

# Projeto de Sistemas RF

## Aula 04 - RS232 e Comunicação Serial

# Apresentação

Na aula passada, nós vimos um pouco sobre os dispositivos de comunicação e sobre alguns protocolos de comunicação. Na aula de hoje, vamos falar de transmissões seriais e do protocolo de comunicação RS-232. Vamos descer até o nível dos fios e visualizar como fazemos a transmissão de dados serialmente entre dispositivos, além de entender como utilizar o protocolo USART para comunicar um microcontrolador com um dispositivo de comunicação.

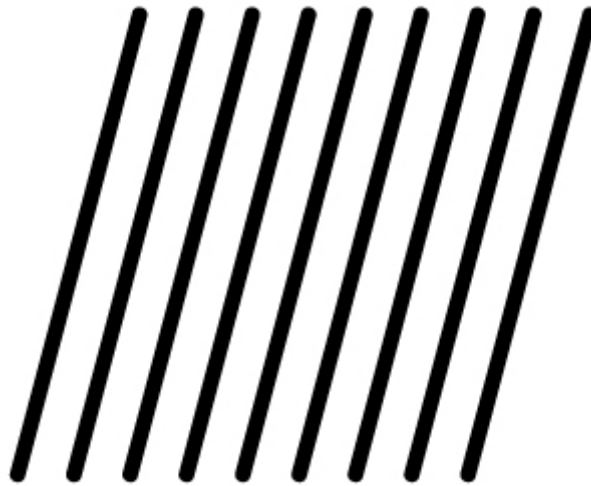
## Objetivos

Ao final desta aula, você deverá ser capaz de:

- Entender os conceitos de transmissão paralela e serial.
- Entender os conceitos de transmissão serial síncrona e assíncrona.
- Entender como o protocolo USART pode ser utilizado na transmissão de dados sem fio.
- Entender a diferença da USART para o RS-232.

# Transmissão Paralela

Vamos começar falando de transmissões paralelas. Você tem ideia do que seja isso? Ao ler a palavra "paralela", o que passa pela sua cabeça? Ela me lembra matemática... geometria... retas paralelas! Você lembra o que são as retas paralelas? São aquelas retas que nunca se cruzam, como você pode ver na **Figura 1**.



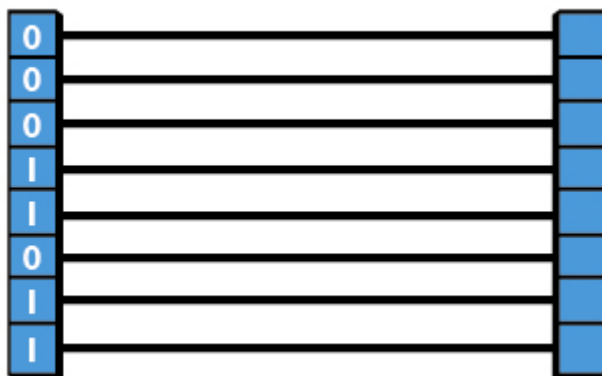
**Figura 1** - Retas paralelas.

Agora, imagine que você tem um dispositivo que consegue medir a temperatura de uma sala. Você deseja transmitir o valor medido pelo dispositivo para um computador e assim poder montar um bonito gráfico da variação da temperatura na sala durante um dia. A temperatura fica armazenada no dispositivo em formato binário. Se ele usa 8 bits (1 Byte) para armazenar o valor e a temperatura atual é 27 °C, a memória do dispositivo irá armazenar o valor binário:

$$27_{10} = 00011011_2$$

Agora, temos que transferir o dado binário para o computador. Imagine que temos oito perninhas no dispositivo e oito entradas no computador. Se conectarmos cada uma das pernas com uma das entradas do computador usando oito fios, podemos transferir todo o

dado de uma vez só! Seria preciso apenas definir um nível de tensão para representar o bit "1", por exemplo 5V, e outro para representar o bit "0", que pode ser por exemplo 0V. Você pode ver isso na animação abaixo.



**Figura 2** - Transmissão paralela.

O que você acabou de ver é um exemplo de uma comunicação paralela! Você consegue perceber uma relação entre a **Figura 2** e a **Figura 1**? Note que os fios estão todos paralelos um ao outro! Cada um é utilizado para transmitir uma parte do dado de forma independente (ou seja, transmitir um bit), e não há cruzamento de informação entre eles. Computadores antigos costumavam ter uma "porta paralela", que continha muitos pinos, usados geralmente para conexão com impressoras. O exemplo da **Figura 2** é uma versão simplificada do que acontecia nessa tal "porta paralela": um conjunto de bits sendo transmitido de uma só vez para o outro lado.

## Transmissão Serial

Agora, imagine se você só tivesse um fio disponível para conectar o dispositivo e o computador. Como você realizaria a transmissão dos dados? É aí que a transmissão serial é utilizada. A palavra "serial" lembra alguma coisa em série... como uma série de televisão, em que uma história é contada em capítulos, os quais são exibidos em ordem. Para entender bem a história, você precisa assistir aos episódios na ordem de exibição. Da mesma forma, é a transmissão serial. Nela, o dado é transmitido bit a bit pelo mesmo fio, sendo montado no lado do receptor.

Assim que o receptor tem um dado completo, ele passa o dado para o computador e fica pronto para receber outro do transmissor. A animação a seguir ilustra bem essa ideia.



**Figura 3** - Transmissão serial.

É assim que uma transmissão serial funciona. Ao longo dos anos, as transmissões seriais foram ficando mais populares do que as paralelas, devido ao custo mais barato (veja que ela usa um número bem menor de fios!) e à alta taxa de transmissão que atingem, mesmo enviando os dados bit-a-bit. Com o tempo, as portas paralelas foram sumindo dos computadores. Hoje em dia, a maioria dos dispositivos utiliza a tecnologia USB, que também se baseia em comunicação de dados no formato serial, mas adiciona várias outras funcionalidades que não existiam em protocolos seriais mais antigos, como o RS-232. Porém, esses protocolos ainda são muito utilizados na indústria para a comunicação simples entre dispositivos, como por exemplo microcontroladores e dispositivos transmissores RF.

## Se liga!

É importante prestar atenção na ordem de envio dos bits na porta serial para que eles sejam montados da forma correta do outro lado da transmissão. Por exemplo, se o dado 0001 1011 que representa os 27 °C medidos fosse enviado a partir do bit mais à direita para o mais à esquerda, porém montado no receptor do bit mais à esquerda para o mais à direita, teríamos o dado 1101 1000 no computador, o que significaria uma temperatura de 216 °C!

## Detalhes da Comunicação Serial

Já vimos que, na comunicação serial, os dados são enviados em uma sequência de bits por um mesmo meio, que, no nosso exemplo, era um fio. Imagine que você agora tem superpoderes e é capaz de ver a energia sendo aplicada ao fio. Vamos supor que a tensão de 5V representa um bit 1, enquanto 0V representa o bit 0. Os mesmos 27 °C seriam vistos por seus olhos da seguinte forma:



### Vídeo 01 - Comunicação Serial

Interessante, não? Utilizando dois valores distintos de tensão podemos codificar qualquer sequência de bits que desejarmos. Mas, note que temos um problema aí. Se eliminarmos o quadro com os bits do vídeo anterior, você vai sentir dificuldade em definir quantos "0"s ou "1"s estão sendo transmitidos em sequência. Ou seja, você não vai estar sincronizado com a transmissão dos dados, pois você não sabe quando começa a transmissão e nem a velocidade na qual eles estão sendo transmitidos.

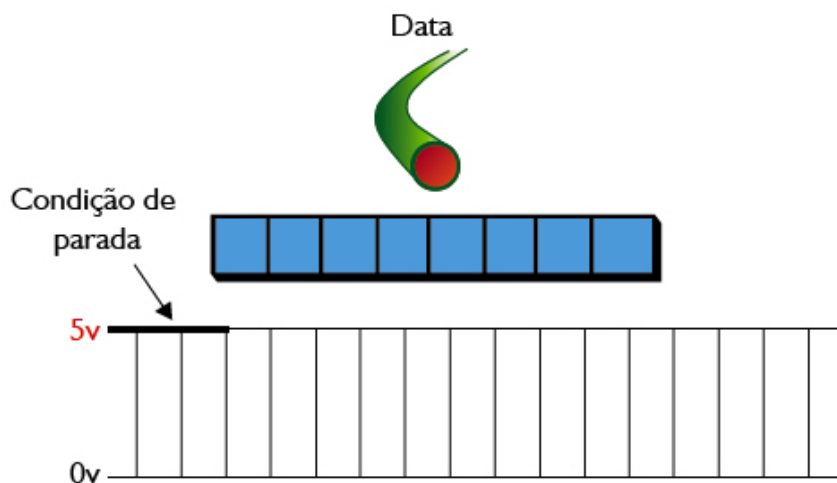
Vamos supor que você tem também acesso a um outro fio que você pode consultar para saber qual a velocidade de transmissão dos bits no primeiro fio. Cada vez que eu aplicar 5V nele, significa que tem um novo bit para ser lido no fio dos dados. Observe o mesmo exemplo, mas agora com o fio auxiliar mostrando diversas velocidades.



### Vídeo 02 - Comunicação Serial Síncrona

## Transmissão Síncrona e Assíncrona

Você entendeu bem como é feita a transmissão utilizando dois fios? Esse tipo de transmissão é chamada de "transmissão síncrona". O fio por onde passam os bits é chamado de "fio de dados", ou "DATA", em inglês; e o outro fio é chamado de "temporizador", mas é bem mais conhecido pelo nome em inglês "CLOCK". Essa transmissão é denominada síncrona porque os bits estão sendo transmitidos pelo cabo de dados em sincronia com o CLOCK. Porém, essa não é a única forma de se transmitir os dados serialmente. Existe outra forma de transmitir os dados sem precisar de um CLOCK. Para isso, é necessário alguma maneira de informar o lado receptor que um dado vai ser transmitido, e outra para informar que a transmissão terminou. Observe a imagem a seguir:



**Figura 6** - Transmissão serial assíncrona.

Nessa imagem, mais uma vez a temperatura de 27 °C é transmitida, mas observe que adicionamos dois bits ao pacote: um para informar o receptor que estamos prestes a enviar um dado, e outro para informar que o dado terminou. Eles são chamados de "bit de início" ou "START BIT" e "bit de término" ou "STOP BIT". O START BIT sempre tem o valor lógico "0", e o STOP BIT o valor lógico "1". Quando não desejamos transmitir, mantemos o valor lógico de "1" na linha, que é conhecido como "condição de parada" ou "STOP CONDITION". Assim que quisermos enviar mais um dado, transmitimos um bit "0" para alertar o receptor e, aí sim, começamos a transmitir o dado. No fim, mandamos um bit "1" para dizer que terminamos. E se quisermos transmitir mais um dado logo depois? Simples. Só transmitir de novo o "START BIT" e mandar outro dado. Entendeu? Observe o gráfico na atividade e tente extrair os Bytes sendo transmitidos de forma assíncrona.

## Se liga!

A "condição de parada" na transmissão serial assíncrona não é definida como "1" à toa. Quando o sistema de telefonia surgiu, percebeu-se que mantendo a condição de parada como nível lógico alto era a forma mais fácil de detectar falhas na linha: se não existisse tensão, significava que algo estava errado! É por isso que os telefones fixos fazem um "Tuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu" quando tiramos do gancho antes de fazer uma ligação: ele está funcionando quando "dá linha"! E da mesma forma, quando ele está "mudo", você sabe que não vai conseguir fazer uma ligação porque ele não está funcionando.

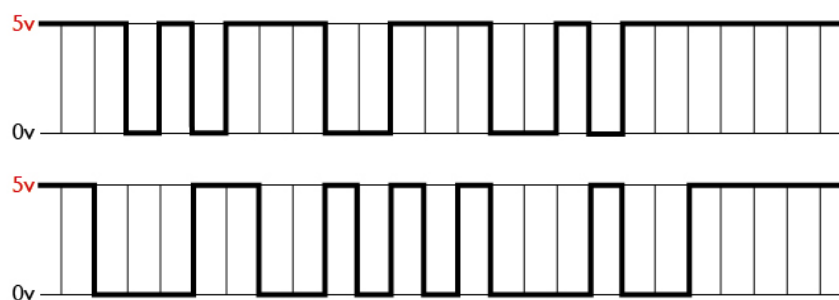


## Se liga!

Até agora só falamos da transmissão de pacotes de 8 bits de tamanho. O problema é que com 8 bits, o maior número decimal que podemos transmitir é 255. E se quisermos transmitir algo maior, como 2012, por exemplo? Nesse caso, temos que dividir o dado em conjunto de 8 bits e transmitir o dado parte por parte, e, então, remontar do outro lado. Exemplo: o número 2012 em binário é "11111011100". A gente pode dividi-lo em 2 Bytes assim: "00000111 11011100". Se transmitirmos primeiro "11011100" e depois "00000111", podemos remontar o dado do lado do receptor e teremos novamente o dado 2012 em decimal. Mas, atenção! Os dados têm que ser montados da mesma forma do lado do receptor! Por exemplo, se montássemos os Bytes que representam 2012 de forma invertida, iria resultar no valor decimal 56327!

## Atividade 01

1. Na figura abaixo, 4 Bytes estão sendo transmitidos de forma assíncrona. Descubra quais são esses Bytes. Marque os "bits de início", "bits de término" e os períodos em que não está sendo transmitido nada na linha.



**Figura 7** - Transmissão serial assíncrona: Exemplo com 4 Bytes.

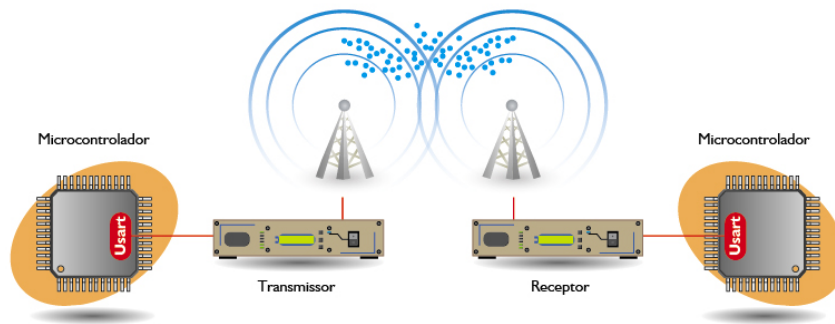
# Protocolos UART e RS-232

Vimos até então como transmitir informação serialmente de forma síncrona e assíncrona. É importante entender a diferença: na síncrona, o envio dos bits é controlado pelo sinal de clock, enquanto que na assíncrona são definidos bits para informar o início e o fim de transmissão. Além disso, em ambas as formas, tanto o transmissor quanto receptor precisam estar configurados para funcionar na mesma velocidade e com um mesmo tamanho de dados (geralmente 1 Byte).

## Se liga!

As transmissões síncronas necessitam de mais hardware do que as assíncronas, pois precisam de um gerador de clock e uma linha a mais de transmissão. Por outro lado, em uma transmissão assíncrona, são enviados mais bits do que o necessário para transmitir um dado, devido ao START BIT e o STOP BIT, gerando uma queda na quantidade de bits que realmente contém informação. Esse efeito é chamado de *overhead*.

As transmissões assíncronas que você estudou são geradas por um dispositivo chamado de UART — *Universal Asynchronous Receiver/Transmitter*. As transmissões síncronas são geradas pelo dispositivo chamado USART — *Universal Synchronous/Asynchronous Receiver/Transmitter*. Eles são os responsáveis por transformar um dado paralelo em serial. A maioria dos microcontroladores tem uma forma de comunicação UART ou USART, e você só precisa habilitar o uso das portas, realizar algumas configurações de funcionamento e conectar o microcontrolador com o dispositivo a se comunicar. Podemos, por exemplo, conectar um transmissor de rádio frequência em um micro através da UART, e no outro lado conectar o receptor a outro micro também através de USART, criando assim uma forma de transmitir dados por um meio sem fio.



**Figura 8** - Transmitindo os dados por um meio sem fio.

A UART e a USART utilizam as tensões padrões de circuitos digitais: 5V para nível lógico alto e 0V para nível lógico baixo. Mas, existem outros padrões. O padrão RS-232, por exemplo, foi usado por muito tempo como padrão de comunicação entre terminais de computadores e alguns periféricos, como teclados e mouses. Ainda hoje, é utilizado em comunicação industrial. Ele define algumas coisas, como os níveis de tensão (-12V para o nível lógico "1" e +12V para o nível lógico "0"), formato dos conectores e outros sinais usados para fazer o controle de dados. Nós não vamos utilizar diretamente o RS-232 na disciplina, mas é importante saber o que ele é. As comunicações em RS-232 geralmente utilizam UART ou USART para transformar os dados de paralelo para serial antes de transmitir no formato RS-232, e novamente de serial para paralelo na ponta do receptor.

## Se liga!

Os conceitos de UART/USART e RS-232 são frequentemente confundidos, pois ambos estão relacionados a transmissões seriais. A diferença principal é que UART/USART tratam da conversão de paralelo para serial e transmissão serial desses dados, usando níveis lógicos padrões dos circuitos digitais. Já o RS-232 é um padrão voltado para a indústria, definindo níveis lógicos que são melhores para transmissão em ambientes com muito ruído, além dos formatos dos conectores e sinais para controlar o fluxo de dados entre dispositivos. Isso é importante, pois assim vários dispositivos de diferentes marcas e modelos podem ser utilizados em conjunto em um mesmo local, desde que todos saibam como se comunicar pelo protocolo RS-232.

# Leitura Complementar

- Neste link, existe bastante informação sobre comunicação serial e protocolos.  
<[http://pt.wikipedia.org/wiki/Comunica%C3%A7%C3%A3o\\_serial](http://pt.wikipedia.org/wiki/Comunica%C3%A7%C3%A3o_serial)>
- Este link possui definições e explicações mais detalhadas dos sistemas de USART.  
<<http://en.wikipedia.org/wiki/USART>>
- Este link possui informação detalhada sobre o protocolo RS-232 e exemplos de frames.  
<<http://pt.wikipedia.org/wiki/RS-232>>

## Resumo

Vimos, nesta aula, dois tipos diferentes de transmissões seriais: a síncrona e a assíncrona. Vimos que as comunicações seriais são mais utilizadas do que as paralelas e podem ser utilizadas em comunicações sem fio. Para transmitir serialmente, é preciso que as duas pontas da transmissão estejam sincronizadas. Isso pode ser feito através de um fio de CLOCK, no caso síncrono, e através de bits de início e término de transmissão, no caso assíncrono, desde que ambas as pontas estejam configuradas para uma mesma velocidade. Por fim, vimos que o RS-232 é apenas um padrão de comunicação que define valores de tensões para os níveis lógicos 0 e 1, formato dos conectores, além de alguns outros sinais para controlar o fluxo de dados.

## Autoavaliação

1. Qual a diferença entre comunicação paralela e serial? Quais as vantagens e desvantagens de cada uma? Qual é a mais utilizada e por quê?
2. Qual a diferença entre comunicação síncrona e assíncrona?
3. O que pode acontecer se o transmissor e receptor estiverem fora de sincronia?

4. O que é RS-232? O que ele define? Quais as diferenças entre ele e a USART?
5. Faça os gráficos que demonstrem a transmissão síncrona e assíncrona dos seguintes valores decimais: 02, 20, 200, 222.

Obs: Utilize um sinal de CLOCK e um de dados na síncrona, enquanto na assíncrona utilize somente o sinal de dados.

## Referências

EXTREME electronics. Disponível em:  
<<http://extremeelectronics.co.in/avr-tutorials/rs232-communication-the-basics/>>. Acesso em: 11 abr. 2012.

\_\_\_\_\_. RS232 Communication: the level conversion. Disponível em:  
<<http://extremeelectronics.co.in/avr-tutorials/rs232-communication-the-level-conversion/>>. Acesso em: 11 abr. 2012.

SERIAL and UART Tutorial . Disponível em:  
<<http://www.freebsd.org/doc/en/articles/serial-uart/>>. Acesso em: 11 abr. 2012.