

Projeto final EA871 - Proposta

Thiago M. Pavão – 247381, Vinícius Esperança Mantovani – 247395

Introdução

A proposta de projeto é implementar o jogo *Simon*, conhecido pelo produto físico da Figura 1. O objetivo do jogo é acertar a sequência de cores mostradas. A sequência incrementa seu tamanho em um quando o jogador consegue inserí-la com sucesso, começando em 1 de tamanho e aumentando até que a memória do jogador falhe.



Figura 1: Jogo Simon, figura retirada de [1]

O jogo conhecido tem por padrão 4 botões/cores, que podem acender e formar a sequência, nossa ideia é fornecer a opção para o usuário de selecionar o mapa de jogo, podendo selecionar entre o clássico (4 posições), 5 posições ou até mesmo 9, para os mais experientes. A escolha do mapa será feita antes do começo da partida.

Periféricos

A interação do usuário com o jogo será via controle remoto de televisão, o sinal será detectado com o sensor da Figura 3, o sinal decodificado segue o protocolo de transmissão NEC [2], que se baseia nos tempos de diversos pulsos. Para medir os tempos utilizaremos o módulo TPM com a função *Input Capture*. Isto será feito por interrupção quando há mudança no sinal do sensor, ou seja, nas bordas de subida e de descida. Alguns testes foram feitos e conseguimos ler o sinal com sucesso.

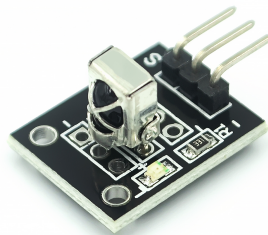


Figura 2: Sensor receptor de infra-vermelho, utilizado para ler o sinal do controle remoto

A visualização do jogo será feita por uma matriz de 8x8 LEDs (Figura 3), controlada por MAX7219. A comunicação com o periférico é feita por intermédio do módulo SPI da placa que utilizamos no laboratório. Esta comunicação é serial e síncrona, a comunicação com MAX7219 resulta em escrita nos seus registradores internos, os dados são enviados em pacotes de 16 bits, contendo o endereço do registrador e o valor que se deseja escrever nele. Inicialmente deve ser feita uma inicialização, escrevendo em determinados registradores, e posteriormente, para controlar os LEDs basta escrever em 8 registradores, cada um controlando uma linha da matriz de LEDs. A especificação da comunicação em detalhes será feita seguindo o datasheet do MAX7219 [3].

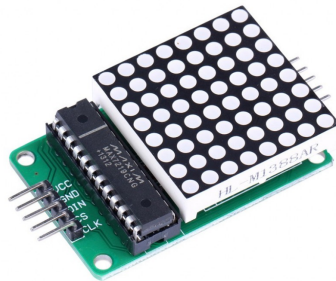


Figura 3: Matriz de LEDs, controlada por MAX7219

Por fim, uma parte importante do jogo é o som, emitido sempre que uma posição é acesa, tanto quando a sequência está sendo apresentada, como quando o usuário está inserindo. Para isto, será utilizado um buzzer passivo, igual ao utilizado no Roteiro 8.

Máquina de Estados

A máquina de estados geral da aplicação consiste de 4 macro estados: *INÍCIO*: Aplicação em stand-by até que ocorra interação do usuário. *SELEÇÃO DE MAPA*: Seleção do mapa (dificuldade) do jogo, dentre as três opções: 4, 5 e 9. *JOGO*: Jogatina em si, em que a sequência é mostrada e deve ser repetida pelo jogador. *RESULTADO*: Mostra para o jogador o número máximo de posições memorizadas por ele, ou seja, o número de posições da última sequência que foi inserida corretamente por completo. Estes estados podem ser vistos na Figura 4.

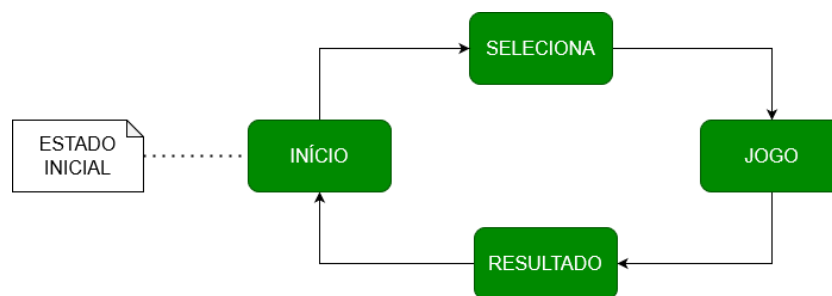


Figura 4: Estados gerais da aplicação

Cada um destes macro estados é composto de estados menores, que serão detalhados a seguir.

INÍCIO

Este macro estado é responsável por mostrar na tela alguma mensagem de inicialização, como “START”, isto será feito na *main*, piscando letra por letra na matriz de LEDs, em loop. Isto continua até que alguma borda seja detectada pelo canal do TPM do sensor IR, que altera para o estado de leitura e realiza a leitura do sinal. Caso haja erro, como um sinal que não era do controle remoto ou um sinal que não pôde ser processado, o estado é retornado para o da mensagem, caso contrário (Leitura bem sucedida do sinal do controle), o fluxo segue para o estado de seleção de mapa. O diagrama deste estado é simples, e pode ser visto na Figura 5.

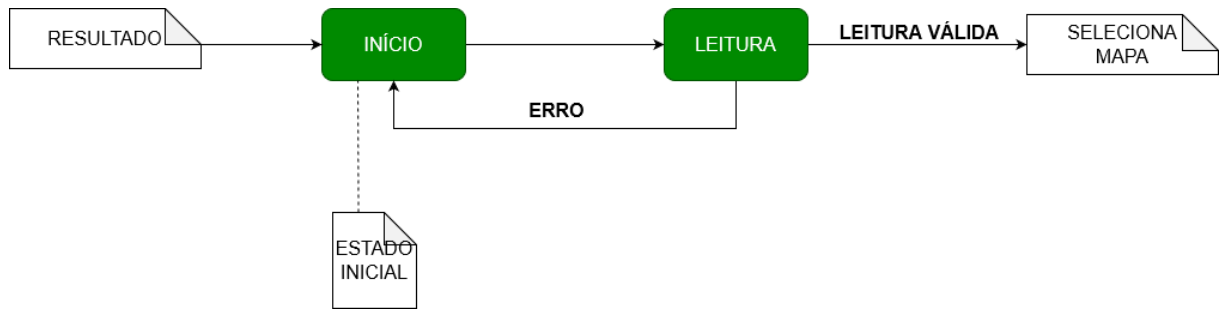


Figura 5: Estado *INÍCIO*

SELEÇÃO DE MAPA

Aqui, o usuário deve escolher o mapa em que deseja jogar, na realidade, o mapa será sempre uma grade 3 por 3, com 9 espaços, mas a seleção permite que o usuário escolha quais quadrados podem acender, e assim fazer parte da sequência. Cada quadrado do mapa é associado a um número, conforme especificado pela Figura 6, portanto cada opção de mapa consiste em um vetor com alguns números, contendo as posições válidas para aquela opção. Isto é utilizado para desenhar o mapa no momento da seleção, enquanto alguma variável armazena o mapa atualmente selecionado.

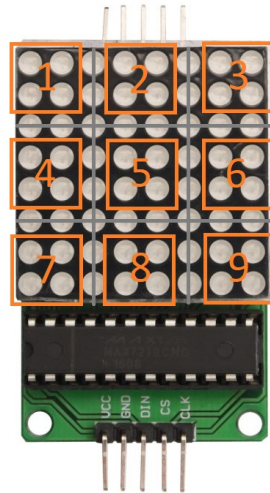


Figura 6: Identificação das posições do tabuleiro.

O usuário pode trocar o mapa selecionado utilizando as setas do controle remoto, quando satisfeito com a decisão, basta pressionar o botão OK para confirmar e começar a jogar. Para implementar este funcionamento, são necessários os estados da Figura 7.

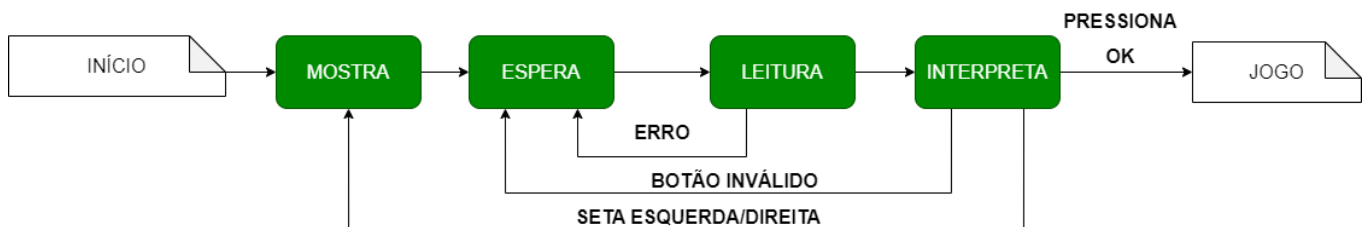


Figura 7: Estado *SELECIONA MAPA*

O estado MOSTRA é responsável por desenhar na matriz de LED o mapa selecionado, assim que isto é feito, o estado muda para ESPERA, onde nada é feito até que haja uma interrupção indicando que houve um possível pressionamento no controle remoto. Quando isto ocorre, o estado é alterado para LEITURA, onde a recepção do sinal é feita. Caso ocorra algum erro, o estado retorna para o de ESPERA, já que não é necessário redesenhar a tela, se o sinal for reconhecido, há duas possibilidades: Se o pressionamento foi da tecla da esquerda ou direita, basta atualizar a variável que armazena o mapa selecionado e retornar ao

estado MOSTRA. Se o pressionamento foi do botão OK, então o fluxo segue para o macro estado de JOGO. Por fim, se o pressionamento foi de algum outro botão, nada precisa ser feito, e o estado é retornado para ESPERA até que outro pressionamento seja feito.

JOGO

Primeiramente, o jogo deve ser inicializado, configurando vetores e variáveis que serão utilizados para controlar o fluxo do jogo, como por exemplo a sequência atual e a quantidade de itens na sequência, esta inicialização será feita no estado INICIALIZA e fará com que a sequência seja iniciada apenas com um item (primeira posição), escolhido aleatoriamente. Agora inicia-se um ciclo de mostrar a sequência para o usuário e receber os pressionamentos verificando se a sequência inserida por ele está correta. Se em algum momento for realizado um pressionamento errado, o estado é alterado para o macro estado RESULTADO. Se toda a sequência for repetida corretamente, mais um item é adicionado à ela e ela é novamente mostrada, reiniciando o ciclo. Podem ocorrer situações específicas enquanto os sinais do controle remoto são recebidos, mas estas são análogas ao que já foi visto anteriormente e pode ser visto na Figura 8.

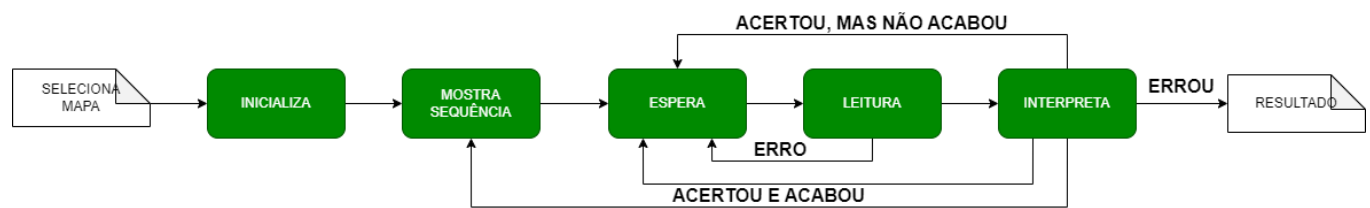


Figura 8: Estado *JOGO*

RESULTADO

Este é o macro estado mais simples, ele é responsável por mostrar na tela o número de posições da última sequência inserida corretamente por completo pelo usuário, ou seja, sua pontuação no jogo. O número é mostrado digito a digito, piscando os caracteres na matriz de LEDs. Para isso, é necessário apenas um micro estado, como pode ser visto na Figura 9.



Figura 9: Estado *RESULTADO*

Máquina de estados completa

Substituindo os macro estados da Figura 4, podemos visualizar a máquina apenas em seus micro estados, estes são os estados que serão realmente implementados na máquina da aplicação. O diagrama completo pode ser visto na Figura 10.

Diagrama de Componentes

Além dos diagramas da máquina de estados, fizemos também o diagrama de componentes para termos uma ideia melhor de como estruturaremos nosso projeto. Assim, conseguiremos organizá-lo de modo mais rápido e inteligível, conforme a Figura 11.

Referências

[1] Página da wikipédia do jogo Simon

[https://en.wikipedia.org/wiki/Simon_\(game\)](https://en.wikipedia.org/wiki/Simon_(game))

[2] Procolo de transmissão NEC

<https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>

[3] Datasheet do MAX7219

<https://www.analog.com/media/en/technical-documentation/data-sheets/max7219-max7221.pdf>