EXPT. NO. : 6

TITLE: Generation of square Wave using timer with interrupt.

```
#include <p18f4550.h>

void timer_isr(void);

extern void _startup (void);
#pragma code RESET_INTERRUPT_VECTOR = 0x1000

void reset (void)
{
      _asm
         goto _startup
          _endasm
}
#pragma code

#pragma code HIGH_INTERRUPT_VECTOR = 0x1008
void high_ISR (void)
{
         _asm

         goto timer_isr
         _endasm    //The program is relocated to execute the interrupt routine timer_iser

}
#pragma code


// This function is executed as soon as the timer interrupt is generated due to timer overflow
#pragma interrupt timer_isr
void timer_isr(void)
{
         TMR0H = 0X6D;                    // Reloading the timer values after overflow
         TMR0L = 0X82;
         PORTDbits.RD0 = ~PORTDbits.RD0;                    //Toggle the PORTB led outputs RB0 - RB3
         INTCONbits.TMR0IF = 0;                  //Resetting the timer overflow interrupt flag
}


void main()
{
   INTCON2bits.RBPU=0;                //To Activate the internal pull on PORTB
         ADCON1 = 0x0F;
         TRISD = 0;
         PORTD=0;
         T0CON = 0x03;                                    //Set the timer to 16-bit mode,internal instruction
cycle clock,1:256 prescaler
         TMR0H = 0x00;             // Reset Timer0 to 0x48E5 TO MAKE DELAY OF 1 SECOND
         TMR0L = 0x00;
         INTCONbits.GIE = 1;                     // Global interrupt enabled
```

```
        T0CONbits.TMR0ON = 1;                   // Start timer0
        while(1);
}


void delay_ms(unsigned int time)
{
 unsigned int i,j;
  for (i=0;i<time;i++)
        for (j=0; j<710;j++);
}
```