**Experiment 6 [LCD Display using 8051]**

**AIM:**

Write an assembly language program for Lcd Display Using 8051.

**SOFTWARE REQUIRED:**

* Keil software 5.
* Proteus 8 software.

**KEIL PROCEDURE:**

1. Open the software, Click on project and open new version project.

2. Create a new project file

3. Enter AT89C51

4. Click NO

5. Click [Ctrl +N] and Type the code

6. Open project and click Build target

7. Open Build target and open source file and ADD, CLOSE

8. Click build target

9. Next debug start and stop

10. Open peripherals and select port 2

11. Now run the program in Debug

12. Open project and click optional properties and in that give output as hex file.

13.Create Hex file

**PROTEUS SOFTWARE:**

1. Open proteus by clicking run as administrator.
2. Open new project and enter the file name.
3. Click next, next, next and finish.
4. Click P symbol and search keyword and place the required components

The components required are:

* AT89C51
* LM016L

1. Connecting pin number 7 from the LCD (LM016L) to pin 1 in the At89c51
2. Likewise, connect pin 8, 9, 10, 11, 12, 13 & 14 from the LCD(LM016L) to the pins 2, 3, 4, 5, 6, 7 & 8 of the AT89c51
3. Connecting pin 4 from the LCD (LM016L) to the pin 21 in the AT89c51
4. Likewise, connect pins 5 & 6 in the LCD (LM016L) to the pins 22 & 23 in the AT89c51
5. Select the hex file
6. Start the simulation process

**PROGRAM:**

**ORG 0000H**

**RS BIT P2.0**

**RW BIT P2.1**

**EN BIT P2.2**

**MOV A,#38H**

**ACALL CMD**

**MOV A,#0EH**

**ACALL CMD**

**MOV A,#80H**

**ACALL CMD**

**MOV A,#06H**

**ACALL CMD**

**MOV A,#'S'**

**ACALL DATA1**

**MOV A,#'U'**

**ACALL DATA1**

**MOV A,#'B'**

**ACALL DATA1**

**MOV A,#'S'**

**ACALL DATA1**

**MOV A,#'C'**

**ACALL DATA1**

**MOV A,#'R'**

**ACALL DATA1**

**MOV A,#'I'**

**ACALL DATA1**

**MOV A,#'B'**

**ACALL DATA1**

**MOV A,#0C0H**

**ACALL CMD**

**MOV A,#'P'**

**ACALL DATA1**

**MOV A,#'R'**

**ACALL DATA1**

**MOV A,#'O'**

**ACALL DATA1**

**MOV A,#'J'**

**ACALL DATA1**

**MOV A,#'E'**

**ACALL DATA1**

**MOV A,#'X'**

**ACALL DATA1**

**MOV A,#'O'**

**ACALL DATA1**

**MOV A,#'N'**

**ACALL DATA1**

**MOV A,#'I'**

**ACALL DATA1**

**MOV A,#'C'**

**ACALL DATA1**

**CMD:ACALL READY**

**MOV P1,A**

**CLR RS**

**CLR RW**

**SETB EN**

**ACALL DELAY**

**CLR EN**

**RET**

**READY:SETB P1.7**

**CLR RS**

**SETB RW**

**H:CLR EN**

**ACALL DELAY**

**SETB EN**

**JB P1.7,H**

**RET**

**DATA1:ACALL READY**

**MOV P1,A**

**SETB RS**

**CLR RW**

**SETB EN**

**ACALL DELAY**

**CLR EN**

**DELAY:MOV R4,#180**

**HERE1:MOV R3,#255**

**HERE2:DJNZ R3,HERE2**

**DJNZ R4,HERE1**

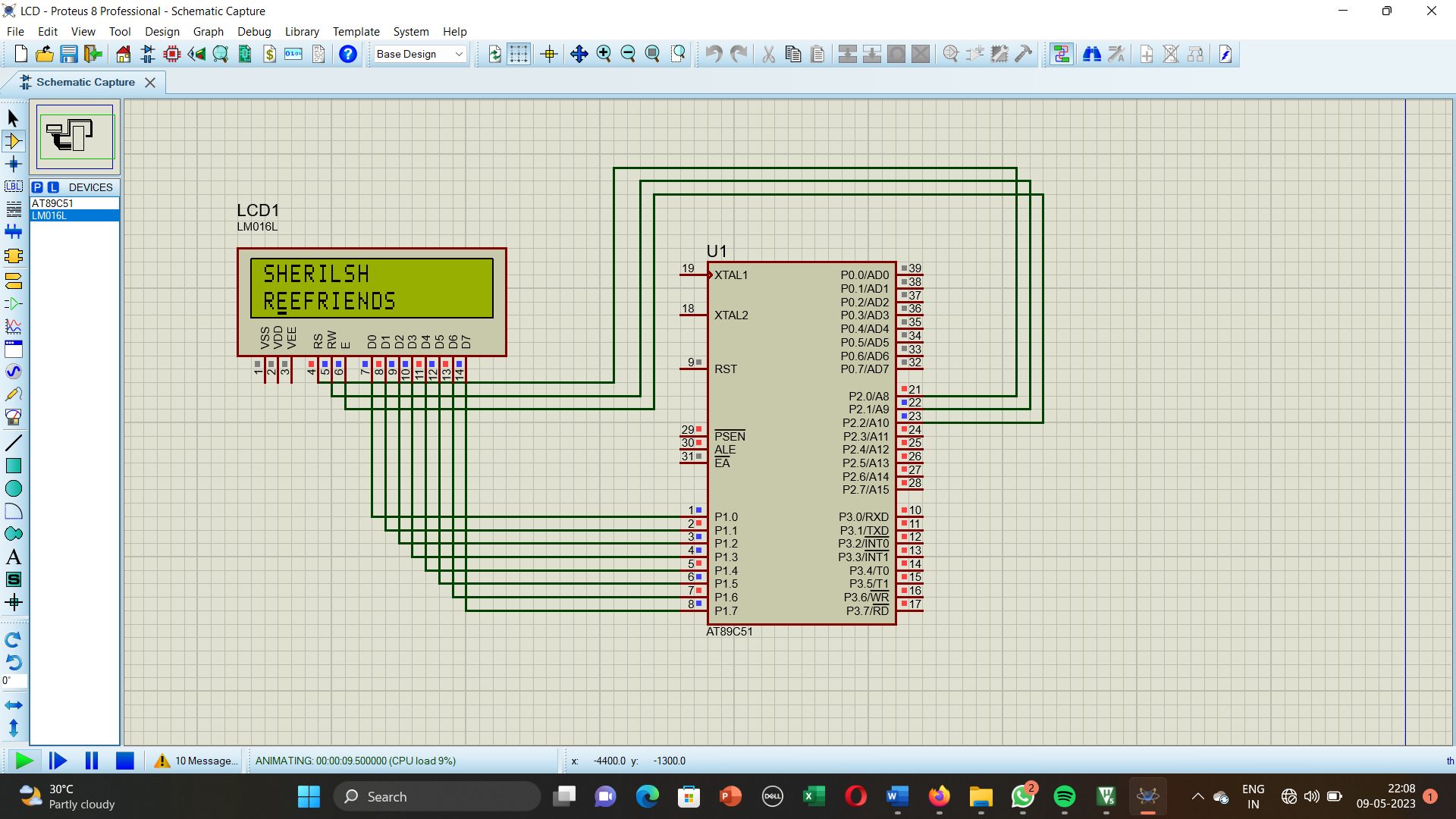
**RET**

**END**

**RESULT:**

Thus the program has been successfully verified and executed.

**OUTPUT:**

****