



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UniFOA

**23 a 25
de outubro**

Submissões abertas até 07/09

Análise comparativa de sinais TTL e RS-232 em comunicação serial: estudo da transmissão sequencial dos caracteres "232" a 9600 bps

Vitor Amadeu Souza¹; 0009-0002-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este trabalho apresenta uma análise comparativa entre os padrões de sinalização TTL (Transistor-Transistor Logic) e RS-232 na transmissão de dados seriais a 9600 bps. Através de simulação computacional, foram gerados e analisados os sinais correspondentes à transmissão dos caracteres ASCII '2', '3' e '2', permitindo a observação das características temporais e de amplitude de cada padrão. Os resultados demonstram as diferenças fundamentais entre os dois meios físicos, incluindo as inversões lógicas características do RS-232 e as amplitudes de tensão distintas. O estudo contribui para o entendimento das tecnologias de comunicação serial amplamente utilizadas em sistemas embarcados e aplicações industriais, fornecendo base teórica e prática para a escolha adequada do protocolo de comunicação em diferentes cenários de aplicação.

Palavras-chave: Comunicação serial. TTL. RS-232. Transmissão de dados. Protocolos de comunicação.



INTRODUÇÃO

A comunicação serial representa um dos fundamentos mais importantes nos sistemas de comunicação digital modernos, sendo amplamente utilizada em aplicações que vão desde microcontroladores até sistemas industriais complexos. A transmissão serial envolve o envio de dados um bit por vez, com a taxa total de bits transmitidos por segundo denominada taxa de transmissão ou baud rate, cujo inverso representa o tempo necessário para transmitir um bit individual (Valvano, 2014). Entre os diversos padrões de comunicação serial, o TTL (Transistor-Transistor Logic) e o RS-232 destacam-se como protocolos fundamentais que definiram as bases para muitas das tecnologias de comunicação contemporâneas.

O padrão TTL, desenvolvido inicialmente pela Texas Instruments na década de 1960, utiliza níveis lógicos baseados em tensões próximas a 0V para nível lógico baixo e aproximadamente 5V para nível lógico alto. A comunicação serial assíncrona é frequentemente referenciada como TTL serial, onde o nível alto de tensão representa lógica 1 e o nível baixo equivale à lógica 0, sendo amplamente implementada em microcontroladores modernos através de interfaces UART (Universal Asynchronous Receiver-Transmitter) integradas (Microcontroller Tips, 2017). Esta implementação tornou o TTL popular em sistemas embarcados e aplicações de baixa complexidade, onde a compatibilidade direta com circuitos integrados digitais é essencial.

Por outro lado, o padrão RS-232, introduzido em 1960 como Recommended Standard 232, representa uma abordagem mais robusta para comunicação serial de longa distância, definindo formalmente os sinais de conexão entre equipamentos terminais de dados (DTE) como computadores e equipamentos de circuito de dados (DCE) (Frenzel, 2015). O RS-232 utiliza níveis de tensão mais elevados e emprega lógica inversa, característica que foi desenvolvida para aumentar a imunidade a ruídos e permitir transmissões confiáveis em distâncias maiores. Os circuitos integrados driver/receiver RS-232 realizam a conversão de níveis necessária entre as interfaces CMOS/TTL e RS-232, sendo fundamentais para a implementação prática deste meio físico (Analog Devices, 2001).

A importância do estudo comparativo destes meios físicos reside na necessidade crescente de sistemas de comunicação eficientes e confiáveis em aplicações industriais modernas.



Com o advento da Internet das Coisas (IoT) e sistemas embarcados cada vez mais complexos, a escolha adequada do meio físico de comunicação pode impactar o desempenho, a confiabilidade e o custo total de um projeto. Além disso, muitos sistemas legados ainda dependem fortemente desta comunicação, tornando fundamental o entendimento de suas características e limitações para projetos de integração e modernização.

A taxa de transmissão de 9600 bits por segundo (bps) foi selecionada para este estudo por representar uma velocidade amplamente utilizada em aplicações, oferecendo um equilíbrio adequado entre velocidade de transmissão e confiabilidade. Esta taxa é particularmente comum em sistemas de monitoramento industrial, equipamentos médicos e dispositivos de telemetria, onde a estabilidade da comunicação é prioritária em relação à velocidade máxima de transmissão (Microcontroller Tips, 2017).

MÉTODOS

Para realizar a análise comparativa dos padrões TTL e RS-232, foi desenvolvido um ambiente de simulação computacional utilizando a linguagem Python com as bibliotecas NumPy e Matplotlib. A metodologia empregada baseou-se na geração sintética de sinais digitais que reproduzem fielmente as características temporais e de amplitude de ambos os protocolos de comunicação.

O processo de simulação iniciou-se com a definição dos dados a serem transmitidos, especificamente três transmissões sequenciais dos caracteres ASCII '2', '3', e '2' (códigos decimais 50, 51, e 50, respectivamente, correspondendo aos padrões binários 00110010, 00110011, e 00110010). Esta sequência "232" foi transmitida caractere por caractere através de uma lista de entrada binária estruturada no código Python, permitindo a observação clara das transições de sinal e padrões repetitivos em ambos os protocolos. A estrutura de transmissão seguiu o formato padrão UART (Universal Asynchronous Receiver-Transmitter), composto por um bit de início (start bit), oito bits de dados, e um bit de parada (stop bit), totalizando dez bits por caractere transmitido (Frenzel, 2015).

Para a simulação do sinal RS-232, foi implementada a lógica inversa característica deste protocolo, onde o nível lógico '0' corresponde a +12V e o nível lógico '1' corresponde a -12V.



Esta inversão lógica, conhecida como "mark" e "space", é uma característica fundamental do RS-232 que contribui para sua robustez em ambientes com interferência eletromagnética. O estado de repouso da linha RS-232 foi configurado como -12V, representando o nível lógico '1' conforme especificado no padrão (Dawoud; Dawoud, 2022).

A geração dos sinais foi realizada através de arrays NumPy que representam os valores de tensão ao longo do tempo, utilizando a função step do Matplotlib para criar visualizações que reproduzem fielmente o comportamento de sinais digitais reais. Os gráficos foram configurados com escala temporal em milissegundos e amplitude apropriada para cada protocolo, permitindo a comparação direta entre os padrões de sinalização.

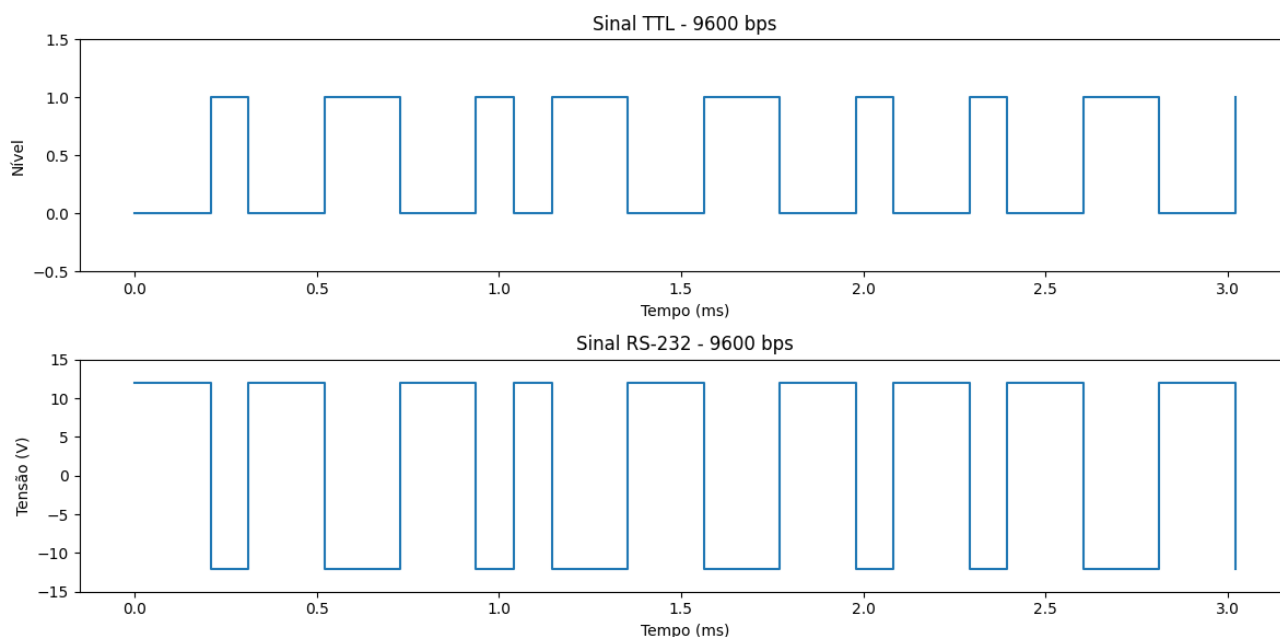
O código-fonte desta pesquisa está disponível para download através do link: <https://github.com/vitor-souza-ime/RS232>.

RESULTADOS E DISCUSSÃO

A análise dos sinais gerados pela simulação revelou características distintas e complementares entre os protocolos TTL e RS-232, conforme demonstrado nos gráficos obtidos e apresentados na Figura 1. O sinal TTL apresentou transições nítidas entre os níveis 0V e 1V, mantendo a lógica positiva esperada onde cada bit '1' dos dados é representado pelo nível alto de tensão e cada bit '0' pelo nível baixo. A duração de cada bit, calculada em aproximadamente 0,104 ms, manteve-se consistente ao longo de toda a transmissão da sequência "232", confirmando a estabilidade temporal do protocolo a 9600 bps.

O padrão de transmissão observado no sinal TTL seguiu a estrutura UART, iniciando cada caractere com um start bit em nível baixo (0V), seguido pelos oito bits de dados na ordem bit menos significativo (LSB) primeiro, e finalizando com um stop bit em nível alto (1V). Esta sequência é visível nos gráficos gerados, onde se observa a transmissão sequencial dos caracteres '2', '3', e '2'. O primeiro caractere '2' (00110010 em binário) gera o padrão de transmissão 0-01001100-1, seguido pelo caractere '3' (00110011 em binário) que produz 0-11001100-1, e finalmente a repetição do caractere '2' com o mesmo padrão inicial, considerando a ordem LSB primeiro característica do protocolo UART.

Figura 1 - Plotagem TTL e RS232



Fonte: O autor.

Em contraste, o sinal RS-232 demonstrou comportamento inverso ao TTL, com amplitudes significativamente maiores e lógica negativa. Os níveis de tensão observados foram +12V para o bit '0' e -12V para o bit '1', resultando em uma excursão total de 24V entre os estados lógicos. Esta característica do RS-232 é fundamental para sua capacidade de transmissão em longas distâncias e imunidade a interferências, uma vez que a amplitude elevada do sinal permite distinguir claramente entre os estados lógicos mesmo na presença de ruído significativo, conforme descrito por Frenzel (2015) em seu handbook sobre interfaces de comunicação serial.

A inversão lógica do RS-232 é observável quando comparada ao sinal TTL correspondente. Enquanto o TTL apresenta nível alto durante o estado de repouso e os bits '1', o RS-232 mantém -12V nestas mesmas condições. Durante os start bits e bits de dados '0', o TTL desce para 0V enquanto o RS-232 sobe para +12V, criando um padrão espelhado que é característico desta diferença fundamental entre os protocolos. Esta inversão de polaridade é uma das principais características que distinguem o RS-232 de outros meios físicos de comunicação serial.



A compatibilidade entre os protocolos é um aspecto vital revelado pela análise. Embora os sinais apresentem amplitudes e polaridades distintas, a estrutura temporal idêntica permite a conversão entre os padrões através de circuitos de interface apropriados. Conversores TTL-RS232, comumente baseados em circuitos integrados dedicados, realizam esta conversão mantendo a integridade temporal dos dados enquanto adaptam os níveis de tensão e lógica conforme necessário, conforme detalhado no trabalho de Nelson (2000) sobre desenvolvimento de comunicações seriais.

CONCLUSÕES

O estudo comparativo entre o TTL e RS-232 demonstrou as características distintivas de cada padrão de sinalização e sua adequação para diferentes aplicações práticas. A análise da transmissão sequencial dos caracteres "232" revelou que, embora ambos os meios físicos compartilhem a mesma estrutura temporal e de dados UART, suas diferenças fundamentais em amplitude, lógica e robustez os tornam adequados para cenários de aplicação específicos e complementares.

O TTL mostrou-se ideal para aplicações de curta distância e alta velocidade, onde a implementação e compatibilidade direta com circuitos digitais são prioritárias. Sua lógica positiva intuitiva e baixos níveis de tensão facilitam a integração com microcontroladores e sistemas embarcados modernos, confirmando sua relevância contínua em aplicações contemporâneas. A estabilidade temporal observada na simulação a 9600 bps sugere que este meio mantém integridade de sinal em condições controladas, sendo apropriado para comunicação intra-sistema e aplicações onde o ambiente eletromagnético é relativamente limpo.

O RS-232, por sua vez, demonstrou características superiores para aplicações que exigem transmissão confiável em longas distâncias e ambientes com interferência eletromagnética. A amplitude elevada do sinal (24V de excursão total) e a lógica inversa contribuem para a robustez da comunicação, justificando sua adoção continuada em aplicações industriais críticas. A capacidade de manter a integridade do sinal em cabos de maior extensão, combinada com a imunidade a ruídos, torna o RS-232 uma escolha preferencial para



sistemas de monitoramento industrial, equipamentos médicos e aplicações onde a confiabilidade da comunicação é fundamental.

A análise temporal confirmou que ambos os protocolos podem coexistir e ser convertidos entre si sem perda de informação, desde que sejam utilizados circuitos de interface apropriados. Esta compatibilidade é essencial para a manutenção de sistemas legados e a integração de tecnologias modernas com equipamentos existentes, demonstrando a importância de compreender ambos os protocolos em projetos de engenharia contemporâneos.

REFERÊNCIAS

ANALOG DEVICES. Fundamentals of RS-232 Serial Communications. Technical Articles, 2001. Disponível em: <https://www.analog.com/en/resources/technical-articles/fundamentals-of-rs232-serial-communications.html>. Acesso em: 31 ago. 2025.

AXELSON, J. Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems. 2nd ed. Madison: Lakeview Research, 2007.

BARR, M. Programming Embedded Systems: With C and GNU Development Tools. 2nd ed. Sebastopol: O'Reilly Media, 2006.

CATSOULIS, J. Designing Embedded Hardware. 2nd ed. Sebastopol: O'Reilly Media, 2005.

COMER, D. Computer Networks and Internets. 6th ed. Boston: Pearson, 2014.

DAWOUD, D. S.; DAWOUD, P. Serial Communication Protocols and Standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX. Aalborg: River Publishers, 2022. Disponível em: <https://www.taylorfrancis.com/books/mono/10.1201/9781003339496/serial-communication-protocols-standards-dawoud-shenouda-dawoud-peter-dawoud>. Acesso em: 31 ago. 2025.

EGGER, D. J. Interfacing to the IBM Personal Computer. 2nd ed. Indianapolis: Howard W. Sams, 1990.

FRENZEL, L. E. Handbook of Serial Communications Interfaces: A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards. Oxford: Newnes, 2015. Disponível em: <https://www.sciencedirect.com/book/9780128006290/handbook-of-serial-communications-interfaces>. Acesso em: 31 ago. 2025.

GANSSELE, J. The Art of Designing Embedded Systems. 2nd ed. Oxford: Newnes, 2008.



4º Congresso Brasileiro
de Ciência e Saberes
Multidisciplinares
**tudo é
ciência**
11º Encontro de Extensão
Universitária do UNIOA

**23 a 25
de outubro**

Submissões abertas até 07/09

HINTZ, K. J.; TABAK, D. Microcontrollers: Architecture, Implementation and Programming. New York: McGraw-Hill, 1992.

HOROWITZ, P.; HILL, W. The Art of Electronics. 3rd ed. Cambridge: Cambridge University Press, 2015.

HYDE, J. USB Design by Example: A Practical Guide to Building I/O Devices. 2nd ed. San Jose: Intel Press, 2001.

MICROCONTROLLER TIPS. Asynchronous serial communication explained (including TTL, UART, and RS232). Microcontroller Tips, 2017. Disponível em: <https://www.microcontrollertips.com/asynchronous-serial-communication-explained-including-ttl-uart-rs232/>. Acesso em: 31 ago. 2025.

NELSON, M. Serial Communications Developer's Guide. San Francisco: IDG Books Worldwide, 2000.

PREDKO, M. Programming and Customizing the 8051 Microcontroller. New York: McGraw-Hill, 1999.

SEED STUDIO. RS232 vs TTL: Beginner Guide to Serial Communication. Seeed Studio Blog, 9 dez. 2020. Disponível em: <https://www.seeedstudio.com/blog/2019/12/11/rs232-vs-ttl-beginner-guide-to-serial-communication/>. Acesso em: 31 ago. 2025.

STALLINGS, W. Data and Computer Communications. 10th ed. Boston: Pearson, 2013.

TANENBAUM, A. S.; WETHERALL, D. J. Computer Networks. 5th ed. Boston: Pearson, 2011.

TINDER, R. F. Engineering Digital Design. 2nd ed. San Diego: Academic Press, 2000.

VALVANO, J. W. Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers. 2nd ed. Austin: Jonathan Valvano, 2014. Disponível em: https://users.ece.utexas.edu/~valvano/Volume1/E-Book/C11_SerialInterface.htm. Acesso em: 31 ago. 2025.