```c
/*
-------------------------------------------------------------
2.996 / 6.971: Biomedical Devices Design Laboratory
Lab Example Software - Timers

This example sets up the 16MHz clock and uses it to drive
Timer A at 1MHz. Timer A then creates a 200Hz PWM waveform
on pins 1.2 and 1.3, one with a 25% duty cylce and one with a
75% duty cycle.

SC - 9/30/2007
-------------------------------------------------------------
*/

#include "msp430x22x4.h"

void main(void)
{
  // stop watchdog timer
  WDTCTL = WDTPW | WDTHOLD;

  // Clock Setup:
  // -------------------------------------------------
  // XT2 not used, LFXT1 set to high-frequency mode
  // no divider for ACLK (full 16MHz)
  BCSCTL1 = XT2OFF | XTS;
  // set MCLK as LFXT1 (16MHz), no divider
  // also set SMCLK as LFXT1, but divide by 4 (4MHz)
  BCSCTL2 = SELM1 | SELM0 | SELS | DIVS1;
  // set LFXT1 to 3-16MHz range
  BCSCTL3 = LFXT1S1;
  // See User's Guide, 5-14 thru 5-16.
  // -------------------------------------------------

  // Pin Setup:
  // -------------------------------------------------
  // set P1.2 and P1.3 as outputs
  P1DIR = BIT2 | BIT3;
  // select P1.2 and P1.3 to be controlled by Timer A
  P1SEL = BIT2 | BIT3;
  // -------------------------------------------------

  // Timer A Setup:
  // -------------------------------------------------
  // clock source = SMCLK (4MHz), divide by 4 (1MHz)
  TACTL = TASSEL1 | ID1;
  // count up to this number, then reset:
  TACCR0 = 5000;  // 5ms period, 200Hz
  // used to set duty cycles:
  TACCR1 = 1250;  // 25% of full period
  TACCR2 = 1250;  // 25% of full period
  // Timer A, output 1 (P1.2) will be set when timer
  // overflows, reset when it counts past TACCR1
  TACCTL1 = OUTMOD2 | OUTMOD1 | OUTMOD0;
  // Timer A, output 2 (P1.3) will be reset when timer
  // overflows, set when it counts past TACCR2
  TACCTL2 = OUTMOD1 | OUTMOD0;
  // start counting
  TACTL |= MC0;
  // Note: No interrupt service routines are needed
  // to generate outputs at the pins, it is handle
  // entirely by the timer.
  // See User's Guide, 12-20 thru 12-24.
  // -------------------------------------------------

  while(1); // loop forever

}
```