```c
/*
    File : pwm.c
    purpose:
            The program demonstrate the how to generate a PWM pulse to control something.
            For example, servo motor. Timer0 will set to 2.5ms, and there will allocate
            8 jobs in timer0. the pwm control is in job1 and every 2.5ms*8 = 20ms will
            execute pwm pulse one.
            In job1, pwm pulse, it will raise the pwm control pin to high, then start
            timer2, when timer2 time up, it will immediately fall the pwm control. and
            generte a pwm pulse.
*/

#include <reg52.h>
#include <stdio.h>

sbit PWM_CON = P1^5;

/*___________ T I M E R 2 ____________________________________________________________*/

void init_Timer2(int count)
{
    RCAP2H = (65536-count)/256;     // register pair for reload.
    RCAP2L = (65536-count)%256;
    T2CON = 0x04;
    IE |=0xA2;
    TH2 = RCAP2H;
    TL2 = RCAP2L;
}

void turnOnTimer2(void)
{
    ET2=1;
    TR2=1;
}

void turnOffTimer2(void)
{
    ET2=0;
    TR2=0;
}

void set_Timer2(int count)
{
    RCAP2H = (65536-count)/256;     // register pair for reload.
    RCAP2L = (65536-count)%256;
    TH2 = (65536-count)/256;
    TL2 = (65536-count)%256;
    turnOffTimer2();
}
/*___________ T I M E R 0 ____________________________________________________________*/

// initalize 16 bits timer0, with time interval equal 2.5ms
void init_Timer0(void)
{
    TMOD |= 0x01;           // set time0 as mode0
    TH0=(65536-2304)/256;   // count 2304 machine cycle
    TL0=(65536-2304)%256;   // 1 machine cycle = 12/11.0598M = 1.085us
    IE |=0x82;              // 2304x1.085u=2.5ms
    TR0=1;
}

// turn on timer0
void turnOnTimer0(void)
{
    TR0=1;
}

// turn off timer0
void turnOffTimer0(void)
{
    TR0=0;
}

/*___________ M A I N  P R O G R A M__________________________________________________*/
```

```c
void main(void)
{
    init_Timer2(1000);
    turnOffTimer2();

    init_Timer0();
    while(1);
}

/*___________ I N T E R R U P T   R O U N T I N E S___________________________________*/

void timer2_ISR (void) interrupt 5
{
    TF2=0;              // cleared by software
    PWM_CON=0;
    turnOffTimer2();    // execute rountine once.
}

void timer0_ISR (void) interrupt 1
{
    static int task_index;
    static int pulse;

    TH0=(65536-2304)/256;   // count 2304 machine cycle
    TL0=(65536-2304)%256;   // 1 machine cycle = 12/11.0598M = 1.085us

    switch(++task_index)
    {
        case 1 :
                 set_Timer2(2000);
                 PWM_CON=1;         //When timer2 time up, it will causes PWM_CON
                 turnOnTimer2();    //fall to low. and generate a PWM.
                 break;
        case 2 : break;
        case 3 : break;
        case 4 : break;
        case 5 : break;
        case 6 : break;
        case 7 : break;
        case 8 : task_index=0; break;
        default : task_index=0;
    }
}
```