

```
#include <avr\io.h>
#include <avr\interrupt.h>
unsigned char  tovCnt;
unsigned long  freq;
void delayby100ms(unsigned char cx);
void main (void)
{
    tovCnt = 0;
    DDRD  &= ~0x40;    // configure T1 for input
    TCCR1A = 0;        // configure Timer 1 to normal mode
    TCCR1B = 0;        // stop Timer 1
    TCNT1H = 0;        // let Timer 1 to count up from 0
    TCNT1L = 0;        // "
    TIFR1 = 1 << TOV1; // clear TOV1 flag
    TIMSK1 = 1 << TOIE1; // enable TOV1 interrupt
    TCCR1B = 0x07;     // enable Timer 1 to count using T1 input as clock source
    sei();             // enable interrupt globally
    delayby100ms(10); // wait for one second
    TCCR1B = 0;        // stop Timer 1
    freq = (long)tovCnt * 65536 + (long)TCNT1;
    while(1);
}
void delayby100ms(unsigned char cx)
{
    TCCR3A = 0;        // configure Timer 1 to normal mode with clock source
    TCCR3B = 0x03;     // set to clk_I/0 / 64
    while(cx){
        TCNT3 = 40536; // let timer1 to count up from 40536
        TIFR3 = 1<<TOV3; // clear TOV1 flag
        while(!(TIFR3 & (1<<TOV3))); // wait until TOV1 is set to 1
        cx--;
    }
}
// -----<
// Timer 1 overflow interrupt service routine.
// -----<
ISR (TIMER1_OVF_vect)
{
    tovCnt++;
}
```