EE 337: Introduction to Timers
Lab 5

29^{th} August, 2016

Following are code templates for the questions in Lab 5 (Introduction to Timers)

- Code format for Q.1 of LabWork: Delay generation by polling

```
ORG 0000H
LJMP MAIN

ORG 0100H
INITIALIZATION:
; Timer0 initialization
MOV TMOD,,#XXH;; Timer0 as 16 bit counter
MOV TH0,,#XXH;
MOV TL0,,#XXH;
MOV R7, #XXH ; Count to keep track of number of overflows
RET

;--------------- MAIN STARTS HERE --------------------
MAIN:
LCALL INITIALIZATION

;-- Delay by polling --;
; Start the timer run
BACK: ; Wait here till Timer does not overflow
; Clear timer flag
;Reduce R7 by 1. If R7 is not zero go to back
;--delay by poling ends--;
OVER: SJMP OVER
END
```

- Code format for Q.2 of LabWork: Delay generation by interrupt

```
ORG 0000H
LJMP MAIN

ORG 00XXH
;Subroutine of Timer0 overflow

ORG 0100H
INITIALIZATION:
```

```
; Timer0 initialization
MOV TMOD,,#XXH;; Timer0 as 16 bit counter
MOV TH0,,#XXH;
MOV TL0,,#XXH;
; Enable Timer0 overflow interrupt
MOV R7, #XXH ; Count to keep track of number of interrupt calls of overflow
RET

;--------------- MAIN STARTS HERE --------------------
MAIN:

LCALL INITIALIZATION
; Configure the Timer0 to run
OVER: SJMP OVER
END
```

- Code format for Pulse width measurement

```
; LCD Pins
LCD_data  equ P2    ;LCD Data port
LCD_rs    equ P0.0  ;LCD Register Select
LCD_rw    equ P0.1   ;LCD Read/Write
LCD_en    equ P0.2   ;LCD Enable

ORG 0000H
LJMP MAIN

ORG 00XXH ; NEGATIVE EDGE ROUTINE of External Interrupt 0
; Display readings on LCD in this routine.

ORG 00XXH ; Timer 0 overflow interrupt routine
; Keep track of number of overflows here

ORG 0100H
;------------------------LCD Initialisation routine-----------------------------------
lcd_init:
mov   LCD_data,#38H  ;Function set: 2 Line, 8-bit, 5x7 dots
clr   LCD_rs         ;Selected command register
clr   LCD_rw         ;We are writing in instruction register
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en
acall delay

mov   LCD_data,#0CH  ;Display on, Curson off
clr   LCD_rs         ;Selected instruction register
clr   LCD_rw         ;We are writing in instruction register
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en

acall delay
```

```
mov   LCD_data,#01H  ;Clear LCD
clr   LCD_rs         ;Selected command register
clr   LCD_rw         ;We are writing in instruction register
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en

acall delay

mov   LCD_data,#06H  ;Entry mode, auto increment with no shift
clr   LCD_rs         ;Selected command register
clr   LCD_rw         ;We are writing in instruction register
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en

acall delay

ret                  ;Return from routine

;-----------------------command sending routine--------------------------------
lcd_command:
mov   LCD_data,A     ;Move the command to LCD port
clr   LCD_rs         ;Selected command register
clr   LCD_rw         ;We are writing in instruction register
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en
acall delay

ret
;-----------------------data sending routine-----------------------------------
lcd_senddata:
mov   LCD_data,A     ;Move the command to LCD port
setb  LCD_rs         ;Selected data register
clr   LCD_rw         ;We are writing
setb  LCD_en         ;Enable H->L
acall delay
clr   LCD_en
acall delay
acall delay
ret                  ;Return from busy routine

;----------------------delay routine-------------------------------------------------
delay:
USING 0
PUSH AR0
PUSH AR1
mov r0,#1
loop2:  mov r1,#255
loop1:  djnz r1, loop1
djnz r0,loop2
POP AR1
```

```
POP AR0
ret

;----------------------- CONVERSION TO ASCII ----------------
ASCIICONV:                  ; binary to ascii converter
; Write your subroutine here
;
RET ; bin to ascii ends here


;--------------- Specific to this application -------------
INITIALIZATION:
; Port and register initialization

; Timer0 initialization
;Configure Timer0 in 16-bit and Gating mode from P3.2
;MOV TH0,,#XXH;
;MOV TL0,,#XXH;

WAIT_NEW_PULSE: JB P3.2, WAIT_NEW_PULSE ; If previous pulse ''HIGH'' is already
; started, Wait till it goes off. This will ensure measuremnt starts exactly at the
; pulse start

;Interrupt Configuration
;MOV 88H, #XXH ;Clear false '1' of IE0 BIT before turning on iterrupt for first time,
; if any.
; Enable required interrupts(Falling edge interrupt on P3.2 and T0 Overflow interrupt)
; Enable global interrupt bit

; Configure Timer0 to run now.
RET
;--------------- MAIN STARTS HERE --------------------
MAIN:

LCALL INITIALIZATION
OVER: SJMP OVER

END
```