

Lab 6: Switches and Timers

1 Problem Statement

Write assembly program to:

1. Read a nibble from slider switch:
 - (a) Configure Port 1 so as to read the input from slider switch.
 - (b) Read a number N from onboard slider switches (P1.3 - P1.0 such that $1 \leq N \leq 16$)
 - (c) Display the value on onboard LEDs
 - (d) Store this nibble as last four bits of memory location 60H.
2. Blink LEDs with an exact time delay:
 - (a) Blink an LED at 1s intervals using 8051 built-in hardware timers instead of software generated time delays.

2 Procedure for accessing data from switch

Sequence of steps to be followed to read and store a nibble

Logic to read a 4 bit number (nibble) from onboard switches and get confirmation from user

ReadNibble: ; Routine to read a nibble and confirm from user

```
loop:
;configure the port P1.0-P1.3 as input (to configure a port as input, set it) and
turn on all 4 leds (to indicate program is ready to accept input)
;wait for 5 sec during which user can give input through switches
;turn off all LEDS
;save the read value in a register
;wait for one sec
;show the read value on LEDS
;wait for 5 sec (during this time delay, user can put all switches to OFF position to
signal that the read value is correct and routine can proceed to next step)
;clear leds
;read the input from switches
;if read value < or > 00h go to loop
;otherwise return and store previously read nibble in location 60H (lower 4 bits).
```

Program to create a delay of 'n' seconds

```
DELAY:
USING 0
PUSH OE0H
PUSH ARO ;saving the contents of the registers which are going to be used
PUSH AR1
PUSH AR2
MOV A, #2n ;Value of '2*n' to create a delay of n seconds is loaded in A
MOV B, #10;Load 10 in B
MUL AB ;Multiply 10 with n to get the number of iterations for the 50ms loop
MOV R0, A ;Move the result to R0 which is used as the iterator in the loop
BACK1: ;following is a nested loop which generates a delay of 50ms. This delay
has been iterated 10*2*n times to get a total delay of 500*2*n ms = n seconds
MOV R2,#200
BACK2:
MOV R1,#OFFH
BACK3:
DJNZ R1, BACK3
DJNZ R2, BACK2
DJNZ R0, BACK1
POP AR2 ;popping the register in the exact reverse order before
POP AR1
POP ARO
POP OE0H
RET ;exiting from the subroutine
```

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 TH0 and TL0 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
```