Table 1

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	**9.84 kΩ**	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	**3.29 V**	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	**0V**	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	**0 mA**	mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	**3.29 V**	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	**.329 mA**	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

Table 2

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	**217Ω**	ohms	with power off and disconnected from circuit (measured with ohmmeter)
	+5 V power supply			(measured with voltmeter relative

2	V_{+5}	**4.99 V**	volts	to ground, *notice that the +5V power is not exactly +5 volts*)
3	TM4C123 Output, V_{PE0} input to 7406	**.674 V**	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
4	7406 Output, V_{k-} LED k-	**2.97 V**	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
5	LED a+, V_{a+} Bottom side of R19	**4.4 V**	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
6	LED voltage	**1.43 V**	volts	calculated as V_{a+} - V_{k-}
7	LED current	**6.5 mA**	mA	calculated as (V_{+5} – V_{a+})/R19 and measured with an ammeter
8	TM4C123 Output, V_{PE0} input to 7406	**94.9 mV**	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
9	7406 Output, V_{k-} LED k-	**3.65 V**	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
10	LED a+, V_{a+} Bottom side of R19	**4.96 V**	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
11	LED voltage	**1.31 V**	volts	calculated as V_{a+} - V_{k-}
12	LED current	**5.95 mA**	mA	calculated as (V_{+5} - V_{a+})/R19 and

		5.45 mA		measured with an ammeter

```
;****************** main.s ***************
; Program written by: Dylan Cauwels, Andrew Han
; Date Created: 2/4/2017
; Last Modified: 2/14/2017
; 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
GPIO_PORTF_LOCK_R  EQU 0x40025520
GPIO_PORTF_CR_R    EQU 0x40025524
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R  EQU 0x4002552C
GPIO_LOCK_KEY      EQU 0x4C4F434B  ; Unlocks the GPIO_CR register
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
      LDR R1, =SYSCTL_RCGCGPIO_R      ;Activate Port F/E Clock
    LDR R0, [R1]
    ORR R0, R0, #0x30
    STR R0, [R1]                   ;Clock Initialize Time
    NOP
    NOP
    LDR R1, =GPIO_PORTF_LOCK_R      ;Unlock PortF Register
    LDR R0, =0x4C4F434B             ;MAYBE YES OR NO NOT SURE WHY
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_CR_R
    MOV R0, #0xFF
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AMSEL_R     ;Disable Analog
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PCTL_R      ;Start GPIO
    MOV R0, #0x00
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DIR_R       ;Set Direction Register
    MOV R0, #0x00                    ;All Input
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AFSEL_R     ;Initialize PortF
    MOV R0, #0
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_AFSEL_R     ;Initialize PortE
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PUR_R       ;Pull Up Resistors
    MOV R0, #0x10
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DEN_R       ;Enable PortF
    MOV R0, #0xFF
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_DEN_R       ;Enable PortE
    MOV R0, #0xFF
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_DIR_R       ;Set Direction Register
    MOV R0,#0x01                   ;0 Output, 1 Input
    STR R0, [R1]



      MOV R3, #1                              ; R3 CONTAINS COUNT FOR DUTY
MULTIPLIER
      MOV R4, #4                              ; R4 CONTAINS COUNT FOR
LEDOFF DUTY MULTIPLIER
      MOV R5, #25                             ; PUTS MULITPLE OF 25 IN R5
mainloop
      BL buttoncheck                          ;CHECKS PORTE1(BUTTON)
checkpoint
```

```
      BL LEDON                                  ;TOGGLE LED ON
      BL LEDOFF                                 ;TOGGLE LED OFF
      BL mainloop                               ;REPEAT
LEDON
      CMP R3, #0
      BEQ next
      LDR R1, =GPIO_PORTE_DATA_R   ;Load PortE Data Address
      LDR R0, [R1]                              ;Loading PortE Data
      MOV R0, #0x01                             ;Toggling LED PE0
      STR R0, [R1]                              ;Storing Result back in
PortE
      MUL R6, R3, R5                            ;NUMBER OF MILISECS, R5 HAS
25, R3 HAS NUMBER OF TIMES TO MULTIPLY BY, PUT IN R6
LOOP
      MOV R2, #19990                            ; COUNT
delay
      SUBS R2, R2, #1                           ; R2 CONTAINS A NUMBER TO
GET UP TO 1MS (16000)
      BNE delay
      SUBS R6, R6, #1                           ; R6 CONTAINS NUMBER OF MS
COUNT
      BNE LOOP
next
      BX LR
LEDOFF
      CMP R4, #0
      BEQ next1
      LDR R1, =GPIO_PORTE_DATA_R        ; LOAD PORTE DATA ADDRESS
      LDR R0, [R1]
      MOV R0, #0x00                             ; OFF
      STR R0, [R1]
      MUL R7, R4, R5                            ;# OF MILISECS, R5 HAS 25
AND, R4 HAS NUMBER OF TIMES TO MULTIPLY BY, PUT IN R7
LOOP1
      MOV R2, #20000                            ; COUNT
delay1
      SUBS R2, R2, #1                           ; R2 CONTAINS A NUMBER TO
GET UP TO 1MS (16000)
      BNE delay1
      SUBS R7, R7, #1                           ; R4 CONTAINS NUMBER OF OFF
MS COUNT
      BNE LOOP1
next1
      BX LR
buttoncheck
      LDR R1, =GPIO_PORTE_DATA_R
      LDR R0, [R1]
      LSR R0, #1
      EOR R0, #0x01
      CMP R0, #0
      BEQ change
      LDR R1, =GPIO_PORTF_DATA_R
      LDR R0, [R1]
      AND R0, #0x10
```

```
      CMP R0, #0
      BEQ breathingMode
      BX LR
change                                                ;WHEN BUTTON IS
PRESSED INCREASES R3 COUNT AND DECREASES R4
      LDR R1, =GPIO_PORTE_DATA_R
      LDR R0, [R1]
      LSR R0, #1
      EOR R0, #0x01
      CMP R0, #0
      BEQ change
      ADD R3, #1
      CMP R3, #6                                      ;SEES IF BUTTON HAS BEEN
PRESSED 5 TIMES (6TH TIME SHOULD SET TO
      BNE change2
      AND R3, #0
change2
      SUBS R4, R4 , #1
      CMP  R4, #0
      BPL back                                        ;IF POSITIVE OR ZERO JUMP,
IF NEGATIVE WANT TO ADD 5 TO GET TO 4(RESET)
      ADD R4, #6
back
      BX LR

breathingMode
      LDR R1, =GPIO_PORTE_DATA_R                      ;turning LED off
      LDR R0, [R1]
      AND R0, #0x0
      STR R0, [R1]
      MOV R8, #1                                            ;R8 = on time
      MOV R9, #40                                           ;R9 = off time
      AND R7, #0x0
loop1
      BL buttonCheck2
      MOV R7, R8
      BL toggle
      BL LOOP1
      MOV R7, R9
      BL toggle
      BL LOOP1
      SUBS R9, #1
      ADD R8, #1
      CMP R9, #0
      BNE loop1
      ADD R9, #1
loop2
      BL buttonCheck2
      MOV R7, R8
      BL toggle
      BL LOOP1
      MOV R7, R9
      BL toggle
      BL LOOP1
```

```
      SUBS R8, #1
      ADD R9, #1
      CMP R8, #0
      BNE loop2
      B breathingMode

toggle
      LDR R1, =GPIO_PORTE_DATA_R
      LDR R0, [R1]
      EOR R0, #0x01
      STR R0, [R1]
      BX LR

buttonCheck2
      LDR R1, =GPIO_PORTF_DATA_R
      LDR R0, [R1]
      AND R0, #0x10
      CMP R0, #0
      BNE checkpoint
      BX LR



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