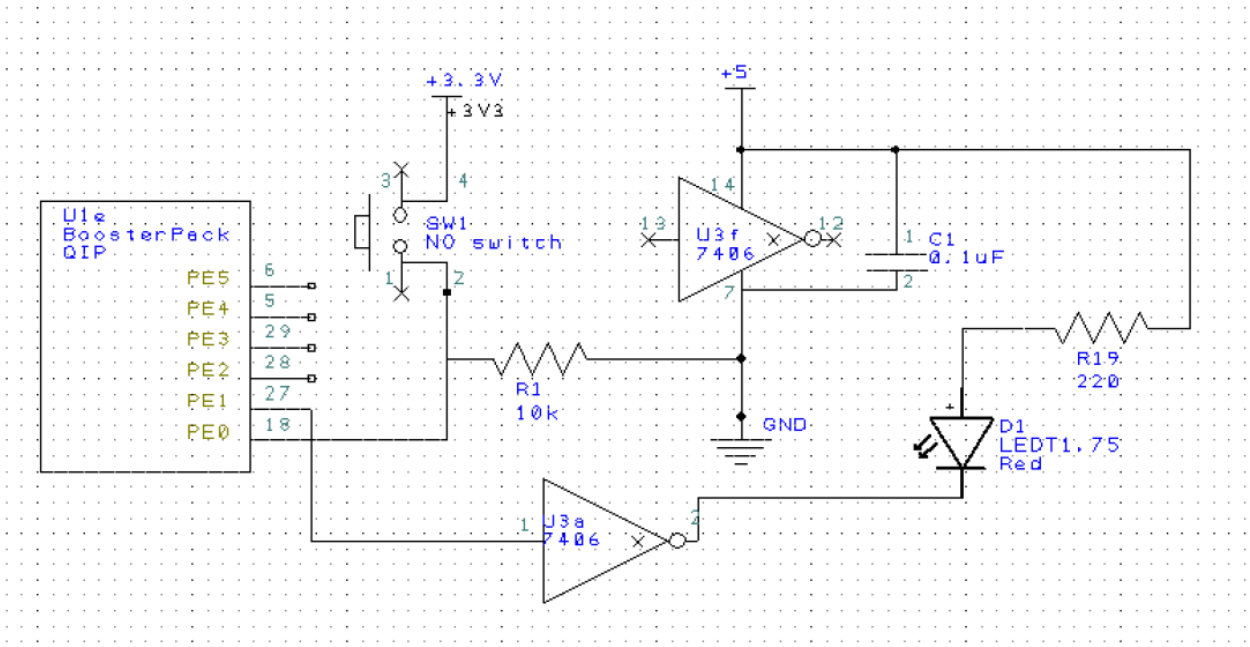
Deliverables

2. Circuit diagram, using PCBArtist, or hand drawn



3. Screenshot like Figure 3.9 showing your debugging in the simulator

4. Switch measurements (Table 3.1)

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9.9k	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.283	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE0}	0.5	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	8	mA	Powered, but switch not pressed $I=V_{PE0}/R1$ (calculated and measured with an ammeter) R19
Input Voltage, V_{PE0}	3.286	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	0.3	mA	Powered and switch pressed $I=V_{PE0}/R1$ (calculated and measured with an ammeter) R1

5. LED measurements (Table 3.2)

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R10	217.7	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	5.02	volts	(measured with voltmeter, *notice that the +5V power is not exactly +5 volts*)
3	TM4C123 Output, V_{PE1} input to 7406	0.08	volts	with **PE1** = 0 (measured with voltmeter)
4	7406 Output, V_{k-} LED k-	3.40	volts	With **PE1** = 0 (measured with voltmeter)
5	LED a+, V_{a+} Bottom side of R10	4.70	volts	with **PE1** = 0 (measured with voltmeter)
6	LED voltage	C- 1.30 M-0.26	volts	calculated as $V_{a+} - V_{k-}$
7	LED current	C-1.02 M-0.0	mA	calculated as $(V_{+5} - V_{a+})/R10$ and measured with an ammeter
8	TM4C123 Output, V_{PE1} input to 7406	3.18	volts	with **PE1** = 1 (measured with voltmeter)

9	7406 Output, V_{k-} LED k-	0.66	volts	with **PE1** = 1(measured with voltmeter)
10	LED a+, V_{a+} Bottom side of R10	2.66	volts	with **PE1** = 1 (measured with voltmeter)
11	LED voltage	C-2.0 M-1.82	volts	calculated as V_{a+} - V_{k-}
12	LED current	C-10.87	mA	calculated as (V_{+5} - V_{a+})/R10 and measured with an ammeter
		M-10.95		

6. Assembly source code of your final program

```
;****************** main.s ***************
; Program written by: Kassandra Smith and Madhumitha Venkataraman
; Date Created: 2/17/2015
; Last Modified: 2/21/2015
; Section Tue 4-5pm     TA: Jenny Chen
; Lab number: 3
; Brief description of the program
;   If the switch is presses, the LED toggles at 8 Hz
; Hardware connections
;  PE0 is switch input  (1 means pressed, 0 means not pressed)
;  PE1 is LED output (1 activates external LED on protoboard)
;Overall functionality of this system is the similar to Lab 2, with five changes:
;1-  the pin to which we connect the switch is moved to PE0,
;2-  you will have to remove the PUR initialization because pull up is no longer needed.
;3-  the pin to which we connect the LED is moved to PE1,
;4-  the switch is changed from negative to positive logic, and
;5-  you should increase the delay so it flashes about 8 Hz.
; Operation
; X         1) Make PE1 an output and make PE0 an input.
; X         2) The system starts with the LED on (make PE1 =1).
; X  3) Wait about 62 ms
; X 4) If the switch is pressed (PE0 is 1), then toggle the LED once, else turn the LED on.
; X  5) Steps 3 and 4 are repeated over and over


GPIO_PORTE_DATA_R       EQU   0x400243FC
GPIO_PORTE_DIR_R        EQU   0x40024400
GPIO_PORTE_AFSEL_R      EQU   0x40024420
GPIO_PORTE_DEN_R        EQU   0x4002451C
GPIO_PORTE_AMSEL_R      EQU   0x40024528
GPIO_PORTE_PCTL_R       EQU   0x4002452C
SYSCTL_RCGCGPIO_R       EQU   0x400FE608
GPIO_PORTE_LOCK_R               EQU   0x40025520
GPIO_PORTE_CR_R                 EQU   0x40025524


              IMPORT  TExaS_Init
      AREA    |.text|, CODE, READONLY, ALIGN=2
      THUMB
      EXPORT  Start
Start
 ; TExaS_Init sets bus clock at 80 MHz
     BL  TExaS_Init ; voltmeter, scope on PD3
; you initialize PE1 PE0


     CPSIE  I    ; TExaS voltmeter, scope runs on interrupts
```

```
;Enable the clock for the port,
;Initialize clock,
LDR R1, =SYSCTL_RCGCGPIO_R            ;activate clock
LDR R0, [R1]
ORR R0, R0, #0x30                      ;set bit 4 to turn on clock
STR R0, [R1] ;put it back
NOP ;wait for stabilization,
NOP

LDR R1, =GPIO_PORTE_LOCK_R ;unlock the lock register
LDR R0, =0x4C4F434B ;unlock GPIO Port E Commit Register
STR R0, [R1]

LDR R1, =GPIO_PORTE_CR_R;enable commit for Port E
MOV R0, #0xFF ;1 means allow access
STR R0, [R1]

LDR R1, =GPIO_PORTE_AMSEL_R ;disable analog functionality
LDR R0, [R1]
BIC R0, #0x03 ;Clear bits 1 and 0
STR R0, [R1]

LDR R1, =GPIO_PORTE_PCTL_R ;configure as GPIO
LDR R0, [R1]
BIC R0, #0x14 ;0 means configure Port E as GPIO
STR R0, [R1]

LDR R1, =GPIO_PORTE_DIR_R ;set direction register
LDR R0, [R1]
ORR R0,#0x02 ;PORTE bit 1 is set to 1
BIC R0,#0x01 ;clear bit 0
STR R0, [R1]

LDR R1, =GPIO_PORTE_AFSEL_R ;disable alternate function select
LDR R0, [R1]
BIC R0, #0x14 ;We don't need the pins' special functions so we set it to 0
STR R0, [R1]

LDR R1, =GPIO_PORTE_DEN_R ;Set DEN so that the bits are useable, Port E digital port
LDR R0, [R1]
ORR R0, #0xFF ;1 enables digital I/O
STR R0, [R1]

LDR R1, =GPIO_PORTE_DATA_R
LDR R0, [R1]
ORR R0, #0x02 ;starting the program with the LED on
STR R0, [R1]
AND R5, R5, #0 ;clearing register 5, to be used as counter for delay
```

```
loop
        ADD R5, #996;
        MUL R5, R5
delay ;delay function
        ADD R5, #-1 ;subtract one from R5                                   ;1 clk cycle
        CMP R5, #0 ;if R5 greater than zero, branch to delay

        BGT delay ;if R5 is equal to zero, proceed                          ;1 clk cycle

        LDR R2, =GPIO_PORTE_DATA_R
        LDR R6, =GPIO_PORTE_DATA_R

        LDR R6, [R6] ;load data from Port E

        AND R6, #0x01 ;masking for bit 0
        CMP R6, #0x01 ;check and see is bit 0 is "1" (switch not pressed)
        BNE turnon ;if switch is not pressed, take the branch

        LDR R6, =GPIO_PORTE_DATA_R
        LDR R6, [R6] ;load data from Port E

        AND R6, #0x02 ;masking for bit 1
        EOR R6, R6, #0x2 ;NOT bit 1
        STR R6, [R2] ;store result back to Port E

        B loop ;Branch back to beginning of loop

turnon ;turning or keeping the LED on
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R2, =GPIO_PORTE_DATA_R
        LDR R1, [R1] ;load data from Port E
        ORR R1, #0x02 ;set bit 1 to "1"
        STR R1, [R2] ;store result back to Port E
        ;end subroutine

    B   loop

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