Memo to: Randy Larimer
From: Johnny Gaddis
Date: 2/22/17
Regarding: EELE 465, Lab 2

Summary:

The purpose of lab 2 was to use the HCS908QG to run a decoder that clocks 5 devices. A transceiver and a Flip-flop were used to read in values from a keypad, and then LED's and an LCD were flashed using two more Flip-flops based on keypad input. This task was accomplished using branching: when a key is pressed we jump to a specific subroutine. In each specific subroutine, they constantly branch to three subroutines: one for writing to the LEDs, one for Polling for keypress, and one for writing a letter to the LCD. This lab led to a greater understanding of the HCS908QG while showing students practical uses of microcontrollers and LCD screens.

Setup:

The wiring required for this lab was a big part of the setup. Everything was already wired but the LCD screen. The code was setup to initialize the LCD. This code took a long time to figure out with lots of bugs in it. It was remedied with the examples provided in the lab 2 folder. The keypad needed a write to LCD subroutine and a set address subroutine.

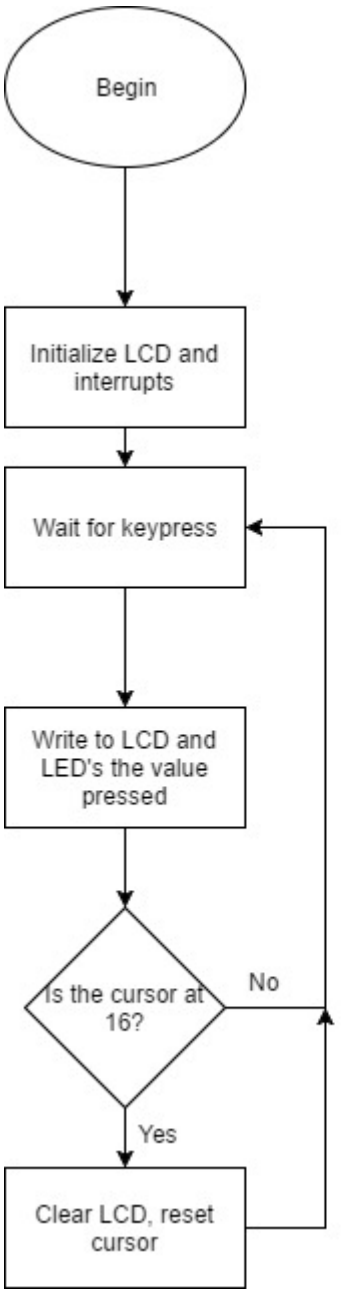
LCD Solution:

Firstly a subroutine was needed for each specific key. Once this was completed we had to write to the LCD with the value of the key and put a binary value on 4 LED's. A global counter was kept track of for the cursor location. Once the cursor location reached too far out the LCD was cleared. Also, a delay was introduced in the main loop so that characters could not be written too fast. One issue encountered was not setting inputs back to outputs on the data bus. Once the issue was noticed it was an easy fix.

Final Thoughts:

This lab taught me how to better interface with devices and user their datasheet to solve problems. There was some research required for this lab like going through datasheets and figuring out how these devices work. The work required to wire up another data bus element and to initialize the LCD in code were the only things difficult about this lab.

Appendix A Flowcharts:



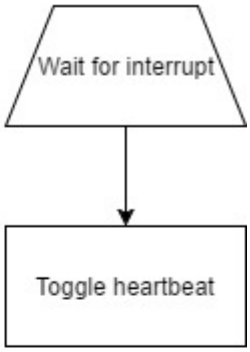


Figure 1: Interrupt Routine and Program Flow