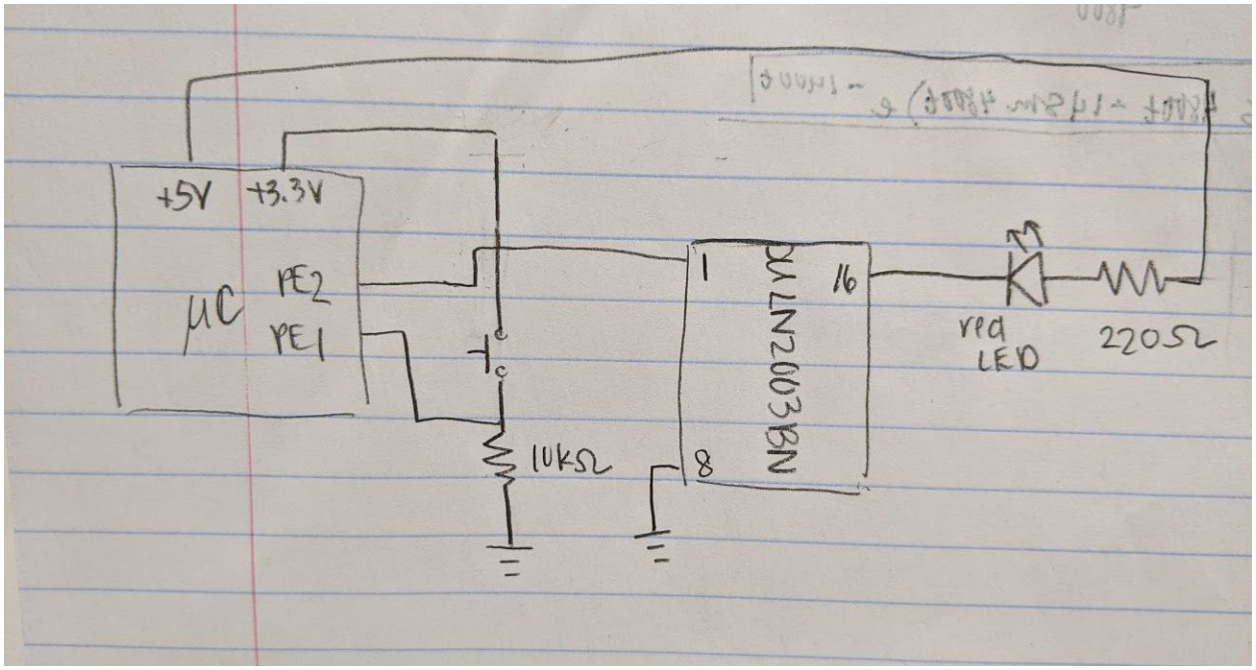
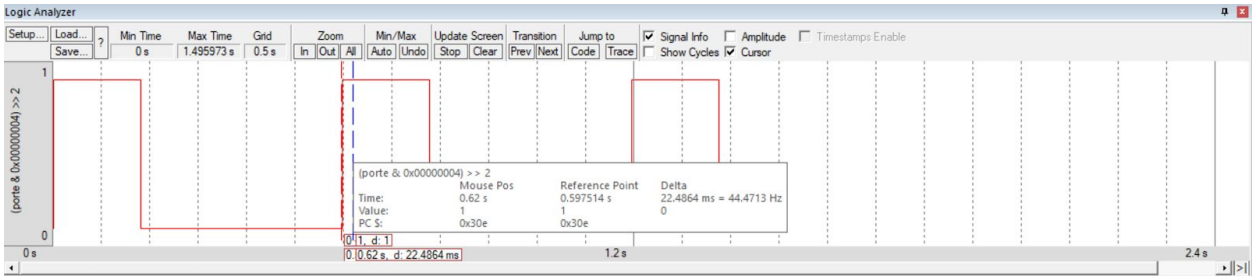


(Logic Analyzer is for code to get accurate results on the real board)

30%



70%

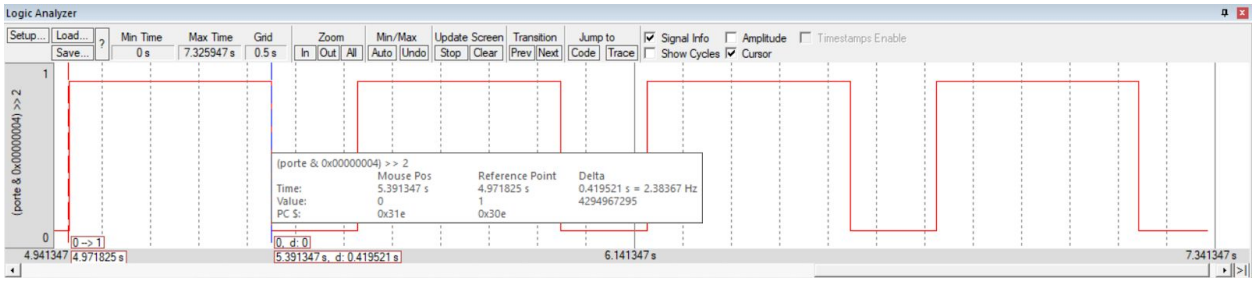


Table 3.1

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9690	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.291	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	0	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	0	mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3.288	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	0.33	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

Table 3.2

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	217.6	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	5.08	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	0.0008	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-	3.775	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)	5.08	volts	with **PE2** = 0 (measured with voltmeter relative to ground). This measurement is also weird, because it too is floating.
6	LED voltage	1.305	volts	calculated as $V_{a+} - V_{k-}$
7	LED current (off)	0	mA	calculated as $(V_{+5} - V_{a+})$/R19 and measured with an ammeter
8	TM4C123 Output, V_{PE2} input to ULN2003B	3.214	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-	0.722	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)	2.56	volts	with **PE2** = 1 (measured with voltmeter relative to ground)
11	LED voltage	1.838	volts	calculated as $V_{a+} - V_{k-}$
12	LED current (on)	11.5 9.68	mA	calculated as $(V_{+5} - V_{a+})$/R19 and measured with an ammeter

```
;****************** main.s ***************
; Program written by: Alison Jin, Roberto Reyes
; Date Created: 2/4/2017
; Last Modified: 2/17/2020
; Brief description of the program
;   The LED toggles at 2 Hz and a varying duty-cycle
; Hardware connections (External: One button and one LED)
;  PE1 is Button input  (1 means pressed, 0 means not pressed)
;  PE2 is LED output (1 activates external LED on protoboard)
;  PF4 is builtin button SW1 on Launchpad (Internal)
;        Negative Logic (0 means pressed, 1 means not pressed)
; Overall functionality of this system is to operate like this
;   1) Make PE2 an output and make PE1 and PF4 inputs.
;   2) The system starts with the the LED toggling at 2Hz,
;        which is 2 times per second with a duty-cycle of 30%.
;        Therefore, the LED is ON for 150ms and off for 350 ms.
;   3) When the button (PE1) is pressed-and-released increase
;        the duty cycle by 20% (modulo 100%). Therefore for each
;        press-and-release the duty cycle changes from 30% to 70% to 70%
;        to 90% to 10% to 30% so on
;   4) Implement a "breathing LED" when SW1 (PF4) on the Launchpad is pressed:
;        a) Be creative and play around with what "breathing" means.
;        An example of "breathing" is most computers power LED in sleep mode
;        (e.g., https://www.youtube.com/watch?v=ZT6siXyIjvQ).
;        b) When (PF4) is released while in breathing mode, resume blinking at 2Hz.
;        The duty cycle can either match the most recent duty-
;        cycle or reset to 30%.
;        TIP: debugging the breathing LED algorithm using the real board.
; PortE device registers
GPIO_PORTE_DATA_R  EQU 0x400243FC
GPIO_PORTE_DIR_R   EQU 0x40024400
GPIO_PORTE_AFSEL_R EQU 0x40024420
GPIO_PORTE_DEN_R   EQU 0x4002451C
; PortF device registers
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_LOCK_KEY     EQU 0x4C4F434B  ; Unlocks the GPIO_CR register
SYSCTL_RCGCGPIO_R  EQU 0x400FE608
```

```
count10         EQU 0x123bf8 ; 1,195,000
hundredms       EQU 0x2477f0 ; 2,390,000
countOn         EQU 0x36ee80 ; 3,600,000
countOff        EQU 0x7f6930 ; 8,350,000
ninetypercent   EQU 0xa41bba ; 10,755,000
onems           EQU 0x4e20  ;20,000 - adding 10% duty cycle for breathing LED
totalms         EQU 0x30d40 ;0.01s -> 200,000 = 100%
breatheIterations  EQU 0xA

        IMPORT  TExaS_Init
        THUMB
        AREA  DATA, ALIGN=2

;global variables go here
        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
 ; Initialization goes here
 ;turn on clock for Port E and F
  LDR R2, =countOn
  LDR R3, =countOff
  LDR R0, =SYSCTL_RCGCGPIO_R
  LDRB R1, [R0]
  ORR R1, #0x30
  STRB R1, [R0]

  ;wait for clock to stabilize
  NOP
  NOP

  ;input/output for Port E
  LDR R0, =GPIO_PORTE_DIR_R
  LDR R1, [R0]
  AND R1, #~0x2
  ORR R1, #0x4
  STR R1, [R0]

  ;digital enable
```

```
  LDR R0, = GPIO_PORTE_DEN_R
  LDR R1, [R0]
  ORR R1, #0x6
  STR R1, [R0]

  ;port f
  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]

  ;port f input
  LDR R0, =GPIO_PORTF_DIR_R
  LDR R1, [R0]
  AND R1, #0xEF
  STR R1, [R0]

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

  ;engage internal pull up resistor for PF4
  LDR R0, =GPIO_PORTF_PUR_R
  LDR R1, [R0]
  ORR R1, #0x10
  STR R1, [R0]
        CPSIE  I        ; TExaS voltmeter, scope runs on interrupts

loop

; main engine goes here
  BL check
  BL lit
  BL high
  BL unlit
  BL low

  MOV R3, R0
```

```
        B       loop



;1 cycle = 0.5 seconds

high SUBS R2, R2, #0x01
  BNE high
  BX LR

low SUBS R3, R3, #0x01
 BNE low
 BX LR

lit LDR R0, =GPIO_PORTE_DATA_R
 LDRB R1, [R0]
 ORR R1, #0x4
 STRB R1, [R0]
 MOV R0, R2
 BX LR

unlit
 MOV R2, R0
 LDR R0, =GPIO_PORTE_DATA_R
 LDRB R1, [R0]
 AND R1, #~0x4
 STRB R1, [R0]
 MOV R0, R3
 BX LR

check
 LDR R0, =GPIO_PORTF_DATA_R ;check PF4 first
 LDR R1, [R0]
 AND R1, #0x10
 LSR R1, #4
 CMP R1, #0 ;negative logic (0 = pressed)
 BEQ breathe

 ;check PE1
 LDR R0, =GPIO_PORTE_DATA_R
 LDR R1, [R0]
 AND R1, #0x2
 LSR R1, #1
```

```
 CMP R1, #1 ;positive logic
 BEQ plus
back
 BX LR

plus
 LDR R0, =GPIO_PORTE_DATA_R
 LDRB R1, [R0]
 AND R1, #~0x4
 STRB R1, [R0]
 LDR R12, =ninetypercent
 CMP R12, R2
 BLE ten
 LDR R12, =hundredms
 ADD R2, R2, R12 ; Adds 100ms
 SUB R3, R3, R12 ; Subs 100ms
release
 LDR R0, =GPIO_PORTE_DATA_R
 LDR R1, [R0]
 AND R1, #0x2
 LSR R1, #1
 CMP R1, #1
 BEQ release
 B back

ten LDR R2, =count10
 LDR R3, =ninetypercent
 B back

breathe
 PUSH {R4,LR}
repeat
 LDR R12, =breatheIterations
 MOV R4, R12
 SUB R4, #1
 LDR R2, =onems
 LDR R3, =totalms
 SUB R3, R2

increase
 BL lit
 BL high
 BL unlit
```

```
 BL low
 MOV R3, R0
 SUBS R12, #0x1
 BNE increase

 SUBS R4, #0x1
 BEQ switch
 BL increment
 LDR R12, =breatheIterations
 B increase

switch
 LDR R12, =breatheIterations
 SUB R4, R12, #1

decrease
 BL lit
 BL high
 BL unlit
 BL low
 MOV R3, R0
 SUBS R12, #0x1
 BNE decrease

 SUBS R4, #0x1
 BEQ Fcheck
 BL decrement
 LDR R12, =breatheIterations
 B decrease

Fcheck
 LDR R0, =GPIO_PORTF_DATA_R ;check PF4 again
 LDR R1, [R0]
 AND R1, #0x10
 LSR R1, #4
 CMP R1, #0 ;negative logic (0 = pressed)
 BEQ repeat

 LDR R2, =countOn
 LDR R3, =countOff
 POP {R4,LR}
 BX LR
```

```
increment
 LDR R0, =onems
 ADD R2, R0
 SUB R3, R0
 BX LR

decrement
 LDR R0, =onems
 SUB R2, R0
 ADD R3, R0
 BX LR

 ALIGN          ; make sure the end of this section is aligned

          END    ; end of file
```