**Topic:** To interface LCD

**Pre-requisite knowledge:** Basic IO programming

**Components required:** Firebird V with 8051 adapter board

**Basic concept:**

* **What is LCD?**

Full form of LCD is Liquid Crystal Display. LCD screen is an electronic display module.

We are going to use 16x2 LCD screen.

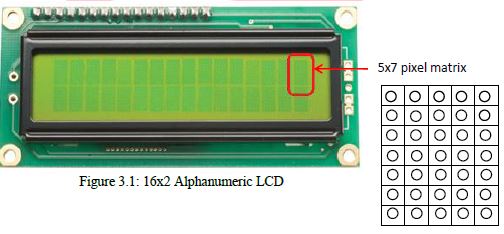
* **What is 16x2 LCD?**

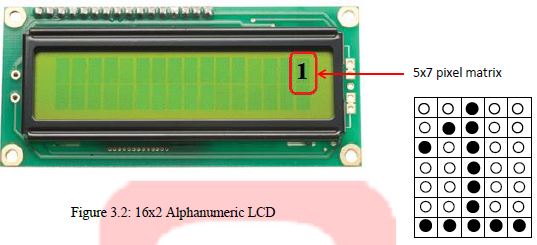
A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix as shown in figure 3.1 and 3.2.

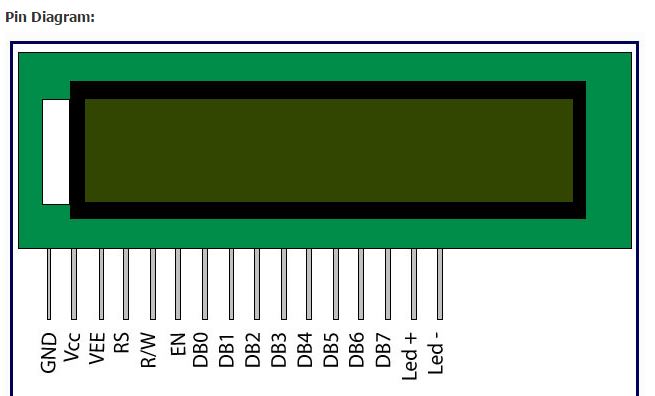


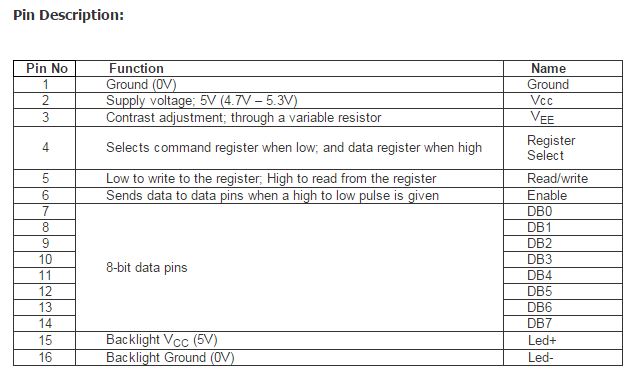
This LCD has two registers, namely, **Command and Data**.

* The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.
* The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.









**Interfacing LCD:**

To interface LCD there can be 2 modes –

1. **8 bit interfacing mode** which requires 8 data lines and 3 control signals and in total of 11 I/O pins.
2. **4 bit interfacing mode** which requires 4 data lines and 3 control signals and in total of 7 I/O pins.

Fire Bird V P89V51RD2 uses the 4-bit interfacing mode to reduce number of I/O lines. Figure 3.3 shows basic LCD interfacing.



**Figure 3.3: LCD interfacing with the microcontroller**

**Control lines:**

LCD requires 3 control lines known as Enable (EN), Register Select (RS) and Read /

Write (R/W). These pins are used for controlling the way LCD works:

* **Register Select:**
* Register select control line is connected to P2.6.
* There are two very important registers inside the LCD.
* If RS =0, the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home, etc.
* If RS = 1, the data register is selected, allowing the user to send data to be displayed on the LCD.
* **Read/Write Select:**
* Read/Write control line is connected to P2.5. This pin is used to select whether we want to write on the LCD or we want to read from it.
* If RW is set a value of 0, it means we are writing to the LCD
* If RW is set a value of 1, it means we are reading from the LCD.
* **Enable pin:**
* The enable pin is used by the LCD to latch information presented to its data pins.
* When data is send to data pins, a high to low pulse must be applied to this pin in order for the LCD to latch the data present in the data pins.

**Data lines:**

Data lines are used to pass the data to LCD.

* There are total eight data pins named from D0 to D7
* Same eight data pins are used to send both, the **command** and the **data**.
* LCD is alpha numeric type and understands only ASCII format.
* In 8-bit mode, all eight data pins are connected with the microcontroller whereas in 4-bit mode only four data pins are connected to the microcontroller.
* In firebird robot, LCD is interfaced in four-bit mode. So Data pin D4 to D7 is only used for interfacing. In this mode higher nibble and lower nibble of commands/data set needs to be sent separately.
* Busy flag:

To know the status of LCD, busy flag is used. The busy flag is D7 and can be read when R/W=1 and RS=0, as follows:

If R/W=1, RS=0. When D7=1(busy flag = 1), the LCD is busy taking care of internal operations and will not accept any new information.

If R/W=1, RS=0. When D7=0 (busy flag = 0), the LCD is ready to receive new information.

**Communicating with the LCD:**

Communication is done in the following sequence of steps

1. Select the register: Command register (RS = 1) or Data register (RS = 0)

2. Select read/write mode: Read (RW = 1) or Write (RW = 0)

5. Send or Receive byte from LCD.

4. Enable the LCD: (E = 1)

5. Disable the LCD: (E = 0)

Important commands which is used for working with LCD:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **No.** |  |  |  | | Instruction | Hex | Decimal |
| 1. | |  | | --- | | Function Set: 8-bit, 1 Line, 5x7 Dots | | 0x30 | 48 |
| 2. | |  | | --- | | Function Set: 8-bit, 1 Line, 5x7 Dots | | 0x38 | 56 |
| 3. | |  | | --- | | Function Set: 4-bit, 2 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 |
| 15. | |  | | --- | | Set DDRAM address or cursor position on display | | 0x80 + address\* | 128 + address\* |
| 16. | |  | | --- | | Set CGRAM address or set pointer to CGRAM location | | 0x40 + address\*\* | 64 + address\*\* |

**Problem Statement:** Display your name on LCD.

(For ease of use a header file is given with almost all the commonly used functions so that you just have to call the predefined functions.)

**Algorithm:**

1. Start

2. Initialize LCD

3. Clear LCD screen

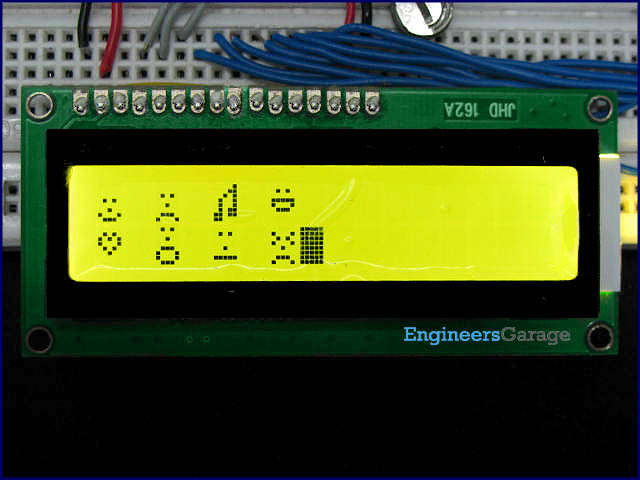
4. Set the cursor to the desired location where data is to be displayed.

5. Display your name by calling appropriate function.

6. Stop

**Extension:**

* Write a program to display “e-Yantra” on 1st line and “Fire Bird V” on 2nd line.
* Write a program to display special characters as follows:



**For more details, you can refer to 8051 Software manual section 6.**