

Assignment-10

```
#include <pic18f4550.h>

#define FREQ_PWM 50 // Frequency of PWM signal in Hz (adjust based on motor needs)

void initPWM() {
    // Set RC1 and RC2 pins as output for PWM signals
    TRISCbits.TRISC1 = 0; // RC1 pin as output (CCP1)
    TRISCbits.TRISC2 = 0; // RC2 pin as output (CCP2)

    // Configure Timer2 for PWM
    T2CONbits.T2CKPS = 0b11; // Prescaler 16 (adjustable as needed)
    T2CONbits.TMR2ON = 1; // Enable Timer2

    // Set the period for PWM
    PR2 = 150; // Period register for 8-bit resolution (For 50Hz PWM on a 8MHz clock)

    // Configure CCP1 module for PWM mode (RC1 pin)
    CCP1CONbits.CCP1M = 0b1100; // PWM mode for CCP1 (RC1)

    // Configure CCP2 module for PWM mode (RC2 pin)
    CCP2CONbits.CCP2M = 0b1100; // PWM mode for CCP2 (RC2)

    // Enable the PWM outputs on CCP1 (RC1) and CCP2 (RC2)
    TRISCbits.TRISC1 = 0; // Set RC1 pin as output for CCP1 PWM
    TRISCbits.TRISC2 = 0; // Set RC2 pin as output for CCP2 PWM

    // Initialize duty cycle to 50% (128 of 255) for both motors
    CCPR1L = 128; // Initial duty cycle for motor 1 (50%)
    CCPR2L = 128; // Initial duty cycle for motor 2 (50%)

    // Enable Global Interrupts (if needed for other interrupt functionalities)
    INTCONbits.GIE = 1;
}

void setPWMDutyCycle(unsigned char motor, unsigned char dutyCycle) {
    // Set the duty cycle of the specified motor (1 for motor 1, 2 for motor 2)
    if (motor == 1) {
        CCPR1L = dutyCycle; // Set duty cycle for motor 1 (CCP1, RC1)
    } else if (motor == 2) {
        CCPR2L = dutyCycle; // Set duty cycle for motor 2 (CCP2, RC2)
    }
}

void delay_ms(unsigned int ms) {
    unsigned int i, j;
    for(i = 0; i < ms; i++) {
        for(j = 0; j < 1000; j++);
    }
}
```

```

}

void main() {
    // Initialize PWM for motors
    initPWM();

    // Example: Set PWM duty cycle to 128 (50%) for both DC motors
    setPWMDutyCycle(1, 128); // 50% duty cycle for motor 1
    setPWMDutyCycle(2, 128); // 50% duty cycle for motor 2

    while(1) {
        // Change duty cycle for motor 1 and motor 2 (adjust for speed control)

        // Example: Gradually increase duty cycle for motor 1 (motor 1 speed control)
        for (unsigned char duty = 0; duty < 255; duty++) {
            setPWMDutyCycle(1, duty); // Gradually increase motor 1 speed
            delay_ms(50);              // Wait for 50ms to observe effect
        }

        // Example: Gradually decrease duty cycle for motor 1 (motor 1 speed control)
        for (unsigned char duty = 255; duty > 0; duty--) {
            setPWMDutyCycle(1, duty); // Gradually decrease motor 1 speed
            delay_ms(50);              // Wait for 50ms to observe effect
        }

        // Example: Gradually increase duty cycle for motor 2 (motor 2 speed control)
        for (unsigned char duty = 0; duty < 255; duty++) {
            setPWMDutyCycle(2, duty); // Gradually increase motor 2 speed
            delay_ms(50);              // Wait for 50ms to observe effect
        }

        // Example: Gradually decrease duty cycle for motor 2 (motor 2 speed control)
        for (unsigned char duty = 255; duty > 0; duty--) {
            setPWMDutyCycle(2, duty); // Gradually decrease motor 2 speed
            delay_ms(50);              // Wait for 50ms to observe effect
        }
    }
}

```

FB7	ECCP1DEL	0x00	0	00000000	'.'
FB8	BAUDCON	0x40	64	01000000	'@'
FBA	CCP2CON	0x0C	12	00001100	'.'
FBB	CCPR2	0x8080	32896	10000000 10000000	'□□'
FBB	CCPR2L	0x80	128	10000000	'□'
FBC	CCPR2H	0x80	128	10000000	'□'
FBD	CCP1CON	0x0C	12	00001100	'.'
FBE	CCPR1	0x0080	128	00000000 10000000	'.'□'
FBE	CCPR1L	0x80	128	10000000	'□'
FBF	CCPR1H	0x00	0	00000000	'.'
FC0	ADCON2	0x00	0	00000000	'.'

FC9	SSPBUF	0x00	0	00000000	'.'
FCA	T2CON	0x07	7	00000111	'.'
FCB	PR2	0x96	150	10010110	'□'
FCC	TMR2	0x07	7	00000111	'.'
FCD	T1CON	0x00	0	00000000	'.'
FCE	TMR1	0x0000	0	00000000 00000000	'..'
FCE	TMR1L	0x00	0	00000000	'.'
FCF	TMR1H	0x00	0	00000000	'.'
FD0	RCON	0x5C	92	01011100	'\'
FD1	WDTCN	0x00	0	00000000	'.'
FD2	HLVDCN	0x05	5	00000101	'.'
FD3	OSCCN	0x48	72	01001000	'H'
FD5	T0CON	0xFF	255	11111111	'y'
FD6	TMR0	0x0000	0	00000000 00000000	'..'
FD6	TMR0L	0x00	0	00000000	'.'
FD7	TMR0H	0x00	0	00000000	'.'
FD8	STATUS	0x02	2	00000010	'.'
FD9	FSR2	0x0000	0	00000000 00000000	'..'
FD9	FSR2L	0x00	0	00000000	'.'
FDA	FSR2H	0x00	0	00000000	'.'
FDB	PLUSW2	0x00	0	00000000	'.'

Variables						
Name	Type	Address	Value	Hexadecimal	Decimal	
<Enter new watch>						
dutyCycle	unsigned char	0x1	'?'; 0x80	0x80	128	
motor	unsigned char	WREG0 (CPU)	NUL; 0x0	0x00	0	

Variables						
Name	Type	Address	Value	Hexadecimal	Decimal	
<Enter new watch>						
i	unsigned int	0x3	0x0000	0x0000	0	
j	unsigned int	0x5	0x015E	0x015E	350	
ms	unsigned int	0x1	0x0032	0x0032	50	