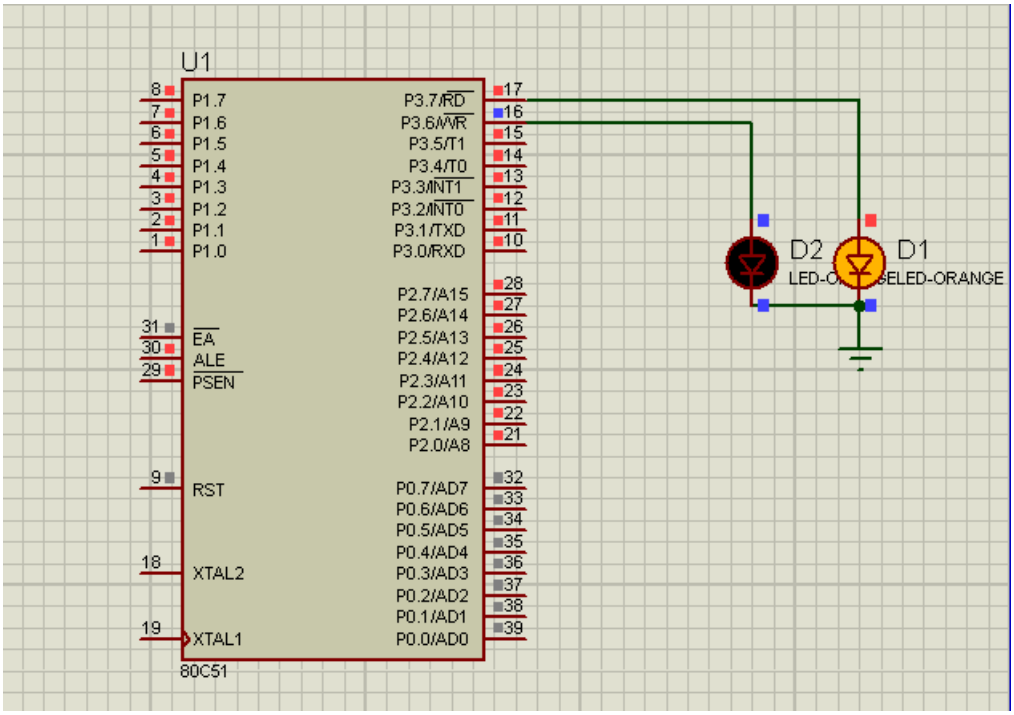
Problem 16



This project uses Timer 1 in mode 1 (16-bit timer) with interrupt triggered every $\cong 50ms$. This $50ms$ is the basic unit of time that is used to make LED blink with T_ON = multiple of 50ms, and T_OFF = another multiple of 50ms.

For example if we set T_ON = 10, and T_OFF = 20 → the actual ON TIME = 10*50ms = 500ms, and OFF TIME = 20*50ms = 1000ms

This is done by using two counter variables, one for the ON_TIME and the other for the OFF_TIME.

Definitions

```
19 LED1_ON_TIME EQU 10
20 LED1_OFF_TIME EQU 20
21 LED2_ON_TIME EQU 50
22 LED2_OFF_TIME EQU 30
23
24 COUNT_ON1 EQU 30H
25 COUNT_OFF1 EQU 31H
26 COUNT_ON2 EQU 32H
27 COUNT_OFF2 EQU 33H
28
29 LED1 EQU P3.7
30 LED2 EQU P3.6
31
32 LED1_STATE EQU 00H
33 LED2_STATE EQU 01H
```

19 to 22, defines the ON/OFF values for two LED (multiple of 50ms)

We need to define a counter for each TIME_ON and TIME_OFF (lines 24-27)

(29-30) Port pins connected to LEDs

We keep the status of the LEDs in two boolean variables as in lines 32,33

Main loop

```
42       org   0000h
43       jmp   Start
44 ORG 001BH
45    LJMP ISR_T1
46 ;=================================================
47 ; CODE SEGMENT
48 ;=================================================
49
50       org   0100h
51 Start:
52
53    MOV TMOD,#10H
54    MOV TL1,#0B0H
55    MOV TH1,#03CH
56    SETB EA
57    SETB ET1
58    SETB TR1
59    CLR LED1
60    CLR LED2
61
62    MOV COUNT_ON1,#LED1_ON_TIME
63    MOV COUNT_OFF1,#LED1_OFF_TIME
64    MOV COUNT_ON2,#LED2_ON_TIME
65    MOV COUNT_OFF2,#LED2_OFF_TIME
66
67 START2:
68    JMP START2
```

First we jump to start label (43), where the interrupt vector of Timer1 overflow will be at address 001Bh.

When Timer1 overflow, the 8051 will be interrupted and execute the code at address 001Bh. So we put a jump to the starting code for our timer handler (45)

At 53, we set Timer 1 in mode 1
At 54-55, we initialize the Timer by 3CB0h, which will cause an overflow after counting 50000 clock cycle, and since the clock of the counter = 12M/12 = 1us→ the time passed will be 50ms

56-57, we enable Timer1 overflow interrupt (note that we must set both EA, and ET1)

ET1 → enable timer 1 overflow interrupt

EA → must be 1 to enable 8051 interrupt

58 → we make Timer1 run by setting TR1

Lines 62-65 initialize the ON/OFF counter for both LEDs

the main loop in our example do nothing! (as all the processing will be done by Timer1 overflow interrupt)

Timer1 Overflow Interrupt

```
71  ISR_T1:
72     MOV TL1,#0B0H
73     MOV TH1,#03CH
74  TEST_LED1:
75     JB LED1_STATE,LED1_ON
76     CLR LED1
77     DJNZ COUNT_OFF1,TEST_LED2
78     SETB LED1_STATE
79     MOV COUNT_OFF1,#LED1_OFF_TIME
80     JMP TEST_LED2
81  LED1_ON:
82     SETB LED1
83     DJNZ COUNT_ON1,TEST_LED2
84     CLR LED1_STATE
85     MOV COUNT_ON1,#LED1_ON_TIME
86  TEST_LED2:
```

First we re-initialize timer 1 by 2^16-50000 = 3CB0 (72, 73) to enable it to count for a time of 50ms again

75→ test if the current state of LED1 is ON; if not, so it is OFF, and we will make

LED1 OFF (76). Then we decrement the OFF counter for LED1 (77) and if it becomes '0', we will change the state of the LED1 to ON (78), and re-initialize the off counter (79)

If LED1 state is ON (81), we make LED ON (82), then decrement the ON-counter (83)-if this counter becomes '0' , we will change the state of LED1 to OFF (84), and re-initialize the ON-counter (85)

The same will be done for LED2 as shown below

```
86  TEST_LED2:
87     JB LED2_STATE,LED2_ON
88     CLR LED2
89     DJNZ COUNT_OFF2,TEST_LED3
90     SETB LED2_STATE
91     MOV COUNT_OFF2,#LED2_OFF_TIME
92     JMP TEST_LED3
93  LED2_ON:
94     SETB LED2
95     DJNZ COUNT_ON2,TEST_LED3
96     CLR LED2_STATE
97     MOV COUNT_ON2,#LED2_ON_TIME
98  TEST_LED3:
99
100 RETI
```