

Desafio - Controle de LEDs com ESP32 e Operadores Bitwise

1 Descrição da Atividade

Nesta atividade deve-se realizar a implementação de um software embarcado utilizando a plataforma ESP32 e o simulador Wokwi¹. O sistema embarcado deve possuir quatro LEDs, ligados a pinos diferentes da ESP32, e um botão, ligado a um outro pino da ESP32. Os pinos ligados aos LEDs devem ser configurados como saída digital e o pino associado ao botão deve ser configurado com a função de entrada digital.

Durante o funcionamento do sistema, cada LED terá um estado que deverá ser armazenado (ou seja, se ele está ligado ou desligado). Com base nesses valores de estado, o software deve configurar as saídas digitais para acender ou apagar os LEDs de acordo com seus estados individuais.

Caso o usuário clique no botão, enquanto o botão estiver apertado, todos os LEDs devem permanecer apagados. Ao soltar o botão os estados dos LEDs podem ter sido atualizados e eles irão ser ligados ou desligados a depender do estado atual. Para gerar uma modificação pseudo-aleatória nos estados dos LEDs, pode-se ficar incrementando a variável que guarda os estados dos LEDs (ou seja, somando 1 a ela) enquanto o botão estiver pressionado. Dessa forma, o valor final dessa variável será diferente a depender do tempo que o usuário permaneceu pressionando o botão.

Outra parte importante do desafio é conseguir utilizar apenas uma variável inteira de 8 bits para armazenar o estado de todos os quatro LEDs, visando obter economia no uso de memória. Para isso, é necessário utilizar operadores bitwise para ter acesso a bits individuais da variável.

Por exemplo, considerando que uma variável inteira de 8 bits (chamada de v) possui o valor 5 em decimal atribuído a ela, logo a representação em binário seria 000000101. Portanto, se usarmos os quatro primeiros bits (os bits são analisados da direita para a esquerda) para representar os estados dos quatro LEDs, teríamos como resultado o LED 1 e o LED 3 acesos e os LEDs 2 e 4 apagados. Caso a variável v após ser incrementada passasse a ter o valor em decimal 15, em binário seria 00001111, o que faria com que todos os LEDs ficassem ligados.

Sabendo que a variável possui essa representação em binário, para acessar bits individuais, precisamos “isolar” os bits para que eles sejam lidos. Para isso, utilizamos dois operadores bitwise, um para deslocar bits para a direita ($>>$) e um para fazer um *and* lógico bit a bit (operador $&$). Por exemplo, para acessar o terceiro bit (da direita para a esquerda) da variável v bastaria realizar a operação $((v >> 2) \& 1)$. Primeiro faz-se o deslocamento de dois bits para a direita (o que leva o terceiro bit a ficar na primeira posição da direita para a esquerda) e depois faz-se o *and* bit a bit com o valor 1 (que em binário é 00000001), o seja ao fazer esse *and* bit a bit, apenas o primeiro bit fica inalterado enquanto que todo o resto vai ser zerado. Após isso, basta verificar se o valor resultante da operação é igual a 0 ou a 1 para descobrir qual era o valor do terceiro bit.

Obs: Foi disponibilizado no Classroom um vídeo com mais explicações. Também foram disponibilizados slides com exemplos de código e figuras que vão auxiliar a resolver o desafio.

¹<https://wokwi.com/>

2 Entrega da Atividade

Envie pelo classroom o código fonte gerado para a implementação do desafio e uma imagem com a tela do simulador.