



PIC Mikrodenetleyicilere Giriş

ANTALYA SINAV ANADOLU LİSESİ ROBOTİK TOPLULUĞU



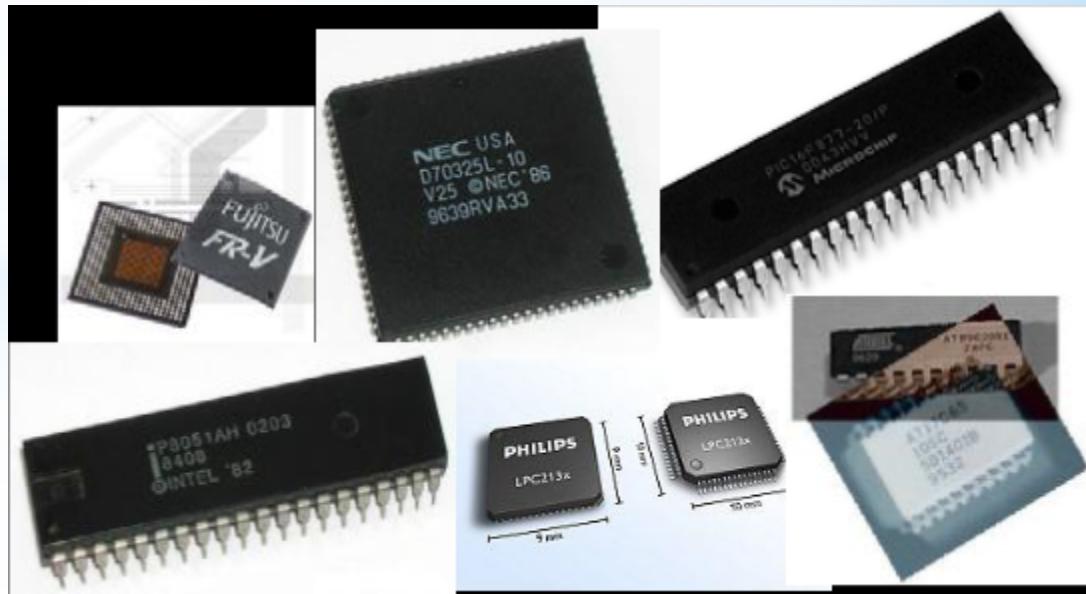
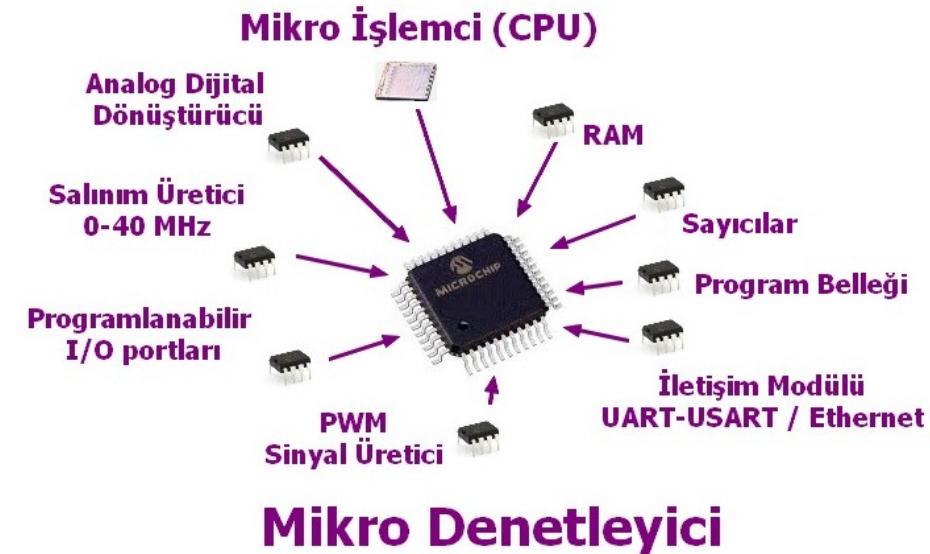
SINAV KOLEJİ

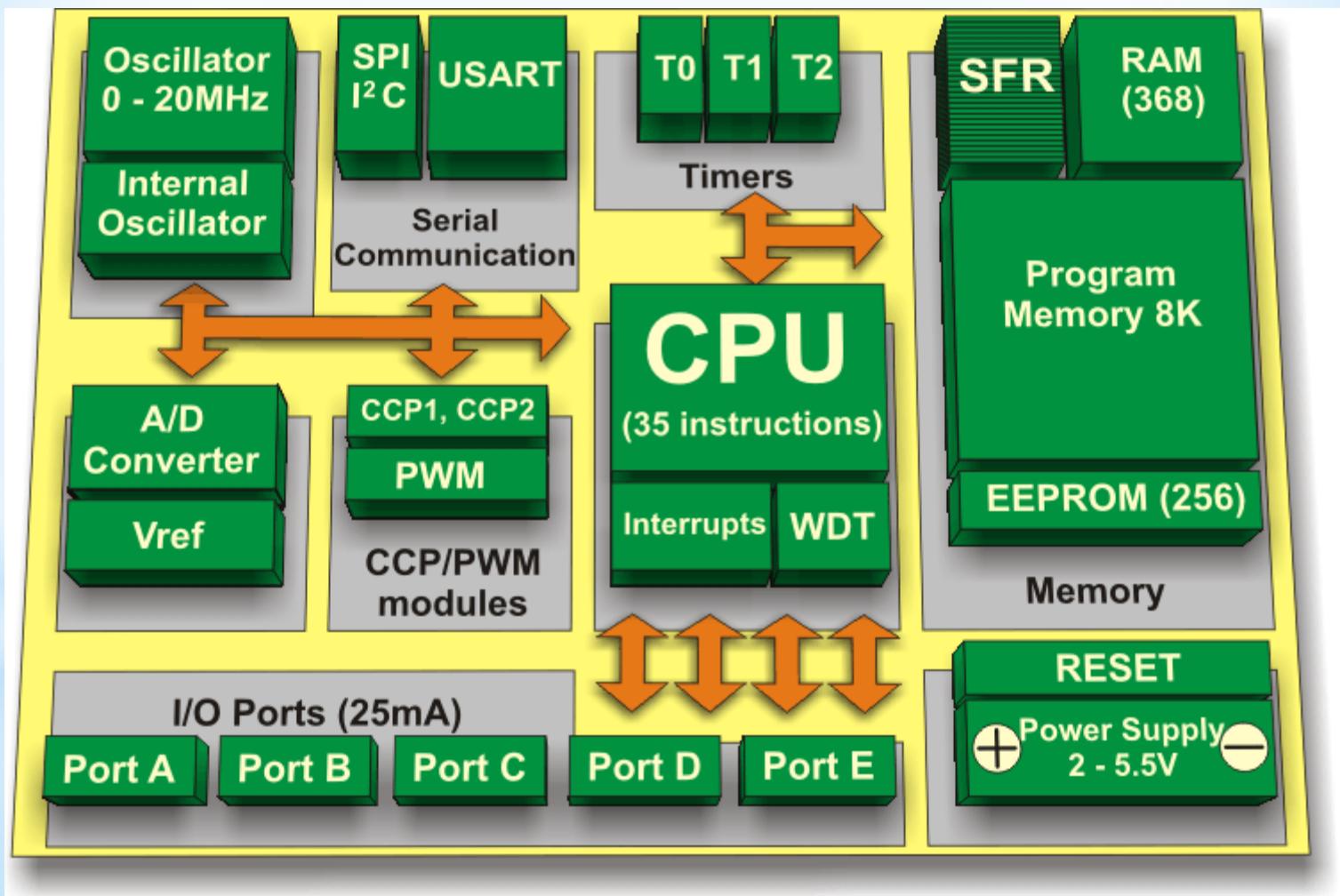


Mikrodenetleyiciler

* Elektronik ve eletromekanik sistemleri yönetmek amacıyla bir merkezi işlem ünitesi (CPU) çevresinde yapılandırılmış bellek (EPROM, EEPROM ve Flash...), programlanabilir giriş ve çıkışlar (input-output), analog-dijital dönüştürücü, sinyal üretici, sayıcı, iletişim arabirimi, kristal osilatör üretici gibi çevre birimlerinin tümleşik bir biçimde yani tek bir yonga şeklinde üretiltiği bir mikro bilgisayardır.

* Mikroişlemciler daha çok bilgisayar sistemleri için uygun bir üretilir. Aşırı yüksek hız gerektirmeyen ve fazla belleğe ihtiyaç duymayan bir kontrol sistemi için mikroişlemci kullanmak hem çok pahalı hem de programlaması zor bir iştir.



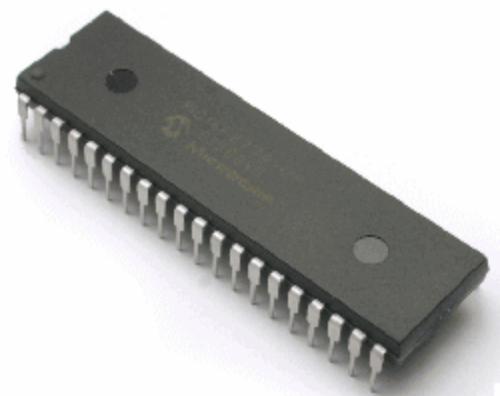
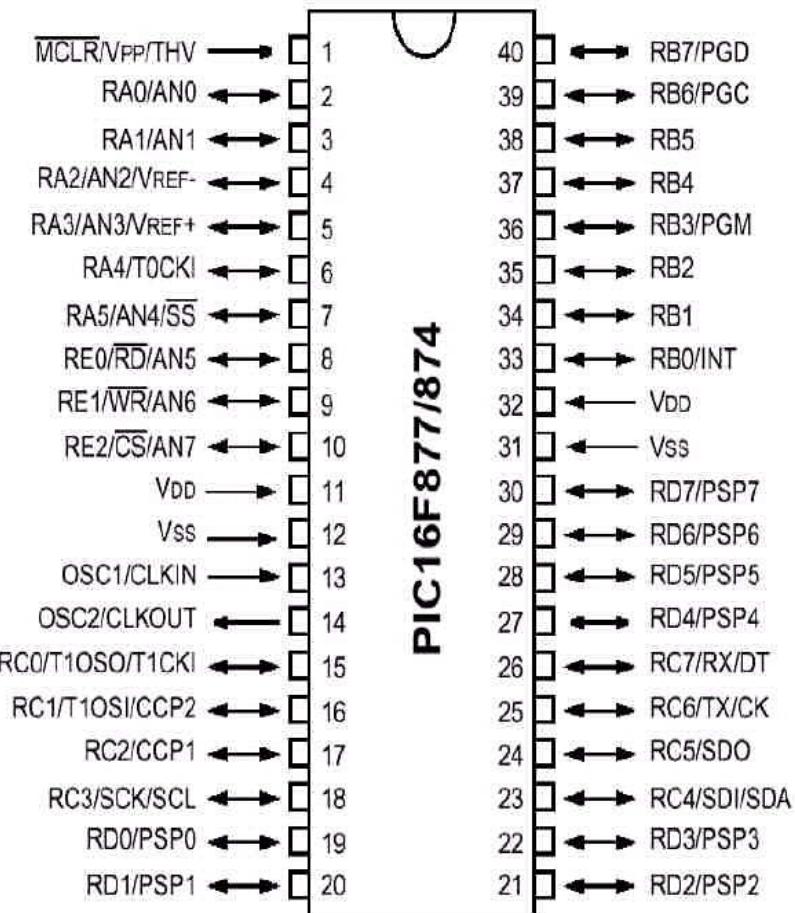


16F877 PIC Mikrodenetleyicisi

- * PIC ismi (Peripheral Interface Controller), Microchip firmasının ürettiği mikrodenetleyicilere verdiği isimdir. PIC ürünlerinde komut kelime boyu ile veri yolu farklı uzunlukta olabilir. Üretilen PIC mikrodenetleyiciler, denetleyicinin veri yolu bit sayısı ve kelime boyuna göre değişik aile isimlerine göre sınıflandırılırlar. (PIC 12F675, PIC16F877, PIC16F628A, PIC18F4520...)
- * PIC16F877 mikrodenetleyicisi PIC ailesinin güçlü bir ürünüdür. Bu ürünün genel özelliklerine bakacak olursak çoğu uyuglama için yeterli bir donanıma sahip olduğunu görürüz.

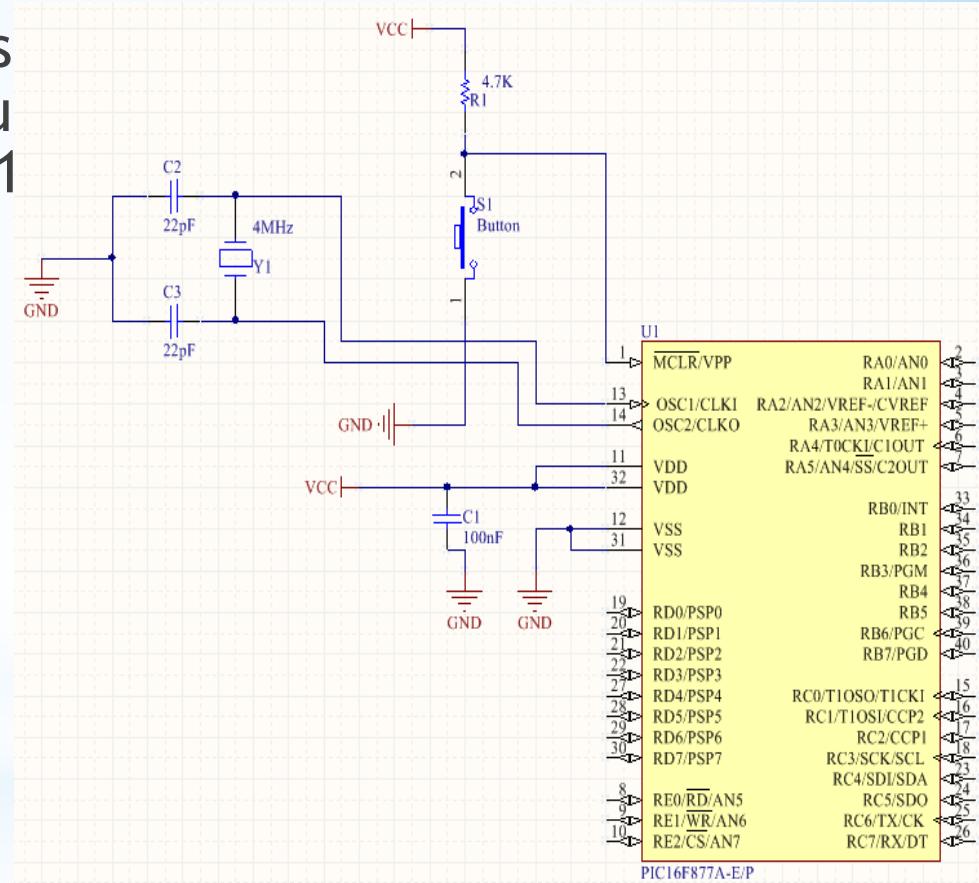
Çizelge 1. PIC16F877 ile PIC16F84'ün karşılaştırılması

| ÖZELLİKLER | PIC16F877 | PIC16F84 |
|---------------------------|---|----------------------|
| Çalışma hızı | DC-20Mhz | DC-10Mhz |
| Program belleği | 8K×14 word Flash ROM | 1K×14 word Flash ROM |
| EEPROM Veri belleği | 256 byte | 64 byte |
| Kullanıcı RAM | 368×8 byte | 68×8 byte |
| Giriş / Çıkış port sayısı | 33 | 13 |
| Timer | Timer0, Timer1, Timer2 | Timer0 |
| A/D çevirici | 8 kanal 10 bit | YOK |
| Capture / Comp. / PWM | 16 bit Capture 16 bit Compare 10 bit PWM çözünürlük | YOK |
| Seri çevresel arayüz | SPI(Master) ve 12C(Master/Slave) modunda SPI portu (senkron seri port) | YOK |
| Paralel slave port | 8 bit, harici RD,WR ve CS kontrollü | YOK |
| USART/SCI | 9 bit adresli | YOK |



Besleme Uçları ve Bağlantıları

* PIC16F877 için besleme uçları Vs (Negatif besleme) 12 ve 31 no'lu pinlere; Vdd (pozitif besleme) 11 ve 32 no'lu pinlere uygulanır. Besleme gerilimleri 2 - 5.5V arasında olabilir. Genelde laboratuvar çalışmalarında 5V idealdir. Devreye enerji verildiği anda meydana gelebilecek gerilim dalgalanmalarından dolayı isteneeyen arızaları önlemek amacıyla Vdd ile Vss arasında kondansatör (100nF) bağlanır.



Osilatör Konfigürasyonları

- * * PIC, hafızasındaki komutları işlemek için bir sinyale ihtiyaç duyar. Bu sinyale clock sinyali denir.
- * PIC16F877'nin osilatör uçları 13(OSC1) ve 14(OSC2) pinleridir.
- * PIC mikrodenetleyicisi, girişine bağlanan osilatör frekansını 4'e bölgerek komut işlemek için kullanır. Bu şekilde bir komutun işlenmesi için gereken zaman:

$$f_{komut} = \frac{f_{osc}}{4}, \quad T_{komut} = \frac{1}{f_{komut}}$$

*

$$f_{komut} = \frac{f_{osc}}{4} = \frac{4 \text{ MHz}}{4} = 1 \text{ MHz} = 1 \times 10^6 \text{ Hz}$$

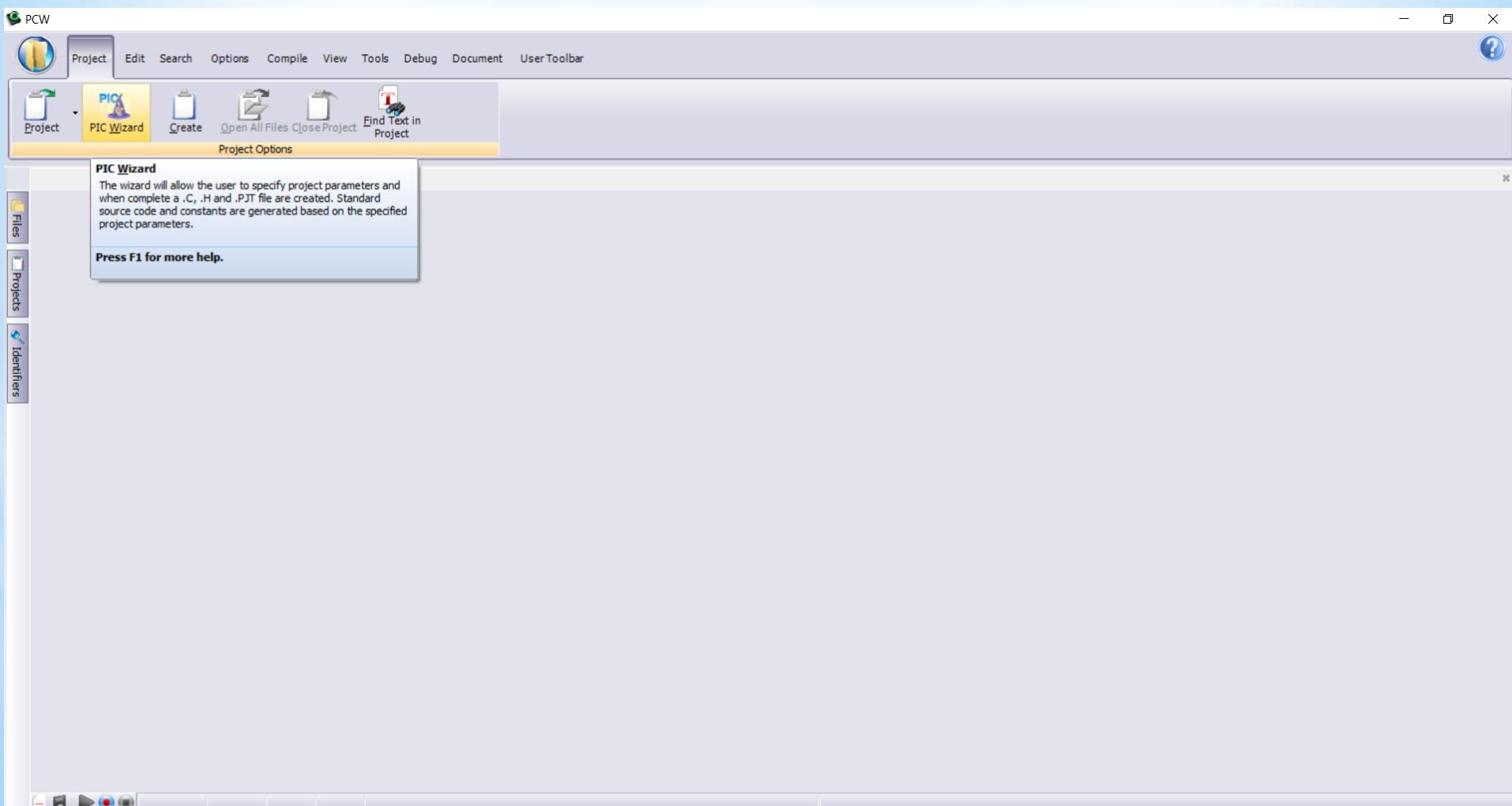
$$T_{komut} = \frac{1}{f_{komut}} = \frac{1}{1 \times 10^6} = 0.000001 \text{ s} = 1 \mu\text{s}$$

PIC denetleycilere bağlanabilen osilatör tipleri şunlardır:

- * LP (Low Power) Kristal Osilatör (32-200 kHz)
- * XT (Crystal) Kristal/Resonatör Osilatör (1-4 MHz)
- * HS (High Speed) Yüksek Hızlı Kristal Osilatör (4-20 MHz)
- * RC (Resistor-Capacitor) Direnç-Kondansatör Osilatör (< 4Mhz)

CCS C ile Program Derlemek

- * İlk olarak proje dosyası oluşturmak için ilk başta «Project» menüsü seçilmeli ve menü altında açılan sekmeden «Project Wizard» sekmesi seçilmelidir.



* Önümüze açılacak pencerede proje dosyasının kaydedileceği yer ve ismi belirtilir. Oluşturulan projenin kaydedileceği yer seçimi ve proje ismi verildikten sonra karşımıza ayar menüsü ve sekmeler karşımıza gelir. «General» sekmesindeki «Device» kısmında kullanılacak PIC modeli, «Oscillator Frequency» kısmında osilatör frekansı, «Fuses» kısmında ise kullanılacak olan PIC başlangıç ayarları yapılır.

The screenshot shows a Microsoft PowerPoint slide titled "sınav kolejİ sunum (rev.2) - Microsoft PowerPoint". The slide contains several numbered notes (7, 8, 9, 10) and a central dialog box for the PIC Wizard.

Notes:

- 7 Osilatör Konfigürasyonları
 - PIC, hafızasındaki komutları işlemek için bir sinyale ihtiyaç duyar. Bu sinyale clock sinyali denir.
 - PIC16F877'inin oscillator uduarı 13(OSC1) ve 14(OSC2) pinesidir.
 - PIC mikrodenetleyici, girişine bağlanan oscillator frekansını 4'e bölerken komut işlemek için kullanır. Bu şekilde bir komutun işlenmesi için gerekilen zaman:
$$t_{\text{komut}} = \frac{f_{\text{osc}}}{4} \quad T_{\text{komut}} = \frac{1}{f_{\text{osc},\mu}}$$
- 8

$$f_{\text{osc}} = \frac{f_{\text{ext}}}{4} = \frac{4 \text{ MHz}}{4} = 1 \text{ MHz} = 1 \times 10^6 \text{ Hz}$$

$$T_{\text{komut}} = \frac{1}{f_{\text{osc}}} = \frac{1}{1 \times 10^6} = 0.000001 \text{ s} = 1 \mu\text{s}$$

PIC denetleyicilerde bulunan oscillator türleri şunlardır:
 LPL (Low Power Kristal oscillator (12-20 kHz))
 XT (Crystal Kristal İstismar oscillator (1-4 MHz))
 HS (High Speed Yüksek Hizi Kristal oscillator (4-20 MHz))
 RC (Resistor-Capacitor Dairesel-Kondansatör oscillator (+400 Hz))
- 9 CCS C ile Program Perlemek
 - İlk etape projenin oluşturulması (önce başta chosen menu'dan new seçilir) ve menü altından açılan seçeneklerden Project选择' seçilmesi gereklidir.
- 10
 - Öncelikle yazılımın çalışacağı portun seçimi (Port选择) yapılıp port numarası (1-4) seçilir. Daha sonra program yüklenme portu (Program选择) seçiliyor ve port numarası (1-4) seçilir.

PIC Wizard Dialog Box:

Project Name: C:\Users\tugdell\Desktop\deneme\deneme_1.pjt

Options Tab:

- General
 - Function Generation
 - Opening brace on the following line
 - Opening brace on the same line
- Device: PIC16C74
- Oscillator Frequency: 20,000,000 Hz
- Enable Integrated Chip Debugging (ICD)
- Restart WDT during calls to DELAY
- Use 16 bit pointers for Full RAM use
- One fuse per line with comments

Fuses Tab:

- Low power osc < 200 khz
- Power Up Timer
- Code protected from reads

Buttons: Ok, Cancel, Help

Text at the bottom: Not eklemek için tıklatın

* «I/O Pins» kısmında PIC ilgili dijital ve analog pinlerinin giriş (input) veya çıkış (output) olarak ayarlandığı kısmıdır. Bizim uylamasını yapacağımız LED yakma söndürme programı için bu kısımda ilgili pinin veya pinlerin giriş ve çıkış portu olup olmayacağıının ayarlanması gerekmektedir.

sinav kolej sunumu (rev.2) - Microsoft PowerPoint

Dosya Giris Ekle Tasarım Geçişler Animasyonlar Slayt Gösterisi Gözden Geçir Görünüm Storyboarding

Yapıtır Kes Kopyala Sıfırla Yeni Slayt Bölüm Pano Slaytlar

19 A A Metin Yönü Metni Hizala SmartArt'a Dönüştür

K T A S abe AA A Verleştir Hızlı Şekil Dolgusu Bul Değiştir Seç

Yazılı Tipi PIC Wizard

File Project Name: C:\Users\tugdell\Desktop\deneme\deneme_1.pjt

General Communications SPI and LCD Timers PIC Timers Analog Other Interrupts Drivers I/O Pins High/Low Voltage Intr Oscillator Config Header Files CAN BUS LCD options MOD BUS BOOT LOADER

Pin names:

| | | |
|----|--------|--------|
| A0 | Input | PIN_A0 |
| A1 | Input | PIN_A1 |
| A2 | Input | PIN_A2 |
| A3 | Input | PIN_A3 |
| A4 | Input | PIN_A4 |
| A5 | Input | PIN_A5 |
| B0 | Output | PIN_B0 |
| B1 | Input | PIN_B1 |
| B2 | Input | PIN_B2 |
| B3 | Input | PIN_B3 |
| B4 | Input | PIN_B4 |
| B5 | Input | PIN_B5 |
| B6 | Input | PIN_B6 |
| B7 | Input | PIN_B7 |
| C0 | Input | PIN_C0 |
| C1 | Input | PIN_C1 |

Enable Pullups on port B List multiple identifiers by separating with commas

Ok Cancel Help

Not eklemek için tıklatin

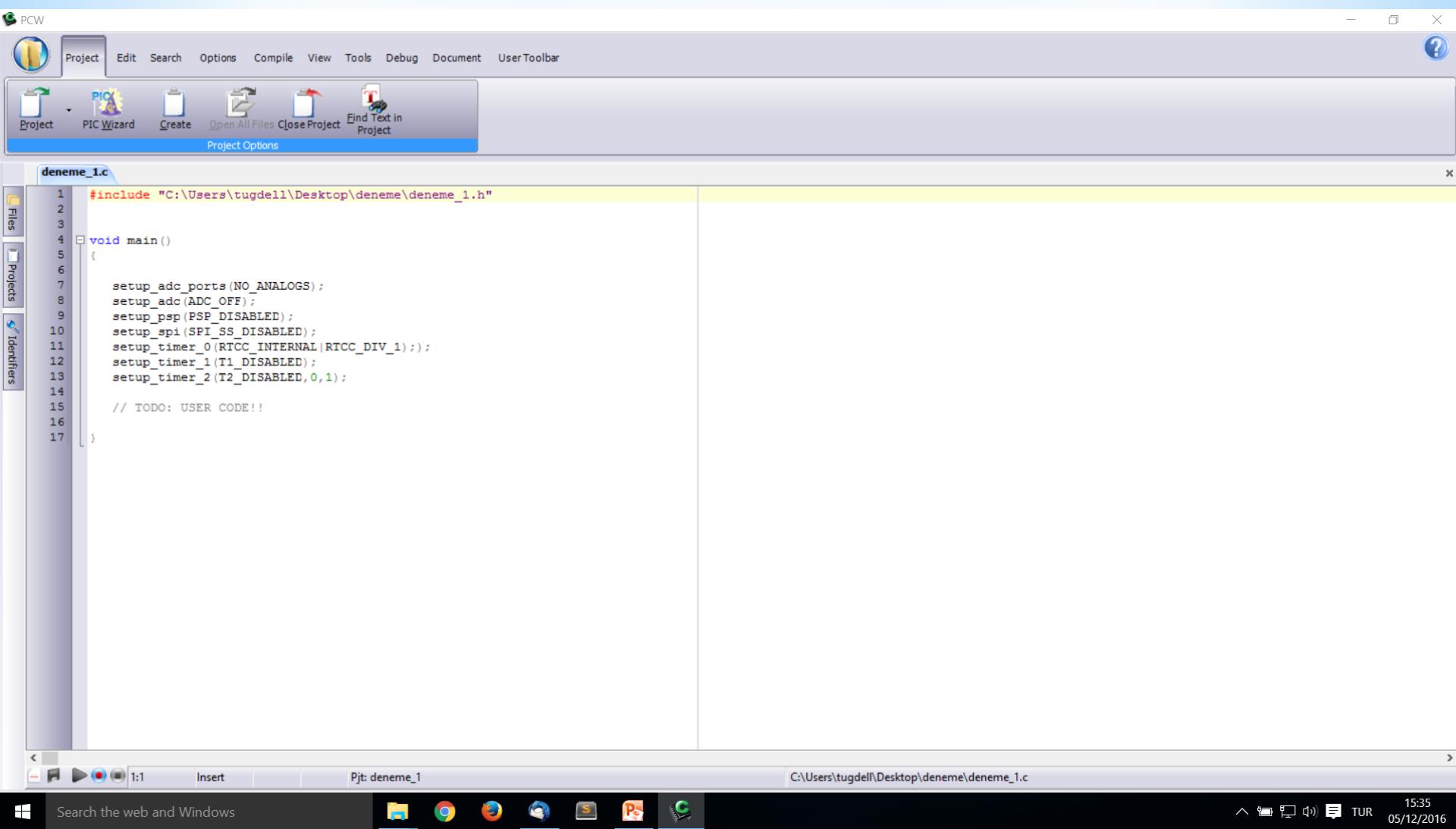
11

çin
tin

Slayt 11 / 11 | "Hava Akımı" | Türkçe | 9683 | 15:30 | 05/12/2016

Search the web and Windows

* İlgili pinlerin seçimi ve ataması yapıldıktan sonra «OK» butonu basılarak oluşturulan proje ile ilgili PIC ayarları tamamlanır ve karşımıza istediğimiz proje ile ilgili kaynak kodları yazacağımız .c uzantılı kaynak dosyası açılır.



The screenshot shows the PCW (PIC C Workbench) software interface. The main window displays a C code editor with the file 'deneme_1.c' open. The code includes setup functions and a main function placeholder. The interface includes a toolbar, menu bar, and various project management tools.

```
1 #include "C:\Users\tugdell\Desktop\deneme\deneme_1.h"
2
3
4 void main()
5 {
6
7     setup_adc_ports(NO_ANALOGS);
8     setup_adc(ADC_OFF);
9     setup_psp(PSP_DISABLED);
10    setup_spi(SPI_SS_DISABLED);
11    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
12    setup_timer_1(T1_DISABLED);
13    setup_timer_2(T2_DISABLED,0,1);
14
15 // TODO: USER CODE!!
16
17 }
```

PCW

Project Edit Search Options Compile View Tools Debug Document User Toolbar

Project PIC Wizard Create Open All Files Close Project Find Text in Project

Project Options

deneme_1.c

Files Projects Identifiers

Pjt: deneme_1 C:\Users\tugdell\Desktop\deneme\deneme_1.c

1:1 Insert

15:35 TUR 05/12/2016

Search the web and Windows

Uygulama 1:LED Yakma-Söndürme

```
*****
```

Uygulama Adı: PIC16F877 ile Led Yakma-Sondurme Programı
Programın Amacı: RB0 pini ile LED'i belirli aralıklarla
yakıp söndürmek

```
***** /
```

// Kullanılacak PIC'in baslık dosyası tanıtılmıyor
`#include <16f877.h>`

// PIC konfigurasyon ayarları:

```
#fuses  
XT,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD
```

// Osilator frekansı seçiliyor:

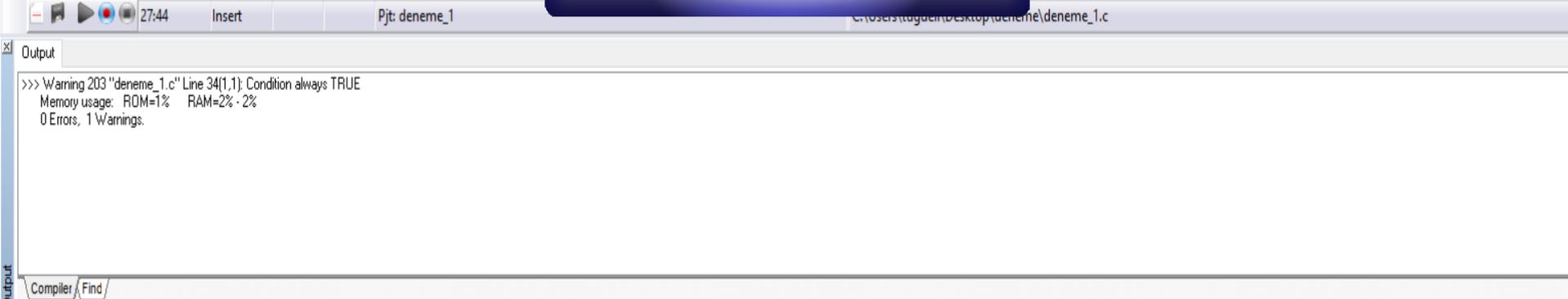
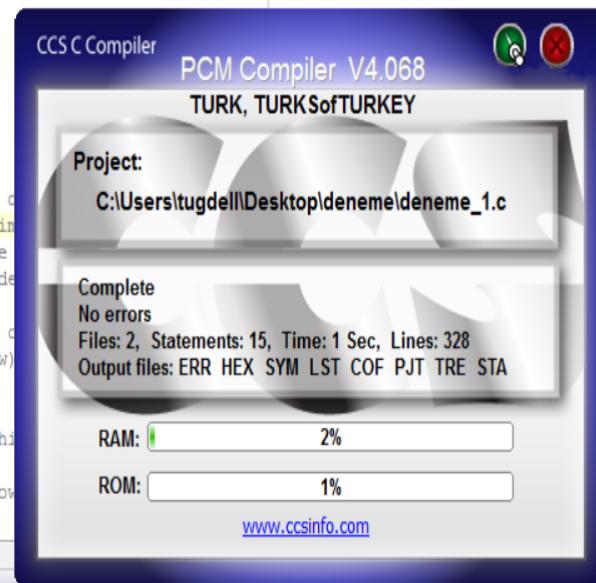
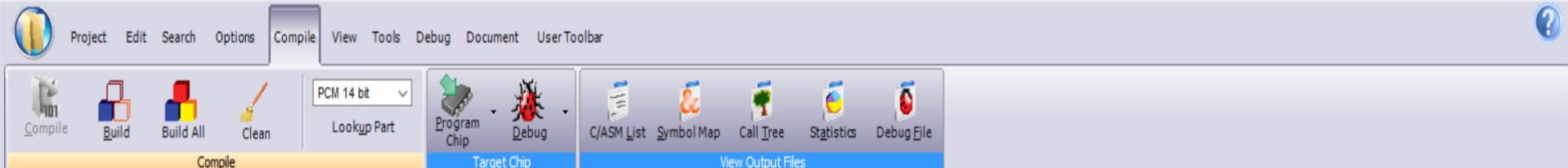
```
#use delay(clock=4000000)
```

// Port yönlendirme komutları PORT(B) için ayarlanıyor

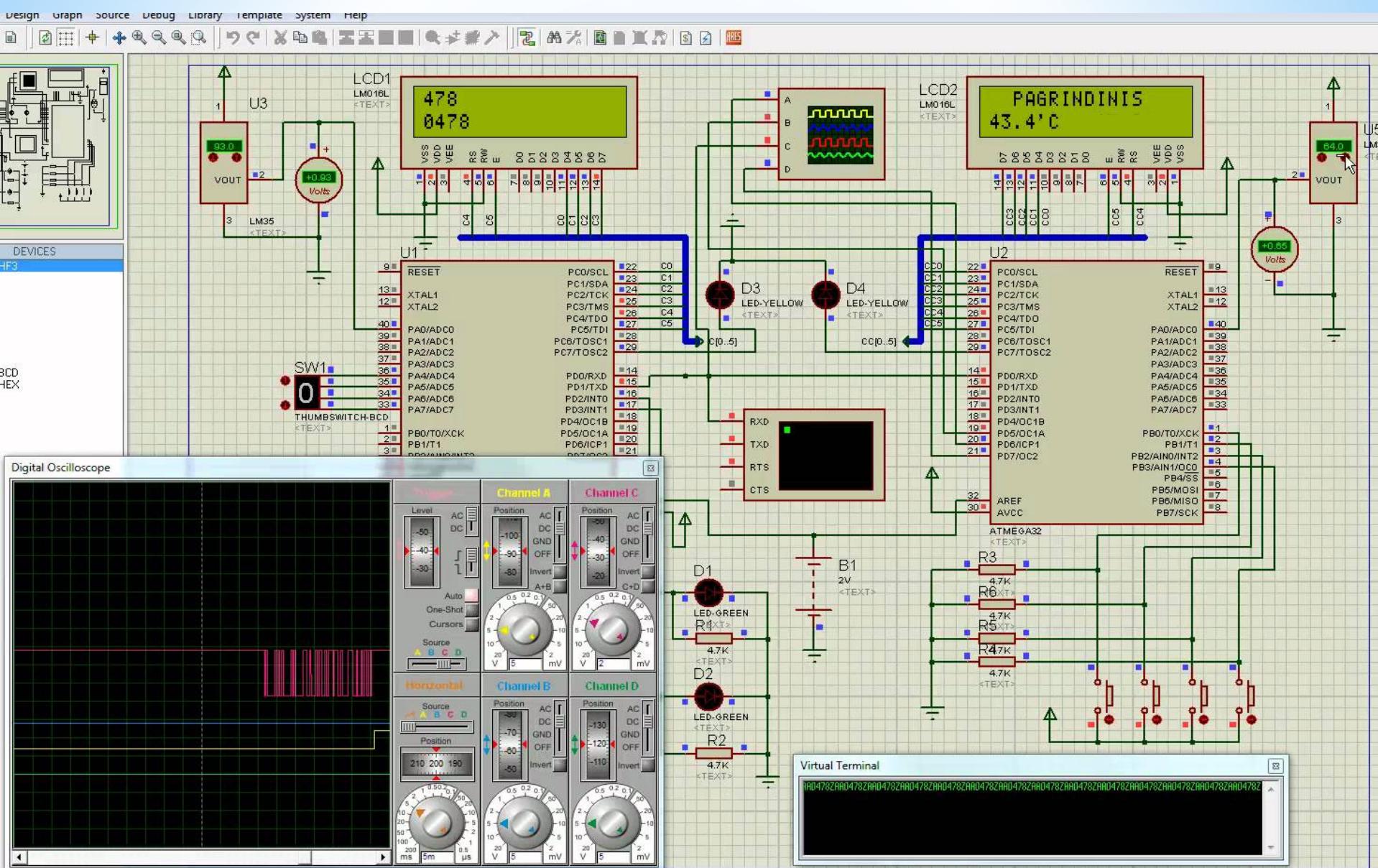
```
#use fast_io(b)
```

```
void main()
{
    setup_adc_ports(NO_ANALOGS); // Analog giris yok
    setup_adc(ADC_OFF); // ADC birimi devre disi
    setup_psp(PSP_DISABLED); // PSP birimi devre disi
    setup_spi(SPI_SS_DISABLED); //SPI interface birimi devre disi
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1); // CCP birimi devre disi
    setup_timer_1(T1_DISABLED); // Timer 1 birimi devre disi
    setup_timer_2(T2_DISABLED,0,1); // Timer 2 birimi devre disi

    set_tris_b(0x00); // PORT(B) komple cikis (output) olarak ayarlanıyor
    output_b(0x00); // PORT(B) komple cikisi sifir (low) yapiliyor
    while(1)
    {
        output_high(pin_b0); //RB0 pin cikisi lojik 1 (high) yapiliyor
        delay_ms(500); // 500ms gecikme yapiliyor
        output_low(pin_b0); //RB0 pin cikisi lojik 0 (low) yapiliyor
        delay_ms(500); // 500ms gecikme yapiliyor
    }
}
```

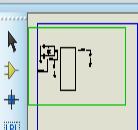


Yazılan Programın Bilgisayar Ortamında Analiz Edilmesi

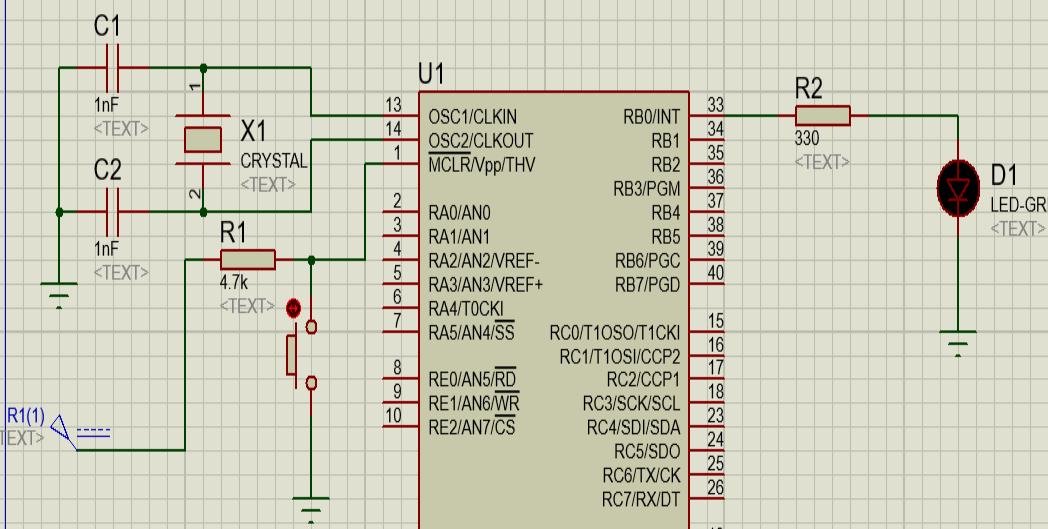




Home Page x Schematic Capture x

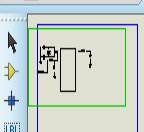


P TERMINALS
DEFAULT
INPUT
OUTPUT
BIDIR
POWER
GROUND
CHASSIS
BUS



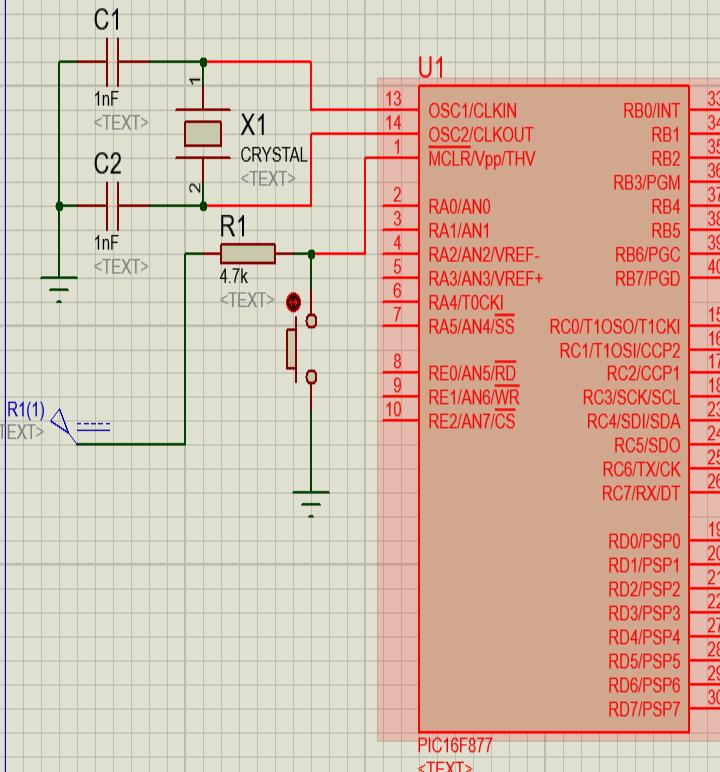


Home Page x Schematic Capture x



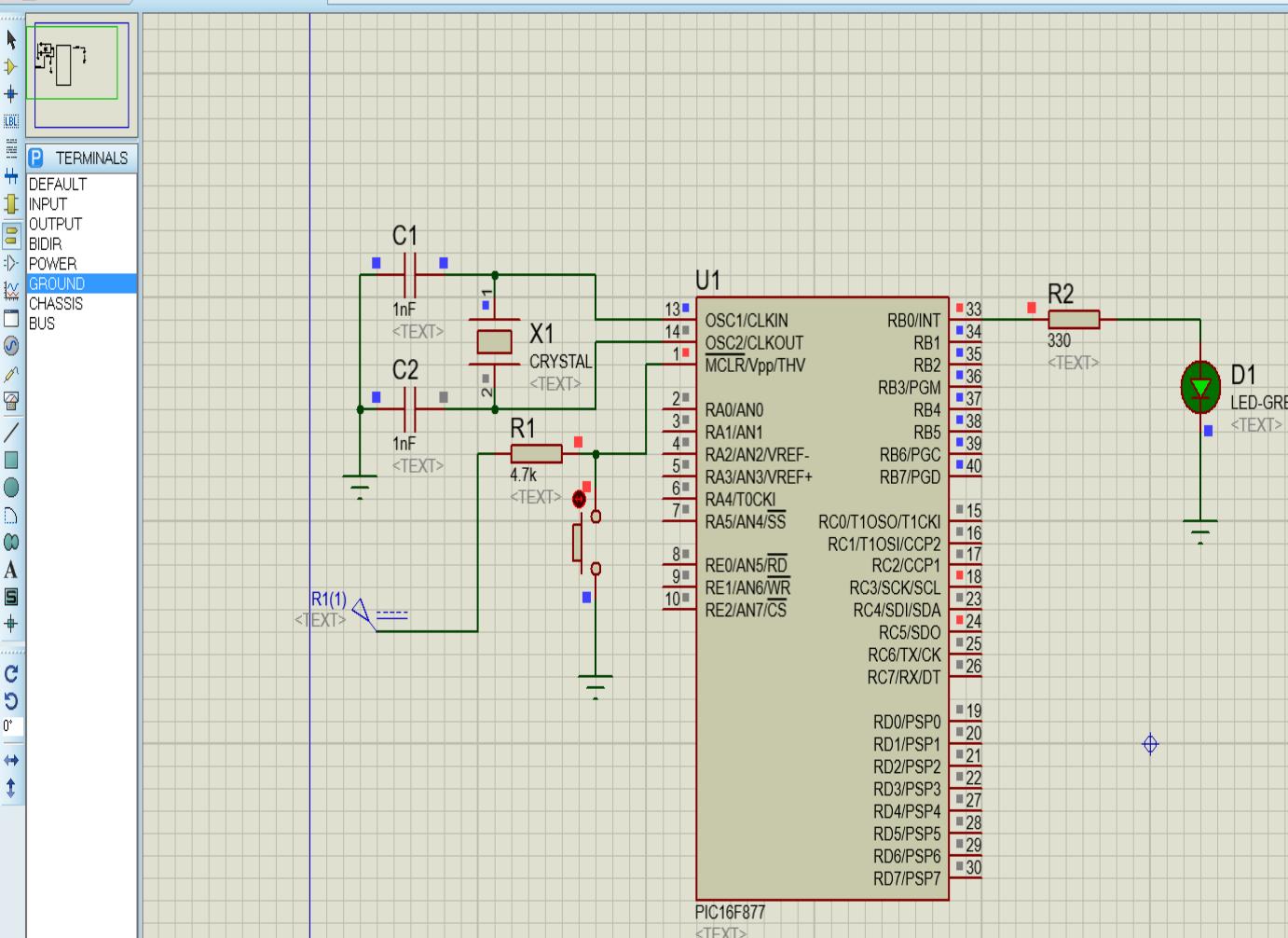
P TERMINALS

- DEFAULT
- INPUT
- OUTPUT
- BIDIR
- POWER
- GROUND**
- CHASSIS
- BUS



Edit Component

| | | | |
|-----------------------------|---|----------------------------------|---------------|
| Part Reference: | <input type="text" value="U1"/> | Hidden: <input type="checkbox"/> | OK |
| Part Value: | <input type="text" value="PIC16F877"/> | Hidden: <input type="checkbox"/> | Help |
| Element: | <input type="button" value="New"/> | Data | |
| PCB Package: | <input type="text" value="DIL40"/> | Hide All <input type="button"/> | Hidden Pins |
| Program File: | <input type="text" value="\\Desktop\\deneme\\deneme_1.hex"/> | Hide All <input type="button"/> | Edit Firmware |
| Processor Clock Frequency: | <input type="text" value="4MHz"/> | Hide All <input type="button"/> | Cancel |
| Program Configuration Word: | <input type="text" value="0x3FFB"/> | Hide All <input type="button"/> | |
| Advanced Properties: | <input type="checkbox"/> Randomize Program Memory? <input type="button" value="No"/> | | |
| Other Properties: | <input type="checkbox"/> Exclude from Simulation <input type="checkbox"/> Attach hierarchy module <input type="checkbox"/> Exclude from PCB Layout <input type="checkbox"/> Hide common pins <input type="checkbox"/> Exclude from Bill of Materials <input type="checkbox"/> Edit all properties as text | | |



Uygulama 2:Button ile LED Kontrol

```
*****
```

Uygulama Adi: PIC16F877 ile Led Yakma-Sondurme Programi
Programin Amaci: RB0 pini ile LED'i RA0 girisine bagli
buton überinden yakip sondurmek

```
***** /
```

```
// Kullanilacak PIC'in baslik dosyasi tanitiliyor  
#include <16f877.h>
```

```
//PIC konfigurasyon ayarları:  
#fuses XT,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD
```

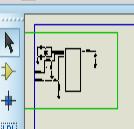
```
// Osilator frekansi seciliyor:  
#use delay(clock=4000000)
```

```
// Port yonlendirme komutları PORT(B) icin ayarlanıyor  
#use fast_io(b)
```

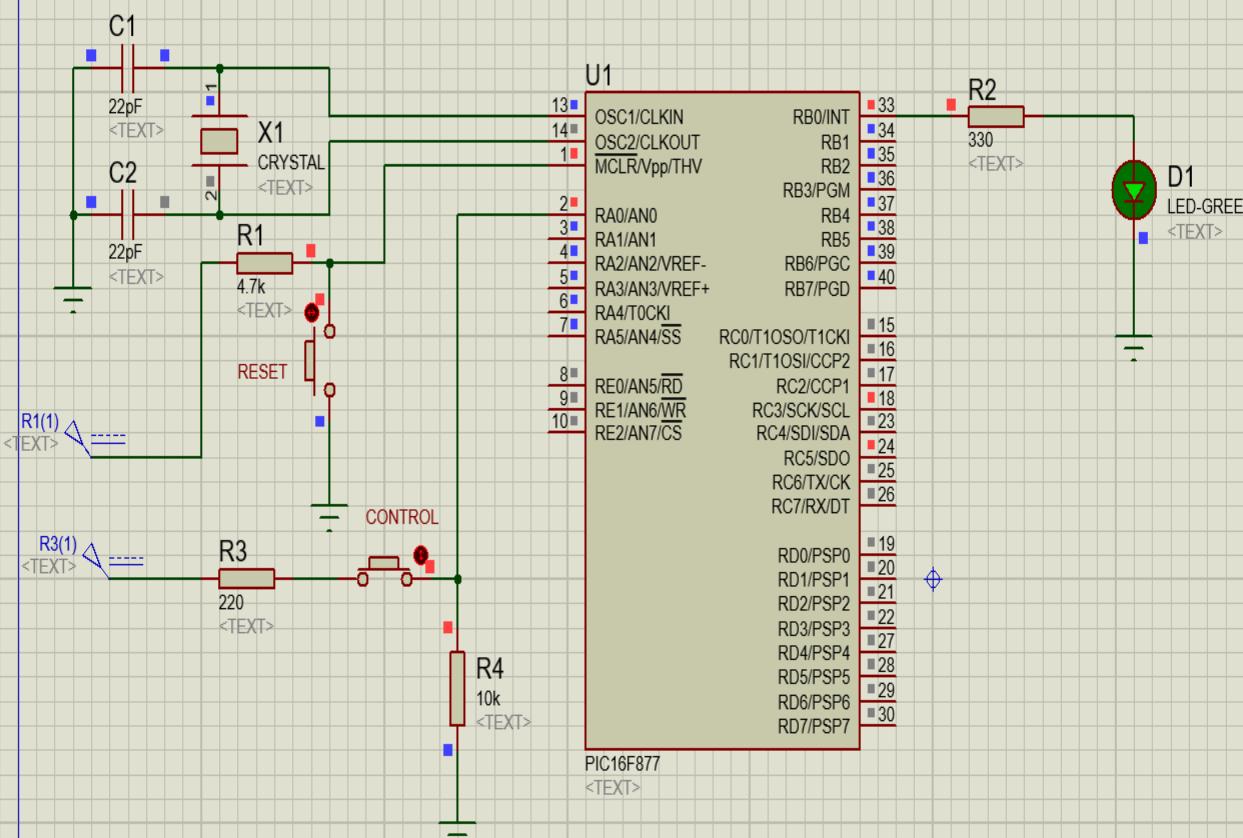
```
int x;
```

```
void main()
{
    setup_adc_ports(NO_ANALOGS); // Analog giriş yok
    setup_adc(ADC_OFF); // ADC birimi devre dışı
    setup_psp(PSP_DISABLED); // PSP birimi devre dışı
    setup_spi(SPI_SS_DISABLED); // SPI interface birimi devre dışı
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1); // CCP birimi devre dışı
    setup_timer_1(T1_DISABLED); // Timer 1 birimi devre dışı
    setup_timer_2(T2_DISABLED,0,1); // Timer 2 birimi devre dışı

    set_tris_b(0x00); // PORT(B) komple çıkış (output) olarak ayarlanıyor
    set_tris_a(0x01); // PORT(A.0) pini komple giriş (input) olarak ayarlanıyor
    output_b(0x00); // PORT(B) komple çıkışının sıfır (low) yapılıyor
    while(1)
    {
        x=input(pin_a0); // PORT(A.0) pini okunuyor x değişkenine atanıyor
        output_bit(pin_b0,x); // x değişkeni RB0 pini çıkışına aktarılıyor
    }
}
```



PL DEVICES
BUTTON
CAP
CRYSTAL
LED-GREEN
PIC16F877
RES



Uygulama 2:LED Toggle Kontrol

```
*****
```

Uygulama Adı: PIC16F877 ile Led Yakma-Sondurme Programı
Programın Amacı: Bu program ile B portuna bağlı 8 adet
LED’i TOGGLE komutu sırayla yakıp sondurmaktır

```
***** /
```

```
// Kullanılacak PIC'in baslik dosyası tanıtılmıyor
#include <16f877.h>
```

```
//PIC konfigurasyon ayarları:
#fuses XT,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD
```

```
// Osilator frekansı seçiliyor:
#use delay(clock=4000000)
```

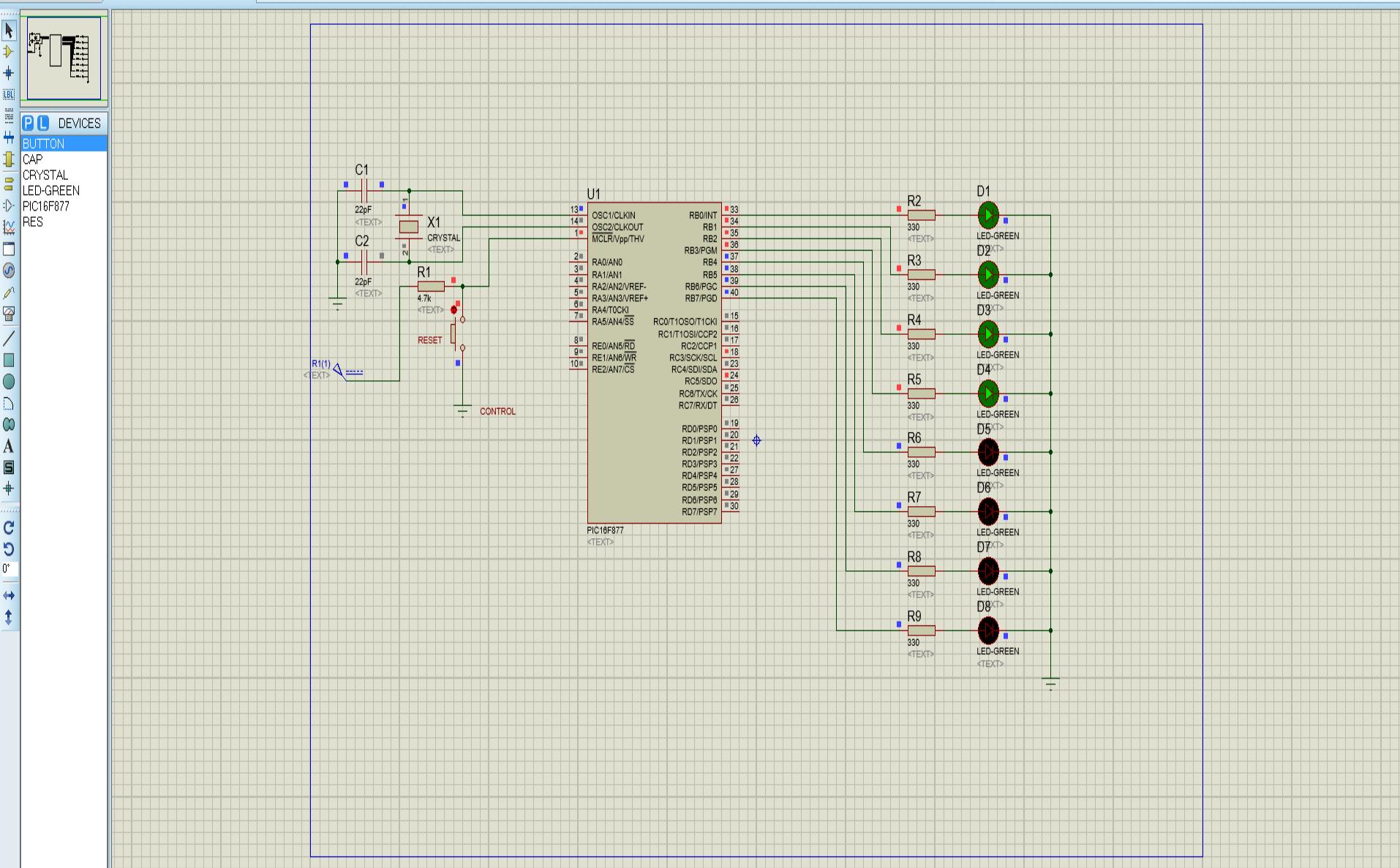
```
// Port yönlendirme komutları PORT(B) için ayarlanıyor
#use fast_io(b)
```

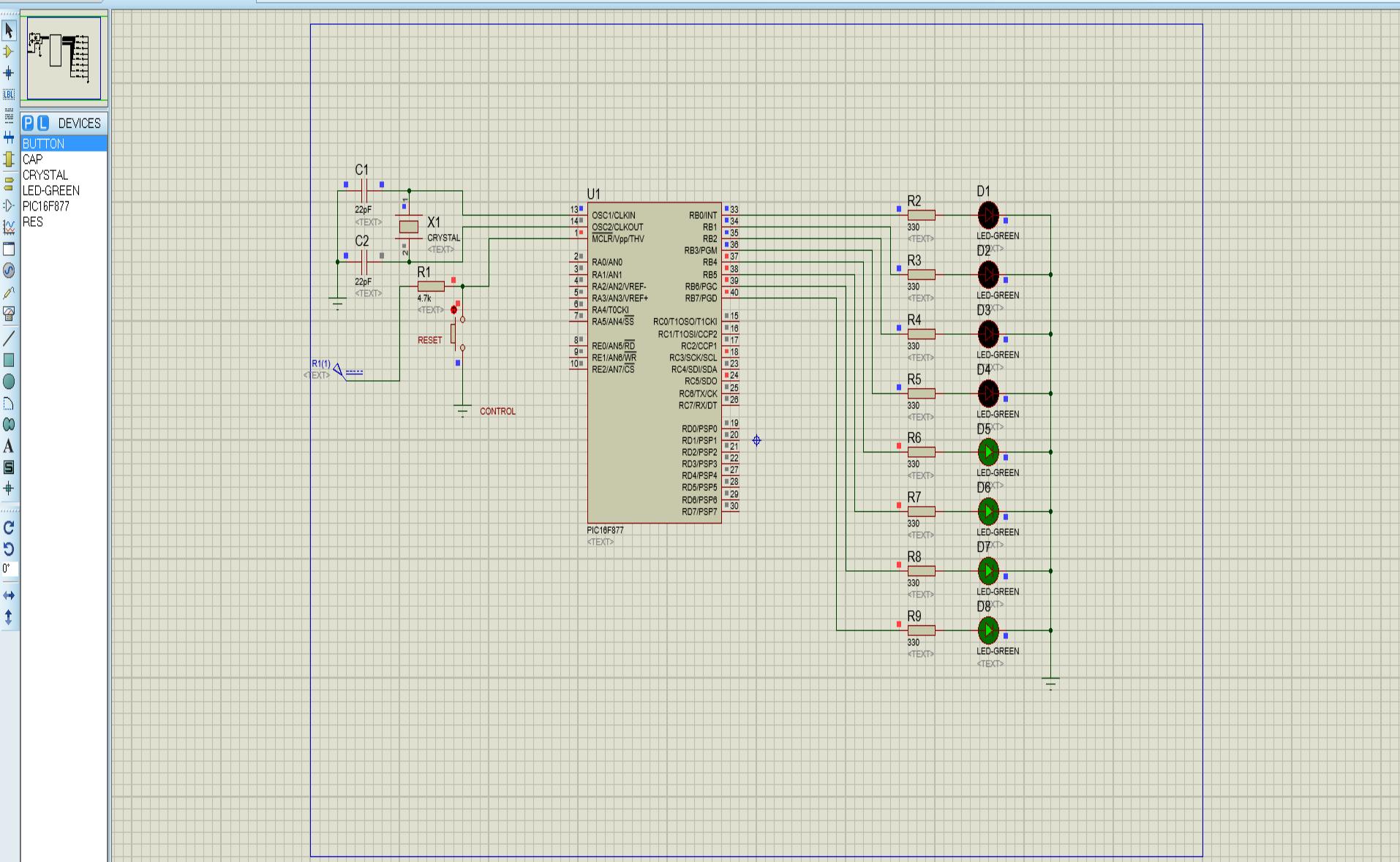
```
void main()
{
    setup_adc_ports(NO_ANALOGS); // Analog giris yok
    setup_adc(ADC_OFF); // ADC birimi devre disi
    setup_psp(PSP_DISABLED); // PSP birimi devre disi
    setup_spi(SPI_SS_DISABLED); //SPI interface birimi devre
disi
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1); // CCP birimi
devre disi
    setup_timer_1(T1_DISABLED); // Timer 1 birimi devre disi
    setup_timer_2(T2_DISABLED,0,1); // Timer 2 birimi devre
disi

    set_tris_b(0x00); // PORT(B) komple cikis (output) olarak
ayarlanıyor
    output_b(0x00); // PORT(B) komple cikisi sıfır (low)
yapılıyor
    while(1)
    {
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b0); // PORT(B.0) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b1); // PORT(B.1) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b2); // PORT(B.2) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b3); // PORT(B.3) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b4); // PORT(B.4) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b5); // PORT(B.5) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b6); // PORT(B.6) cikisi tersleniyor
        delay_ms(500); // 500 ms'lik gecikme veriliyor
        output_toggle(pin_b7); // PORT(B.7) cikisi tersleniyor
    }
}
```



Home Page x Schematic Capture x





TEŞEKKÜRLER...