Laboratório Camada de Enlace

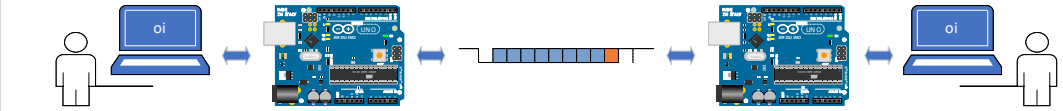
SSC0142 - Redes de Computadores

Prof. Kalinka Regina Lucas Jaquie Castelo Branco

Entrega: 28/05/2025

Objetivo

O objetivo do trabalho é realizar uma comunicação síncrona entre duas plataformas Arduino[1].



Neste trabalho, o Arduino emissor receberá um caractere pela sua porta serial (através do *Serial Monitor* da plataforma) e deverá enviar de forma serial e síncrona ao Arduino receptor, que imprimirá os caracteres recebidos em sua porta serial, exibindo para o usuário através do *Serial Monitor*. Os caracteres deverão ser transmitidos de acordo com o código ASCII (`https://upload.wikimedia.org/wikipedia/commons/d/dd/ASCII-Table.svg`)

Requisitos

- A comunicação entre as duas plataformas deverá ser síncrona, ou seja, deve haver uma sincronização entre o emissor e o receptor através de um *clock*;
- A comunicação deverá implementar o controle de erros através de um bit de paridade;
- A comunicação deverá implementar um *handshake* simples entre o emissor e o receptor.

O Clock

Para a geração do *clock*, poderá ser utilizado o temporizador. Um temporizador ou *timer* é uma estrutura no μC que permite a contagem de tempo a partir de um *clock* interno ou externo.

[1] `https://www.arduino.cc/`

O uso do temporizador é vantajoso porque (1) é feito pelo microcontrolador em *background*, e seu programa pode processar outras coisas enquanto isso e (2) é bastante preciso.

Quando o tempo programado no temporizador é atingido, ele gera uma **interrupção** no μC, o qual interrompe a execução do programa e executa a **rotina de interrupção** associada àquele evento.

Na atividade, o TIMER1 já está pré-configurado no arquivo `Temporizador.h`. Para utilizá-lo, é necessário incluir o arquivo .h no programa. Os métodos disponíveis são:

- `void configuraTemporizador(int baud_rate)`
  (configura o *timer* para interromper `baud_rate` vezes por segundo)
- `void iniciaTemporizador()`
- `void paraTemporizador()`

A rotina de interrupção do TIMER1, ou seja, o método que é chamado toda vez que o *timer* é interrompido, é o `ISR(TIMER1_COMPA_vect)`. Dentro dessa rotina deverá ser inserido o código a ser executado a cada interrupção do *timer*.

Controle de paridade

Deverá ser realizado com uma das possibilidades: paridade *par* ou *ímpar*. Se a paridade é **par**, deve haver um número par de '1's na transmissão; se a paridade é **ímpar**, deve haver um número ímpar de '1's na transmissão. O bit de paridade é utilizado para atingir essa paridade.

Por exemplo, se for ser transmitido o caractere: 'o' (0x6F ou 0110 1111b), em paridade ímpar, o bit de paridade deverá ser '1', uma vez que há 6 '1s' no dado, um número par. No caso de paridade par, já existem 6 '1s', então o bit de paridade será 0.

Handshake

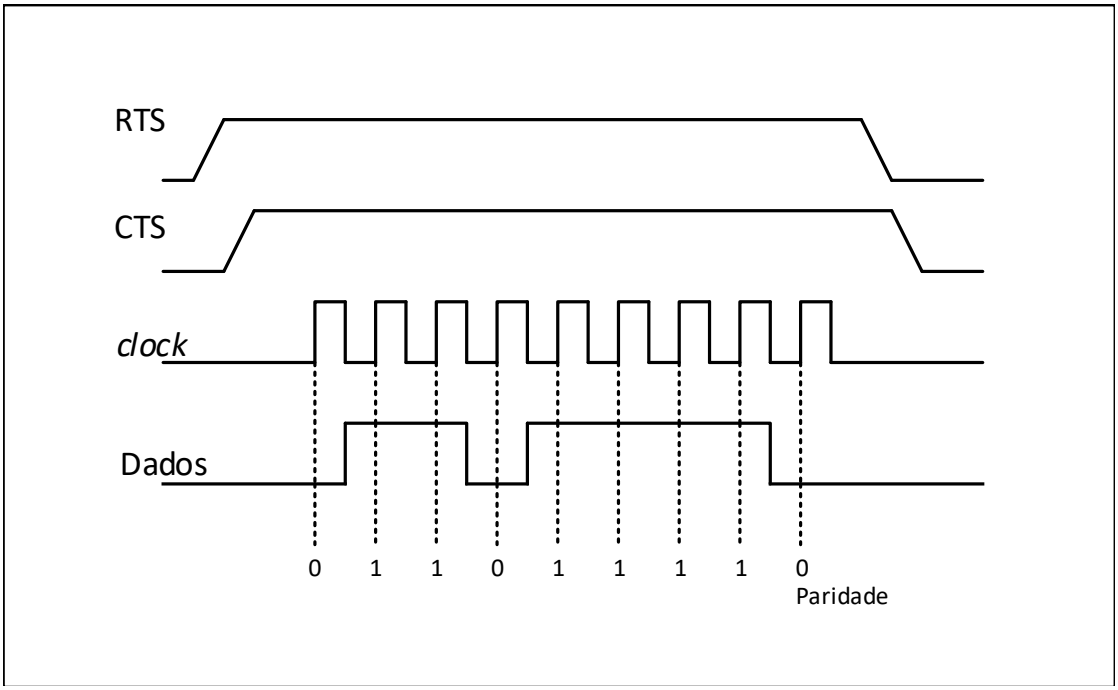
O *handshake* é utilizado como forma de controle de fluxo, como um acordo entre emissor e receptor que a transmissão pode ocorrer. Neste trabalho, serão utilizados dois sinais para realizar o handshake: RTS (do emissor para o receptor) e CTS (receptor para transmissor).

Exemplo dos sinais em uma comunicação:

Na figura abaixo, é exemplificada a transmissão do caractere 'o' (0x6F), conforme as etapas abaixo:

1. O emissor seta o RTS para 1 e espera que o receptor sete o CTS;
2. O receptor, ao perceber o RTS em 1, seta o CTS;

3. O emissor, ao perceber o CTS em 1, inicia a transmissão com o *clock*;
4. a cada *transição de clock* (no exemplo, a transição de 0 para 1), o receptor lê o dado colocado no pino de comunicação;
5. No fim da recepção, o emissor pára o clock e limpa o RTS para 0;
6. O emissor, ao perceber que o RTS foi limpo, toma duas ações:
   - Verifica o bit de paridade e, caso correto, imprime o dado na tela;
   - Coloca o CTS em zero.
7. O emissor, ao perceber o CTS em zero, dá a comunicação como finalizada e espera novo dado no serial monitor para transmissão.



O que entregar

O grupo deve entregar o arquivo do "receptor ou transmissor".ino e também um vídeo de no máximo 3 minutos mostrando o funcionamento do trabalho. Todos os participantes devem participar do vídeo.