Title page: EEDG/CE 6302 Lab Report 2
CE6302, Embedded Systems, 302 Laboratory - 83204
Lab Topic: Accelerometer
Yuyang Hsieh, YXH230019
Lab partner name: Xiongtao Zhang
Group Number 4
Instructor name: Tooraj Nikoubin
TA name: Seyed Saeed
Date: 9/8/2024

1. Objective:

Purpose: Lab 2 is to use the ADC module of MSP432 to input the values of 3-Axis accelerometer which measures g-forces, and display the output values on the LCD, as well as the serial monitor. Finally, use matlab to read the values of the 3-Axis accelerometer sent from MSP430, and visualize the values in a 3D coordinate graph in real time.
Methods used: To read and convert the analog values from 3-Axis accelerometer, use function analogRead() with Pin number as argument, and store the return value. To display the values in the LCD, create an instance of the HX8353E screen, and use myScreen.gText() passing with coordinates, strings, and screen color. This would display strings in the LCD with specific location and color. To receive the message from the serial port, use Serial.print() with values as argument. To split the values in matlab, first print specific strings between each value to distinguish, then use split() in matlab, explicit with corresponding strings, to split each value. To set delay after each reading value, use function delay().
Results: The LCD screen on the MKII displayed the values of the accelerometer in x,y,z axis respectively, with certain update rate. While tilting the device, the values changed correspondingly. In Matlab, the values were received through serial port and were plotted in 3D graph in real time.
Hardware used: SimpleLink MSP432P401R LaunchPad, Educational BoosterPack MKII.
Software used: Energia-1.8.10, Matlab
Major Conclusions: In lab 2, an accelerometer read and display program is made, and Matlab is used to plot 3D graph.

2. Introduction:

2.1 Hardware and Software Background Information

An accelerometer is a device that measures the proper acceleration of an object. Proper acceleration is the acceleration (the rate of change of velocity) of the object relative to an observer who is in free fall (that is, relative to an inertial frame of reference). In lab 2, we used a 3-Axis accelerometer (Kionix KXTC9-2050) which integrated into the MKII board.

MATLAB is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

2.2 Purpose of the Experiment

Understand the working concept of MSP432, know how to convert analog values with ADC, control the LCD screen, be able to communicate through serial port, understand programs in C and utilize Matlab to plot graph.

2.3 Summary of the Experiment

In lab 2, a program is made to read analog values from an accelerometer and convert them to digital and display them on LCD. Matlab is used to receive the data from the serial port and plot it in a 3D graph.

2.4 Findings of the experiment

In Lab 2, the working concept of the MSP432 microcontroller and its connection with peripheral devices was explored. The process of using the serial port to output data from the MSP432 was covered, along with utilizing MATLAB to split the incoming data into arrays and plot a 3D graph for visualization.

3. Explanation:

3.1 Experiment procedure

The programming of the MSP432 microcontroller for reading and displaying values from the accelerometer begins by including the header for the LCD display and the Educational BoosterPack MKII. The analog input pins for the X, Y, and Z axes are initialized, connected to pins 23, 24, and 25, respectively. Serial communication is initiated in the setup() function at a baud rate of 9600 to send data to MATLAB. The program reads analog values from the accelerometer using analogRead() for each axis. A delay is added after each reading to allow MATLAB sufficient time to process the data.The initialization for the LCD screen is performed using myScreen.begin(), which involves hardware setup, connecting the MSP432 microcontroller, and verifying and uploading the code to display the accelerometer values. In MATLAB, a for-loop reads 1000 lines of data from the device, splits the incoming data into an array, converts string values to doubles, and sets limits for the X, Y, and Z values to prevent overflow. A grid is then created to plot the last value, which is continuously updated with data from the accelerometer.

3.2 Experiment Images

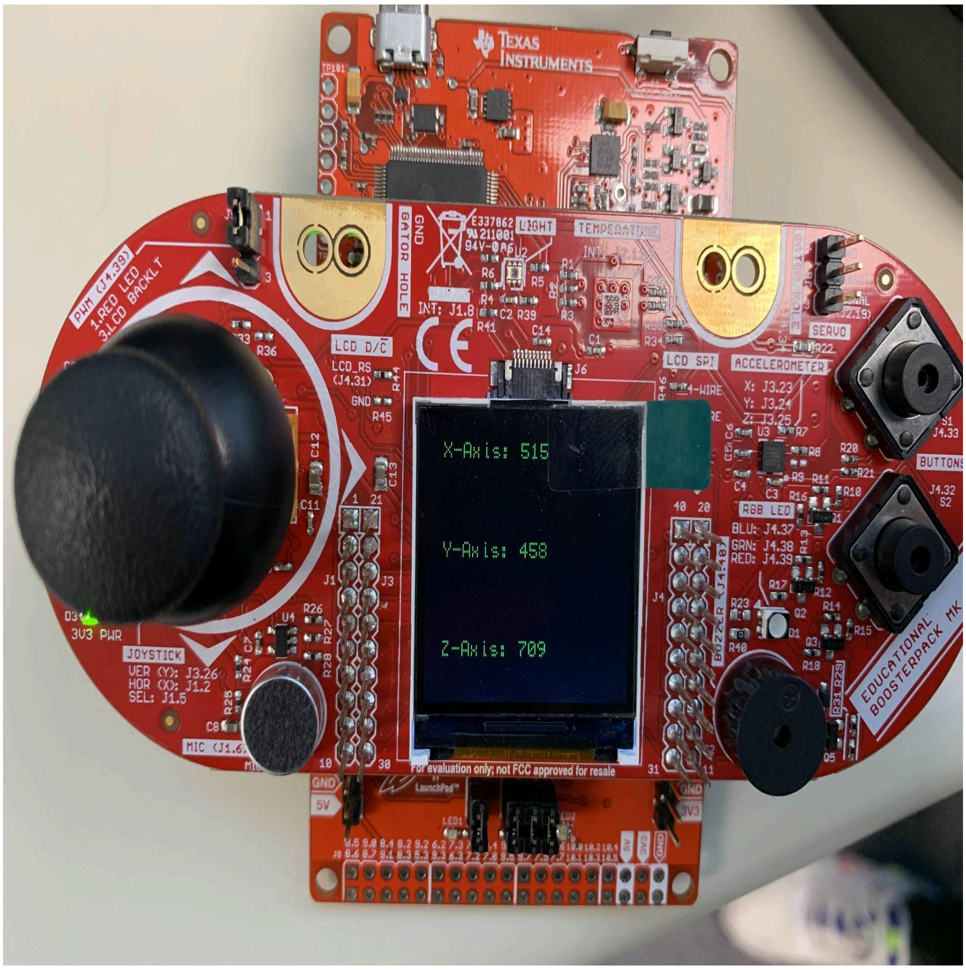

Image 3.2-1: acceleration values on LCD

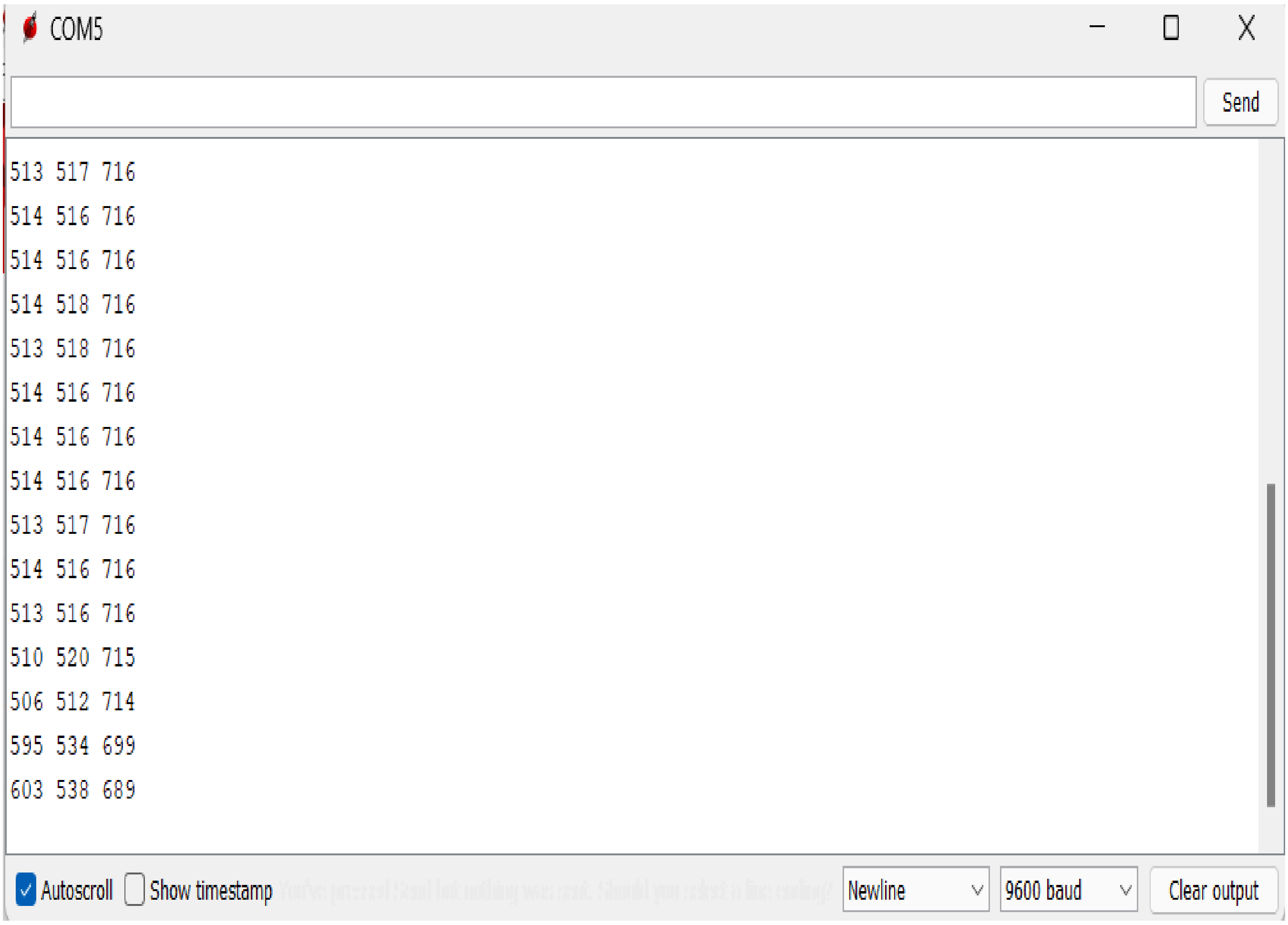

Image 3.2-2: MSP430_Display_Serial_Monitor

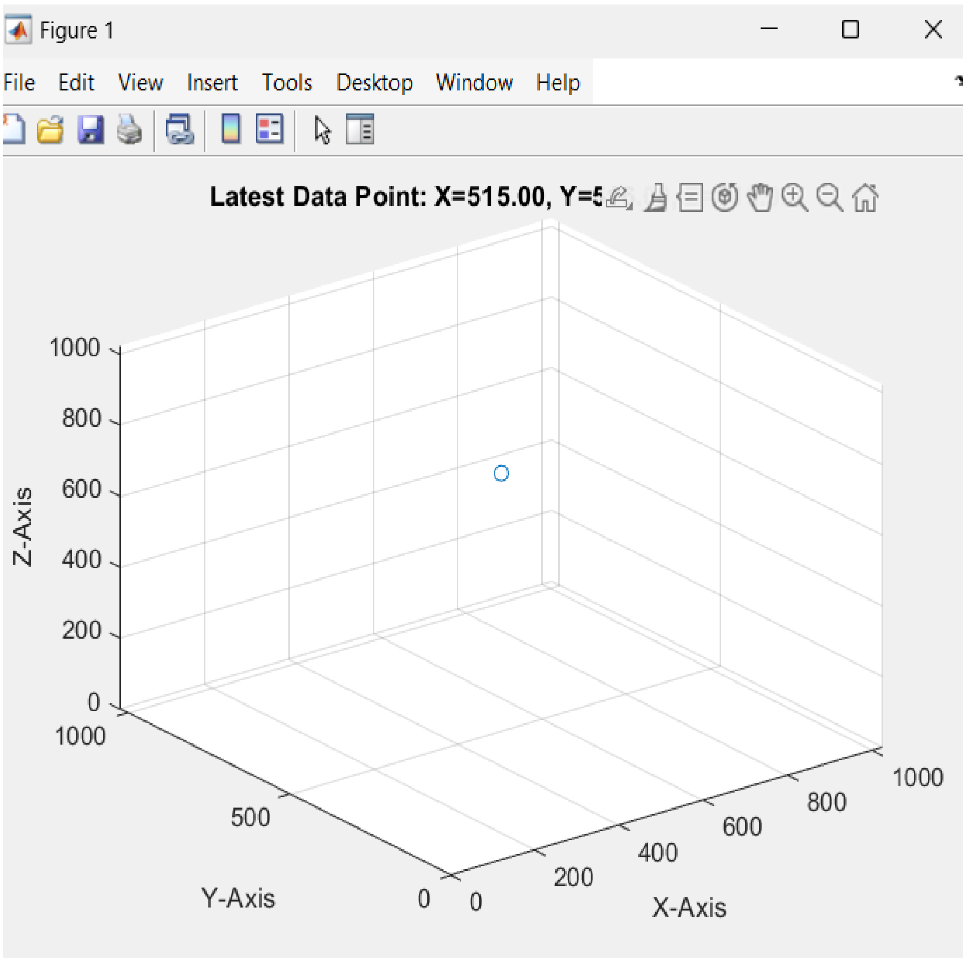

Image 3.2-3: Matlab_3D_Display

3.3 Codes

```
// Core library for code-sense
#include "Energia.h"

// Following includes for Energia only
#include "SPI.h"

// Include the library for the LCD screen
#include "Screen_HX8353E.h"

Screen_HX8353E myScreen;

const int xpin = 23; // x-axis of the accelerometer
const int ypin = 24; // y-axis
const int zpin = 25; // z-axis (only on 3-axis models)

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

  // initialize for sending diagnostic info to computer
  Serial.begin(9600);

  //Initialize the LCD Screen
  myScreen.begin();
  myScreen.clear(blackColour);

  // Set the analog pins as inputs
  pinMode(xpin, INPUT);
  pinMode(ypin, INPUT);
  pinMode(zpin, INPUT);
}
```


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

  // initialize for sending diagnostic info to computer
  Serial.begin(9600);

  //Initialize the LCD Screen
  myScreen.begin();
  myScreen.clear(blackColour);

  // Set the analog pins as inputs
  pinMode(xpin, INPUT);
  pinMode(ypin, INPUT);
  pinMode(zpin, INPUT);
}
```

```
void loop() {
  // Read the analog values from the accelerometer
  int x = analogRead(xpin);
  int y = analogRead(ypin);
  int z = analogRead(zpin);


  int xAccel = x;
  int yAccel = y;
  int zAccel = z;



  // Display the XYZ values on the LCD screen using myScreen.gText()
  myScreen.gText(10, 10, "X-Axis: " + String(xAccel) , greenColour);

  myScreen.gText(10, 55, "Y-Axis: " + String(yAccel) , greenColour);

  myScreen.gText(10, 100, "Z-Axis: " + String(zAccel) , greenColour);


  // Also send the values to the Serial monitor
  Serial.print(xAccel);
  Serial.print(" ");

  Serial.print(yAccel);
  Serial.print(" ");
  Serial.println(zAccel);

  delay(200);
}
```
Image 3.3-1: Energia Code

```matlab
serialPort = 'COM5';
baudRate = 9600;

clear s;
% Create serial object
s = serialport(serialPort, baudRate);



% Set limits for the axis to ensure that the
% acceleration values do not overflow
xLimit = [0, 1024];
yLimit = [0, 1024];
zLimit = [0, 1024];

% Loop for 1000 iterations
for i = 1:1000
    % read the line from the device
    data = readline(s);

    %  split it into an array
    dataArray = split(data, " ");

    % Check if the array has three elements for x, y, and z
    if length(dataArray) == 3
        % Convert string to a double value.
        xData = double(dataArray(1));
        yData = double(dataArray(2));
        zData = double(dataArray(3));

        % Plot x, y and z axis with a marker ‘o’
        plot3(xData, yData, zData, 'o');

        % Set axis limits to avoid overflow
        xlim(xLimit);
        ylim(yLimit);
        zlim(zLimit);

        % Label the axes
        xlabel('X-Axis');
        ylabel('Y-Axis');
        zlabel('Z-Axis');

        % Create a grid
        grid on;
```

```
            % Display the latest data point
            title(sprintf('Latest Data Point: X=%.2f, Y=%.2f, Z=%.2f' ...
                , dataArray(1), dataArray(2), dataArray(3)));

            % Update the plot
            drawnow;
        end
    end

    % Close the serial connection after 1000 readings
    delete(s);
    clear s;
```

Image 3.3-2: Matlab Code

4. Discussions and Conclusions:

4.1 Comparison of your experimental results with the research/theory of the concept with reasoning.

In theory, the g-forces were measured using the 3-axis accelerometer (Kionix KXTC9-2050) on the MKII and compared with expected theoretical outcomes. The signal produced by acceleration varies when the board is moved along the x, y, and z axes due to a combination of movement and gravity. One application of the accelerometer is measuring tilt by sensing changes in these forces. In the lab, the LCD screen displayed different values for the x, y, and z axes while tilting the device at various angles. Using the myScreen.gText() function, the analog values were displayed on the screen, consistent with the theory that the accelerometer measures both gravitational and dynamic forces. This data was then visualized using MATLAB on a laptop by reading the acceleration values from the board, which corresponded to the theoretical understanding.

4.2 Learnings from the experiment

The project involved displaying accelerometer readings in the x, y, and z directions on an LCD screen and transmitting the data to MATLAB for 3D plotting. Initially, the output was formatted using Serial.print("\t") in Energia, but MATLAB failed to render the 3D plot as expected. After troubleshooting, it was determined that MATLAB was not handling the tab characters (\t) in the incoming data stream correctly. Replacing the tab characters with spaces (" ") allowed MATLAB to interpret the data properly, enabling the 3D plot to display as intended.

4.3 Video Link: https://youtu.be/OWyHs7Dauq0

5. References:

Wikipedia: https://en.wikipedia.org/wiki/Accelerometer
Matlab: https://www.mathworks.com/
Texas Instruments: https://www.ti.com/
Texas Instruments, Educational BoosterPack MkII Guide: https://www.ti.com/lit/ug/slau599b/slau599b.pdf
Energia: https://energia.nu/