

Lab 8: Timers with 8051

1 Problem Statement

Write an assembly program to:

1. Blink an LED at 1s intervals using 8051 built-in hardware timers instead of software-generated time delays. Use Timer 0, mode 1 (16-bit) to create the delay.
2. Toggle all the LEDs continuously with a delay of 2s in between. Use Timer 1, 16-bit mode to generate the delay.
3. Write a program to provide the given on/off time to three traffic lights (green, yellow, and red) with timings specified as follows

Light	On time
Green	3 seconds
Yellow	1 seconds
Red	2 seconds

2 Timer Registers of 8051

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

Table 1: TCON register (Timer Control Register)

GATE	C/T'	M1	M0	GATE	C/T'	M1	M0
------	------	----	----	------	------	----	----

Table 2: TMOD register (Timer Mode Register)

M1	M0	Description
0	0	13-bit timer mode
0	1	16-bit timer mode
1	0	8-bit auto-reload mode
1	1	Split timer mode

Table 3: Timer Modes

3 Procedure for generating time delay using timers

```
org 0000h
ljmp main

org 000bh ; Interrupt service routine during Timer 0 (T0) overflow
ljmp interrupt_routine

org 100h
main:
; Take a 16 bit number N as input to load in TL0 and TH0 for the required time delay.
Store upper byte in R1 and lower byte in R0

; Enabling interrupt from T0
setb EA
setb ET0

; Initializing timer count
mov th0, R1 ; TH0 = R1
mov tl0, R0 ; TL0 = R0

; Enabling the timer T0 to run. To start Timer 0, set tr0(TCON.4). Whenever T0
overflows, TF0 will be set (TCON.5), and the currently running program will be
interrupted.
mov tmod, #-h ; setting mode 1 and timer 0
```

```
setb tr0

here: sjmp here

org 300h
interrupt_routine:

; restore the count
mov th0, R1 ; TH0 = R1
mov tl0, R0 ; TL0 = R0

; code to blink LEDs for the desired time period.

reti
end
```