```
;-------------------------------------------------------------------------------
; MSP430 Assembler Code Template for use with TI Code Composer Studio
;
;
;-------------------------------------------------------------------------------
            .cdecls C,LIST,"msp430.h"       ; Include device header file

;-------------------------------------------------------------------------------
            .def    RESET                   ; Export program entry-point to
                                            ; make it known to linker.
;-------------------------------------------------------------------------------
            .text                           ; Assemble into program memory.
            .retain                         ; Override ELF conditional linking
                                            ; and retain current section.
            .retainrefs                     ; And retain any sections that have
                                            ; references to current section.

;-------------------------------------------------------------------------------
RESET       mov.w   #__STACK_END,SP         ; Initialize stackpointer
StopWDT     mov.w   #WDTPW|WDTHOLD,&WDTCTL  ; Stop watchdog timer


;-------------------------------------------------------------------------------
; Main loop here
;-------------------------------------------------------------------------------

_start:

        bis.b #0xFF, P3DIR ;output for 7-seg disp

        bis.b #BIT0, P2DIR ;output for green led
        bis.b #BIT2, P2DIR ;output for blue led
        bis.b #BIT4, P7DIR ;output for red led

        bic.b #BIT5, P1DIR ;set direction for pin 1.5 (tilt switch)
        bis.b #BIT5, P1REN ;enable pullup resistor
        bis.b #BIT5, P1OUT ;set output register to 1

        bic.b #BIT2, P1DIR ;set direction for Pin 1.2 (red led)
        bis.b #BIT2, P1REN ;enable pullup resistor
        bis.b #BIT2, P1OUT ;set output register to 1

        bic.b #BIT3, P1DIR ;set direction for Pin 1.3 (blue led)
        bis.b #BIT3, P1REN ;enable pullup resistor
        bis.b #BIT3, P1OUT ;set output register to 1

        bic.b #BIT4, P1DIR ;set direction for Pin 1.4 (green led)
        bis.b #BIT4, P1REN ;enable pullup resistor
        bis.b #BIT4, P1OUT ;set output register to 1
```

```
tilt_check      call #red_test
                        bit.b #BIT5, P1IN       ;check if switch is on/off
                                jnz increase            ;jump to increase label if
on
                                jz  decrease            ;jump to decrease label if
off
                        ret

tilt_checku call #red_test                      ;check led buttons
                        bit.b #BIT5, P1IN               ;test tilt switch
                                jz decrease                     ;jump to decrease if
tilt switch off
                                ret                                     ;else return

tilt_checkd call #red_test                      ;check led buttons
                        bit.b #BIT5, P1IN               ;test tilt switch
                                jnz increase            ;jump to increase if on
                                ret                                     ;else return

red_test    bit.b #BIT2, P1IN     ;check if red button is on/off
                        jz red_light      ;if on, jump to red_light label

blue_check  bit.b #BIT3, P1IN     ;check if blue button is on/off
                                jz blue_light     ;if on, jump to blue_light label

green_check bit.b #BIT4, P1IN     ;check if green button is on/off
                                jz green_light    ;if on, jump to green_light label

                                ret                                 ;return to call
command

red_light       bic.b #BIT4, P7OUT              ;turn on red led
                        bis.b #BIT2, P2OUT              ;turn off blue led
                        bis.b #BIT0, P2OUT              ;turn off green led

        jmp tilt_check

blue_light      bis.b #BIT4, P7OUT              ;turn off red led
                        bic.b #BIT2, P2OUT              ;turn on blue led
                        bis.b #BIT0, P2OUT              ;turn off green led

        jmp tilt_check

green_light     bis.b #BIT4, P7OUT              ;turn off red led
                        bis.b #BIT2, P2OUT              ;turn off blue led
                        bic.b #BIT0, P2OUT              ;turn on green led

        jmp tilt_check
        ret
```

```
increase        mov.b #00010001b, P3OUT         ;display 0 on 7-seg disp
                        call #delay                             ;jump to
code from delay label
                        call #tilt_checku                       ;jump back
to red_test label and test for any RGB buttons

                mov.b #10011111b, P3OUT         ;display 1 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #00110010b, P3OUT         ;display 2 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #00010110b, P3OUT         ;display 3 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #10011100b, P3OUT         ;display 4 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #01010100b, P3OUT         ;display 5 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #01010000b, P3OUT         ;display 6 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #00011111b, P3OUT         ;display 7 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #00010000b, P3OUT         ;display 8 on 7-seg disp
                        call #delay
                        call #tilt_checku

                mov.b #00010100b, P3OUT         ;display 9 on 7-seg disp
                        call #delay
                        call #tilt_checku

                        jmp tilt_check                          ;jump back
to test if tilt switch is off

decrease        mov.b #00010100b, P3OUT         ;display 9 on 7-seg disp
                        call #delay
                        call #tilt_checkd
```

```
                        mov.b #00010000b, P3OUT         ;display 8 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #00011111b, P3OUT         ;display 7 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #01010000b, P3OUT         ;display 6 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #01010100b, P3OUT         ;display 5 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #10011100b, P3OUT         ;display 4 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #00010110b, P3OUT         ;display 3 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #00110010b, P3OUT         ;display 2 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #10011111b, P3OUT         ;display 1 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                        mov.b #00010001b, P3OUT         ;display 0 on 7-seg disp
                                call #delay
                                call #tilt_checkd

                                jmp tilt_check                  ;jump back to test
if tilt switch is on


delay   mov.w #0x00ffffff, R12                  ;sets high number in register 6
timer           dec R12                                 ;decreases the number stored
in register 6
                        jnz timer                               ;jumps back a line
until register 6 is 0
                        ret


                nop
```

```
;-------------------------------------------------------------------------------
; Stack Pointer definition
;-------------------------------------------------------------------------------
            .global __STACK_END
            .sect   .stack

;-------------------------------------------------------------------------------
; Interrupt Vectors
;-------------------------------------------------------------------------------
            .sect   ".reset"                ; MSP430 RESET Vector
            .short  RESET
```