EE 445L - Lab 1: Fixed-point Output

Joshua Bryant jmb6357 James Morris jsm3288

September 12, 2014

1 Objective

The main objectives of this lab were as follows: to introduce and ourselves with the provided lab equipment, to familiarize ourselves with Keil μVision4 for the ARM Cortex M processor, to develop a set of useful fixed-point output routines that we can use in future labs, and to re-familiarize ourselves with fixed-point arithmetic and the differences in properties between fixed-point and floating-point.

2 Analysis and Discussion

1. It is good to design fixed.c to not be tied specifically to any low-level display routines to allow the code to remain portable. For instance, instead of calling the hardware specific function **ST7735_OutString()**, calling the generic **printf()** function allows whatever code calling our functions to specify what output is appropriate.

2. Guaranteeing that the decimal point is in the same physical position regardless of the number being printed is important for visual formatting on output screens. A standard format allows for easy reading of the information and for any form of formatted output where one may only want to modify the integer portion of the number and not the mantissa.

3. Fixed point numbers are useful if higher precision is needed in a number as well as less hardware complexity being required to operate on fixed point numbers. Floating point should be used if a larger range is required.

4. Binary fixed-point is preferable for scaling values by powers of 2 since this results in fast bit-shifts in the hardware. Binary fixed-point also has the benefit of being able to exactly represent fractional powers of two. Decimal fixed-point is useful for exactly representing fractional powers of 10 since these can only be approximated by binary fixed-point.

5. One example of an application for fixed-point would be real-time digital signal processing where using a microcontroller that supports floating point arithmetic would be too costly to be commercially viable.

6. Yes, it is possible to use floating-point on the ARM Cortex M4. It would require using an M4F processor and would require more complicated hardware in the chip used. Using floating point, in general, is also slower than using fixed-point arithmetic.