

Jogo: *Super Bit Smasher*

Escreva um programa que implemente um jogo com as seguintes características:

- Existem 3 botões, um para cada operação lógica (AND, OR, XOR). O OR estará sempre disponível, mas a disponibilidade do AND e XOR dependerá de um valor aleatório. O mapeamento dos botões será o seguinte: AND-*pino 4*, OR-*pino 3*, XOR-*pino 2*;
- O jogo começa por gerar dois valores de 8 bits: o *target* e o ponto de partida;
- Em cada *round* de jogo, deve-se ler uma cadeia de caracteres numérica via porta série **correspondente a um inteiro decimal**, convertê-la para tipo inteiro (**ver nota no final do enunciado**) e aplicar a operação lógica selecionada pelo botão ao valor do ponto de partida, gerando um novo valor;
- Quando o valor atingido for igual ao *target* (ou se o tempo expirar), termina o jogo;
- Existem 4 LEDs, que mostram quanto tempo resta (cada um simboliza $\frac{1}{4}$ do tempo limite, ligados nos pinos digitais 8 a 11).

Modalidade de jogo.

A. Início do jogo:

- São gerados dois números aleatórios de 8 bits, **convertidos em binário e apresentados ao jogador**: o *target* e o ponto de partida;
 - É também gerado um número aleatório entre 1 e 3 (inclusivé), que determina se as operações AND e/ou XOR estarão disponíveis durante o jogo, pela seguinte regra:
 - bit 2 ativo -> AND disponível; bit 1 ativo -> XOR disponível.
- O OR estará sempre disponível. **O jogador será notificado das operações disponíveis.**

B. Em cada *round* (o jogo deverá permitir *rounds* sucessivos, com tempo limite):

- O jogador deverá inserir um número (**em decimal**), premindo *Enter*. De seguida, o número inserido deverá ser mostrado ao jogador, em binário;
- Quando um dos botões ativos for premido, o valor inicial será atualizado, aplicando-se o operador selecionado e número inserido. **O novo valor será impresso;**

O jogo terminará quando o jogador transformar o valor inicial no valor *target*, ou se o tempo expirar (armazenado numa variável e definido pelo programador), recomeçando de seguida.

Nota: para este trabalho, não se exige a implementação da funcionalidade de *debouncing*.

Devem recorrer a funções e modularizar o vosso código tanto quanto possível.

Poderão visualizar uma demonstração (sem os LEDS) em: <https://youtu.be/W2ib-gOITYw>

O trabalho deverá ser realizado de forma **individual**, e submetido no Inforestudante até dia 16/12, pelas 23:59. **A submissão deverá incluir:** o código desenvolvido; um diagrama do circuito (feito em *tinkercad* ou outra ferramenta – **não é pedida uma implementação em tinkercad**); uma foto/vídeo do circuito montado; um relatório breve documentando a implementação (máx. 2 páginas de texto – figuras e excertos de código não serão contabilizados para o limite). Este trabalho estará sujeito a defesa, caso se julgue necessário.

Sobre a avaliação: Quaisquer situações irregulares serão tratadas em consonância com os regulamentos em vigor na UC.

Rudimentos necessários: leitura de dados via porta série e alguns conselhos práticos

(experimentar no Arduino – não utilizar o Tinkercad)

De modo recíproco à escrita via porta série (*Serial.print*, *Serial.println*), também é possível fazer leituras de cadeias de caracteres a partir da porta série. Por exemplo:

```
String a;
int valor;

void setup() {
    Serial.begin(9600);
    Serial.println("Introduzir nome");
}

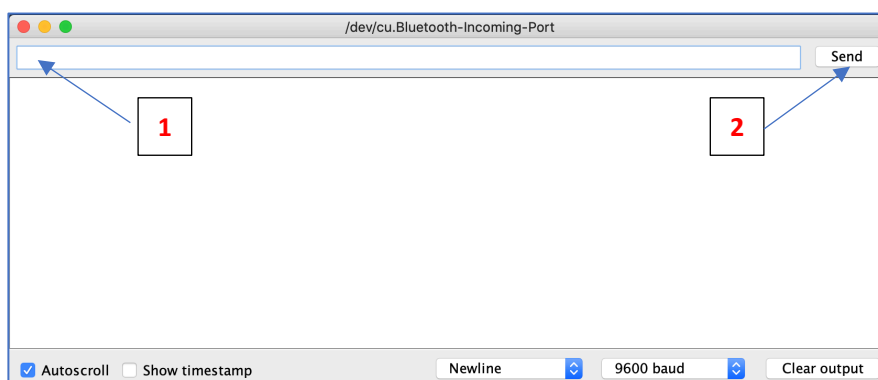
void loop() {

    // Ler uma string terminada por Enter
    // Primeiro passo: verificar se existem dados na porta serie para receber, com o Serial.available()
    if (Serial.available() > 0)
    {
        // Segundo passo: ler os dados utilizando o readStringUntil (o que sera o '\n')
        a= Serial.readStringUntil('\n');
        Serial.println("O seu nome e:"+a);

        //Conversao para inteiro, demonstracao
        valor=a.toInt();
        valor=valor+1;
        Serial.println(valor);

        //A funcao String() converte numeros em cadeias de caracteres
        //o segundo parametro, que e opcional, permite especificar a base (como no println)
        Serial.println("O valor binário e:"+String(valor,BIN));
    }
}
```

Este exemplo de código lê uma cadeia de caracteres a partir da porta série, que é armazenada na variável **a**, do tipo *String*. Esta variável pode ser convertida num inteiro, utilizando o método **toInt()**. Sempre que no *Serial Monitor* introduzir uma cadeia de caracteres no campo **1 (ver figura)** e premir a tecla **Enter** ou o botão **Send** (ver passo “2”), a cadeia de caracteres será enviada para o Arduino.



NOTA MUITO IMPORTANTE: no *Tinkercad*, a função *String()* tem um bug com alguns anos, que nunca foi corrigido. As conversões de inteiro para cadeia de caracteres feitas por esta função dão frequentemente erros, quanto utilizadas no simulador (por exemplo, *String(124)* dá “122” como resultado).