

```

;***** main.s *****
; Program written by: HYDER SHAD, MIRA SEHGAL
; Date Created: 1/22/2016
; Last Modified: 2/7/2016 12:58 AM
; Section: THURSDAY 2-3 PM
; Instructor: V. JANAPA
; Lab number: 2
; Brief description of the program
; The overall objective of this system an interactive alarm
; Hardware connections
; PF4 is switch input (1 means SW1 is not pressed, 0 means SW1 is
pressed)
; PF3 is LED output (1 activates green LED)
; The specific operation of this system
;   1) Make PF3 an output and make PF4 an input (enable PUR for PF4).
;   2) The system starts with the LED OFF (make PF3 =0).
;   3) Delay for about 100 ms
;   4) If the switch is pressed (PF4 is 0), then toggle the LED once,
else turn the LED OFF.
;   5) Repeat steps 3 and 4 over and over

SYSCTL_RCGCGPIO_R EQU 0x400FE608
GPIO_PORTF_DATA_R EQU 0x400253FC
GPIO_PORTF_DIR_R EQU 0x40025400
GPIO_PORTF_AFSEL_R EQU 0x40025420
GPIO_PORTF_PUR_R EQU 0x40025510
GPIO_PORTF_DEN_R EQU 0x4002551C
GPIO_PORTF_LOCK_R EQU 0x40025520
GPIO_PORTF_CR_R EQU 0x40025524
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R EQU 0x4002552C
GPIO_LOCK_KEY EQU 0x4C4F434B
countdown EQU 0x4E200 ; delay subroutine countdown set to 320,000 to
achieve 100ms toggle delay
        AREA    |.text|, CODE, READONLY, ALIGN=2
        THUMB
        EXPORT Start

;
-----
```

INIT

```

; 1) activate clock for Port F
LDR R0, =SYSCTL_RCGCGPIO_R
LDR R1, [R0]
ORR R1, #0X20      ;SET BIT 5 HIGH TO ENABLE PORT F CLOCK
STR R1, [R0]
NOP
```

```

        NOP          ;ALLOW TIME TO FINISH ACTIVATING (2+ CYCLES)
        NOP

; 2) enable pull-up resistors for negative logic

        LDR R0, =GPIO_PORTF_PUR_R ;PUR LIMITS CURRENT APPLIED TO PIN
ON THE MICROCONTROLLER
        LDR R1, [R0]
        ORR R1, #0X10           ;PULL UP RESISTOR ON BIT 4 BECAUSE PF4
IS INPUT USING NEGATIVE LOGIC
        STR R1, [R0]

; 3) set direction register

        LDR R0, =GPIO_PORTF_DIR_R      ;SET BIT 3 HIGH FOR IT TO BE
OUTPUT
        LDR R1, [R0]
        ORR R1, #0X08
        STR R1, [R0]

; 4) disable analog functionality

        LDR R0, =GPIO_PORTF_AMSEL_R ;DISABLE ANALOG CAPABILITIES,
SOLEY DIGITAL I/O
        MOV R1, #0X0
        STR R1, [R0]

; 5) configure as GPIO

        LDR R0, =GPIO_PORTF_PCTL_R
        MOV R1, #0X0              ;CLEAR PORT CONTROL FIELD TO SET UP
PINS FOR GPIO
        STR R1, [R0]

; 6) regular port function

        LDR R0, =GPIO_PORTF_AFSEL_R ;DISABLE ALT FUNCTIONS FOR PIN BY
SETTING BITS TO ZERO
        MOV R1, #0
        STR R1, [R0]

; 7) enable digital port

        LDR R0, =GPIO_PORTF_DEN_R      ;R1 =
&GPIO_PORTD_DEN_R
        ORR R1, #0X18                ;ENABLE DIGITAL I/O ON
PINS 5-2
        STR R1, [R0]

; 8) exit initialization process

```

```

BX LR

;-----

Start    BL INIT          ;initialize port f for I/O
        LDR R0, =GPIO_PORTF_DATA_R ;load R0 with address for port
f data register

loop     LDR R1, [R0]
        CMP R1, #0X10   ;compare instruction sets condition codes for
the following conditional branch instruction
        BEQ loop      ;if PF4 (bit 4) is not pressed/activated,
continue checking bit, else if PF4 is pressed, continue to toggle LED
        ORR R1, #0X08    ;set bit 3 to 1 for LED ON (FREINDLY
CODE USING ORR)
        toggle  STR R1, [R0]  ;toggle LED, write LED ON to data register

        BL delay           ;time
delay   EOR R1, #0X08    ;set bit 3 to 0 for LED OFF (FREINDLY CODE
USING EOR)
        STR R1, [R0]    ;write LED OFF to data register
        BL delay           ;time delay
        B  loop            ;repeat check for switch activation

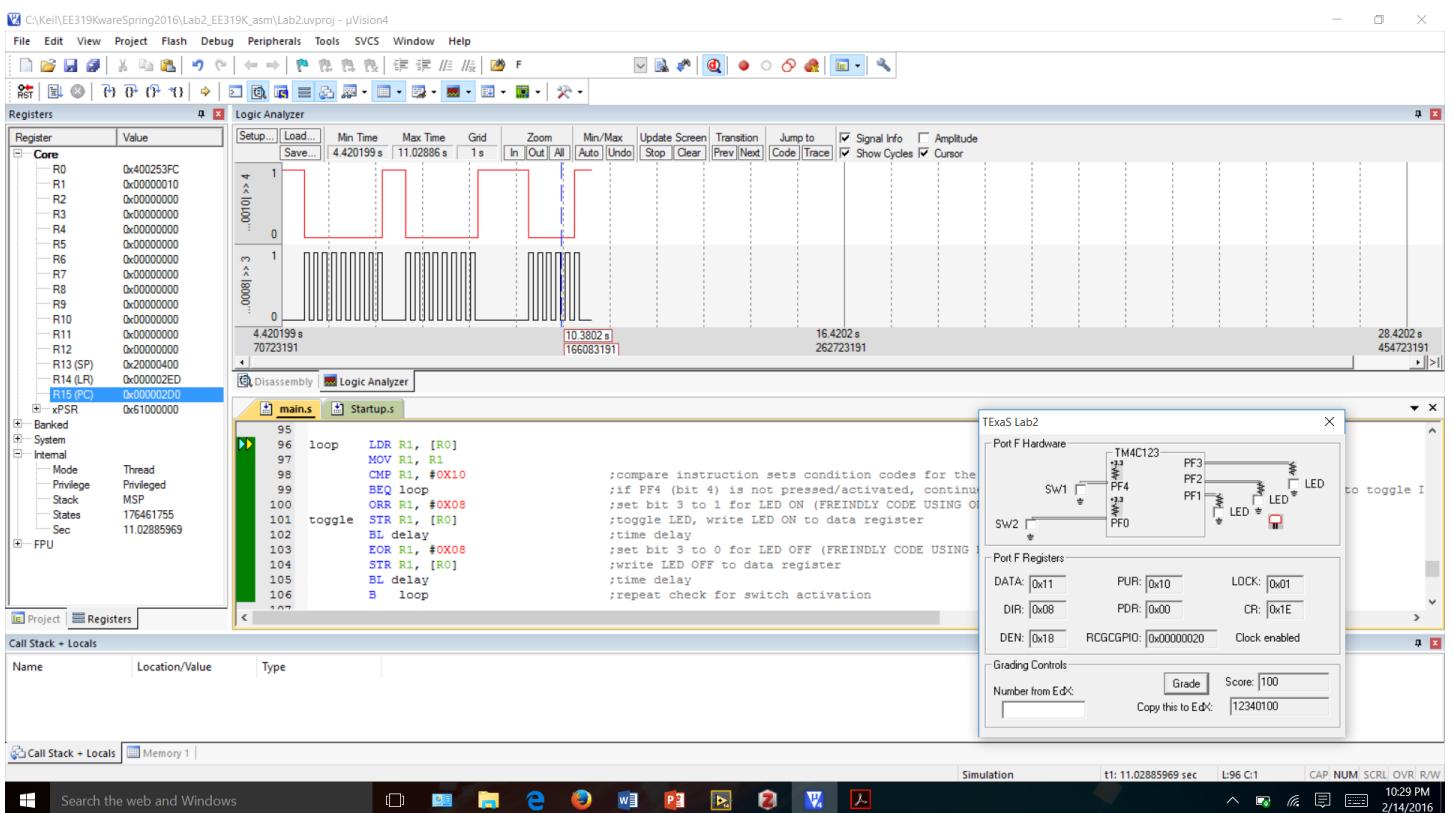
delay   LDR R3, =countdown

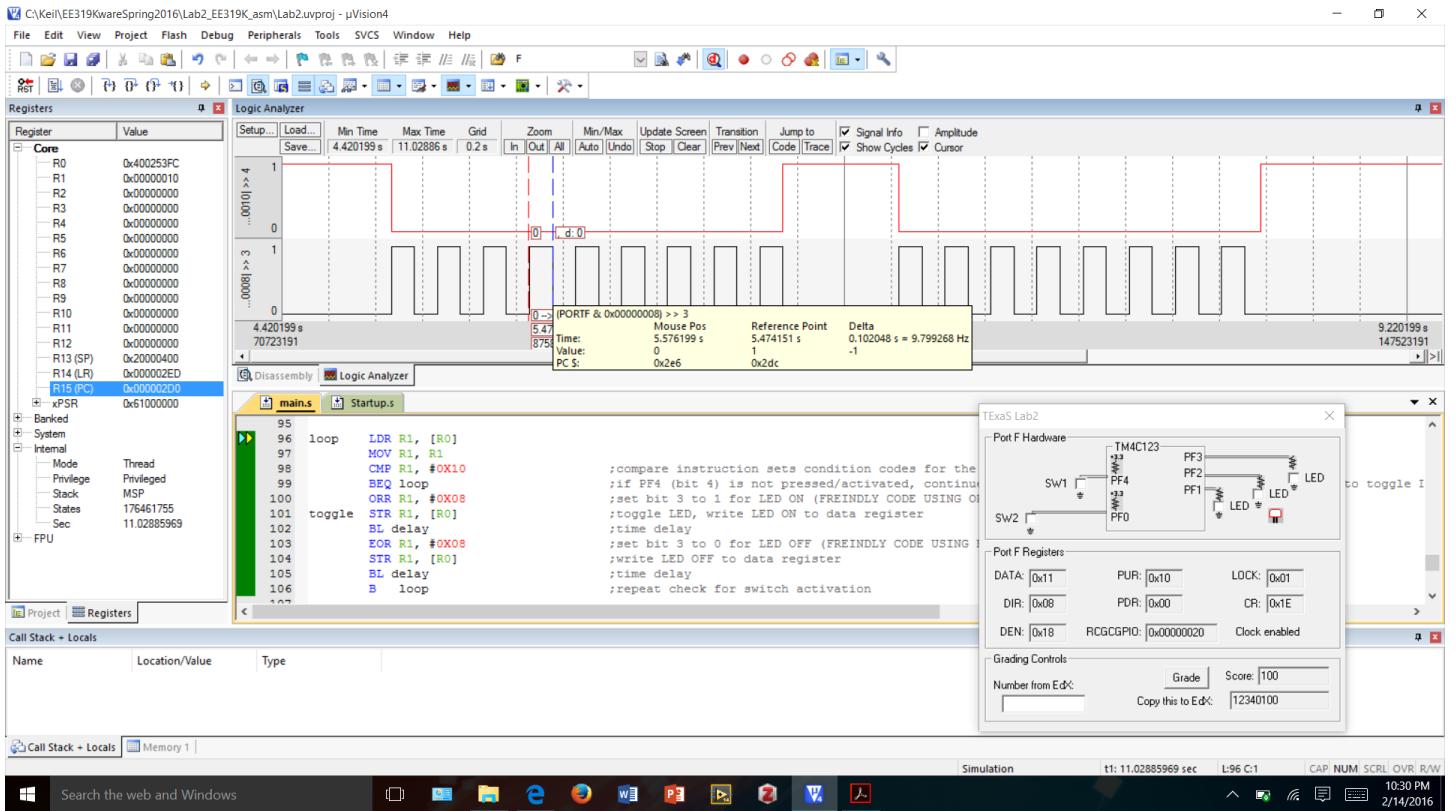
count   ADD R3, #-1      ;subtract 1 from countdown

        CMP R3, #0      ;compare to see if countdown has reached 0
        BNE count      ;count not 0, then loop to count and subtract 1
again, count = 0, then exit subroutine
exit    BX LR          ;exit delay subroutine, use link register to
return to main program code

;-----
-----
```

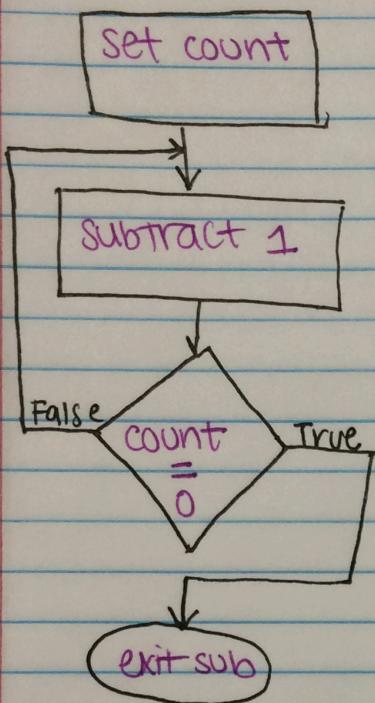
ALIGN ;make sure the end of this section is aligned
END ;end of file





Mira Sehgal
Hyder Shah

Flowchart Delay



Pseudocode Delay

- ① set a value to count down from
- ② subtract one
- ③ compare count with 0
True = exit subroutine
False = loop to subtract 1 again
- ④ exit subroutine

Delay Subroutine 100ms delay Calculation:

delay LDR R₃, = count down

Count ADD R₃, # -1
CMP R₃, # 0
BNE Count
BX LR

} delay takes 5 cycles to complete

1 cycle = 0.0625 μs

5 cycles = 0.3125 μs

$\frac{100 \text{ ms}}{0.3125 \mu\text{s}} = 320,000$ repetitions of the delay
required to achieve 100ms toggle delay

Countdown = 320,000 = 0x4E200