

The diagram shows a PIC16F877A microcontroller (U1) interfaced with a relay (RL1) and a buzzer. The PIC is configured with a 10k pull-up resistor (R1) on the MCLR pin (pin 1). The oscillator circuit consists of a crystal (X1) and two 22pF capacitors (C1, C2). The PIC's RB0/INT pin (pin 33) is connected to the base of a BC547BP transistor (Q1) through a 4.7k resistor (R2). The transistor's emitter is grounded, and its collector is connected to the relay coil (RL1) through a 1k resistor (R4). The relay coil is also connected to a 12V battery (B1) and a 5V buzzer. The PIC's RC0/T1OSO/T1CKI pin (pin 15) is connected to the buzzer through a 1k resistor (R6). The PIC's RC1/T1OSI/CCP2 pin (pin 16) is connected to the buzzer through a 1k resistor (R5). The PIC's RC2/CCP1 pin (pin 17) is connected to the buzzer through a 1k resistor (R3). The PIC's RC3/SCK/SCL pin (pin 23) is connected to the buzzer through a 1k resistor (R4). The PIC's RC4/SDI/SDA pin (pin 24) is connected to the buzzer through a 1k resistor (R5). The PIC's RC5/SDO pin (pin 25) is connected to the buzzer through a 1k resistor (R6). The PIC's RC6/TX/CK pin (pin 26) is connected to the buzzer through a 1k resistor (R7). The PIC's RC7/RX/DT pin (pin 27) is connected to the buzzer through a 1k resistor (R8). The PIC's RD0/PSPO pin (pin 19) is connected to the buzzer through a 1k resistor (R9). The PIC's RD1/PSPI pin (pin 20) is connected to the buzzer through a 1k resistor (R10). The PIC's RD2/PSPI pin (pin 22) is connected to the buzzer through a 1k resistor (R11). The PIC's RD3/PSPI pin (pin 27) is connected to the buzzer through a 1k resistor (R12). The PIC's RD4/PSPI pin (pin 28) is connected to the buzzer through a 1k resistor (R13). The PIC's RD5/PSPI pin (pin 29) is connected to the buzzer through a 1k resistor (R14). The PIC's RD6/PSPI pin (pin 29) is connected to the buzzer through a 1k resistor (R15). The PIC's RD7/PSPI pin (pin 30) is connected to the buzzer through a 1k resistor (R16).

```
void main() {
    int bt=0;
    Trisb=0x00; //as output
    Trisc=0xff; // as input
    portb=0x00;
    while(1)
    {

        //button
        if(portc.f0==1)
        {
            delay_ms(150);
            if(portc.f0==1)
            {
                bt++;

                if(bt==10)
                {
                    bt=0;
                }
            }
        }

        if(portc.f1==1)
        {
```

```
if(portc.f1==1)
{
```

```
delay_ms(150);
if(portc.f1==1)
{
    bt++;

    if(bt==10)
    {
        bt=0;
    }
}

if(portc.f0==1)
{
    portb.f0=1;//turn on relay

}

if(portc.f1==1){
    portb.f0=0;//turn off relay
}

}
```