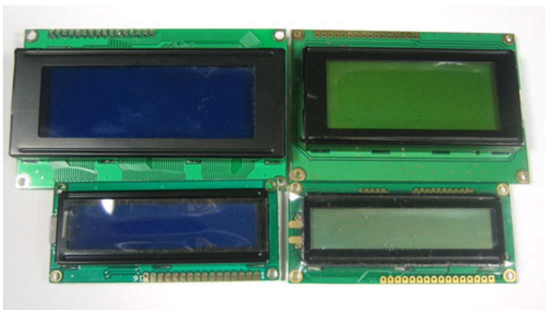
Experiment -8: LCD Display

- Type of LCD display and interfacing mode
- Connector details for LCD on the kit
- Port configuration with for LCD
- LCD code table
- LCD Library routines
    - lcd_init()
    - lcd_comm()
        - write_cn()
    - lcd_data()
        - write_dn()
- command and data variables – temp1 and temp2
- Sequence of lcd library routines
- example for initializing, passing command and data to the LCD using library routines
- Typical list of LCD commands
- Hands On
    - Display a character on LCD using the library routines

Advantages of LCD

- Low power consumption since voltage controlled
- Easy to read in bright light
- Declining cost
- Ability to display alpha numeric’s, special characters, graphics
- Available in different modes and types
- Intelligent controller and LCD display panels

Different types of LCD



- Although they display only text, they do some in many shapes: from top left we have a 20x4 with white text on blue background, a 16x4 with black text on green, 16x2 with white text on blue and a 16x1 with black text on gray.

- The good news is that all of these displays are 'swappable' - if you build your project with one you can unplug it and use another size.

- Your code may have to adjust to the larger size but at least the wiring is the same!

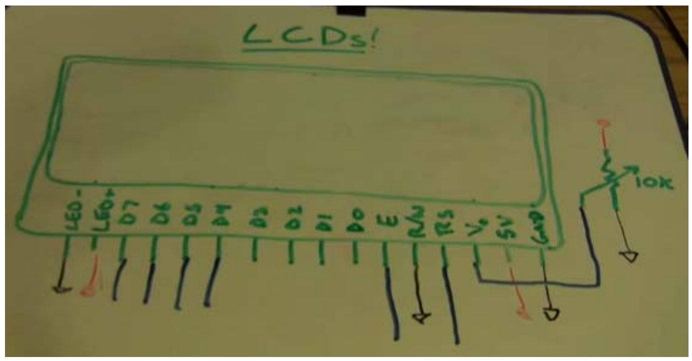
- Easy to program

- 2 rows with 20/40 chars in each row

- Each char is a 5x8 or 5x11 matrix

LCD driver applications

LCD pin details



Get Ready with LCD

- Vcc and Gnd for power to the LCD panel
- LED+ and LED- through series resistor for backlight to LCD panel
- V0 connected to wiper part of 10K Potentiometer to control the contrast of the display
  - On power
    - Power supply logic
    - Backlight light up
    - Vary contrast to see first line of rectangular

Bus Wiring

- There are 11 bus lines: **D0** through **D7** (8 data lines) and **RS**, **EN**, and **RW**.
- D0-D7 is the pins that have the raw data we send to the display.
  - Use in 8b mode for speedy operation
  - Use in 4b mode to save pins and less speed
- The **RS** pin lets the microcontroller tell the LCD whether it wants to display that data (as in, an ASCII character) or whether it is a command byte (like, change position of the cursor).
- The **EN** pin is the 'enable' line we use this to tell the LCD when data is ready for reading.
- The **RW** pin is used to set the direction - whether we want to write to the display (common) or read from it (less common)

Alphanumeric 16x2 LCD display
LCD Interface:

A 16X2 Alphanumeric LCD Display with back light is provided along with the ALSEMB-EVAL-03 board. The LCD interfaced using 4 – bit mode. The connector details of LCD are as shown in the table below

Connector Pin No.	Signal Name	Port line
1	GND	GND (Ground)
2	VCC (+5V)	VCC (+5V)
3	VEE (Preset For LCD Contrast)	**R9** (10K Potentiometer)
4	RS (Resistor Select) Line	P2.7
5	R/W (Read/Write)	P2.5
6	EN (LCD Enable) Line	P2.6
7	Data D0	No connection
8	Data D1	No connection
9	Data D2	No connection
10	Data D3	No connection
11	Data D4	P2.0
12	Data D5	P2.1
13	Data D6	P2.2
14	Data D7	P2.3
15	VCC (+5V)	P2.4 through **R11**
16	No connection	No connection

RS = 0 for sending Command to the LCD

RS = 1 for sending Data to the LCD

R/W = 0 for reading from the LCD

R/W = 1 for writing to the LCD

EN = 0 for disabling the LCD

EN = 1 for enabling the LCD

To display any character on LCD micro controller has to send its ASCII value to the data bus of LCD.

For e.g. to display 'AB' microcontroller has to send two hex bytes 41h and 42h respectively.

LCD display used here is having 16x2 sizes.

It means 2 lines each with 16 characters

LCD initialization

- **Major task in LCD interfacing is the initialization sequence.**
  - In LCD initialization you have to send command bytes to LCD.
  - Here you set the
    - interface mode,
    - display mode,
    - address counter increment direction,
    - set contrast of LCD,
    - horizontal or vertical addressing mode,
    - Color format.

  - This sequence is given in respective LCD driver datasheet.

  - Studying the function set of LCD lets you know the definition of command bytes. It varies from one LCD to another.

  - If you are able to initialize the LCD properly 90% of your job is done.

- **Sending data bytes to DD RAM of LCD**

  - Next step after initialization is to send data bytes to required display data RAM memory location.

  - Firstly set the address location using address set command byte and then send data bytes using the DDRAM write command.

  - To address specific location in display data RAM one must have the knowledge of how the address counter is incremented.

- **Mapping LCD pins with Port**

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
RS	EN	R/W	Vcc (+5V)	D7	D6	D5	D4

LCD Code Table

```
//LCD code table for displaying strings
unsigned char arr1[17] = {"<string to display in first line of length 16 chars>"};
unsigned char arr2[17] = {"<string to display in second line of length 16 chars>"};

(or)

//LCD Code table for a key press
unsigned char LCD_CODE[16] = {'0','4','8','C',
                              '1','5','9','D',
                              '2','6','A','E',
                              '3','7','B','F'       };
```

LCD Library Routines

lcd_init()

Form the 8b control word to write sequence of control bytes in to the LCD data bus

Ex:

temp1 = 0x28; // Set retrun home (2) and display off and cursor off
lcd_comm();
delay(500);

lcd_comm()

Split the command byte in to upper and lower nibbles and pass it to *wr_cn()*

wr_cn()

Form the 8b control word to write the command byte in to the LCD data bus

lcd_data()

Split the data byte into upper and lower nibbles and pass it to *wr_dn()*

wr_dn()

Form the 8b control word to write the data byte in to the LCD data bus

Command and Data variables used in LCD library

- *'temp1'* for passing command to the LCD command register via lcd_comm() API
- *'temp2'* for passing data to the LCD data register via lcd_data() API

Example for initializing the LCD (RS=0, EN=1, R/W=0, Vcc=1)

```
lcd_init();         //call to LCD library routine for initialization
```

Example for passing command to LCD (RS=0, EN=1, R/W=0, Vcc=1)

```
temp1 = 0x28;       //temp1 variable for passing command byte to LCD
                    //internally LCD works in 4b mode
                    // upper nibble 2 – command for return home
                    // lower nibble 8 - command for display off, cursor off

lcd_comm();         //call to LCD library routine to form the control word and pass the command
```

Example for passing data to LCD (RS=1, EN=1, R/W=0,Vcc=1)

```
temp2 = 0x41        //temp2 variable for passing data byte to LCD
                    //internally LCD works in 4b mode
                    //byte 0x41 is to display 'A' on LCD

lcd_data();         //call to LCD library routine to form the control word and pass the data
```

Commands of LCD

1	-	clear display screen
2	-	return home
4	-	decrement counter
6	-	increment counter
5	-	shift display right
7	-	shift display left
8	-	display off, cursor off
A	-	display off, cursor on
C	-	display on, cursor off
E	-	display on, cursor blinking
F	-	display on, cursor blinking
10	-	shift cursor position to left
14	-	shift cursor position to right
18	-	shift the entire display to left
1C	-	shift the entire display to right
80	-	force cursor to beginning of first line
C0	-	force cursor to beginning of second line
38	-	2 lines and 5x7 matrixes

//Pseudo Code to display string or key pressed
//include the lcd library file

//LCD code table for displaying strings – init array to display the string
Or
//LCD code table for key press – ascii code for each key to be displayed (chars in single quotes)

//init the LCD – lcd_init() api call

// Display the data from first position of first line – temp1=0x80
// call the api to write command into LCD– lcd_comm()

//loop through the array of chars to be displayed or detect any key press
//assign the ascii value of char to be displayed to data variable
//call the api to write data into LCD – lcd_data()

LCD

Types of LCDs

Pin out Diagram

Control line description

EN (450 ns) / RW / RS / Contrast (1.8 – 3.6V) / Backlight

LCD Busy concept – Delay of 10ms

Initialize LCD with command bytes

LCD function set – command options

Clear display – 01

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Function
0	0	0	0	0	0	0	0	0	1	Clear LCD and memory, home cursor

Display and cursor home – 02 or 03

0 0 0 0 0 0 0 0 1 x Clear and home cursor only

Character entry mode – 1 I/D S – 04 to 07 - Increment or decrement / Display shift on of off

0 0 0 0 0 0 0 1 1/0(I/D) S Screen action as display character written

S=1/0: Shift screen/cursor

I/O=1/0: cursor R/L, screen L/R

Display on or off and cursor – 1 D U B – 08 to 0F

Display on or off

Cursor underlines or off

Cursor blinks or off

0 0 0 0 0 0 1 D CU B D=1/0:Screen on/off

C=1/0:Cursor on/off

B=1/0:Cursor blink/no blink

Display / cursor shift – 1 D/C R/L x x – 10 to 1F

Display shfit or cursor move

Right or left shift

0 0 0 0 0 1 S/C R/L x x S/C: 1/0:screen/Cursor

R/L: Shift one space R/L

Function set – 1 8/4 2/1 10/7 – 20 to 3F

8b or 4b interface (set interface length)

2 line or 1 line mode

5x10 or 5x7 dot format for each character

0 0 1 (8/4)DL (2/1)N (10/7)F 0 0 DL=1/0:8/4 Bits per Character

N=1/0; 2/1 Rows of Characters

F=1/0;5*10/5*7Dots/Character

Set CGRAM Address – 0 1 A A A A A A – 40 to 7F

0 1 [A] Character address Write to character RAM address after this

Set display address – 1 A A A A A A A – 80 to FF

1 [A] Display data address Write to display RAM address after this

0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

0 1 BF Current address BF=1/0:busy/Notbusy

1 0 Character type Write byte to last RAM chosen

1 1 Character type Read byte from last RAM chosen

LCD initialization command bytes (3 bytes)

4/8 bit mode +1/2 line mode + 5x7 or 5x10 dot format

Toggles display + cursor and cursor blink on or off

Display shift + cursor increment mode

Clearing the display and cursor home

01h – command byte

Writing text

Send ascii bytes of characters to be displayed

RS = 1 to display text

00-0F – UDF characters

10-1F and 80-9F – blank characters

20-7F – proper ascii bytes to display

A0 – DF and E0 – FF – display Japanese and greek characters

Memory mapping

LCD can display only 16 locations at a time (00-0F) and second line from 40H

Cursor positioning

First line address stars from 00

Second line address starts from 40

Set cursor position command byte => 80

So first line start address => 80 + 00 = 80

Second line start address => 40 + 80 = C0

Ex: 10th location of second line=> 4A + 80 = CA

Display and cursor shifting

10H command byte

18H – to shift whole display to left by one character

Starts showing hidden locations in the left

10H – to shift the cursor position to the right by one position

Cursor at the right most position

No longer visible after the instruction

Clear – 01H – brings display and cursor back to home

User Defined Graphics

40H – CGRAM mode

Character generator internal RAM of LCD – upto 8 UDF characters storage

After 40H – all the data values stored in CGRAM unless come out of CGRAM mode

Data value – 8b – upper 3b discarded and lower 5b taken for pixel values for each row

7 rows makes one character (5x7 dot format)

More data will be taken to next CGRAM address

Access each character in CGRAM mode by 00-07 or 08-0F since CGRAM is volatile

Ex: a man may be drawn by => 40 (CGRAM mode) and 0E, 11, 0E, 04, 1F, 04, 0A, 11 (pixel values for 7 rows)

Points to remember

- EN set and cleared before and after sending every byte
- Clear RS for command and set for display text
- Init LCD before using any purpose
- Delay after each instruction for atleast 10ms
- Adjust contrast using POT after power up
- Character written after last char of first line – will not auto appear on second line –need to move cursor to second line
- Change LCD if nothing works
- Display shifting – send home command after shift operation – only then memory address will be valid

Liquid Crystal Display

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sand witched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

One each polarisers are pasted outside the two glass panels. These polarisers would rotate the light rays passing through them to a definite angle, in a particular direction

When the LCD is in the off state, light rays are rotated by the two polarisers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarisers, which would result in activating / highlighting the desired characters.

The LCD’s are lightweight with only a few millimeters thickness. Since the LCD’s consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

The LCD s doesn’t generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD’s have long life and a wide operating temperature range.

Changing the display size or the layout size is relatively simple which makes the LCD’s more customer friendly.

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics. The LCDs have even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

This section describes the operation modes of LCD’s then describe how to program and interface an LCD to 8051 using Assembly and C.

LCD operation:

In recent years the LCD is finding widespread use replacing LED s (seven-segment LED s or other multi-segment LED s).This is due to the following reasons:

1. The declining prices of LCDs.
2. The ability to display numbers, characters and graphics. This is in contrast to LED which is limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, there by relieving the CPU of the task of refreshing the LCD. In the case of LED s, they must be refreshed by the CPU to keep on displaying the data.
4. Ease of programming for characters and graphics.

LCD pin description:

The LCD discussed in this section has 14 pins. The function of each pin is given in table.

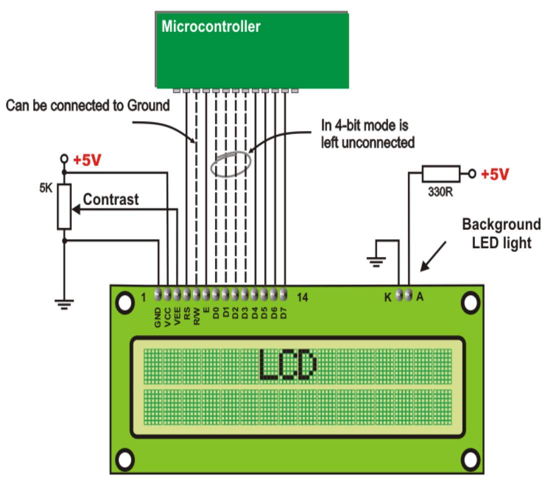


TABLE 1: Pin description for LCD

Pin	symbol	I/O	Description
1	Vss	--	Ground
2	Vcc	--	+5V power supply
3	VEE	--	Power supply to control contrast
4	RS	I	RS=0 to select command register RS=1 to select data register
5	R/W	I	R/W=0 for write R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

The LCD can display a character successfully by placing the

1. Data in Data Register
2. Command in Command Register of LCD

1. Data corresponds to the ASCII value of the character to be printed. This can be done by placing the ASCII value on the LCD Data lines and selecting the Data Register of the LCD by selecting the RS (Register Select) pin.
2. Each and every display location is accessed and controlled by placing respective command on the data lines and selecting the Command Register of LCD by selecting the (Register Select) RS pin.

The commonly used commands are shown below with their operations.

TABLE 2: LCD Command Codes

Code (hex)	**Command to LCD Instruction Register**
1	Clear display screen
2	Return home
4	Decrement cursor
6	Increment cursor
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor on
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Uses:

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data.

The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range.

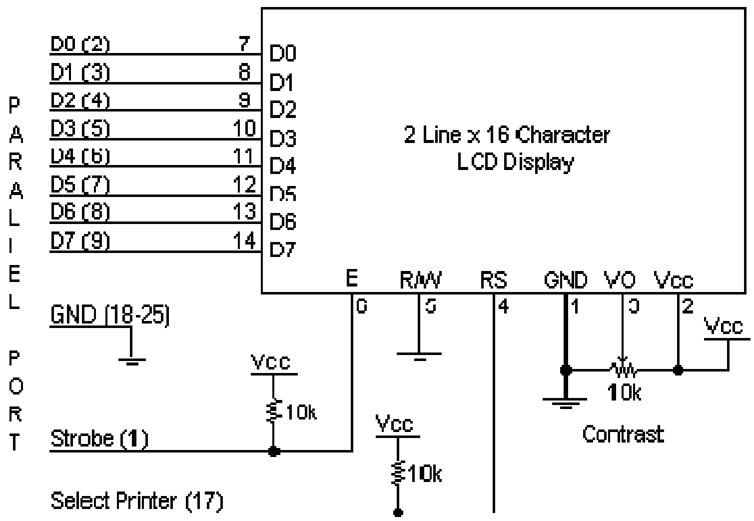
These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics.

LCD (LIQUID CRYSTAL DISPLAY)

Description:

LCD’s can add a lot to your application in terms of providing a useful interface for the user, debugging an application or just giving it a "professional" look. The most common type of LCD controller is the Hitachi 44780, which provides a relatively simple interface between a processor and an LCD. Using this interface is often not attempted by inexperienced designers and programmers because it is difficult to find good documentation on the interface, initializing the interface can be a problem and the displays themselves are expensive.

PIN DIAGRAM:



PIN DESCRIPTION:

Pin	Symbol	I/O	Description
1	GND	-	Ground
2	Vcc	-	+5V power supply
3	VEE	-	Contrast control
4	RS	I	command/data register selection
5	R/W	I	write/read selection
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

As you would probably guess from this description, the interface is a parallel bus, allowing simple and fast reading/writing of data to and from the LCD.

Above is the quite simple schematic pin diagram.

The LCD panel's Enable and Register Select is connected to the Control Port.

The Control Port is an open collector / open drain output.

While most Parallel Ports have internal pull-up resistors, there is a few which don't.

Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction.
Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag, which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here.

You can use a bench power supply set to 5v or use a onboard +5 regulator.

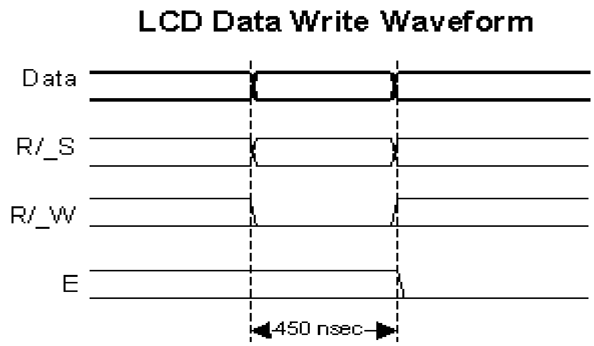
The 2 line x 16 character LCD modules are available from a wide range of manufacturers and should all be compatible with the HD44780.

The diagram to the right shows the pin numbers for these devices. When viewed from the front, the left pin is pin 14 and the right pin is pin 1.

LCDs can be added quite easily to an application and use as few as three digital output pins for control.

As for cost, LCDs can be often pulled out of old devices or found in surplus stores for less than a dollar.

The most common connector used for the 44780-based LCDs is 14 pins in a row, with pin centers 0.100" apart.



As you would probably guess from this description, the interface is a parallel bus, allowing simple and fast reading/writing of data to and from the LCD.

This waveform will write an ASCII Byte out to the LCD's screen.

The ASCII code to be displayed is eight bits long and is sent to the LCD either four or eight bits at a time.

If four-bit mode is used, two "nibbles" of data (Sent high four bits and then low four bits with an "E" Clock pulse with each nibble) are sent to make up a full eight-bit transfer. The "E" Clock is used to initiate the data transfer within the LCD.

Sending parallel data, as either four or eight bits are the two primary modes of operation.

While there are secondary considerations and modes, deciding how to send the data to the LCD is most critical decision to be made for an LCD interface application.

Eight-bit mode is best used when speed is required in an application and at least ten I/O pins are available.

Four-bit mode requires a minimum of six bits.

To wire a microcontroller to an LCD in four-bit mode, just the top four bits (DB4-7) are written to.

The "R/S" bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD. If the Bit is set, then the byte at the current LCD "Cursor" Position can be read or written. When the Bit is reset, either an instruction is being sent to the LCD or the execution status of the last instruction is read back (whether or not it has completed).

The different instructions available for use with the 44780 are shown in the table below:

R/S	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Instruction/Description
4	5	14	13	12	11	10	9	8	7	Pins
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	0	1	*	Return Cursor and LCD to Home Position
0	0	0	0	0	0	0	1	ID	S	Set Cursor Move Direction
0	0	0	0	0	0	1	D	C	B	Enable Display/Cursor
0	0	0	0	0	1	SC	RL	*	*	Move Cursor/Shift Display
0	0	0	0	1	DL	N	F	*	*	Set Interface Length
0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

"*" - Not Used/ Ignored. This bit can be either "1" or "0"

Before you can send commands or data to the LCD module, the Module must be initialized. For eight-bit mode, this is done using the following series of operations:

1. Wait more than 15 msecs after power is applied.
2. Write 0x030 to LCD and wait 5 msecs for the instruction to complete
3. Write 0x030 to LCD and wait 160 µsecs for instruction to complete
4. Write 0x030 AGAIN to LCD and wait 160 µsecs or Poll the Busy Flag
5. Set the Operating Characteristics of the LCD
    - Write "Set Interface Length"
    - Write 0x010 to turn off the Display
    - Write 0x001 to Clear the Display
    - Write "Set Cursor Move Direction" Setting Cursor Behavior Bits
    - Write "Enable Display/Cursor" & enable Display and Optional Cursor

Once the initialization is complete, the LCD can be written to with data or instructions as required. Each character to display is written like the control bytes, except that the "R/S" line is set. During initialization, by setting the "S/C" bit during the "Move Cursor/Shift Display" command, after each character is sent to the LCD, the cursor built into the LCD will increment to the next position (either right or left). Normally, the "S/C" bit is set (equal to "1") along with the "R/L" bit in the "Move Cursor/Shift Display" command for characters to be written from left to right (as with a "Teletype" video display).

The "Ninth Character" is the position of the Ninth character on the first line.

Most LCD displays have a 44780 and support chip to control the operation of the LCD. The 44780 is responsible for the external interface and provides sufficient control lines for sixteen characters on the LCD. The support chip enhances the I/O of the 44780 to support up to 128 characters on an LCD. From the table above, it should be noted that the first two entries ("8x1", "16x1") only have the 44780 and not the support chip. This is why the ninth character in the 16x1 does not "appear" at address 8 and shows up at the address that is common for a two line LCD.

LIQUID CRYSTAL DISPLAY(LCD):

2x16 lcd

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Here we are using a 2x16 LCD i.e 2 rows and 16 columns

FEATURES:

•Display construction……………16 Characters * 2 Lines

•Backlight……………………………LED(B/5.0V)

•Viewing direction…………………6 o’clock

•Operating temperature…………… Indoor

•Driving voltage…………………… Single power

•Driving method……………………1/16 duty,1/5 bias

•Type………………………………… COB (Chip On Board)

•Number of data line………………8-bit parallel

INTERNAL STRUCTURE:

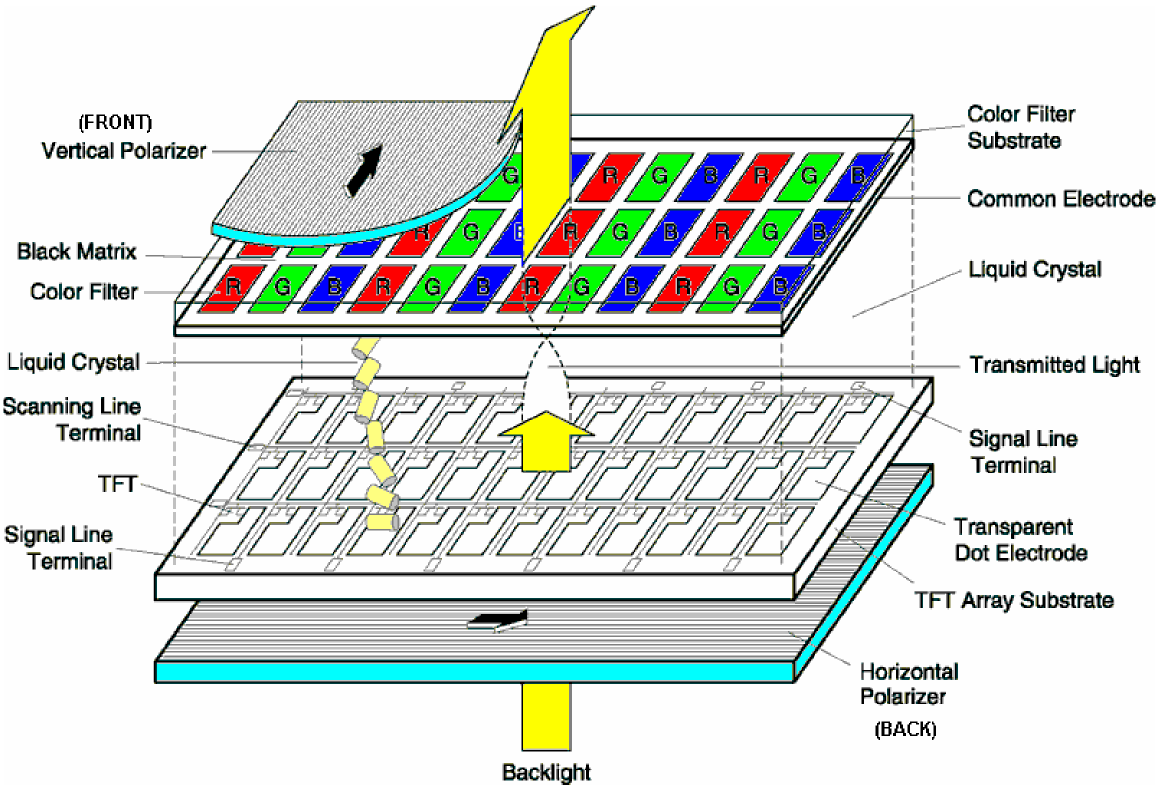
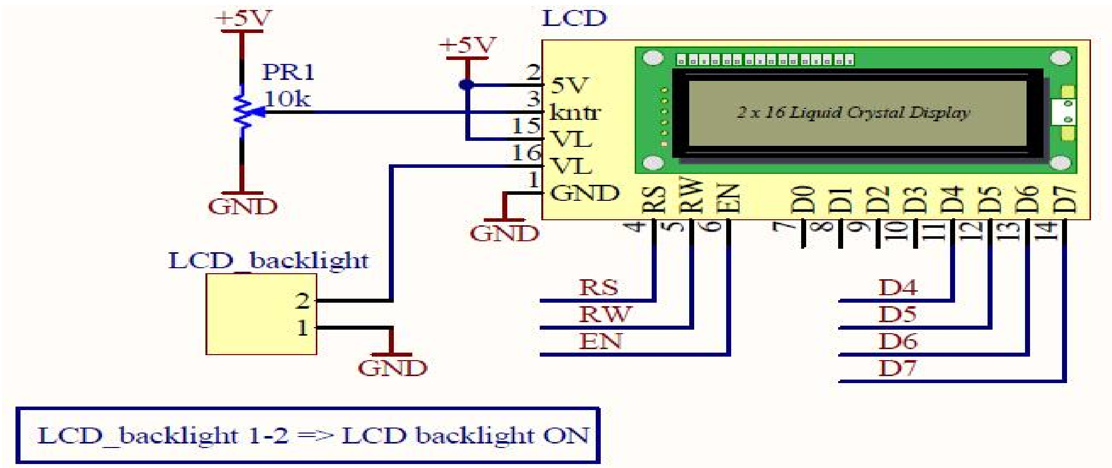


Fig 18: Internal structure of LCD

LCDs are more energy efficient and offer safer disposal than CRTs. Its low electrical power consumption enables it to be used in battery-poweredelectronic equipment. It is an electronically-modulated optical device made up of any number of pixels filled with liquid crystals and arrayed in front of alight source (backlight) or reflector to produce images in colour or monochrome. The earliest discovery leading to the development of LCD technology, the discovery of liquid crystals, dates from 1888.[1] By 2008, worldwide sales of televisions with LCD screens had surpassed the sale of CRT units.

PIN DESCRIPTION:



lcd internal connections

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Pin Description of lcd

LCD COMMANDS DESCRIPTION

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Function
0	0	0	0	0	0	0	0	0	1	Clear LCD and memory, home cursor
0	0	0	0	0	0	0	0	1	0	Clear and home cursor only
0	0	0	0	0	0	0	1	1/0	S	Screen action as display character written S=1/0:Shift screen/cursor I/O=1/0:cursor R/L, screen L/R
0	0	0	0	0	0	1	D	C	B	D=1/0:Screen on/off C=1/0:Cursor on/off B=1/0:Cursor blink/no blink
0	0	0	0	0	1	S/C	R/L	0	0	S/C: 1/0:screen/Cursor R/L: Shift one space R/L
0	0	0	0	1	DL	N	F	0	0	DL=1/0:8/4 Bits per Character N=1/0; 2/1 Rows of Characters F=1/0;5*10/5*7Dots/Character
0	0	0	1	Character address						Write to character RAM address after this
0	0	1		Display data address						Write to display RAM address after this
0	1	BF		Current address						BF=1/0:busy/Notbusy
1	0			Character type						Write byte to last RAM chosen
1	1			Character type						Read byte from last RAM chosen

Design Considerations:

Display Format : 16 characters (W) x 2 lines (H)

General Dimensions : 80.0 mm (W) x 36.0 mm (H) x 9.5 mm (T)

Character Size : 2.95 mm (W) x 4.35 mm (H)

Character Pitch : 3.65 mm (W) x 5.05 mm (H)

Viewing Area : 64.0 mm (W) x 13.8 mm (H)

Dot Size : 0.55 mm (W) x 0.50 mm (H)

Dot Pitch : 0.60 mm (W) x 0.55 mm (H)

Display Type : Positive or Negative

LC Fluid : STN Yellow-Green

Backlight LED : Optional

Polarizer Mode : Reflective

View Angle : 6 o’clock or 12 o’clock

Controller : S6A0069 orEquivalent

Temperature Range : 0oC to 50oC (Operating); -20oC to 70oC (Storage)

Circuit Diagram for LCD interfacing with AT89s52:

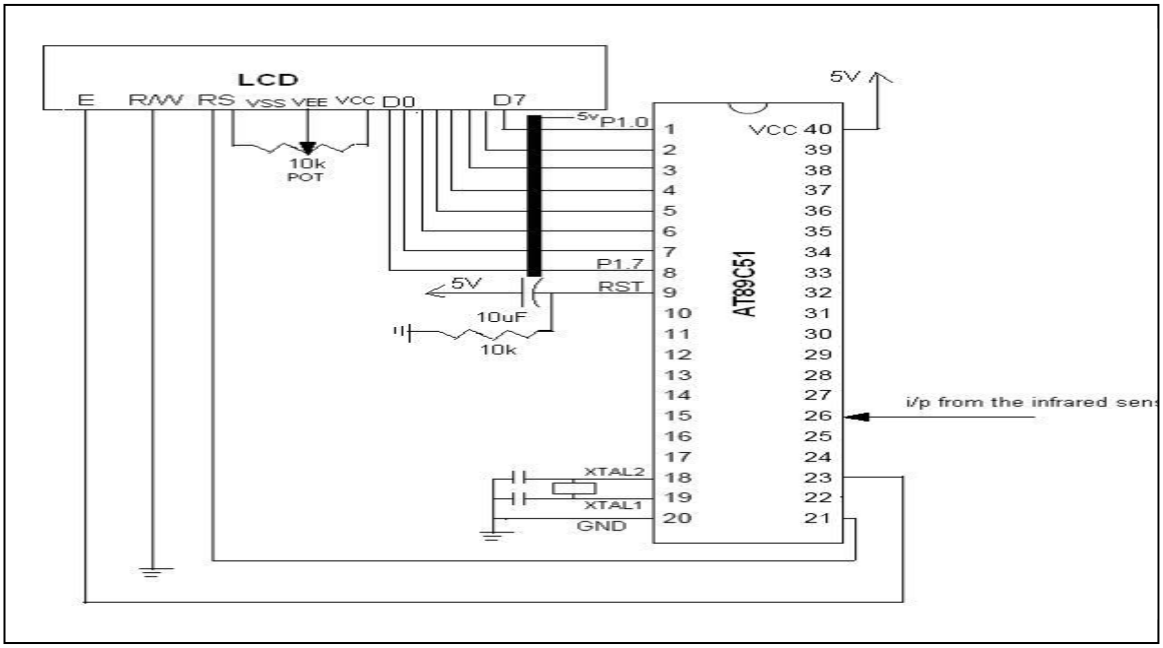


Fig20: Circuit Diagram for LCD interfacing with AT89s52

LCD is used in this project for the purpose of displaying messages. LCD interfacing consists of several parts like AT89C451 microprocessor, 2 × 16 line LCD are main components needed. AT89C51 is a 40 pin DIP micro processor.

LCD is a 2 line 16 pin device, 2 lines means it contains 2 rows to display.To develop a protocol to interface this LCD with 89S52 first we have to understandhow they functions. These displays contain two internal byte-wide registers, one for command and second for characters to be displayed.

There are three control signals called R/W, DI/RS and En. Select By making RS/DI signal 0 you can send different commands to display. These commands are used to initialize LCD, to display pattern, to shift cursor or screen etc. AT89S52 can be divided in to 4 ports, and each port consists of 8 pins. All the data lines of LCD are connected with port P1. i.e., data lines D0-D1 are connected to port P1 i.e., to pin numbers 1 to 8 through a SIL, SIL is a few ohms of resistance connected to withstand the large voltages and currents.'EN' pin is connected with P2.0, 'DI' (RS) is connected with P2.1 and R/W pin is connected with P2.2. i.e., the three pins are connected to the port two. The operation ofLCD depends upon these three pin only. For the pins 18 and 19 a crystal oscillator circuit is connected to generate clock signals to the micro processor to enable its pins. And 20th pin is grounded with oscillator

DESCRIPTION

EN:

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS:

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which sould be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW:

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command

Vcc, Vss, V_{EE} :

While Vcc and Vss provide +5v and ground, respectively, Vee is used for controlling LCD contrast.

D0 - D7:

The 8 - bit data pins, D0 - D7, are used to send information to the LCD or read the contents of the LCD's internal registers.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character