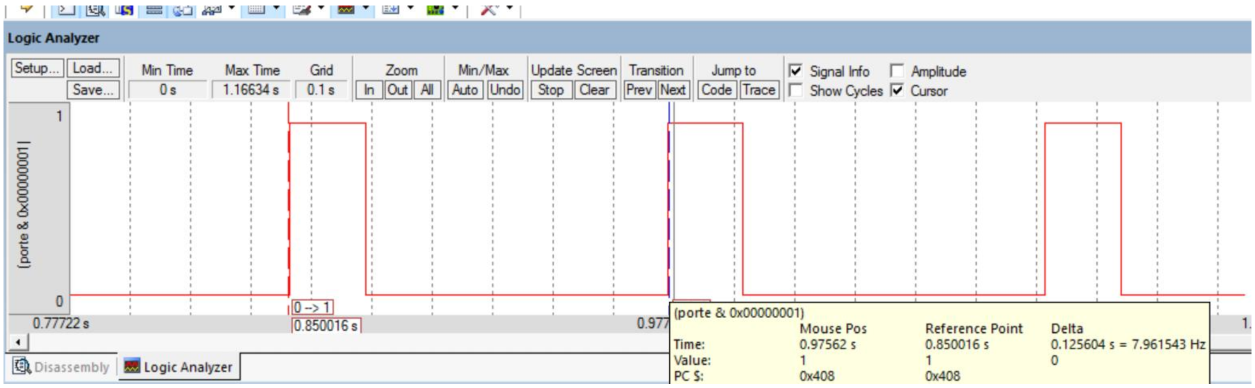
	MIN/Only Value	MAX
Resistance of the 10K ohm resistor	10K ohms	
Supply Voltage	3.3 V	
Input Voltage (not pressed)	0 V	
Resistor Current (not pressed)	0 A	
Input Voltage (pressed)	3.29 V	
Resistor Current (pressed)	0.3 mA	
Resistance of the 220 ohm resistor	220 ohms	
Power Supply Voltage	4.93 V	
TM4C123 Output, V_{PE0} input to 7406	1.42 V	2.7 V
LED k- 7406 Output, V_{k-}	3.01 V	3.77 V
Bottom side of R19 LED a+, V_{a+}	4.45V	5.07V
LED Voltage	3V	3.76V
LED Current	2.7 mA	

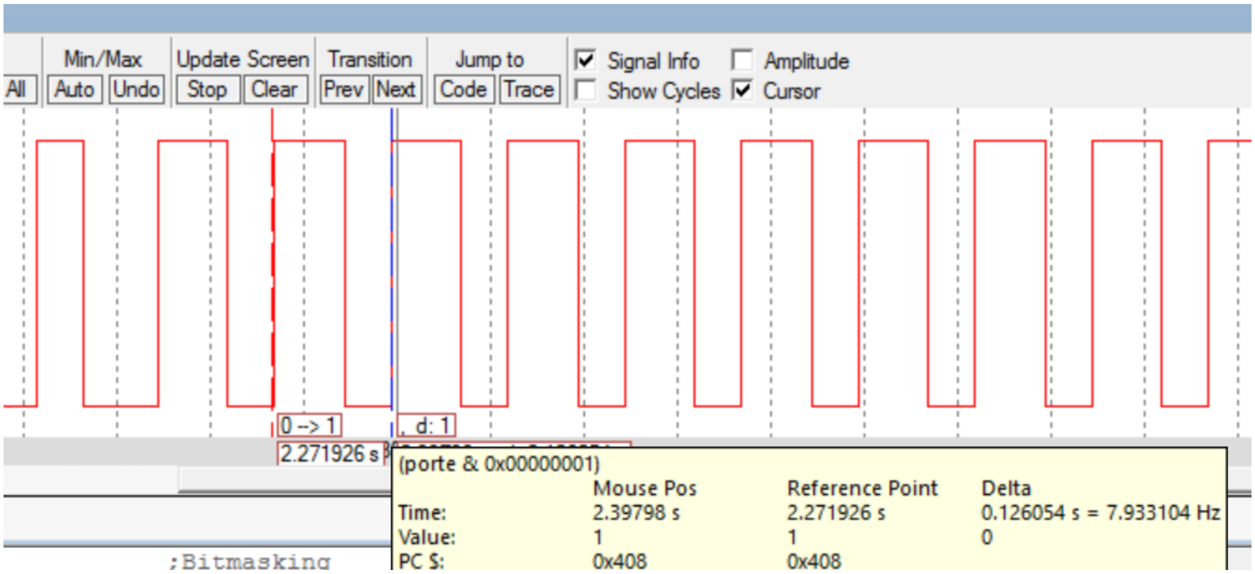


Min Time: 0 s
Max Time: 1.16634 s
Grid: 0.1 s

☑ Signal Info ☐ Amplitude
☐ Show Cycles ☑ Cursor

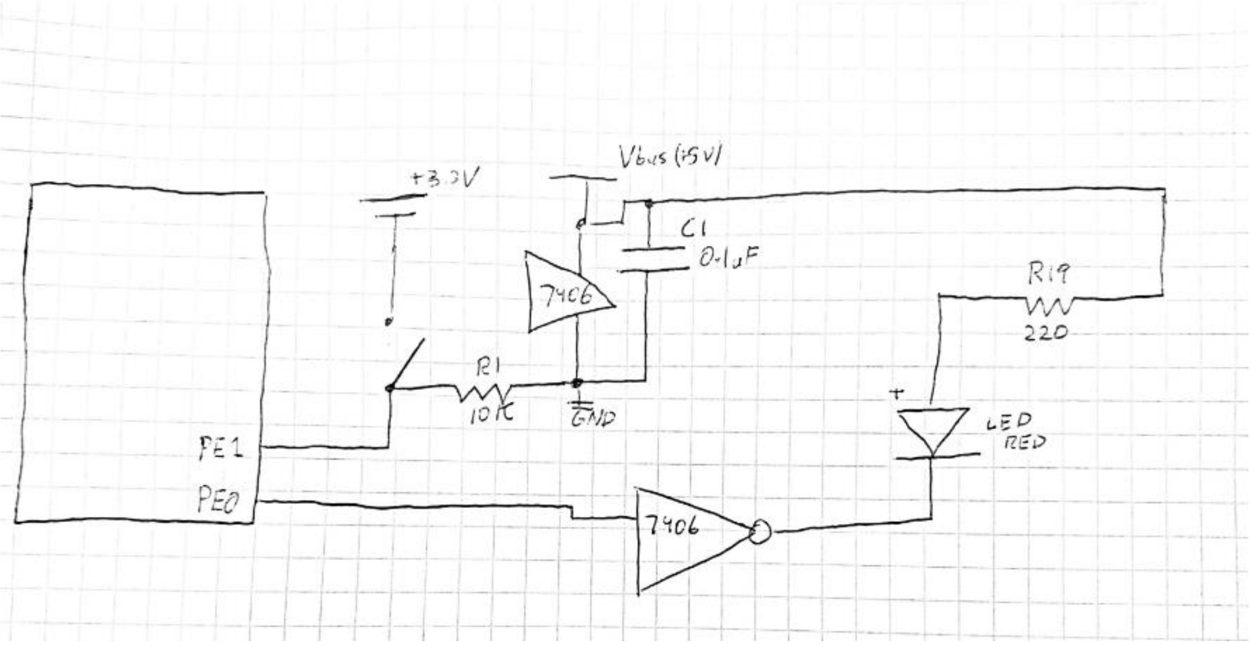
(porte & 0x00000001)

	Mouse Pos	Reference Point	Delta
Time:	0.97562 s	0.850016 s	0.125604 s = 7.961543 Hz
Value:	1	1	0
PC $:	0x408	0x408	



```
(porte & 0x00000001)
          Mouse Pos      Reference Point   Delta
Time:     2.39798 s      2.271926 s        0.126054 s = 7.933104 Hz
Value:    1              1                 0
PC $:     0x408          0x408
```

```
;Bitmasking
```



```
;****************** main.s ***************
; Program written by: *Sebastian Guillen Vargas, Michael Niemer*
; Date Created: 2/4/2017
; Last Modified: 2/4/2017
; 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 La            unchpad (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
;Constants for 8Hz frequency. Format:Percentage On Off
TwentyOn                    EQU 0x001E8480
TwentyOff                   EQU 0x0007A120
FortyOn                             EQU 0x0016E360
FortyOff        EQU 0x000F4240
SixtyOn                     EQU 0x000F4240
SixtyOff        EQU 0x0016E360
EightyOn         EQU 0x0007A120
EightyOff                   EQU 0x001E8480
;Constants for Breathingfrequency. Same format
Brth                        EQU 0x0000C350
Uno                                 EQU 0x00000032
PercentCnt                  EQU 0x000003E7
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
    CPSIE  I    ; TExaS voltmeter, scope runs on interrupts

;Initializing Port F;
            LDR R1, =SYSCTL_RCGCGPIO_R      ; 1) activate clock for Port F
            LDR R0, [R1]
   ORR R0, R0, #0x20             ; set bit 5 to turn on clock
   STR R0, [R1]
   NOP
   NOP                           ; allow time for clock to finish
   LDR R1, =GPIO_PORTF_DIR_R       ; 5) set direction register
   MOV R0,#0x00                  ; PF4 is input
   STR R0, [R1]
   LDR R1, =GPIO_PORTF_AFSEL_R     ; 6) regular port function
   MOV R0, #0                    ; 0 means disable alternate function
   STR R0, [R1]
   LDR R1, =GPIO_PORTF_PUR_R       ; pull-up resistors for PF4,PF0
   MOV R0, #0x10                 ; enable weak pull-up on PF0 and PF4
   STR R0, [R1]
   LDR R1, =GPIO_PORTF_DEN_R       ; 7) enable Port F digital port
   MOV R0, #0xFF                 ; 1 means enable digital I/O
   STR R0, [R1]

;Initializing Port E
   LDR R1, =SYSCTL_RCGCGPIO_R      ; 1) activate clock for Port E
            LDR R0, [R1]
   ORR R0, R0, #0x10             ; set bit 5 to turn on clock
   STR R0, [R1]
   NOP
```

```
  NOP
       LDR R1, =GPIO_PORTE_DIR_R   ; 5) set direction register
  MOV R0,#0x01          ; PE0 is output
  STR R0, [R1]
  LDR R1, =GPIO_PORTE_AFSEL_R   ; 6) regular port function
  MOV R0, #0          ; 0 means disable alternate function
  STR R0, [R1]
  LDR R1, =GPIO_PORTE_DEN_R   ; 7) enable Port E digital port
  MOV R0, #0xFF         ; 1 means enable digital I/O
  STR R0, [R1]

loop


Default20
       LDR R1, =GPIO_PORTF_DATA_R
       LDR R0, [R1]
       CMP R0, #0                                               ;Checking to see if PF4 is
pressed
       BNE Begin20                                              ;If not, continue duty cycles
       BL Breathe                                               ;If so, LED breathes
       B  BeginLEDoff                                    ;When Breathing ends, LED goes Off


Begin20
       BIC R2, R2                                               ;R2 needs to be cleared since it
is an indicator of when the button was pressed
       LDR R1, =GPIO_PORTE_DATA_R
       LDR R0, [R1]                                      ;Loading input from Port E
       AND R0, R0, #0x02                                 ;Bitmasking
       CMP R0, #2                                               ; Checking to see if PE1 is
pressed
       BNE skip20
```

```
        BL  Poll                                        ; Poll to see if button is released, R2
returns 0x02 if pushed AND released


skip20

        LDR R0, =TwentyOn                               ;R0 will serve as the constant for the On
time period

        LDR R3, =TwentyOff                              ;R3 will serve as the constant for the Off
time period

        CMP R2, #2

        BEQ next40                                          ; If it is pressed, then it does to
40%

        BL dutyloop                                     ; If not than the 20% duty cycle is called

        B Default20                                         ; If PE1 hasn't been pressed at
all, then 20% continues


next40

        LDR R1, =GPIO_PORTF_DATA_R

        LDR R0, [R1]

        CMP R0, #0                                          ;Checking to see if PF4 is
pressed

        BNE Begin40                                         ;If not, continue duty cycles

        BL Breathe                                          ;If so, LED breathes

        B  BeginLEDoff                                  ;When Breathing ends, LED goes Off

Begin40

        AND R2, R2, #0

        LDR R1, =GPIO_PORTE_DATA_R

        LDR R0, [R1]                                    ;Loading input from Port E

        AND R0, R0, #0x02                               ;Bitmasking

        CMP R0, #2                                          ; Checking to see if PE1 is
pressed

        BNE skip40

        BL  Poll                                        ; Poll to see if button is released, R2
returns 0x02 if pushed AND released
```

```
skip40

        LDR R0, =FortyOn

        LDR R3, =FortyOff

        CMP R2, #2

        BEQ next60                                          ; If pressed, 60% is called

        BL dutyloop                                         ; 40% is called

        B next40                                            ; If PE1 hasn't been pressed at
all, then 40% continues


next60

        LDR R1, =GPIO_PORTF_DATA_R

        LDR R0, [R1]

        CMP R0, #0                                          ;Checking to see if PF4 is
pressed

        BNE Begin60                                         ;If not, continue duty cycles

        BL Breathe                                          ;If so, LED breathes

        B  BeginLEDoff                                 ;When Breathing ends, LED goes Off

Begin60

        AND R2, R2, #0

        LDR R1, =GPIO_PORTE_DATA_R

        LDR R0, [R1]                                   ;Loading input from Port E

        AND R0, R0, #0x02                              ;Bitmasking

        CMP R0, #2                                          ; Checking to see if PE1 is
pressed

        BNE skip60

        BL  Poll                                       ; Poll to see if button is released, R2
returns 0x02 if pushed AND released

skip60

        LDR R0, =SixtyOn

        LDR R3, =SixtyOff

        CMP R2, #2

        BEQ next80                                          ; If pressed, 80% is called
        BL dutyloop                                         ; 60% is called
        B next60                                            ; 60% continues


next80
        LDR R1, =GPIO_PORTF_DATA_R
        LDR R0, [R1]
        CMP R0, #0                                          ;Checking to see if PF4 is
pressed
        BNE Begin80                                         ;If not, continue duty cycles
        BL Breathe                                          ;If so, LED breathes
        B  BeginLEDoff                                ;When Breathing ends, LED goes Off
Begin80
        AND R2, R2, #0
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]                                  ;Loading input from Port E
        AND R0, R0, #0x02                             ;Bitmasking
        CMP R0, #2                                          ; Checking to see if PE1 is
pressed
        BNE skip80
        BL  Poll                                      ; Poll to see if button is released, R2
returns 0x02 if pushed AND released
skip80
        LDR R0, =EightyOn
        LDR R3, =EightyOff
        CMP R2, #2
        BEQ next100                                         ; If pressed, 100% is called
        BL dutyloop                                         ; 80% is called
        B next80                                            ; 80% continues


next100
        LDR R1, =GPIO_PORTF_DATA_R
```

```
	LDR R0, [R1]
	CMP R0, #0						;Checking to see if PF4 is pressed
	BNE Begin100						;If not, continue duty cycles
	BL Breathe						;If so, LED breathes
	B  BeginLEDoff					;When Breathing ends, LED goes Off
Begin100
	AND R2, R2, #0
	LDR R1, =GPIO_PORTE_DATA_R
	LDR R0, [R1]					;Loading input from Port E
	AND R0, R0, #0x02					;Bitmasking
	CMP R0, #2						; Checking to see if PE1 is pressed
	BNE skip100
	BL  Poll						; Poll to see if button is released, R2 returns 0x02 if pushed AND released
skip100
	CMP R2, #2
	BEQ LEDoff
	MOV R0, #0x01
	STR R0, [R1]
	B next100


LEDoff
	LDR R1, =GPIO_PORTF_DATA_R
	LDR R0, [R1]
	CMP R0, #0						;Checking to see if PF4 is pressed
	BNE BeginLEDoff						;If not, continue duty cycles
	BL Breathe						;If so, LED breathes
```

```
BeginLEDoff

        AND R2, R2, #0

        LDR R1, =GPIO_PORTE_DATA_R

        LDR R0, [R1]                                            ;Loading input from Port E

        AND R0, R0, #0x02                                       ;Bitmasking

        CMP R0, #2                                                      ; Checking to see if PE1 is pressed

        BNE skip0

        BL  Poll                                                ; Poll to see if button is released, R2 returns 0x02 if pushed AND released

skip0

        CMP R2, #2

        BEQ Default20

        MOV R0, #0x00

        STR R0, [R1]

        B LEDoff



        B    loop


dutyloop


LoopOn  SUBS R0, R0, #1                                         ; Decrementing Counter

          BNE LoopOn                                            ; Counter, only branches when at 0

                LDR R1, =GPIO_PORTE_DATA_R

                LDR R0, [R1]                                    ; Loading data from Port E into register

                ORR R0, #0x01                                   ; Toggling PE0

                STR R0, [R1]                                    ; Toggling PE0


LoopOff         SUBS R3, R3, #1                                 ; Decrementing Counter
```

```
            BNE     LoopOff                                   ; Counter, only
branches when at 0

            LDR R0, [R1]                        ; Loading data from Port E into register

            AND R0, #0x00                       ; Toggling PE0

            STR R0, [R1]                        ; Toggling PE0

            BX LR                                 ; Loop back up


Poll

            LDR R2, [R1]                              ;Loading input from Port E

            AND R2, R2, #0x02                         ;Bitmasking

            EOR R2, R0, R2                            ; Checking to see if PE1 is
released

            CMP R2, #2

            BNE Poll

            BX LR


Breathe                                                       ;The Breathe function
goes through the duty cycles without checking to see

      MOV R4, LR                                              ;if PE1 is pressed. R13 is
pushed because the Breathe subroutine calls the


Bloop                                                         ;dutyloop subroutine in
order to have a delay, however new variable,

      LDR R0, =Brth                                   ; Brth#On/Off accounts for a faster Hz
in order to have breathing effect.

      MOV R3, #0

      LDR R1, =Uno                                        ;R0 will be when the light is on,
R3 will be for when its off

      LDR R12, =PercentCnt                        ;R1 is one percent of the constant
"Brth". R12 is the counter ((1/Uno)-1)


Loopception
```

```
        LDR R1, =Uno                                    ;R1 must be reloaded for
security

        SUB R0, R0, R1                                  ;Certain Percentage of constant
"Brth" is deducted

        ADD R3, R3, R1                                  ;Certain Percantage of constant
"Brth" is added

  MOV R5, R3

        MOV R6, R0                                          ; Registers in the
dutyloop subroutine modify R3 and R0. Saving in R5, R6.

        LDR R7, =GPIO_PORTF_DATA_R              ;Checking to see if PF4 is not pressed anymore

        LDR R8, [R7]

        CMP R8, #0

        BNE Leave

        BL dutyloop

        MOV R0, R6                                          ; Reloading registers

        MOV R3, R5

        SUBS R12, R12, #1                               ; Once counter reaches zero,
program can skip branch

        BNE Loopception


        LDR R3, =Brth                                   ; Same routine as Loopception
EXCEPT the R3 and R0 is reversed

        MOV R0, #0                                          ; This gives the illusion
of the LED coming down from its brightest peak

        LDR R1, =Uno                                    ; and back to its off state.

        LDR R12, =PercentCnt

Loopception2

        LDR R1, =Uno

        SUB R3, R3, R1

        ADD R0, R0, R1

  MOV R5, R0

        MOV R6, R3

        LDR R7, =GPIO_PORTF_DATA_R              ;Checking to see if PF4 is not pressed anymore
        LDR R8, [R7]

        CMP R8, #0

        BNE Leave

        BL dutyloop

        MOV R3, R6

        MOV R0, R5

        SUBS R12, R12, #1

        BNE Loopception2

        B       Bloop                                   ;Bloop serves to not Push R13
every time

Leave

        MOV LR, R4

        BX LR




    ALIGN    ; make sure the end of this section is aligned

    END      ; end of file
```