

Deneyin Amacı

8051 deney setini kullanarak 4'lü 7 Segment Display'de 9'dan 1'e kadar olan rakamların sırayla gösterilmesi.

Kullanılan Araçlar:

8051 Deney Seti
+5 V DC Güç Kaynağı

Deneyde Yapılanlar:

Keil uVision programında *Assembly* dili kullanılarak, bu mikro denetleyiciye uygun program yazıldı(yazılan kodlar diğer sayfada verilmiştir, display'lerden birisi arızalı olduğu için 3 tanesi kullanılabilmiş ve kod da ona göre yazılmıştır). 7 Segment Display'lerde gösterilecek karakter P2, hangi Display'in çalışacağı ise P3 portuna bağlıdır. P2 ve P3 portlarındaki verinin belirli aralıklarla değiştirilmesi ile rakamların sırayla gösterilmesi uygulamasının gerçekleşmesi sağlandı.

7 Segment Display'lerin yanış süreleri için bir gecikme fonksiyonu yazıldı. P2 ve P3'deki verinin her değişiminden sonra gecikme fonksiyonu ACALL komutu ile çağırıldı ve gecikme için gereken kodlar çalıştırıldıktan sonra RET komutu ile ana programa dönülerek bir sonraki rakamın, bir sonraki 7 Segment Display'de yakılması ve portlardaki güç kesilmediği sürece yanması için sonsuz döngüye sokulması sağlandı. *Keil uVision*'da yazılan programın, deney setine yüklenmek üzere *hex dosyası oluşturuldu*.

Deney seti, *RS232 seri portu* ile bilgisayara bağlandıktan sonra *Atmel – Flip* programı açılarak *AT89C51RC2* mikro denetleyicisi seçildi, deney seti *Program moduna* alınıp *Reset tuşuna* basıldıktan sonra *Atmel – Flip* programında *RS232* bağlantısı *COM1* portu üzerinden *115200 Baud rate* ile sağlandı. *Keil uVision* tarafından oluşturulan programımızın *hex hali*, *Atmel – Flip* programında seçildikten sonra *Operations Flow* bölümünden *Run* tuşuna basılarak yazdığımız program deney setine yüklenmiş oldu.

Deney seti üzerindeki *gereksiz switch'ler kapatılıp* P2 ve P3'e ait *Pull-up'lar* ile birlikte kullandığımız 7 Segment Display'lere ait *switchler* açıldı. *Mod* ayarı *Devre* olarak ayarlandı ve *Reset tuşuna* basıldı.

Deneyde Yapılanlar – Assembly Kodu:

```

    ORG 0000H
    SJMP ANA
    ORG 0030H
ANA:  MOV P2, #0F6H      /*9 karakteri*/
      MOV P3, #080H      /*Soldan 1. Display*/
      ACALL gecikme
      MOV P2, #0FEH      /*8 karakteri*/
      MOV P3, #020H      /*Soldan 3. Display, soldan 2. bozuk*/
      ACALL gecikme
      MOV P2, #0E0H      /*7 karakteri*/
      MOV P3, #010H      /*Soldan 4. Display */
      ACALL gecikme
      MOV P2, #0BEH      /*6 karakteri*/
      MOV P3, #080H      /*Soldan 1. Display */
      ACALL gecikme
      MOV P2, #0B6H      /*5 karakteri*/
      MOV P3, #020H      /*Soldan 3. Display, soldan 2. bozuk */
      ACALL gecikme
      MOV P2, #066H      /*4 karakteri*/
      MOV P3, #010H      /*Soldan 4. Display */
      ACALL gecikme
      MOV P2, #0F2H      /*3 karakteri*/
      MOV P3, #080H      /*Soldan 1. Display */
      ACALL gecikme
      MOV P2, #0DAH      /*2 karakteri*/
      MOV P3, #020H      /*Soldan 3. Display, soldan 2. bozuk */
      ACALL gecikme
      MOV P2, #060H      /*1 karakteri*/
      MOV P3, #010H      /*Soldan 4. Display */
      ACALL gecikme
      SJMP ANA
gecikme: MOV R6, #00H
tekrar: MOV R7, #00H
        DJNZ R7, $
        MOV R7, #00H
        DJNZ R7, $
        DJNZ R6, tekrar
        RET
        END

```

Deneyde Yapılanlar – Açıklamalar:

P3'e yüklenen verilere göre yanan 7 Segment Display'ler(soldan sağa) aşağıdaki gibidir:

1.Display: #080H	2.Display: #040H	3.Display: #020H	4.Display: #010H
------------------	------------------	------------------	------------------

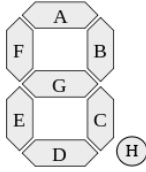
2. Display bozuk olduğundan, kodumuzda da görüldüğü gibi #040H değeri kullanılmadı.

7 Segment Display'ler yalnızca tek tek değil, aynı anda birden fazlası da yakılabilir. P3 portunun 4 biti (MSB) kullanılarak hangi Display'in aktifleştireceği seçilebilir. Örneğin P3'e 10100000, yani hexadecimal olarak #0A0 yollanırsa, soldan 1. ve 3. Display'ler aktifleştirecektir. Hangi Display'in yanacağını ilk 4 bite bakarak anlamamız mümkündür.

P2'ye yüklenen verilere göre 7 Segment Display'de görünecek rakamlar aşağıdaki gibidir:

1: #060H	2: #0DAH	3: #0F2H
4: #066H	5: #0B6H	6: #0BEH
7: #0E0H	8: #0FEH	9: #0F6H

Değerler şu mantıkla hesaplanır:



Bu deney setinde gönderilecek değerler ABCDEFGH sırasıyla gönderilir.

Örneğin 5 karakterini göndermek için AFGCD LED'lerinin yanması gerekir. AFGCD LED'lerine 1, diğerlerine 0 değeri verilip, hesaplanan 8 bitlik değer hexadecimal'e çevrilerek P2'ye gönderilir.

Sonuç ve Tartışma:

Assembly dili kullanarak 7 Segment Display uygulamaları yapıldı. 7 Segment Display'deki LED'lerin hangi mantıkla yakıldığı öğrenildi.

Deney setimizdeki P2 portunun 7 Segment Display'e giden veriyi aktarmak, P3 portunun ise hangi 7 Segment Display'i aktif olacağı verisini aktarmak için kullanıldığı öğrenildi.

R6 ve R7 kaydedicilerindeki veriler de döngüye sokularak gecikme olması sağlandı, böylece 7 Segment Display'lerin yanma süresi gözümüzün görebileceği kadar uzadı.