

```

#include <p18f4550.h>
void timer_isr(void);
extern void _startup (void);

// Interrupt vectors
#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_isr
}
#pragma code

// Interrupt service routine for Timer0
#pragma interrupt timer_isr
void timer_isr(void)
{
    TMR0H = 0x00; // Reload TMR0 high byte for 5 ms
    TMR0L = 0xE9; // Reload TMR0 low byte for 5 ms (calculated value)

    PORTDbits.RD0 = ~PORTDbits.RD0; // Toggle PORTD LED (output square wave)

    INTCONbits.TMR0IF = 0; // Clear the Timer0 interrupt flag
}

void main()
{
    // Initialize system
    INTCON2bits.RBPU = 0; // Activate internal pull-ups on PORTB
    ADCON1 = 0x0F;        // Disable all analog inputs
    TRISD = 0;            // Set PORTD as output
    PORTD = 0;            // Initialize PORTD (clear all LEDs)

    // Set up Timer0
    T0CON = 0x03;          // Set Timer0 to 16-bit mode, internal instruction
cycle clock, 1:256 prescaler
    TMR0H = 0x00;          // Set TMR0 high byte for 5 ms delay (part of the 100
Hz square wave)
    TMR0L = 0xE9;          // Set TMR0 low byte for 5 ms delay (calculated for
100 Hz)

    // Enable global interrupt and Timer0 interrupt
    INTCONbits.GIE = 1;    // Enable global interrupt
    T0CONbits.TMR0ON = 1;  // Turn on Timer0

    while (1);             // Infinite loop, the interrupt will handle the rest
}

// Function to generate a 1 ms delay
void delay_ms(unsigned int time)
{
    unsigned int i, j;
    for (i = 0; i < time; i++)

```

```
    for (j = 0; j < 710; j++);  
}
```