Experiment No. 3

PROBLEM DEFINITION: To interface 16x2 Liquid Crystal Display(LCD) to 8051 Microcontroller and write an 8051 ‘C’ code to display the message “GITCSE”

Objectives of the Experiment:

1. To demonstrate the interfacing of 16x2 LCD with 8051 Microcontroller.
2. To develop an 8051 ‘C’ code to display the message “GITCSE”

LCD Interfacing

- LCD is very commonly used electronic display module and having a wide range of applications such as calculators, laptops, mobile phones etc.
- 16×2 character is very basic module which is commonly used in electronics devices and projects.
- It can display 2 lines of 16 characters.
- Each character is displayed using 5×7 or 5×10 pixel matrix.

LCD Interfacing

- LCD can be interfaced with microcontroller in 4 Bit or 8 Bit mode.
- These differs in how data is sent to LCD.
- In 8 bit mode to write a character, 8 bit ASCII data is sent through the data lines D0 – D7 and data strobe is given through EN of the LCD.
- LCD commands which are also 8 bit are written to LCD in similar way.

LCD interfacing

- 4 Bit Mode uses only 4 data lines D4 – D7.
- In this mode 8 bit character ASCII data and command data are divided into two parts and send sequentially through data lines.
- The idea of 4 bit communication is used save pins of microcontroller.
- 4 bit communication is a bit slower than 8 bit communication but this speed difference can be neglected since LCDs are slow speed devices.
- Thus 4 bit mode data transfer is most commonly used.

LCD Interfacing

- **Lcd4_Init () :** These function will initialize the LCD module which is connected to pins defined by following bit addressable variables.

LCD Initialization

- //LCD Module Connections

```
sbit RS(Register Select) = P2^7;
sbit EN (Enable)         = P2^6;
sbit R/W(Read/Write)     = P2.5;
sbit Data D4             = P2^0;
sbit Data D5             = P2^1;
sbit Data D6             = P2^2;
sbit Data D7             = P2^3;
Data D0, D1, D2, D3      = No connection
//End LCD Module Connections
```

LCD Initialization

- 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

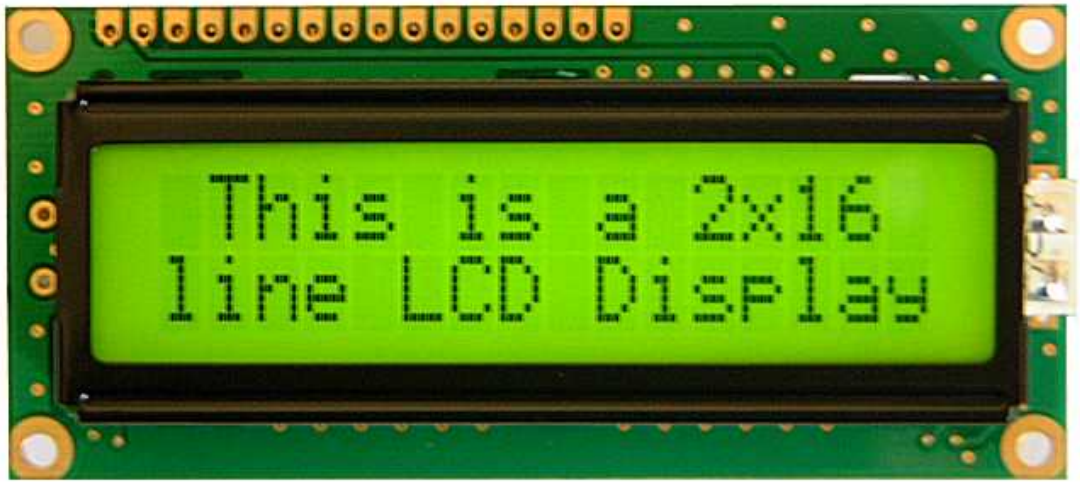
LCD Interfacing

- **Lcd4_Clear() : These functions will clear the LCD screen when interfaced with 8051 in 4 bit mode.**
- **Lcd4_Set_Cursor() : These functions are used to set the cursor position on the LCD screen. By using this function we can change the position of character and string displayed by the following functions.**
- **Lcd4_Write_Char() : These functions are used to write a character to the LCD screen.**

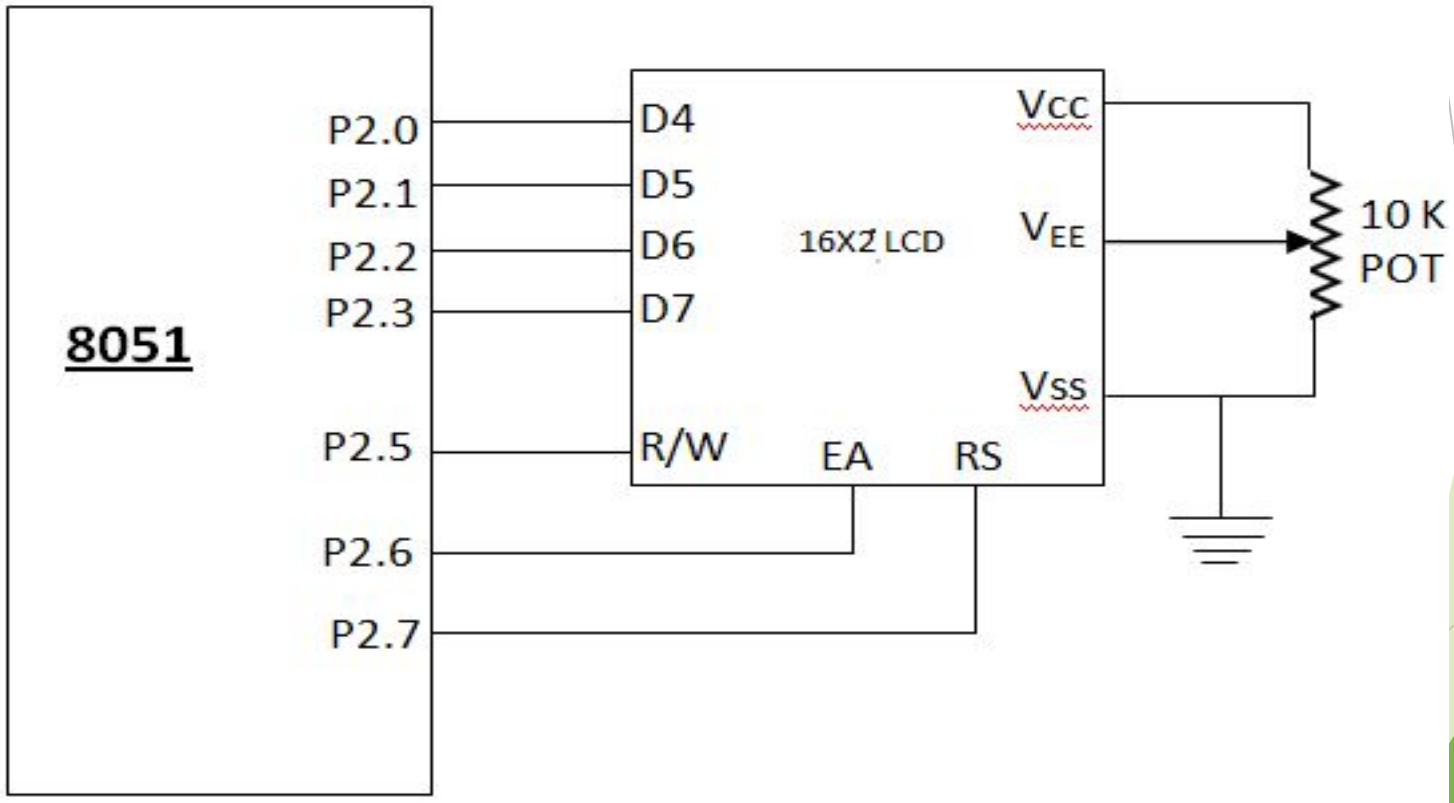
LCD Interfacing

- **Lcd4_Shift_Left() : These functions are used to shift display left without changing the data in display RAM.**
- **Lcd4_Shift_Right() : These functions are used to shift display right without changing the data in display RAM.**

LCD Interfacing



LCD (4-bit mode) interfacing diagram



AUXR - Auxiliary Register (8Eh)

7	6	5	4	3	2	1	0
DPU	-	M0	XRS2	XRS1	XRS0	EXTRAM	AO

Bit Number	Bit Mnemonic	Description
7	DPU	**Disable Weak Pull-up** Cleared by software to activate the permanent weak pull-up (default) Set by software to disable the weak pull-up (reduce power consumption)
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	M0	**Pulse length** Cleared to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 6 clock periods (default). Set to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 30 clock periods.
4	XRS2	**XRAM Size** XRS2 XRS1 XRS0 XRAM size 0 0 0 256 bytes 0 0 1 512 bytes 0 1 0 768 bytes (default) 0 1 1 1024 bytes 1 0 0 1792 bytes
3	XRS1	
2	XRS0	

STEP 1 : INCLUDE THE HEADER FILE ‘’at89c51ed2.h’’

STEP 2 : INCLUDE THE HEADER FILE <intrins.h> AND ADD LCD ROUTINE FILE IN THE SOURCE GROUP (The **INTRINS.H** include file contains prototypes for routines that instruct the compiler to generate in-line intrinsic code.)

STEP 3 : DECLARE LCD FUNCTION PROTOTYPES: INITIALIZATION, COMMAND, AND DATA . **lcd_init(), lcd_comm() AND lcd_data().**

STEP 4 : DECLARE VARIABLES arr, temp1, temp2, i=0.

STEP 5 : BEGIN MAIN

STEP 6 : INITIALIZE **AUXR=0x10** TO ACCESS FULL EXTERNAL RAM. **lcd_init()**

STEP 7: **temp1=0X80**; WRITE COMMAND; **lcd_comm();** TO DISPLAY THE DATA **“GITCSE”** FROM FIRST LINE.

STEP 8: INITIALIZE FOR LOOP TO DISPLAY 6 CHARACTERS. For(i=0; i<6; i++)

STEP 9: LOAD temp2 WITH CHARACTERS: arr[i]

STEP10: WRITE DATA; **lcd_data**

STEP11: TO DISPLAY ANOTHER STRING(EXAMPLE: MICROCONTROLLER) IN SECOND LINE REPEAT STEPS 7 TO 10 WITH temp1=0XC0 AND FOR LOOP INITIALIZED WITH 0 TO NUMBER OF CHARACTERS IN SECOND STRING .

STEP12: REPEAT FOREVER; WHILE(1)

```
#include "at89c51ed2.h"
#include <intrins.h>

// LCD FUNCTION PROTOTYPE
void lcd_init(void);
void lcd_comm(void);
void lcd_data(void);

unsigned char xdata arr[16]={“GITCSE"};
unsigned char temp1=0x00;
unsigned char temp2;
unsigned int i=0;
```

```
void main(void)
{
    AUXR = 0x10;                    // Accessing Full XRAM
    lcd_init();
    temp1 = 0x80;                   // To display from the first line
    lcd_comm();                     //  Command writing
    for(i=0;i<6;i++)
    {
        temp2 = arr[i];
        lcd_data();                 // Data writing
    }
    while(1)
  {
  }
 }
```

Connection Details

- Port 2 to CN6 of Microcontroller Evaluation Board.

Learning Outcomes of the Experiment

At the end of the session, students should be able to :

- Interface 16x2 LCD to 8051 Microcontroller
- Write ‘C’ code to display the message “GITCSE” on LCD display.
- Understand the applications of LCD for various purposes.

Inquiry based learning

- To interface 16x2 Liquid Crystal Display(LCD) to 8051 Microcontroller and write an 8051 ‘C’ code to display the messages on both the lines of LCD