

Laboratório 06

Objetivos:

- Usar a entrada RS-232 do nosso kit para receber dados de um outro periférico RS-232 (no caso, de um computador)
- Se familiarizar com um novo padrão.

Descrição:

Você fará com que o nosso kit esteja sempre pronto para receber caracteres pela porta RS-232, onde o código do caractere recebido é mostrado nos LEDs discretos (ou nos displays de 7 segmentos, dado que seu componente já está pronto).

Implementação:

Basicamente a comunicação serial RS-232, possui a seguinte padronização: O sistema fica em *idle*, com nível lógico '1'. Assim, o primeiro bit chama-se *start bit* e é obviamente em '0'. Ele sinaliza o início da comunicação;

- Os próximos (5, 6, 7 ou) 8 bits são de dados, sendo que o menos significativo é o primeiro a ser enviado, na sequência do *start bit*;
- O (7º, 8º, 9º ou) 10º bit é o de paridade, quando existente. A paridade pode ser par (*even parity*), sendo '1' se o número de '0's for par; ímpar (*odd parity*), sendo '1' se o número de '0's for ímpar; ou sem paridade (*none*);
- O (8º, 9º, 10º ou) 11º bit é chamado de *stop bit* e será sempre '1'. A novidade é que o *stop bit* pode ter a duração de 1, 1.5 ou 2 bits.

No nosso caso, usaremos o chamado "8-N-1", como é conhecida a configuração de palavra de 8 bits, paridade none e 1 bit de stop.

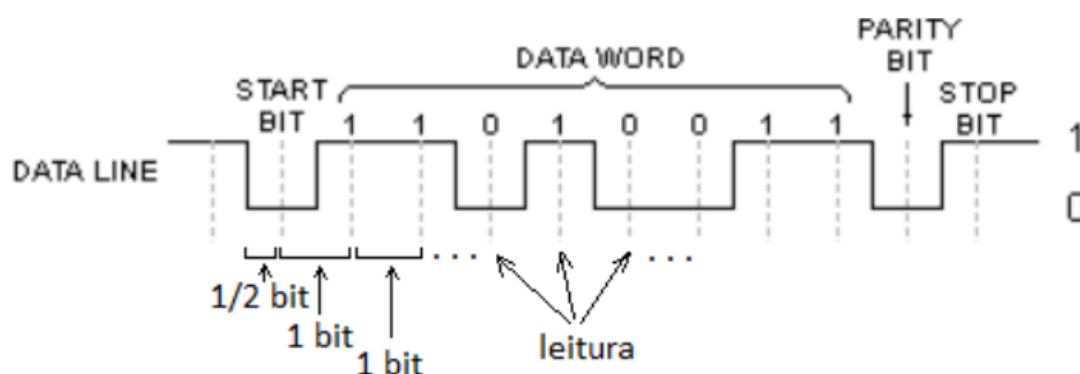
A outra questão importante é sabermos em que velocidade de transmissão trabalharemos. O padrão admite várias possibilidades de baud rate, como é chamada a taxa máxima de troca de bits por segundo. As mais comuns geralmente suportadas são 75, 110, 300, 1.200, 2.400, 4.800, 9.600, 19.200, 38.400, 57.600 e 115.200

bits/segundo. Na nossa implementação, para padronizarmos os testes, vamos usar por convenção a taxa de 115.200 bps (bps = notação mais usada).

Observe a descrição do padrão no Capítulo 7 do User Guide do nosso kit. Este laboratório tem uma simplificação, uma vez que usaremos *parity none*, então seu código não precisará ter tratamento de erro.

Repare que ao contrário da implementação do laboratório anterior (PS/2), o padrão RS-232 não traz um sinal de clock com transição sincronizada com a metade do bit de informação, onde o dado está absolutamente estável para ser lido – aquilo era mesmo uma molezinha, não é mesmo? 😊...

Neste laboratório, a partir da identificação (*falling_edge*) do *start bit*, você terá que fazer o circuito aguardar até a chegada do meio do *start bit*, para depois passar a contar o tempo em que se fará a leitura de cada bit. Repare que depois de aguardar por meio bit, seu contador poderá ser usado para aguardar o tempo do bit inteiro, migrando assim do meio de um bit ao meio do próximo bit – sempre lendo o dado no seu momento mais estável.



(fonte da figura: Wikipedia)

Para testar, será usado um aplicativo emulador de terminal, bem como um outro programa será usado para o redirecionamento da porta serial para uma porta USB do computador.

Importante:

O nosso emulador de terminal, assim como vários outros, sempre manda um caractere de comando após o caractere enviado. É muito típico ser o CR (carriage return) ou ser o chamado LF + CR (line feed + carriage return). Usaremos configurado para a primeira opção, daí o seu circuito precisar armazenar (e mostrar) o caractere que vem antes do código ASCII número 13₁₀.