

WP3-CETELI-3-IoT

Atividade Prática	Pushbutton + HC-SR04 + Servo Motor	Data	21 de julho de 2023
Monitor	Diego Alves Amoedo	Aula	04
Professor	Andrey Ruben Rieiro Bessa		
INTERLIGAR OS COMPONENTES PUSHBUTTON + HC-SR04 + SERVO MOTOR			

Sumário

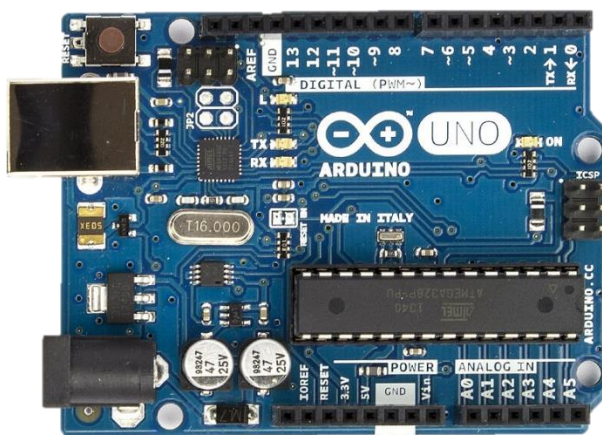
1. OBJETIVOS.....	2
2. MATERIAIS	2
• Arduino MEGA (ou placa compatível).....	2
• Sensor de distância HC-SR04	2
• Cabos de conexão	2
• Servo motor	2
• Botão (pushbutton)	3
3. CONFIGURAÇÃO DO AMBIENTE	3
4. IMPLEMENTAÇÃO DO SCRIPT	3
5. EXPLICAÇÃO DO CÓDIGO.....	6
6. MONTAGEM DO CIRCUITO	7
7. COMPILAÇÃO E UPLOAD DO CÓDIGO	7
8. EXECUÇÃO E OBSERVAÇÃO DOS RESULTADOS	7
9. CONCLUSÃO.....	8
10. DIAGRAMA DE CLASSES DO CÓDIGO.....	8
11. FLUXOGRAMA DO CÓDIGO	9

1. OBJETIVOS

O objetivo deste relatório é fornecer um guia passo a passo para a implementação do script que utiliza o sensor HC-SR04 para medir distâncias e controlar um servo motor. O script faz parte do Projeto SUPER - Projeto para Educação e Pesquisa 2023, específico para a tarefa WP3-CETELI-3-IoT, sob a orientação do instrutor Andrey Bessa. O experimento envolve a medição de distâncias usando o sensor HC-SR04 e o acionamento de um servo motor com base nessa medição.

2. MATERIAIS

- Arduino MEGA (ou placa compatível)



- Sensor de distância HC-SR04



- Cabos de conexão



- Servo motor



- Botão (pushbutton)



3. CONFIGURAÇÃO DO AMBIENTE

Antes de começar, é necessário configurar o ambiente de desenvolvimento. Siga os passos abaixo:

- Faça o download e a instalação da IDE do Arduino no site oficial (<https://www.arduino.cc/en/software>).
- Conecte o Arduino ao computador utilizando um cabo USB.
- Abra a IDE do Arduino e verifique se o Arduino é reconhecido corretamente nas portas disponíveis (no menu "Ferramentas" > "Porta").
- Certifique-se de que as bibliotecas "NewPing" e "Servo" estão instaladas. Caso não estejam, vá em "Sketch" > "Incluir Biblioteca" > "Gerenciar Bibliotecas" e procure por "NewPing" e "Servo". Clique em "Instalar".

4. IMPLEMENTAÇÃO DO SCRIPT

A seguir, apresentamos o código do script que utiliza o sensor HC-SR04 e o servo motor. O script está comentado para facilitar a compreensão das diferentes partes.

```
1 // Projeto: SUPER - Projeto para Educação e Pesquisa 2023
2 // Task: WP3-CETELI-3-IoT
3 // Instrutor: Andrey Bessa
4
5 // Incluindo as bibliotecas necessárias
6 #include <NewPing.h>
7 #include <Servo.h>
8
9 // Definindo os pinos para o trigger e o echo do sensor
10 #define TRIGGER_PIN 2
11 #define ECHO_PIN 3
12
13 // Definindo o pino para o botão
14 #define BUTTON_PIN 4
15
16 // Definindo os pinos para o servo motor
17 #define SERVO_PIN 5
18
19 // Criando uma instância do objeto NewPing para o sensor HC-SR04
20 NewPing sonar(TRIGGER_PIN, ECHO_PIN);
21
22 // Criando uma instância do objeto Servo para controlar o servo motor
23 Servo servoMotor;
24
25 // Variável para controlar o estado do sistema
26 bool sistemaLigado = false;
27
28 void setup() {
29 // Inicializando a comunicação serial
30 Serial.begin(9600);
31
32 // Configurando o pino do botão como entrada
33 pinMode(BUTTON_PIN, INPUT_PULLUP);
34
35 // Anexando o servo motor ao pino correspondente
```

```
36 servoMotor.attach(SERVO_PIN);
37
38 // Movendo o servo motor para a posição inicial (desligado)
39 servoMotor.write(0);
40 }
41
42 // Variável para controlar o último estado do botão
43 int ultimoEstadoBotao = HIGH;
44
45 void loop() {
46 // Verificando o estado do botão
47 int estadoBotao = digitalRead(BUTTON_PIN);
48
49 // Verificar se o botão foi pressionado
50 if (ultimoEstadoBotao == HIGH && estadoBotao == LOW) {
51     sistemaLigado = !sistemaLigado;
52     if (sistemaLigado) {
53         Serial.println("Sistema ligado!");
54     } else {
55         servoMotor.write(0);
56         Serial.println("Sistema desligado!");
57     }
58 }
59
60 // Atualizar o último estado do botão
61 ultimoEstadoBotao = estadoBotao;
62
63 if (sistemaLigado) {
64     // Realizando a medição da distância
65     unsigned int distance = sonar.ping_cm();
66
67     // Verificando se a medição foi bem-sucedida
68     if (distance == 0) {
69         Serial.println("Erro na medição. Distância não detectada.");
70     } else {
```

```
71 // Imprimindo a distância no monitor serial
72 Serial.print("Distância: ");
73 Serial.print(distance);
74 Serial.println(" cm");
75
76 // Verificando se a distância é menor que 10 cm
77 if (distance < 10) {
78     // Ligar o servo motor
79     servoMotor.write(90); // Angulo para acionar o servo motor
80     delay(5000); // Aguarda 5 segundos
81     servoMotor.write(0); // Desliga o servo motor
82 }
83 }
84 }
85
86 // Aguardando um pequeno intervalo de tempo antes de realizar a
87 próxima medição
88 delay(1000);
89 }
```

5. EXPLICAÇÃO DO CÓDIGO

O código é dividido em várias seções. Ele inclui as bibliotecas necessárias para o uso do sensor HC-SR04 e do servo motor. Em seguida, os pinos são definidos para o sensor, o botão e o servo motor. São criadas instâncias dos objetos NewPing e Servo para utilizar o sensor e o servo motor, respectivamente.

No setup(), a comunicação serial é inicializada, o pino do botão é configurado como entrada e o servo motor é movido para a posição inicial (0 graus).

No loop(), o estado do botão é verificado. Se o botão for pressionado pela primeira vez, o sistema é ligado. Se o botão for pressionado novamente, o sistema é desligado. Se o sistema estiver ligado, o script realiza a medição da distância usando o sensor HC-SR04. Se a distância for menor que 10 cm, o servo motor é acionado para uma posição específica (90 graus) por 5 segundos antes de ser desligado novamente.

6. MONTAGEM DO CIRCUITO

Para utilizar o script, siga a seguinte montagem do circuito:

- Conecte o pino TRIGGER do sensor HC-SR04 ao pino 2 do Arduino.
- Conecte o pino ECHO do sensor HC-SR04 ao pino 3 do Arduino.
- Conecte o pino do botão ao pino 4 do Arduino.
- Conecte o pino de controle do servo motor ao pino 5 do Arduino.
- Conecte o Arduino ao computador utilizando um cabo USB.

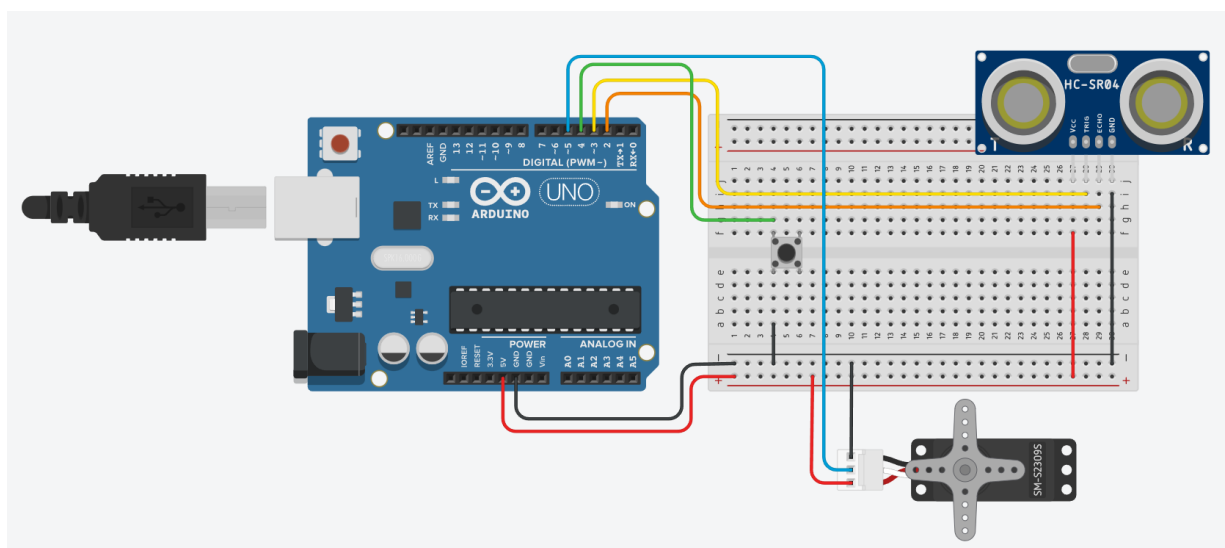


Figura 1 - Interligação dos componentes

7. COMPILAÇÃO E UPLOAD DO CÓDIGO

- Verifique se o Arduino está selecionado corretamente no menu "Ferramentas" > "Placa".
- Verifique se a porta correta do Arduino está selecionada em "Ferramentas" > "Porta".
- Clique no botão "Verificar" (ícone de marca de seleção) para compilar o código. Verifique se não há erros de compilação.
- Se a compilação for bem-sucedida, clique no botão "Enviar" (ícone de seta) para fazer o upload do código para o Arduino.

8. EXECUÇÃO E OBSERVAÇÃO DOS RESULTADOS

Após o upload do código, abra o monitor serial da IDE do Arduino ("Ferramentas" > "Monitor Serial" ou pressione Ctrl + Shift + M). Certifique-se de que a taxa de transmissão esteja definida como 9600 bps.

Agora você poderá observar as medições de distância e o acionamento do servo motor conforme a proximidade detectada pelo sensor HC-SR04. Certifique-se de testar o funcionamento pressionando o botão para ligar e desligar o sistema.

9. CONCLUSÃO

Neste relatório, foi apresentado um guia passo a passo para a implementação do script que utiliza o sensor HC-SR04 e o servo motor. O script é adequado para alunos de graduação que desejam aprender sobre o uso desses componentes e a integração com o Arduino. Espera-se que este relatório tenha fornecido informações suficientes para que os alunos possam compreender e implementar o script com sucesso.

Caso haja alguma dúvida adicional, é recomendado revisar a documentação específica dos componentes (sensor HC-SR04 e servo motor) para obter mais detalhes sobre suas características e funcionalidades.

10. DIAGRAMA DE CLASSES DO CÓDIGO

Neste diagrama, temos três classes principais: **NewPing**, **Servo** e **Sistema**.

A classe **NewPing** representa a biblioteca utilizada para controlar o sensor **HC-SR04**, que é utilizado para medir a distância. A classe **Servo** representa a biblioteca utilizada para controlar o servo motor. A classe **Sistema** representa o código principal, que gerencia o sistema, faz as medições e controla o servo motor.

As associações indicam que a classe **Sistema** possui instâncias das classes **NewPing** e **Servo** para realizar as operações de medição e controle do servo motor. Além disso, a classe **Sistema** também possui atributos privados para os pinos utilizados no sistema: **TRIGGER_PIN**, **ECHO_PIN**, **BUTTON_PIN** e **SERVO_PIN**. Esses atributos são utilizados internamente pelo código na classe **Sistema**.

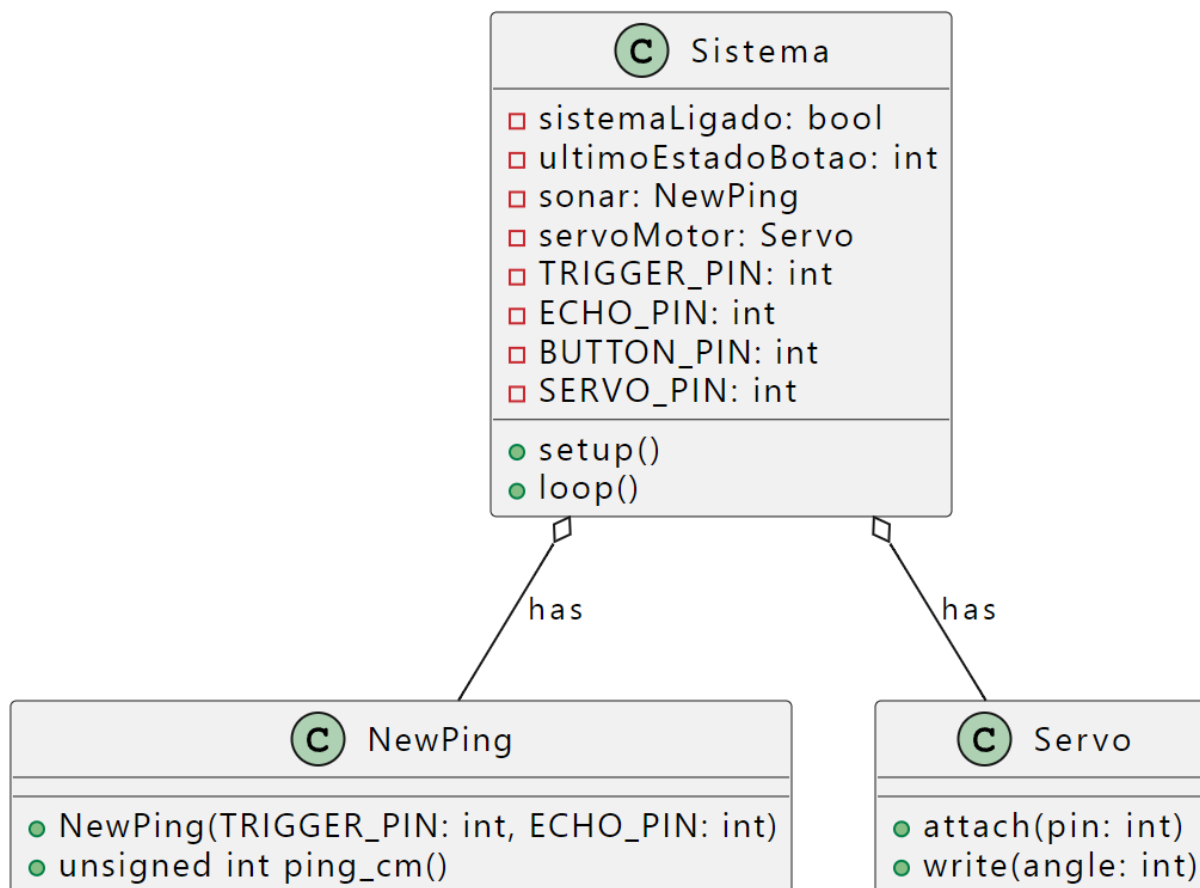


Figura 2 - Diagrama de classes do Código

11. FLUXOGRAMA DO CÓDIGO

Neste fluxograma, iniciamos com o bloco start e, em seguida, executamos o método Setup(). Em seguida, entramos em um loop infinito repeat while (true) que representa a função loop() do código.

Dentro do loop, verificamos se o botão foi pressionado (Ler Botão) e, se for o caso, verificamos se o sistema está ligado (Sistema Ligado?). Se o sistema estiver ligado, medimos e verificamos a distância. Se a distância for menor que 10 cm, ligamos o servo motor, aguardamos 5 segundos e desligamos o servo motor. Em caso negativo, ou seja, se o sistema não estiver ligado ou o botão não foi pressionado, o sistema permanece desligado.

O fluxograma continua em um loop, esperando 1 segundo antes de executar o próximo ciclo do loop repeatend.

Finalmente, o fluxograma é encerrado com o bloco stop.

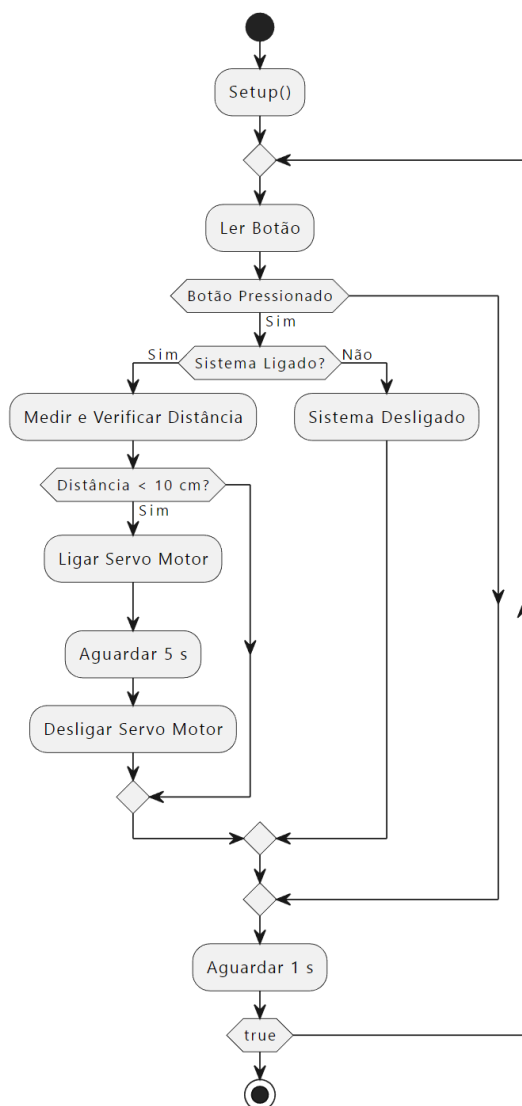


Figura 3 - Fluxograma do Código