Using 3 channels of Timer_A

```
// Red LED toggled by Channel 0 each 0.75 seconds (9000 cycles @ 12 KHz)
// Green LED toggled by Channel 1 each 0.3 seconds (3600 cycles @ 12 KHz)
// Both LEDs turned on/off each 4 seconds (48000 cycles @ 12 KHz)

#include <msp430g2553.h>
#define redLED 0x01
#define greenLED 0x40

int main(void) {
    WDTCTL = WDTPW | WDTHOLD;

    // Source ACLK from VLO
    BCSCTL1 &= ~XTS;                   // XTS=0
    BCSCTL3 &= ~LFXT1S_3;        // Clear LFXT1S
    BCSCTL3 |= LFXT1S_2;         // LFXT1S = 2 (VLO)

    P1DIR |= (redLED|greenLED);
    P1OUT &= ~(redLED|greenLED);

    // (Continuous mode) (ACLK)
    TACTL = MC_2 | TASSEL_1 | ID_0 | TACLR;

    TACCR0 = (9000-1);
    TACCTL0 &= ~CCIFG;
    TACCTL0 |= CCIE;

    TACCR1 = (3600-1);
    TACCTL1 &= ~CCIFG;
    TACCTL1 |= CCIE;

    TACCR2 = (48000-1);
    TACCTL2 &= ~CCIFG;
    TACCTL2 |= CCIE;

    _low_power_mode_3();         // Also enables global interrupts
}

#pragma vector = TIMER0_A0_VECTOR
__interrupt void TA0_ISR() {
       P1OUT ^= redLED;
       TACCR0 += 9000;
       // Flag cleared automatically
}
```

```
#pragma vector = TIMER0_A1_VECTOR
__interrupt void TA1_ISR() {
    static int flashing = 1;

    // Channel 1
    if((TACCTL1 & CCIFG) != 0) {
        P1OUT ^= greenLED;
        TACCR1 += 3600;
        TACCTL1 &= ~CCIFG;
    }

    // Channel 2
    if((TACCTL2 & CCIFG) != 0) {
        if(flashing==1) {
            P1OUT &= ~(redLED | greenLED);   // Turn LEDs off
            TACCTL0 &= ~CCIE;                       // Disable interrupt
            TACCTL1 &= ~CCIE;
            flashing = 0;
        }
        else {
            TACCR0 = TA0R+9000;                     // Retime new interval
            TACCR1 = TA0R+3600;
            TACCTL0 &= ~CCIFG;                      // Clear flag
            TACCTL1 &= ~CCIFG;
            TACCTL0 |= CCIE;                        // Toggle interrupt
            TACCTL1 |= CCIE;
            flashing = 1;
        }

        TACCR2 += 48000;
        TACCTL2 &= ~CCIFG;
    }
}
```