**Experiment 10 [Led Chaser using 8051]**

**AIM:**

Write an assembly language program for **Led Chaser 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 PROCEDURE:**

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
* Led Bar graph Red
* Crystal
* Choose terminal mode, select ground and place it two times
* Cap

1. Connecting pin number 21 to 1 in Led Bargraph
2. Likewise, connect pin 22, 23, 24, 25, 26, 27 & 28 to the pins 2, 3, 4, 5, 6, 7 & 8 in the Led Bargraph
3. Connecting ground to the Led Bargraph(pin 20)
4. Connecting pins 13, 14, 15, 16, 17, 18, 19, 20 to the ground
5. Connecting pins 19 & 18 of the AT89C51 to the ends of the two capacitors
6. Now, connect both the capacitors together
7. Connect the ground to the Capacitors
8. Connecting two ends of the Crystal
9. Change the frequency of the Crystal to 16MHZ
10. Change the Capacitance to 33pF
11. Select the Hex file
12. Start the simulation process

**PROGRAM:**

**ORG 0000H**

**UP:MOV P2, #01H**

**ACALL DELAY**

**MOV P2, #02H**

**ACALL DELAY**

**MOV P2, #04H**

**ACALL DELAY**

**MOV P2, #08H**

**ACALL DELAY**

**MOV P2, #10H**

**ACALL DELAY**

**MOV P2, #20H**

**ACALL DELAY**

**MOV P2, #40H**

**ACALL DELAY**

**MOV P2, #80H**

**ACALL DELAY**

**SJMP UP**

**DELAY: MOV R4, #255**

**H1:DJNZ R4,H1**

**RET**

**END**

**RESULT:**

Thus the program has been successfully verified and executed.

OUTPUT:

