**ÖNSÖZ**

Bu çalışmada elektronik devreler ve mikro denetleyici PIC16F84A kullanılarak led ışık oyunu tasarımı amaçlanmıştır. Çalışmada devre çalıştırıldığı andan itibaren sırası ile ilk dört ışık sağa doğru kayarak yanacak daha sonra tüm ledler sönecek ve son 4 led sola doğru kayarak yanacak ve tüm ledler söncek.Bu işlem devre kapatana kadar devam ediyor.

**İÇİNDEKİLER**

**ÖNSÖZ --------------------------------------------------------------------------------- 1**

**İÇİNDEKİLER ------------------------------------------------------------------------------ 2**

**1.GİRİŞ -------------------------------------------------------------------------------------- 3**

**1.1 PROJEDE KULLANILAN MALZEMELER ------------------------------------ 3**

**1.2 KULLANILAN MALZEMELERİN İŞLEVLERİ VE ÖZELLİKLERİ ------- 3**

**2. YAPILAN ÇALIŞMALAR ------------------------------------------------------------ 4**

**2.2 MİKRO-İŞLEMCİ KODU ----------------------------------------------------------- 5**

**3.SONUÇ ------------------------------------------------------------------------------------ 8**

**KAYNAKLAR ------------------------------------------------------------------------------ 13**

**1.GİRİŞ**

**1.1 PROJEDE KULLANILAN MALZEMELER**

Mikroişlemci kontrollü led ışık oyunu tasarımımızda, PIC16F84A ve PIC deneme kartı kullandık.

**1.2 KULLANILAN MALZEMELERİN İŞLEVLERİ VE ÖZELLİKLERİ**

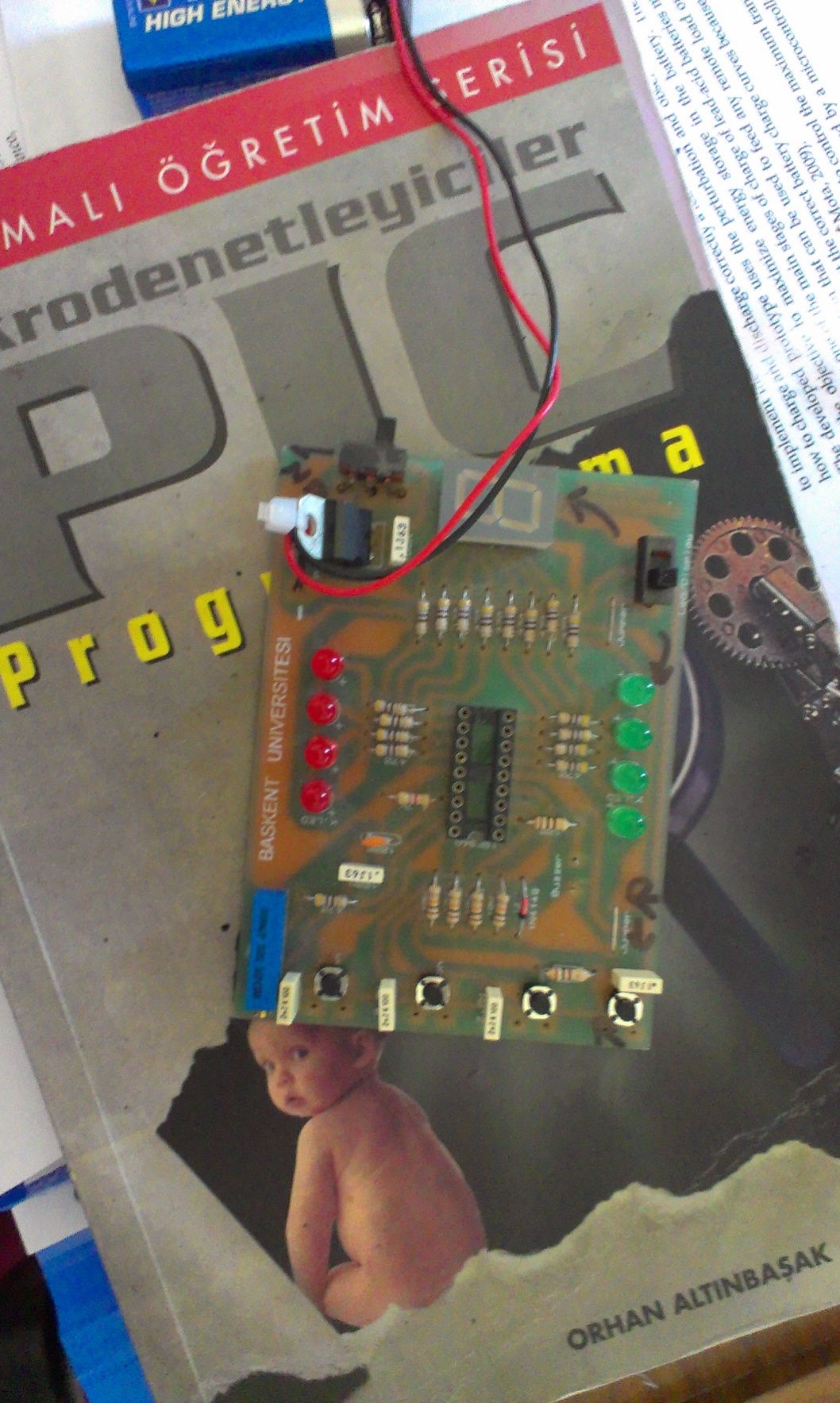
Gerçekleştirilen projede Microchip firmasının ürettiği PIC16F84A mikro işlemcisi kullanılmıştır. Mikro işlemcinin özellikleri datasheet’te mevcuttur.[1]



PIC16F84A

Mikro işlemcimiz ile beraber RC osilatör kullanılmıştır.

Mikro işlemcimizi çalıştırmak için PIC16F84 deneme kartını kullandık.

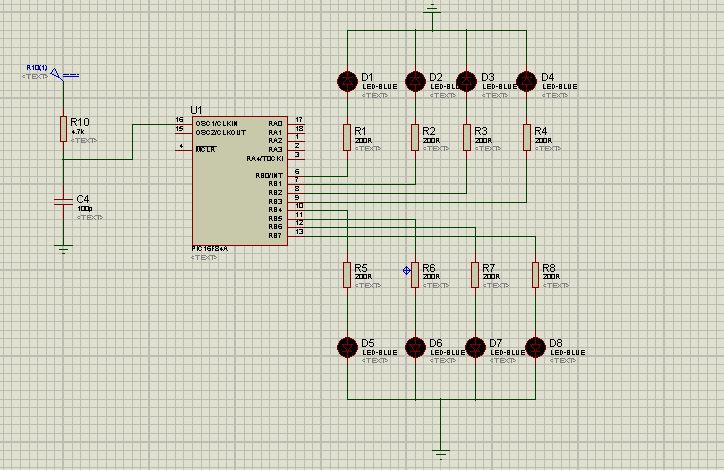


**2. YAPILAN ÇALIŞMALAR**

Mikro denetleyicili led ışık oyunumuzda Mikrochip firmasının MPLAB IDE v8.87 adlı programını kullanarak mikro denetleyici kodumuzu yazdık.

Simülasyon yapmak için Labcenter Electronics firmasının PROTEUS 7 PROFESSIONAL 7.7 SP2 programının ISIS yardımcı programını kullandık. Bu program sayesinde devremizin simülasyonunu önceden yaptık ve test ettik.

**Simülasyon Devresi;**

****

**2.2 MİKRO-İŞLEMCİ KODU**

**TIMER ZAMANININ HESAPLANMASI**

1/RC= ise yaklaşık olarak 1KHz ‘e eşittir.

T=1/f = 1msn

1msn/4= 250 µsn

2sn gecikme için TIMER da kullanacağımız sayaçlara 37 yazmamız yeterli olacaktır.

**AKIŞ DİYAGRAMI:**

****

**KOD :**

LIST P=16F84A

INCLUDE "P16F84A.INC"

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_PWRTE\_ON & \_RC\_OSC

SAYAC2 EQU H'0D'

SAYAC EQU H'0C'

SAYAC3 EQU H'0B'

ORG 0X00

CLRF PORTB

BSF STATUS,5

CLRF TRISB

BCF STATUS,5

TEKRAR

MOVLW H'01'

MOVWF PORTB

SAG

CALL TIMER

RLF PORTB,F

BTFSS PORTB,3

GOTO SAG

GOTO SON

SON

CALL TIMER

MOVLW B'00000000'

MOVWF PORTB

CALL TIMER

MOVLW H'80'

MOVWF PORTB

GOTO SOL

SOL

CALL TIMER

RRF PORTB,F

BTFSS PORTB,4

GOTO SOL

CALL TIMER

MOVLW B'00000000'

MOVWF PORTB

CALL TIMER

GOTO TEKRAR

TIMER

MOVLW H'37'

MOVWF SAYAC

DONGUl

MOVLW H'37'

MOVWF SAYAC2

DONGU2

MOVLW H'37'

MOVWF SAYAC3

DONGU3

DECFSZ SAYAC3, F

GOTO DONGU3

DECFSZ SAYAC2, F

GOTO DONGU2

DECFSZ SAYAC, F

GOTO DONGUl

RETURN

END

**İŞLEM BASAMAKLARI**

1. Config ayarları yapıldı. RC osilatör modu ayarlandı.
2. Gecikme için sayaçlar tanımlandı.
3. Başlama adresi yazdık.
4. PORTB yi temizledik.
5. STATUS’ UN beşinci bitini 1 yaparak BANK1'e geçtik.
6. PORTB'yi çıkış olarak tanımladık.
7. STATUS’ UN beşinci bitini 0 yaparak BANK0'a geçtik.
8. Tekrar birinci ledi yakması için TEKRAR döngüsünü tanımladık
9. İlk ledi yaktık.
10. Birinci ledden 4.lede kayarak gitmesi için döngü kurduk.
11. TIMER gecikme döngüsünü çağırdık.
12. 1.ledden başlayarak yaktığımız ledleri 4.lede kadar kaydıran komut.
13. RB3'e geldiğimizi kontrol edip SON döngüsüne gideceğimiz komut.
14. RB3 ledi yanmadıysa SAG döngüsüne git.
15. RB3 ledi yandıysa SON döngüsüne git.
16. SON döngüsü tüm ledleri sıfırlar. Ve 8. ledi yakar.
17. TIMER gecikme döngüsü çağırdık.
18. Ledleri söndürdük.
19. Sola kayması için 8. ledi yaktık.
20. 8. ledden 5. lede kaydıran komut.
21. RB4'in yanıp yanmadığını kontrol ettik.
22. RB4 deki led yanmadıysa döngüye devam et.
23. RB4 deki led yandıysa ledleri söndür.
24. Tekrar döngüsüne git.
25. RC osilatör için gecikme döngüsü kurduk.

**3.SONUÇ**

Devreyi çalıştırdığımızda ilk önce PORTB’nin 0.bitine (RB0) bağlı 1.led yanmıştır. Daha sonra 2sn gecikme ile PORTB’nin 0.bitinden (RB0) 3.bitine kadar ledlerimiz sağa doğru kayarak yanmıştır.

İşlem sırası;

RB0 🡪 RB1 🡪 RB2 🡪 RB3

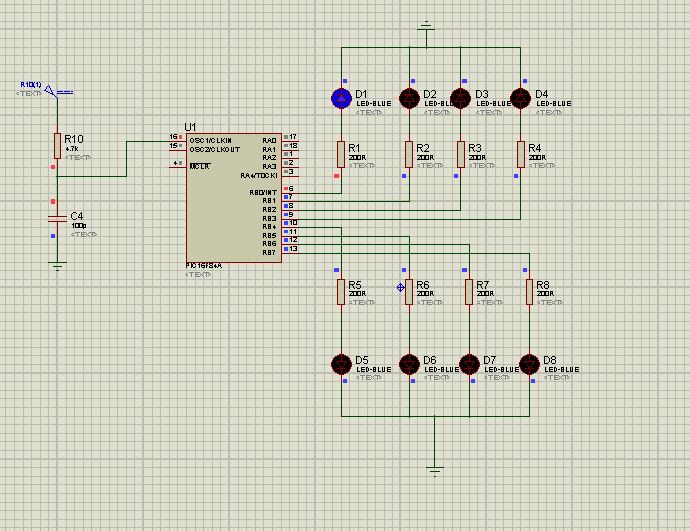
Bu işlemler tamamlandıktan sonra tüm ledler sönmüştür. Devam eden işlem PORTB’nin 7.bitinden (RB7) 4. bitine kadar (RB4) ledlerimiz sola doğru kayarak yanmıştır.

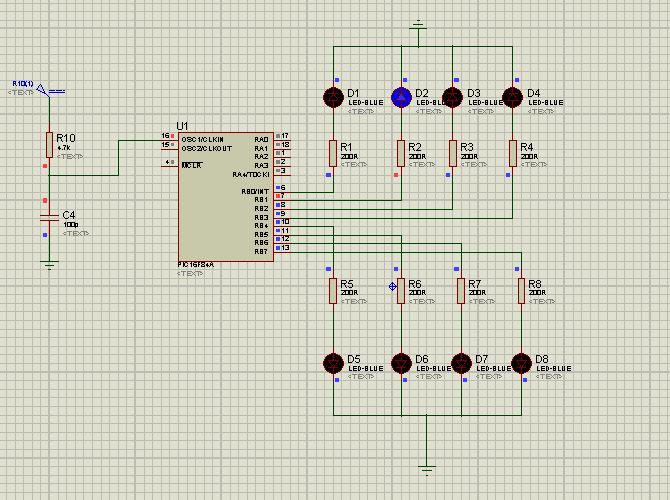
İşlem sırası;

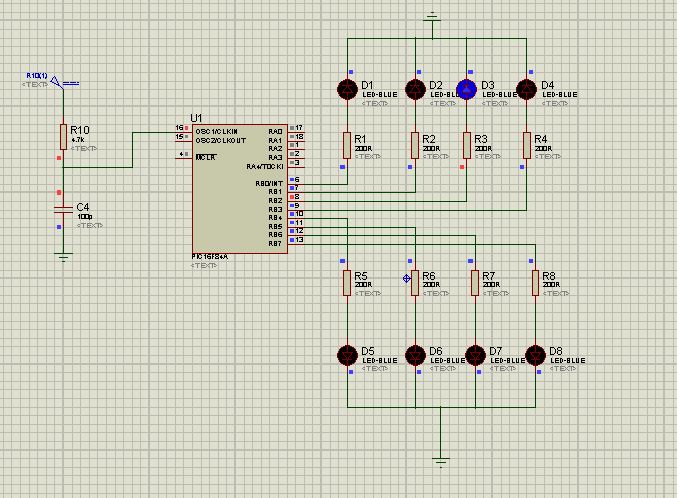
RB7 🡪 RB6 🡪 RB5 🡪 RB4

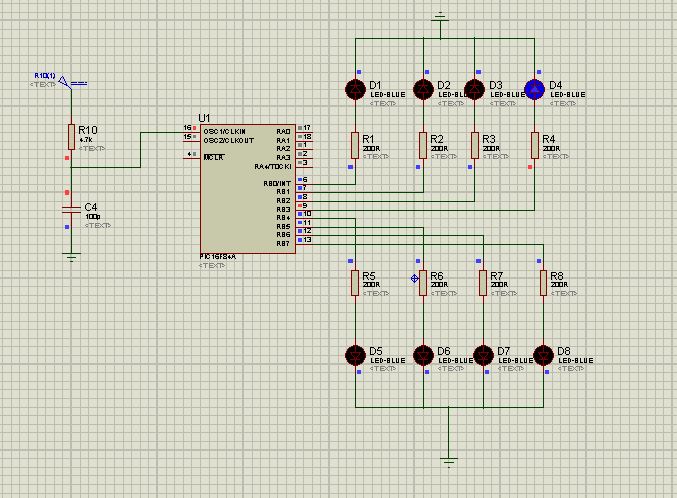
Bu işlemler sürekli bir şekilde devam etmesi için tekrar döngüsü oluşturulmuştur.

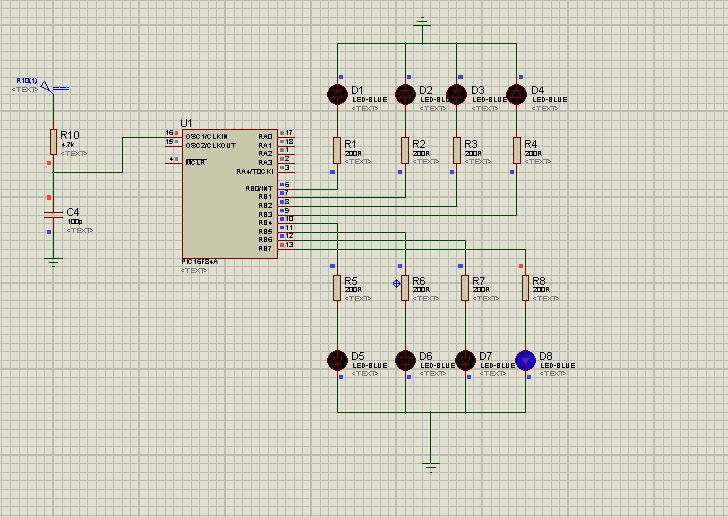
**Simülasyon Sonuçları;**

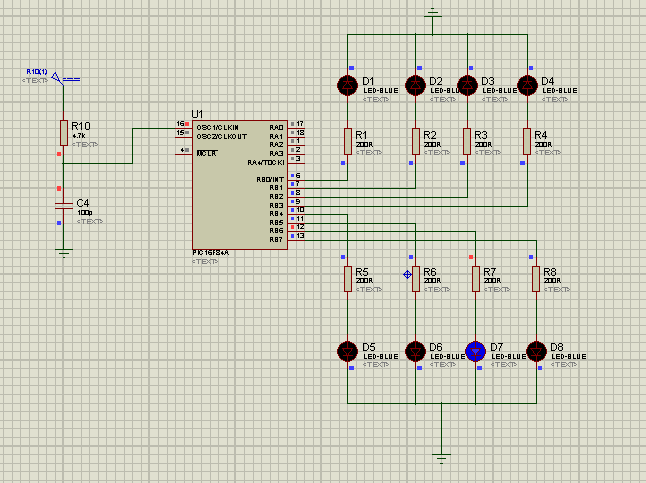
****

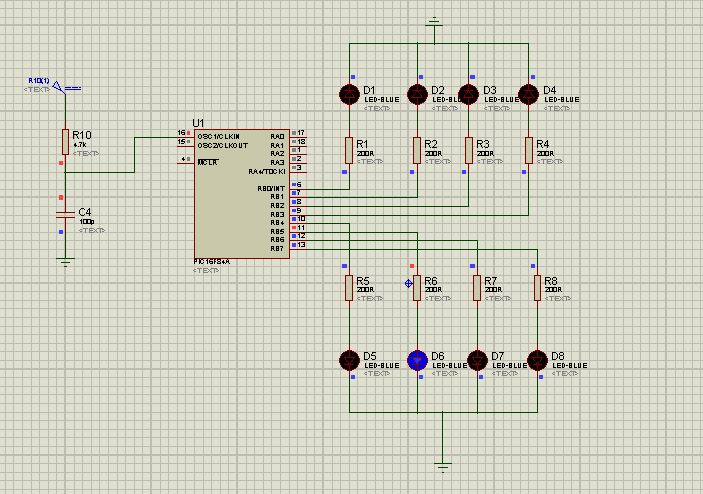
****

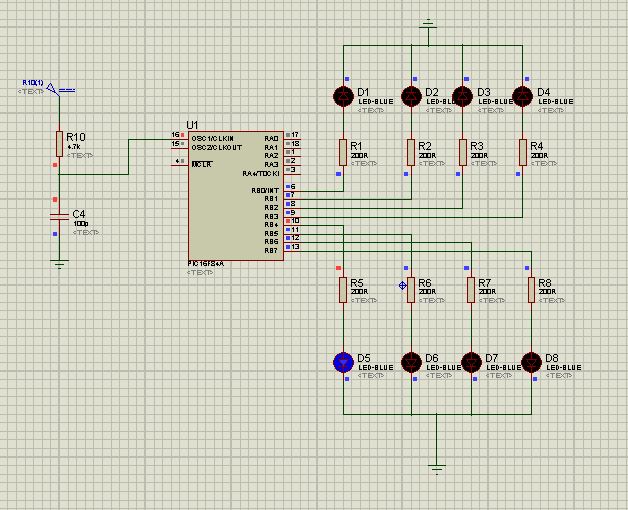
****

****

****

****

****

****

**KAYNAKLAR**

1. <http://ww1.microchip.com/downloads/en/devicedoc/35007b.pdf>
2. **MİKRODENETLEYİCİLER VE PIC PROGRAMLAMA**

**– ORHAN ALTINBAŞAK**

1. **PIC16F84 UYGULAMALARI - ENGİN TEKİN --METİN BEREKET**