

LCD INTERFACING

LCD modules can be used to display numbers, letters etc. It has widespread use than seven segment LEDs.



Above LCD module has 14 pins. These pins are grouped into three categories:

1. Power pins (VSS, VCC, VEE)
2. Control pins (RS, R/W, E)
3. Data/Command pins (D0-D7)

There are two registers in LCD:

1. Data register : Contains data to be displayed
2. Command register : Contains command to configure LCD operations

To operate LCD, three steps are to be followed:

1. Initialize the LCD.
2. Send any of the commands to the LCD.
3. Send the character to be shown on the LCD.

Following table shows some of the commands to operate LCD. For example to clear the LCD screen, send the command 0x01.

(Hex) Register

1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
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 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
28	2 lines and 5 × 7 matrix (D4-D7, 4-bit)
38	2 lines and 5 × 7 matrix (D0-D7, 8-bit)

Initializing the LCD

- To initialize the LCD for 5 × 7 matrix and 8-bit operation, the following sequence of commands should be sent to the LCD: 0x38, 0x0E, and 0x01

Sending commands to the LCD

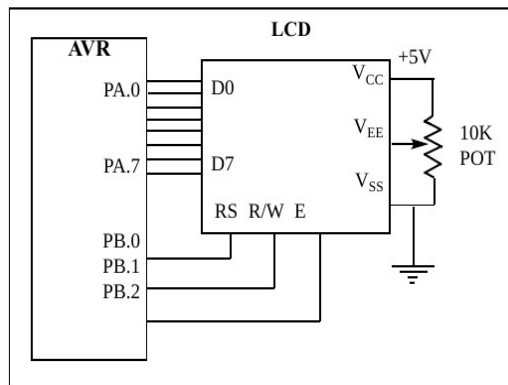
- To send any of the commands to the LCD, make pins RS and R/W = 0 and put the command number on the data pins (D0-D7).

- Then send a high-to-low pulse to the E pin to enable the internal latch of the LCD
- After each command we should wait about 100 μ s to let the LCD module run the command (Clear LCD and Return Home commands are exceptions to this rule.)

Sending data to the LCD

- To send data to the LCD, make pins RS = 1 and R/W = 0
- Then put the data on the data pins (D0–D7) and send a high-to-low pulse to the E pin to enable the internal latch of the LCD

Figure shows the connection diagram of AVR with LCD (with 8 bit data)



LCD Programming in C

Example: Write an AVR C program to display a message in LCD in 8 bit, 16x2 display mode. Use PortA for 8 bit data and PortB pins to control LCD.

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
void lcd_init();
void lcdCommand(unsigned char cmd);
void lcdData(unsigned char data);
void lcdPrint(char *str);
void lcd_init()
{
    DDRA= 0xFF;
    DDRB= 0xFF;
    PORTB &= 0xFB;    // LCD En=0
    _delay_ms(10);
    lcdCommand(0x38);
    lcdCommand(0x0F);
    lcdCommand(0x01);
    _delay_ms(10);
    lcdCommand(0x06);
}
```

```

void lcdCommand(unsigned char cmd)
{
    PORTA = cmd;
    PORTB &= 0xFE;    //RS=0 for command
    PORTB &= 0xFD;    // R/W=0
    PORTB |=0x04;     //EN=1
    _delay_ms(10);
    PORTB &=0xFB;     // EN=0
    _delay_ms(10);
}

void lcdData(unsigned char data)
{
    PORTA = data;
    PORTB |= 0x01;    //RS=1 for data
    PORTB &= 0xFD;    // R/W=0
    PORTB |=0x04;     //EN=1
    _delay_ms(10);
    PORTB &=0xFB;     // EN=0
    _delay_ms(10);
}

void lcdPrint(char *str)
{
    unsigned char i=0;
    while (str[i] != '\0')
    {
        lcdData(str[i]);
        i++;
    }
}

```

```
    }  
}
```

```
int main(void)  
{  
  
    lcd_init();  
    lcdCommand(0x80);    //cursor to beginning of first line  
    lcdPrint("Welcome To");  
    lcdCommand(0xC0);  
    lcdPrint("AKNM GPC");  
  
    while (1);  
    return 0;  
}
```