Interface 2

EMBEDDED DIPLOMA

LCD (Liquid crystal display)

- Liquid crystal display, or LCD is a very commonly used device in electronics projects to display data and interact with users.
- LCDs contain characteristics, when electric current is applied to back ground. Liquid crystals do not emit light by them selves' special crystals, which change their optical them, this makes them visible, on a contrast like LEDs.
- Therefore you need light to see them, usually the surrounding light is enough to read the display, yet in case of dark environments it is hard to read the display.
- Most LCDs therefore contain an optional backlight, to produce sufficient contrast, which makes reading easy in dark environment

LCD Types

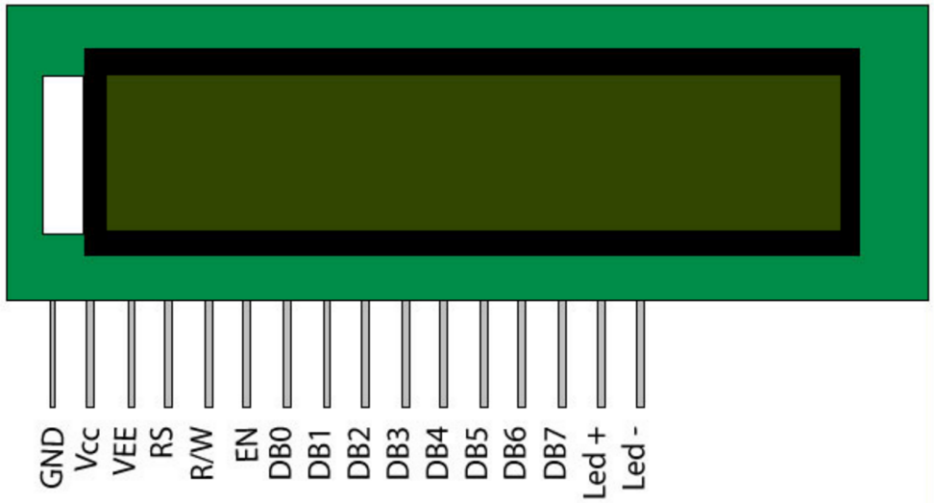
- There are two basic types of LCDs available, Character LCDs and Graphic LCDs:

1. Character LCD

2. Graphical LCD

Character LCD

- Character LCDs are manufactured by a number of manufacturers, in various sizes.
- The characteristics of LCD are defined by the number of text lines it has, and the number of characters per line.
- Thus a 20 x 4 character LCD would have four lines of text data, having 20 characters per line.
- 6 x 2 is the most convenient size and most commonly used in electronics projects.



Interfacing with Character LCD

- The character LCDs, contain onboard controller, with a connector to communicate with the parent microcontroller.
- There are usually 14 pins for communication and two pins for a backlight LED, if that is there.
- Thus a total of 16 pin connector is usually required.

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
(+)	(-)	D7	D6	D5	D4	D3	D2	D1	D0	E	RW	RS	VEE	VCC	GND

Interfacing with Character LCD

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Interfacing with Character LCD

- **LCD can work in 2 modes**
  - 4bit mode: only four data pin are connected to microcontroller
  - 8bit mode: the whole 8 data pin connected to microcontroller

This is a good reference for LCD : https://www.8051projects.net/lcd-interfacing/lcd-4-bit.php

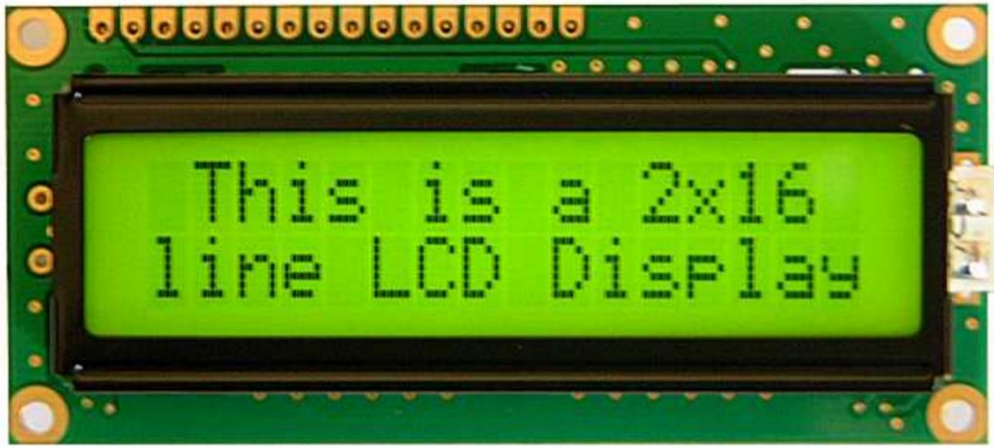
Also you can read this article about LCD: https://goo.gl/YwQ9lG

Interfacing with Character LCD

- We are going to learn how to use LCD in **4-bit mode**.
- There are many reasons why sometime we prefer to use LCD in **4-bit mode** instead of 8-bit.
- One basic reason is **lesser number** of pins are needed to interface LCD.
- In 4-bit mode the data is sent in **nibbles**, first we send the **higher** nibble and then the **lower** nibble.

Interfacing with Character LCD

- There is two mode for configuration :
  - Data mode: to display data on LCD
  - Command mode: to shift data, set a specific location, clear screen …etc



Interfacing with Character LCD

Below is a brief list of useful commands which are used frequently while working on the LCD.

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
11	Shift entire display right	0x1C	30
12	Move cursor left by one character	0x10	16
13	Move cursor right by one character	0x14	20
14	Clear Display (also clear DDRAM content)	0x01	1

Common command

- 0x80: to start writing from 1st line first place so 0x81 writing on 2nd place until 0x8f
- 0xc0: to start writing from 2nd line first place so 0xC1 writing on 2nd place until 0xcf
- 0x01: to clear the display
- 0x0E: to turn on the cursor
- 0x0C: to turn off the cursor

Interfacing with Character LCD

- To enable the 4-bit mode of LCD, we need to follow special sequence of initialization that tells the LCD controller that user has selected 4-bit mode of operation. Following is the reset sequence of LCD.

1. Configure all required pins connected to lcd as output (D7,D6,D5,D4, EN,RS,RW)
2. Put RW = 0 as we need to write only on LCD
3. Send the first init value (0x2) to start config i`n 4bit mode
4. Send the first init value (0x28) to work at 4bit mode
5. Send the first init value (0x01) to clear LCD
6. Send the first init value (0x0c) to turn cursor off
7. Send the first init value (0x06) to make curser increment to right
8. Send the first init value (0x80) Set the curser at the beginning of LCD display
9. Wait 20 mille second.

Interfacing with Character LCD

- To write a **command** for example 0x28

follow the below steps:

1. Put RS = 0 to write command
2. Put **highest** nibble of 0x28 on d7 d6 d5 d4 (2);
3. set **En = 1** then **wait 1 mille** second, set **En = 0**, then **wait 1 mille** second.
4. Put **lowest** nibble of 0x28 on d7 d6 d5 d4 (8);
5. set **En = 1** then **wait 1 mille** second, set **En = 0**, then **wait 1 mille** second.

Interfacing with Character LCD

- To write a **data** for example number 12

follow the below steps:

1. Put RS = 1 to write data
2. Convert 12 to ascii code “the result is 2 ascii number represent 12 which is 49 and 50”
3. Follow the same steps in the previous slide from number 2

Note:

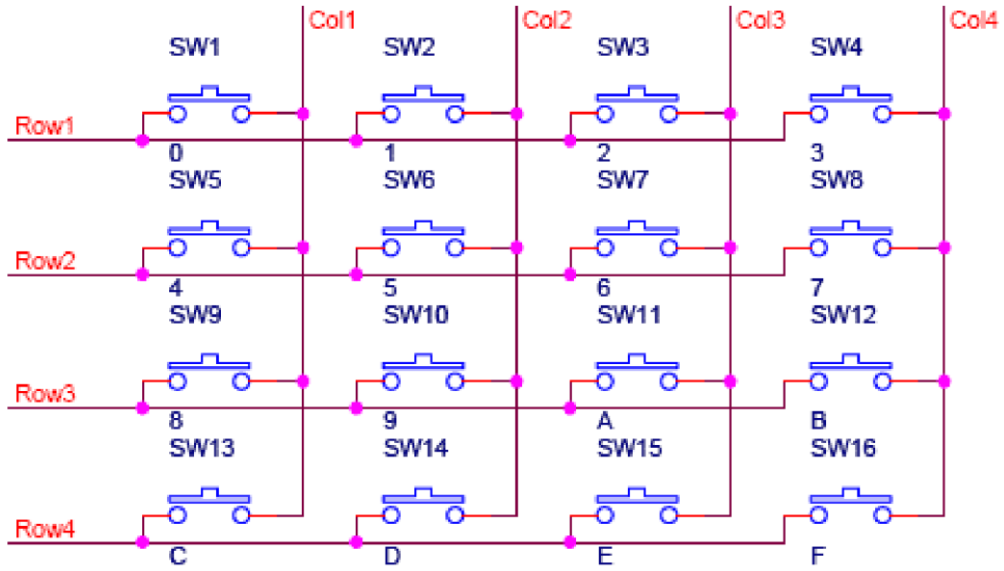
You now have 2 number 1 and 2 which are 49 and 50 then send both of them

Keypad

- There are many methods depending on how you connect your keypad with your controller, but the basic logic is same.
- A common technique: Set the columns as I/p and we drive the rows making them o/p, this whole procedure of reading the keyboard is called scanning.

Matrix Keypad Interfacing

- In order to detect which key is pressed from the matrix:
  - we make row lines low one by one and read the columns. Lets say we first make Row1 low,
  - then read the columns. If any of the key in row1 is pressed will make the corresponding column as low.
  - If second key is pressed in Row1, then column2 will give low. So we come to know that key 2 of Row1 is pressed. This is how scanning is done.



Working with keypad

1. Configure 4 pin as input and make them pulled up, 2. Configure 4 pin as output

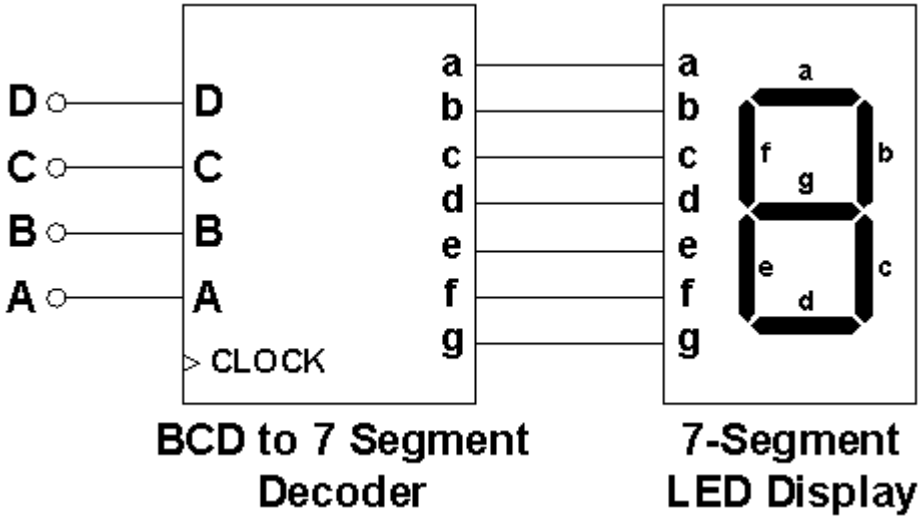
2. send 1110 to the 4 output pin and start to scan/check the value of the 4 input pin

3. if one of the 4 input pin is zero this mean you pressed on a keypad button, return the value of this pressed button

4. repeat step 3 and 4 but change the output to be 1101, 1011, 0111 and so on….

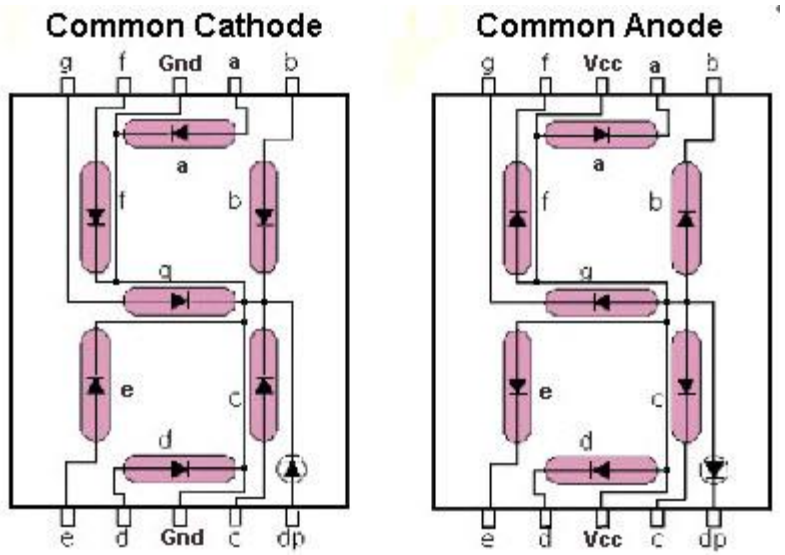
Seven segment

- Consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown.
- Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed.
- An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point (DP).



Seven segment

- The displays common pin is generally used to identify which type of 7-segment display it is.
- As each LED has two connecting pins, one called the “Anode” and the other called the “Cathode”, there are therefore two types of
- LED 7-segment display called:
  - Common Cathode (CC)
  - Common Anode(CA).



Working with Seven segment

- Multiplexing
  - Often we need to use two, three or more Seven Segment Displays and that too using only a single MCU, but one problem that we face is the lack of I/O pins in the MCU, as one Seven Segment Display would take 8 pins, and so three Seven Segment Displays would take 24 pins.
  - We mean by multiplexing of seven-segment display is that we will be using only 7 output ports to give the display on all of the SSDs
- So what is the solution?
  - One possibility is that we use a bigger MCU with more I/O pins.
  - Another much better and recommended solution to this problem is to multiplex the Seven Segment Displays

Working with Seven segment

1. Configure the 4 pins of the IC and En1,En2 as output.
2. To display a number on the first seven segment put En1 = 1, En2 = 0 and put the number on the 4 pins
3. To display a number on the second seven segment put En2 = 1, En1 = 0 and put the number on the 4 pins
4. To display a two digits number put the first digit on the 4 pin then En1=1, En2=0, wait 1 ms, then put second digit on the 4 pins and En1=0, En2=1 then wait 1ms and put this steps in a loop to be repeated.

Contact Me

Email: sameh.m.afifi@gmail.com

Facebook: https://www.facebook.com/Same7Afifi

linkedIn: https://www.linkedin.com/in/sameh-afifi-8389173a/

Phone: 01127346781

Work phone: 224135537