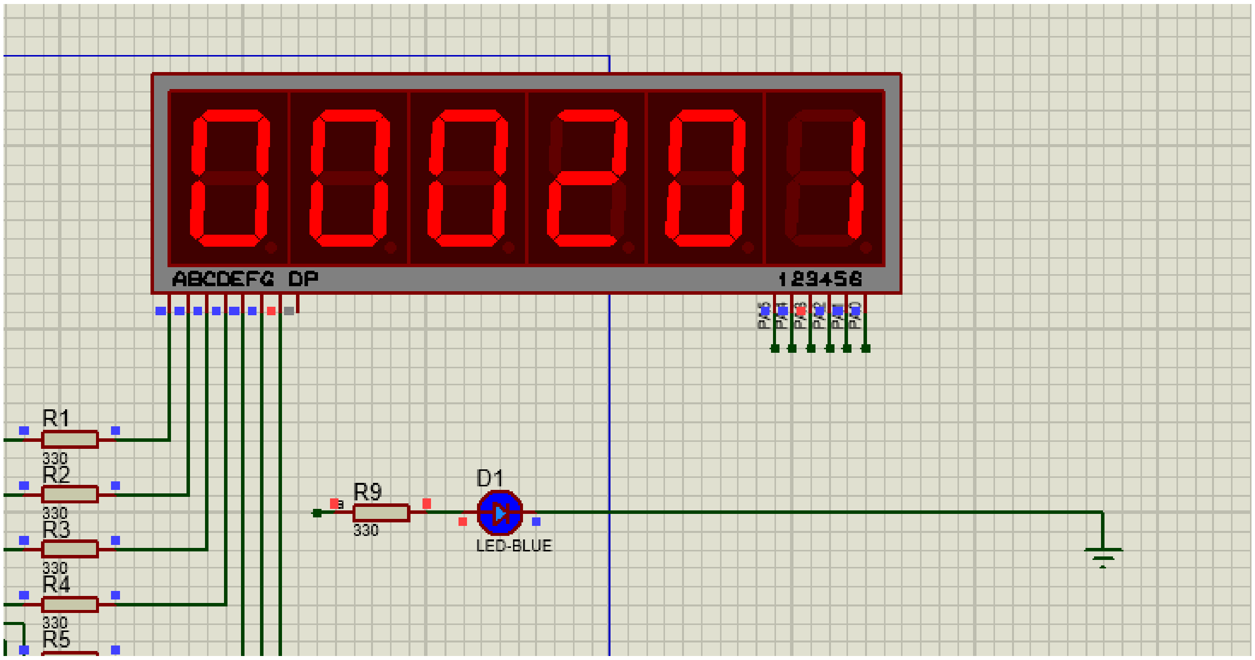
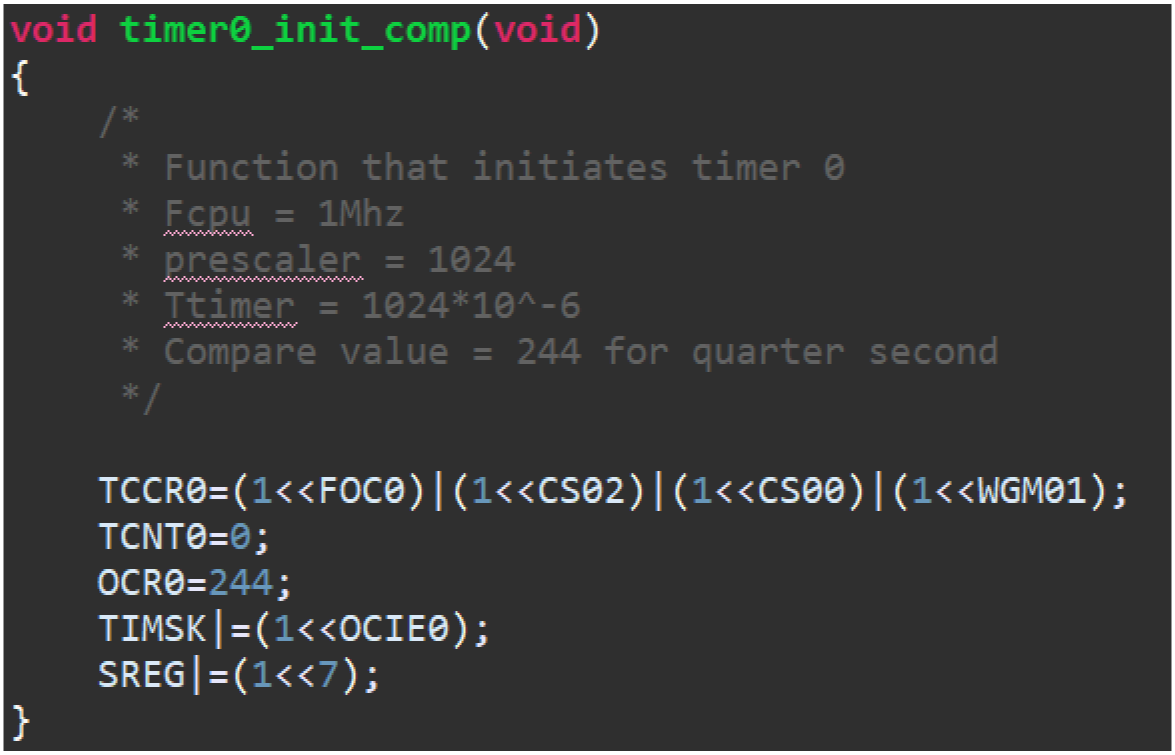
In the project I have added a function which works as a timer that makes a LED blink after reaching a preset value which makes the stopwatch works as a timer as well.



This function is implemented by using Timer_0 which gives an interrupt each 0.25 second.



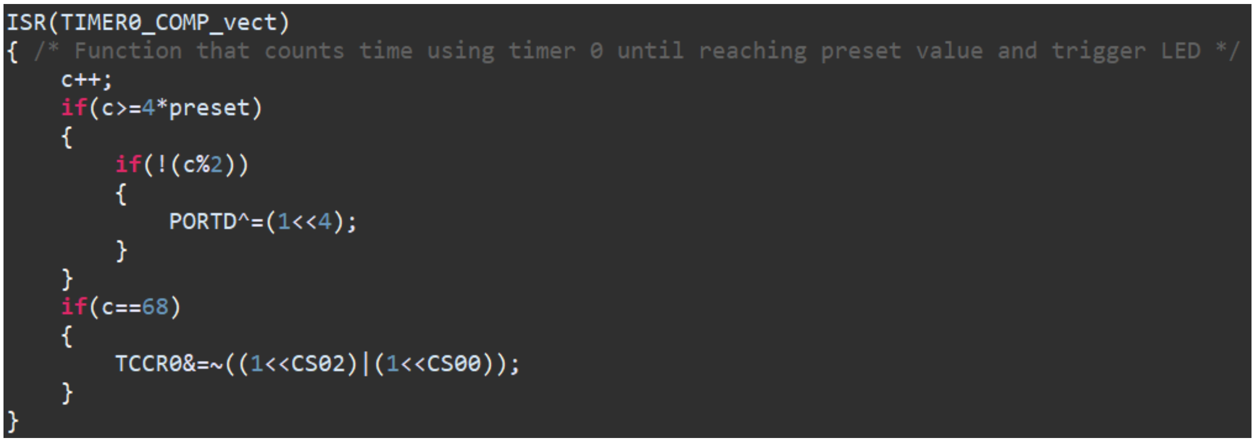
```
void timer0_init_comp(void)
{
    /*
     * Function that initiates timer 0
     * Fcpu = 1Mhz
     * prescaler = 1024
     * Ttimer = 1024*10^-6
     * Compare value = 244 for quarter second
     */

    TCCR0=(1<<FOC0)|(1<<CS02)|(1<<CS00)|(1<<WGM01);
    TCNT0=0;
    OCR0=244;
    TIMSK|=(1<<OCIE0);
    SREG|=(1<<7);
}
```

The ISR of the interrupt caused by Timer_0 increments the values c,

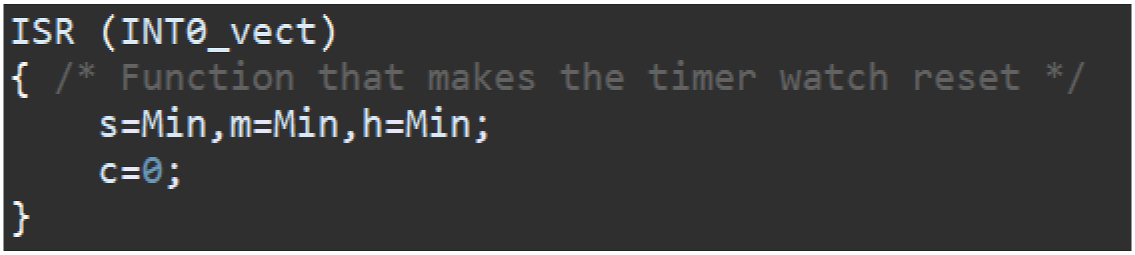
The preset value is specified by the user in units of seconds,

When the value of c reaches the value of seconds specified by the user, it blinks the LED each half a second.

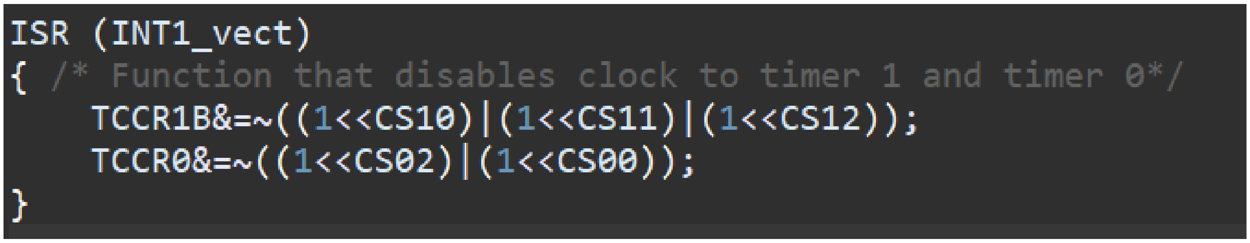


```
ISR(TIMER0_COMP_vect)
{ /* Function that counts time using timer 0 until reaching preset value and trigger LED */
    c++;
    if(c>=4*preset)
    {
        if(!(c%2))
        {
            PORTD^=(1<<4);
        }
    }
    if(c==68)
    {
        TCCR0&=~((1<<CS02)|(1<<CS00));
    }
}
```

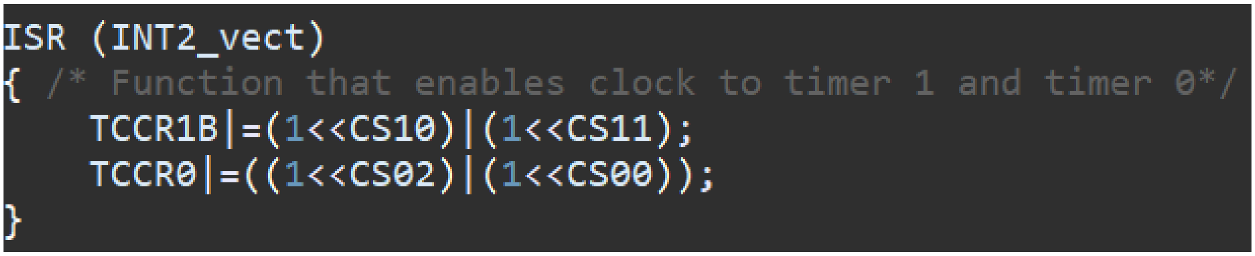
Also, the operation of this function is added to the other ISRs of the project to guarantee seamless performance.



```
ISR (INT0_vect)
{ /* Function that makes the timer watch reset */
    s=Min,m=Min,h=Min;
    c=0;
}
```



```
ISR (INT1_vect)
{ /* Function that disables clock to timer 1 and timer 0*/
    TCCR1B&=~((1<<CS10)|(1<<CS11)|(1<<CS12));
    TCCR0&=~((1<<CS02)|(1<<CS00));
}
```



```
ISR (INT2_vect)
{ /* Function that enables clock to timer 1 and timer 0*/
    TCCR1B|=(1<<CS10)|(1<<CS11);
    TCCR0|=((1<<CS02)|(1<<CS00));
}
```