

# Aula 6: Acesso à porta serial

## Apresentação

---

Na computação, a comunicação serial é, ainda, uma das mais utilizadas. Uma porta serial é uma interface de comunicação por meio da qual a informação é transferida, para dentro ou para fora, um bit por vez (em contraste com uma porta paralela).

Durante a maior parte da história dos computadores pessoais, os dados foram transferidos por meio de portas seriais RS232 para dispositivos como modems, terminais e vários periféricos.

Nesta aula, utilizaremos funções em C que permitirão utilizar a comunicação serial por RS232, um aprendizado importante para a criação de interfaces de softwares que realizam a aquisição de dados de hardware externo em aplicações de software básico.

Também será vista uma forma de testar a comunicação sem hardware externo, com a utilização de programa que virtualiza portas seriais e simula a ligação entre elas.

## Objetivos

---

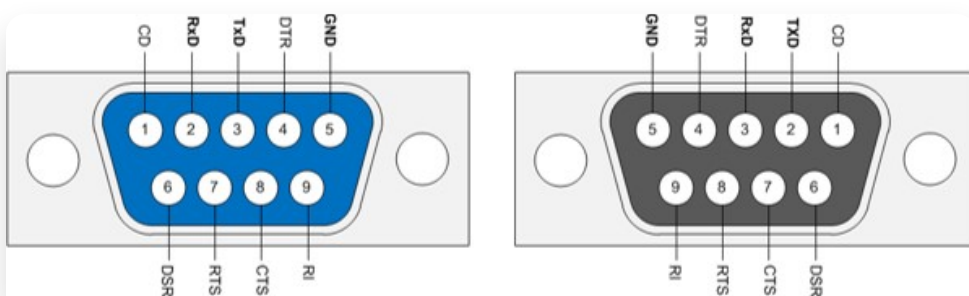
- Identificar o hardware serial e o protocolo RS232 para tráfego de dados por essa porta;
- Operar o uso das funções em C para acesso a portas seriais.

## Porta serial RS232

---

Recommended Standard (RS) significa padrão recomendado. Nos anos 60, um comitê de padrões agora conhecido como Electronic Industries Association (EIA) desenvolveu uma interface para conectar terminais de computadores a modems. Ao longo dos anos, isso foi atualizado: A versão mais comum do padrão é RS232C (às vezes conhecida como EIA232); e a mais recente é o RS232E.

- 1 O padrão define as características elétricas e mecânicas da conexão – incluindo os níveis de tensão, a taxa máxima de bits e a função dos sinais e pinos do handshake, que significa aperto de mão e estabelece o processo pelo qual duas ou mais máquinas informam que reconheceram umas às outras e, a partir daí, estão prontas para iniciar a comunicação.
- 2 Se o RS232 é um padrão, por que não posso simplesmente usar um cabo-padrão para conectar duas portas RS232 e esperar que elas conversem entre si? A resposta é que o padrão RS232 foi criado para uma situação específica: Conectar computadores a modems. Qualquer outro uso estaria, portanto, fora do padrão.
- 3 O padrão definia como os computadores – chamados de Data Terminal Equipment (DTE) – fariam a conexão com os modems – denominados Data Communication Equipment(DCE). O padrão dizia que os computadores deveriam estar equipados com um conector de nove vias (25 vias no início), enquanto os modems deveriam ter um soquete (conector fêmea)de nove vias, como mostrado na figura 1.

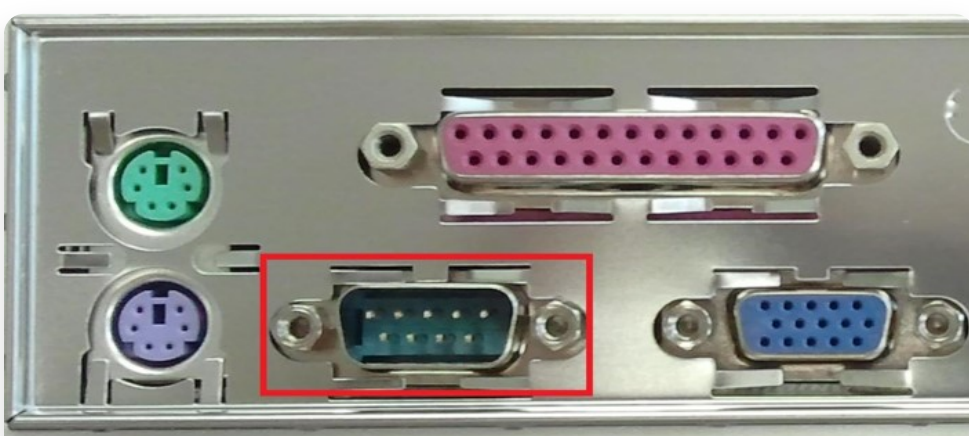


**Figura 1:** Conector DTE macho de nove vias (azul) e DCE fêmea de nove vias (Fonte: [wikimedia](#)).

O cabo de interconexão entre um computador e um modem era pino a pino (pino1-pino1, pino2-pino2etc). É importante observar que um sinal que é uma saída de um computador é uma entrada para um modem e vice-versa. Isso significa que você nunca pode distinguir apenas pelo nome do sinal se é uma entrada ou uma saída de um determinado equipamento. Os principais sinais da figura 1e suas direções de fluxo são:

- GND: Nível de tensão de referência zero para todos os sinais de controle.
- TxD (Transmit Pin): Transmite dados do DTE para o DCE.
- RxD (Receive Pin): Envia dados do DCE para o DTE.
- DTR (Data Terminal Ready): DTE está pronto para aceitar o pedido.
- CD ou DCD (Data Carrier Detect): DCE aceita uma operação de um DTE localizado em um local remoto.
- DSR (Data Set Ready): DCE está preparado para enviar e receber as informações.
- RI (RingIndicator): Detecta o toque de entrada na linha telefônica.
- CTS (Clear to Send): DTE pede que o DCE envie os dados.
- RTS (Request to Send): DTE está preparado para enviar os dados.

## Porta serial para comunicação com outros dispositivos



**Figura 2:** Conector de porta serial DB9 macho nos computadores. (Fonte: Wikimedia)

A porta serial sempre foi parte integrante dos computadores e é considerada uma das mais básicas conexões externas, junto com a porta paralela. Normalmente disponível em um conector de nove pinos (DB9 macho marcado na figura 2), foi usada como interface para periféricos diversos. O sistema Windows as identifica como portas COMx, onde x é o número da porta.

RS232 é o nome também atribuído ao protocolo de comunicação serial utilizado, que envolve o envio de dados bit a bit no tempo, em uma linha de comunicação. A despeito dos demais sinais utilizados em uma comunicação serial ainda mais antiga, com modems, como vimos na figura 1, uma comunicação simples pode ser feita com a ligação de pinos de transmissão (TX), recepção (RX) e terra (GND), conforme mostra a figura 3.

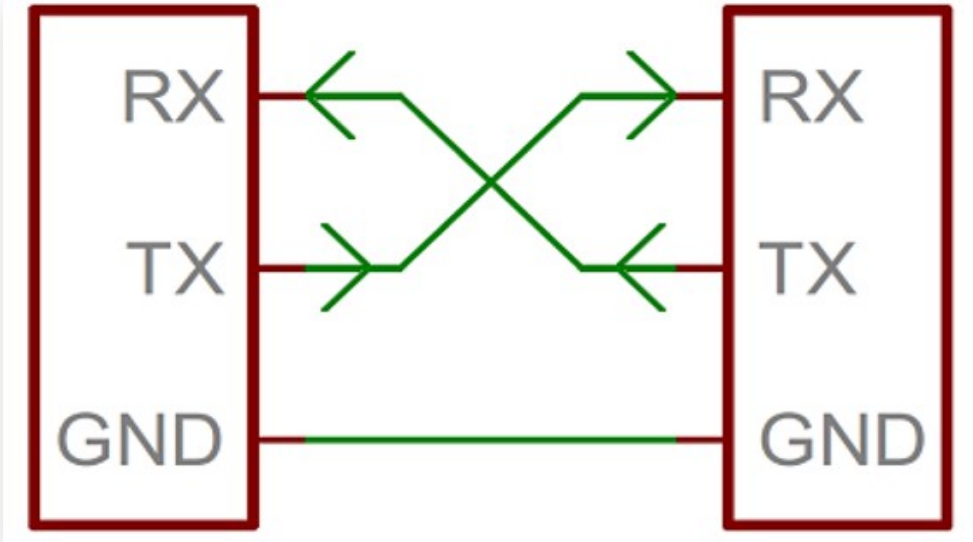


Figura 3: Pinos básicos para comunicação serial RS232

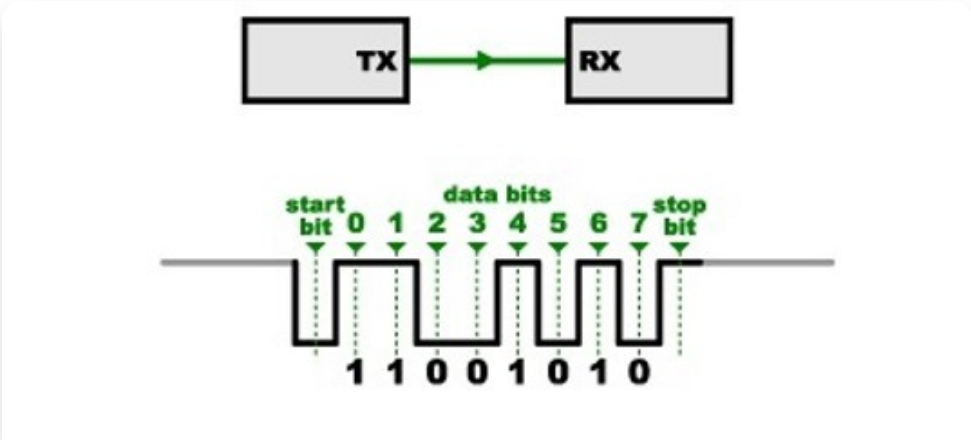


Figura 4: Um pacote de dados típico em comunicação serial RS232

O pino de transmissão envia um pacote de bits que será recebido bit a bit pelo pino receptor. Cada pacote enviado (figura 4) contém: Um start bit que indica o início da mensagem; um ou dois stop bits para indicar o fim da mensagem; e cinco a nove bits de informação (data bits). Pode usar ainda um bit de paridade para evitar a recepção de erros.

O RS232 especifica também os níveis de tensão. Observe que alguns desses níveis de tensão são negativos e também podem atingir  $\pm 15V$ . Um balanço de tensão maior torna o RS232 mais resistente a interferências.

### Estabelecendo a comunicação

Para estabelecer a comunicação serial por RS232, devemos obter todas as informações disponíveis sobre o instrumento ou dispositivo. Precisamos saber:

O número da porta de comunicação (COM) na qual se conectou o dispositivo.

A taxa de transmissão (Baud Rate) do dispositivo.

O número de bits de dados e a paridade.

O protocolo RS232 basicamente pode ser estabelecido com o envio de dados de forma assíncrona, ou seja, sem a necessidade de um sinal de sincronismo. Nesse caso, um parâmetro muito importante é o Baud Rate, que especifica a velocidade de recepção e envio, sendo que os dois dispositivos devem utilizar a mesma taxa. A unidade é bits por segundo (bps). Taxas de comunicação assíncrona comumente usadas são 9600, 33600 e 115200 bps.

# Protocolo RS232 via USB

Computadores atuais não disponibilizam portas seriais no padrão RS232, em função de os periféricos terem migrado para o padrão Universal Serial Bus(USB). O protocolo RS232, no entanto, ainda é muito difundido para a ligação de instrumentos mais antigos em PCs ou nas placas de microcontroladores com interface serial, usadas no desenvolvimento de controles, sistemas Internet das Coisas (IoT) e automação em residências e indústrias.

A alternativa para a ligação desses dispositivos é o conversor Serial-USB, embutido nas placas de desenvolvimento para microcontroladores mais comuns, como as de Arduíno, LaunchPad MSP430 ou PIC, mas que também pode ser adquirido separadamente.

## Comentário

Esses conversores têm um circuito integrado (CI ou chip) que converte os dados entre os protocolos RS232 e USB, criando portas COM virtuais no computador que podem ser usadas como as antigas portas seriais, inclusive com o mesmo protocolo RS232.

## Funções para acesso às portas RS232

O acesso à porta serial em Windows pode ser feito com a função CreateFile(), que cria ou abre um arquivo ou dispositivo de entrada ou saída (E/S). A função retorna um identificador (HANDLE) que pode ser usado para acessar o arquivo ou dispositivo de E/S. Depois de abrir uma porta serial, é possível fechá-la com o uso da função CloseHandle().

## Exemplo

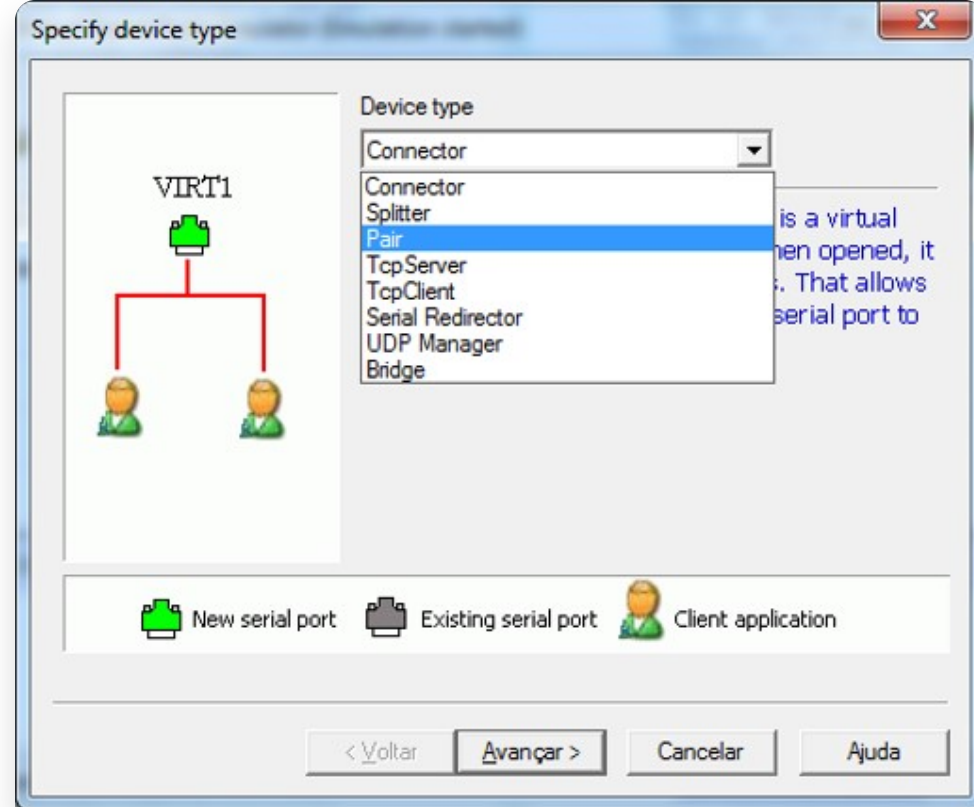
Veja um [exemplo](#) que pode ser executado com o ambiente DEV C++ e mostra a abertura e o fechamento de uma porta serial.

- 1

A primeira função declara um identificador ou manipulador (HANDLE) para acessar a porta serial. Em seguida, a função CreateFile recebe sete argumentos, como a porta COM a ser aberta e o modo de acesso, retornando, em caso de sucesso, um identificador válido para a variável hComm.
- 2

A porta COM disponível no sistema pode ser verificada no gerenciador de dispositivos. Em geral, é enumerada uma porta COM1. Caso não exista, o teste da porta poderá ser feito com a enumeração de portas virtuais por meio do programa [Free Virtual Serial PortsEmulator](#).
- 3

Com o programa instalado, clique em Create new device e, em Device type, escolha Pair (figura 5). Escolha as duas COM que desejar para criar um par virtual de portas interligadas. Isso servirá para os testes de códigos de acesso à porta serial.



**Figura 5:** Criação de um par de portas COM virtuais no programa Free Virtual Serial Ports Emulator

## Leitura e escrita na porta serial

No Windows, configurações como taxa de transmissão, número de bits de início/parada, formatos de dados etc., para a porta serial, são controlados pela estrutura do DCB.

Para configurar a estrutura do DCB, usamos duas funções:

1

### Função GetCommState()

recupera as configurações de controle atuais da porta serial.

2

### Função SetCommState()

Os timeouts ajudam a evitar que o programa espere indefinidamente até que os dados cheguem. Eles ajudam a ler ou escrever chamadas para retornar depois de um período de tempo definido.

## Definir tempos-limite (timeouts)

Os timeouts ajudam a evitar que o programa espere indefinidamente até que os dados cheguem. Eles ajudam a ler ou escrever chamadas para retornar depois de um período de tempo definido.

## Escrevendo dados na porta serial

A gravação de dados na porta serial aberta é realizada pela função WriteFile(). A função WriteFile() pode ser usada para gravar nos arquivos e nas portas de E/S.

## Exemplo

Veja os exemplos de programação a seguir:

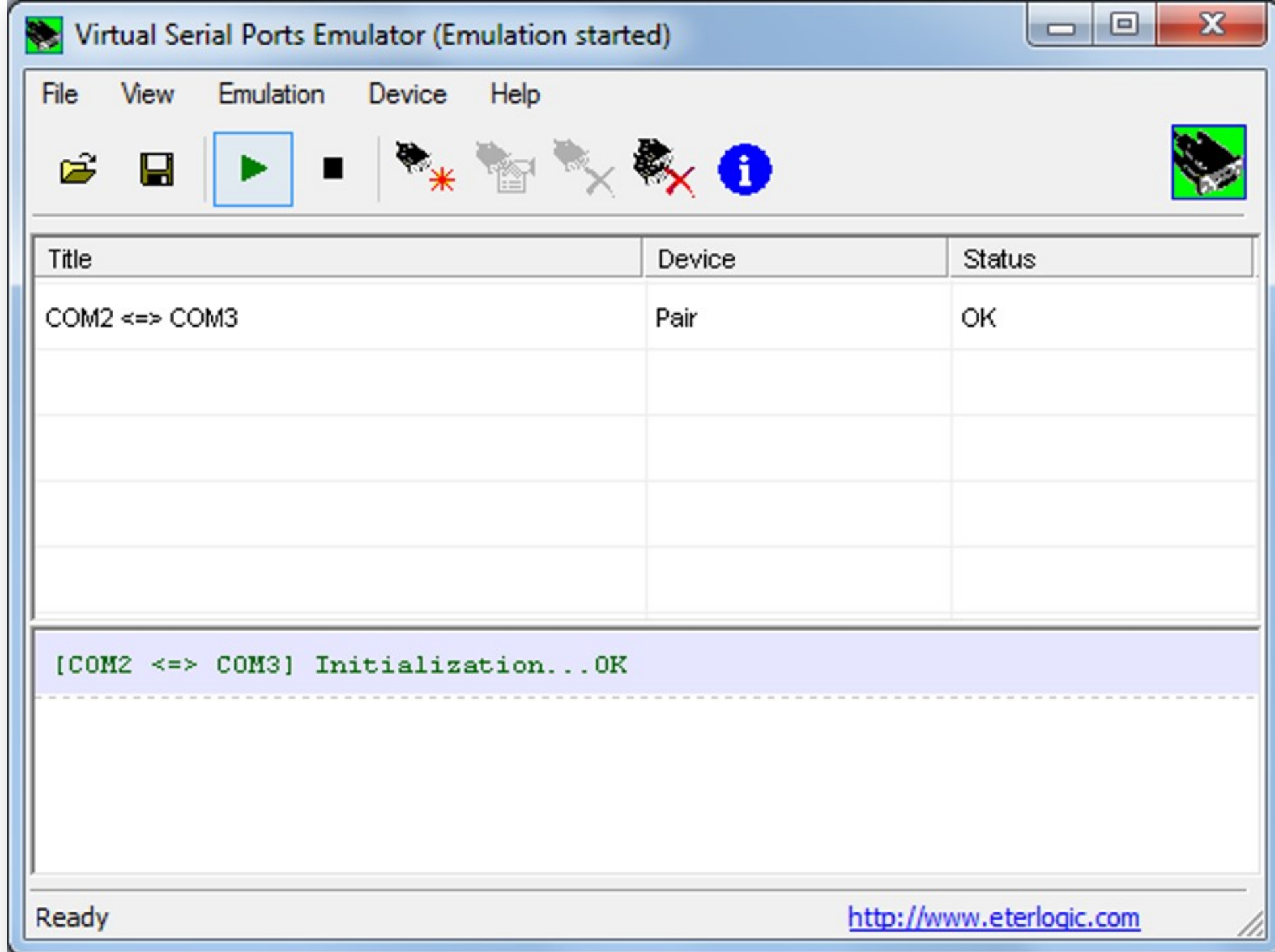
- [Programa para escrita na porta serial](#)
- [Programa para leitura na porta serial](#)

**Executando o programa para escrita e o programa para leitura ao mesmo tempo, junto com o programa para enumeração de portas virtuais, com COM3 e COM2 interligadas, é possível testar a comunicação serial escrevendo em um programa e recebendo em outro, continuamente.**

Observe que os programas de leitura e escrita não podem ser executados a partir do Dev C++. Pelo menos um deles, ou os dois, deve ser executado a partir do respectivo arquivo .exe que é criado na mesma pasta do arquivo-fonte.

Veja o programa Free Virtual Serial PortsEmulator interligando as COM2 e COM3, o programa de escrita e o programa de leitura, respectivamente:





**Figura 6:** Ligação das portas virtuais COM2 e COM3 estabelecidas no programa Free Virtual Serial PortsEmulator.

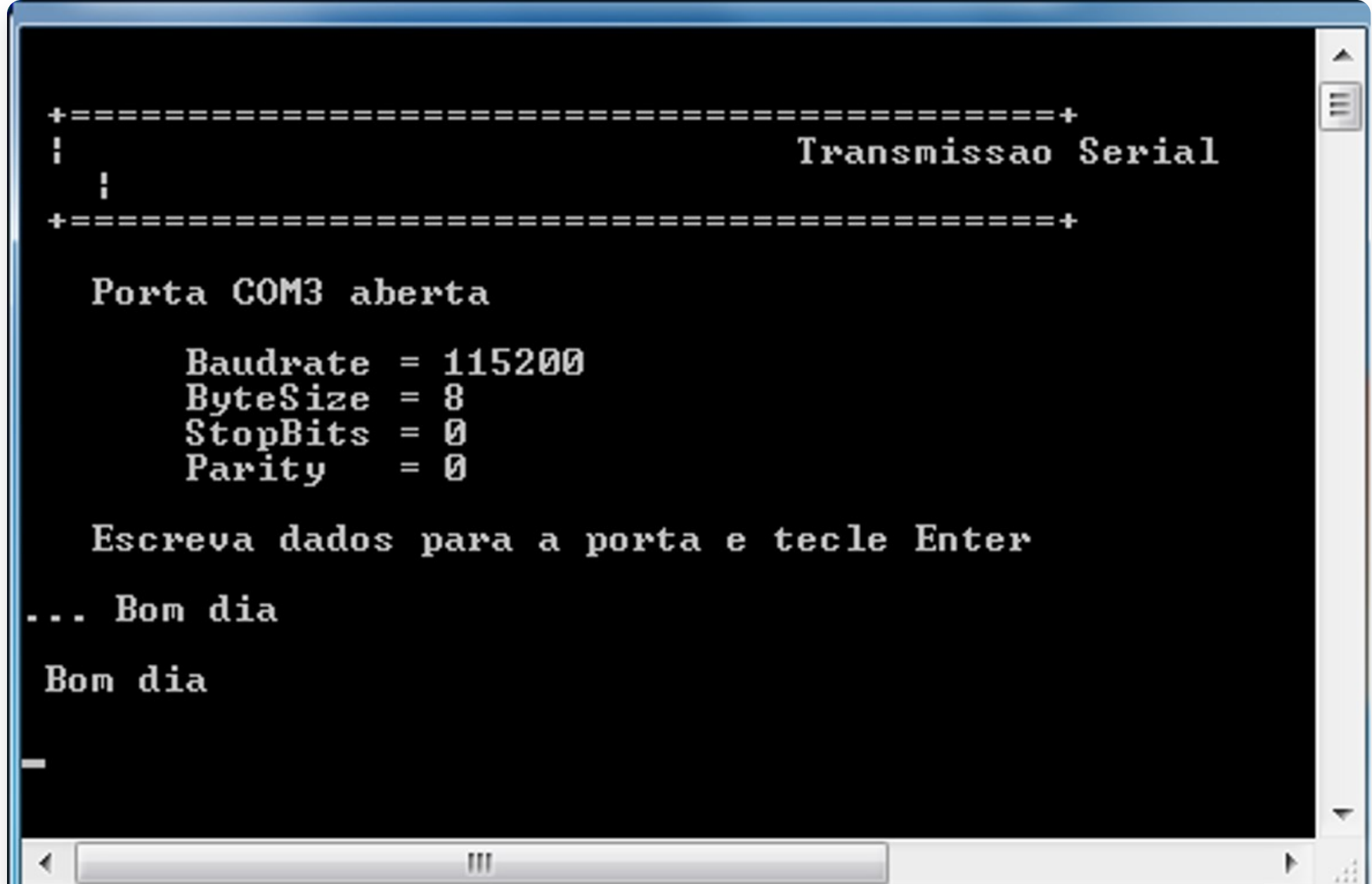


Figura 7: Programa de escrita na porta serial.

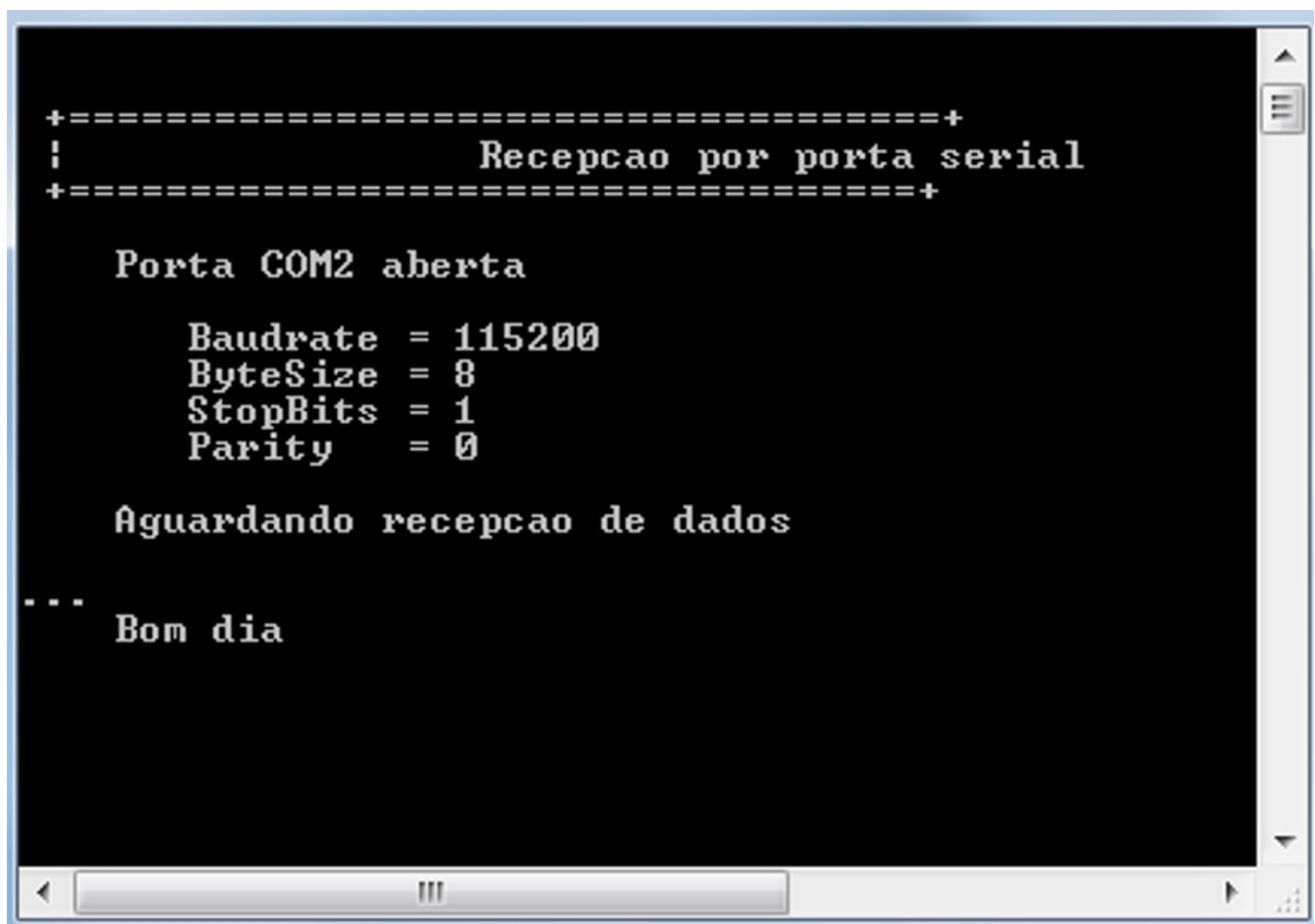


Figura 8: Programa de recepção pela porta serial.



# Atividade

---

O entendimento adequado do funcionamento do hardware serial é fundamental para usar a interface de E/S de dados. Resolva as questões 1, 2 e 3.

1. A transferência de dados no RS232 é feita em \_\_\_\_\_.

- a) formato assíncrono de início (start) e parada (stop).
  - b) formato síncrono de início (start) e parada (stop) .
  - c) formato isócrono.
  - d) formato EBDIC.
- 

2. A transferência de dados no RS232 é feita em \_\_\_\_\_.

- a) formato assíncrono de início (start) e parada (stop).
  - b) formato síncrono de início (start) e parada (stop) .
  - c) formato isócrono.
  - d) formato EBDIC.
- 

3. O que significa um baud rate de 9.600 bps na comunicação serial?

4. Altere os programas de recepção e escrita mostrados para que o de recepção responda automaticamente a pelo menos três perguntas escolhidas por você, e essas respostas apareçam no terminal de escrita.

5. Quais motivos fariam um projetista escolher RS232 para comunicação em distâncias mais longas?

6. Quais as diferenças básicas entre as interfaces RS232 e USB?

7. Como são transmitidos os bits de dados em RS232?

## Notas

Título modal <sup>1</sup>

---

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

## Título modal <sup>1</sup>

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

## Referências

TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas operacionais modernos**. 4. ed. São Paulo: Pearson, 2015.

## Próxima aula

- O essencial da programação de software básico voltada para computadores em um único chip: Os microcontroladores;
- Aquisição de dados com microcontroladores;
- Programação cliente/servidor.

## Explore mais

Embora inexistente nos computadores atuais, ainda é comum encontrar em laboratórios a interface RS232 em equipamentos, o que muitas vezes suscita a atualização do software supervisorio, que controla e recebe dados desse equipamento, para versões mais novas de sistemas operacionais.

Avaliar esses trabalhos é uma forma de aprender sobre o tema, que pode vir a desafiá-lo no futuro. O material a seguir mostra alguns exemplos de uso da interface RS232 e outros protocolos, como o Modbus, em trabalhos de automação em laboratórios e indústrias. Leia os textos:

- [Extração do protocolo de comunicação para o controle de um dispositivo através do monitoramento dos sinais da porta serial \(RS232\)](#) .
- [Interoperabilidade entre dispositivos de diferentes fabricantes via protocolo Modbus](#) .
- [Comunicação ZigBee aplicada em um sistema de controle](#).