

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:

EXP7 SEMESTER\MPMC\LAB\EXP 7 upproj - μ-Vision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x0
r7	0xffff
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x00
pc	0x0000
PC \$	C000012
states	38
sec	0.000019.
psw	0x00

Disassembly

```

21: DELAY_LOOP1: DJNZ R6, _DELAY_LOOP1
C0:0012 DEFN DJNZ R6,_DELAY_LOOP1(C:0012)
22:    ;DJNZ R6, _DELAY_INNER
C0:0014 DFFA DJNZ R7,_DELAY_INNER(C:0010)

6 START:    MOV P1, #0FFH ; Release Port1 (pull-ups) - make sure pins are high by default
7     CLR A
8     SETB P1.0           ; Turn ON LED (assuming LED anode -> P1.0, cathode -> GND via resistor)
9     ACALL _DELAY         ; Call delay
10    CLR P1.0            ; Turn OFF LED
11    ACALL _DELAY         ; Call delay
12    SJMP MAIN_LOOP       ; Repeat forever
13
14:   -----
15: _DELAY:      MOV R7, #0FFH ; Outer loop count (255)
16: _DELAY_INNER: MOV R6, #0FFH ; Inner loop count (255)
21: DELAY_LOOP1: DJNZ R6, _DELAY_LOOP1
22: DJNZ R7, _DELAY_INNER
23: RET
24:
25: END

```

Project Registers

Command

Running with Code Size Limit: 2K
Load "E:\8th SEMESTER\MPMC\LAB\Objects\EXP 7"

Call Stack + Locals

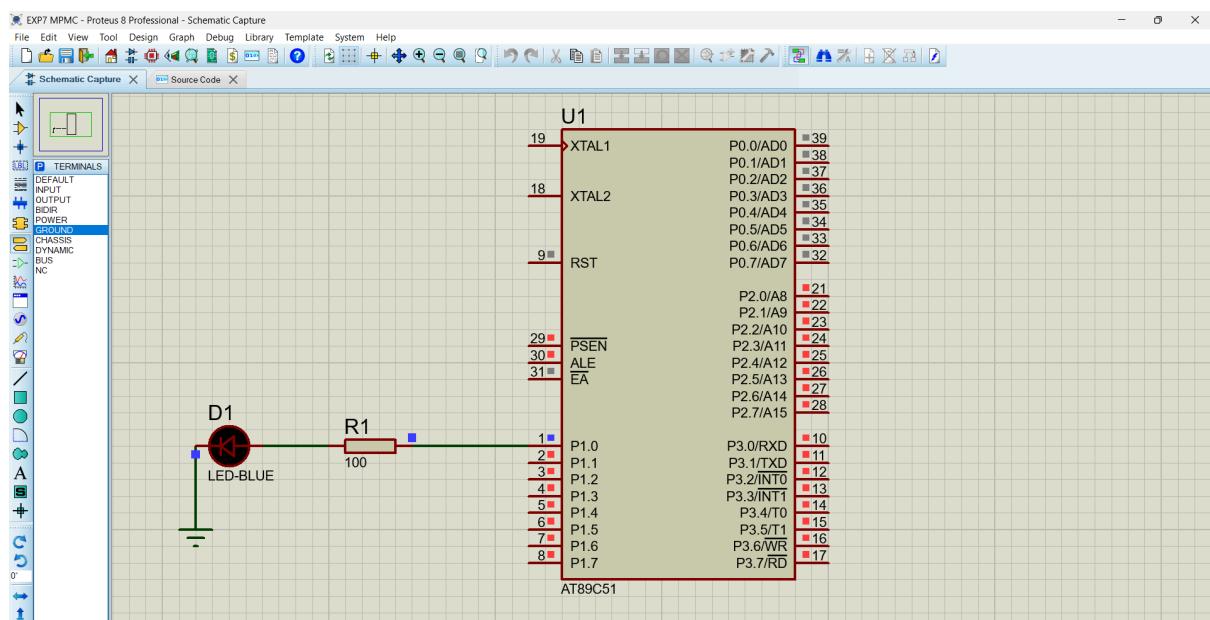
Name	Location/Value	Type
EXP7	C0000E	
EXP7	C00003	

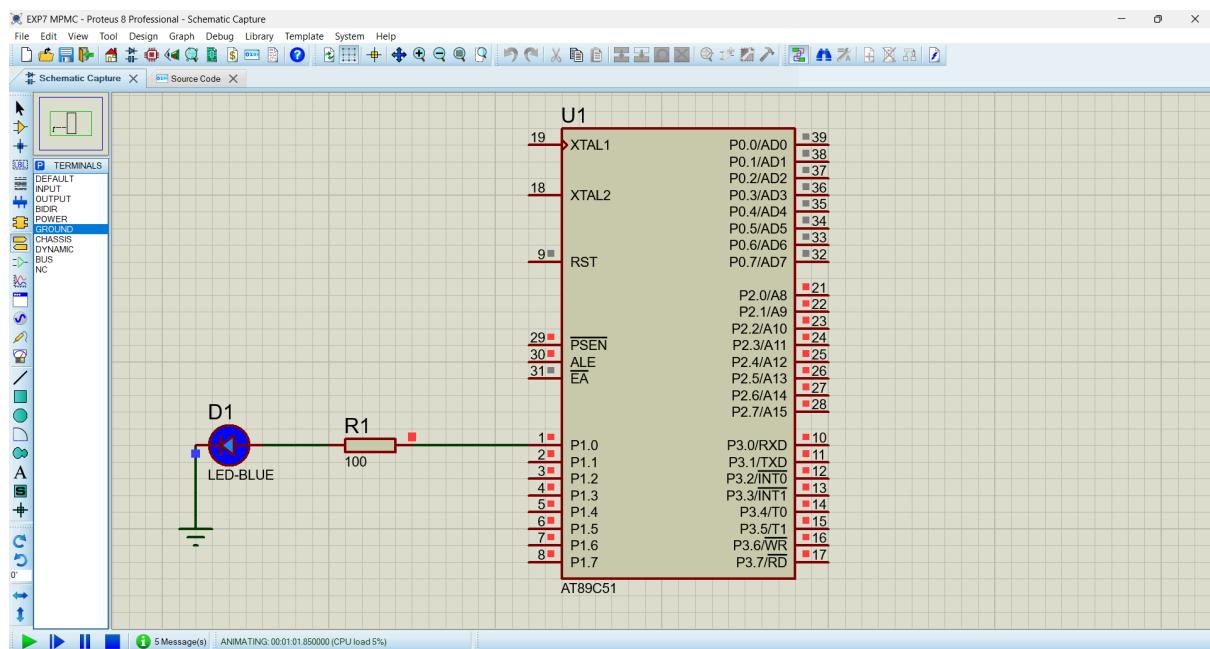
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE

Indents selected text right one tab stop

Simulation t: 0.00001900 sec I:21 C:1 CAP NUM SCRL OVR R/W

PROTEUS OUTPUT:





Result:

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