

# Lab3

Negin Kheirmand

9831023

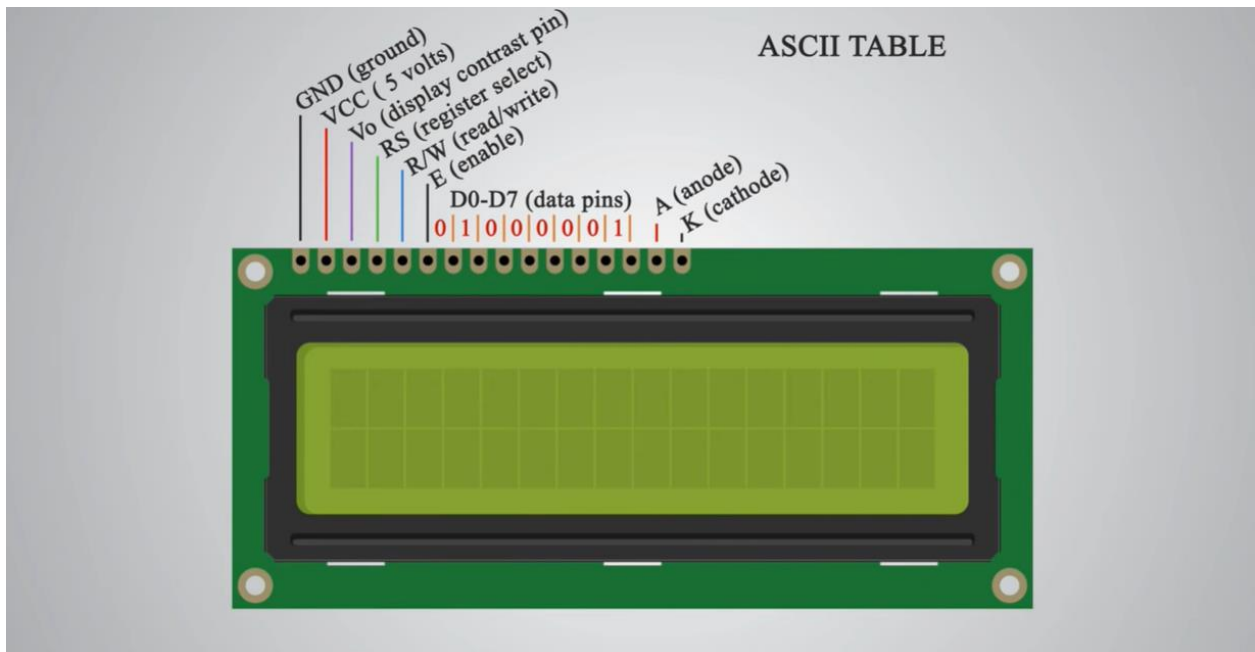
This week's lab is about working with LCD.

## IC & Modules I used:

- Arduino Nano
- 4\*4 Matris Keypad
- a 220  $\Omega$  resistors & a 100k potentiometer (& some jumper wires)
- LCD Display 20\*4 (CV4204)
- Keypad

Codes for the Lab:

1. Part1 : sketch\_RunningNegin directory
2. Part2 : sketch\_PassVerification directory
3. Part3 : sketch\_calculatino directory
4. Part4 : sketch\_MadChar directory



Disclaimer: the LCD I used had 4 rows and 20 columns (but since the pinout is the same as the picture above I used it)

## CONNECTOR PIN ASSIGNMENT

Pin No.	Symbol	Function	Pin No.	Symbol	Function
1	Vss	Ground	9	DB2	Data Bus
2	VDD	Supply voltage for logic	10	DB3	Data Bus
3	Vo	Input voltage for LCD	11	DB4	Data Bus
4	RS	Register Select Input	12	DB5	Data Bus
5	R/W	Read/Write	13	DB6	Data Bus
6	E	Enable Signal	14	DB7	Data Bus
7	DB0	Data Bus	*15	BL+	Supply voltage for Backlight (+VE)
8	DB1	Data Bus	*16	BL -	Supply voltage for Backlight (-VE)

**Note (\*) : Pin 15, 16 are used for backlight version**

My LCD also supported 4bit and 8bit data.

## GENERAL DESCRIPTION

Display mode : 20 characters x 4 lines LCD module

Interface : 4-bit or 8-bit parallel

Driving method : 1/16 duty, 1/5 bias

Controller IC : Sitronix ST7066U or equivalent

For the detailed information, please refer to the IC specifications

For more info about the display refer to CV4204\_LCD\_doc.pdf file.

---

*Experiment Reports:*

---

**First experiment is a simple sketch to show my name on the display and move it right-wards and at the end of the column shift it a row down.**



## What I learnt in this experiment:

- 4 row LCD displays (at least mines) have only 2 controllers, one of them controls the 1<sup>st</sup> and 3<sup>rd</sup> row and the other one controls the 2<sup>nd</sup> and 4<sup>th</sup> row, that's why if you try to write a word in the first row and it does not fit in a it, instead of wrapping it correctly and just writing the rest of it in the next row, the display will skip one row and display it in the 3<sup>rd</sup> row. Reference: <https://forum.arduino.cc/t/20x4-lcd-screen-only-prints-to-line-0-and-2/121300/3>
  - Quote:
  - "a 4-line LCD display is really two 2-line display controllers connected together. For some reason, the designers used one controller for lines 1 and 3, and the second controller for lines 2 and 4."
  - Also I didn't know how to name them in the code, this helped me: <https://forum.arduino.cc/t/analog-pins-for-keypad/134054>
- You can control the light of the backlight of the LCD display if you use a potentiometer and wire it to the 15<sup>th</sup> and 16<sup>th</sup> pin of the display(which are the anode and cathode supply voltage for the backlight).Which answers the question:
  - **Why potentiometer is used in LCD Arduino?**
    - On an LCD the potentiometer is used to adjust the bias level of the LCD - that is the contrast. You need to use it to set a voltage between Vcc and Vee, which you feed into Vo. That is, a voltage somewhere between +5V and -5V. You can't do that with one resistor.
- DON'T EVER USE FEMALE-TO-FEMALE JUMP WIRES TO CONNECT YOUR LCD DISPLAY TO YOUR ARDUINO BOARD:
  - This took me a lot of time to figure out, at first I kept getting weird characters that looked kind of similar to what I wanted the display to show but not completely. After a lot of searching I came across this: reference: <https://forum.arduino.cc/t/how-to-fix-all-lcd-problems-read-this/100051>

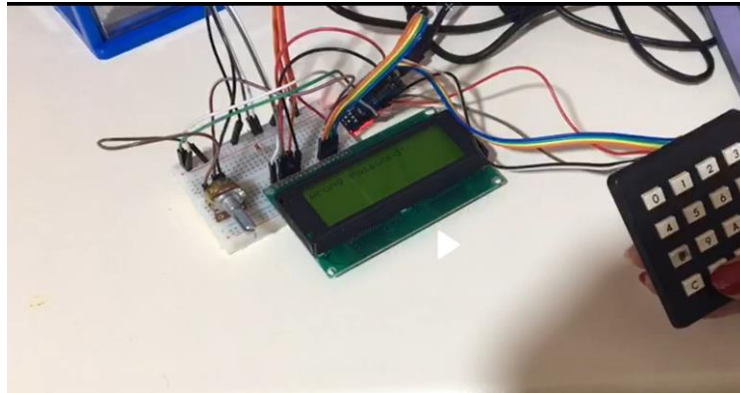
## Pinout:

- \* LCD pin 1(vss) to gnd
- \* LCD pin 2(vcc) to 220Ohm resistor (resistors other pin is connected to 5V)
- \* LCD pin 3(V0 -display contrast pin-) to gnd
- \* LCD pin 4(RS -register select-) to D6 (Arduino)
- \* LCD pin 5(R/w) to gnd
- \* LCD pin 6(Enable) to D7 (Arduino)
- \* LCD pin 7 (Data0) -> unused
- \* LCD pin 8 (Data1) -> unused
- \* LCD pin 9 (Data2) -> unused
- \* LCD pin 10(Data3) -> unused
- \* LCD pin 11(Data4) to D2 (Arduino)
- \* LCD pin 12(Data5) to D3 (Arduino)
- \* LCD pin 13(Data6) to D4 (Arduino)
- \* LCD pin 14(Data7) to D5 (Arduino)
- \* LCD pin 15(Anode) to 10K potentiometer's middle pin(output)->I actually used a 100k potentiometer and changed its value in range of 1k to 10k only (I didn't have a 10k pot)
- \* LCD pin 16(Cathode) to 5v

This is going to be pretty much the circuit used in all of the other experiments, so I will specify it just here.

---

Second experiment is a simple Lock Sketch that verifies the input numbers given by the keypad a



#### What I learnt in this experiment:

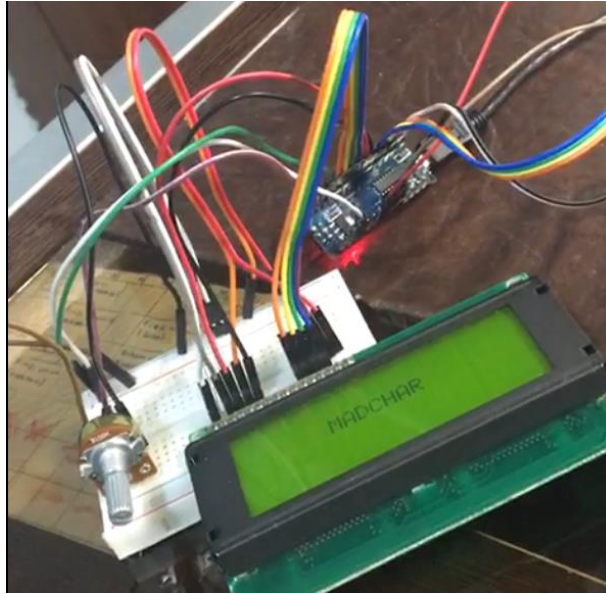
- You can use analog pins as digital:
  - Found out about this because I couldn't fit an 8 pin keypad and a 16 pin display LCD which needed at least 6 digital pins to work correctly. If I knew about this sooner, I wouldn't have used serial print in the last lab in order to replace the LED's I could not connect to the Arduino because I didn't have enough digital pins.
  - Reference: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>
  - Quote: **"The analog input pins can be used as digital pins, referred to as A0, A1, etc. The exception is the Arduino Nano, Pro Mini, and Mini's A6 and A7 pins, which can only be used as analog inputs."**
- There were also people who said the 0 and 1 pins also could not be used because of serial (which is used when printing ...) but I couldn't figure out if they meant digital or analog (though I suspect they meant analog so I did not use those in any of the experiments). Digital pin number 13 of the Arduino cannot be used to read data and should be only used as output. I passed a lot of time on this too, since I was trying my best to use every single one of the digital pins in the Arduino which were less than what I already needed for this lab, but no matter what I did it wouldn't work, turns out you can't use it for a keypad since the pins connected to keypad are input\_pullup mode.
  - Source: <https://forum.arduino.cc/t/unable-to-use-free-pins-with-keypad-library/211191>
  - Quote : **"The Arduino Nano has a led + resistor connected to pin 13, you get into trouble if you use that as input."**

---

Third experiment is a simple calculator Sketch that supports only one operator (or less)



**Forth experiment was a moving char in the display with a zig-zag like pattern**



### Questions:

#### ❖ Functions:

- LiquidCrystal()
  - Description: Creates a variable of type LiquidCrystal. The display can be controlled using 4 or 8 data lines. If the former, omit the pin numbers for d0 to d3 and leave those lines unconnected. The RW pin can be tied to ground instead of connected to a pin on the Arduino; if so, omit it from this function's parameters
- begin()
  - Description: Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands.
- clear()
  - Description: Clears the LCD screen and positions the cursor in the upper-left corner.
- setCursor()
  - Description: Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed.
- write()
  - Description: Write a character to the LCD.
- noDisplay()
  - Description: Turns off the LCD display, without losing the text currently shown on it.
- print()
  - Description: Prints text to the LCD.
- scrollDisplayLeft()
  - Description: Scrolls the contents of the display (text and cursor) one space to the left.
- autoscroll()
  - Description: Turns on automatic scrolling of the LCD. This causes each character output to the display to push previous characters over by one space. If the current text direction is left-to-right (the default), the display scrolls to the left; if the current direction is right-to-left, the display scrolls to the right. This has the effect of outputting each new character to the same location on the LCD.

1. Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
2. Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
3. Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V. (for backlight: these is one of the pins you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.)
4. Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
5. Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation). **Since in my experiments I only needed to write to the display it was wired to gnd.**
6. Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
7. Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3 –**which is the interface I used, I wired d4-d7 of the display to 4 of the digital pins in Arduino-**, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
8. Pin15 (+ve pin of the LED): This pin is connected to +5V. (for backlight: these is one of the pins you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.)
9. Pin 16 (-ve pin of the LED): This pin is connected to GND. (for backlight: these is one of the pins you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.)