Harley Ross

Hmr566

Dalton Altstaetter

Dea528

Dr. Bard

1/26/14

Lab 1 Objectives

The objectives for this lab were to become familiar with the IDE and implement fixed point values for different data types. We became familiar with the compiler by building a new project, debugging, and coding.

The first task was to convert an unsigned 32-bit number to a fixed-point string with resolution 0.01 and ranging from 0.00 to 999.99. If the number is out of range then it had output \*\*\*.\*\*.

The second task is to print a 32-bit unsigned fixed-point string on to the OLED or simulator.

The third task is to convert a signed 32-bit number to a signed fixed-point string with resolution 0.001 and ranging from -9.999 to 9999. Inputs are expected to be between -9999 and 9999 and everything out of this range is stored as \*.\*\*\*.

The fourth task is to print that signed 32-bit fixed-point number to the OLED or simulator.

The fifth task is to convert an unsigned 32-bit binary fixed-point number with resolution 1/256. Input values range from 0 to 255,999 and output values range from 0.00 to 999.99. Inputs out of this range are stored as \*\*\*.\*\*.

The sixth task is to print that unsigned 32-bit binary fixed-point number to the OLED or simulator.

Lab 1 Analysis and Discussion

1) It is a good design because the code becomes more modular. It allows a higher level of abstraction for future use and this modularity makes it a simpler task for debugging purposes.

2) The hardware is limited by the amount of characters it can display.

3) Floating-point is used for larger ranges of values than the hardware has in bits to represent. If your ranges of values are large relative to the number of available bits floating-point is more appropriate. Fixed-point values are more precise and are used when when the bits of the hardware can represent all of the values in the data’s range. Fixed-point is also more computationally efficient.

4) Use binary fixed-point when doing computations in the computer that do not necessarily need to be displayed. Binary fixed-point is faster since it can be accomplished by bit-shift operations. Decimal fixed- point should be used when having to display those values that are computed so that it can be easily interpreted by people.

5) An example for fixed point applications could be financial documents or any measuring device, such as a gas pump that has three digits in front and behind the decimal point.

6) Floating point does not come standard, but it could be implemented through software that converts the value to and from floating-point values.

Extra Credit: Fixed- point arithmetic is almost always faster than floating-point arithmetic.