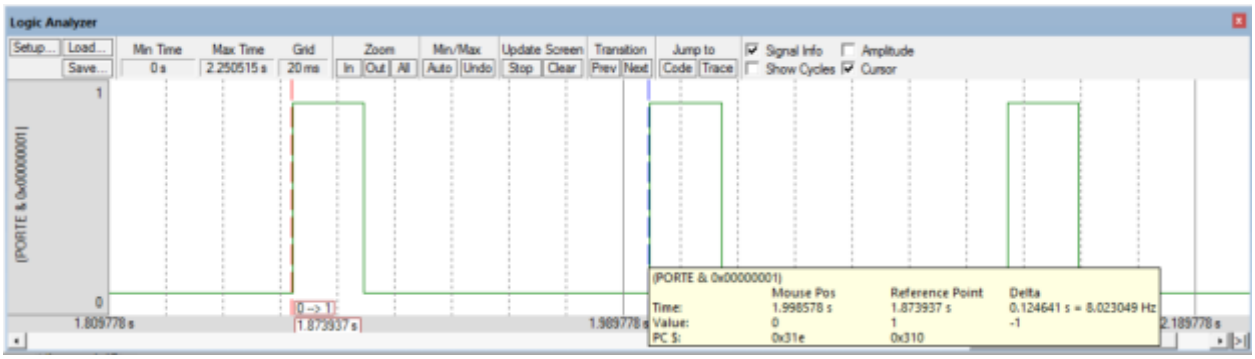
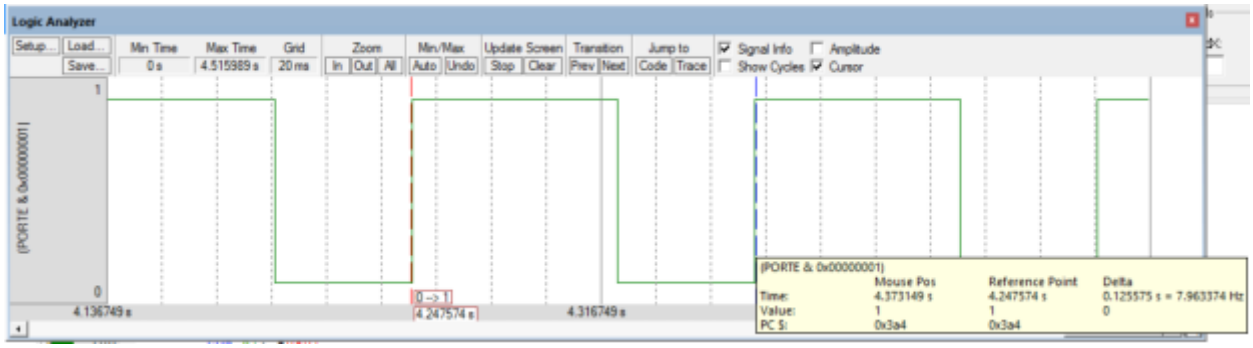
Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9.93k	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.27	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	13.9m	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	Calculated: ~0mA Measured: 0mA	mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3.27	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	Calculated: 0.327 Measured: 0.33	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

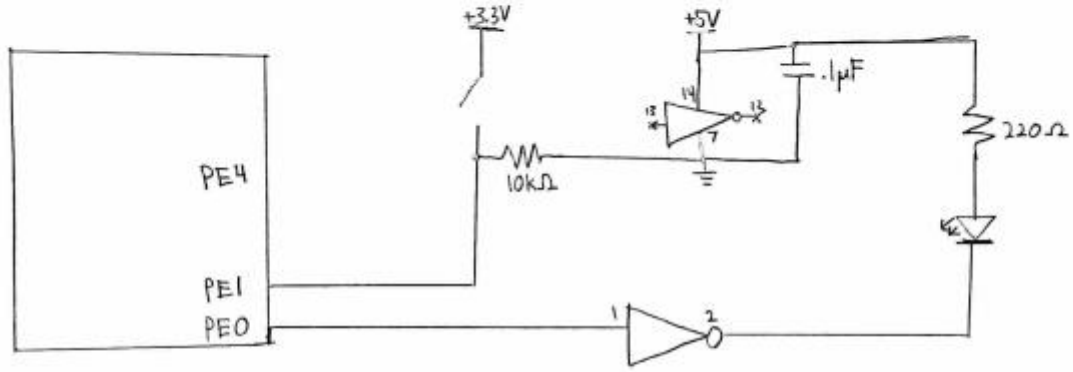
Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	217	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	5.12	volts	(measured with voltmeter relative to ground, *notice that the +5V power is not exactly +5 volts*)

3	TM4C123 Output, V_{PE0} input to 7406	84.9m	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
4	7406 Output, V_{k-} LED k-	4.94	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
5	LED a+, V_{a+} Bottom side of R19	5.11	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
6	LED voltage	0.17	volts	calculated as $V_{a+} - V_{k-}$
7	LED current	Calcultaed:0.77 Measured: 0	mA	calculated as $(V_{+5} - V_{a+})$/R19 and measured with an ammeter
8	TM4C123 Output, V_{PE0} input to 7406	72.6m	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
9	7406 Output, V_{k-} LED k-	156.5m	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
10	LED a+, V_{a+} Bottom side of R19	2.08	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
11	LED voltage	1.924	volts	calculated as $V_{a+} - V_{k-}$
12	LED current	Calculated:8.74 Measured:10.05	mA	calculated as $(V_{+5} - V_{a+})$/R19 and measured with an ammeter


Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 2.250515 s
Grid 20 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Amplitude
Show Cycles
Cursor
(PORTE & 0x00000001)
1
0
0 -> 1
1.809778 s
1.873937 s
1.989778 s
2.189778 s
(PORTE & 0x00000001)
Mouse Pos
Reference Point
Delta
Time: 1.998578 s 1.873937 s 0.124641 s = 8.023049 Hz
Value: 0 1 -1
PC $: 0x31e 0x310

;****************** main.s ***************


Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 4.515989 s
Grid 20 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Amplitude
Show Cycles
Cursor
(PORTE & 0x00000001)
1
0
0 -> 1
4.136749 s
4.247574 s
4.316749 s
(PORTE & 0x00000001)
Mouse Pos
Reference Point
Delta
Time: 4.373149 s 4.247574 s 0.125575 s = 7.963374 Hz
Value: 1 1 0
PC $: 0x3a4 0x3a4


+3.3V
+5V
.1μF
220Ω
10kΩ
PE4
PE1
PE0

```
; Program written by: Michael Blume, Jordan Pamatmat
; Date Created: 2/4/2017
; Last Modified: 2/12/2017
; Brief description of the program
;   The LED toggles at 8 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)
;  PE0 is LED output (1 activates external9 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 PE0 an output and make PE1 and PF4 inputs.
;   2) The system starts with the the LED toggling at 8Hz,
;      which is 8 times per second with a duty-cycle of 20%.
;      Therefore, the LED is ON for (0.2*1/8)th of a second
;      and OFF for (0.8*1/8)th of a second.
;   3) When the button on (PE1) is pressed-and-released increase
;      the duty cycle by 20% (modulo 100%). Therefore for each
;      press-and-release the duty cycle changes from 20% to 40% to 60%
;      to 80% to 100%(ON) to 0%(Off) to 20% to 40% 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 8Hz.
;         The duty cycle can either match the most recent duty-
;         cycle or reset to 20%.
;      TIP: debugging the breathing LED algorithm and feel on the simulator is impossible.
; 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
DELAY10   EQU  0x00003D000 ; this is 12.5ms which is 10% of a 8HZ frequency
DELAY80   EQU  0x0001E8000
DELAY20   EQU  0x00007A000 ; this is 25ms which is 20% of a 8HZ frequency
DELAY40   EQU  0X0000F4000
DELAY60   EQU  0x00016E000
DELAY5    EQU  0x0000030CC  ; this is 0.625ms which is 5% of a 80HZ frequency
DELAY100  EQU  0x00003CFF1
DELAY1    EQU     0x0000009C3
DELAY1001  EQU  0x00003D02D


SYSCTL_RCGCGPIO_R  EQU 0x400FE608
      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
```

```
  LDR R0, =SYSCTL_RCGCGPIO_R  ; Turn on the clock for Port E and Port F
  LDR R1, [R0]
  ORR R1, #0x30
  STR R1, [R0]
  NOP
  NOP

  LDR R0, =GPIO_PORTE_DIR_R
  LDR R1, [R0]
  BIC R1, #0x02   ; Make PE1 an input = 0
  ORR R1, #0x01 ; Make PE0 an output = 1
  STR R1, [R0]

  LDR R0, =GPIO_PORTF_DIR_R
  LDR R1, [R0]
  BIC R1, #0x10 ; Make PF4 an input = 0
  STR R1, [R0]

  LDR R0, =GPIO_PORTE_AFSEL_R
  LDR R1, [R0]
  BIC R1, #0x03  ; Turn off alternate functions for Port E
  STR R1, [R0]

  LDR R0, =GPIO_PORTF_AFSEL_R
  LDR R1, [R0]
  BIC R1, #0x10  ; Turn off alternate functions for Port F
  STR R1, [R0]

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

  LDR R0, =GPIO_PORTE_DEN_R
  LDR R1, [R0]
  ORR R1, #0x03   ; Enable PE1,PE0
  STR R1, [R0]

  LDR R0, =GPIO_PORTF_DEN_R
  LDR R1, [R0]
  ORR R1, #0x10   ; Enable PF4
  STR R1, [R0]

  LDR R0, =GPIO_PORTE_DATA_R
  LDR R12, =GPIO_PORTF_DATA_R

   CPSIE  I    ; TExaS voltmeter, scope runs on interrupts

; THIS IS THE LOOP THAT OF A 20% DUTY CYCLE AT 8HZ
LOOP20
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY20
DE201  SUBS R2, R2, #1
  BNE DE201
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY80
```

```
DE202 SUBS R2, R2, #1
  BNE DE202

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT20
  BL BREATHING
NEXT20
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP20
  BL CLEAR
  B LOOP40

; THIS IS A LOOP OF 40% DUTY CYCLE AT 8HZ
LOOP40
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY40
DE401  SUBS R2, R2, #1
  BNE DE401
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY60
DE402 SUBS R2, R2, #1
  BNE DE402

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT40
  BL BREATHING
NEXT40
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP40
  BL CLEAR
  B LOOP60

; THIS IS A LOOP OF 60% DUTY CYCLE AT 8HZ
LOOP60
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY60
DE601  SUBS R2, R2, #1
  BNE DE601
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY40
DE602 SUBS R2, R2, #1
```

```
  BNE DE602

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT60
  BL BREATHING
NEXT60
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP60
  BL CLEAR
  B LOOP80

; THIS IS A LOOP OF 80% DUTY CYCLE AT 8HZ
LOOP80
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY80
DE801  SUBS R2, R2, #1
  BNE DE801
  EOR R1, #0x01
  STR R1, [R0]
  LDR R2, =DELAY20
DE802 SUBS R2, R2, #1
  BNE DE802

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT80
  BL BREATHING
NEXT80
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP80
  BL CLEAR
  B LOOP100

; THIS IS A LOOP OF 100% DUTY CYCLE AT 8HZ
LOOP100
  LDR R1, [R0]
  ORR R1, #0x01
  STR R1, [R0]

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT100
  BL BREATHING
NEXT100
```

```
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP100
  BL CLEAR
  B LOOP0

; THIS IS A LOOP OF 0% DUTY CYCLE AT 8HZ
LOOP0
  LDR R1, [R0]
  BIC R1, #0x01
  STR R1, [R0]

  LDR R11, [R12]
  AND R11, #0x10
  CMP R11, #0x10
  BEQ NEXT0
  BL BREATHING
NEXT0
  LDR R1, [R0]
  AND R5, R1, #0x02 ; R5 is pressed will be a 1, if not pressed it will be a 0
  ORR R6, R6, R5 ; if it is pressed put 1 in R6,
  EOR R8, R5, R6 ; if turned on then off this will be a 1
  CMP R8, #0x02
  BNE LOOP0
  BL CLEAR
  B LOOP20

CLEAR
  AND R6, R6, #0
  BX LR

BREATHING ; R7, R9, AND R10 ARE NOT USED ANYWHERE ELSE IN THE PROGRAM
  PUSH {LR, R8}
  LDR R10, =DELAY1
  LDR R9, =DELAY1001
  AND R8, R8, #0
  MOV R7, #1

; implement breating, increase by 1% everytime until reaches 100%, then decrease 1% until reaches 0%
INCREASE
  PUSH {R7, R9}
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
DEB1  SUBS R7, R7, #1
  BNE DEB1
  EOR R1, #0x01
  STR R1, [R0]
DEB2 SUBS R9, R9, #1
  BNE DEB2

  POP {R7, R9}
  LDR R11, [R12] ; check to see if it is not pressed anymore
  AND R11, #0x10
  CMP R11, #0x10
```

```
  BNE NEXTI
  POP {LR, R8}
  BX LR    ; jump back to the program

NEXTI
  CMP R8, #100
  BEQ DEC   ; IF WE REACH 100% WE NEED TO GO THE OTHER WAY
INC  ADD R8, R8, #1  ; COUNTER TO SEE IF WE REACHED 100%
  ADD R7, R7, R10
  SUB R9, R9, R10
  B INCREASE

DECREASE
  PUSH {R7, R9}
  LDR R1, [R0]
  EOR R1, #0x01
  STR R1, [R0]
DEB3  SUBS R7, R7, #1
  BNE DEB3
  EOR R1, #0x01
  STR R1, [R0]
DEB4 SUBS R9, R9, #1
  BNE DEB4

  POP {R7, R9}
  LDR R11, [R12] ; check to see if it is not pressed
  AND R11, #0x10
  CMP R11, #0x10
  BNE NEXTD
  POP {LR, R8}
  BX LR

NEXTD
  CMP R8, #0
  BEQ INC
DEC  SUB R8, R8, #1
  ADD R9, R9, R10
  SUB R7, R7, R10
  B DECREASE

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