



```

// PIC16F877A Configuration Bit Settings

// 'C' source line configuration statements

// CONFIG
#pragma config FOSC = EXTRC      // Oscillator Selection bits (RC 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
#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
#pragma config CP = OFF          // Flash Program Memory Code Protection bit (Code protection off)

// #pragma configuration statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>

//defining clock frequency for the PIC controller
#define _XTAL_FREQ 6000000

//function declarations
void init(void);
void pwmchange(void);

//variable declaration
unsigned char pwmhigh10, pwmlow10;
unsigned char pwmhigh50, pwmlow50;
unsigned char pwmhigh80, pwmlow80;

void main(void) {
    init();
    while (1) {
        pwmchange(); // calling function PWMchange for changing percentage of duty cycle
    }
}

void init(void) {
    //PIC controller has 2 PWM modules, we are using PWM1 which is connected to RC2 pin
    //thus RC2 is set as output, other pins are configured input
    TRISC = 0xFB; //1111 1011 RC2 is set as output
    //3rd and 2nd bit as to be HIGH for enabling PWM mode
    //4th and 5th bit are dedicated for storing LSB data of CCP1 register (duty cycle)
    CCP1CON = 0x3C; //00111100
    //CCP1 register is for setting duty cycle for PWM,
    //Calculations has to be done as follows
    //CCP1 = (duty cycle * FOSC(controller clock))/Timer pre-scale value
    CCP1L = 0x2E; //0010 1110 - MSB of duty cycle
    //PIC controller timer module T2CON is used as timing reference for PWM
    //2nd bit has to be HIGH for enabling timer and bit 1,0 has to be set according to the required prescale value
    //bit 1 is set HIGH for 1:16 pre-scaler

```

```

T2CON = 0x06;
//PR2 register is for setting the total duration of one period(PWM)
//calculation: (FOSC/(4*pwm_freq*TIMER pre-scale value))-1
// -1 is optional could be used or unused according to the practical results
//PR2 = (6000000/(4*1000*16)) = 94(0x5E)
PR2 = 0x5E;
//duty cycle calculation for 10 per:
//Duty cycle time = 0.1/pwm_freq = 0.1/1000
//0.0001*6000000/16 = 38(0x26)-0010 0110
//LSB 2 bit has to be set in CCP1CON(4,5th bit)= 0010 0000(0x20)
//other MSB in CCP1L = 0000 1001(0x09)
pwmhigh10=0x09;
pwmlow10=0x20;
//duty cycle calculation for 50 per:
//Duty cycle time = 0.5/pwm_freq = 0.5/1000
//0.0005*6000000/16 = 188(0xBC)-1011 1100
//LSB 2 bit has to be set in CCP1CON(4,5th bit)= 0000 0000
//other MSB in CCP1L = 0010 1111(0x2F)
pwmhigh50=0x2F;
pwmlow50=0x00;
//duty cycle calculation for 80 per:
//Duty cycle time = 0.8/pwm_freq = 0.8/1000
//0.0008*6000000/16 = 300(0x12C)- 0001 0010 1100
//LSB 2 bit has to be set in CCP1CON(4,5th bit)= 0000 0000
//other MSB in CCP1L = 0100 1011(0x4B)
pwmhigh80=0x4B;
pwmlow80=0x00;

```

```

void pwmchange()

```

```

{
    //assigning MSB 8 bit of 10 percent duty cycle value to CCP1L
    CCP1L = pwmhigh10;
    //resetting the 4th and 5th bit of CCP1CON and setting only 4th and 5th bit(mask value is same as pwmlow10)
    CCP1CON= (CCP1CON&0xCF)|pwmlow10;
    __delay_ms(3000);
    //assigning MSB 8 bit of 50 percent duty cycle value to CCP1L
    CCP1L = pwmhigh50;
    //resetting the 4th and 5th bit of CCP1CON and setting only 4th and 5th bit(mask value is same as pwmlow50)
    CCP1CON= (CCP1CON&0xCF)|pwmlow50;
    __delay_ms(3000);
    //assigning MSB 8 bit of 80 percent duty cycle value to CCP1L
    CCP1L = pwmhigh80;
    //resetting the 4th and 5th bit of CCP1CON and setting only 4th and 5th bit(mask value is same as pwmlow80)
    CCP1CON= (CCP1CON&0xCF)|pwmlow80;
    __delay_ms(3000);
}

```