



**KARADENİZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR ORGANİZASYONU
LABORATUVARI**



SERİ GİRİŞ / ÇIKIŞ (SERIAL I/O)

1. GİRİŞ

Bilgisayarların dış dünya ile haberleşmeleri kullanıcıya veya kullanım ihtiyacına göre seri veya paralel olarak gerçekleştirilebilir. Paralel giriş / çıkışta 8 bit data, 1 bit data hazır, 1 bit data istek, 1 bit GND olmak üzere 11 tel ile iletişim gerçekleştirilir. Bu haberleşme yöntemi hızlıdır. Ancak uzak mesafelerle yapılan haberleşmede maliyet çok fazladır. Seri haberleşmede biri veri, diğeri GND olmak üzere iki tel yeterli olduğundan dolayı seri haberleşmeye geçilmiştir.

Ayrıca haberleşmede gürültüden kurtulma çok önemlidir. Paralel haberleşmede lojik-0 için 0V, lojik-1 için 5V kullanılmaktadır. Burada 1-2V arası belli bir değere karşı düşmemektedir. Ancak gürültü bu değeri aşabilir. Seri haberleşmede ise lojik-1 (-3)-(-25)V, lojik-0 ise (3)-(25)V arasında kabul edildiğinden gürültülere daha dayanıklıdır. Yani $\pm 25V = 50V$ dalgalanma aralığı vardır.

Seri giriş/çıkış arayüzleri bilgisayarla paralel, dış dünya ile seri haberleşen elemanlardır. Bilgisayardan aldıkları paralel veriyi kendi içerisinde seriye çevirip dışarıya verirken, dışarıdan aldığı seri veriyi paralele çevirip bilgisayara verir.

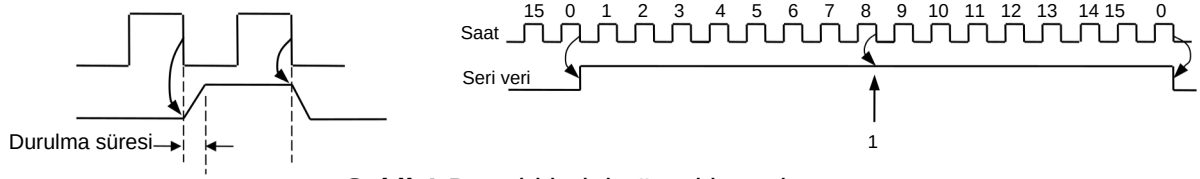
2. SERİ HABERLEŞME

2.1 Seri Haberleşmede Bit Örneklenmesi

Seri haberleşmede gönderilen data bitlerinin uygun anlarda örneklenerek algılanması zorunludur. Çünkü datanın hazır olduğunu gösteren herhangi bir işaret alıcı ile verici arasında gönderilmez. Gönderen saat işareti ile data hattına koyulan data bitinin istenilen seviyeye ulaşması için belirli bir durulma süresinin geçmesi gerekmektedir. Alıcı cihaz bu biti data hattını bu durulma süresi geçtikten sonra örnekleyerek almalıdır. Örneklemenin data bit süresinin ortasında yapılması en uygunudur ve bu amaçla alıcı tarafta bit ortasının bulunmasını sağlayan bir donanım kurulur. Orta noktanın bulunmasında kullanılacak saat işaretinin frekansı aşağıdaki bağıntı ile verilebilir:

$$\text{Saat işaretinin frekansı} = \text{Seri data hızı} \times K_s$$

K_s katsayısı bitin ortasının bulunması için 1, 16 veya 64 seçilebilir. Buna göre data kelimesinin örnekleme zamanlaması aşağıdaki gibi verilebilir.

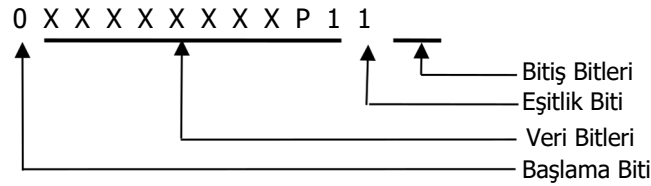


Şekil-1 Data bitlerinin örnekleme şekli

3. SERİ GİRİŞ/ÇIKIŞ PROTOLÜ

3.1 Asenkron Seri Haberleşme

Senkron ve asenkron olmak üzere iki seri haberleşme yöntemi vardır. Asenkron haberleşmede bir seri veri bit dizisinin başında ve sonunda datanın gönderilmeye başlandığını ve bittiğini belirten başlama ve bitiş bitleri mevcuttur. Buna göre, veri 0 başlama biti ile başlar ve 1, 1.5 veya 2 adet lojik-1 sonlandırma bitleri ile sonlanır. Bit dizisinde bu çerçevelendirme bitlerinin kullanılması bilgi akış hızını belirli oranda düşürmesine rağmen senkronizasyonu sağlamakta ve donanım gereksinimi belirli oranda azaltmaktadır. Veri yapısı içinde yer alan ve bilgi akışında güvenilirliği artıran eşitlik biti de bilgi akış hızını düşüren diğer bir etkidir. Buna göre 8-bitlik bir bilgi, başlama biti, eşitlik (parity) biti ve sonlandırma bitleri ile Şekil-2'deki gibi çerçevelenerek gönderilebilir.



Şekil-2 Asenkron seri veri biriminin biçimi

Bilgilerin içerebileceği bit sayısı 5-8 arasında değişirken, eşitlik biti yalnız bir tanedir.

3.2 Senkron Seri Haberleşme

Senkron haberleşme yönteminin gerçekleştirilmesi daha pahalı olmasına rağmen asenkron haberleşmeye göre daha hızlıdır. Başlama ve bitiş bitleri gerektirmeyen bu haberleşmede senkronizasyon, senkron karakter aracılığı ile gerçekleştirilir. İlk önce verici, senkronizasyon karakterini gönderir. Sync karakterini bulabilmek için seri G/Ç arayüz cihazının özel avlanma veya araştırma moduna sahip olması gerekir. Senkron haberleşmede, bir karakter 5-8 data biti ve bir de eşitlik bitinden oluşacak şekilde kurulan data kelimeleri ile gerçekleştirilmektedir.

Bitiş biti kullanılmadığından, senkronizasyonu korumak için genellikle bir saat işareti senkron veriye eşlik eder. Senkron seri veri, telefon hattından gönderileceği zaman, saat işareti için ayrı bir kanal sağlamak mümkün değildir. Bu durumda veri ve saati tek bir kanala kodlayacak özel bir senkron modem kullanılır.

Senkron veri dizisinin SYNC karakteri ile asenkron veri dizisinin çerçeveleme bitleri arasında bir benzerlik vardır. SYNC karakteri bir blok senkron veri karakterini çerçeveler. Başlama ve bitiş bitleri ise bir asenkron veri dizisinde her veri karakterini çerçeveler.

Genellikle aynı iki kavrammış gibi belirtilen baud rate ve bit rate aslında farklı kavramlardır. Baud rate adını Fransız mucit Emile Baudot'un isminin kısaltmasından almıştır. O zamanlarda kullanılan teleyazıcının (teleprinter) daha sonra telgraf (telegraph) olmuştur- herbiri 5 data ve 1 stop biti olmak üzere toplam 6 bitten oluşan mors kodu karakterlerini göndermek için kullandıkları "saat frekansı"na verilen isimdir. Fransız şehir şebekesi 50 Hz'lik alternatif akım kullandığı için o zamanlar telgraf için 50 baud'luk bir saat frekansı kullanılmıştır. Daha sonraları RS-232 seri haberleşme standardı ile diğer elektronik cihazlar iletişim kurmuşlardır. 1950'li yıllarda bilgisayarların yaygınlaşmaya başlaması ile modemler kullanılmıştır. Telgraftaki sistemin aksine – her saat darbesinde 1 bit – modemler sinyali çeşitli parametreleri kullanarak [gerilim, faz farkı vs.] işleyerek her saat darbesinde birden fazla bit gönderebilme özelliğine sahip cihazlardır. Bit rate (BPS-Bit Per Second) adından da anlaşılacağı gibi saniyede gönderilen bit sayısı ya da veri miktarını temsil etmektedir. Örneğin her saat darbesinde module işaret ile 8 bit iletildiği zaman Bit Rate = Baud Rate * 8 olmaktadır. Fakat modem bu işlemleri otomatik olarak gerçekleştirdiğinden saniyede alınan veri miktarı olan bit rate çoğu zaman baud rate olarak adlandırılmaktadır.

4. Deneye Hazırlık

Deneye gelmeden önce deney föyünün bir ön bilgi teşkil etmesi açısından okunması, deney süresinin verimli kullanılması açısından önemlidir. Ayrıca deney föyünden elde edilemeyen bilgiler çevrimiçi kaynaklar kullanılarak araştırılabilir. Deneye gelmeden önce hazırlık sorularının çözülmesi veya araştırılması gerekir. Bu soruları araştırırken öğreneceğiniz bilgiler deneyde anlatılanları pekiştirmenizi sağlayacaktır.

Bu deneyde, Tinkercad üzerinde iki Arduino kartı arasında seri haberleşme işlemi gerçekleştirilecektir. Bir Arduino "sender" (gönderici) olarak veri gönderecek, diğer Arduino ise "receiver" (alıcı) olarak bu veriyi alıp seri monitör üzerinden görüntüleyecektir.

Gerekli Adımlar

1. Tinkercad'e Giriş

- Tinkercad hesabınıza giriş yapın.
- Üst menüden "Devreler" seçeneğini seçin.
- "Yeni Devre Oluştur" butonuna tıklayın.

2. Arduino Kartlarını Ekleme

- Sağdaki bileşenler panelinden iki adet Arduino kartını devre alanına sürükleyin.
- Bir Arduino kartı "Sender", diğer Arduino kartı ise "Receiver" olarak adlandırılacaktır.

3. Kablolama İşlemi

- Sender Arduino'nun TX (transmit) pini, Receiver Arduino'nun RX (receive) pinine bağlanır.
- Receiver Arduino'nun TX pini, Sender Arduino'nun RX pinine bağlanır.
- Her iki Arduino'nun GND (toprak) pinleri bir kablo ile birbirine bağlanır.

Not: Kabloların doğru bağlandığından emin olun. Yanlış bağlantı veri aktarımını engeller.

Sender Arduino Kodu

```
char mystr[10] = "Hello"; // Gönderilecek veri
```

```
void setup() {  
  // Seri haberleşmeyi 9600 baud hızında başlat  
  Serial.begin(9600);  
}
```

```
void loop() {  
  Serial.write(mystr, 5); // Seri veri gönder  
  delay(1000); // 1 saniye bekle  
}
```

- **char mystr[10]:** Gönderilecek karakter dizisi ("Hello").
- **Serial.write(mystr, 5):** mystr dizisindeki ilk 5 karakteri seri porta gönderir.
- **delay(1000):** Her gönderim arasında 1 saniyelik gecikme oluşturur.

Receiver Arduino Kodu

char mystr[10]; // Gelen veriyi saklamak için değişken

```
void setup() {
  // Seri haberleşmeyi 9600 baud hızında başlat
  Serial.begin(9600);
}
```

```
void loop() {
  Serial.readBytes(mystr, 5); // Seri porttan veri oku ve değişkene ata
  Serial.println(mystr); // Alınan veriyi Seri Monitörde yazdır
  delay(1000); // 1 saniye bekle
}
```

Kodların Arduino Kartlarına Yüklenmesi

- Sender Arduino için yukarıdaki "Sender Arduino Kodu" eklenir.
- Receiver Arduino için yukarıdaki "Receiver Arduino Kodu" eklenir.
- Kodlar tamamlandıktan sonra "Başlat" butonuna tıklayın.

Test ve Çalışma

- Çalıştırma sonrası, Receiver Arduino'nun Seri Monitöründen alınan veriyi görebilirsiniz.
- Seri Monitör açıldığında her saniyede bir "Hello" yazısı görüntülenmelidir.
- Alıcı tarafta Baud Rate değerini değiştirip çıktıdaki değişimi gözlemleyiniz.

5. Deney Tasarımı ve Uygulanması

Deneyden önce öğrencilerin deneye ne kadar hazır olduklarını ölçme bakımından hazırlık soruları sorulabilir. **Deneye hazırlık aşamasında yapılması gerekenlerin dikkatlice okuyup uygulaması önerilir.**

6. Deneye Hazırlık Soruları

1. Haberleşmede kullanılan giriş/çıkış arayüzleri nelerdir? Aralarındaki temel farkları vererek açıklayınız.
2. Seri haberleşmenin kullanım alanları nelerdir? Örnekler vererek açıklayınız.
3. RS232 standardı nedir?
4. Baud Hızı nedir?
5. Asenkron seri haberleşme nasıl gerçekleşmektedir?
6. Senkron seri haberleşme nasıl gerçekleşmektedir?

7. Deney Raporu

Deneyde anlatılan kısımların pekiştirilmesi ve kalıcı bilgi haline getirilmesi açısından tekrar edilmelidirler. Deney raporu, **kısmen** deney föyündeki bilgilerden yararlanılarak hazırlanabilir. Deney raporunun alınma amacı öğrencinin gerekli kazanımları alıp almadığının görülmesidir. Rapor 1 hafta içinde deney sorumlusuna uzam veya deney sorumlusunun belirttiği platform üzerinden teslim edilir.