

Tarang Khandpur – tk8435

Karime Saad – ks38728

Lab 1 Deliverables

Objectives:

Our objectives for this lab are to develop a solid understanding of how to use the TM4C123 Microcontroller, the LCD, and the Keil IDE. In addition to this fundamental knowledge of interfacing with the lab components, we also aim to program a set of functions to output fixed point conversion for base10 and base2. Familiarizing ourselves with the Micro-Controller, LCD and Keil will allow us to complete the subsequent labs without spending time on the “setup” phase of the project. A good grasp of Keil’s processes and modes also enables us to debug and test our program efficiently. Furthermore, we will manipulate input data for a set of points to draw a particular shape in order to fit our LCD parameters. This will require scaling the data appropriately and offsetting it to achieve the desired image. These goals together encompass our objectives for Lab1.

Analysis and Discussion:

1) In what way is it good design to minimize the number of arrows in the call graph for your system?

-- Minimizing the arrows in the call graph is analogous to reducing the number of function calls, which simply translates to less instructions therefore quicker execution.

2) Why is it important for the decimal point to be in the exact same physical position independent of the number being displayed? Think about how this routine could be used with the ST7735_SetCursor command.

-- The position of the decimal point helps to clarify the level of precision and resolution that the routine is expecting, and it also indicates the correct conversion of the value. If the position of the decimal were different for various inputs the meaning of the conversion would be different as well. By placing the decimal in the same position the difference in input values can be easily compared. Also to retain the precision of the value, instead of simply dividing by the resolution we had to multiply/shift and store each individual digit. By doing so, we kept the conversion accurate but now required the decimal to be printed separately from the actual calculation. If the function assigned the decimal to different positions it would be difficult to reliably set the cursor to account for changed process.

3) When should you use fixed-point over floating point? When should you use floating-point over fixed-point?

-- You use fixed point for precise calculation, and where the range is known and possible to compute. For example if you were calculating exchanges at a bank, you would want to use fixed point to compute an accurate result, and you would know the maximum possible exchange amount. The resolution for a money transaction would be a fixed .001 to retain the value.

-- You use floating point when the range is large and not known. For example, if we were measuring various temperatures across the universe, we would like to represent the values easily and therefore must have a variable resolution to accommodate from the coldest point in the universe to the hottest point in the universe.

4) When should you use binary fixed-point over decimal fixed-point? When should you use decimal fixed-point over binary fixed-point?

-- We should use binary fixed point over decimal fixed point when performing mathematical calculations. Binary fixed point is more efficient because the process of multiplication and division only involves shifting bits which do not add to the execution time. However, when displaying numbers, it is preferred to use decimal fixed point since it is better understood by humans.

5) Give an example application (not mentioned in the book) for fixed-point. Describe the problem, and choose an appropriate fixed-point format. (No software implementation required).

-- As stated earlier, money transactions are ideal examples for fixed point usage since the range of possible values is known and the precision is limited to the currency. To further detail a scenario, let's say you are a bank that accepts transactions from a minimum of 1 cent to 1 billion dollars. That is a known range and easily represented in fixed point format since the resolution is fixed to .001 (to account for rounding of the cent's value).

6) Can we use floating point on the ARM Cortex M4? If so, what is the cost?

-- Yes, we are able to utilize floating point format on the ARM Cortex M4. The associated cost requirement would be additional RAM to store and calculate operations on the floating point values.