**T. Y. B. Tech (Electrical and Computer Engineering)**

**Trimester: V Subject: Microcontroller and Applications**

**Name: Shreerang Mhatre Class: TY**

**Roll No: 52 Batch: A3**

**Experiment No: 04**

**Name of the Experiment:** **Interfacing of LCD**

**Performed on: 03/10/2023** 

**Submitted on: 04/11/2023**



**Aim:** Write C program for interfacing of 16x2 LCD with C8051F340 in 8-bit mode.

**Apparatus:** EPBF340 Board, ASK25 board, Connectors

**Theory:**

LCD has the ability to display letters, numbers and characters. A 16x2 LCD can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix.

**LCD pin descriptions:**

**Vcc, Vss and Vee:**

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

**Register Select (RS):**

There are two very important registers inside the LCD. The RS pin is used for their selection as follows.

1. RS = 0: the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home.
2. RS = 1: the data register is selected, allowing the user to send the data to be displayed on the LCD.

**Read/write (R/W):**

R/W input allows the user to write information to the LCD or read information from it. R/W = 1 when reading, R/W = 0 when writing.

**Enable (EN):**

The enable pin is used by the LCD to latch information presented to its data pins. When data is supplied to data pins, a high to low pulse must be applied to the pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of 450ns wide.

**Data bus (D0 – D7):**

The 8-bit data pins, D0-D7 are used to send the information to the LCD or read the contents of the LCD’s internal registers. To display the numbers and letters, we send ASCII codes to these pins while making RS=1.

There are also instruction command codes that can be sent to the LCD to clear the display or blink the cursor.

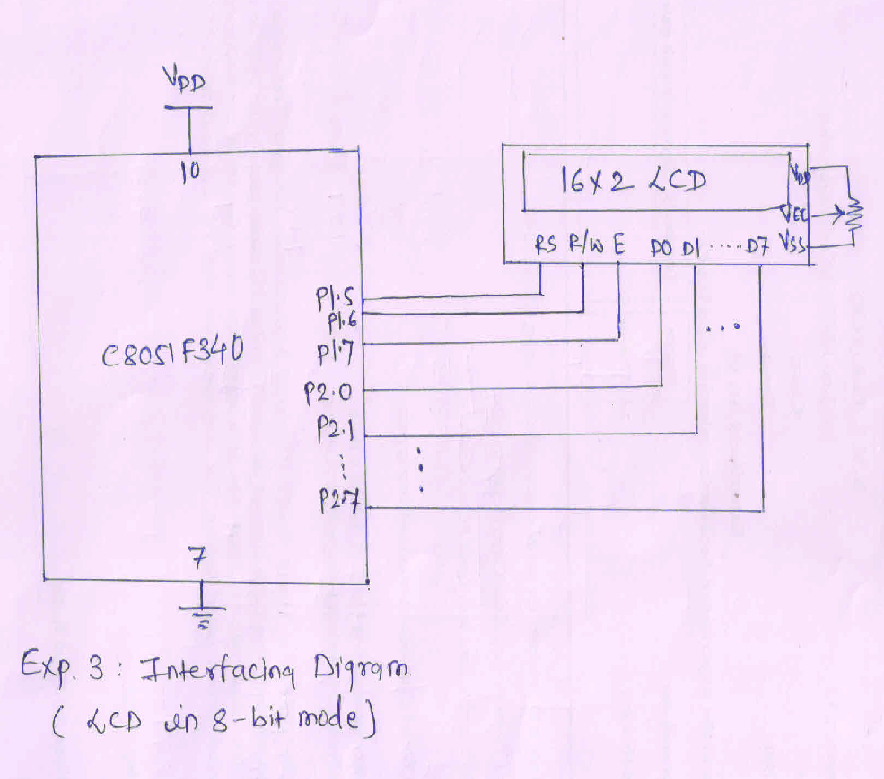
We also use RS = 0 to check the busy flag bit to see if the LCD is ready to receive information. The busy flag is D& and can be read when R/W = 1 and RS=0. When D7 =1, the LCD is busy taking care of internal operations and will not accept any new information. When D7 = 0, the LCD is ready to receive new information.

Table 3.1 Pin Assignment of 16x2 LCD

| **Pin number** | **Symbol** | **Level** | **I/O** | **Function** |
| --- | --- | --- | --- | --- |
| 1 | Vss | - | - | Power supply (GND) |
| 2 | Vcc | - | - | Power supply (+5V) |
| 3 | Vee | - | - | Contrast adjust |
| 4 | RS | 0/1 | I | 0 = Instruction input 1 = Data input |
| 5 | R/W | 0/1 | I | 0 = Write to LCD module 1 = Read from LCD module |
| 6 | E | 1, 1->0 | I | Enable signal |
| 7 | DB0 | 0/1 | I/O | Data bus line 0 (LSB) |
| 8 | DB1 | 0/1 | I/O | Data bus line 1 |
| 9 | DB2 | 0/1 | I/O | Data bus line 2 |
| 10 | DB3 | 0/1 | I/O | Data bus line 3 |
| 11 | DB4 | 0/1 | I/O | Data bus line 4 |
| 12 | DB5 | 0/1 | I/O | Data bus line 5 |
| 13 | DB6 | 0/1 | I/O | Data bus line 6 |
| 14 | DB7 | 0/1 | I/O | Data bus line 7 (MSB) |
| 15 | VB+ | 1 | - | Backlight Supply |
| 16 | VB- | 0 | - |

In 8-bit mode eight data pins are used. 8-bit ASCII value of a character is sent at a single time period and displayed on the LCD.

**Interfacing Diagram:**

****

*Figure 3.1 Interfacing Diagram of 16x2 LCD with C8051F340*

**Hardware Connections:**

Connect flat cable between PL3 connector of ASK25 and PL3 connector of EPBF340 board.

Table 3.1 Hardware connections between EPBF340 and ASK25 board for LCD Interfacing

|  |  |  |
| --- | --- | --- |
| **Pin Connection** | **PL3 Connector of ASK25** | **PL3 Connector of EPBF340** |
| 6 | RS | P1.5 |
| 7 | R/W | P1.6 |
| 8 | EN | P1.7 |
| 9 |  |  |
| 10 | D0 | P2.0 |
| 11 | D1 | P2.1 |
| 12 | D2 | P2.2 |
| 13 | D3 | P2.3 |
| 14 | D4 | P2.4 |
| 15 | D5 | P2.5 |
| 16 | D6 | P2.6 |
| 17 | D7 | P2.7 |
| 19 | 5V | 5.0 V |
| 20 | GROUND | GND |

**Program:** Attach printout of the tested code.

**Result:**

String should be displayed on the LCD.

**Conclusion:**

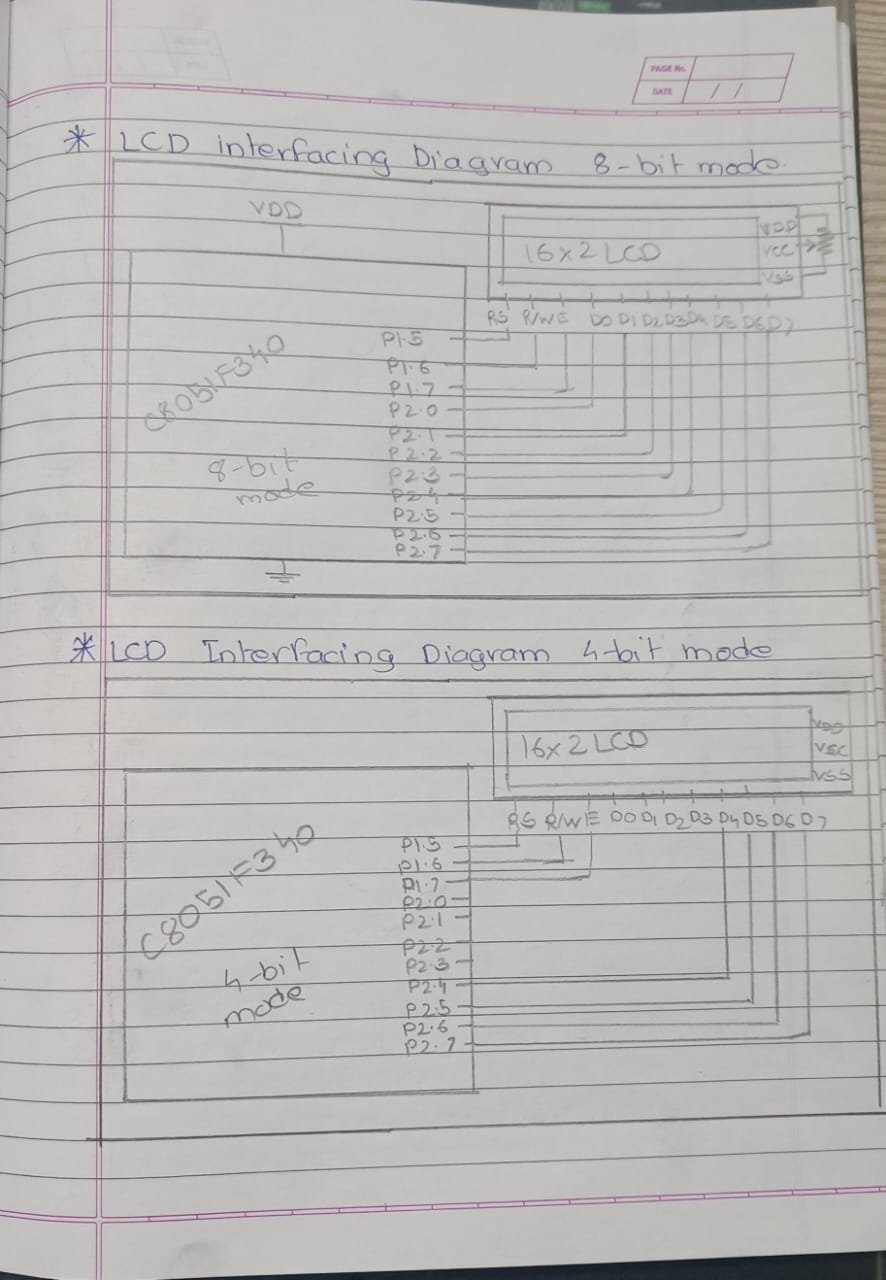
**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Study Question:**

1. Explain the 4-bit mode of LCD.
2. Explain the significance of RS pin and list commands of LCD.
3. Explain buzy flag.

**Additional link:**

1. [*https://www.electronicshub.org/interfacing-16x2-lcd-8051/*](https://www.electronicshub.org/interfacing-16x2-lcd-8051/)

**LCD Interfacing Diagrams:**

**Code for normal LCD Interfacing with C8051F340**

// Exp 4 Basic LCD Interfacing

/\*

Name: Shreerang Mhatre

Rollno: 52

Class: TY

\*/

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x38);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    Write\_Data\_LCD('W');

    DelayMs(50);

    Write\_Data\_LCD('P');

    DelayMs(50);

    Write\_Data\_LCD('U');

    DelayMs(50);

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++)

    {

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    LCD\_RS=0;

    LCD\_RW=0;

    P2=command;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    LCD\_RS=1;

    LCD\_RW=0;

    P2=character;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

**LCD Interfacing with C8051F340 in 8-bit Mode:**



**CODE: LCD displaying Name In 8-bit mode**

// Exp 4 LCD displaying Name In 8-bit mode

/\*

Name: Shreerang Mhatre

Rollno: 52

Class: TY

\*/

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    unsigned char name[]={"SHREERANG"};

    int i;

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x38);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    for(i=0;name[i]!='\0'; i++)

    {

        Write\_Data\_LCD(name[i]);

        DelayMs(50);

    }

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++)

    {

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    LCD\_RS=0;

    LCD\_RW=0;

    P2=command;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    LCD\_RS=1;

    LCD\_RW=0;

    P2=character;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

**LCD Interfacing with C8051F340 in 4-bit Mode:**



**CODE: LCD displaying Name In 4-bit mode**

// Exp 4 LCD displaying Name In 4-bit mode

/\*

Name: Shreerang Mhatre

Rollno: 52

Class: TY

\*/

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    unsigned char name[]={"SHREERANG"};

    int i;

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x28);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    for(i=0;name[i]!='\0'; i++)

    {

        Write\_Data\_LCD(name[i]);

        DelayMs(50);

    }

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++){

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    P2=(command & 0xF0);

    LCD\_RS=0;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

    P2=(command & 0x0F)<<4;

    LCD\_RS=0;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    P2=(character & 0xF0);

    LCD\_RS=1;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

    P2=(character & 0x0F)<<4;

    LCD\_RS=1;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

