Deepak Bharadwaj.R (2020H1400229H)
Shri Lakshmi.A (2020H1400217H)

Implementing Alpha-Numeric Characters in JHD 16X2 LCD Display

In 8-bit mode LCD operation

Objective:

The objective of this project is to write a Linux Kernel driver for 16x2 LCD in 8-bit mode and demonstrate its working in Raspberry Pi 3b+ interfaced with JHD 16x2 LCD.

Summary:

A kernel driver has been developed for 16x2 LCD display in 8-bit mode. The driver module contains code for normal LCD initialization and supports basic features for write operation. The commands and data are transmitted using 8 data pins. Control signals otherwise employed are RS signal and Enable port. RW port of LCD is assumed to always operate in write mode.

Components:

- Raspberrypi 3b+
- JHD162A 16x2 LCD display
- 10k pot

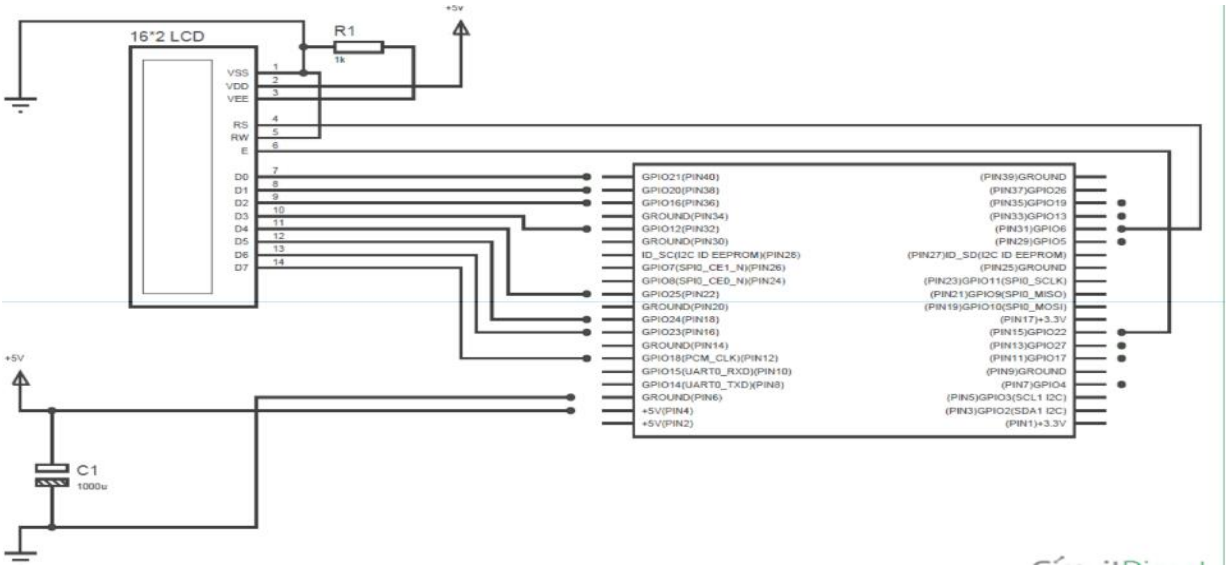
Software tools:

- VNC Viewer
- Putty

Mode of Operation:

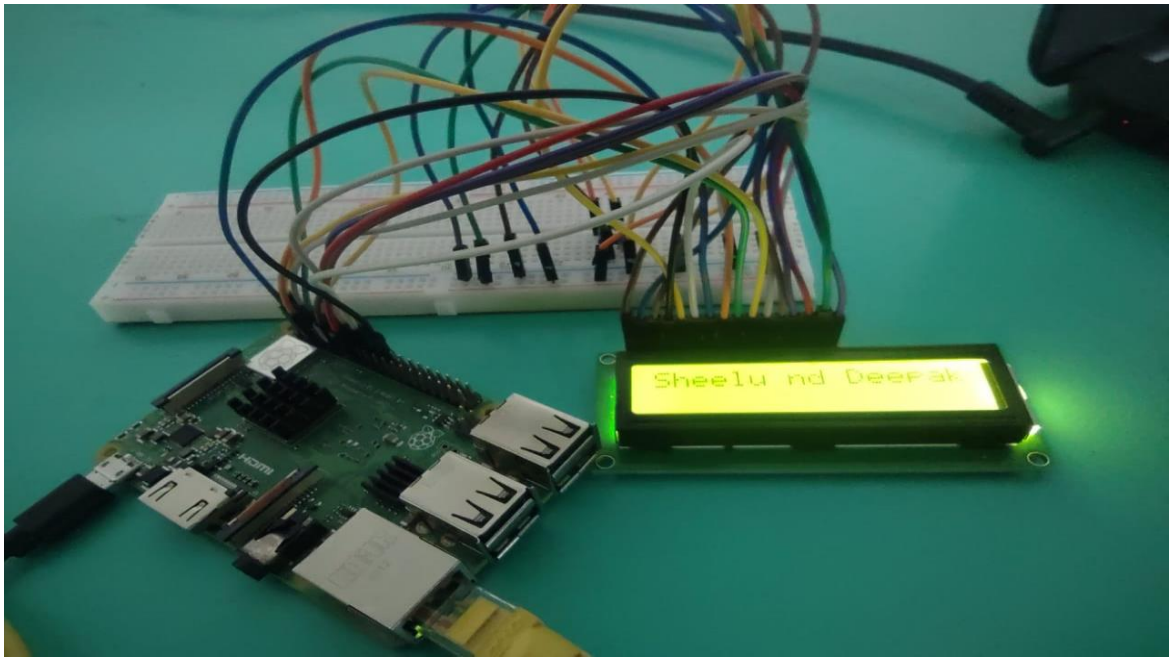
- 8-bit mode
- Only write operation
- Two line with maximum character limit at 16

Circuit connections:



Deepak Bharadwaj.R (2020H1400229H)
Shri Lakshmi.A (2020H1400217H)

Actual setup:



GPIO:

IO signals	Raspberrypi	
	GPIO no	**Pin no**
RS	25	22
E	24	18
Data 0	2	3
Data 1	3	5
Data 2	14	8
Data 3	15	10
Data 4	23	16
Data 5	17	11
Data 6	18	12
Data 7	22	15

Feature:

- Clear Display
- Print on Line 1
- Print on Line 2
- Print with character offset
- Cursor ON
- Cursor OFF

Deepak Bharadwaj.R (2020H1400229H)
Shri Lakshmi.A (2020H1400217H)

Usage:

Kernel code:

1. Upload ***lcd_8bit*** folder into raspberrypi workspace.
2. In the terminal, navigate to the directory that contains ***lcd_8bit*** folder using ***cd ~/$(lcd_8bit)*** command.
3. The folder contains ***lcd_8bit.c*** file and ***Makefile***.
4. Make the driver using the command: ***sudo make all***
5. Load the compiled kernel module using command: ***sudo insmod lcd_8bit.ko***
6. Loaded module can be checked in kernel space using ***dmesg*** or in procfs using command: ***cat /proc/devices | grep lcd_8bit***
7. Enable permission of generated ***lcd_8bit.ko*** using command***: sudo chmod 666 /dev/lcd_8bit***
8. The driver can be tested in command prompt using command: ***echo "String of your choice" > /dev/lcd_8bit***
9. The module is unloaded using command: ***sudo rmmod lcd_8bit***
10. The module can be cleaned using command:

User code:

- Load the kernel code in raspberrypi workspace by following the above instructions till step 7.
- Navigate to user folder inside ***lcd_8bit*** folder using ***cd ~/$(user)*** command.
- The folder contains ***user.c*** file. Compile it using ***gcc user.c -o user.o***
- Run the compiled file using ***./user.o***
- The command space instructs the user with Input options and possible command options. Type in the values and observe the LCD Display.
- Run steps 9 and 10 for unloading the driver from kernel model that disables user application inherently.