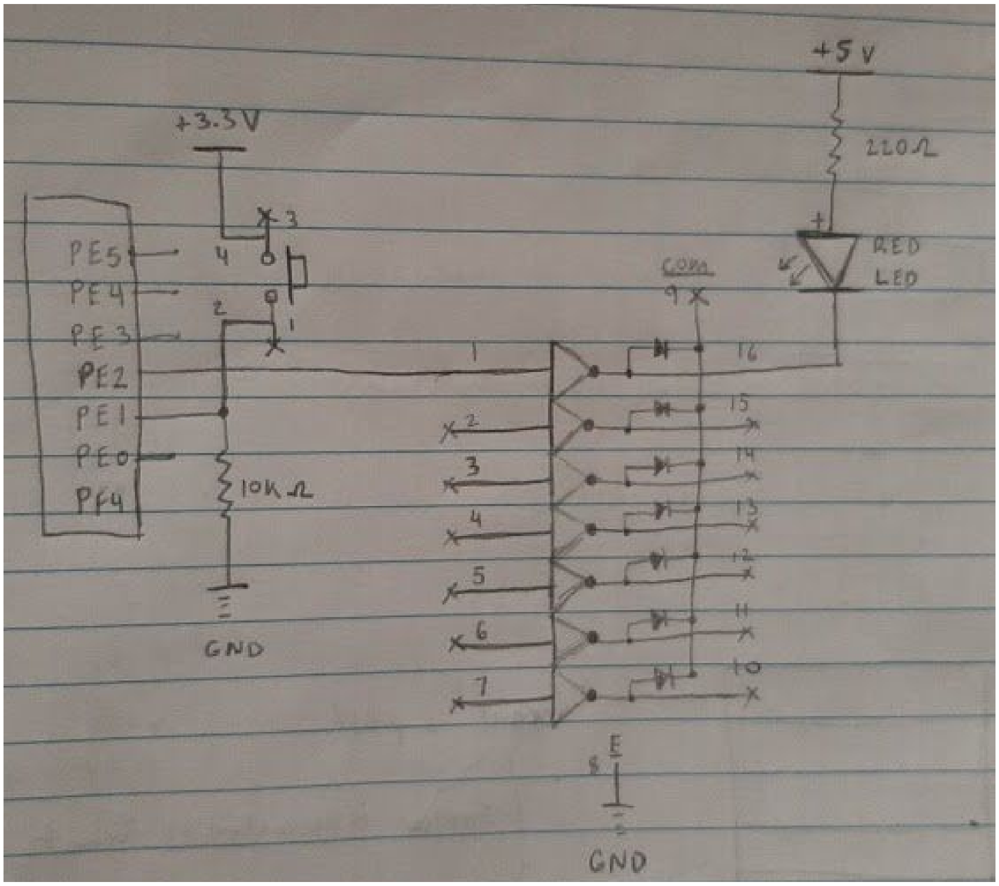
Mervin John & Pranesh Satish

EE 319H

February 19, 2020

Lab 3 Deliverables



Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9.91 kOhms	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.31 V	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	0.0 V	volts	Powered, but with switch not pressed (measured with voltmeter)
	0.00 mA		Powered, but switch not pressed

Resistor current		mA	$I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3.30 V	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	0.33 mA	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

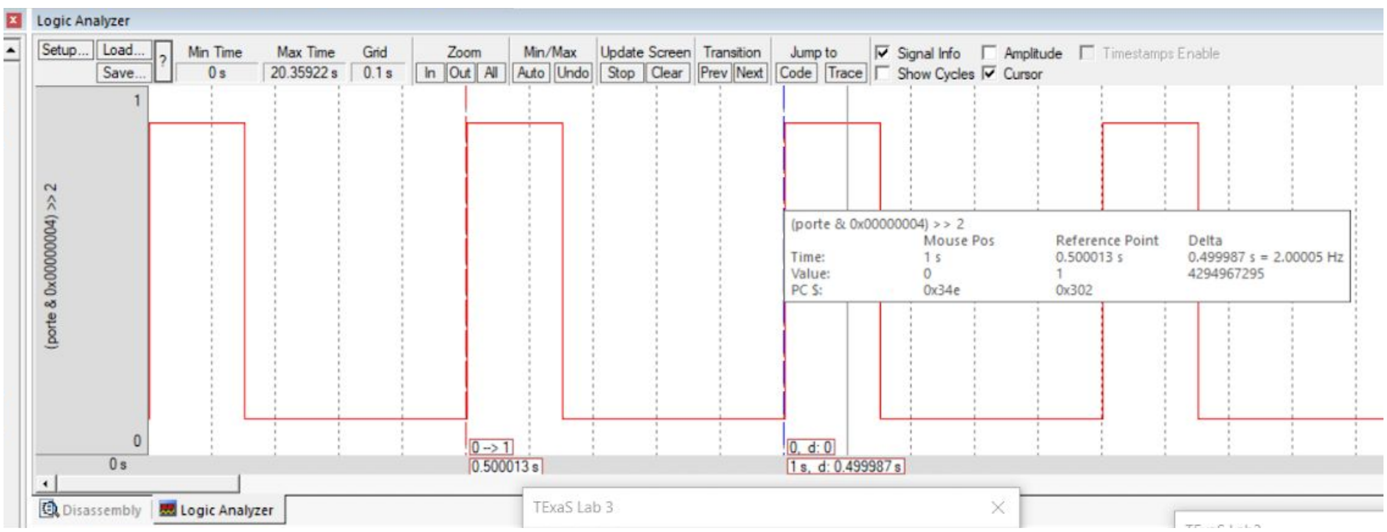
Table 3.1. Switch measurements.

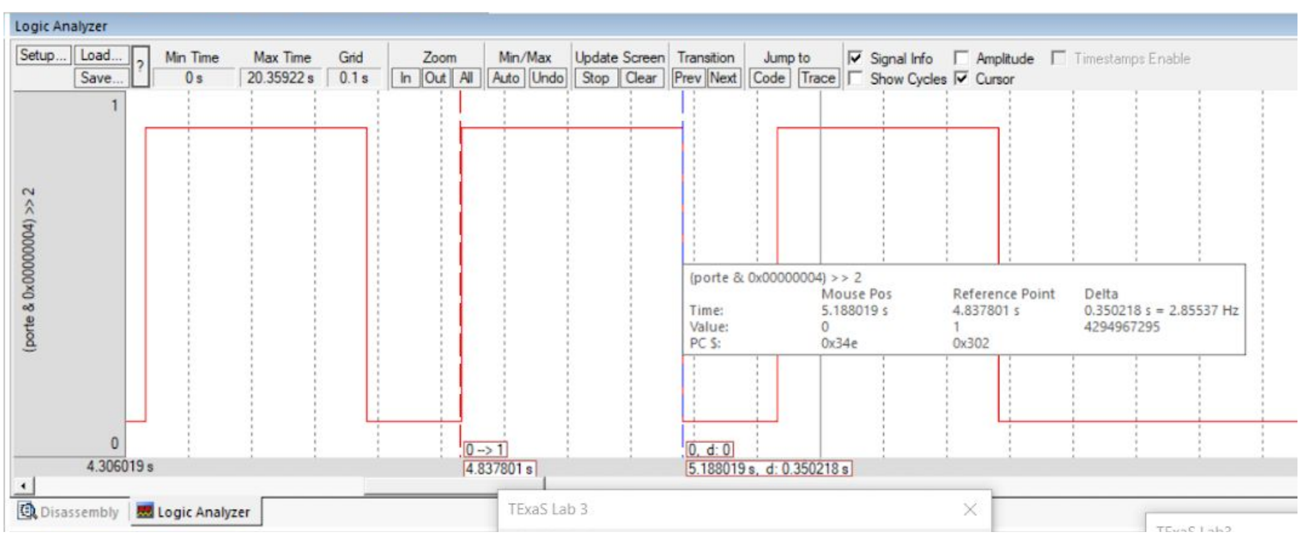
Warning: NEVER INSERT/REMOVE WIRES/CHIPS WHEN THE POWER IS ON.

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	218.5 Ohms	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	5.02 V	volts	(measured with voltmeter relative to ground, *notice that the +5V power is not exactly +5 volts*)
3	TM4C123 Output, V_{PE2} input to ULN2003B	0.0 V	volts	with **PE2** = 0 (measured with voltmeter relative to ground). We call this V_{OL} of the TM4C123.
4	ULN2003B Output, pin 16, V_{k-} LED k-	3.61 V	volts	with **PE2** = 0 (measured with voltmeter relative to ground). This measurement will be weird, because it is floating.
5	LED a+, V_{a+} Bottom side of R19 (anode side of LED)	5.03 V	volts	with **PE2** = 0 (measured with voltmeter relative to ground). This measurement is also weird, because it too is floating.
6	LED voltage	1.42 V	volts	calculated as V_{a+} - V_{k-}
7	LED current (off)	0.0 mA	mA	calculated as (V_{+5} - V_{a+})/R19 and measured with an ammeter

8	TM4C123 Output, V_{PE2} input to ULN2003B	3.23 V	volts	with **PE2** = 1 (measured with voltmeter relative to ground). We call this V_{OH} of the TM4C123.
9	ULN2003B Output pin 16, V_{k-} LED k-	0.724 V	volts	with **PE2** = 1 (measured with voltmeter relative to ground). We call this V_{OL} or $V_{CE(sat)}$ of the ULN2003B.
10	LED a+, V_{a+} Bottom side of R19 (anode side of LED)	2.67 V	volts	with **PE2** = 1 (measured with voltmeter relative to ground)
11	LED voltage	1.94 V	volts	calculated as V_{a+} - V_{k-}
12	LED current (on)	10.76 mA 19.48 mA	mA	calculated as (V_{+5} - V_{a+})/R19 and measured with an ammeter

Table 3.2. LED measurements (assuming the 220 Ω resistor is labeled R19 in FIgure 3.8).





Mervin John & Pranesh Satish

EE 319H

February 19, 2020

```
+Start
 ; TExaS_Init sets bus clock at 80 MHz
    BL  TExaS_Init ; voltmeter, scope on PD3
 ; Initialization goes here
    LDR R0, =SYSCTL_RCGCGPIO_R
       LDRB R1, [R0]
       ORR R1, #0x20
       STRB R1, [R0]

       NOP
       NOP

       LDR R0, =GPIO_PORTF_LOCK_R
       LDR R1, =GPIO_LOCK_KEY
       STR R1, [R0]
       LDR R0, =GPIO_PORTF_CR_R
       LDR R1, [R0]
       ORR R1, #0xFF
       STR R1, [R0]

       LDR R0, =GPIO_PORTF_DIR_R
       LDR R1, [R0]
       AND R1, #0xFF
       STR R1, [R0]

       LDR R0, =GPIO_PORTF_DEN_R
       LDR R1, [R0]
       ORR R1, #0x10
       STR R1, [R0]

       LDR R0, =GPIO_PORTF_PUR_R
       LDR R1, [R0]
       ORR R1, #0x10
       STR R1, [R0]

       LDR R0, = SYSCTL_RCGCGPIO_R ;intialize port E clock
       LDR R1, [R0]
       ORR R1, #0x10
       STR R1, [R0]
       NOP
       NOP
       LDR R0, = GPIO_PORTE_DIR_R ;port E2 set to output
       MOV R1, #0x04
       STR R1, [R0]
       LDR R0, = GPIO_PORTE_DEN_R ;temp digital enable to test on sim
       MOV R1, #0x06
       STR R1, [R0]
```

```
    CPSIE  I    ; TExaS voltmeter, scope runs on interrupts
      ;these are some registers I used because I was too lazy to make variables

      LDR R5, = 3333333 ;The base length of a cycle
      LDR R9, = 2999000 ;The max value we want to let a high run
      LDR R6, = 666666    ;amount incrementing duration of high
      LDR R4, = 1000000  ;length of high in the beginning

loop
; main engine goes here
;FACT: (20M) cycles == 1s

      LDR R0, = GPIO_PORTE_DATA_R
      LDR R1, [R0]
      LDR R2, [R0]          ;sees if PE1 is turned on or off
      BFC R2, #1,#1
      CMP R2, R1
      BMI change            ;if CMP is negative that means PE1 is turned on, so then branch change the cycle
return
    ;LDR R0, = GPIO_PORTE_DATA_R
      LDR R1, [R0]
      ORR R1, #0x04
      STR R1, [R0]
      BFC R2, #0, #31
      ADD R2, R4, #0

      LDR R10,= GPIO_PORTF_DATA_R
      LDR R11, [R10]
      ORR R11, #0x10
      LDR R12, [R10] ;check PF4 turned "on"
      CMP R11, R12
      BMI breathe
Delay1
      ADD R8, R8, #0
      LDR R1, [R0]
      LDR R7, [R0]          ;sees if PE1 is turned on or off
      BFC R7, #1,#1
      CMP R7, R1
      BMI change
      SUBS R2, R2, #1
      BNE Delay1

      LDR R10,= GPIO_PORTF_DATA_R
      LDR R11, [R10]
      ORR R11, #0x10
      LDR R12, [R10] ;check PF4 turned "on"
      CMP R12, R11
      BMI breathe

      LDR R1, [R0]
      BFC R1, #2, #2
      STR R1, [R0]
```

```
      BFC R3, #0, #31
      SUBS R3, R5, R4
Delay2
    ADD R8, R8, #0
    LDR R1, [R0]
      LDR R7, [R0]           ;sees if PE1 is turned on or off
      BFC R7, #1,#1
      CMP R7, R1
      BMI change
      SUBS R3, R3, #1
      BNE Delay2

      LDR R10,= GPIO_PORTF_DATA_R
      LDR R11, [R10]
      ORR R11, #0x10
      LDR R12, [R10] ;check PF4 turned "on"
      CMP R12, R11
      BMI breathe
    B    loop


change
      CMP R4, R9             ;check if R4 is already at 3M
      BPL Zero
      ADD R4, R4, R6 ;R4+=2M

loop2     ;sits here till PE1 turned off
     LDR R0, = GPIO_PORTE_DATA_R
     LDR R1, [R0]
     LDR R2, [R0]            ;sees if PE1 is turned on or off
     BFC R2, #1,#1
     CMP R2, R1
     BMI loop2    ;if CMP is negative that means PE1 is turned on, so then branch change the cycle
      B return

Zero
      LDR R4, =1000000    ;1M
      ;B return

loop3 ;its here till PE1 turned off
     LDR R0, = GPIO_PORTE_DATA_R
     LDR R1, [R0]
     LDR R2, [R0]            ;sees if PE1 is turned on or off
     BFC R2, #1,#1
     CMP R2, R1
     BMI loop3               ;if CMP is negative that means PE1 is turned on, so then branch change the cycle


      B return

breathe
     PUSH {R4,R5}
     PUSH {R6,R7}
```

```
    PUSH {R8,R9}
    PUSH {R10,R11}
    PUSH {R12,R0}
   ;PUSH{R4-R12,LR}
    LDR R4,= 20000 ;REMEBER TO PUT R4-R9 ON STACK!!!!!!
    LDR R5,= 1000
    LDR R6,= 0
    LDR R8,= 0x30

     LDR R10,= GPIO_PORTF_DATA_R
     LDR R11, [R10]
     BFC R11, #4,#4
     LDR R12, [R10] ;check PF4 turned "OFF"
     CMP R12, R11
     BMI stopBreathing

bLoop
    ADD R6, R6, R5
    ORR R7, #0x00
    ORR R7, #0x00
    CMP R6, R4
    BPL bLoopZero

    LDR R10,= GPIO_PORTF_DATA_R
     LDR R11, [R10]
     BFC R11, #4,#4
     LDR R12, [R10] ;check PF4 turned "OFF"
     CMP R12, R11
     BMI stopBreathing

InnerLoop       ; this makes it so the light can keep looping at a certain brightness for a little longer
    ADD R7, R6, #0
    LDR R0,= GPIO_PORTE_DATA_R
    LDR R1, [R0]
    ORR R1, #0x04
    STR R1, [R0]
bLoopDelay1
    SUBS R7, R7, #1
    BNE bLoopDelay1
    ORR R7, #0x00
    SUBS R7, R4, R6
    LDR R0,= GPIO_PORTE_DATA_R
    LDR R1, [R0]
    BFC R1, #2, #2
    STR R1, [R0]

     LDR R10,= GPIO_PORTF_DATA_R
     LDR R11, [R10]
     BFC R11, #4,#4
     LDR R12, [R10] ;check PF4 turned "OFF"
     CMP R11, R12
     BMI stopBreathing
```

```
bLoopDelay2
      SUBS R7, R7, #1
      BNE bLoopDelay2
      ORR R7, #0x00
      SUBS R8, R8, #1
      BNE InnerLoop
      LDR R8,= 0x30
      B bLoop

       LDR R10,= GPIO_PORTF_DATA_R
       LDR R11, [R10]
       BFC R11, #4,#4
       LDR R12, [R10] ;check PF4 turned "OFF"
       CMP R11, R12
       BMI stopBreathing

bLoopZero
      LDR R4,= 20000
      LDR R5,= 1000
      LDR R6,= 0
      LDR R9,= 0
      LDR R8,= 0x30

bLoopZero1
      ADD R9, R9, R5
      SUBS R4, R4, R5
      CMP R6, R4
      BPL breathe
bLoopInner          ;makes the light stay on longer
      SUBS R7, R4, R6 ;ADD R7, R4, #0 flip with line 267
      LDR R0,= GPIO_PORTE_DATA_R
      LDR R1, [R0]
      ORR R1, #0x04
      STR R1, [R0]
bLoopDelay3
      SUBS R7, R7, #1
      BNE bLoopDelay3

       LDR R10,= GPIO_PORTF_DATA_R
       LDR R11, [R10]
       BFC R11, #4,#4
       LDR R12, [R10] ;check PF4 turned "OFF"
       CMP R11, R12
       BMI stopBreathing


      ADD R7, R9, #0 ;SUBS R7, R4, R6 flip with line 257
      LDR R0,= GPIO_PORTE_DATA_R
      LDR R1, [R0]
      BFC R1, #2, #2
      STR R1, [R0]
bLoopDelay4
      SUBS R7, R7, #1
```

```
        BNE bLoopDelay4

         LDR R10,= GPIO_PORTF_DATA_R
         LDR R11, [R10]
         BFC R11, #4,#4
         LDR R12, [R10] ;check PF4 turned "OFF"
         CMP R11, R12
         BMI stopBreathing

        SUBS R8, R8, #1
        BNE bLoopInner
        LDR R8,= 0x30
        B bLoopZero1
stopBreathing
        POP{R0,R12}
        POP{R11,R10}
        POP{R9,R8}
        POP{R7,R6}
        POP{R5,R4}
        B Start

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

Demonstration

(both partners must be present, and demonstration grades for partners may be different)

You will show the TA your program operation on the actual TM4C123 board. The TA may look at your data and expect you to understand how the data was collected and how the switch and LEDs work. Also be prepared to explain how your software works and to discuss other ways the problem could have been solved. Why the ULN2003B was used to interface the LED? I.e., why did we not connect the LED directly to the TM4C123. What would the flashing LED “look” like if the delay were 1ms? How would you modify the software to change the rate at which LED flickers? What operating point (voltage, current) exists when the LED is on? Sketch the approximate current versus voltage curve of the LED. Explain how you use the resistor value to select the operating point. What is the difference between a positive logic and negative logic interface for the switch or the LED? We may test to see if you can measure voltage, current and/or resistance with your meter (so bring your meter to the demonstration).

Please make note of which TA checked you out. The name of the TA will greatly help you when resolving any grading issues later.

Deliverables

(Items 2, 3, 4, 5, and 6 are one pdf file uploaded to Canvas, have this file open during demo.) **Your software solution and this pdf must be committed to GitHub.**

1. Lab 3 grading sheet (TA prints). You fill out the information at the top.
2. Circuit diagram (hand-drawn or optionally using PCB Artist)
3. Screenshots like Figure 3.10a,b showing your debugging in the simulator
4. Switch measurements (Table 3.1)
5. LED measurements (Table 3.2)

6. Assembly source code of your final program
7. Optional Feedback : http://goo.gl/forms/rBsP9NTxSy