EXTERNAL INTERRUPT

```
#include <pic18f4550.h>

#define RELAY_PIN LATAbits.LATA4

void interrupt extint_isr(void)
{
   unsigned int i;
   if(INT1F)
   {
      INT1F = 0;
      INT1IE = 0;
      RELAY_PIN = ~RELAY_PIN;
      for(i=0; i<10000; i++);      //small delay for debouncing
      INT1IE = 1;
   }
}

int main()
{
   ADCON1 = 0x0F;         //set pins as Digital
   TRISAbits.TRISA4 = 0;   //set relay pin RA4 as output
   TRISBbits.TRISB1 = 1;   //Interrupt pin as input
   RELAY_PIN = 1;

   INT1IE  =   1;                 //Enable external interrupt INT1
   INTEDG1 =   0;                 //Interrupt on falling edge
   GIE     =   1;                 // Enable global interrupt

   while(1);
}
```

LED BLINKING

```
#include <p18f4550.h>


void delay(unsigned int time)
{
   unsigned int i,j;
   for(i=0;i<time;i++)
      for(j=0;j<5000;j++);
}


void main(void)
{
  TRISB = 0x00;
  LATB = 0xFF;

  while(1)                    //Loop forever;
  {
     LATB = ~LATB;
     delay(200);
  }
}
```

TEMPERATURE

```
#include <pic18f4550.h>
#include <stdio.h>

#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB

unsigned char str[16];

void lcd_delay(unsigned int time)
{
 unsigned int i , j ;

   for(i = 0; i < time; i++)
   {
     for(j=0;j<100;j++);
   }
}



void SendInstruction(unsigned char command)
{
    LCD_RS = 0;             // RS low : Instruction
    LCDPORT = command;
    LCD_EN = 1;             // EN High
    lcd_delay(10);
    LCD_EN = 0;             // EN Low; command sampled at EN falling edge
    lcd_delay(10);
}
```

```
void SendData(unsigned char lcddata)
{
    LCD_RS = 1;             // RS HIGH : DATA
    LCDPORT = lcddata;
    LCD_EN = 1;             // EN High
    lcd_delay(10);
    LCD_EN = 0;             // EN Low; data sampled at EN falling edge
    lcd_delay(10);
}

void InitLCD(void)
{
   ADCON1 = 0x0F;
   TRISB = 0x00; //set data port as output
   TRISAbits.RA0 = 0; //RS pin
   TRISAbits.RA1 = 0; // EN pin

   SendInstruction(0x38);     //8 bit mode, 2 line,5x7 dots
   SendInstruction(0x06);          //entry mode
   SendInstruction(0x0C);          //Display ON cursor OFF
   SendInstruction(0x01);     //Clear display
   SendInstruction(0x80);     //set address to 0
}

void LCD_display(unsigned int row, unsigned int pos, unsigned char *ch)
{
   if(row==1)
      SendInstruction(0x80 | (pos-1));
   else
      SendInstruction(0xC0 | (pos-1));
```

```
    while(*ch)
        SendData(*ch++);
}

void ADCInit(void)
{
    TRISEbits.RE2 = 1;              //ADC channel 7 input

    ADCON1 = 0b00000111;            //Ref voltages Vdd & Vss; AN0 - AN7 channels Analog
    ADCON2 = 0b10101110;            //Right justified; Acquisition time 4T; Conversion clock Fosc/64
}

unsigned short Read_Temp(void)
{
    ADCON0 = 0b00011101;     //ADC on; Select channel;
    GODONE = 1;             //Start Conversion

    while(GO_DONE == 1 );    //Wait till A/D conversion is complete
    return ADRES;                   //Return ADC result
}

int main(void)
{
    unsigned int temp;
    InitLCD();
    ADCInit();
    LCD_display(1,1,"Temperature:");
    while(1)
    {
        temp = Read_Temp();
        temp = ((temp * 500) / 1023);
```

```
        sprintf(str,"%d'C  ",temp);
        LCD_display(2,1,str);
        lcd_delay(9000);
    }
    return 0;
}
```

TIME BUZZER

```
#include <pic18f4550.h>                    /* Contains PIC18F4550 specifications */

#define Buzzer LATAbits.LATA5              /* Define buzzer pin */
unsigned int count = 0;

void interrupt Timer1_ISR()
{
   if(TMR1IF==1)
   {
   //TMR1=0xCF2C;
   TMR1L = 0x20;
   TMR1H = 0xD1;
   count ++;

   if (count >= 1000) //measure upto 1000 ms i.e. 1 seconds
   {
      Buzzer = ~Buzzer;    /* Toggle buzzer pin  */
      count = 0;  //reset count
   }
   TMR1IF = 0; //timer1 overflow flag to 0
   }
}


void main()
{
   TRISB=0;                              /* Set as output port */
   TRISAbits.TRISA5 = 0;      //set buzzer pin RA5 as output
   GIE=1;                  /* Enable Global Interrupt */
   PEIE=1;                  /* Enable Peripheral Interrupt */
```

```
    TMR1IE=1;                /* Enable Timer1 Overflow Interrupt */
    TMR1IF=0;

    /* Enable 16-bit TMR1 register,no pre-scale,internal clock, timer OFF */
    T1CON=0x80;        /*1:8 prescale*/
    TMR1L = 0x20;
    TMR1H = 0xD1;
    TMR1ON=1;               /* Turn ON Timer1 */

    while(1);
}
```

UART

```
#include<p18F4550.h>
#include<stdio.h>
#define Fosc 48000000UL

void InitUART(unsigned int baudrate)
{
  TRISCbits.RC6 = 0;                    //TX pin set as output
  TRISCbits.RC7 = 1;                    //RX pin set as input

  SPBRG = (unsigned char)(((Fosc /64)/baudrate)-1);
  BAUDCON = 0b00000000;                  //Non-inverted data; 8-bit baudrate generator

  TXSTA = 0b00100000;                   //Asynchronous 8-bit; Transmit enabled; Low speed baudrate
select
  RCSTA = 0b10010000;                   //Serial port enabled; 8-bit data; single receive enabled
}

void SendChar(unsigned char data)
{
    while(TXSTAbits.TRMT == 0);          //Wait while transmit register is empty

    TXREG = data;                       //Transmit data
}

void putch(unsigned char data)
{
    SendChar(data);
}

unsigned char GetChar(void)
```

```
{
   while(!PIR1bits.RCIF);                //Wait till receive buffer becomes full
   return RCREG;                         //Returned received data
}



void main(void)
{
   InitUART(9600);

   printf("\r\nHello MicroPIC-18F: Enter any Key from Keyboard\r\n");

   while(1)
   {
      printf("%c! ",GetChar());           //Receive character from PC and echo back
   }

 while(1);
}
```