

Alunos:	Nota:
1 -	
2 -	
3 -	Data:

Atividade Prática 2

Implementação de um jogo de memória.

1. Objetivo:

Modelagem e implementação de um jogo de memória em forma de sequência luminosa para divulgação do curso de Eng. Elétrica.

2. Descrição do Problema:

Na **Figura 1** apresenta-se o esquema do jogo a ser implementado e na **Figura 2** apresenta-se o protótipo montado.

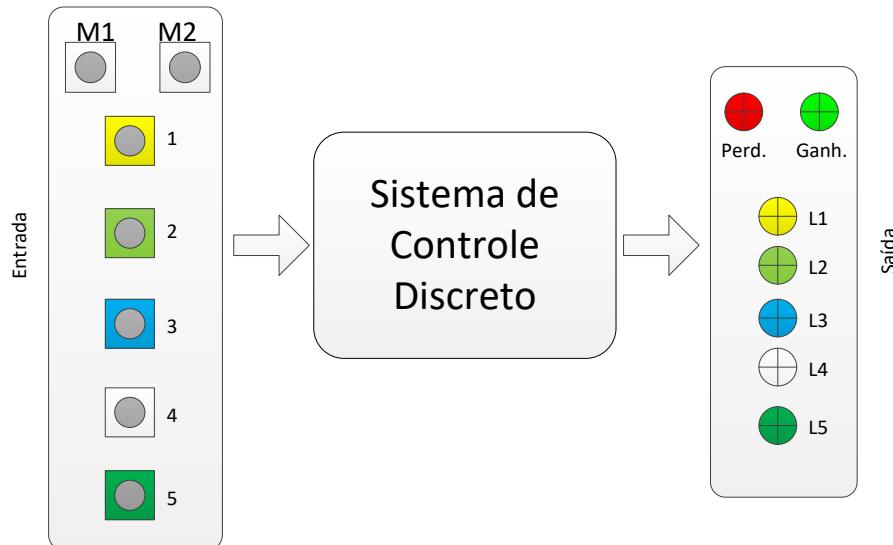


Figura 1: Esquema de implementação do jogo da memória.



Figura 2: Protótipo montado.

O jogo consiste em reproduzir através de 5 botoeiras sequenciais luminosas apresentadas por um conjunto de 5 leds correspondentes a cada botão.

Neste jogo serão implantados dois modos de operação, os quais são descritos a seguir:

Modo 1: A cada rodada apresenta-se um número fixo de sequências, mudando apenas a velocidade com que as sequências são apresentadas

Modo 2: A cada rodada acrescenta-se um elemento na sequência, mantendo fixo o tempo que um LED é iluminado.

3. Atividades a serem desenvolvidas:

O botão M1 faz que o jogo inicie pelo modo 1 e o Botão M2 inicia o jogo pelo modo 2. A seguir descreve-se como cada modo deve funcionar.

Modo 1: Para efeitos didáticos serão implementadas apenas **duas sequências de 4 sinais luminosos**, em que a primeira sequência apresenta-se de uma forma lenta, enquanto que a segunda sequência se apresenta de uma forma mais rápida.

A lógica de controle deve atender às seguintes especificações:

1. Para dar início ao jogo, o botão “M1” deve ser pressionado.
2. Logo após o passo anterior deve ser apresentada a primeira sequência predeterminada de 4 sinais luminosos. Cada sinal luminoso deve durar 500 ms ligado. Após desligado um sinal, não precisa esperar para ligar o próximo sinal.
3. Após a etapa anterior o jogador deve pressionar a mesma sequência que lhe foi apresentada. Caso a sequência digitada estiver errada, o sinal luminoso “perdeu” deve acender e jogo termina. Caso contrário passa para a próxima etapa.

4. Considerando que o jogador acertou a primeira sequência, deve ser apresentada a segunda sequência predeterminada de 4 sinais luminosos. Cada sinal luminoso deve durar 300 ms ligado.
5. A seguir o jogador deve pressionar a mesma sequência que lhe foi apresentada. Caso a sequência digitada estiver errada, o sinal luminoso “perdeu” deve acender e o jogo termina, caso contrário, o sinal “ganhou” deve acender.
6. Em qualquer situação o botão “M1” ou “M2” deve levar o sistema para o estado inicial, mesmo que o jogo tenha ficado sem concluir.

Modo 2: Para efeitos didáticos serão implementadas **apenas 4 sequências**, começando o primeiro com um caractere, a segunda com dois caracteres, e assim sucessivamente até a quarta sequência de 4 caracteres. Considerar um tempo de iluminação de cada LED fixo e igual a 300 ms.

A lógica de controle deve atender às seguintes especificações:

1. Para dar início ao jogo, o botão “M2” deve ser pressionado.
2. Logo após o passo anterior deve ser apresentado o primeiro sinal luminoso que deve durar 300 ms.
3. A seguir o jogador deve pressionar o botão correspondente. Caso o botão estiver errado, o sinal luminoso “perdeu” deve acender e jogo termina. Caso contrário passa para a próxima etapa.
4. Considerando que o jogador acertou uma sequência, deve ser apresentada a próxima sequência acrescentando mais um sinal luminoso a cada rodada. Considere que cada nova rodada contém a sequência do passo anterior mais um novo sinal.
5. Em caso de o jogador acertar 4 sequências consecutivas, o sinal luminoso ganhou deve acender.
6. Sempre que um botão for pressionado na sequência errada o sinal luminoso “perdeu” deve acender e o jogo termina.
7. Em qualquer situação o botão “M1” ou “M2” deve levar o sistema para o estado inicial, mesmo que o jogo tenha ficado sem concluir.

Desenvolver e implementar uma lógica de controle discreto utilizando autômatos. A implementação pode ser por autômatos de Mealy ou Moore. Deverão ser apresentados o projeto do autômato e a implementação em microcontrolador.

A lógica desenvolvida deve ser somente por autômatos, ou seja, não será aceito como válido se for utilizado comandos adicionais de lógica de programação (**if, case, for, while**) para contornar problemas não resolvidos pela lógica de autômatos.