```
1  ;-------------------------------------------------------------------------------
2  ; MSP430 Assembler Code Template for use with TI Code Composer Studio
3  ;
4  ;
5  ;-------------------------------------------------------------------------------
6              .cdecls C,LIST,"msp430.h"       ; Include device header file
7
8  ;-------------------------------------------------------------------------------
9              .def    RESET                   ; Export program entry-point to
10                                             ; make it known to linker.
11 ;-------------------------------------------------------------------------------
12             .data                           ; Assemble into program memory.
13             .retain                         ; Override ELF conditional linking
14                                             ; and retain current section.
15             .retainrefs                     ; And retain any sections that have
16                                             ; that have references to current
17                                             ; section
18
19 count:      .space 2                        ; count of timer interupts since the program
   started
20 count0:     .space 2                        ; Value of count variable at the time counter is
   started
21 count1:     .space 2                        ; Value of count variable at the time counter is
   stopped
22 result:     .space 2                        ; result = count1 - count0
23
24 cstatus:    .space 1                        ; Zeroth bit of cstaus stores your counter's
   status bit. 0 means counter stopped, 1 means counter running
25
26 ;-------------------------------------------------------------------------------
27             .text                           ; Assemble into program memory.
28             .retain                         ; Override ELF conditional linking
29                                             ; and retain current section.
30             .retainrefs                     ; And retain any sections that have
31                                             ; references to current section.
32
33 ;-------------------------------------------------------------------------------
34 RESET       mov.w   #__STACK_END,SP         ; Initialize stackpointer
35 StopWDT     mov.w   #WDTPW|WDTHOLD,&WDTCTL  ; Stop watchdog timer
36
37
38 ;-------------------------------------------------------------------------------
39 ; Main loop here
40
41     ; P1.0 is connected to the red LED
42     bic.b   #BIT0, &P1OUT  ; start with LED off
43     bis.b   #BIT0, &P1DIR  ; Set P1.0 as output pin
44
45     ; Configure Push Button S1 (Port 1, Pin 1)
46     bis.b   #BIT1, &P1REN           ; Enable Resistor
47     bis.b   #BIT1, &P1OUT           ; Pull-up resistor, since S1 is active low
48     bis.b   #BIT1,  &P1IE           ; Enable P1.1 interrupts
49     bis.b   #BIT1,  &P1IES          ; Enable Loweing Edge
50     bic.b   #BIT1,  &P1IFG
51
52     ; Configer timer B0, connected to ACLK, to raise interrupts
53     bis.w #TBCLR, &TB0CTL
```

```
54      bis.w #TBSSEL__ACLK, &TB0CTL
55      bis.w #MC__CONTINUOUS, &TB0CTL
56      bis.w #TBIE, &TB0CTL
57      bic.w #TBIFG, &TB0CTL
58
59  UnlockGPIO:  bic.w   #LOCKLPM5,&PM5CTL0      ; Disable the GPIO power-on default
60
61      nop                             ; no operation
62      bis.w   #GIE|LPM3,   SR         ; sleep in LPM3 mode and wait for interrupts
63      nop                             ; no operation
64
65  ;-------------------------------------------------------------------------
66  ;                           ISRs
67  ;-------------------------------------------------------------------------
68
69  PORT1_ISR:
70
71          xor.b     #BIT0, &P1OUT     ; Toggle P1.0
72          xor.b     #BIT0, &cstatus   ; Toggle counter status flag
73
74          bit.b     #BIT0,    &cstatus    ; if cstatus' 0th bit is 1 (counter runniing)
75          jnc IF1_EXIT
76                    mov.w   &count,  &count0
77  IF1_EXIT:
78
79          bit.b     #BIT0,    &cstatus    ; if cstatus' 0th bit is 0 (counter stopping)
80          jc  IF2_EXIT
81                    mov.w &count,  &count1
82                    mov.w &count1, &result
83                    sub.w &count0, &result  ;result = count1 - count0
84
85  IF2_EXIT:
86
87          bic.b     #BIT1, &P1IFG         ; Multisourced Interrupt so clear interrupt flag
88      reti
89
90  ;-------------------------------------------------------------------------
91
92  Timer_Overflow:
93          inc.w   &count              ; increase count by 1
94          bic.w   #TBIFG, &TB0CTL     ; Multisourced Interrupt so clear interrupt flag
95      reti
96
97  ;-------------------------------------------------------------------------
98  ; Stack Pointer definition
99  ;-------------------------------------------------------------------------
100             .global __STACK_END
101             .sect   .stack
102
103 ;-------------------------------------------------------------------------
104 ; Interrupt Vectors
105 ;-------------------------------------------------------------------------
106             .sect   ".reset"                ; MSP430 RESET Vector
107             .short  RESET
108
109             .sect   ".int37"                ; PORT1 Vector
110             .short  PORT1_ISR
```

```
111
112             .sect   ".int50"                ; TIMER Vector
113             .short  Timer_Overflow
114
115
```