

# Project 3 Interrupt

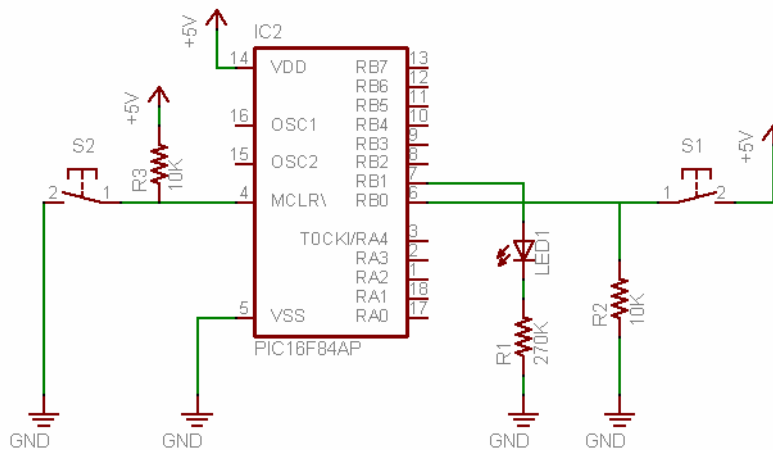
## โจทย์

สร้างวงจรและเขียนโปรแกรมสำหรับ PIC microcontroller ใช้ควบคุมการเปิดปิดไฟ LED ด้วยสวิตช์ในลักษณะ กดติด กดดับ โดยใช้ external interrupt ผ่านทาง RB0/INT (pin 6)

## ขั้นตอนการปฏิบัติ

### 1) การต่อวงจร

ต่อลักษณะเดียวกับวงจรใน Project 1 แต่จะต้องบังคับให้สวิตช์อยู่ที่ตำแหน่งขา RB0/INT (ขาที่6) ส่วน LED สามารถวางไว้ที่ pin ของ port ได้ก็ได้เพียงแต่เขียนโปรแกรมให้ตรงตามวงจร



รูปที่ 1 แผนผังวงจรสำหรับ Project 3

### 2) ขั้นตอนการเขียนโปรแกรม

#### เตรียมตัวก่อนเขียนโปรแกรม

แม้ว่าเป้าหมายของโปรเจกต์จะเหมือนกับโปรเจกต์ 1 แต่่วิธีการแตกต่างกัน ในโปรเจกต์แรกเราใช้โปรแกรมในการวนลูปเพื่อคอยเช็คว่ามีกรกดสวิตช์หรือยัง เราเรียกการวนลูปแบบนี้ว่า polling ซึ่งมีข้อเสียคือจะต้องเสียเวลาหน่วยประมวลผลมากในการคอยรอเช็คสถานะของสวิตช์ วิธีการแก้ปัญหานี้ก็คือการใช้ interrupt ซึ่งเป็นฮาร์ดแวร์ที่คอยตรวจสอบสถานะของพอร์ทซึ่งเมื่อสถานะมีการเปลี่ยนแปลงจะเกิดการขัดจังหวะโปรแกรมให้หยุดทำงานอื่นชั่วคราว แล้วมาจัดการงานที่ต้องการทำเมื่อเกิดการเปลี่ยนแปลงสถานะของพอร์ทนั้นก่อน ซึ่งวิธีการนี้ทำให้หน่วยประมวลผลมีเวลาว่างที่พร้อมจะไปทำงานอย่างอื่นได้ นอกจากการเปลี่ยนแปลงสถานะของพอร์ทแล้ว PIC ยังสามารถรองรับ interrupt จากเหตุการณ์อื่นๆ ได้อีกมากซึ่งเราจะได้ศึกษาในโปรเจกต์ต่อไป

ศึกษารายละเอียดข้อมูลเกี่ยวกับการใช้ interrupt ได้จากหัวข้อที่ 14.6 หน้า 101 จะเห็นว่า PIC16F628A มี interrupt ถึง 10 แหล่ง แต่ที่เราจะใช้ในครั้งนี่คือ External Interrupt RB0/INT ซึ่งมี register ที่จำเป็นต้องกำหนดค่าดังนี้คือ

TRISB0 ให้เป็น 1 (input) สำหรับรับสัญญาณที่มาเป็น interrupt

GIE ให้เป็น 1 สำหรับเปิดการใช้งานกลุ่ม interrupt feature ทั้งหมด (ดู FIG 14-14 หน้า 101)

INTE ให้เป็น 1 สำหรับเปิดการใช้งาน External Interrupt RB0/INT

นอกจากฟังก์ชัน main แล้วเราจะต้องเขียนฟังก์ชันสำหรับทำงานที่ต้องการเมื่อเกิด interrupt ซึ่งมีรูปแบบดังนี้

```
Static void interrupt isr(void)
```

```
{
```

```
    ...//โปรแกรมที่ต้องการให้ microcontroller ทำเมื่อเกิด interrupt
```

```
}
```

โดย สามารถเปลี่ยนชื่อฟังก์ชัน isr ได้แต่ส่วนอื่นต้องคงไว้ และเขียนโปรแกรมที่ต้องการให้ microcontroller ทำเมื่อเกิด interrupt ในวงเล็บ {} แต่ภายในฟังก์ชันนี้เป็นฟังก์ชันที่รองรับการทำงานของ interrupt ทั้งหมด ดังนั้นจึงต้องมีการตรวจสอบอีกขั้นหนึ่งว่าเป็น interrupt จากแหล่งใด โดยใน project นี้เราต้องการใช้ External Interrupt RB0/INT จึงต้องตรวจสอบ flag bit ชื่อ INTF ดังนี้

```
if(INTF==1)
```

```
{
```

```
    ...//โปรแกรมที่ต้องการให้ microcontroller ทำเมื่อเกิด external interrupt ที่ขา
```

```
    //RB0/INT
```

```
}
```

โดยเราจะเขียนโปรแกรมที่ต้องการให้ไมโครคอนโทรลเลอร์ทำเมื่อเกิด external interrupt ที่ขา RB0/INT ภายในเงื่อนไขดังกล่าว

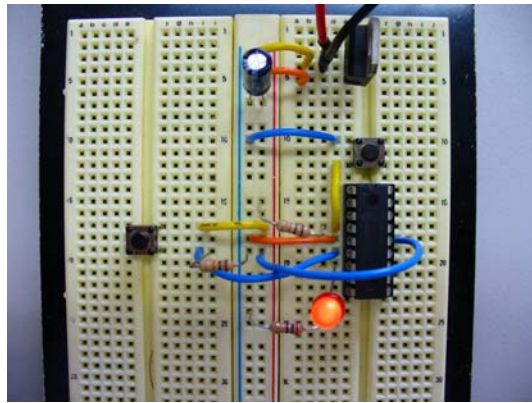
### โปรแกรม project03.c

```
#include<pic.h>
__CONFIG(UNPROTECT & LVPDIS & BOREN & MCLREN & PWRTDIS & WDTDIS & INTIO );

main(void)
{
    TRISB0=1;          //set port RB0 as input port (switch)
    TRISB1=0;          //set port RB1 as output port (LED)
    GIE=1;             //Enable global interrupt to use any interrupt source
    INTE=1;            //Enable external interrupt.
    while(1)
    {
    }
}
static void interrupt isr(void)
{
    if(INTF==1)                //Check, is interrupt from port RB0/INT
    {
        if(RB1==1)RB1=0;        //If the light is on, turn it off
        else RB1=1;             //If not, turn it on
        INTF=0;                 //Clear INT interrupt flag
    }
}
```

### 3) การทดสอบการทำงาน

การทดสอบการทำงานของวงจรเหมือนกับ Project 1 ทุกประการ กล่าวคือเมื่อกดสวิตช์หนึ่งครั้ง ไฟจะสว่างขึ้น และเมื่อกดสวิตช์อีกครั้งหนึ่งไฟจะดับเช่นเดิม แม้ว่าเราจะไม่สามารถสังเกตความแตกต่างระหว่างการทำงานของโปรเจกต์ทั้งสองได้ด้วยตา แต่ขอให้ทราบว่า การใช้ interrupt เป็นวิธีการที่ดีกว่า ทำให้เรามีหน่วยประมวลผลที่ว่างที่จะไปทำงานอย่างอื่นได้ ถ้าเราสังเกตจากในโปรแกรมจะเห็นว่าในวงเล็บของลูป while(1) ไม่มีคำสั่งอะไรเลย แสดงว่าโปรแกรมจะรอโดยไม่ทำอะไรเลยแม้ยังไม่มีการกดสวิตช์ ดังนั้นหากกำหนดให้มิงานอื่นต้องทำในวงเล็บนี้ก็จะสามารถทำได้เต็มที่



รูปที่ 2 การต่อวงจรบน breadboard สำหรับ Project 3