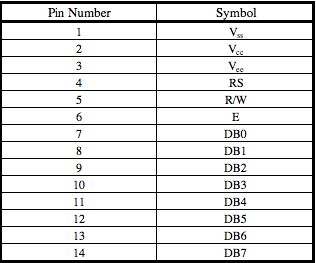
**Introduction:**

In this second lab assignment we integrated a simple LCD display with our BeagleBone micro controller. The LCD was first mounted onto our breadboard, and then wired to the GPIO pins on the BeagleBone. These GPIO pins were accessed in much the same manner as our first lab project. Two different programs were written that interfaced with the display; the first is a program that allows the user to type on the computer keyboard and send this text to the LCD display; The second is a simple guessing game in which the user tries to guess a random number generated by the program. The LCD for this second program displays the current state of the game.

**Design Specifications:**

Hardware Implementation:

Before anything could be run, the LCD display had to first be connected to the BeagleBone correctly. To do this, GPIO pins on the BeagleBone were designated for each of the pins of the display. The following table (**table 1**) was used to direct our wiring.

A potentiometer was connected in series with the Vee to give the user some control over the darkness of the LCD display. The higher the resistance of the potentiometer, the **DARKER/LIGHTER** the screen became. After thorough testing with the oscilloscope it was realized that GPIO 117 of the BeagleBone was not performing as expected. It seemed to be producing an oscillating triangular wave without fail. We therefore opted not to use this pin in our final design. Once all of the pins of the display were wired, the LCD could be initialized through software processes.

Software Implementation:

To get the LCD display up and running, a specific set of initialization instructions had to be sent to the display on boot up. In this initialization process, we specified our font size (5x7 pixel characters), the entry mode, and the cursor display status. Once this initialization was complete we could begin sending display data to the screen.

For the first program, text that was sent to the LCD had to first be decrypted before it could be interpreted by the display. A program was written to convert text entered by the user into ASCII binary values. These ASCII values were then read by the standard character table on the LCD RAM and drawn on the display. The display was configured so as to be large enough to display two lines, each of 16 characters.

The second program also made use of the LCD display, this time to show the state of a number guessing game. In this game the user is given an arbitrary number of guesses (the default is 6) to correctly guess a random number generated by the program. When the user makes an incorrect guess, the program generates a hint of whether to guess a larger or smaller value. The display is utilized in this game to show whether the game has been won or not, the number of guesses the user has left, and the higher/lower hint.