

**Deneyin Amacı**

8051 deney setini kullanarak Knight Rider'daki LED animasyonu ve 3'er 3'er atlayan animasyon gibi çeşitli kayan LED uygulamalarının yapılması.

**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). Yakılacak olan LED'ler P0 portuna bağlı olduğundan P0'daki verinin belirli aralıklarla değiştirilmesi ile kayan LED uygulamasının gerçekleşmesi sağlandı. LED'lerin yanış süreleri için bir gecikme fonksiyonu yazıldı ve P0'daki 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 LED'in yakılması ve P0'daki 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 P0'a ait Pull-up'lar* ile birlikte kullandığımız *LED'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 P0, #01H
     ACALL gecikme
     MOV P0, #02H
     ACALL gecikme
     ....
     MOV P0, #80H
     ACALL gecikme
     SJMP ANA
gecikme:
     MOV R6, #00H
tekrar: MOV R7, #00H
        DJNZ R7, $
        MOV R7, #00H
        DJNZ R7, $
        DJNZ R6, tekrar
        RET
        END
```

P0'a yüklenen verilere göre yanan LED'ler aşağıdaki gibidir:

1.LED: #01H	2.LED: #02H	3.LED: #04H	4. LED: #08H
5.LED: #10H	6.LED: #20H	7.LED: #40H	8.LED: #80H

Knight Rider'daki gibi kayan LED uygulaması için sırasıyla;

"#01H, #02H, #04H, #08H, #10H, #20H, #40H, #80H, #40H, #20H, #10H, #08H, #04H, #02H" verileri, aralarına gecikme fonksiyonu çağırılarak P0'a yüklenmiştir.

3'er 3'er atlayan kayan LED uygulaması için sırasıyla:

"#01H, #08H, #40H, #80H, #10H, #02H" verileri, aralarına gecikme fonksiyonu çağırılarak P0'a yüklenmiştir.

### Sonuç ve Tartışma:

Assembly dili kullanarak kayan LED uygulamaları yapıldı. P0 portuna sırasıyla veriler gönderilerek LED'lerin istenen sırayla yanması sağlandı. R6 ve R7 kaydedicilerindeki veriler de döngüye sokularak gecikme olması sağlandı, böylece LED'lerin yanma süresi gözümüzün görebileceği kadar uzadı.