

# **Experiment-7:**

## **LED Interfacing Using 8051**

### **Aim:**

To interface an LED with the 8051 microcontroller and control its operation.

### **Apparatus Required:**

1. Laptop with Keil uVision software
2. Proteus Design Suite

### **Circuit Diagram Setup in Proteus:**

1. Open Proteus and create a new project.
2. Add the following components from the library:
  - 8051 Microcontroller (AT89C51)
  - LED
  - Resistor (330Ω)
  - Ground (GND) connection
3. Connect the LED's anode to P1.0 of the microcontroller through a 330Ω resistor.
4. Connect the cathode of the LED to GND.
5. Save the design and proceed to programming in Keil.

### **Algorithm:**

1. Configure P1.0 as an output port.
2. Set P1.0 HIGH to turn ON the LED.
3. Introduce a delay.
4. Set P1.0 LOW to turn OFF the LED.
5. Introduce a delay.
6. Repeat the process continuously.

## **Program:**

### **Program (Keil - 8051 Assembly)**

```
; led_blink.asm - Blink LED on AT89C51 P1.0
; Assemble with Keil for AT89C51, produce HEX for Proteus simulation.

        ORG 0000H      ; Reset vector

START:    MOV P1, #0FFH  ; Release Port1 (pull-ups) - make sure pins are high
          by default
          CLR A
MAIN_LOOP:   SETB P1.0    ; Turn ON LED (assuming LED anode -> P1.0,
             cathode -> GND via resistor)
             ACALL DELAY  ; Call delay
             CLR P1.0    ; Turn OFF LED
             ACALL DELAY  ; Call delay
             SJMP MAIN_LOOP ; Repeat forever

; -----
; DELAY subroutine
; Nested loops using R7 (outer) and R6 (inner)
; Adjust values for longer/shorter delays
; -----
DELAY:    MOV R7, #0FFH ; Outer loop count (255)
DELAY_INNER:  MOV R6, #0FFH ; Inner loop count (255)
DELAY_LOOP1: DJNZ R6, DELAY_LOOP1
             DJNZ R7, DELAY_INNER
             RET

END
```

## Output:

## KEIL OUTPUT:

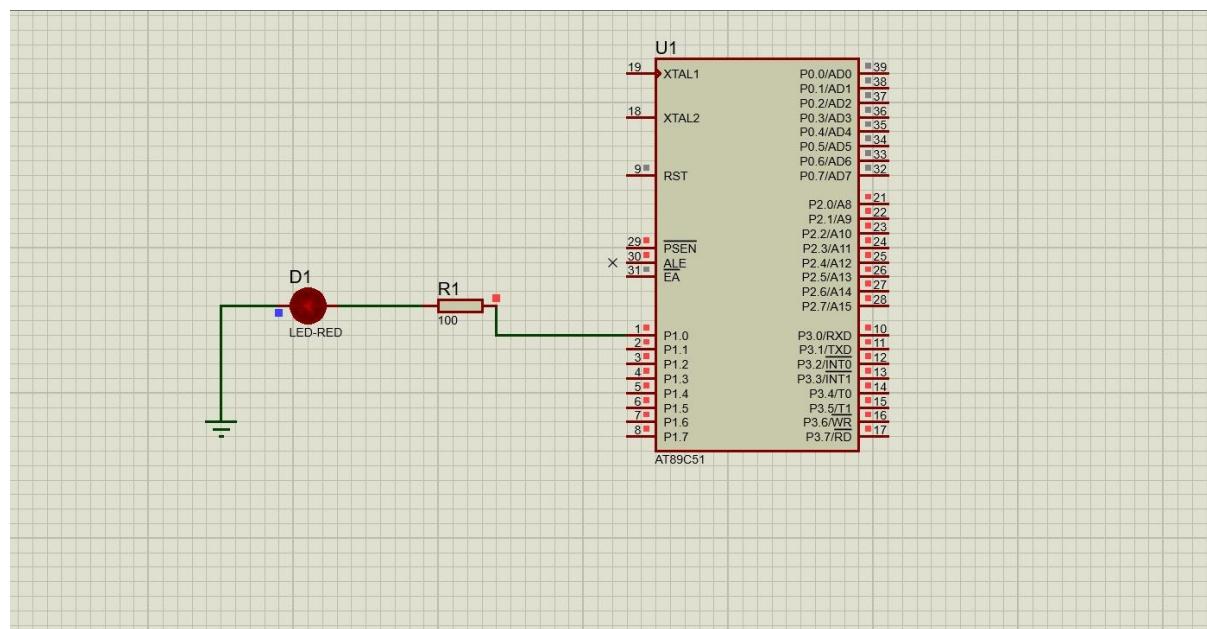
The screenshot shows the Keil uVision IDE interface with the following details:

- Title Bar:** C:\Keil\_v5\CS1\Examples\HELLO\mpmc exp6.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbar:** Standard tools for file operations, project management, and simulation.
- Registers View:** Shows registers R0 through R7 with their current values.
- Disassembly View:** Displays the assembly code for the program:

```
37: SJMP STOP ; Infinite loop to halt the program
C:0x0018 00FE SJMP STOP(C:0018)
C:0x001A 00 NOP
C:0x001B 00 NOP

mmpmc exp6.asm
20 MOV R5, A ; Store difference in R5 (Result: 10H)
21
22 ; --- MULTIPLICATION (Low byte in R6, High byte in B/R7) ---
23 MOV A, R1 ; Load first operand into A
24 MOV B, R2 ; Load second operand into B
25 MUL AB ; Multiply A and B (Product: A=Low byte, B=High byte)
26 MOV R6, A ; Store low byte in R6 (Result: 60H, as 30H * 20H = 600H)
27 ; The high byte is already in B register, you could move B to R7 if desired: MOV R7, B
28
29 ; --- DIVISION (Quotient in A, Remainder in B) ---
30 MOV A, R1 ; Load dividend into A
31 MOV B, R2 ; Load divisor into B
32 DIV AB ; Divide A by B (Quotient in A, Remainder in B)
33 MOV R7, A ; Store quotient in R7 (Result: 01H, integer division)
34 ; The remainder is in B, you could store it in R0 if desired: MOV R0, B
35
36 STOP:
37 SJMP STOP ; Infinite loop to halt the program
38
39 END
```
- Project View:** Shows the project structure with files like mmpmc\_exp6.asm.
- Call Stack + Locals View:** Shows the variable MPMC\_EXP6 with a value of C:0x0003.
- Command Line:** Displays build logs: "Running with Code Size Limit: 2K" and "Load C:\Keil\_v5\CS1\Examples\HELLO\Objects\mmpmc\_exp6".
- Bottom Status Bar:** Shows memory usage, simulation time (t1: 0.00001850 sec), and other system information.

## PROTEUS OUTPUT:



## **Result:**

The LED interfacing with the 8051 microcontroller has been successfully implemented and simulated using Keil and Proteus.