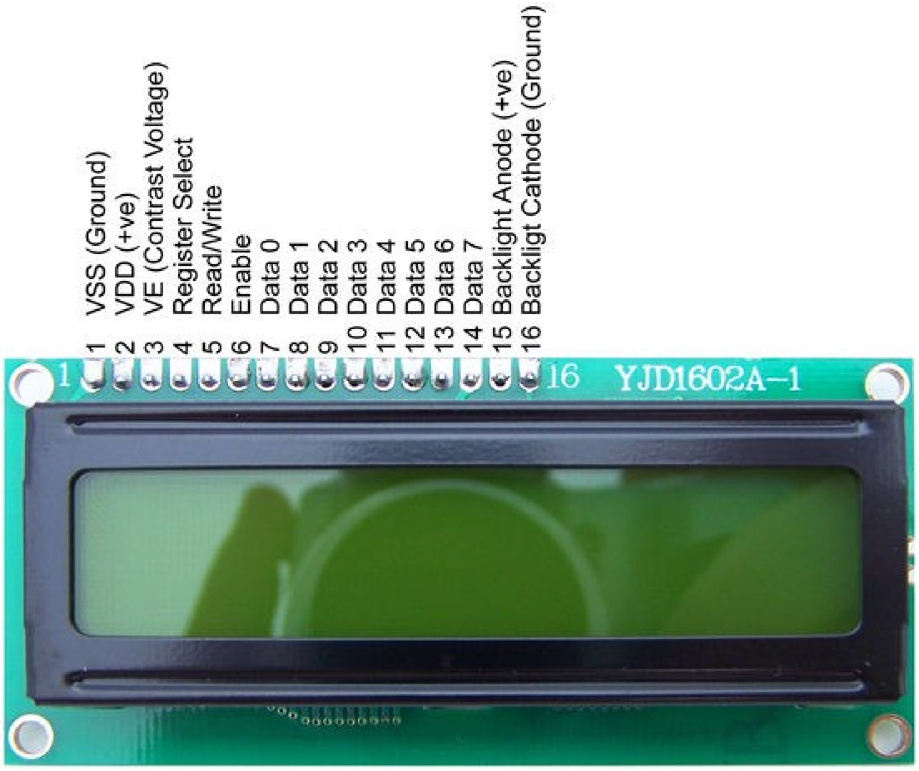
Lab Number 3

In this lab you will make use of the ultrasonic sensor circuit created in the previous lab. Instead of displaying the distance to the closest object on the screen, you will instead display this information on an lcd text display. The displays we will be using have a standard wiring pattern and any other display should work similarly. There are displays with I2C interface boards which have only four connections but they require specialized libraries and modification of the Raspberry Pi startup sequence in order to function properly.

The wiring is very important for the display to function and will require seven GPIO pins. Each of these pins will be set as output.

The image and wiring table below show the connections to be made and the superscript numbers in the table show the order of the pins to be used in the LCD function call.



G	+	Contras	Reg	Read	Ena	D0	D1	D2	D3	D	D	D	D	Bk	G

ND	5V	t	Sel[1]	/ Write	ble[2]					4[3]	5[4]	6[5]	7[6]	light[7]+3.3V	ND
Main Power for display		Display Contrast. Should go to the center pin of a potentiometer	Command when high and data when low	Always GND or voltage must be controlled	Used to decide when to send data to the display	No Connection	No Connection	No Connection	No Connection	Data	Data	Data	Data	This can be 5V or 3.3V (less bright)	GND for back light

We are using the display in 4-bit mode so we need no connection on the D0 – D3 lines. The software library will break the data down into two parts and send it to the display in the proper order. This saves four pins but will take twice as long to send the data. This shouldn't be a problem for anything we will need to do.

Once you have the LCD connected, you will need to define the size of the LCD for the software. In this case we are using a 16 character X 2 row (16x2) display. You should then define the pins being used before any data is written to the LCD. When finished, be sure to call the shutdown function to clear the LCD and turn off the display and back light.

I have created the myLCD.py library for you to use and it will give you all of the basic functions necessary. To use the myLCD library, you should have GPIO imported and set for BCM mode. You would then do the following to get started:

```
import myLCD
```

start(GPIO,rows, cols) will be used to define the size of the display

LCD(GPIO,RS[1], E[2], D4[3], D5[4], D6[5], D7[6], BKL[7]) is used to define the pins that will be used for each of the lines in the table above.

lcd_init(GPIO) is used to clear the screen and put it into an initialized state.

Now you can use any of the following functions to display your data and control the LCD as needed.

backlight(GPIO,state) is used to turn the back light on (True) or off (False)

lcd_string(GPIO,message, line) is used to display text on the screen

lcd_shiftleft(GPIO)

lcd_shiftright(GPIO) shift the display left and right by one character

lcd_shutdown(GPIO) is used to clear the display, and clean up any other LCD properties that were set

lcd_init(GPIO) can be used the clear the display at any time.

I will update this library if we require more functions later. You can look at the library code if you are interested in how the functions actually work.