```c
/*
    Name: Interrupt.c
    Purpose:
        provide the function for easier access the interrupt, or timer setting.
*/


/*___________ I N C L U D E S _____________________________________________________*/

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

/*___________ B A U D   R A T E   9 6 0 0 _____________________________________________*/

// com port with 9600 baud with crystal 11.0592MHz.
void init_uart(void)
{
    ES = 0;                 //disable serial port interrupt
    SCON  = 0x50;           //serial port mode, enable reception
    TMOD &=0x0F;            // Clear only Timer 1 control bits
    TMOD |= 0x20;           //Timer1 set up in Mode2 - 8 bits auto-reload timer
    TH1=253;
    TI    = 1;              //enable transmit interrupt
    TR1   = 1;
}

/*___________S E T   T I M E R 1 _____________________________________________________*/

// 16bit auto reload timer2. Before set the timer1, wait until all the data is transmited.
void set_timer1(int count)
{
    while (!TI);            //wait until all the data sended out to RS232
    EA = 1;
    TR1 = 0;
    TF1 = 0;
    TMOD &=0x0F;            // Clear only Timer 1 control bits
    TMOD |=0x10;
    TH1=(65536-count)/256;  // count 2304 machine cycle
    TL1=(65536-count)%256;  // 1 machine cycle = 12/11.0598M = 1.085us
}

// turn on/ off the timer1, return is it serial mode or not.
void turnOnTimer1(void) {   ET1=1;  TR1=1;  }
void turnOffTimer1(void){   ET1=0;  TR1=0;  }

/*___________S E T   T I M E R 2 _____________________________________________________*/

// 16bit auto reload timer2
void init_timer2(int count)
{
    RCAP2H = (65536-count)/256;     // register pair for reload.
    RCAP2L = (65536-count)%256;
    IE = 0x80;
    TH2 = RCAP2H;
    TL2 = RCAP2L;
}

//turn on/off the timer2, report state
void turnOnTimer2(void) {   ET2=1;  TR2=1;  }
void turnOffTimer2(void){   ET2=0;  TR2=0;  }
bit isTimer2On(void)    {   return TR2;     }

/*___________S E T   T I M E R 0 _____________________________________________________*/

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

// turn on/0ff timer0
```

```
77  void turnOnTimer0(void) {   TR0=1;  }
78  void turnOffTimer0(void){   TR0=0;  }
79
80
81
82
83
84
85
86
```

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

/*___________ I N T E R R U P T  R O U N T I N E S_____________________________*/
//execute the interrupt routine one time
void timer1_ISR (void) interrupt 3 using 3
{
    TF1=0;              //cleared by software
    /*
        user program code
        .
        .
        .
    */
}

//execute the interrupt routine one time
void timer2_ISR (void) interrupt 5
{
    TF2=0;              // cleared by software
    /*
        user program code
        .
        .
        .
    */

}

```

```
/*
    Name : InterruptConfig.h
    purpose:
        header for the frequencyCounter.c and Interrupt.c

*/


/*_____________ P R O T O C O L ____________________________________________________________*/

// Interrupt.c
void init_uart(void);               // baud rate 9600
void set_timer1(int count);         // Timer1
void turnOnTimer1(void);
void turnOffTimer1(void);

void init_timer2(int count);        // Timer2, int .count. the number of pulse.
void turnOnTimer2(void);
void turnOffTimer2(void);
bit isTimer2On(void);

void init_Timer0(int count);        // Timer0
void turnOnTimer0(void);
void turnOffTimer0(void);


```