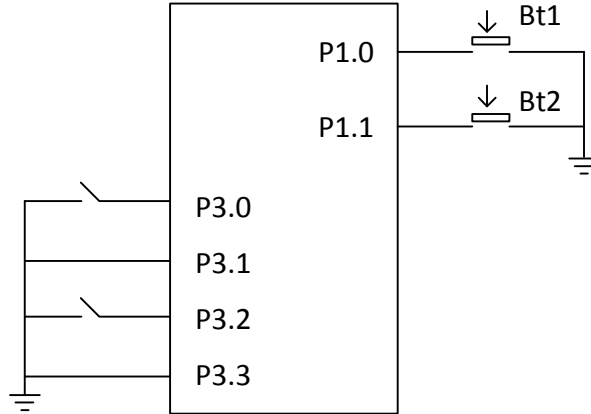


21. P3'te yüksek nibble'daki deęer bilinmiyor. P3'ün düşük nibble'da 10d deęeri vardır. Bt1 butonuna basıldığında p3'ten alınan düşük nibble deęerin (10d=0Ah) 2 katını, Bt2 butonuna basıldığında ise 3 katını alınız. Sonucu P2'da gözlemleyiniz.

Pinlerin kontrolünü (butonlara basılıp basılmadığını) jb veya jnb komutu kullanılarak 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 3 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, dięer 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. 22. Sorudaki programı, Keil programını adım adım çalıştırırken acall kodu çalışmadan önce döneceğiniz adresi belirleyiniz. Sp deęişimleri 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 acall code is not run)

24. Aşağıdaki programı yazınız. Veritabanı etiketinin program adresini gözlemleyiniz. Ayrıca veri tabanındaki deęerleri ve bunların program hafızadaki adreslerini gözlemleyiniz. Program adresi hem debug modunda dissassembly 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  ; ayarlari yapilir
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 sekilde daha rahat olusturulabilirdi.
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.** 24. 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.)

- a. 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.)

- b. 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.** 26. 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)

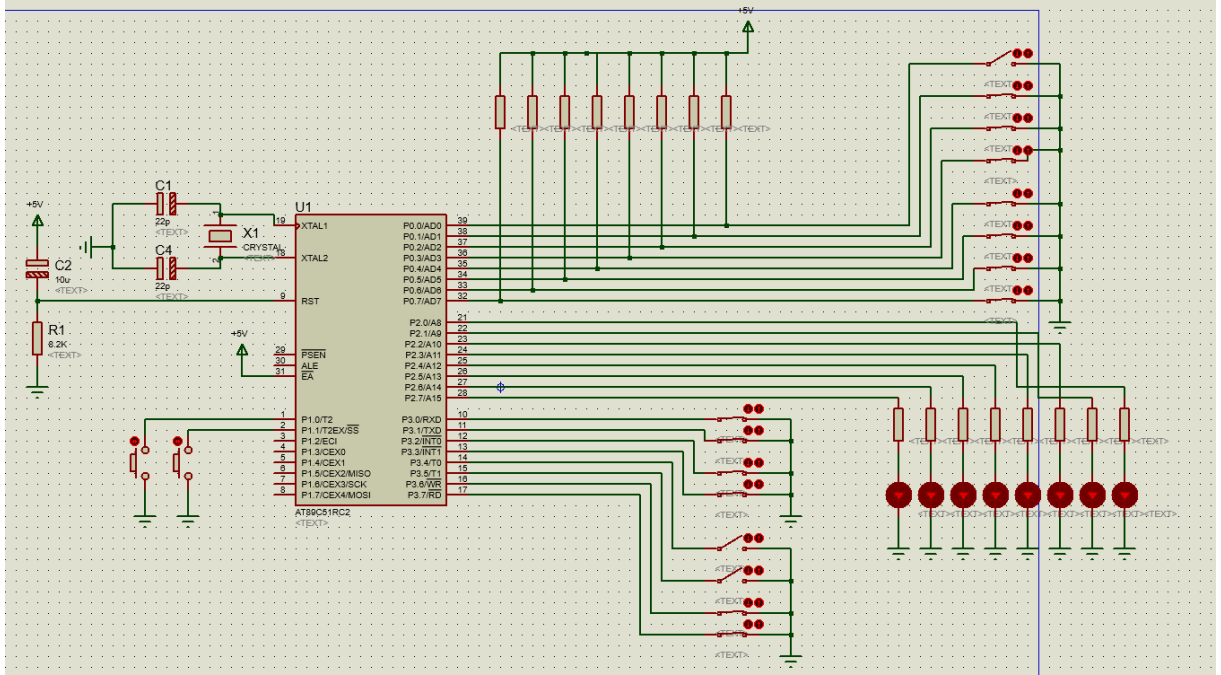
- 30.** 24. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri toplayalım ve sonucu p0'da gözlemleyelim.

(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.



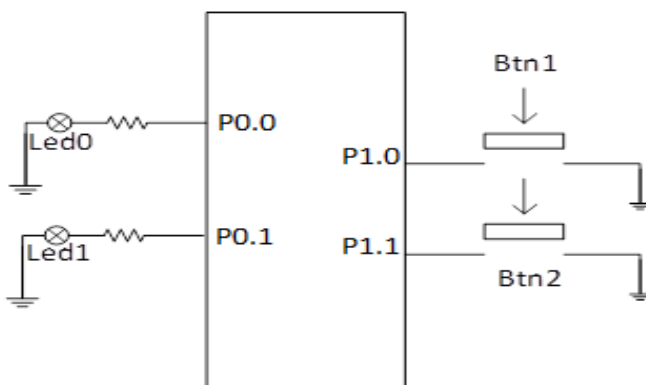
Yukarıda verilen devreyi kurunuz. P0 ve P3 değerleri, pinlere bağlı olan anahtarlardan sağlanacaktır.

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.)

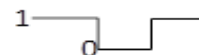
bekle: jb bt1,bekle

x: jnb bt1,x



Butonlara basıp elimizi çektiğimizde
Pinlerde oluşan lojik değişim

(The logic change on the pins when we
press the buttons and pull our hand)



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)

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

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.)

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

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 adress 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)	
Adres	Veri
49h	160d
48h	2d
47h	80d
46h	4d
45h	40d
44h	8d
43h	20d
42h	16d
41h	10d
40h	32d

Tablo2 (Algoritma ile otomatik olarak sıralanacak veriler)	
Adres	Veri
39h	2d
38h	4d
37h	8d
36h	10d
35h	16d
34h	20d
33h	32d
32h	40d
31h	80d
30h	160d