

Relatório de Tecnologia da Informática

Jogo: Conversão-Relâmpago



UNIVERSIDADE D
COIMBRA

André Filipe Costa Colaço

UC - 2020220301

Introdução

O objetivo deste jogo consiste em gerar um número aleatório, convertê-lo para binário e apresentá-lo nos 5 LED's. O jogador deve premir o botão sucessivamente, fazendo avançar o contador apresentado no monitor, para tentar adivinhar o número que os LED's indicam em binário. Sempre que o número chegar a 32, este retorna a 1.

Número em binário

Para passar o número gerado aleatoriamente para binário e, de seguida, para os LED's criei uma função chamada "gerarbinario". Para lermos o último bit de um número, podemos utilizar operadores bitwise. Para ler os outros bits tenho de fazer sucessivos Shift's para a direita e ler o último bit após cada Shift. Ou seja, nesta função implementei 5 condições, uma para cada LED (cada bit). Por exemplo, na primeira condição, se o último bit for 1 acende o LED no pino 10 e se for 0 este não acende. Na segunda condição, faz um Shift para a direita e se o último bit desse número for 1 acende o LED no pino 9 e se for 0 não acende. Para os outros LED's é de forma análoga.

Uso do botão

Para poder usar o botão, tive de implementar no meu código o debounce e criei uma função chamada "debounce" para tal. Ora, quando o botão é premido, este devolve-nos valores de HIGH para LOW e LOW para HIGH, alternadamente. Isto faz com que, quando o botão seja premido uma vez, o programa assuma mais leituras, ou seja, em vez de aparecer uma vez que o botão é premido aparece dez vezes.

Para resolver isto, é usado o debounce. Primeiro, é lido o estado do botão. Depois, se este estado mudar, o tempo do debounce é resetado. Posteriormente, se o estado do botão não mudar durante 50 milissegundos e se o estado do botão mudar, então o estado anterior deste passa a ser o atual. Caso o estado do botão esteja em HIGH, então o contador vai incrementando um (se este chegar a 32 volta para um) e imprime o resultado deste. Posto isto, a leitura do botão é guardada para o próximo loop.

Reset

Para implementar o reset, pus o código dentro da função do debounce. Se o botão for premido, ou seja, o estado do botão estiver em HIGH e se o estado do botão não mudar durante um segundo então o jogo recomeça.

Vitória/Derrota

Para ver se o jogador acertou ou não no número, criei uma função chamada “resultado”. Caso passem 15 segundos desde o começo ou recomeço do jogo, então vão ser verificadas duas situações:

- ✓ Se o número que o jogador pôs através do botão for igual ao número gerado aleatoriamente, então é impressa uma mensagem de parabéns e, de seguida, é executado o efeito de varrimento dos LED's. Para tal, criei um ciclo, que desliga todos os LED's e depois criei outro ciclo para acender os LED's, sequencialmente, com um delay de meio segundo.
- ✓ Se o número for diferente do número gerado, então é impressa uma mensagem a dizer que erraste e, também, a dizer qual tinha sido o número gerado aleatoriamente.

Independentemente do resultado do jogo, ou seja, se o jogador ganhou ou não, este volta a recomeçar o jogo.

Conclusão

Para acabar, achei o trabalho mais difícil do que parece. A parte mais complicada, para mim, foi mesmo no código, mais especificamente no debounce. Contudo, fez com que pensássemos sobre as várias maneiras de fazer este jogo. Quando acabei de escrever o código, depois de muitas tentativas, até achei interessante e útil o jogo.