

```
//Original code is here,  
//http://osx-launchpad.blogspot.com/2010/11/breathing-led-effect-with-launchpad.html  
//This is slight modification to work with Ti CCS devtool
```

```
#include "msp430g2553.h"
```

```
int index = 0;    // Index to PWM's duty cycle table (= brightness)
```

```
const unsigned char curve[] = {  
    1,    1,    1,    1,    1,    1,    1,    1,  
    1,    1,    1,    1,    1,    1,    1,    1,  
    1,    1,    1,    2,    2,    2,    2,    2,  
    2,    2,    3,    3,    3,    3,    3,    3,  
    4,    4,    4,    4,    4,    5,    5,    5,  
    5,    6,    6,    6,    6,    7,    7,    7,  
    8,    8,    8,    8,    9,    9,    9,    10,  
    10,   10,   11,   11,   11,   12,   12,   13,  
    13,   13,   14,   14,   15,   15,   15,   16,  
    16,   17,   17,   18,   18,   18,   19,   19,  
    20,   20,   21,   21,   22,   22,   23,   23,  
    24,   24,   25,   25,   26,   26,   27,   27,  
    28,   29,   29,   30,   30,   31,   31,   32,  
    33,   33,   34,   34,   35,   36,   36,   37,  
    38,   38,   39,   39,   40,   41,   41,   42,  
    43,   43,   44,   45,   46,   46,   47,   48,  
    48,   49,   50,   50,   51,   52,   53,   53,  
    54,   55,   56,   56,   57,   58,   59,   59,  
    60,   61,   62,   62,   63,   64,   65,   66,  
    66,   67,   68,   69,   70,   70,   71,   72,  
    73,   74,   75,   75,   76,   77,   78,   79,  
    80,   80,   81,   82,   83,   84,   85,   86,  
    87,   87,   88,   89,   90,   91,   92,   93,  
    94,   95,   95,   96,   97,   98,   99,   100,  
    101,  102,  103,  104,  105,  106,  106,  107,  
    108,  109,  110,  111,  112,  113,  114,  115,  
    116,  117,  118,  119,  120,  121,  122,  122,  
    123,  124,  125,  126,  127,  128,  129,  130,  
    131,  132,  133,  134,  135,  136,  137,  138,  
    139,  140,  141,  142,  143,  144,  145,  146,  
    147,  148,  149,  150,  151,  152,  153,  154,  
    155,  156  
};
```

```
int main(void)  
{  
    // Stop watchdog  
    WDTCTL = WDTPW + WDTHOLD;  
  
    // Set clock to 1 MHz  
    DCOCTL= 0;  
    BCSCTL1= CALBC1_1MHZ;  
    DCOCTL= CALDCO_1MHZ;  
  
    // SMCLK = 1 MHz / 8 = 125 KHz (SLAU144E p.5-15)  
    BCSCTL2 |= DIVS_3;
```

```

// Make P1.6 (green led) an output. SLAU144E p.8-3
P1DIR |= BIT6;

// P1.6 = TA0.1 (timer A's output). SLAS694C p.41
P1SEL |= BIT6;

// PWM period = 125 KHz / 625 = 200 Hz
TACCR0 = 625;

// Source Timer A from SMCLK (TASSEL_2), up mode (MC_1).
// Up mode counts up to TACCR0. SLAU144E p.12-20
TACTL = TASSEL_2 | MC_1;

// OUTMOD_7 = Reset/set output when the timer counts to TACCR1/TACCR0
// CCIE = Interrupt when timer counts to TACCR1
TACCTL1 = OUTMOD_7 | CCIE;

// Initial CCR1 (= brightness)
TACCR1 = 0;

// LPM0 (shut down the CPU) with interrupts enabled
__bis_SR_register(CPUOFF | GIE);
}

// This will be called when timer counts to TACCR1.
#pragma vector=TIMER0_A1_VECTOR
__interrupt void Timer_A(void)
{
    int new_ccr1 = 1;

    // Clear interrupt flag
    TACCTL1 &= ~CCIFG;

    if (index < 500) {
        new_ccr1 = curve[index++ >> 1];
    } else if (index < 1000) {
        new_ccr1 = curve[(999 - index++) >> 1];
    } else {
        index = 0;
    }

    // Wait to set the new TACCR1 until TAR has gone past it, so that we
    // don't get interrupted again in this period.
    while (TAR <= new_ccr1);
    TACCR1 = new_ccr1;
}

```