

Timer_A Output, LED duty cycle

```
// Varying the duty cycle (brightness) of LED by Timer_A

// Green LED is lit at 100 Hz with a varying duty cycle
// Green LED is controlled by Timer_A0 in output mode 7 (reset/set)

// At a flashing rate of 100 Hz and ACLK=12 KHz, the period is 120 cycles
// The LED duty cycle is either of (20, 40, 60, 80 cycles) out of 120 cycles
// This gives 4 brightness levels

// The brightness level is updated each 1 second
// This time is measured by Timer_A1 in the up mode
// Note: The two Timer_A are used (Timer_A0 and Timer_A1)

#include <msp430g2553.h>
#define redLED 0x01
#define greenLED 0x40
#define pushButton 0x08

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);
    P1SEL |= greenLED; // P1.6 controlled as TA0.1

    P1DIR &= ~pushButton;
    P1REN |= pushButton;
    P1IFG &= ~pushButton;
    P1IE |= pushButton;

    // (Up mode) (ACLK)
    TACTL = MC_1 | TASSEL_1 | ID_0 | TACLK;
    TACCR0 = (120-1); // 100 Hz

    // Channel 1 output mode 7 (reset/set)
    TACCTL1 = OUTMOD_7;
    TACCR1=0;

    // Timer_A1
    TA1CTL = MC_1 | TASSEL_1 | ID_0 | TACLK;
    TA1CCR0 = 12000;
    TA1CCTL0 &= ~CCIFG;
    TA1CCTL0 |= CCIE;

    _low_power_mode_3(); // Also enables global interrupts
}
```

```
#pragma vector = TIMER1_A0_VECTOR
__interrupt void TA1_0_ISR(){
    if( TACCR1 == 80)
        TACCR1 = 0;
    else TACCR1 += 20;
}
```