

Mikroişlemciler 1 Çalışma Soruları/Microprocessor Questions for Exercise

1. DJNZ komutu ve dolaylı adresleme kullanarak aşağıda verilen adreslere karşılardaki verileri oluşturunuz.

By using the DJNZ command and indirect addressing, create the data given below to the addresses which is opposite of them.

RAM	
adres	veri
10h	10h
0Fh	0Fh
0Eh	0Eh
.	.
.	.
.	.
01h	01h
00h	xxxx

2. Yukarıda hazırladığınız ram adreslerindeki verilerin bilinmediği varsayılarak 10h adresinden 01h adresine kadar olan verileri toplayınız ve P0'da toplam değeri gösteriniz. DJNZ döngüsünde yapınız.

Assuming that the data in the ram addresses above is not known, add all data from 10h to 01h by using the djnz loop and show the total value on P0.

3. $r0 = \#5d$ değeri veriniz. DJNZ komutu kullanarak döngü ile $5*4*3*2*1$ işlemini yapınız ve çarpım değerini p0'da gözlemleyiniz.

(Give $r0 = \#5d$ value. Perform process $5*4*3*2*1$ with a cycle using the DJNZ command and observe the multiplication value in p0.)

4. $r0 = \#10d$ değeri veriniz. DJNZ komutu kullanarak döngü ile $10+9+8+7+6+5+4+3+2+1$ değerini p0'da gözlemleyiniz.

(Give $r0 = \#10d$ value. Perform process $10+9+8+7+6+5+4+3+2+1$ with cycle using the DJNZ command and observe the total value in p0.)

5. Nibble : 8 bitlik bir verinin 2 adet 4 bitlik verilerine verilen addır. Örneğin 84h gibi bir verinin 8 ve 4 sayıları ayrı ayrı nibble olarak adlandırılır.

(It is the name given to two 4-bit data of an 8-bit data. For example, the 8 and 4 numbers of data such as 84h are called nibble separately.)

Adres	Yüksek Nibble (LN)	Düşük Nibble (HN)
R0=#12h	1	2
R1=#34h	3	4

R0'in LN'si + R1'in HN'sini toplayıp p0 da gözlemleyiniz. $P0 = 1+4=5$ olacak.

(Observe in P0 after add the LN of R0 and HN of R1. $P0 = 1+4=5$)

R0'in HN'si + R1'in LN'sini toplayıp p1 da gözlemleyiniz. $P1 = 2+3=5$ olacak.

(Observe in P1 after add the HN of R0 and LN of R1. $P1 = 2+3=5$)

6. Aşağıda verilen adreslerdeki verileri bir döngü içinde DJNZ ve inc veya dec kullanarak oluşturunuz.

(Using the DJNZ, inc and dec, create the data given below to the addresses which is opposite of them in cycle.)

Adres	HN	LN
04h	3	3
03h	3	4
02h	3	5
01h	3	6
00h	xxxx	

7. Yukarıda oluşturduğunuz verilerin bilinmediği varsayılarak (sadece adreslerin bilindiği varsayılıyor) bir döngü içinde sadece LN lerini toplayıp P0 da gözlemleyiniz. Sonuç=P0=3+4+5+6

(Assuming that the data in the addresses above is not known, add the LN data and show the total value on P0. P0= 3+ 4+ 5+ 6)

8. 6. Soruda oluşturduğunuz verilerin bilinmediği varsayılarak sadece HN'leri toplayıp sonucu P0 da gözlemleyiniz. P0=3+3+3+3=12d olacak

(Assuming that the data in the addresses in sixth question is not known, add the HN data and show the total value on P0. . P0= 3+ 3+ 3+ 3= 12d)

9. Adreslerdeki verilerin bilinmediği varsayılarak verilen ok yönlerindeki verileri toplayınız. Sonucu P0 da gözlemleyiniz. P0=3+4+3+6=16d Döngü içinde yapmanıza gerek yoktur.

(Assuming that the data in the addresses is not known, add the data which is in arrow direction and show the total value on P0. P0= 3+ 4+ 3+ 6= 16d. There is no need to do in the cycle.)

Adres	HN	LN
04h	3	3
03h	3	4
02h	3	5
01h	3	6
00h	0	4

10. p1.0 pininden 0 değeri gelmiş ise P3'e bağlı 8 ledin durmaksızın (bir döngü içerisinde) sadece sağdan sola doğru (p3.0 dan p3.7'ye doğru) teker teker sırasıyla yanmasını gerçekleştiren kodu yazınız. (rl veya rlc komutunu bir döngü içinde kullanmadan yapınız.)

(If the value of p1.0 is 0, please write the code that performs unceasingly respectively the one by one flashing of 8 LEDs connected to P3 (in the one cycle) from right to left (p3.0 to p3.7). (Do not use rl or rlc in a cycle)

11. Soru için rl komutunu 1 kez kullanarak döngü içinde kodu tekrar yazınız.

(In the above question, please write again using rl code one time in the cycle)

12. Soru için p1.0 pininden 0 değeri gelmiş ise soldan sağa, p1.1'den de 0 değeri gelmiş ise sağdan sola ledleri tek tek sırasıyla yakınız.

(If the value of p1.0 is 0, please write the code that performs unceasingly respectively the one by one flashing of 8 LEDs connected to P3 (in the one cycle) from left to right and If the value of p1.1 is 0, please write the code that performs unceasingly respectively the one by one flashing of 8 LEDs connected to P3 (in the one cycle) from left to right)

13. CJNE komutu kullanarak 5! (faktöriyel 5) işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

(Using the CJNE code, perform the process of 5! (faktöriyel 5). Show the result on P0)

14. CJNE komutu ile aşağıda verilen adreslere dolaylı adresleme kullanarak karşılarındaki verileri yazınız. (Tabloda 16 adres ve veri vardır)

(Using with the CJNE command indirect addressing, create the data given below to the addresses which is opposite of them.(There are 16 addresses and data in the table))

Adres	Veri
1FH	4FH
1EH	4EH
.	.
.	.
11H	41H
10H	40H

15. CJNE komutu ile aşağıda verilen adreslere dolaylı adresleme kullanarak karşılarındaki verileri yazınız. (Tabloda 16 adres ve veri vardır)

(Using with the CJNE command indirect addressing, create the data given below to the addresses which is opposite of them.(There are 16 addresses and data in the table))

Adres	Veri
1FH	70D
1EH	68D
1DH	66D
.	.
.	.
11H	42D
10H	40D

16. P1 bağı 4 button vardır (P1.0-P1.1-P1.2-P1.3). (Butonlara basınca, pine gelen değerler 0 olmaktadır.) Dolasıyla hiçbir pine basılmamışsa P1'den gelen değer 15 olmaktadır. Yüksek Nibble'dan gelen veriler maskeleyeyle sıfırlanmalıdır. P1'den bu şekilde gelecek verinin 1'den gelen değere kadar toplamını p3'te gözlemleyiniz. (Örneğin porttan gelen veri 10 ise 1'den 10'a kadar olan sayıların toplamını p3'te gözlemleyiniz.)

There are 4 buttons connected to P1 (P1.0-P1.1-P1.2-P1.3). (When you press the buttons, the values from the pin are 0.) If no pin is pressed, the value from P1 is 15. Data from the high Nibble must be reset by masking. (For example, if the data from the port is 10, observe the sum of the numbers from 1 to 10 in p3.) Observe the adding of data which is come from P1 in this way until value that come from P1 on P3. (For example, if the data from the port is 10, observe the sum of the numbers from 1 to 10 in p3.)

17. Bir üst soruda yapılan toplama işlemini alt fonksiyonda yapınız.

(Please perform the process of collecting which is done in the above question, in the sub-function)

18. p3'ten gelen veri p0'dan gelen veriye eşit ise p1.1 bağı ledi, aksi takdirde p1.2 ledini yakınız. JZ komutunu kullanınız.

(If the data from P3 is equal to the data from P0, turn on the P1.1 LED, otherwise burn the P1.2 LED. Use the JZ command.)

19. P2-P3= işleminin sonucu pozitif ise p1.1=5v yapınız aksi takdirde p1.2=5v yapınız.

(If the result of process of P2-P3 is positive, perform the P1.1 = 5V, otherwise p1.2=5v)

20. Aşağıda verilen 16 bitlik sayıları çıkarınız. Sonuç pozitif ise p1.1=5v yapınız aksi takdirde p1.2=5v yapınız.

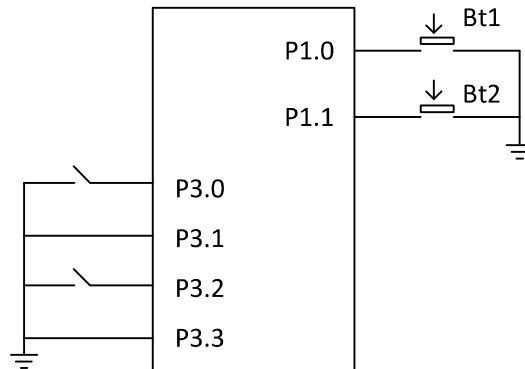
(Substract the 16-bit numbers given below. If the result is positive, perform the p1.1=5v, otherwise perform p1.2=5v.)

1105h- 080Ah

21. P3'te yüksek nibble'daki değer bilinmiyor. P3'ün düşük nibble'da 10d değeri vardır. Bt1 butonuna basıldığında p3'ten alınan düşük nibble değer (10d=0Ah) 2 katını, Bt2 butonuna basıldığında ise 3 katını alınız. Sonucu P2'da gözlemleyiniz.

Pinlerin kontrolünü (butonlara basılıp basılmadığını) jb veya jnb komutu kullanarak yapılacaktır.

(The value in high nibble in P3 is unknown. There is 10d in the low nibble of P3. When the Bt1 button is pressed ,multiply with 2 the value of low nibble which is got from P3 and When the Bt2 button is pressed ,multiply with 2 the value of low nibble which is got from P3. Show the result in P2)



22. Yukarıdaki çarpma işlemlerini alt programda yapınız. Bir alt programınız olacak o sadece 2 katını alacak, diğer bir alt programınız ise 3 katını alacak.

(Perform the above multiplication processings in the subprogram. in the first subprogram , Multiply with 2 the value .In the second subprogram ,multiply with 3 the value)

23. Sorudaki programı, Keil programını adım adım çalıştırırken acall kodu çalışmadan önce döneceğiniz adresi belirleyiniz. Sp değişimleri ile sp adreslerin gösterdiği içeriklerini gözlemleyiniz.

(While the program of the 22th question is run step by step in the keil program, define the adress before a call code is not run)

24. Aşağıdaki programı yazınız. Veritabanı etiketinin program adresini gözlemleyiniz. Ayrıca veri tabanındaki değerleri ve bunların program hafızadaki adreslerini gözlemleyiniz. Program adresi hem debug modunda disassembly penceresinde hemde memory kısmında c:0x00 şeklinde gözlemleyebilirsiniz.

(Write the program which is below. Observe the program address of the database label. Also observe the values in the database and their address in the program memory. You can observe the program address in both the debug mode which is in the assembly window and the memory part which is shaped c: 0x00.)

```
cs1.a51
1  #include "aduc841.h"
2  org 0000
3  sjmp ayar
4  ayar:
5      ; burada mikro
6      ; ayarları yapılır
7  basla:
8      mov r0, #04h
9      mov a, #33h
10
11      sjmp basla
12
13  veritabani: db 19d,
14               8d,
15               17d,
16               6d,
17               15d,
18               4d,
19               13d,
20               2d,
21               11d,
22               0d // 10 adet veri
23  ;veritabani: db 19d,8d,17d,6d,15d,4d,13d,2d,11d,0d // bu şekilde daha rahat oluşturulabilirdi.
24  end
```

Bundan sonraki sorularda veri tabanındaki değerler bilinmediği sadece veri tabanındaki veri sayısının 10 olduğu bilinerek aşağıdaki soruları cevaplayınız.

(Please answers the next questions that the values are in unknown in database and which the data numbers is known 10 in database)

25. Soruda hazırladığınız programı şu şekilde güncelleyelim: Veri tabanındaki değerlerin bilinmediği sadece veri tabanında 10 veri olduğu bilgisi ile veri tabanındaki 8. değeri (yani 1d verisini) p0 da gösteriniz.

(Please update the program which prepared in 24th question in that: With the knowledge that the values are in unknown in database and which the data numbers is known 10 in database, show 8th value of database in P0.)

26. 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir 10 luk döngü ile (djnz) p0 da tek tek veri tabanındaki bilgileri aşağıda verilen a ve b adımlarında verilen şekilde p0 da gözlemleyiniz.

(Please update the program which prepared in 24th question in that: In a 10-loop cycle, observe the data in (djnz)p0 as given in steps a and b below.)

- İlk önce veri tabanındaki 10. veriden başlayarak geriye doğru 0. veriye kadar olan verileri p0 da tek tek gözlemle. (0d->p0'da, ardından 11d->p0'da, ardından 2d->p0'da.....son olarak 19d->p0'da olacak şekilde)

(Firstly, observe the data starting from the tenth to zeros data respectively one by one in P0.)

- Yine djnz kullanarak 0. veriden 10. veriye kadar olan verileri tek tek p0'da gözlemle.

(Using the DJNZ command, observe the datas which is from the zeros data to tenth data one by one in P0.)

27. Sorudaki kodu cjne ile hem a hem de b şıkkı için yapınız. Bu sefer b şıkkı kolaylıkla yapılabilir.

(Perform code that is in 26th question using the DJNE for both a and b. This time the b is performed easily.)

28. 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Veri tabanında bir arama yapma programı yapılacaktır. Örneğin veri tabanında 6d değeri var ise p0.1 yakınız, yok ise p0.2 yakınız. Örneğin veri tabanında 16d değeri var ise p0.1 yakınız yok ise p0.2 yakınız.

(Please update the program which prepared in 24th question in that: In database, the searchable program will be performed. For example, if there is 6d value in the database, turn on the p0.1 otherwise p0.2. For example, if there is 16d value in the database, turn on the p0.1 otherwise p0.2.)

29. 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri tek tek p0'da gözlemleyiniz. Yani 1. 3. 5. 7. ve 9. sıradaki bilgileri.

(Please update the program which prepared in 24th question in that: Observe single index information in the database one by one with cycle. So the knowledge of lines of the 1th,3th,5th,7th and 9th)

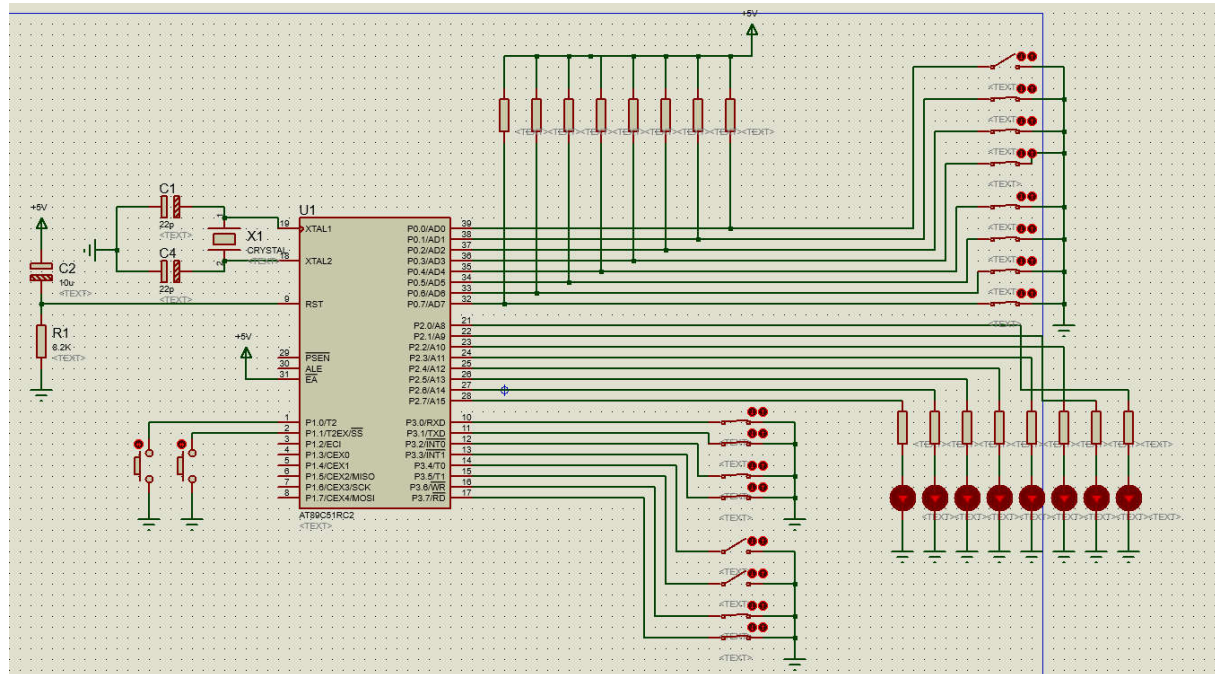
30. 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri toplayalım ve sonucu p0'da gözlemleyelim.

(Please update the program which prepared in 24th question in that: Add knowledges that is single index in the database and observe the result in P0.)

31. 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri toplayalım ve sonucu r7'ye yazalım, ardından çift indisli bilgileri toplayalım ve sonucu r6'ya yazalım. Ve r7-r6 çıkartma işlemini gerçekleştiriniz. Sonuç pozitif ise p0.1 deki ledi yakınız, negatif ise p0.2deki ledi yakınız.

(Please update the program which prepared in 24th question in that: Add knowledges that is single index in the database and write the result in r7 then add knowledges that is double index and write the result in r6. Subtract the r6 from r7. If the result is positive, turn on the LED in P0.1 otherwise the LED in P0.2)

32.



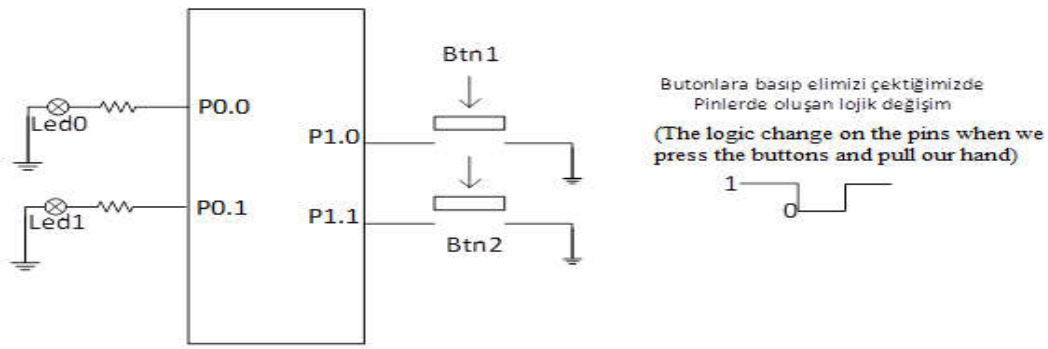
Yukarıda verilen devreyi kurunuz. P0 ve P3 değerleri, pinlere bağlı olan anahtarlardan sağlanacaktır.

Bt1 butonuna basılıp-bırakıldığında p3'ten alınan yüksek nibble değeri ile P0'dan alınan düşük nibble değerini karşılaştıran; bu değerler birbirine eşitse P0'ın 4 katını alan ve sonucu P2'ye bağlı ledleri yakarak gösteren, değerler birbirine eşit değilse iç-içe döngü ile gecikme oluşturarak P2'ye bağlı ledleri bu gecikme ile yakıp söndüren 8051 assembly program kodunu yazınız. Programı Proteousta oluşturulan devre üzerinde çalıştırarak gösteriniz.

(Comparing the lower nibble value from P0 with the higher nibble value from p3 given in the options above when the Bt1 button is pressed and released ; if these values are equal to each other, multiply with 4 the value of low nibble which is got from P0 and show the result in P2. if these values aren't equal to each other, flash the leds connected to the P2 with this delay by creating a delay with an intertwined loop write the code of the 8051 assembly program that flashes the LEDs connected to the P2 with this delay by creating a delay with the nested loop. Run the program on the Proteousta generated circuit.)

bekle :jb bt1,bekle

x :jnb bt1,x



33. Portların başlangıçtaki durumları hep lojik 1'dir. Başlangıçta tüm ledleri söndürünüz.

Btn1 butonuna basılıp-çekildiğinde (p1.0=1_0_1) led0 yakınız.

(The initial state of the ports is always logic 1. Switch off all leds at first. When the Btn1 buttons are pressed and then pulled , turn on LED0)

34. Eğer Btn1 butonuna basılıp-çekildiğinde (p1.0=1_0_1) led0 yakınız ve led1 söndürünüz, eğer Btn2 butonuna basılıp-çekildiğinde led1 yakınız ve led0 söndürünüz.

(If When the Btn1 buttons are pressed and then pulled,turn on LED0 and turn off LED1, If When the Btn2 buttons are pressed and then pulled,turn on LED1 and turn off LED0.)

35. Btn1 butonuna arka arkaya iki kez basılıp-çekilmişse Led0 ledini yakınız.

(If the Btn1 button is pressed twice in succession, turn on the LED0.)

36. Bu biraz ters bir soru:

Btn1 basık iken (lojik 0'da iken), btn2 basılıp-çekilirse led0 ledini yakınız. Aksi halde ledler sönük kalsın.

(When Btn1 is pressed (when logic is 0), if btn2 is pressed and then pulled, turn on led0 otherwise turn off)

37. İki adet 3x3 matrisin toplama işlemi gerçekleştirilecektir. Aşağıda verilen matris, verilerin ram'de tutulacağı adresleri vermektedir.

(Two 3x3 matrices will be added. The matrix that is given below,give addresses which is will be datas in RAM)

$$\begin{bmatrix} 11h & 12h & 13h \\ 14h & 15h & 16h \\ 17h & 18h & 19h \end{bmatrix} + \begin{bmatrix} 21h & 22h & 23h \\ 24h & 25h & 26h \\ 27h & 28h & 29h \end{bmatrix} = \begin{bmatrix} 31h & 32h & 33h \\ 34h & 35h & 36h \\ 37h & 38h & 39h \end{bmatrix}$$

Aşağıda verilen matriste, adreslerde tutulacak verilerin elde ediliş formatı verilmiştir. Toplanacak iki matrisin değerleri ayrı ayrı iki adet iç içe döngüde oluşturulacaktır.

(In the matrix given below, the format of the data to be kept at the addresses is given. The values of the two matrices to be collected will be generated separately in two nested cycles.)

$$\begin{bmatrix} 11h = 1x1 & 12h = 1x2 & 13h = 1x3 \\ 14h = 2x1 & 15h = 2x2 & 16h = 2x3 \\ 17h = 3x1 & 18h = 3x2 & 19h = 3x3 \end{bmatrix} + \begin{bmatrix} 21h = 3x3 & 22h = 3x2 & 23h = 3x1 \\ 24h = 2x3 & 25h = 2x2 & 26h = 2x1 \\ 27h = 1x3 & 28h = 1x2 & 29h = 1x1 \end{bmatrix} = \begin{bmatrix} 31h & 32h & 33h \\ 34h & 35h & 36h \\ 37h & 38h & 39h \end{bmatrix}$$

Ardından ayrı bir iç içe döngüde iki matrisin elemanları toplanarak sonuç matrisinde verilecek adreslere değerler yazılacaktır.

(Then, in a separate nested loop, the elements of the two matrices will be added and the values will be written to the addresses to be given in the result matrix.)

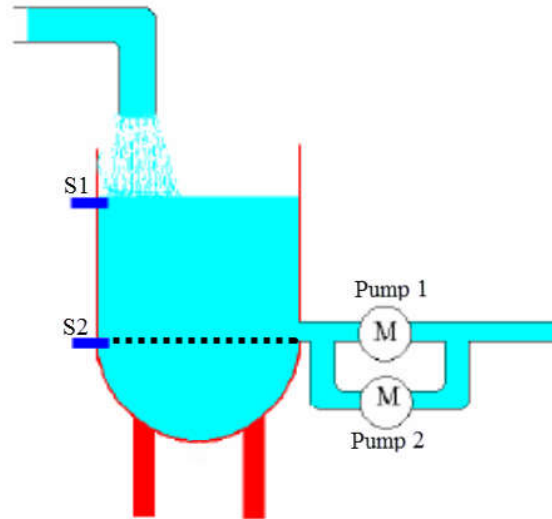
38. Tablo1'de verilen ram adres ve içeriklerini 1 ya da 2 döngü içinde oluşturunuz. Ardından Tablo 1'deki verileri sıralama işlemine tabi tutarak, Tablo2'deki gibi elde ediniz. (Tavsiye: ilk önce tek numaraları adreslere bir döngü içinde, ardından çift numaralı adreslere bir döngü içinde veriler oluşturulabilir. Ardından basit herhangi bir sıralama algoritması araştırılarak bu algoritma ile asm kodları yazılabilir.)

Create The address and contents which is given in Table 1, in one or two cycle. Then obtain the data that is in Table 1, like Table 2. (Recommendation: The datas can be create in the addressess which has first a single number then has double number. After that The asm code can be wrote with algorithm Which is by searching any simple listing

Tablo1 (Döngü ile otomatik oluşturulacak veriler)		Tablo2 (Algoritma ile otomatik olarak sıralanacak veriler)	
Adres	Veri	Adres	Veri
49h	160d	39h	2d
48h	2d	38h	4d
47h	80d	37h	8d
46h	4d	36h	10d
45h	40d	35h	16d
44h	8d	34h	20d
43h	20d	33h	32d
42h	16d	32h	40d
41h	10d	31h	80d
40h	32d	30h	160d

39. Şekildeki depo S1 seviyesinde dolu S2 seviyesinde ise boştur. Depo dolduğunda P1 pompası çalışarak depoyu boşaltacak, boşaldığında kendiliğinden duracaktır. Boşaltma sırasında P1 arızalanırsa P2 otomatik devreye girerek boşaltmaya devam edecektir. P1 pompasının arızalanması; sinyal lambası ile her iki pompanın da arızalanması ise sesli alarm ile ikaz edilecektir.

The tank in the figure is full at level S1 and empty at level S2. When the tank is full, pump P1 will empty as run the tank and it will stop automatically when empty. If P1 fails during unloading, P2 will automatically switch on and continue draining. Failure of pump P1; the signal lamp and the failure of both pumps will be warned by an audible alarm.



Girişler/Inputs	Pin	Cıkışlar/Outputs	Pin
S1 sensöründen gelen veri	P1.0	Pompa 1	P2.0
S2 sensöründen gelen veri	P1.1	Pompa 2	P2.1
Pompa 1 Arıza Sensörü	P1.2	Pompa Arıza Sinyal Lambası	P3.0
Pompa 2 Arıza Sensörü	P1.3	Pompa Arıza Ses İkazı	P3.1

(Sensörlerden 0 verisi geldiğinde aktiftir, Lamba ve Arıza sistemi 5V Logic 1 olduğunda aktif olurlar)

40. Otobüste ön, orta ve arka olmak üzere 3 tane kapı vardır. Bu kapılardan ön kapı yolcu biniş kapısı, orta ve arka kapı ise yolcu iniş kapısıdır. Orta ve arka kapılardan yolcuların inmek için bastıkları birer buton ve butona basıldığını gösterecek lambalar bulunmaktadır. Her kapının buton ve lambası ayrıdır. Şoförün kapıları kontrol etmesi için otobüsün konsolunda bir panel bulunmaktadır. Yolcuların inme ve binme şartları aşağıdaki gibidir.

- İnme butonlarından birine basıldığında şoförün kontrol panelindeki kapıya ait lamba ve kapıdaki lamba yanacak.
- Şoför kontrol panelindeki yanan lambalara ilişkin butonlara bastığında ilgili kapılar açılacak ama buton lambaları sönmeyecek.
- Şoför kapıları kapatmak için ilgili butonlara ikinci kez bastığında kontrol panelindeki lamba ve kapıdaki lamba sönecek ve kapılar kapanacak.

- d. Yolcunun binmesi için ön kapı kontrol panelindeki butonla açılacak (şoför ilgili butona bastığında ön kapı açılacak). Paneldeki lamba yanacak. Butona tekrar bastığında paneldeki lamba sönecek ve kapı kapanacak.

There are 3 doors on the bus, front, middle and rear. The front door is the boarding door, the middle and the rear door is the exit door. There are buttons which is press by passenger to exit from the middle and the rear door, and leds will be show that the button is pressed. The button and lamp of each door are separate. There is a panel on the bus console for the driver to check the doors. The conditions of rearing and exiting of passengers are as follows.

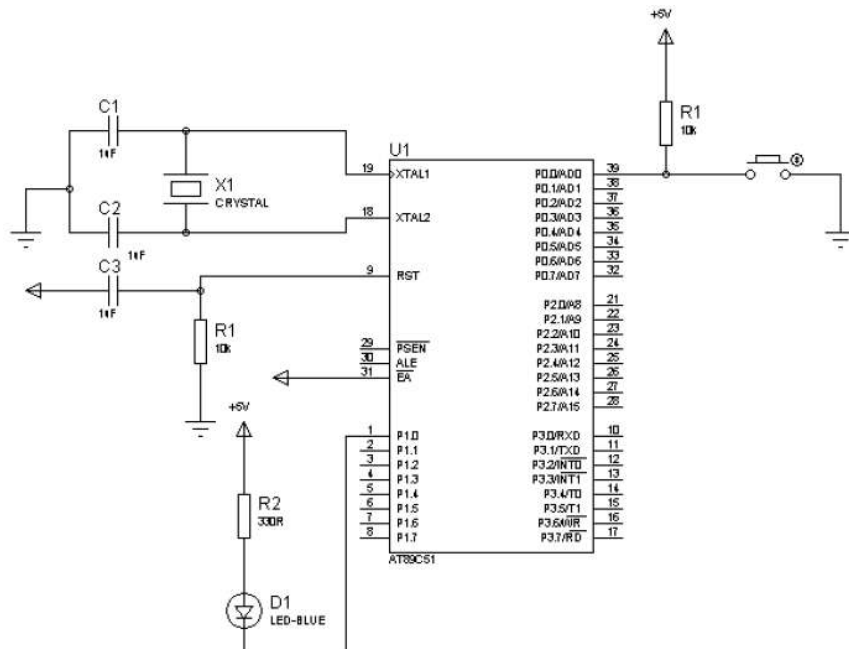
- When one of the exiting buttons is pressed, the lamp of the door in the driver's control panel and the lamp in the door will light.
- The corresponding doors will be opened when the driver presses the buttons on the control panel lights, but the button lamps will not go out.
- When the driver presses the relevant buttons a second time to close the doors, the lamp in the control panel and the lamp in the door will go out and the doors will close.)
- The front door will be opened with the button on the control panel for the passenger to ride (the front door will be opened when the driver presses the corresponding button). The lamp in the panel will light. When the button is pressed again, the lamp in the panel will turn off and the door will close.

Girişler	Pin	Çıkışlar	Pin
Yolcu orta kapı butonu	P1.0	Orta Kapı Lambası	P2.0
Yolcu arka kapı butonu	P1.1	Arka Kapı Lambası	P2.1
Şoför orta kapı butonu	P1.2	Panel Arka Kapı Lam.	P2.2
Şoför arka kapı butonu	P1.3	Panel Orta Kapı Lam	P2.3
Şoför ön kapı butonu	P1.4	Panel Ön Kapı Lam	P2.4
		Ön Kapı ON/OFF	P3.0
		Orta Kapı ON/OFF	P3.1
		Arka Kapı ON/OF	P3.2

Not: Tüm butonlar basıldığında “0” bırakıldığında ise “1” durumundadır.

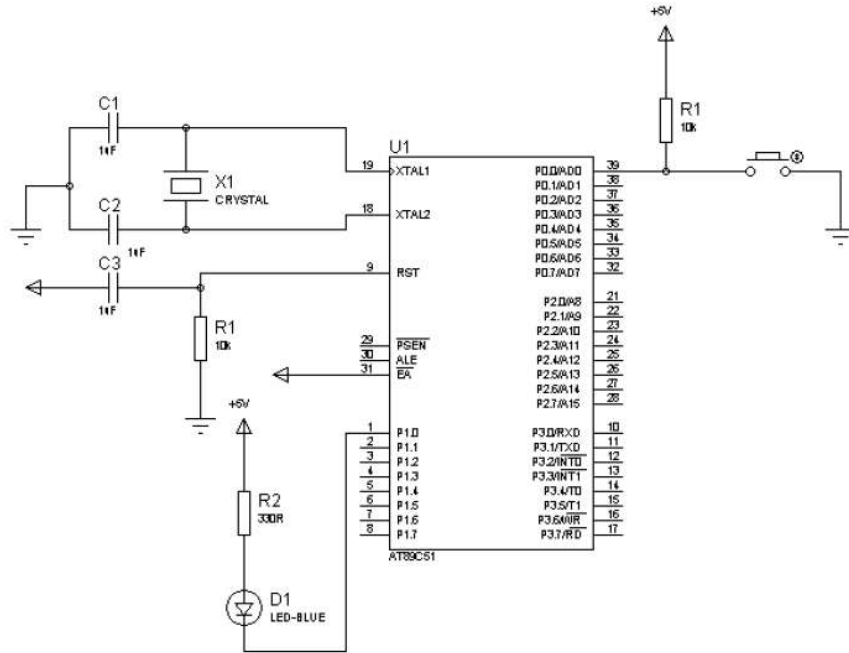
41. Aşağıdaki devrede bulunan P0.0 butonuna **basıldığı anda** P1.0’a bağlı ledi 1 s aralıklarla yakıp-söndüren 8051 assembly program kodunu yazarak proteous benzetimine uygulayınız. Zamanlayıcı-0 16 Bit yoklama modunda kullanılacaktır.

(In the circuit below ,When P0.0 button is pressed, apply the code of 8051 assembly program that burns and extinguishes led connected to P1.0 at intervals of 1 s in proteoid simulation. Timer-0 will be used in 16 Bit poll mode.)



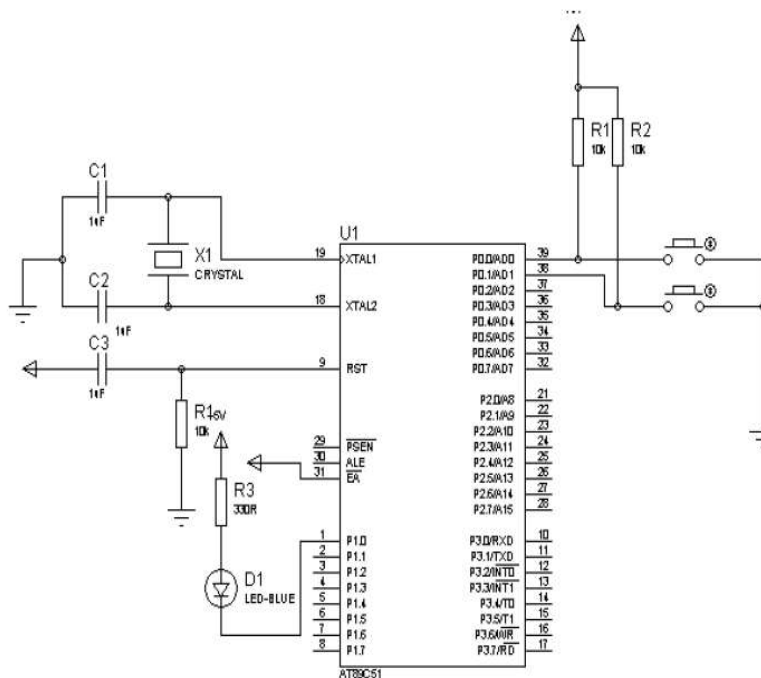
42. Aşağıdaki devrede bulunan P0.0 butonuna **basılıp-bırakıldığında** P1.0'a bağlı ledi 1 s aralıklarla yakıp-söndüren 8051 assembly program kodunu yazarak proteous benzetimine uygulayınız. **Zamanlayıcı-1** 16 Bit yoxlama modunda kullanılacaktır.

(In the circuit below, When P0.0 button is pressed and released, apply the code of 8051 assembly program that burns and extinguishes leds at intervals of 1 s in proteoid simulation. Timer-0 will be used in 16 Bit poll mode.)



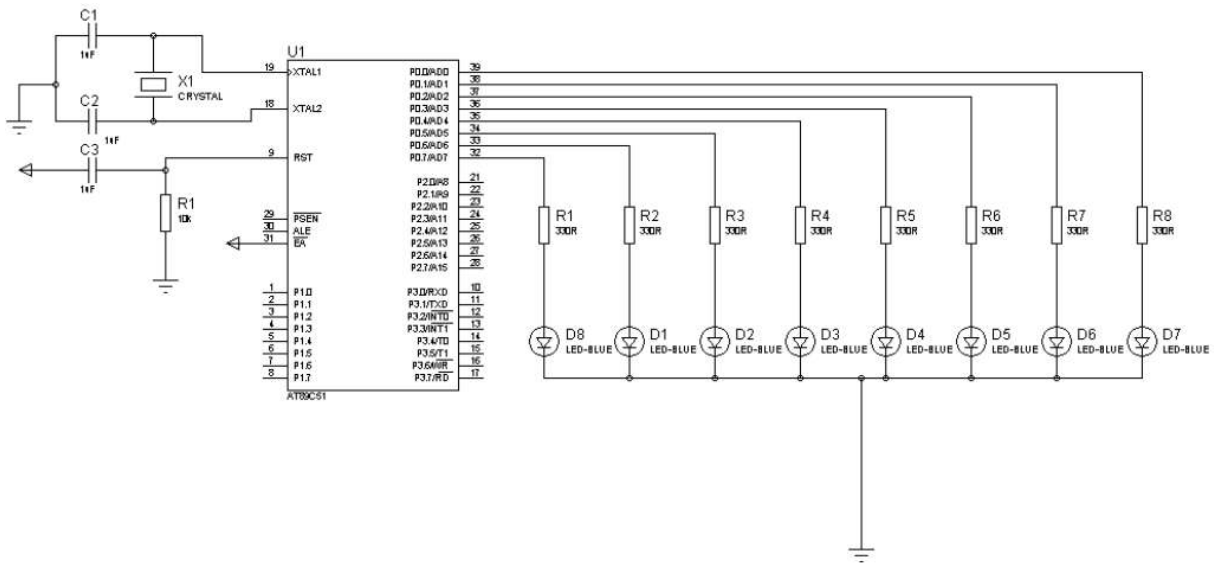
43. Aşağıdaki devrede bulunan P0.0 butonuna basılınca P1.0'a bağlı ledi yakan ve yakılı halde tutan, P0.1 butonuna basılınca ledi söndüren ve sönük tutan 8051 assembly program kodunu yazınız.

(In the circuit below ,When P0.0 button is pressed, apply the code of 8051 assembly program that burns and continuous to led connected to P1.0 at intervals of 1 s in proteoid simulation or When P0.1 button is pressed, apply the code of 8051 assembly program that extinguishes and continuous to led connected to P1.0 at intervals of 1 s in proteous simulation.)



44. Aşağıdaki devrede bulunan 8 adet ledi, binary olarak 0.8 saniye aralıklarla sıralı yakan 8051 assembly program kodunu yazınız. Zamanlayıcı-2 16 Bit otomatik yükleme modunda kullanılacaktır. (Örnek: 0000 0001, 0000 0010, 0000 0011,....., 1111 1111)

In the circuit below, Type the code of the 8051 assembly program, which issequentially burn the 8 piece of led as binary at 0.8 second intervals. Timer-2 will be used in 16-bit auto-load mode.

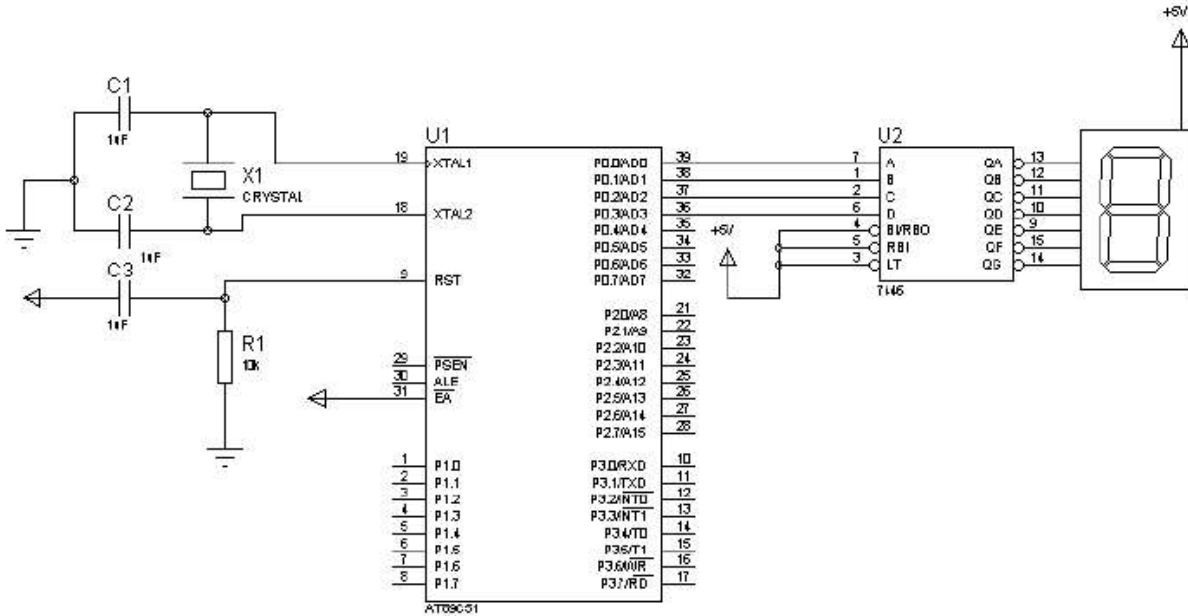


45. Aşağıdaki devrede bulunan 7 segment displayde sıralı olarak 0'dan – 9'a kadar rakamları yazdıran programı yazınız.

(Örnek: 0.1.2.3.4.5.6.7.8.9.0.1...)

Write the program that prints numbers as from 0 to 9 in sequence in the 7-segment display in the following circuit.

Devre Şeması:

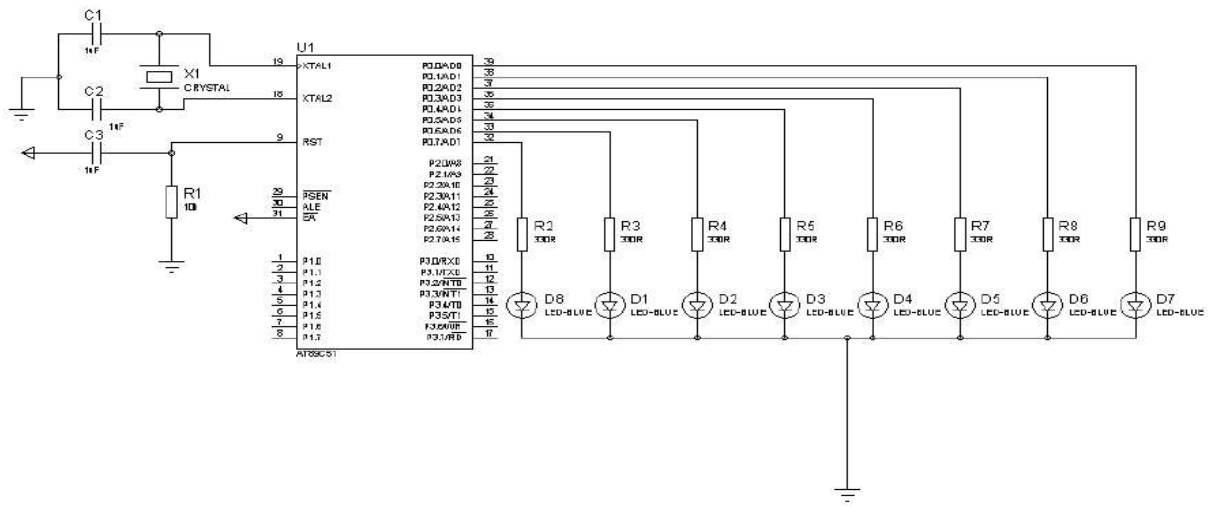


46. Aşağıdaki devrede bulunan ledleri Johnson sayıcı sıralamasıyla yakan programı yazınız.

Write the program which burn leds which is with ranking of counter of Johnson in the following circuit.

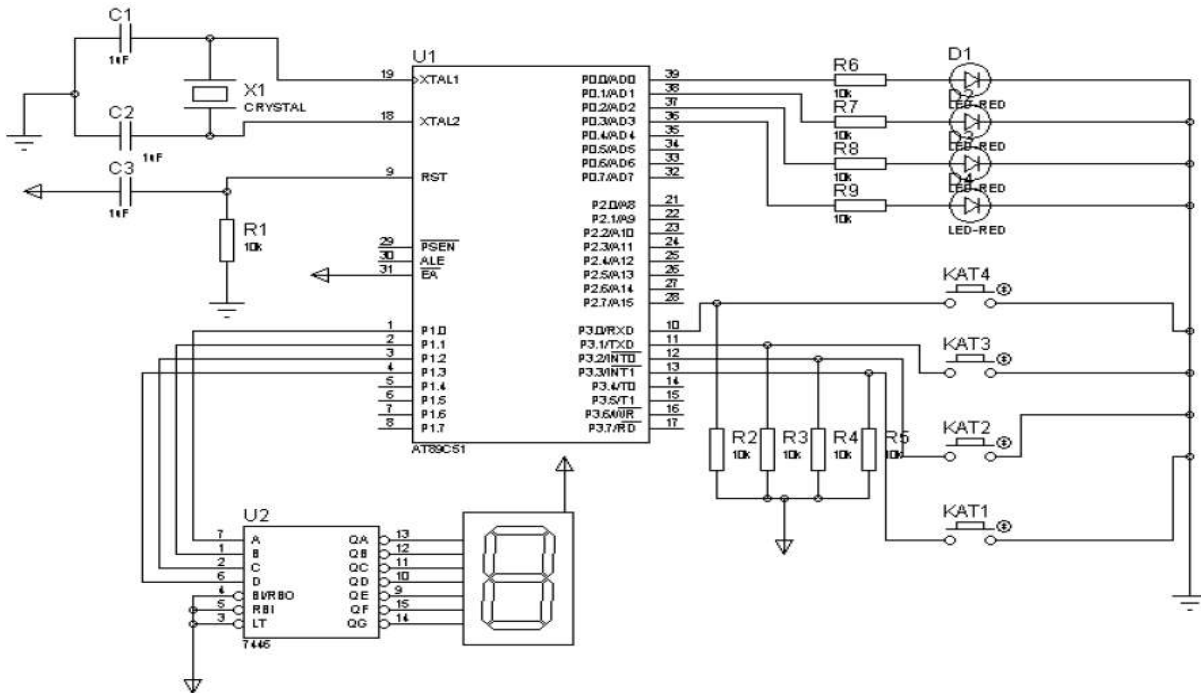
Johnson Sayıcı Tablosu:

Clock Pulse	Q ₀	Q ₁	Q ₃	Q ₄
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



47. Aşağıdaki asansör devresini çalıştıracak olan 8051 assembly program kodunu yazınız. Asansör hangi kattan çağrılırsa o kata kadar gidecek ve asansörün geçtiği kat numarası display ve ledler üzerinde görüntülenecektir. Her kat arası geçiş 1 saniye sürmektedir. (Kat numaraları yukarıdan aşağıya doğru azalmaktadır.)

Write the assembly program code 8051, which will run lift circuit which is in the following circuit. The lift will go up to that floor if it is called from the which floor and the floor number of the lift will be displayed on the display and leds. Switching between each floor takes 1 second. (Floor numbers are decreasing from top to bottom.)



48. Aşağıda bir hırsız alarm devresi görülmektedir. Eğer hareket sensörü bir hareket algıarsa 10 saniye saymaya başlayacaktır. 10 saniye bitene kadar durdur butonuna basılmazsa alarm çalacaktır. Alarm P2.5 portuna bağlı led olarak gösterilmiştir. Sensör P3.2, Durdur butonu P3.3 girişine bağlanmıştır. Programı kesme (interrupt) kullanarak gerçekleştiriniz.

In the following circuit, a burglar alarm circuit is seen. If the motion sensor detects a motion, it will start counting for 10 seconds. The alarm will sound if the stop button is not pressed until 10 seconds have elapsed. The alarm is shown as LED connected to P2.5 port. Sensor P3.2, Stop button is connected to input P3.3. Perform the program using interrupt.

Devre Şeması: