EE445L lab1 report

Salim Memon (sam6345)

Priscilla Chang (pc23384 )

A) Objectives: -

The objective of this lab was to refresh our usage of Keil environment, and get accustomed with the Stelaris LCD screen. This was done by writing software in C, that dealt with fixed point and producing the results on a screen incorporating graphical capabilities. The software consisted of two main files, the first one being a header files called fixed.h. The header file contained within it the definitions for four different functions, and the second file, fixed.c, contained the implementations of those functions. The first and second part of the function aimed at converting integers to decimal and binary fixed points. The third function focused on initializing the display, followed by the final function that drew the pixels onto the LCD based on their coordinates.

A secondary objective of this lab was to teach us about the difference between using fputc and printf methods. The fputc method outputs a character to the screen, whereas the printf method called the fputc method for every character within the string. The advantage of using the printf method was simplicity in coding, however the code ran at a slower speed. Fputc method would create complicated code but would provide an optimized version of the code.

E)

1) By decreasing the number of arrows in our call graph, the function becomes modular. In other words, this lab could be reused for future projects with minimal modifications and be able to produce results on the LCD screen.

2) The decimal must remain in the same physical place as to obtain alignment for the values that would be displayed on the screen. Since they were in the same physical place, it allowed the user to be able to easily view the output on the screen. Since we knew that the output characters had to be at max length of 6 characters, the correct alignment confirmed that no additional values were produced on the screen, for example, the value 1.23 would only produce "1.23" and no extra characters.

3) If high speed is required with a fixed range of values, as well as precision not being as much of a concern, then fixed point should be used over floating point.

When precision is of high priority, speed is not as big of a deal, and the range of values is large floating point is chosen over fixed point.

4) When the range of values is bounded, and speed is of the highest priority, binary fixed points are used as opposed to fixed points.

The fixed point code is usually not that easy to be recognized and so it is usually always preferable to use binary fixed points, however one can manually confirm the overflow bits using fixed points.

5) Example Application: AA microwave used within households does not have requirements for high precision as the general user would not enter the weight of the material being heater at anything more than 2 decimal places. Also by avoiding the use of floating points it would decrease the costs incurred by the hardware, and provide a faster output.

6) ARM M4 allows the use of floating points. The costs incurred by using floating points is the power consumption due to using additional hardware, and along with higher computational time.