**Paralel Haberleşme:** 8 biti ayrı ayrı kanallardan aktarabiliriz. Herhangi bir anda 8 biti gönderebiliriz.

**Seri Haberleşme:** 8 biti arka arkaya tek bir hattan göndeririz. Burada 8 biti arka arkaya gönderdiğimiz için biraz zaman ihtiyaç duyarız. Hattan tasarruf, zamandan kayıp.

**Senkron ve Asenkron Haberleşme:**

Senkron haberleşmede veri ile birlikte bir de saat (clock) sinyali kullanılır. Veri saat sinyali ile eş zamanlı olarak iletilir.

Asenkron iletişimde ise saat sinyali yoktur. (İki tarafında hızı aynı olarak çalışmalı)

***UART (SERİ HABERLEŞME) (Universal Asynchronous Receiver Transmitter) Nedir ?***

* Evrensel bir eşzamansız alıcı-verici, veri biçiminin ve iletim hızlarının yapılandırılabilir olduğu eşzamansız **seri iletişim için bir bilgisayar donanım aygıtıdır**.
* UART'ın temel işlevi, **bilgisayarlar, mikrodenetleyiciler, sensörler ve diğer cihazlar** arasında veri iletişimi sağlamaktır.
* UART, **evrensel asenkron alıcı / verici** anlamına gelir ve iki cihaz arasında **seri veri alışverişi** için  basit, iki telli bir protokoldür.
* UART çok basittir ve her iki yönde de iletmek ve almak için verici ve alıcı arasında yalnızca iki kablo kullanır.
* Her iki uçta da **toprak bağlantısı** vardır.
* UART’ta iletişim **tek yönlü** (veriler yalnızca bir yönde gönderilir), **yarı çift yönlü** (her iki taraf konuşur ancak bir seferde yalnızca bir tane) veya **tam çift yönlü** (her iki taraf aynı anda iletebilir) olabilir.
* UART’taki veriler **çerçeveler (frame)** şeklinde iletilir.
* UART’ın en büyük avantajlarından biri **asenkron** olmasıdır – verici ve alıcı ortak bir saat sinyalini paylaşmaz. Bu, protokolü büyük ölçüde basitleştirse de, verici ve alıcıya belirli gereksinimler getirir. Bir saati paylaşmadıklarından, aynı bit zamanlamasına sahip olmak için her iki uç da aynı, önceden ayarlanmış hızda iletmelidir.
* Nerdeyse tüm mikrodenetleyicilerde bulunan bir birimdir.

**Başlangıç ve bitiş bitleri**, kullanıcı verilerinin nerede başladığını ve bittiğini belirtmek veya verileri “*çerçevelemek*” için kullanılır.

**Eşlik Biti (Parite Biti)**

* Tek bit hatalarını algılamak için isteğe bağlı bir **eşlik biti (parite biti)** kullanılabilir. Veri bütünlüğünü korumak için kullanılır.
* Eşlik biti kullanılarak **çok güvenli olmayan hata denetimleri** yapılabilir. Çok güvenli olmamasının sebebi iki bit birden bozulmaya uğrarsa bunu tespit edemez.

UART hala yaygın olarak kullanılan bir seri veri protokolüdür ancak son yıllarda bazı uygulamalarda yerini **SPI, IC, USB ve Ethernet** gibi teknolojiler almıştır.

**Asenkron**, paylaşılan saat olmadığı anlamına gelir, bu nedenle UART’ın çalışması için bağlantının her iki tarafında aynı bit veya baud hızı yapılandırılmalıdır.

**Baud Rate:** UART ile iletilen verilen hızı bu şekilde adlandırılır. Saniyede iletilen bit sayısını ifade eder. (Genellikle 300-115200 bps 300’ün katları arasında değişir)

**Tam ve Yarı İletim:** UART, tam çift yönlü veya yarı çift yönlü iletişim için kullanılabilir. Tam çift yönlü iletişimde hem veri gönderilebilir hem de alınabilirken, yarı çift yönlü iletişimde sadece bir cihaz veri gönderebilir ya da alabilir.

**UART Kullanım Alanları:**

* Seri veri iletişimi gerektiren mikrodenetleyiciler ve sensörler.
* Bilgisayarlarla ve diğer cihazlarla seri port üzerinden iletişim.
* GPS alıcıları, bluetooth modülleri, Wi-Fi modülleri gibi kablosuz iletişim cihazları.
* Otomotiv sistemlerde, telemetri uygulamalarında ve daha birçok endüstriyel uygulamada.

Son zamanlarda UART kullanımının popülaritesi azalmaya başladı. Bunun yerine; **SPI** ve de **I2C** gibi protokoller kullanılmaya başlanmıştır.

Belirli bir seri bağlantı noktası üzerinden iletişim kurmanın yerine, birçok modernleşmiş bilgisayar ve çevre birimi artık Ethernet ve USB gibi teknolojileri kullanmaya başlamıştır. Aynı zamanda UART, çok basit, düşük maliyette ve uygulanması da kolay olduğundan hala **çok daha düşük hızlı ve çok daha düşük verimli uygulamalar için** kullanılmaktadır.

**UART Haberleşme Protokolleri Çalışma Prensibi**

metin, diyagram, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Genel olarak, UART tam çift yönlü haberleşme modu şeklinde çalışmaktadır. Kısacası, aynı anda veri gönderebilir ya da alabilir. USART, 5 ve 9 bit arasında olan data uzunluğuna sahip verileri taşıma özelliğine sahip olmaktadır. Fakat genelde 8 ya da 9 bitlik kullanımlar tercih edilmektedir.**

**UART, saatleri kullanmamaktadır.**

**ekran görüntüsü, metin, dikdörtgen, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**UART birimlerinde RX, TX ve GND uçları bulunur.**

**TX: Veri iletim hattı.**

**RX: Veri alma hattı.**

**GND: Toprak bağlantısı. Verinin düzgün bir şekilde iletilmesini ve cihazlar arasındaki güç ve sinyallerin kararlı kalmasını sağlar.**

**Haberleşecek birimlerin RX ve TX uçları çapraz olarak bağlanır. GND bağlantıları ortaklanır.**

**Full Duplex: İki tarafın da aynı anda konuşabileceği haberleşme yapısıdır. İki tarafta veri iletebilir ve veri alabilir.**

**Half Duplex: Telsiz mantığı. Biri konuşur diğeri dinler. Aynı anda gerçekleşmez.**

**Simplex: Biri konuşur diğeri dinler.**

**USART** **(Universal Synchronous Asynchronous Receiver Transmitter)**

* **USART seri iletişim kurmak için mikrodenetleyiciler içerisinde bulunan çevre biriminin adıdır.**
* **USART** ise hem senkron hem de asenkron olarak çalışabilir.
* UART’a göre daha gelişmiş bir protokoldür.
* Haberleşme mantığı aynı şekilde çalışır ancak USART aynı zamanda **senkron haberleşmeleri de gerçekleştirebilir.**
* Yeni çıkan bir mikroişlemcinin datasheet’ine (elektronik ve teknolojik ürünlerin teknik özellikleri, ayrıntıları vs.) baktığınız zaman bu birimleri genelde USART birimi olarak görüyoruz çünkü USART aynı zamanda UART’ı da kapsayan bir birim olarak tasarlanmıştır.
* RS232 ise fiziki bir katmandır.
* Genellikle işlemcinin UART çıkışları 3.3V veya 5V seviyelerindedir.
* Uzun mesafelerde elektriksel olarak daha iyi bir şekilde iletebilmek için RS232 çeviriciler kulanılır.
* İki işlemciyi USART üzerinden haberleştirebiliriz.

**UART ve USART Nasıl Çalışır ?**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Haberleşme bir start bitiyle başlar.

Start biti 🡺 1 bit

Stop biti 🡺 1 veya 2 bit

Veri gönderdiğimiz yapı bir byte olmak zorunda değil. 5,6,7,8 veya 9 bit olabilir. Ama en çok kullanılan 8 bitlik yapıdır.

Pariti biti 🡺 Hata kontrolünde kullanılır.

Bir bit start, bir bit stop sayarsak **bir byte göndermek** için aslında 10 bit gönderilir.

**UART ve USART Haberleşme Protokollerinin Avantajları**

– Genelde tüm yapılan iletişim işlevlerinde **saat sinyali gerekli olmamaktadır.**

– Kullanımı oldukça **uygun maliyetli** olmaktadır.

– Veri iletişimi konusunda **2 kablo yeterli** olmaktadır.

– En çok kullanım sağlanan **seri standartlar** olmaktadır.

– Genelde eşlik bitleri kullanımı sağlanarak, **hata denetimi** de yapılmaktadır.

– Kullanımında **değişken özellikte veri paketi** yer almaktadır. Veri paket yapısı da **her iki taraf da aynı ayarlandığı sürece** değişiklik yapılabilir.

– Çok iyi belgelenmiş olan UART ve USART haberleşme protokolleri **çok sık kullanılmaktadır.**

**UART ve USART Haberleşme Protokollerinin Dezavantajları**

– Genelde birden daha çok *Master Slave* işlevlerini **desteklemez**.

– UART haberleşmesinde yer alan **baud hızı**, cihazların birbiri içinde **aynı** olmak zorundadır.

– Veri aktarım hızları daha önce sabitlenmektedir ve diğerinden de farklı olamaz.

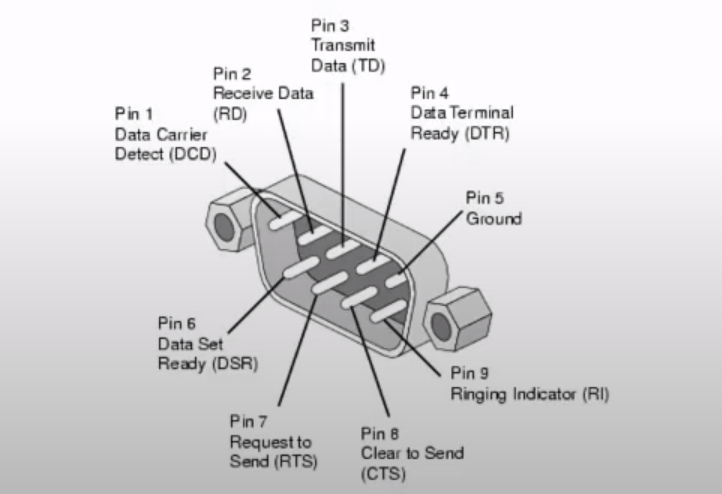
– Veri çerçevesi boyutları maksimum ölçülerde **9 bit ile sınırlı** olmaktadır.

– *Birden daha çok* ana sistemi de **desteklemez**.

**Master Slave:** Master-Slave, birincil cihazın kontrolü altındaki ikincil cihazları yönlendirerek koordinasyon sağlar.

**RS232**

Geçmişte hemen hemen bütün PC’lerde bulunan bir seri iletişim fiziksel standardıdır. Noktadan noktaya iletişimde en yaygın kullanılan protokollerden birisi idi.



**Bu port yokken nasıl mikrodenetleyicilerle haberleşebiliriz ?**

Uart – USB Converter’lar kullanabiliriz. Bunları kullandığımızda bilgisayarda sanal comport oluşur ve biz bu şekilde mikrodenetleyicilerle iletişim kurabiliriz.