

## Class 5 PWM - Class Program

```
/*
 * File:  main.c
 * Author:  sagar
 *
 * Created on 1 April, 2025, 3:34 PM
 */

// CONFIG
#pragma config FOSC = HS      // Oscillator Selection bits (HS 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)

#include <xc.h>
#include<stdint.h>

#define _XTAL_FREQ 2000000
void pwmInit(void);
void pwmDutyCycleUpdate(void);
uint8_t pwm10_lower, pwm10_upper;
uint8_t pwm50_lower, pwm50_upper;
uint8_t pwm80_lower, pwm80_upper;

#include <xc.h>
#include<stdint.h>

#define _XTAL_FREQ 2000000 // Define oscillator frequency for delay
void pwmInit(void);
void pwmDutyCycleUpdate(void);

void main(void)
{
    pwmInit();
    while (1)
    {
        pwmDutyCycleUpdate();
    }
}
```

```

void pwmInit(void)
{
    TRISC = 0xFB; // Make RC2 (CCP1) an output
    CCP1CON = 0x0C; // PWM mode, 10-bit
    T2CON = 0x06; // Timer2 ON, prescaler 1:16
    PR2 = 0x5E; // Set period
    pwm10_lower = 0x02;
    pwm10_upper = 0x09;

    pwm50_lower = 0x00;
    pwm50_upper = 0x2F;

    pwm80_lower = 0x00;
    pwm80_upper = 0x4B;

    TMR2 = 0; // Reset Timer2 counter
    TMR2ON = 1; // Turn on Timer2
}

void pwmDutyCycleUpdate(void)
{
    /***** 10% duty cycle *****/
    CCPR1L = pwm10_upper; // Upper 8 bits
    CCP1CON &= ~(0x3 << 4);
    CCP1CON |= (pwm10_lower << 4); // Lower 2 bits
    __delay_ms(3000);

    /***** 50% duty cycle *****/
    CCPR1L = pwm50_upper;
    CCP1CON &= ~(0x3 << 4);
    CCP1CON |= (pwm50_lower << 4);
    __delay_ms(3000);

    /***** 80% duty cycle *****/
    CCPR1L = pwm80_upper;
    CCP1CON &= ~(0x3 << 4);
    CCP1CON |= (pwm80_lower << 4);
    __delay_ms(3000);
}

```