

```

/*-----+
|
|           AT89C51 as an Frequency Counter
|
| Name: FrequencyCounter.c
|
| Purpose:
|   Using 8951 as an Frequency Counter, count the pules that trigger the interrupt
|   pin. Each sec, the timer overflow rountine will print out the pules number
|   And that is a reflection of frequency.
|-----*/

/*_____ I N C L U D E S _____*/

#include <reg51.h>
#include <stdio.h>
#include <ctype.h>           //use toint()

unsigned int frequency = 0;
unsigned int ia = 20;       //20x0.05s=1s

// com port with 9600 baud with crystal 11.0592MHz.
void init_uart(void)
{
    SCON = 0x50;
    TMOD |= 0x20;
    TH1 = 253;
    TR1 = 1;
    TI = 1;
}

// initialize 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;
}

// initialize 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
}

/*
   implement the pulse counting for the detection of frequency
*/
void main(void)
{
    init_uart();          // 9600 baud @ 11.0592MHz
    init_int0();          // enable INT0
    init_Timer0();        // 0.05s time interval of time 0;

    printf("Frequency Counter\n");
    P1=0xcc;
    while(1);
}

// when external interrupt occur, count will increase 1 and toggle P1
void ExInt(void) interrupt 0
{
    frequency++;
}

// when timer0 time up, execute the code.
void timer1_ISR (void) interrupt 1
{
    TH0=(65536-46900)/256; // reset the timer 0.
    TL0=(65536-46900)%256; //fine tune. as crystal may not exactly 11.0598MHz
    --ia;
    if(ia<=0)             // each second print out the number of pules
    {                       // that represent the frequency.

```

```
    printf("\rFrequency : %d      \r",frequency);  
    frequency=0;           // reset.  
    P1=~P1;  
    ia=20;  
    }  
}
```