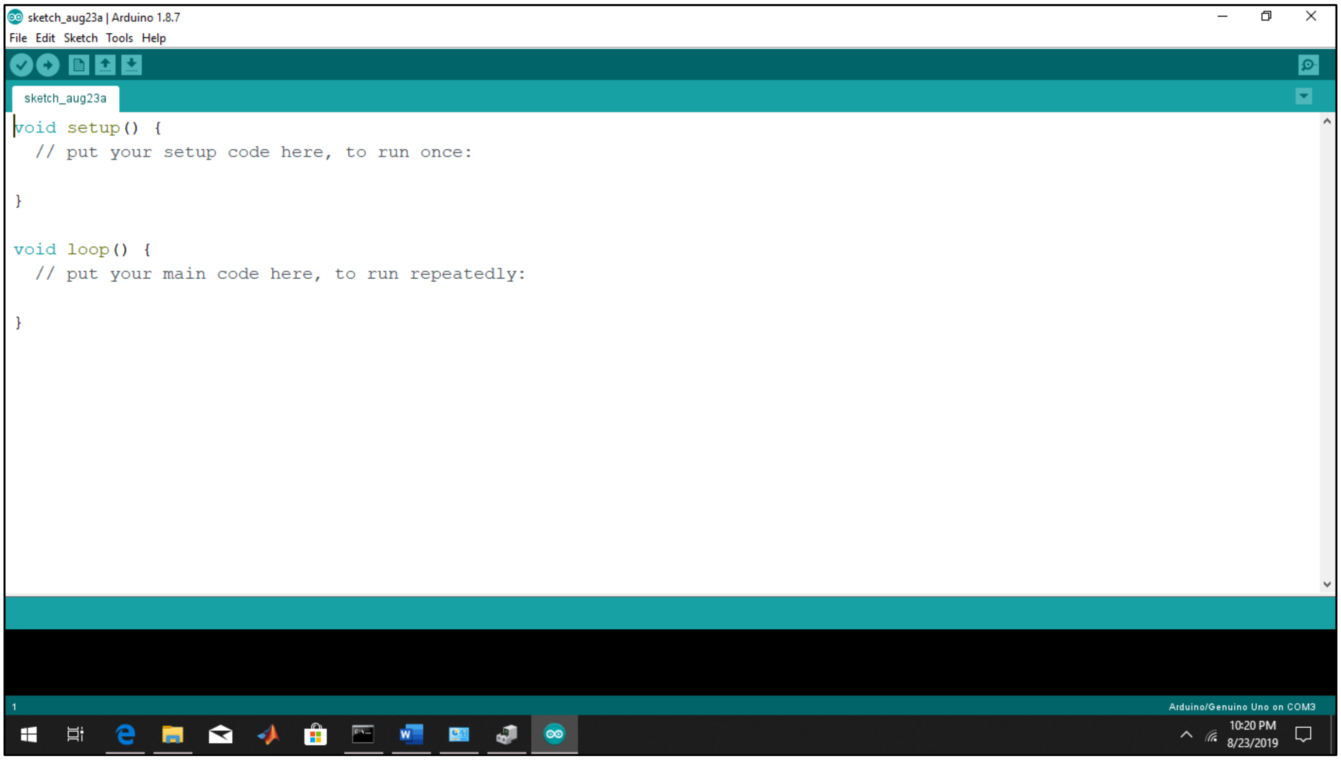
1. Install The open-source Arduino Software (IDE), compatible with your Operating System.

I. Go to Arduino IDE download page

II. Download the software that is compatible with your Operating System i.e. Windows (⊞), Linux (🐧) or MacOS X (OS X).

III. Install the software using the downloaded setup.

IV. After successful installation run the Arduino IDE, a window similar to figure-2 will appear showing a blank Arduino sketch.


sketch_aug23a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_aug23a
1
Arduino/Genuino Uno on COM3
10:20 PM
8/23/2019

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 2 Screenshot showing Arduino IDE running on Windows 10

2. Download the .zip package from the source https://github.com/merajalishah/adxl345vibration and extract it to any suitable location in your local disc. The extracted folder will have a name "adxl345vibration"

3. Install required libraries for adxl345.

I. From the Menu bar of Arduino IDE, go to *Sketch > Include library > Add .Zip library...* and locate the two libraries .Zip files (*Adafruit_Sensor-master.zip* and *Adafruit_ADXL345-master.zip*) present in *adxl345vibration/ Libraries* folder. This will add the sensor's library to your Arduino directory.

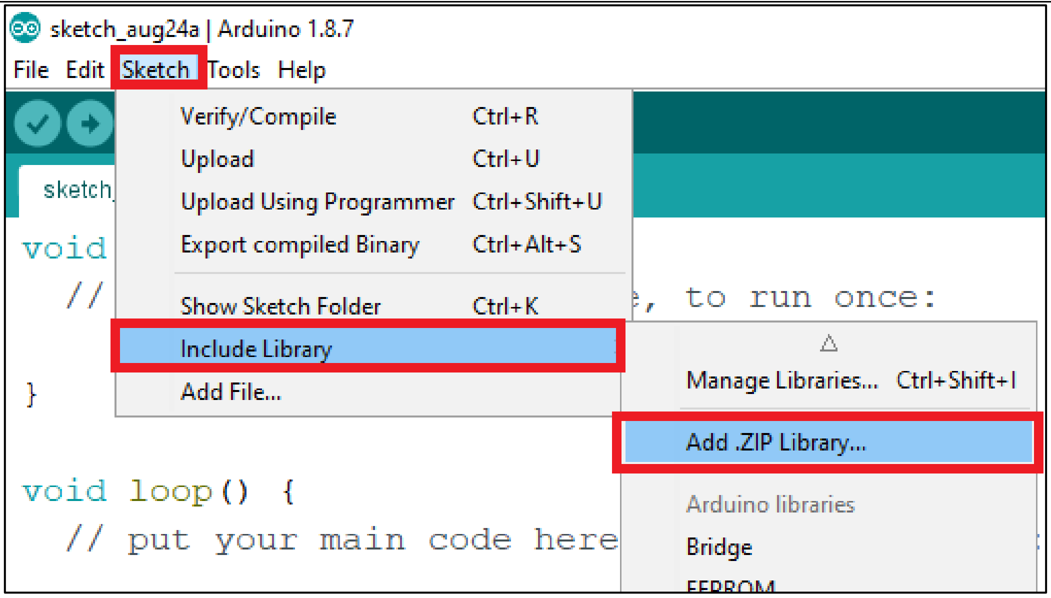

sketch_aug24a | Arduino 1.8.7
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Bridge
EEPROM
sketch
void
//
, to run once:
}
void loop() {
// put your main code here

Figure 3 Screenshot of Arduino main window showing option to add library from zip file

Uploading the sketch using Arduino IDE to your Arduino UNO Microcontroller board

This section helps you to upload the Arduino sketch to your Arduino UNO board.

Follow below steps in sequence

1- From the menu bar of Arduino IDE go to File > Open… and look for the “Arduino Sketches” folder in the directory where the downloaded file was extracted. In the “Arduino Sketches” folder select *Print_Accel_y_LSBs.ino.* This will open the sketch (as shown in figure 4) prepared by the author to print only the raw y-axis acceleration (see labelled y-direction printed on the Adxl345 PCB) for vibration measurement.

2- Connect the Arduino UNO board to your Computer.

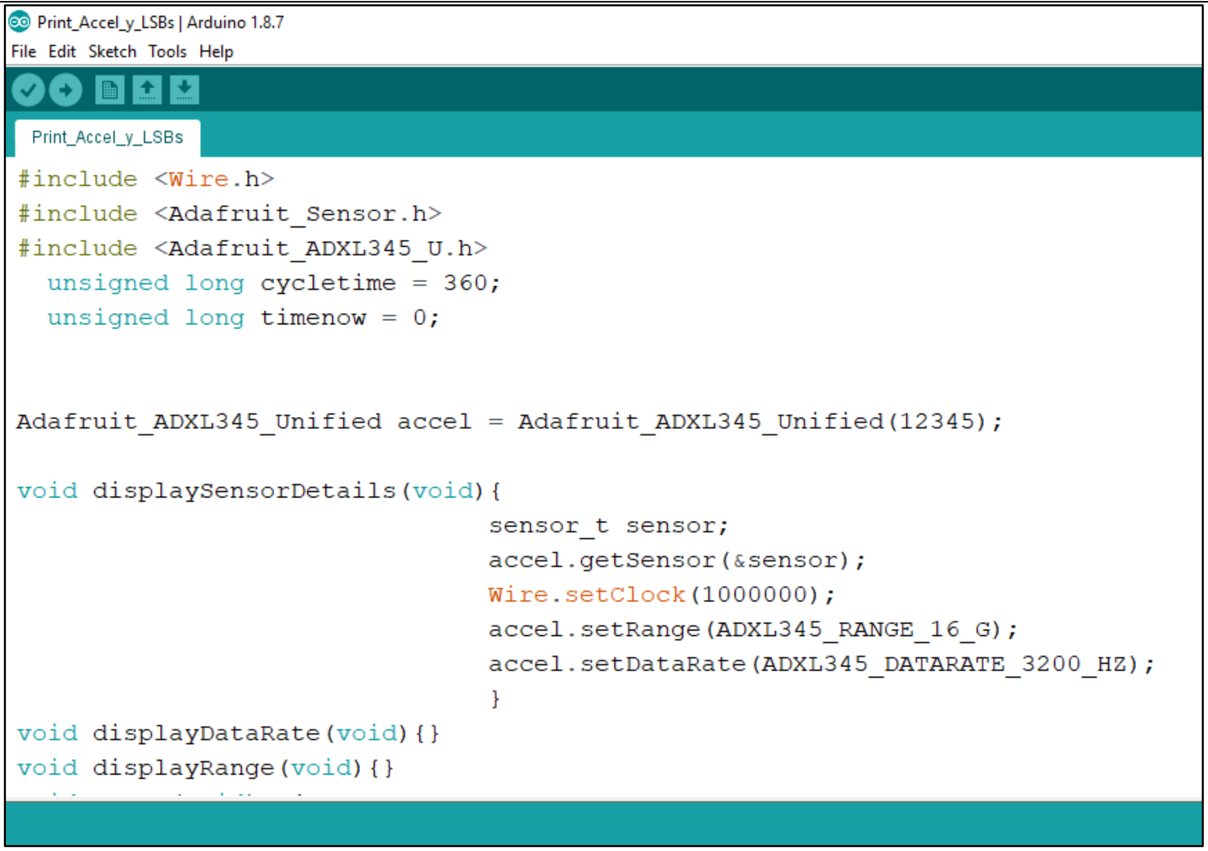
3- From the menu bar go to Tools > Board: and select Arduino/Genuino Uno (select according to the board you are using).

4- From the menu bar go to Tools > Port and select the port to which Arduino Board is attached. (The correct port is the one which disappears by detaching the Arduino board or alternatively you can use device manager on Windows to find the correct port)

5- Now upload the sketch via the upload button present on the top left region of Arduino IDE screen.

6- Once the code is successfully uploaded, open the Serial Monitor (Ctrl+Shift+M) and set the Baud rate to 2000000(used in the sketch), a window similar to figure-5 should appear showing live y-axis acceleration values from the sensor.

7- Now you must leave the board connected to you PC and close the Arduino serial monitor and move on to the next section below.

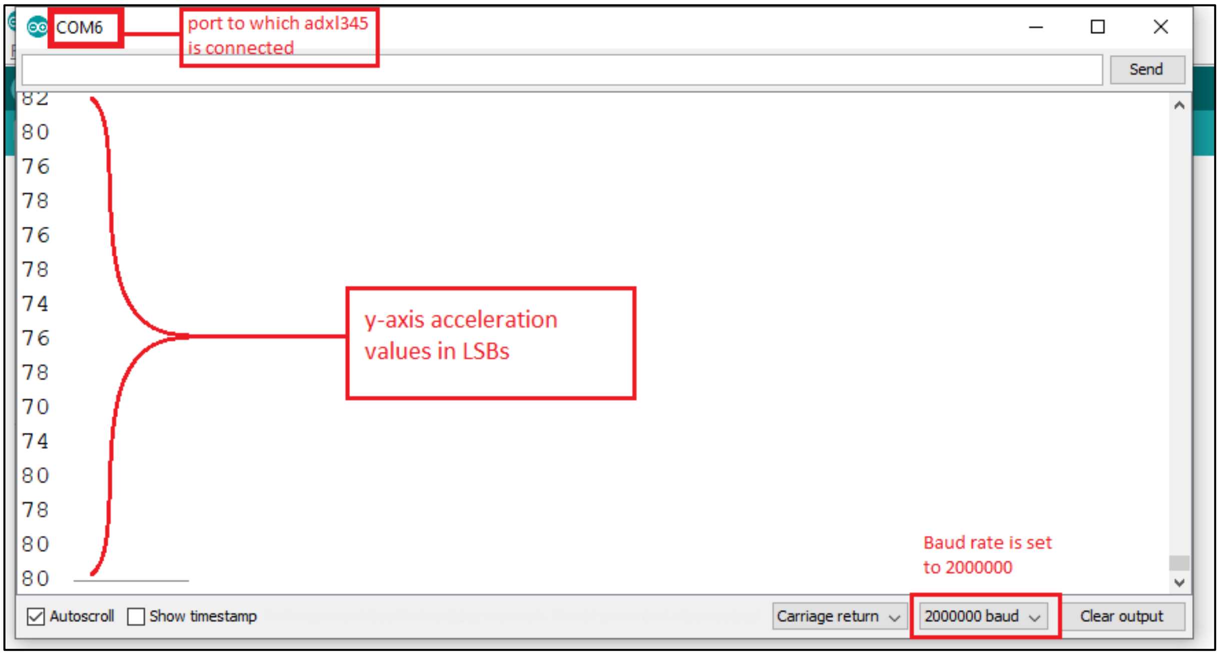

Print_Accel_y_LSBs | Arduino 1.8.7
File Edit Sketch Tools Help
Print_Accel_y_LSBs

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
  unsigned long cycletime = 360;
  unsigned long timenow = 0;


Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

void displaySensorDetails(void){
                               sensor_t sensor;
                               accel.getSensor(&sensor);
                               Wire.setClock(1000000);
                               accel.setRange(ADXL345_RANGE_16_G);
                               accel.setDataRate(ADXL345_DATARATE_3200_HZ);
                               }
void displayDataRate(void){}
void displayRange(void){}
```

Figure 4 Screenshot of Arduino IDE window showing the sketch "Print_Accel_y_LSBs" opened



port to which adxl345 is connected

```
82
80
76
78
76
78
74
76
78
70
74
80
78
80
80
```

y-axis acceleration values in LSBs

Baud rate is set to 2000000

☑ Autoscroll ☐ Show timestamp

Carriage return

2000000 baud

Figure 5 Screenshot of Serial monitor window showing live y-axis acceleration values coming from the ADXL345 sensor (Note that these acceleration values are in LSBs with a conversion 1LSB = 0.04m/s^2)

Visualizing & Logging the live Acceleration time-history and Spectrum of accelerometer data

This section helps you to setup the “Telemetry Viewer” application so that the accelerometer data coming from Adxl345 sensor can be visualized in time as well as in frequency domain (FFT of data)

Follow below steps in sequence

1- Download and install a compatible version of java runtime environment (JRE) from the official oracle website.

2- After Successful installation of JRE, go to “adxl345vibration/Monitoring and Data Logging” folder.

3- You will see three items namely “ModifiedTelemetryViewer.jar”, “Run.bat” and “Layout.txt”.

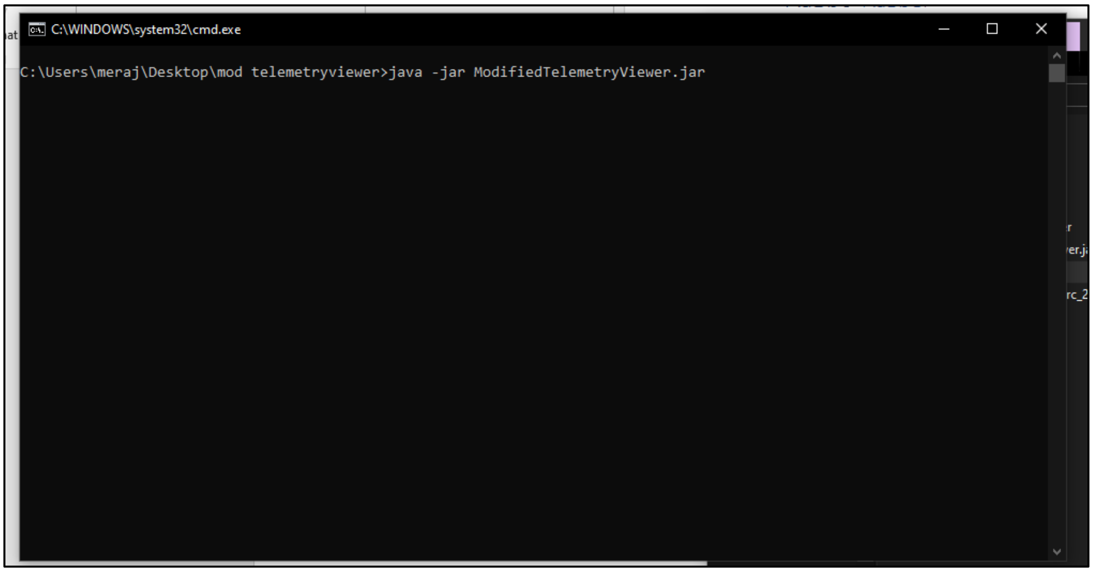
4- Run the “Run.bat” file, a command prompt similar to Figure-6 should appear. Wait for a while and a window similar to Figure-7 should appear. This is the main window of the “Telemetry Viewer” app.

5- On the bottom right corner of Telemetry Viewer app set the following fields (also shown in figure-8)

- Sampler Rate (Hz): 2743
- CSV
- UART: COMx (Selection should be made according to the port used by the Arduino see Figure-5 above)
- Baud Rate: 2000000 (Same as used in the Arduino)

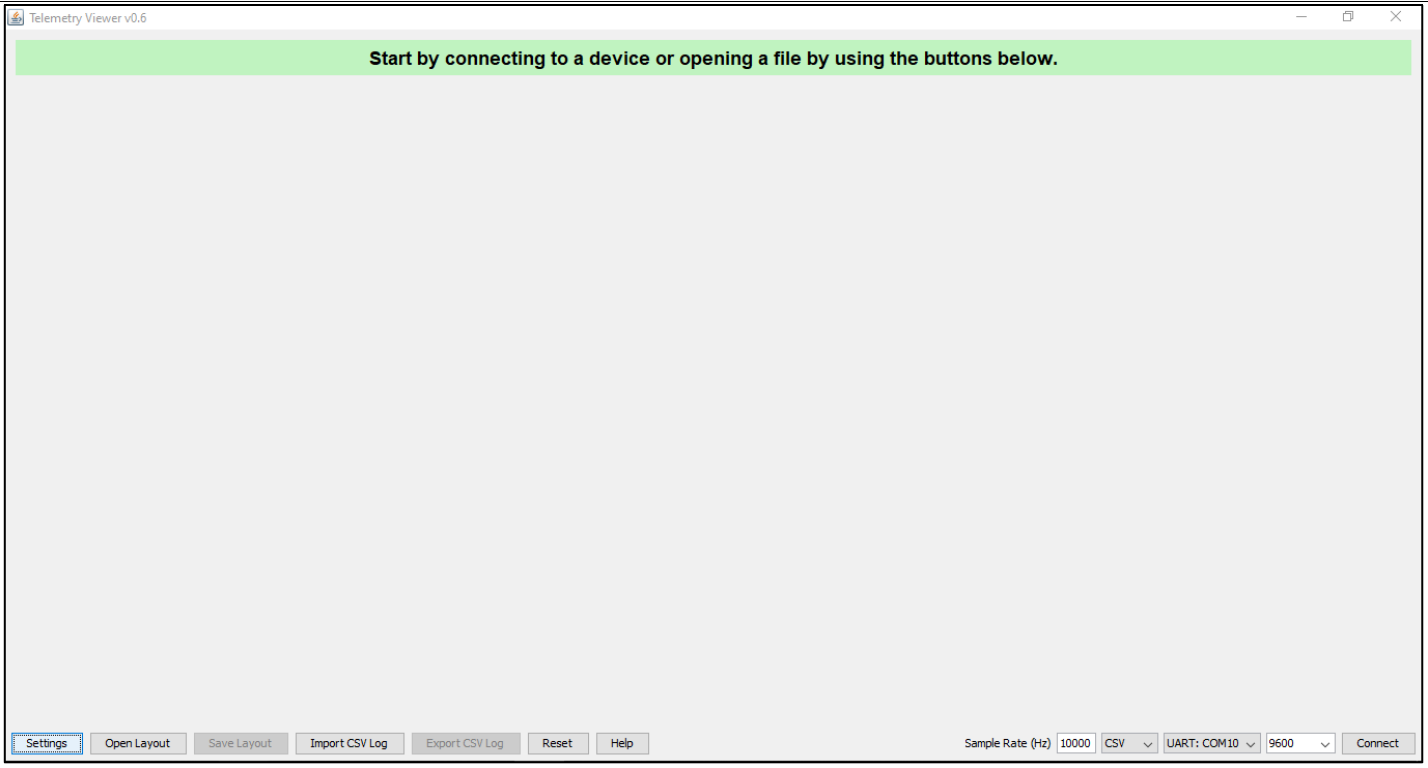
6- From the bottom left corner of main Telemetry viewer window go to “Open Layout” and navigate to the “Layout.txt” file present in “Monitoring and Data Logging” folder. The Telemetry viewer window should appear (as shown in figure-9) showing the live Acceleration time-history and its Spectrum.

7- Now, in order to save(log) the accelerometer data, go to “Export CSV Log” from the bottom of your Telemetry Viewer Screen and provide a name for your CSV log file. The logged CSV file can be used to import the recorded accelerometer readings in MATLAB for further analysis using some well-developed tools like Tom Irvine’s Vibrationdata GUI (https://vibrationdata.wordpress.com/2013/05/29/vibrationdata-matlab-signal-analysis-package/).



```
C:\Users\meraj\Desktop\mod telemetryviewer>java -jar ModifiedTelemetryViewer.jar
```

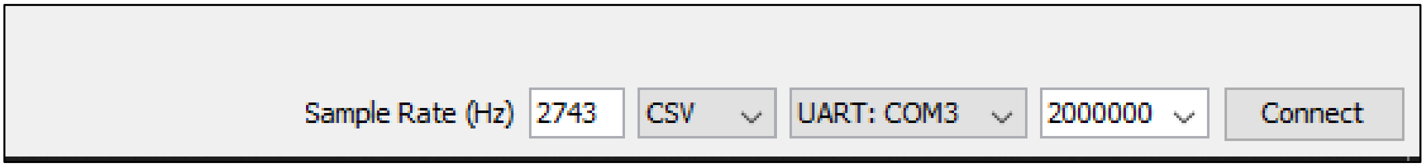
Figure 6 Screenshot of Command prompt that appears after opening “Run.bat” file.



Start by connecting to a device or opening a file by using the buttons below.

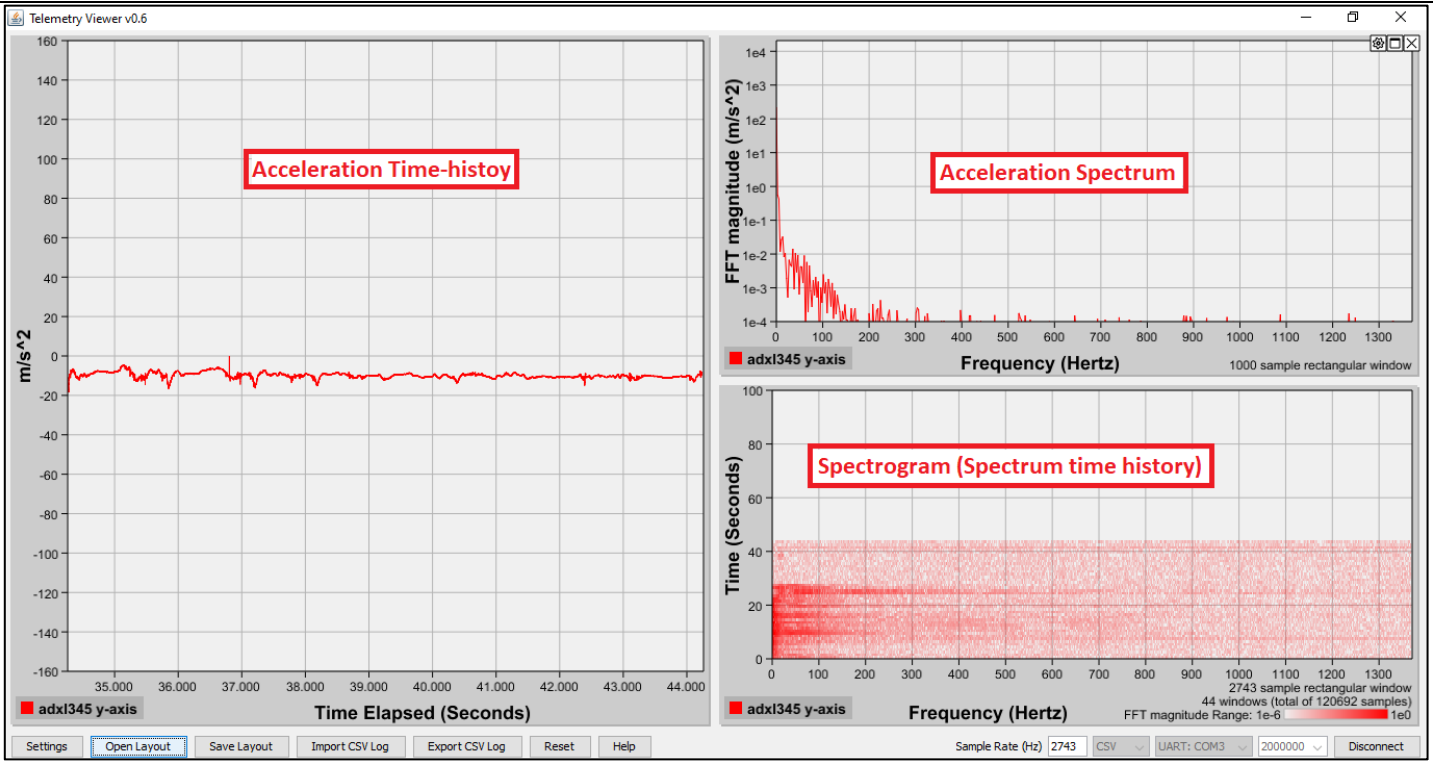
Sample Rate (Hz) 10000 CSV UART: COM10 9600

Figure 7 Screenshot showing the main window of the “Telemetry Viewer” app.



Sample Rate (Hz) 2743 CSV UART: COM3 2000000

Figure 8 Screenshot of Telemetry viewer showing the fields that needs to be set. Note: In my case the port COM3 was used by the Arduino. You must select your own correct Arduino port.



Sample Rate (Hz) 2743 CSV UART: COM3 2000000

Figure 9 Screenshot of Telemetry Viewer app showing Acceleration time-history, Spectrum and Spectrogram of ADXL345 sensor’s real-time data