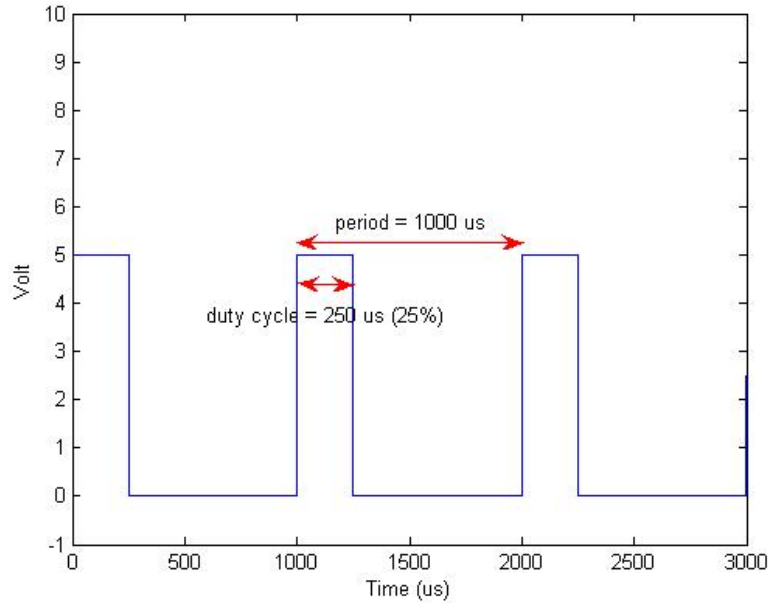


Project 2 Power Width Modulator (PWM)

โจทย์

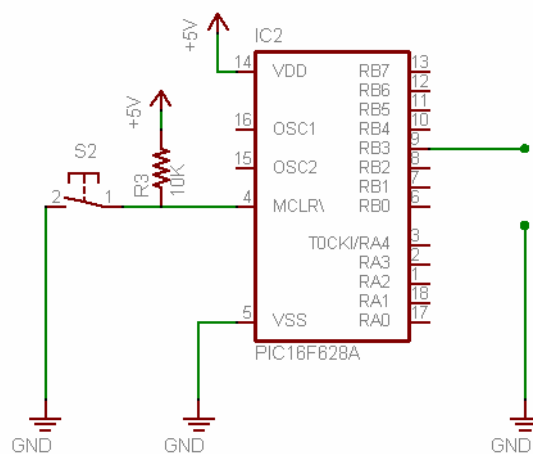
สร้าง PWM ความถี่ = 1 KHz (Period = 1 ms) โดยมี duty cycle = 25% (duty cycle = 250 us) โดยให้ใช้ PWM module (hardware) ของ PIC16F628A ดังรูปที่ 1



รูปที่ 1 รูปกราฟ PWM ที่ต้องการจะสร้างในโปรเจกต์นี้

ขั้นตอนการปฏิบัติ

1) การออกแบบวงจร



รูปที่ 2 แผนผังวงจรที่ใช้สำหรับสร้าง PWM ด้วย PIC

รูปที่ 2 แสดงแผนผังวงจรที่เราจะใช้ในโปรเจกต์นี้ เราจะใช้ขา RB3/CCP1 (ขา 9) ซึ่งทำหน้าที่ได้สองอย่างคือเป็น I/O พอร์ต หรือเป็น Capture/Compare/PWM (CCP) ก็ได้ ซึ่งเราจะต้องกำหนดค่า register เพื่อบอกให้ฮาร์ดแวร์รู้ว่าเราต้องการให้ขา

นี้ทำหน้าที่เป็น CCP ซึ่งหมายความว่าเราจะไม่สามารถใช้ RB3 เป็น I/O พอร์ตได้ในเวลาเดียวกัน นอกจากนี้เราได้ต่อสวิตช์เพื่อใช้สำหรับการ reset ไมโครคอนโทรลเลอร์ไว้ด้วย

2) ขั้นตอนการเขียนโปรแกรม

เตรียมตัวก่อนเขียนโปรแกรม

ในโปรเจกต์นี้เราจะต้องใช้ SFR register มากกว่าในโปรเจกต์แรก เพื่อสั่งงานให้ PWM module สร้างสัญญาณ PWM ตามที่เราต้องการได้ การใช้งานฮาร์ดแวร์โมดูลที่มีหน้าที่เฉพาะเช่น PWM นี้ จะช่วยประหยัดเวลาการทำงานของหน่วยประมวลผล ทำให้หน่วยประมวลผลว่าง และพร้อมที่จะทำงานอื่นได้ทันที PIC แต่ละเบอร์ก็จะมี module ที่ทำหน้าที่เฉพาะแตกต่างกันไป ซึ่งเป็นสิ่งหนึ่งที่เราต้องใช้ในการพิจารณาเลือกเบอร์ PIC ให้เหมาะกับงานของเรา

ดู data sheet ในหัวข้อ 11.0 Capture/Compare/PWM (CCP) Module หน้า 61 และ 64-65 และในหัวข้อ 8.0 Timer2 Module หน้า 50-51 เพื่อทำความเข้าใจวิธีการใช้งาน PWM module ซึ่งสามารถสรุปได้ดังนี้

1. ตั้งค่า PWM period โดยกำหนดค่าที่ register PR2 ซึ่งหาได้โดยคำนวณจากสูตร

$$\text{PWM period} = [(PR2)+1] \times 4 \times T_{\text{osc}} \times (\text{TMR2 prescale value})$$

จากโจทย์เราต้องการให้

$$\text{PWM period} = 1 \text{ ms}$$

$$T_{\text{osc}} = 1/f = 0.25 \text{ us} \text{ (เนื่องจากใช้ internal oscillator ดูความถี่ได้จากหัวข้อ 14.2.6 หน้า 95)}$$

$$\text{TMR2 prescale value} = 4 \text{ (เลือกใช้ค่า 1,4 หรือ 16 ที่จะทำได้ PR2 ไม่เกิน 256)}$$

แทนค่าจะได้

$$PR2 = \text{PWM period} / [4 \times T_{\text{osc}} \times (\text{TMR2 prescale value})] - 1$$

$$PR2 = 1 \text{ ms} / [4 \times 0.25 \text{ us} \times 4] - 1$$

$$PR2 = 254$$

2. ตั้งค่า PWM duty cycle โดยกำหนดค่าที่ register CCPR1L และ CCP1CON<5:4> (8 + 2 = 10 บิต)

ค่า CCPR1L และ CCP1CON<5:4> หาได้โดยคำนวณจากสูตร

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \times T_{\text{osc}} \times (\text{TMR2 prescale value})$$

จากโจทย์เราต้องการให้

$$\text{PWM duty cycle} = 250 \text{ us}$$

$$T_{\text{osc}} = 1/f = 0.25 \text{ us} \text{ (เนื่องจากใช้ internal oscillator ดูความถี่ได้จากหัวข้อ 14.2.6 หน้า 95)}$$

$$\text{TMR2 prescale value} = 4 \text{ (ใช้ค่าเท่ากับค่าที่ใช้ในการคำนวณ PWM period)}$$

แทนค่าจะได้

$$\text{CCPR1L:CCP1CON<5:4>} = (\text{PWM duty cycle}) / [T_{\text{osc}} \times (\text{TMR2 prescale value})]$$

$$\text{CCPR1L:CCP1CON<5:4>} = 250 \text{ us} / (0.25 \text{ us} \times 4)$$

$$\text{CCPR1L:CCP1CON<5:4>} = 250 \text{ (เลขฐานสิบ)}$$

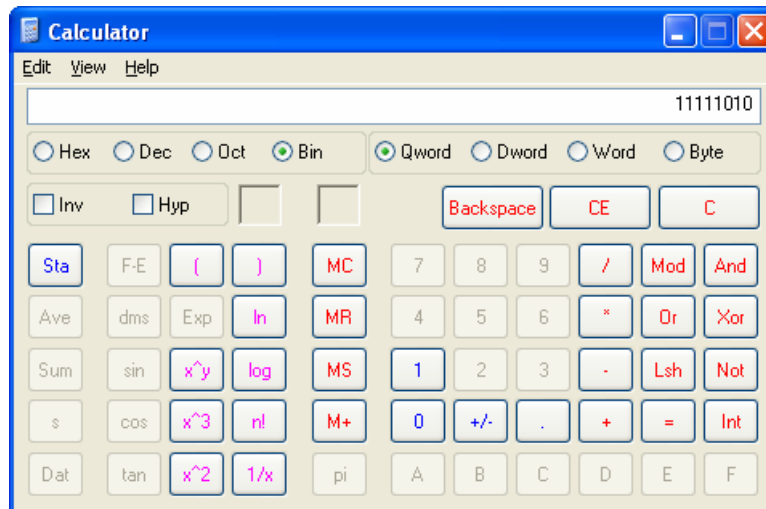
$$\text{CCPR1L:CCP1CON<5:4>} = 0011111010 \text{ (เลขฐานสอง)}$$

ดังนั้นจะได้ว่า

$$\text{CCPR1L} = 00111110 \text{ (8 บิตทางซ้าย)}$$

CCP1CON<5:4> = 10 (2 บิตทางขวา)

หมายเหตุ เราสามารถที่จะแปลงเลขฐาน 2, 10, 16 ได้สะดวกโดยใช้โปรแกรม Calculator ที่มาพร้อมกับ Window



3. กำหนดค่า SFR register ที่จำเป็นอื่นๆ
 - กำหนดให้ขา RB3/CCP1 เป็นขาเอาต์พุตของสัญญาณ โดยกำหนดให้ TRISB3 = 0
 - กำหนดให้ขา CCP เป็น PWM mode โดยกำหนดที่ CCP1CON (CCP1M3:CCP1M0 = 11xx โดยที่ x คือค่า 0 หรือ 1 ก็ได้)
 - กำหนดให้ TMR2 prescale value (ดู data sheet ในหัวข้อที่ 8.0 Timer2 Module หน้า 50-51) ให้เท่ากับค่าที่คำนวณไว้ (= 4) โดยกำหนดที่กลุ่ม register T2CON (T2CKPS1:T2CKPS0 = 01)
4. Enable Timer2 เพื่อเริ่มทำงาน โดยกำหนดให้ TMR2ON = 1

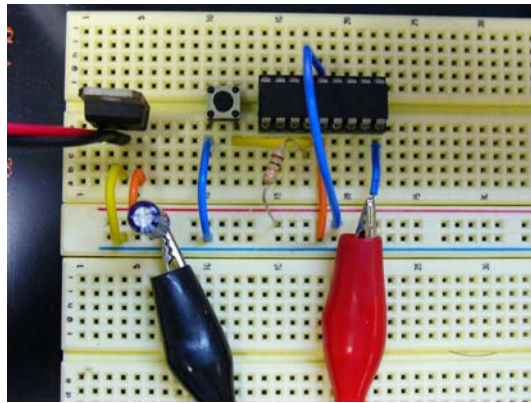
โปรแกรม project02.c

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

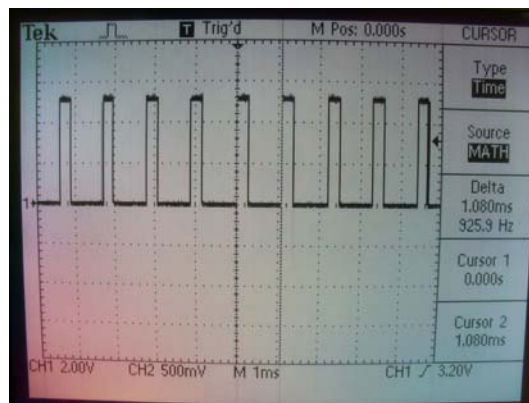
main(void)
{
    PR2=249;           //set register for PWM period
    CCPR1L=0x3E;      //set register for PWM duty cycle bit 2-9
    CCP1X=1;          //                ~                bit 1
    CCP1Y=0;          //                ~                bit 0
    TRISB3=0;         //set pin NO. 9 as output
    CCP1M3=1;         //set CCP1CON as PWM mode bit 3
    CCP1M2=1;         //                ~                bit 2
    T2CKPS1=0;        //set Timer2 prescale value at T2CON bit 1
    T2CKPS0=1;        //                ~                bit 0
    TMR2ON=1;         //start pulse
}
```

3) การทดสอบการทำงาน

ต้องวงจรตามแผนผังวงจรข้างต้นดังแสดงในรูปที่ 1 จ่ายไฟให้กับ PIC แล้วให้ใช้ oscilloscope วัดสัญญาณที่ขา RB3/CCP1 จะได้สัญญาณ PWM ตามที่โจทย์กำหนด ดังแสดงในรูปที่ 4



รูปที่ 3 การต่อวงจรบน breadboard เพื่อสร้าง PWM



รูปที่ 4 สัญญาณ PWM ที่ได้จาก PIC