Relatório do 1º Trabalho prático individual

Objetivo: Criar um programa que represente o jogo Whack-a-LED, deve acender aleatoriamente um LED e, o jogador deve pressionar o botão correspondente ao LED aceso. O jogo continua até que todos os LEDs sejam acionados corretamente.

1. Estrutura e Inicialização

- O código define variáveis para controlo do botão, LEDs, e tempos de resposta, como buttonPin, blinkDelay, delayLED, entre outras. O botão é configurado no pino 9 com INPUT_PULLUP, e os LEDs são mapeados entre os pinos 2 e 8.
- No setup(), o código inicializa todos os pinos de LEDs como saídas e também os desliga. O gerador de números aleatórios é ativado com randomSeed(analogRead(0)).

2. Função Principal (Loop)

- No loop principal, o código verifica se o botão foi pressionado por 2 segundos para iniciar ou reiniciar o jogo com a função primir2sec().
- Se o jogo já estiver iniciado, ele entra na função ronda(), que gerencia o acendimento aleatório dos LEDs e a interação com o botão.

3. Funções do Jogo

- **Iniciar e Reiniciar o Jogo**: Quando o jogo começa, a função startGame() faz os LEDs piscarem três vezes para sinalizar o início, e escolhe aleatoriamente um LED para acender. Caso o jogo seja reiniciado, a função resetGame() apaga todos os LEDs e faz eles piscarem duas vezes para indicar a reinicialização.
- **Rondas e Interações**: A função ronda() alterna entre acender LEDs aleatórios e verificar se o botão foi pressionado no momento correto. Se o jogador pressionar o botão no LED correto, ele é marcado como "acertado". Se todos os LEDs forem acionados corretamente, o jogo é vencido e todos os LEDs piscam três vezes.
- **Controle de LEDs e Debouncing**: A lógica de controle de LEDs envolve alternar entre acender e apagar LEDs com as funções toggleAllLEDs() e apagarAtivo(). Além disso, o botão tem um controle de debounce para evitar múltiplas leituras de um único pressionamento, com a função debounceDelay.

4. Lógica de Vitória

- A vitória é verificada com a função checkVictory(), que retorna true se todos os LEDs forem acionados corretamente, ou seja, se todas as variáveis de acerto (como acert2, acert3, etc.) forem true.

5. Funções de Auxílio

- **Controle de LEDs**: As funções blinkAllLEDs() e toggleAllLEDs() permitem piscar todos os LEDs de forma sincronizada. A função selectRandomLED() escolhe aleatoriamente um LED entre os pinos 2 e 8 para acender.
- **Controle de Ações do Jogador**: Quando o jogador pressiona o botão, a função meteAtivo() marca o LED como "acertado". A função acertou() verifica se o LED foi corretamente pressionado.

Conclusão

Este trabalho teve como base criar um código onde conseguíssemos conciliar ligar leds, controlar se estavam ligados ou não, e criar um algoritmo parra os manter ligados enquanto eles iam mudando aleatoriamente, sempre em atenção se o jogador quisesse reiniciar o jogo apertava o botão 2 segundos. O mesmo termina quando todos os leds forem acertados, ou seja, estiver tudo ligado.

Alexandre José Martins Rodrigues

2022249408