



Konu : Pot. ile LED Parlaklığı Kontrolü

Örnek Kod

```
#include<16f876a.h> // Ön işlemci fonksiyon ayarları yapılır
#device ADC = 10 // Pic Wizard ile otomatik olarak da ayarlanabilir.
//#fuses HS // ADC, 10 bit olacak şekilde ayarlanır.
#fuses XT // High Speed - Yüksek Hızlı Kristal veya Seramik Resonatör kullanılacak.
#fuses NOMDT // XTal - Kristal veya Seramik Resonatör
#fuses NOLVP // No Watch Dog Timer - WDT kullanılmayacak.
#fuses NOPROTECT // No Low Voltage Programming - LVP kullanılmayacak.
// Program Belleğine (ROM) yazılan verilerin okunmasına izin verir.

#USE delay(clock = 4000000) // PIC kütüphanelerinden unutulmuş kodları bulmak için;
// #include kısmındaki tanımlanmış .h uzantılı yere sağ tıklayıp
// "Open File at cursor" kısmından kütüphaneye girilebilir.
// Bu kütüphaneden istenen kelime "Ctrl+F" ile rahatça bulunabilir.

int analog_veri_oku ( int ); // Kullanılacak fonksiyon önceden tanımlanır.
int LED_Parlaklik = 0; // Ana ( Main ) fonksiyondan önce tanımlanması gerekmektedir.
// Global bir değişken tanımlanır.

void main(){
    setup_adc(ADC_CLOCK_INTERNAL); // ADC aktif hale getirilir.
    setup_adc_ports(AN0); // A0 pini analog giriş(input) olarak ayarlanır.

    setup_ccp1(CCP_PWM); // CC1 bacağı PWM olarak kullanılacağı tanımlanır.
    setup_ccp2(CCP_OFF); // CC2 bacağı pasif hale getirilir.
    setup_timer_2(T2_DIV_BY_16,255,1); // Timer2 ayarları yapılır.
    // Kristal sinyali 16'ya bölünür.
    // Timer-2 sayıcısının sınır değeri 255 olarak belirlenir.

    set_tris_a(0x0f); // A0-3 pinleri Giriş(Input), A4-6 pinleri ise Çıkış(Output) olarak ayarlanır.
    set_tris_b(0x00); // B portu Çıkış(Output) olarak ayarlanır.
    set_tris_c(0x00); // C portu Çıkış(Output) olarak ayarlanır.

    output_b(0x00); // Portlar sıfırlanır.
    output_c(0x00);

    while(1){
        LED_Parlaklik = analog_veri_oku(0); // Fonksiyondan döndürülen değer "LED_Parlaklik" değişkenine eşitlenir.
        set_pwm1_duty( LED_Parlaklik ); // "LED_Parlaklik" değişken değeri oranında Duty-Cycle oluşturan PWM Sinyali oluşturulur.
        delay_ms(50); // 50 milisaniye gecikme oluşturulur.
    }

    int analog_veri_oku( int Kanal ){
        set_adc_channel(Kanal); // "Kanal" değerindeki analog bacadan( Ör: AN0 pini ) veri okunur.
        delay_us(10); // 100 mikrosaniye gecikme oluşturulur.
        return (read_adc()/4 ) ; // ADC'den 10 bitlik değer okunur.
        // PWM fonksiyonu 8 bitlik değer döndürür.
        // Bu yüzden ADC'de okunan değer 4'e bölünür.
    }
}
```

Proteus Çizimi

