

Projeto de Sistemas RF

Aula 05 - Comunicação Serial RS232 (Hardware)

Apresentação

Nesta aula, iremos falar especificamente sobre o padrão de comunicação RS232 (serial). Iremos focar somente na parte de hardware da comunicação. Veremos aspectos físicos, como ligações, níveis de tensão, conectores etc. Em outra aula iremos abordar os aspectos de software (registradores, contadores, dentre outros).

Objetivos

Ao final desta aula, você deverá ser capaz de:

- Entender os aspectos físicos do protocolo RS232.
- Entender, em relação ao hardware, como funciona o protocolo RS232.
- Estudar os circuitos externos e ligações necessárias para comunicação RS232.

Quadro e Encapsulamento de Dados

Como vimos em aulas passadas, comunicações seriais têm que seguir algum tipo de protocolo. No caso desta aula, estamos tratando com uma comunicação serial que segue o protocolo RS232. Esse protocolo tem uma forma específica de organizar a forma com que os dados são transmitidos, lembrando que por se tratar de uma comunicação serial, é sempre transmitido um bit por vez.

Quando queremos transmitir uma informação de um ponto a outro, normalmente essa informação não "cabe" em um único bit. O que normalmente se deseja transmitir é algo como uma mensagem composta por vários caracteres, que por sua vez são formados por diversos bits. Veja, por exemplo, o caso de transmitir a palavra "SERIAL". Essa palavra é composta por 6 caracteres (nesse caso específico, os caracteres são todas letras maiúsculas). Como sabemos, para representar cada um desses caracteres, precisamos de 7 bits. Dessa maneira, para transmitir a palavra SERIAL precisamos de 42 bits.

Cada padrão de comunicação serial estabelece uma forma com que esses 42 bits são transmitidos. Em geral eles são divididos em grupos e cada grupo transmite um pedaço da mensagem. Esses grupos são chamados de quadros. Cada quadro, além de dividir a mensagem, contém informações sobre a transmissão (começo do quadro, tamanho etc.). A **Figura 1** ilustra como seria um exemplo de sequência de bits para transmitir a palavra SERIAL.

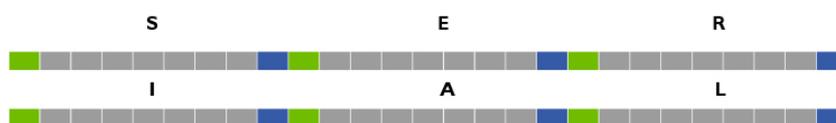


Figura 1 - Exemplo de quadros na transmissão da palavra SERIAL.

Nesse exemplo, cada quadro transmite uma letra e é formado por 9 bits: o primeiro e último bit são bits de informação do quadro (começo, término, checagem etc.); os bits do meio do quadro armazenam a letra (nesse caso, cada letra ocupa 7 bits).

As informações adicionais presentes no início de cada quadro são chamadas de preâmbulo (ou cabeçalho do quadro) e as presentes no final são chamadas de sufixo do quadro.

Dependendo do tamanho do preâmbulo e sufixo do quadro, será necessário transmitir mais bits do que a informação em si e a esse fenômeno dá-se o nome de *overhead*.

Se liga!

Lembra o nome da tabela que representa os códigos de caracteres e seus respectivos símbolos? É a famosa "tabela ASCII".

RS232 (Especificação do Quadro)

Iremos agora detalhar a composição do quadro RS232. O protocolo RS232, como todos os protocolos, tem suas especificações. Esse padrão é definido pelo EIA (*Electronic Industries Association*) e segundo ele o quadro RS232 possui 3 partes: cabeçalho, dados e sufixo.

Start bit – O cabeçalho define o começo da palavra a ser transmitida e consiste de um bit com valor 0. Esse bit denomina-se start bit.

Bits de dado – O dado a ser transmitido pode ter tamanho variado e em geral é de 7 ou 8 bits. Esse tamanho deve ser acordado com as partes transmissora e receptora do sistema, pois não há como o receptor saber que, após o start bit, virão 7 ou 8 bits.

Bit de paridade – O sufixo do quadro RS232 pode conter um bit denominado bit de paridade. Esse bit nem sempre está presente e sua presença também deve ser acordada entre transmissor e receptor. Caso esteja presente, esse bit indica a paridade do sinal. Mas o que vem a ser a paridade do sinal? Paridade é simplesmente uma indicação de que o número de bits iguais a 1 na mensagem é par ou não. Por exemplo, se

tivermos uma mensagem binária 01001010 ela terá 3 bits iguais a 1 e, portanto, o bit de paridade estaria desligado (paridade par). Se a mensagem fosse 01010110 então o bit estaria ligado (4 bits iguais a 1).

Stop bit – Os últimos bits de um quadro RS232 são os bits de parada (podem ser 1 ou 2 e novamente isso deve ser acordado entre transmissor e receptor). Esses bits são bits iguais a 1 e indicam que a mensagem terminou.

String de Configuração

É muito comum em sistemas que se comunicam via RS232 existir uma pequena sequência de caracteres (string) que identifica a configuração de uma porta serial (que utiliza o protocolo RS232). Essa string é composta por um número que indica quantos bits de dados existem. Nela são utilizadas as letras N, P ou E para indicar a paridade (N = sem paridade, P = paridade par e E = paridade ímpar) e um número para indicar a quantidade de stop bits.

Exemplos de string de configuração:

8N1 (8 bits de dados, sem paridade e 1 stop bit)

7P2 (7 bits de dados, paridade par e 2 stop bit)

8E1 (8 bits de dados, paridade ímpar e 1 stop bit)

Especificações de Hardware

O padrão RS232 como um todo não descreve só como os bits têm que ser arranjados nos quadros. Ele também nos informa como devem ser as partes "físicas" envolvidas na comunicação.

A seguir, temos resumidos alguns aspectos importantes do padrão RS232. Como veremos em aulas futuras, esses padrões servirão apenas para nos guiar sobre as conexões, pois eles especificam a comunicação

com fio. Como iremos utilizar transmissores sem fio, não iremos obedecer esta parte do padrão (embora seja importante saber para realizar testes).

Sinais e linhas físicas – A primeira coisa que nos vem à cabeça quando pensamos em comunicar serialmente (mandando um bit por vez) é: "Por onde estes bits vão passar e como faz para transmitir e receber dados ao mesmo tempo?" Na aula passada, vimos que os bits são representados por tensões elétricas em um fio. Para que tanto um dispositivo quanto o outro possam enviar e receber mensagens, precisa-se de 2 fios (3 se contar com o terra ou referência). Esses fios no protocolo RS232 são denominados de TX e RX. Existem chamadas "linhas" no protocolo, mas focaremos aqui nas "linhas" TX e RX.

Conector – Para que possamos ligar os dispositivos, os padrões especificam até o conector a ser utilizado. No caso específico da comunicação serial utilizando o padrão RS232, os conectores mais comuns são o DB9 e o DB25 (com 9 e 25 pinos, respectivamente). A **Figura 2** ilustra esses conectores.



Figura 2 - Conectores utilizados no padrão RS232.

Fonte: Extraído de: <http://www.farnell.com.br/>. Acesso em: 10 maio 2012

Tensões – As tensões elétricas usadas para representar um bit 1 ou 0 no padrão RS232 são também especificadas. Nesse caso, os bits 0 são representados por tensões que podem variar de 3 a 15 volts. O bit 1 é representado por tensões entre -3 e -15 volts (isso mesmo, é "ao contrário", bits 1 são negativos e bits 0 são positivos).

Velocidade – O fator mais importante para nossa disciplina do padrão RS232 é a velocidade de transmissão. O padrão especifica diversas velocidades que podem ser utilizadas. Alguns exemplos dessas velocidades são (em bits por segundo ou bauds): 1200, 4800, 9600, 115200.

Mesmo essas velocidades sendo padronizadas, não é estritamente necessário que elas sejam obedecidas exatamente. Se for combinado entre o transmissor e o receptor uma velocidade diferente dessas, a única coisa que deve ser garantida é que ambos (transmissor e receptor) operem nessa mesma velocidade.

Se liga!

Atenção! A velocidade de transmissão será um ponto importante quando estivermos utilizando o padrão RS232 para transmitir sem fio, pois nessa hora quem vai ditar a velocidade máxima possível vai ser o dispositivo transmissor.

Circuitos Auxiliares (MAX232)

Quando estamos utilizando fios para transmitir dados seriais, já vimos que o padrão exige que representemos os bits por tensões negativas (para bits 1) e positivas (para bits 0). Além de serem bipolares (ter tensão positiva e negativa), essas tensões são normalmente altas (maiores do que 10 V). Isso dificulta a implementação de circuitos que utilizem o padrão RS232. Essa dificuldade se dá porque praticamente em todos os circuitos os dispositivos digitais utilizam os componentes de transmissão (no caso da serial as USART) com tensões de 0 a 5 V.

Felizmente, existem pequenos circuitos integrados que convertem esses níveis de tensão usados em dispositivos digitais (entre 0 e 5 V) para níveis específicos do padrão RS232. O exemplo mais famoso desses dispositivos é o MAX232. Ele é um pequeno CI que recebe em sua entrada tensões de 0 a 5 V e fornece na sua saída as tensões correspondentes aos

níveis do padrão RS232 (no caso do MAX232, de -12 a 12). Para que isso seja possível, o MAX232 tem que ser conectado a um pequeno grupo de capacitores de valor relativamente alto para poder formar o circuito elevador de tensão. A **Figura 3** ilustra essas tensões.

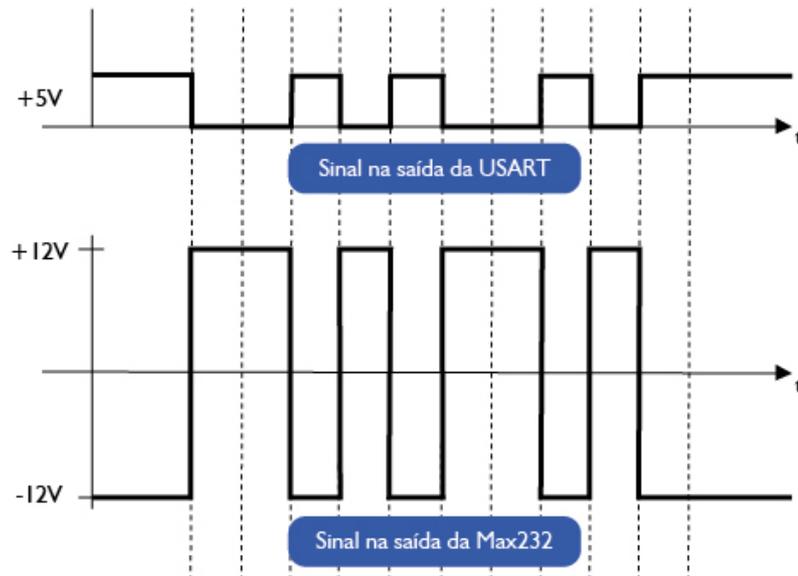


Figura 3 - Níveis de tensão na saída da USART e na saída do MAX232.

Fonte: Adaptado de: <http://www.best-microcontroller-projects.com/how-rs232-works.html>. Acesso em: 10 maio 2012

Se liga!

Afinal, por que estas tensões tão estranhas (negativas e altas) do protocolo RS232? Na verdade, elas servem para garantir que os dados possam ser transmitidos em cabos com uma distância relativamente alta. Isso porque os efeitos capacitivos de cabos grandes deformariam muito os bits, dificultando a identificação de bits 0 e 1 (caso fossem utilizadas tensões pequenas entre 0 e 5V).

Leitura Complementar

- Nesse site, você encontrará mais informações sobre o CI MAX 232. <<http://en.wikipedia.org/wiki/MAX232>>
- Essa página mostra os diversos tipos de ligações possíveis entre conectores DB9 e DB25. <<http://www.lammertbies.nl/comm/cable/RS-232.html>>
- Nesse link, você obterá mais informações sobre o protocolo que usaremos bastante neste curso (o RS-232). <<http://en.wikipedia.org/wiki/RS-232>>

Resumo

Nesta aula, aprendemos algumas características físicas do protocolo RS-232. Vimos como ocorre sua composição em termos de quadro, ligações físicas e conectores. Com esta aula estamos prontos para aprender a ligar fisicamente dois dispositivos. Nas próximas aulas, iremos realizar essa ligação (inicialmente com fios e posteriormente com transmissores sem fio).

Autoavaliação

1. O que é um "quadro" em uma transmissão serial?
2. Quais os bits que compõem um quadro no padrão RS232?
3. Como seria a string de configuração para uma transmissão com 8 bits de dados, com bit de paridade par e 2 stop bits?
4. A string 7N1 especifica qual configuração de bits?
5. Por que é necessário o uso de circuitos auxiliares como o MAX232 no padrão RS232?

Referências

TORRES, Gabriel. **Hardware**: curso completo. 3. ed. São Paulo: Axcel Books, 1999.