```
;****************** main.s ***************
; Program written by: Trevor Barrett, Raed
; Date Created: 2/4/2017
; Last Modified: 1/18/2019
; Brief description of the program
;   The LED toggles at 2 Hz and a varying duty-cycle
; Hardware connections (External: One button and one LED)
;  PE2 is Button input  (1 means pressed, 0 means not pressed)
;  PE3 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 PE3 an output and make PE2 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
count              EQU 0xA9054
countinc           EQU 0x1520A8
count10            EQU 0xA90
count20            EQU 0x1520
count30            EQU 0x1FB0;2
count40            EQU 0x2A40
count50            EQU 0x34D1
count60            EQU 0x3F61
count70            EQU 0x49F2
count80            EQU 0x5481
count90            EQU 0x5F12
countShort         EQU 0x30D40   ;200000,   10% of 2,000,000
countLong          EQU 0x1B7740    ;180000, 90% of 2,000,000
countAdd           EQU 0x61A80    ;400000, 20% of 2,000,000
countMinus         EQU 0xFFF9E580  ;-400000, -20% of 2,000,000
countNeg           EQU 0xFFE488C0  ;-180000

       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
```

```
 73    ; Initialization goes here
 74       LDR R6, =countShort             ;making R2 the short delay length number
 75       LDR R7, =countLong
 76       LDR R1, =SYSCTL_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  Loop2
155          LDR R1, =GPIO_PORTF_DATA_R  ;continually checks if button is pressed
156          LDR R0, [R1]                ;once button is released, count is incremented
157          AND R0, #0x10
158          ROR R0, #4
159          CMP R0, #0
160          BEQ Breathe
161          LDR R1, =GPIO_PORTE_DATA_R  ;checks if buttons is pressed,
162          LDR R0, [R1]                ;branches to SWITCH if button is pressed
163          AND R0, #0x04
164          ROR R0, #2
165          CMP R0, #1
166          BEQ SWITCH
167          SUBS R4,R4,#0x01
168          BNE Loop2
169          BX LR
170
171
172  Breathe
173          LDR R1, =GPIO_PORTE_DATA_R      ;toggle LED on
174          LDR R0, [R1]
175          ORR R0, R0, #0x08
176          STR R0, [R1]
177          MOV R5, #10
178          LDR R3, =count10
179          LDR R4, =count90
180          BL ten
181  SE1     MOV R5, #10
182          LDR R3, =count20
183          LDR R4, =count80
184          BL ten
185  SE2     MOV R5, #10
186          LDR R3, =count30
187          LDR R4, =count70
188          BL ten
189  SE3     MOV R5, #10
190          LDR R3, =count40
191          LDR R4, =count60
192          BL ten
193  SE4     MOV R5, #10
194          LDR R3, =count50
195          LDR R4, =count50
196          BL ten
197  SE5     MOV R5, #10
198          LDR R3, =count60
199          LDR R4, =count40
200          BL ten
201  SE6     MOV R5, #10
202          LDR R3, =count70
203          LDR R4, =count30
204          BL ten
205  SE7     MOV R5, #10
206          LDR R3, =count80
207          LDR R4, =count20
208          BL ten
209  SE8     MOV R5, #6
210          LDR R3, =count90
211          LDR R4, =count10
212          BL ten
213  SE9     MOV R5, #10
214          LDR R3, =count80
215          BL ten
```

```
216   SE10    MOV R5, #10
217           LDR R3, =count70
218           BL ten
219   SE11    MOV R5, #10
220           LDR R3, =count60
221           BL ten
222   SE12    MOV R5, #10
223           LDR R3, =count50
224           BL ten
225   SE13    MOV R5, #10
226           LDR R3, =count40
227           BL ten
228   SE15    MOV R5, #10
229           LDR R3, =count30
230           BL ten
231   SE16    MOV R5, #10
232           LDR R3, =count20
233           BL ten
234   SE17    MOV R5, #10
235           LDR R3, =count10
236           BL ten
237           B Breathe
238
239           ;ON
240   ten     LDR R1, =GPIO_PORTE_DATA_R      ;toggle LED on
241           LDR R0, [R1]
242           EOR R0, R0, #0x08
243           STR R0, [R1]
244           ADD R2, R3, #0
245   Re1     LDR R1, =GPIO_PORTF_DATA_R  ;checks if buttons is pressed,
246           LDR R0, [R1]                ;branches to SWITCH if button is pressed
247           AND R0, #0x10
248           ROR R0, #4
249           CMP R0, #0
250           BNE loop
251           SUBS R2,R2,#0x01
252           CMP R2, #0
253           BNE Re1
254           ;OFF
255           LDR R1, =GPIO_PORTE_DATA_R      ;toggle LED off
256           LDR R0, [R1]
257           EOR R0, R0, #0x08
258           STR R0, [R1]
259           ADD R2, R4, #0
260   Re2     LDR R1, =GPIO_PORTF_DATA_R  ;checks if buttons is pressed,
261           LDR R0, [R1]                ;branches to SWITCH if button is pressed
262           AND R0, #0x10
263           ROR R0, #4
264           CMP R0, #0
265           BNE loop
266           SUBS R2,R2,#0x01
267           CMP R2, #0
268           BNE Re2
269           SUBS R5, R5, #1
270           CMP R5, #0
271           BNE ten
272           ;B SE1
273           BX LR
274
275
276        CPSIE  I    ; TExaS voltmeter, scope runs on interrupts
277   loop
278   ; main engine goes here
279
280       LDR R1, =GPIO_PORTE_DATA_R      ;toggle LED on
281       LDR R0, [R1]
282       EOR R0, R0, #0x08
283       STR R0, [R1]
284
285       BL ShortDelay                       ;delay keeps LED on
286
287       LDR R1, =GPIO_PORTE_DATA_R      ;toggle LED (off)
```

```
288       LDR R0, [R1]
289       EOR R0, R0, #0x08
290       STR R0, [R1]
291
292       BL LongDelay                        ;delay keeps LED off
293
294        B    loop
295
296
297        ALIGN      ; make sure the end of this section is aligned
298        END        ; end of file
299
300
```