**EE 474 Lab 2**

**Inputs, Outputs, and Time**

Jonathan Lockwood

Ahmed Boustani

David Anderson

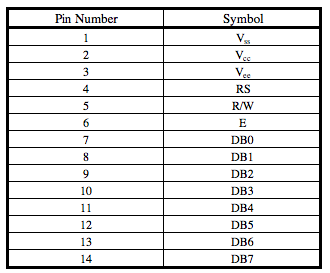
**INTRODUCTION:**

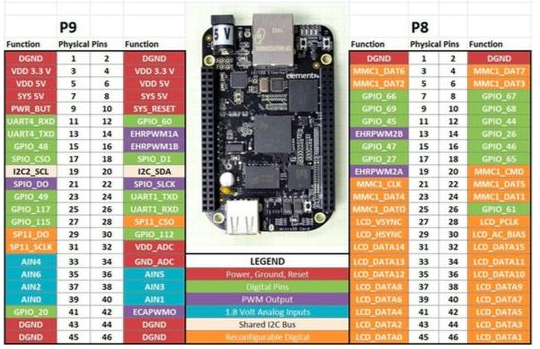
Throughout this lab, we continued our work on the Beaglebone Black and explored more aspects and capabilities of the device. This lab introduced an LCD display that can be hooked up to the breadboard and pins of the Beaglebone to be able to handle communication from the user. The GPIO pins were hooked up to the LCD display and the initialization process began. To test the scheduler that is inside of the system, we performed a jitter by showing the output waveform on the oscilloscope. We were then able to communicate with the LCD display through the terminal on the computer. Finally, we created a function that performed basic mathematical operations with the output being displayed on the LCD.

**OVERVIEW:**

The first task we had to complete was to initialize the LCD display. We began by wiring up the LCD with the GPIOs from the Beaglebone. Since there are 16 pins on the LCD display, it was important to double check the pinout diagram on the Beaglebone to make sure we did not mix up the GPIOs that were used. Since specific pins on the LCD corresponded to specific pins on the Beaglebone, it was necessary to know the functionality of each pin number. The pin assignments for the LCD and Beaglebone can be seen in Table 1 and Figure 1, respectfully.

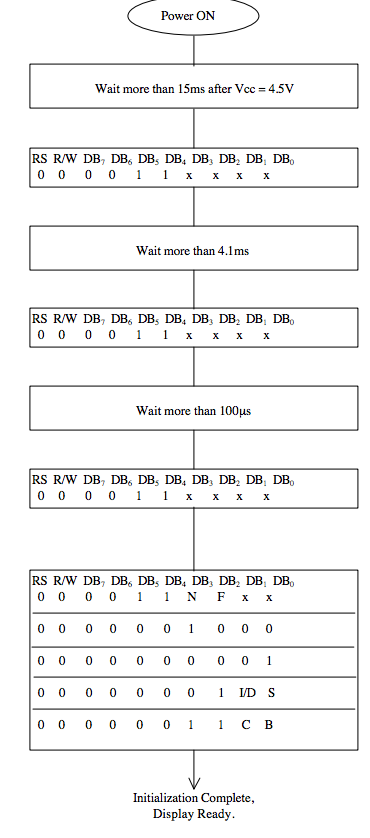
**Table 1:** LCD pin assignments





**Figure 1:** Beaglebone Black pin assignments

Once the pins were all hooked up correctly, we needed to initialize the system. We chose to operate the device in 8-bit mode. The block diagram can be seen in Figure 2 below.



**Figure 2:** 8-bit initialization block diagram

Since the first 4 bits are always 0 in every function set command, we chose not to include them as parameters in our code to minimize the number of variables used. After correctly initializing the LCD display, the cursor began blinking in the first position of the display. To test the scheduler jitter, we used another GPIO pin to be the system monitor. The output waveform was then shown on the oscilloscope and a simple function was used to make the signal go from high to low. After the initialization of the LCD, we began sending text to the display from the terminal. To accomplish this task, we set up a pipe that passes the output of the command to the input of the display. Pipes allow two commands to be connected together so the output of one program can be the input to the other. Once we were able to successfully communicate to the LCD display, we began the final task: a simple calculator. Our design was to be able to type out a mathematical operation on the terminal and once we press “enter”, the answer to the equation will be shown on the LCD display.

**FUNCTIONS:**

Our first program allowed us to communicate with the LCD display. We sent the initialization code into the LCD screen to get the cursor to appear. We also had used a pipe to read in the text that the user inputted. Additionally, we used scripts to streamline the initializing process and our calculator function. Our calculator allowed us to perform a simple mathematical operation and have the output displayed on the LCD display.

**RESULTS:**

We were able to successfully initialize the LCD display and communicate with it effectively. We had initial difficulty initializing the display but after careful examination we realized we made some minor mistakes. At first, we weren’t sure if the problem was software or hardware related. We began debugging the code on a physical level by implementing an LED to see if it would light up if the bit was a ‘1’. We came to the conclusion that there was a hardware issue since the code seemed to be working fine. After taking another look at the wiring, we realized we incorrectly assigning one of the GPIO pins to the display in our code. In our program we declared that a certain GPIO pin was being used while in the physical system, we never actually wired that certain pin to the LCD display. Another issue we had was figuring out how to print to the second line on the LCD. To fix this problem, we realized that we needed to use the addresses on the second line which started at 40H. Despite these problems, we were able to successfully send text to the LCD display through a pipe as well as testing the scheduler jitter on the oscilloscope. Finally, we were able to successfully program the system so we could input a mathematical expression into the system and the LCD would display the correct output.

**CONCLUSION:**

In conclusion, we were able to successfully communicate and send text to the LCD display. We were able to become even more familiar with the GPIO pins of the Beaglebone by correctly assigning them to the pins of the LCD display. Enhancing our skills in the C language, we were able to initialize the LCD display correctly and send text from the terminal to appear on the LCD. Finally, we developed a program to perform math calculations while having the output and equation being displayed on the LCD. Throughout this lab we learned more about the Beaglebone and working with the LCD display and making it fully functional.