

External Interrupt

```
#include <xc.h>
// CONFIG
#pragma config FOSC = HS      // Oscillator Selection bits (RC oscillator)
#pragma config WDTE = OFF     // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF    // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF    // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF      // Low-Voltage (Single-Supply) In-Circuit
Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be
used for programming)
#pragma config CPD = OFF      // Data EEPROM Memory Code Protection bit
(Data EEPROM code protection off)
#pragma config WRT = OFF      // Flash Program Memory Write Enable bits
(Write protection off; all program memory may be written to by EECON
control)
#pragma config CP = OFF       // Flash Program Memory Code Protection
bit (Code protection off)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#define _XTAL_FREQ 20000000
main()
{
    char i;
    TRISC0=TRISC1=0;
    TRISD0=0;
    RD0=0;
    RC0=1;
    RC1=0;
    GIE=1;           //enabling global interrupt
    PEIE=1;          //enabling periferal interrupt
    INTF=0;          //clearing interrupt flag
    INTE=1;          //interrupt enable
    while(1)
    {
        RC0=~RC0;
        RC1=~RC1;
        for(i=0;i<=100;i++)
            __delay_ms(10);
    }
}

interrupt isr()
{
    char i;
    RD0=1;
    for(i=0;i<=100;i++)
        __delay_ms(10);
    RD0=0;
    INTF=0;          //clearing interrupt flag
}
```