

**Instituição:** Instituto Federal de Educação, Ciência e Tecnologia - Paraíba (IFPB).

**Disciplina:** Microprocessadores e Microcontroladores.

**Professor:** Fagner de Araujo Pereira.

Alunos: \_\_\_\_\_

## Exercícios – Programação de GPIO no STM32F407

**Realize a programação do GPIO no STM32F407, escolhendo corretamente os pinos, para implementar as seguintes tarefas:**

*(\*Utilizar resistores limitadores de  $470\Omega$  para os LEDs externos)*

*(\*Solicite ao professor o acesso à biblioteca do Display LCD)*

- ☐ 1. Piscar um LED numa frequência de 5Hz, com largura de pulso de 100ms.
- ☐ 2. Acionar um *buzzer* para emitir ciclicamente o seguinte sinal sonoro: 1 *beep* curto, aguarda um tempo e, em seguida, 2 beeps curtos, aguardando mais um tempo.
- ☐ 3. Acender dois LEDs com diferentes intensidades de brilho, alterando alternadamente a intensidade em tempo de execução, semelhantemente ao que foi feito na questão 6: Simultaneamente, enquanto um LED estiver aumentando o brilho, o outro deve diminuir o brilho.
- ☐ 4. Apresentar uma contagem binária de 8 bits a partir de 8 LEDs externos. (*obs.: para facilitar a implementação do código, conecte os LEDs a pinos contíguos de um mesmo GPIO e use a função `GPIO_Write_Port()` para acionar simultaneamente todos os pinos de uma porta*).
- ☐ 5. Implementar o efeito do “LED andante” nos 8 LEDs da questão anterior: ative o primeiro LED e faça com que o LED aceso se desloque do primeiro ao último: Ligue o primeiro LED e, após um tempo, apague esse LED enquanto acende o segundo, e assim sucessivamente até o oitavo LED. O único LED aceso em cada passo deve se deslocar de um extremo a outro e em seguida fazer a operação no sentido inverso.
- ☐ 6. Simular o funcionamento de um cruzamento viário com dois semáforos de trânsito usando 6 LEDs coloridos.
- ☐ 7. Apresentar uma contagem hexadecimal (crescente e decrescente) de um dígito em um display de 7 segmentos. A contagem deve incrementar de 0x0 até 0xF e em seguida decrementar até 0x0 novamente, repetindo o processo indefinidamente.
- ☐ 8. Exibir o seu primeiro nome na primeira linha de um display de cristal líquido, o segundo nome na segunda linha, o terceiro nome na terceira e uma contagem regressiva de 10 até 0 na quarta linha, repetindo o processo indefinidamente.
- ☐ 9. Controlar um micro servomotor para que ele alterne suave e lentamente a posição do seu eixo, iniciando na posição de 0° até a posição de 180° e depois retornando à posição de 0°, repetindo o processo indefinidamente.
- ☐ 10. Implementar o controle de velocidade de um motor DC para que ele acelere suavemente até sua velocidade máxima e em seguida desacelere suavemente e repita o mesmo procedimento no sentido contrário, repetindo o processo indefinidamente. Utilize uma ponte H como drive de potência para o motor e uma frequência PWM de 2kHz.

- ☐ 11. Realizar o acionamento de uma carga em corrente alternada (AC) de 220V por meio de um relé eletromecânico, usando um sensor óptico de detecção de objetos: Quando o objeto se aproximar do sensor, a carga deve ser acionada. Quando o objeto se distanciar, a carga deve ser desligada.
- ☐ 12. Realizar o acionamento de um motor de passo bipolar. O eixo do motor deve fazer dois giros completos no sentido horário e depois um no sentido anti-horário, repetindo o processo indefinidamente. Realize o acionamento através de passo completo. (*obs.: utilize duas pontes H como drive de potência*).
- ☐ 13. Reproduzir em duas saídas com dois LEDs os estados de duas entradas com dois *push-buttons*.
- ☐ 14. Utilizar uma entrada com um *push-button* para alternar o estado lógico de uma saída conectada a um LED.
- ☐ 15. Utilizar dois *push-buttons* (K0 e K1) para acionar um LED com a seguinte regra: o LED só deve acender se o usuário pressionar os dois botões, mas primeiro o botão K0 deve ser pressionado e só depois o botão K1.
- ☐ 16. Utilizar dois *push-button* (K0 e K1) para acionar um LED com a seguinte regra: o LED só deve acender se o usuário pressionar os dois botões, mas primeiro o botão K0 e em seguida o botão K1 dentro de no máximo 1 segundo após K0 ser pressionado. Se o tempo for maior que esse, o LED não deve acender.
- ☐ 17. Controlar um micro servomotor para que ele alterne lenta e suavemente a posição do seu eixo entre 0° e 180° por meio de dois *push-buttons*, sendo um para aumentar o ângulo de posicionamento do eixo e outro para diminuir.
- ☐ 18. Utilizar um teclado de membrana 4x4, fazendo sua decodificação, apresentando a última tecla pressionada em um display de 7 segmentos.
- ☐ 19. Fazer a leitura de um sensor de proximidade ultrassônico, semelhante aos sensores de ré e estacionamento de automóveis. Escolher 3 limiares de distância (D1=10cm, D2=20cm e D3=30cm), sinalizando a distância medida por meio de um buzzer. Se a distância medida for maior que 30cm, o buzzer não deve ser acionado. Se a distância medida estiver entre 30cm e 20cm, o buzzer deve emitir *beeps* a uma frequência baixa. Se a distância medida estiver entre 20cm e 10cm, o buzzer deve emitir *beeps* a uma frequência intermediária. Se a distância medida for menor que 10cm, o buzzer deve emitir *beeps* a uma frequência alta.
- ☐ 20. Utilizando 4 botões e 4 Leds externos, implementar o antigo jogo da memória “Genius”. (Para criar aleatoriedade, utilize a função `Random_Number()` da biblioteca `Utility.h`, que retorna um número inteiro sem sinal de 32 bits).