





Parameter	Value	Units	Conditions
Resistance of the 10k Ω resistor, R1	9,840	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, V _{+3,3}	3.28	volts	Powered (measured with voltmeter)
Input Voltage, V _{PE2}	3.23	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	0	mA	Powered, but switch not pressed $I = V_{PE2} / R1$ (calculated and measured with an ammeter)
Input Voltage, V _{PE2}	0.1	millivolts	Powered and with switch pressed (measured with voltmeter)
Resistor current	450	mA	Powered and switch pressed $I = V_{PE2} / R1$ (calculated and measured with an ammeter)

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	216	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	4.99	volts	(measured with voltmeter relative to ground, <i>notice that the +5V power is not exactly +5 volts</i>)
3	TM4C123 Output, V_{PE3} input to ULN2003B	1.1	volts	with PE3 = 0 (measured with voltmeter relative to ground). We call this V_{OL} of the TM4C123.
4	ULN2003B Output, pin 16, V_k LED k-	0	volts	with PE3 = 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)	0.001	volts	with PE3 = 0 (measured with voltmeter relative to ground). This measurement is also weird, because it too is floating.
6	LED voltage	0.001	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_{PE3} input to ULN2003B	3.22	volts	with PE3 = 1 (measured with voltmeter relative to ground). We call this V_{OH} of the TM4C123.
9	ULN2003B Output pin 16, V_k LED k-	0.7	volts	with PE3 = 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)	2.8	volts	with PE3 = 1 (measured with voltmeter relative to ground)
11	LED voltage	2.1	volts	calculated as $V_{a+} - V_k$.
12	LED current (on)	20	mA	calculated as $(V_{+5} - V_{a+})/R19$ and measured with an ammeter

```

1  ;***** main.s *****
2  ; Program written by: Trevor Barrett, Raed
3  ; Date Created: 2/4/2017
4  ; Last Modified: 1/18/2019
5  ; Brief description of the program
6  ;   The LED toggles at 2 Hz and a varying duty-cycle
7  ; Hardware connections (External: One button and one LED)
8  ;   PE2 is Button input  (1 means pressed, 0 means not pressed)
9  ;   PE3 is LED output (1 activates external LED on protoboard)
10 ;   PF4 is builtin button SW1 on Launchpad (Internal)
11 ;   Negative Logic (0 means pressed, 1 means not pressed)
12 ; Overall functionality of this system is to operate like this
13 ;   1) Make PE3 an output and make PE2 and PF4 inputs.
14 ;   2) The system starts with the the LED toggling at 2Hz,
15 ;       which is 2 times per second with a duty-cycle of 30%.
16 ;       Therefore, the LED is ON for 150ms and off for 350 ms.
17 ;   3) When the button (PE1) is pressed-and-released increase
18 ;       the duty cycle by 20% (modulo 100%). Therefore for each
19 ;       press-and-release the duty cycle changes from 30% to 70% to 70%
20 ;       to 90% to 10% to 30% so on
21 ;   4) Implement a "breathing LED" when SW1 (PF4) on the Launchpad is pressed:
22 ;       a) Be creative and play around with what "breathing" means.
23 ;           An example of "breathing" is most computers power LED in sleep mode
24 ;           (e.g., https://www.youtube.com/watch?v=ZT6siXyIjvQ).
25 ;       b) When (PF4) is released while in breathing mode, resume blinking at 2Hz.
26 ;           The duty cycle can either match the most recent duty-
27 ;           cycle or reset to 30%.
28 ;       TIP: debugging the breathing LED algorithm using the real board.
29 ; PortE device registers
30 GPIO_PORTE_DATA_R EQU 0x400243FC
31 GPIO_PORTE_DIR_R EQU 0x40024400
32 GPIO_PORTE_AFSEL_R EQU 0x40024420
33 GPIO_PORTE_DEN_R EQU 0x4002451C
34 ; PortF device registers
35 GPIO_PORTF_DATA_R EQU 0x400253FC
36 GPIO_PORTF_DIR_R EQU 0x40025400
37 GPIO_PORTF_AFSEL_R EQU 0x40025420
38 GPIO_PORTF_PUR_R EQU 0x40025510
39 GPIO_PORTF_DEN_R EQU 0x4002551C
40 GPIO_PORTF_LOCK_R EQU 0x40025520
41 GPIO_PORTF_CR_R EQU 0x40025524
42 GPIO_LOCK_KEY EQU 0x4C4F434B ; Unlocks the GPIO_CR register
43 SYSTCTL_RCGCGPIO_R EQU 0x400FE608
44 count EQU 0xA9054
45 countinc EQU 0x1520A8
46 count10 EQU 0xA90 ; 10% count
47 count20 EQU 0x1520 ; 20% count
48 count30 EQU 0x1FB0 ; 30% count
49 count40 EQU 0x2A40 ; 40% count
50 count50 EQU 0x34D1 ; 50% count
51 count60 EQU 0x3F61 ; 60% count
52 count70 EQU 0x49F2 ; 70% count
53 count80 EQU 0x5481 ; 80% count
54 count90 EQU 0x5F12 ; 90% count
55 countShort EQU 0x30D40 ;200000, 10% of 2,000,000
56 countLong EQU 0x1B7740 ;180000, 90% of 2,000,000
57 countAdd EQU 0x61A80 ;400000, 20% of 2,000,000
58 countMinus EQU 0xFFFF9E580 ;-400000, -20% of 2,000,000
59 countNeg EQU 0xFFE488C0 ;-180000
60
61 IMPORT TExaS_Init
62 THUMB
63 AREA DATA, ALIGN=2
64 ;global variables go here
65
66
67 AREA |.text|, CODE, READONLY, ALIGN=2
68 THUMB
69 EXPORT Start
70 Start
71 ; TExaS_Init sets bus clock at 80 MHz
72 BL TExaS_Init ; voltmeter, scope on PD3

```

```

73      ; Initialization goes here
74      LDR R6, =countShort           ;making R2 the short delay length number
75      LDR R7, =countLong
76      LDR R1, =SYCTL_RCGCGPIO_R    ;start clock for port E
77      LDR R0, [R1]
78      ORR R0, R0, #0x30
79      STR R0, [R1]
80      NOP
81      NOP
82      LDR R1, =GPIO_PORTE_DIR_R     ;set PE3 as an output, PE2 as an input
83      LDR R0, [R1]
84      ORR R0, R0, #0x08
85      STR R0, [R1]
86      LDR R1, =GPIO_PORTE_DEN_R     ;enable digital I/O for PE2, PE3
87      LDR R0, [R1]
88      ORR R0, R0, #0x0F
89      STR R0, [R1]
90      LDR R1, =GPIO_PORTF_DIR_R     ;set PF4 as input
91      MOV R0, #0x00
92      STR R0, [R1]
93      LDR R1, =GPIO_PORTF_DEN_R     ;enable digital I/O for PF4
94      LDR R0, [R1]
95      ORR R0, #0x10
96      STR R0, [R1]
97      LDR R0, =GPIO_LOCK_KEY
98      LDR R1, =GPIO_PORTF_LOCK_R
99      STR R0, [R1]
100     LDR R0, =GPIO_PORTF_CR_R
101     LDR R1, [R0]
102     ORR R1, #0xFF
103     STR R1, [R0]
104     LDR R0, =GPIO_PORTF_PUR_R
105     LDR R1, [R0]
106     ORR R1, #0x10
107     STR R1, [R0]
108
109
110     B loop
111
112     SWITCH
113
114         LDR R1, =GPIO_PORTE_DATA_R ;continually checks if button is pressed
115         LDR R0, [R1]               ;once button is released, count is incremented
116         AND R0, #0x04
117         ROR R0, #2
118         CMP R0, #1
119         BEQ SWITCH
120         LDR R0, =countNeg          ;checks to see if the short delay number has become
121         ADD R0, R6, R0
122         CMP R0, #0
123         BEQ reset                  ;90%, which means that the duty cycle must reset to short delay
being 10%
124         LDR R4, =countAdd          ;adding 20% to the shorter delay, R2
125         LDR R5, =countMinus        ;subtracting 20% from the long delay R3
126         ADD R6, R6, R4
127         ADD R7, R7, R5
128         B cont
129     reset LDR R6, =countShort
130         LDR R7, =countLong
131     cont  BX LR
132
133     LongDelay
134         ADD R5, R7, #0
135     Loop1
136         LDR R1, =GPIO_PORTF_DATA_R ;continually checks if button is pressed
137         LDR R0, [R1]               ;once button is released, count is incremented
138         AND R0, #0x10
139         ROR R0, #4
140         CMP R0, #0
141         BEQ Breathe
142         LDR R1, =GPIO_PORTE_DATA_R ;checks if buttons is pressed,
143         LDR R0, [R1]               ;branches to SWITCH if button is pressed

```

```
144         AND R0, #0x04
145         ROR R0, #2
146         CMP R0, #1
147         BEQ SWITCH
148         SUBS R5,R5,#0x01
149         BNE Loop1
150         BX LR
151
152     ShortDelay
153         ADD R4, R6, #0
154
155     Loop2
156         LDR R1, =GPIO_PORTF_DATA_R ;continually checks if button is pressed
157         LDR R0, [R1] ;once button is released, count is incremented
158         AND R0, #0x10
159         ROR R0, #4
160         CMP R0, #0
161         BEQ Breathe
162         LDR R1, =GPIO_PORTE_DATA_R ;checks if buttons is pressed,
163         LDR R0, [R1] ;branches to SWITCH if button is pressed
164         AND R0, #0x04
165         ROR R0, #2
166         CMP R0, #1
167         BEQ SWITCH
168         SUBS R4,R4,#0x01
169         BNE Loop2
170         BX LR
171
172     Breathe
173     ; This loop cycles through duty cycles at a rate
174     ; such that the LED appears to 'breathe'
175     ; in testing the breathing function was most smooth
176     ; when LED was started as ON which is why the loop
177     ; first toggles on the LED
178     ; In this loop R5 serves as a loop counter so that the
179     ; LED stays at a certain duty cycle for a longer time
180     ; making the breathe more noticable
181     ; R3 holds the count that keeps the light ON
182     ; R4 holds the count that keeps the light OFF
183         LDR R1, =GPIO_PORTE_DATA_R ;toggle LED on
184         LDR R0, [R1]
185         ORR R0, R0, #0x08
186         STR R0, [R1]
187         ;10% duty cycle
188         MOV R5, #10
189         LDR R3, =count10
190         LDR R4, =count90
191         BL breath
192         ;20% duty cycle
193         MOV R5, #10
194         LDR R3, =count20
195         LDR R4, =count80
196         BL breath
197         ;30% duty cycle
198         MOV R5, #10
199         LDR R3, =count30
200         LDR R4, =count70
201         BL breath
202         ;40% duty cycle
203         MOV R5, #10
204         LDR R3, =count40
205         LDR R4, =count60
206         BL breath
207         ;50% duty cycle
208         MOV R5, #10
209         LDR R3, =count50
210         LDR R4, =count50
211         BL breath
212         ;60% duty cycle
213         MOV R5, #10
214         LDR R3, =count60
215         LDR R4, =count40
```

```

216         BL breath
217         ;70% duty cycle
218         MOV R5, #10
219         LDR R3, =count70
220         LDR R4, =count30
221         BL breath
222         ;80% duty cycle
223         MOV R5, #10
224         LDR R3, =count80
225         LDR R4, =count20
226         BL breath
227         ;90% duty cycle
228         MOV R5, #20
229         LDR R3, =count90
230         LDR R4, =count10
231         BL breath
232         ;80% duty cycle
233         MOV R5, #10
234         LDR R3, =count80
235         BL breath
236         ;70% duty cycle
237         MOV R5, #10
238         LDR R3, =count70
239         BL breath
240         ;60% duty cycle
241         MOV R5, #10
242         LDR R3, =count60
243         BL breath
244         ;50% duty cycle
245         MOV R5, #10
246         LDR R3, =count50
247         BL breath
248         ;40% duty cycle
249         MOV R5, #10
250         LDR R3, =count40
251         BL breath
252         ;30% duty cycle
253         MOV R5, #10
254         LDR R3, =count30
255         BL breath
256         ;20% duty cycle
257         MOV R5, #10
258         LDR R3, =count20
259         BL breath
260         ;10% duty cycle
261         MOV R5, #10
262         LDR R3, =count10
263         BL breath
264         B Breathe
265
266 breath   LDR R1, =GPIO_PORTA_DATA_R      ;toggle LED on
267          LDR R0, [R1]
268          EOR R0, R0, #0x08
269          STR R0, [R1]
270          ADD R2, R3, #0
271 again    LDR R1, =GPIO_PORTF_DATA_R      ;checks if buttons is pressed,
272          LDR R0, [R1]                    ;branches to loop if button is realeased thus resuming regular
273          LDR R0, [R1]
274          AND R0, R0, #0x10
275          ROR R0, #4
276          CMP R0, #0
277          BNE loop
278          ;delay to keep LED on
279          SUBS R2, R2, #0x01
280          CMP R2, #0
281          BNE again
282          LDR R1, =GPIO_PORTA_DATA_R      ;toggle LED off
283          LDR R0, [R1]
284          EOR R0, R0, #0x08
285          STR R0, [R1]
286          ADD R2, R4, #0

```



```
287  again2  LDR R1, =GPIO_PORTF_DATA_R  ;checks if buttons is pressed,
288          LDR R0, [R1]                ;branches to SWITCH if button is pressed
289          AND R0, #0x10
290          ROR R0, #4
291          CMP R0, #0
292          BNE loop
293          ;delay to keep LED off
294          SUBS R2,R2,#0x01
295          CMP R2, #0
296          BNE again2
297          ;overall loop counter
298          SUBS R5, R5, #1
299          CMP R5, #0
300          BNE breath
301
302          BX LR
303
304
305          CPSIE I      ; TExaS voltmeter, scope runs on interrupts
306  loop
307
308          LDR R1, =GPIO_PORTE_DATA_R    ;toggle LED on
309          LDR R0, [R1]
310          EOR R0, R0, #0x08
311          STR R0, [R1]
312
313          BL ShortDelay                  ;delay keeps LED on
314
315          LDR R1, =GPIO_PORTE_DATA_R    ;toggle LED (off)
316          LDR R0, [R1]
317          EOR R0, R0, #0x08
318          STR R0, [R1]
319
320          BL LongDelay                   ;delay keeps LED off
321
322          B      loop
323
324
325          ALIGN      ; make sure the end of this section is aligned
326          END        ; end of file
327
328
329
```