D:\Project_Development\TaskSharingDemonstration\TaskSharingMain.c 04/21/03 12:40:00

```
/*------------------------------------------------------------------------------+
|                                                                               |
|                            Multi-Tasking of 8 jobs                            |
|                                                                               |
| Name: TaskSharingMain.c                                                       |
| Purpose:                                                                      |
|       Demostrate the usage of implementing the multitasking jobs, from job 1 to 8 |
|       Each 50ms will execute the 1 job, therefore for job 1 to execute again. It  |
|       is required to wait 7 jobs complete, that is 7x50ms.                    |
| Example:                                                                      |
|       50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms            |
|         ^      ^      ^      ^      ^      ^      ^      ^      ^             |
|       job1   job2   job3   job4   job5   job6   job7   job8   job1  ..... again.. |
+------------------------------------------------------------------------------*/
#include <reg51.h>
#include <stdio.h>
#include "TaskSharing\InterruptConfig.h"

void main(void)
{
    init_uart();                // baud rate 9600
    init_Timer0();              // start Timer 0
    turnOnTimer0();


    printf("\nHello");
    while(1);
}
```

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

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

#include <reg51.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)
{
    SCON  = 0x50;
    TMOD |= 0x20;
    TH1   = 253;
    TR1   = 1;
    TI    = 1;
}


/*___________ T I M E R 0 ___________________________________________________*/


// initalize 16 bits timer0, with time interval equal 0.05s
void init_Timer0(void)
{
    TMOD |= 0x01;            // set time0 as mode0
    TH0=(65536-46079)/256;  // count 46080 machine cycle
    TL0=(65536-46079)%256;  // 1 machine cycle = 12/11.0598M = 1.085us
    IE |=0x82;              // 46080x1.085u=0.05s
    TR0=1;
}

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

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

// check whether timer 0 is on or off. ON return 1, OFF return 0;
bit isTimer0ON(void)
{
    if(TR0)
        return 1;
    else
        return 0;
}


/*___________ I N T E R R U P T _____________________________________________*/


// initalize external interrupt 0 (P3.2)
void init_int0(void)
{
    PX0=1;      //Define Int0 high priority
    IE0=0;      //External Interrupt 0 edge flag, set when external interrupt
                //detected,cleared when interrupt is processed.
    IT0=1;      //set to specific falling edge produce interrupt
    EX0=1;      //enable External Interrupt 0
    EA=1;       //enable all interrupt
}

// enable external interrupt 0
void enINT0(void)
{
    EX0=1;
}

// disable external interrupt 0
void disINT0(void)
{
    EX0=0;
}
```

D:\Project_Development\TaskSharingDemonstration\TaskSharing\TaskSharing.c 04/21/03 12:37:00

```
/*----------------------------------------------------------------------------------+
|                                                                                   |
| Name: TaskSharing.c                                                               |
| Purpose:                                                                          |
|       The c files provde the functions to implement the multitasking jobs, 1-8    |
|       Each 50ms will execute the 1 job, therefore for job 1 to execute again. It  |
|       is required to wait 7 jobs complete, that is 7x50ms.                        |
| Example:                                                                          |
|       50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms , 50ms                |
|         ^      ^      ^      ^      ^      ^      ^      ^      ^                |
|       job1   job2   job3   job4   job5   job6   job7   job8   job1  ..... again.. |
+----------------------------------------------------------------------------------*/

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

#include <reg51.h>
#include <stdio.h>
#include "InterruptConfig.h"

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

/*
    when external interrupt occur, count will increase 1 and toggle P1
*/
void ExInt(void) interrupt 0
{
    printf("\nExternal Interrupt occur");
}


/*
    when timer0 time up 50ms, execute  the code.
*/
void timer0_ISR (void) interrupt 1
{
    static task_index;
    TH0=(65536-46079)/256; // count 46080 machine cycle
    TL0=(65536-46079)%256; // 1 machine cycle = 12/11.0598M = 1.085us
    switch(++task_index)
    {
         case 1 : printf("1"); break;
         case 2 : printf("2"); break;
         case 3 : printf("3"); break;
         case 4 : printf("4"); break;
         case 5 : printf("5"); break;
         case 6 : printf("6"); break;
         case 7 : printf("7"); break;
         case 8 : printf("8"); task_index=0; break;
         default : task_index=0;
    }
}
```

```
/*
    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 init_Timer0(void);             // Timer0
void turnOnTimer0(void);
void turnOffTimer0(void);
bit isTimer0ON(void);
void init_int0(void);               // External Interrupt 0
void enINT0(void);
void disINT0(void);

//FrequencyCounter.c
int getFrequency(void);             // counter the number of pules within one second.
```