Experiment No. 4	**Batch: ____**	**Roll No.: _______**
Title : Interfacing of 16X2 LCD to 89C51/89S52 in 8 bit and 4 bit mode using delay and busy flag		
Date of Performance :		
Date of Submission :		
Checked By (Sign and Date):		
Remarks:		

Experiment No: 4

AIM: Interfacing of 16x2 LCD with microcontroller 8051/89S52 in 8 bit and 4 bit mode by using time delay and checking busy flag

OBJECTIVES:

1. To understand the interfacing concept of LCD in 8 bit and 4 bit mode
2. To understand Delay and Busy bit programming and execution
3. To understand various commands for initialization of LCD

SOFTWARE REQUIRED:- KEIL µVision IDE, ISPPgm programmer
OPERATING SYSTEM: - WINDOWS 8/10/XP
HARDWARE MATERIAL REQUIRED:-
1. PC-256 RAM, 1.8GHZ, 40GB HDD loaded with above software.
2. AT89S52 Development board, 16x2 LCD module, Connecting wires and power supply through USB port.

REFERENCES:

1. Mazidi, McKinlay "The 8051 Microcontroller and Embedded Systems".
2. Datasheet LCD

Interfacing Diagram (8 bit mode)

Program Statement 1:

Write a assembly language Program to interface LCD in **8 bit mode** by using **time delay** with 89c51/89s52 and display message ‘WELCOME TO VIIT’ such That ‘WEL COME’ will be display on line 1 and ‘TO VIIT’ will be displayed on line 2.

ALGORITHM TO DISPLAY MESSAGE "WELCOME TO VIIT" ON LCD

1. **Start**
2. Define Port for LCD data lines
3. Define Port pins for RS, RW and EN

4. **Initialize type of LCD interface with Microcontroller**
   (To initialize LCD as 8 bit data length, no. of lines 2, resolution 5x7 dots,
   Command word is 0 0 1 DL N F 0 0 = 0011 1000 = 38H,
   refer command table in datasheet time required to execute this command by LCD 40 µsec)
5. Call command subroutine to send this command to LCD module.

6. **Turn on the display and cursor**
   (Command word is 0 0 0 0 1 D C B = 0000 1110 = 0EH)
   (Time required to execute this command by LCD 40 µsec)
7. Call command subroutine to send this command to LCD module.

8. **Clear LCD display, Memory and cursor at home position;**
   (Command word is 0 0 0 0 0 0 0 1 = 01H)
   (Time required to execute this command by LCD 82 µsec to 1.64 µsec)
9. Call command subroutine to send this command to LCD module.

10. **Set the cursor move direction and enable disable the display**
    (To shift cursor to left for next character; Command word is
    0 0 0 0 0 1 I/D S = 0000 0110 = 06H)
    (Time required to execute this command by LCD 40 µsec)
11. Call command subroutine to send this command to LCD module.

12. **Select 1st line and position to display massage (80h) (40 µsec)**
13. Call command subroutine to send this command to LCD module.

14. **Load ASCII code of message into accumulator.**
15. Call Display subroutine to send to LCD module.

16. **Select 2nd line and position to display massage (0C0H) (40 µsec).**
17. Call command subroutine to send this command to LCD module.

18. **Load ASCII code of message into accumulator**
19. Call Display subroutine to send to LCD module

20. Stop

ALGORITHM FOR COMMAND SUBROUTINE (FOR 8 BIT MODE)

1. Check whether LCD is busy or ready for communication, **call delay subroutine** (Provide sufficient time to LCD to complete previous work and to be ready for accepting next information)
2. Copy contents (i.e. command) from register A to Port interface with LCD
3. Select command register RS=0 for command, RS should be low
4. Select LCD in write mode R/W=0 for write command in command register.
5. Enable LCD, E=1 Enable pin should be high To low transition

E = high to low / negative edge triggered —-> Write
E = low to high / positive edge triggered —-> Read

6. Give LCD some time (Time around 1-2 machine cycle)
7. Low Pulse EN=0 to LCD.
8. Return to main program

ALGORITHM FOR DISPLAY SUBROUTINE (FOR 8 BIT MODE)

1. Check whether LCD is busy or ready for communication, **call delay subroutine** (Provide sufficient time to LCD to complete previous work and to be ready for accepting next information)
2. Copy contents (i.e. Data to display) from register A to Port interface with LCD
3. Select Display/Data register RS=1 for Display/Data, RS should be high.
4. Select LCD in write mode R/W=0 for write data in display/data register.
5. Enable LCD, E=1 Enable pin should be high To low transition

E = high to low / negative edge triggered —-> Write
E = low to high / positive edge triggered —-> Read

6. Give LCD some time (Time around 1-2 machine cycle)
7. Low Pulse EN=0 to LCD.
8. Return to main program

ALGORITHM FOR DELAY SUBROUTINE (AROUND 2 msec)

1. ____
2. ____
3. ____
4. ____
5. ____
6. ____

Calculation of delay:

____

____

____

____

____

____

____

____

Program Statement 2.

Write a assembly language Program to interface LCD in **8 bit mode** by using **busy flag** with 89c51/89s52 and display message 'WELCOME TO VIIT' such That 'WEL COME' will be display on line **4th position of line 1** and 'TO ENTC' will be displayed on **4th position of line 2.**

ALGORITHM TO DISPLAY MESSAGE "WELCOME TO VIIT" ON LCD
(Same as main algorithm)

ALGORITHM FOR COMMAND SUBROUTINE (FOR 8 BIT MODE)
Same as in command subroutine
(Instead of calling delay subroutine call check busy flag subroutine)

ALGORITHM FOR DISPLAY SUBROUTINE (FOR 8 BIT MODE)
Same as in command subroutine
(Instead of calling delay subroutine call check busy flag subroutine)

You can use subroutine for checking busy flag or just a big (and safe) delay.

1. Initialize LCD Port as a input, as we are checking status of D7 bit
2. Set R/W Pin of the LCD HIGH(read from the LCD)
3. Select the instruction register by setting RS pin LOW
4. Enable the LCD by Setting the enable pin LOW to HIGH transition to read LCD data
5. The most significant bit of the LCD data bus is the state of the busy flag (1=Busy,0=ready to accept instructions/data). The other bits hold the current value of the address counter. If D7 =1 LCD is busy jump to step 4 for continuous monitoring else return to main program.

If the LCD never come out from "busy" status because of some problems, The program will "hang," waiting for DB7 to go low. So in a real applications it would be wise to put some kind of time limit on the delay--for example, a maximum of 100 attempts to wait for the busy signal to go low. This would guarantee that even if the LCD hardware fails, the program would not lock up.

Interfacing Diagram (4 bit mode)

Program Statement 3.

Write a assembly language Program to interface LCD in **4 bit mode** by using **time delay** with 89c51/89s52, display and rotate message 'WELCOME TO VIIT' on **line 1** from right to left continuously.

ALGORITHM TO DISPLAY MESSAGE "WELCOME TO VIIT" ON LCD

1. **Start**
2. Define Port for LCD data lines
3. Define Port pins for RS, RW and EN

4. **Initialize type of LCD interface with Microcontroller**
   (To initialize LCD as 4 bit data length, no. of lines 2, resolution 5x7 dots,
   Command word is 0 0 1 DL N F 0 0 = 0010 1000 = 28H,
   refer command table in datasheet time required to execute this command by LCD 40 µsec)
5. Call command subroutine to send this command to LCD module.

6. **Turn on the display and cursor**

(Command word is 0 0 0 0 1 D C B = 0000 1110 = 0EH)
(Time required to execute this command by LCD 40 µsec)

7. Call command subroutine to send this command to LCD module.

8. **Clear LCD display, Memory and cursor at home position;**
   (Command word is 0 0 0 0 0 0 0 1 = 01H)
   (Time required to execute this command by LCD 82 µsec to 1.64 µsec)
9. Call command subroutine to send this command to LCD module.

10. **Set the cursor move direction and enable disable the display**
    (To shift cursor to left for next character and digit right to left; Command word is
    0 0 0 0 0 1 I/D S = 0000 0111 = 07H)
    (Time required to execute this command by LCD 40 µsec)
11. Call command subroutine to send this command to LCD module.

12. **Select 1st line and last position to display and rotate massage (8fh)(40 µsec)**
13. Call command subroutine to send this command to LCD module.

14. **Load ASCII code of message into accumulator.**
15. Call Display subroutine to send to LCD module.

16. Goto step 14 for continuous operation

17. Stop

ALGORITHM FOR COMMAND SUBROUTINE (FOR 4 BIT MODE)

1. Check whether LCD is busy or ready for communication, **call delay subroutine** (Provide sufficient time to LCD to complete previous work and to be ready for accepting next information)
2. Copy contents (i.e. command) from register A to temporary register suppose R0
3. Mask higher nibble of A (as we are sending higher nibble first)
4. Swap the higher nibble to LSB side (as lower port pins of port are connected to LCD data lines)
5. Send data to LCD using respective port
6. Select command register RS=0 for command, RS should be low
7. Select LCD in write mode R/W=0 for write command in command register.
8. Enable LCD, E=1 Enable pin should be high to low transition
9. Give LCD some time (Time around 1-2 machine cycle)
10. Low Pulse EN=0 to LCD.
11. Take original data from R0
12. Mask lower nibble
13. Send data to LCD using respective port
14. Select command register RS=0 for command, RS should be low
15. Select LCD in write mode R/W=0 for write command in command register.
16. Enable LCD, E=1 Enable pin should be high To low transition
17. Give LCD some time (Time around 1-2 machine cycle)
18. Low Pulse EN=0 to LCD.
19. Return to main program.

ALGORITHM FOR DISPLAY/DATA SUBROUTINE (FOR 4 BIT MODE)

1. Check whether LCD is busy or ready for communication, **call delay subroutine** (Provide sufficient time to LCD to complete previous work and to be ready for accepting next information)
2. Copy contents (i.e. command) from register A to temporary register suppose R0
3. Mask higher nibble of A (as we are sending higher nibble first)
4. Swap the higher nibble to LSB side (as lower port pins of port are connected to LCD data lines)
5. Send data to LCD using respective port
6. Select data/display register RS=1 for command, RS should be high
7. Select LCD in write mode R/W=0 for write command in command register.
8. Enable LCD, E=1 Enable pin should be high to low transition
9. Give LCD some time (Time around 1-2 machine cycle)
10. Low Pulse EN=0 to LCD.
11. Take original data from R0
12. Mask lower nibble
13. Send data to LCD using respective port
14. Select data/display register RS=1 for command, RS should be high
15. Select LCD in write mode R/W=0 for write command in command register.
16. Enable LCD, E=1 Enable pin should be high To low transition
17. Give LCD some time (Time around 1-2 machine cycle)
18. Low Pulse EN=0 to LCD.
19. Return to main program.

Program Statement 4.

Write a assembly language Program using **MOVC** instruction to interface LCD in **8 bit mode** by using **time delay** with 89c51/89s52, display ‘WELCOME TO ENTC’ on **line 1**.

ALGORITHM TO DISPLAY MESSAGE "WELCOME TO VIIT" ON LCD

(Only main algorithm)

____

____

____

____

____

____

____

____

____

____

____

____

Conclusion:

8bit/4bit mode:________________________________

____

Using delay and busy flag:______________________

____

Using MOVC instruction:_________________________

____

LAB 4
WORKSHEET

1. How many rows and columns are present in a 16*2 alphanumeric LCD?

a) rows=2, columns=32 b) rows=16, columns=2

c) rows=16, columns=16 d) rows=2, columns=16

2. How many data lines are there in a 16*2 alphanumeric LCD?

a) 16 b) 8 c) 1 d) 0

3. Which pin of the LCD is used for adjusting its contrast?

a) pin no 1 b) pin no 2 c) pin no 3 d) pin no 4

4. For writing commands on an LCD, RS bit is

a) set b) reset c) both of the mentioned d) none of the mentioned

5. Which command of an LCD is used to shift the entire display to the right?

a) 0x1C b) 0x18 c) 0x05 d) 0x07

6. Which command is used to select the 2 lines and 5*7 matrix of an LCD?

a) 0x01 b) 0x06 c) 0x0e d) 0x38

7. What changes are to be made to send data to an LCD?

a) set the R/W bit b) set the E bit c) set the RS bit d) all of the mentioned

8. For reading operation from an LCD what changes in the software are introduced?

a) low to high pulse at E pin b) R/W pin is set high c) both of the mentioned

d) none of the mentioned

9. Which instruction is used to select the first row first column of an LCD?

a) 0x08 b) 0x0c c) 0x80 d) 0xc0

10. The RS pin is an input/output for a controller?

a) input b) output c) both of the mentioned d) none of the mentioned