EE445L – Lab 1: Fixed Point Output

Kyle O’Brien and Maykei Nguyen

Bard Spring 2012 – TA: Nachi

1. **OBJECTIVE**

The goals for this lab are to get familiarized with Keil uVision software and our Stellaris Development Board as well as investigate the use and implications of fixed point numbers and arithmetic. Because the ARM Cortex M3 microprocessor does not have a floating point logic unit as a part of its microarchitecture, fixed point arithmetic will have to be implemented everywhere exact calculations are required.

This lab requires us to understand the different aspects of fixed point numbers such as precision and variable integers. In addition, we must understand how to create appropriate fixed point numbers for different situations and how to preform arithmetic upon them. By doing this lab were able to produce code that accepts fixed point numbers in both decimal and binary format then print them in human-readable decimal format on the OLED screen.

1. **ANALYSIS AND DISCUSSION**
   1. In what way is it good design of fixed.c that there is no arrow directly from the fixed.c module to the rit128x96x4.c module in the call graph for your system?

*A call to the rit128x96x4 module would be subverting the main function and the OLED output functions. Ignoring these controlling programs and interacting directly with the OLED driver could possibly corrupt a something that the driver is already trying to output under direction from the out functions.*

* 1. Why is it important for the decimal point to be in the exact same physical position independent of the number being displayed?

*When the decimal is in the same point, it makes the display “pretty” and readable because all the numbers are lined up.*

* 1. When should you use fixed-point over floating point? When should you use floating-point over fixed-point?

*You should use fixed point over floating point when the range of values is known and relatively small. Floating point should be used to accommodate a large range of values.*

* 1. When should you use binary fixed-point over decimal fixed-point? When should you use decimal fixed-point over binary fixed-point?

*You should use binary fixed-point when you are not expecting system interaction with a human operator. The Arm processor is very efficient in bit-shifting so binary-fixed point is much faster to process. When expecting system interaction with humans, you should use decimal fixed-point.*

* 1. Give an example application (not mentioned in this lab assignment) for fixed-point. Describe the problem, and choose an appropriate fixed-point format. (no software implementation required).

*Fixed point could be used to implement a simple kitchen timer. The range would be from 0:00 to 59:59 (Minutes:Seconds). For this application you would need a resolution of 0.01 and a way to alter basic arithmetic calculations to accommodate the 60-place value for seconds. The most appropriate format would be 32bit fixed-point decimal.*

* 1. Can we use floating point on the Arm Cortex M3? If so, what is the cost?

*No we cannot use floating point on the Arm Cortex M3.*

* 1. (EXTRA CREDIT) Is fixed-point or floating-point arithmetic faster on the Pentium w/MMX?

*Fixed-point arithmetic is faster on the Pentium w/MMX. Using fixed-point arithmetic (integer commands), we can execute two instructions at a time as opposed to only one floating point instruction.*