AI & SUSTAINABILITY IN VET EDUCATION

Nome do(s) autor(es): Sucessos Criativos, Lda

Título do WP: WP3 – IA-SAVES@ProjectHub

Referência de Atividade: 3.2 – Modelo de projeto para IA-SAVES@ProjectHub

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Índice

1. Título do Projeto: 4
2. Categoria do Projeto: 4
3. Descrição do Projeto: 4
4. Palavras-chave: 4
5. Público-alvo: 4
6. Objetivos do Projeto: 4
7. Cronologia do Projeto: 5
8. Recursos Fornecidos: 5
9. Ferramentas e Tecnologias Utilizadas: 6
10. Plano de Implementação Passo a Passo: 7
Passo 1: Prepare o Raspberry Pi para Interação de Hardware 7
Passo 2: Ligue o Sensor de LDR 7
Passo 3: Escreve o programa Python para Detetar Luz 8
11. Critérios de Avaliação: 12
12. Línguas Disponíveis: 12
13. Colaboradores: 12
17. Feedback e Melhoria Contínua: 13

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

1. Título do Projeto:

Primeiros Passos com o Kit AI-VET: Medição de Distância com um Sensor Ultrassónico

---

2. Categoria do Projeto:

Fundamentos da Microeletrónica

---

3. Descrição do Projeto:

Este projeto introduz os formandos à medição de distância utilizando o sensor ultrassónico HC-SR04. Os participantes irão ligar o sensor a um Raspberry Pi 5 e desenvolver um script em Python para medir e mostrar valores de distância em tempo real. Esta atividade prática serve de base para aplicações de automação e IoT, como contentores inteligentes, robótica ou sistemas energeticamente eficientes.

---

4. Palavras-chave:

Raspberry Pi, Python, HC-SR04, GPIO, Sensor de Distância, Automação, Deteção Ambiental

---

5. Público-alvo:

Educadores, aprendizes e criadores, com ou sem experiência prévia, que estão a iniciar o seu percurso na microeletrónica e computação física.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

6. Objetivos do Projeto:

- Ligue e configure o sensor ultrassónico HC-SR04 com um Raspberry Pi 5
- Meça distâncias do mundo real usando processamento digital de sinal em Python
- Reforçar competências em acesso GPIO, codificação e entrada/saída baseada em sensores

---

7. Cronologia do Projeto:

A duração estimada é de aproximadamente **2,5 a 3 horas**, incluindo a montagem do ambiente e a montagem física.

---

8. Recursos Fornecidos:

- Kit AI-VET (Raspberry Pi 5, LED, Breadboard, Jumpers, Sensor Ultrassónico HC-SR04)

MATERIAL	
Raspberry Pi 5	
Fonte de alimentação USB Tipo-C de 27W para Raspberry Pi 5	

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Cabos Flat Jumper F/F	
Breadboard	
FIO EM FILA ÚNICA PARA PLACAS DE ENSAIO	
HC-SR04P - SENSOR ULTRASSÓNICO 3V~5V	

---

9. Ferramentas e Tecnologias Utilizadas:

- Raspberry Pi OS
- Python 3 (em ambiente virtual)
- `Biblioteca LGPIO`
- Geany IDE
- Terminal (bash)

---

10. Plano de Implementação Passo a Passo:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Passo 1: Prepare o Raspberry Pi para Interação de Hardware

Este projeto baseia-se diretamente nas tarefas concluídas em "Começar com o Raspberry Pi 5: Configuração, Configuração e Hello World em Python." e "Piscar um LED". Se está a começar por aqui, ou precisa de uma atualização, deve primeiro passar pelos passos essenciais de configuração para preparar o seu Raspberry Pi para controlar componentes físicos.

Neste passo, irá:

- Instale o Raspberry Pi OS usando a ferramenta Raspberry Pi Imager.
- Configure definições básicas do sistema, como Wi-Fi, layout do teclado e fuso horário.
- Ative interfaces chave incluindo SPI, I2C, 1-Wire e VNC usando o menu de configuração do Raspberry Pi.
- Identifique o endereço IP do Raspberry Pi usando o ifconfig e ligue-se remotamente via TigerVNC Viewer.
- Realizar uma atualização completa do sistema
- Instalar a biblioteca lgpio

Passo 2: Ligue o Sensor Ultrassónico

Neste passo, vais ligar o sensor ultrassónico HC-SR04 ao Raspberry Pi. O sensor funciona enviando um pulso ultrassónico a partir do pino TRIG (gatilho) e esperando que este volte para o **pino ECHO** (eco). O tempo que o eco demora a regressar é usado para calcular a distância a um objeto.

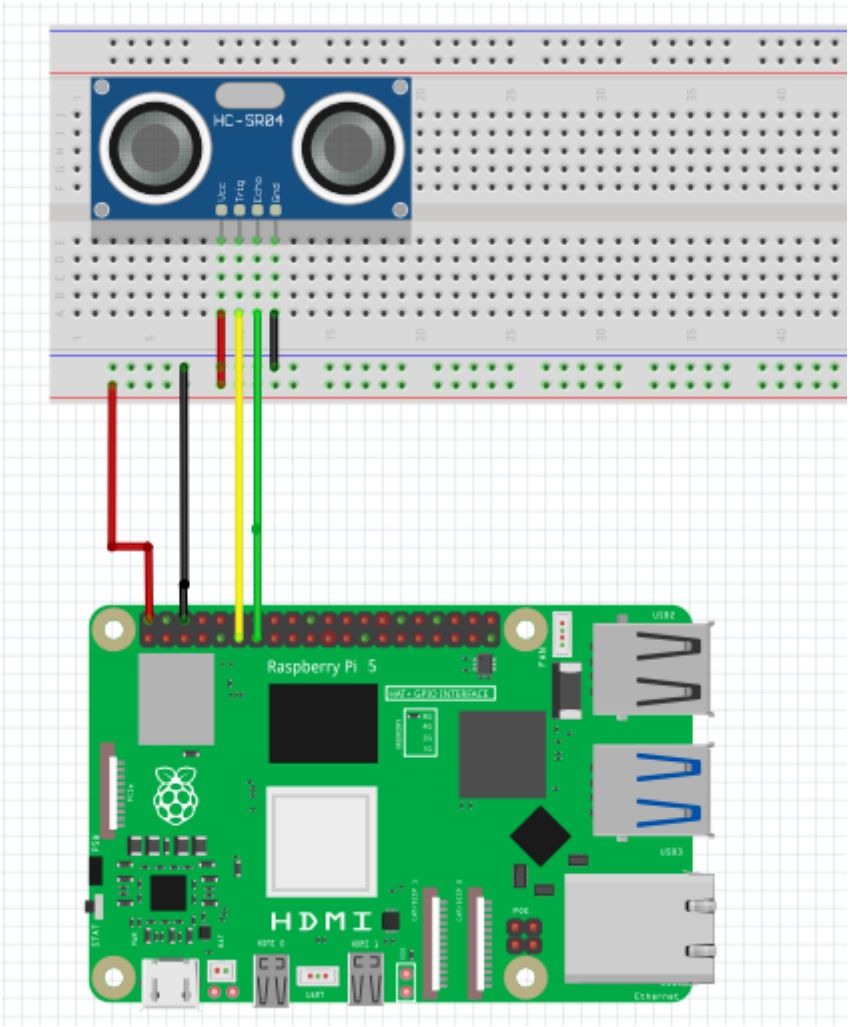
- **TRIG**: Envia um pulso sonoro de alta frequência (sinal de disparo) quando ativado.
- **ECHO**: Recebe o pulso refletido; o intervalo de tempo entre o TRIG e o ECHO ajuda a determinar a distância.

Ligações:

- **VCC** → Raspberry Pi 5V (pino 2)
- **GND** → Raspberry Pi GND (pino 6)
- **TRIG** → GPIO 17 (pino 11)
- **ECHO** → GPIO 27 (pino 13)

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.



Passo 3: Escreva o programa Python para medir distâncias

Com o sensor ligado, vais agora escrever um programa em Python que verifica a distância do objeto e imprime uma mensagem no terminal com base no resultado.

Abra o Geany e introduza o seguinte código:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
Importação da LGPIO
Tempo de importação

# Defina pinos GPIO para o sensor ultrassónico
TRIGGER_PIN = pino GPIO 17  # ligado ao pino de disparo do sensor ultrassónico
ECHO_PIN = pino GPIO 27     # ligado ao pino Echo do sensor ultrassónico

# Inicializar o chip GPIO (o chip GPIO padrão é "/dev/gpiochip0")
h = lgpio.gpiochip_open(0)

# Defina o pino de gatilho como saída e o pino de eco como entrada
lgpio.gpio_claim_output(h, TRIGGER_PIN)
lgpio.gpio_claim_input(h, ECHO_PIN)

# Função para medir distâncias usando o sensor ultrassónico
def measure_distance():
    """
    Envia um pulso para o sensor ultrassónico e calcula a distância
    com base na duração do eco retornado.
    """
    # Enviar um pulso de 10 microssegundos para ativar o sensor
    lgpio.gpio_write(h, TRIGGER_PIN, 1)
    time.sleep(0.00001)  # Espera 10 microssegundos
    lgpio.gpio_write(h, TRIGGER_PIN, 0)

    # Espera que o eco suba (início do pulso do eco)
    start_time = tempo.tempo()
    enquanto lgpio.gpio_read(h, ECHO_PIN) == 0:
        start_time = tempo.tempo()  # Capturar a hora de início quando o pulso
é detetado

    # Espera que o eco baixe (fim do pulso do eco)
    end_time = tempo.tempo()
    enquanto lgpio.gpio_read(h, ECHO_PIN) == 1:
        end_time = tempo.tempo()  # Capturar o fim do tempo quando o pulso
termina

    # Calcular a duração do pulso
    pulse_duration = end_time - start_time

    # Calcular a distância em centímetros (velocidade do som é 34300 cm/s)
    # Dividimos por 2 porque o som viaja até ao objeto e volta
    distância = (pulse_duration * 34300) / 2

     Distância de retorno

Experimente:
    embora Verdadeiro:
        # Meça a distância usando o sensor ultrassónico
        distance_cm = measure_distance()
        print(f'Distância: {distance_cm:.2f} cm')  # Imprimir a distância em
centímetros

        # Espera 0,3 segundos antes da próxima medição
        tempo.sono(0.3)

exceto o KeyboardInterrupt:
    # Lidar com a interrupção do utilizador (Ctrl + C) para parar o programa
    # Limpar fechando o chip GPIO
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
    lgpio.gpiochip_close(h)
    print ("GPIO limpo!")
```

Guarda o ficheiro como **UltraSound.py.**

Decomposição do Código:

Configuração e Importações: O script começa por importar o lgpio para gerir os pinos GPIO e o tempo para controlar o timing e os atrasos entre leituras do sensor.

Definição e Inicialização dos Pinos: Estão definidos dois pinos GPIO: GPIO 17 para o pino TRIG e GPIO 27 para o pino ECHO. Estes são então configurados usando lgpio.gpio_claim_output() e lgpio.gpio_claim_input(), respetivamente, após a abertura do chip GPIO.

Função de Medição: A função measure_distance() ativa o sensor ultrassónico ao enviar um pulso de 10 microssegundos. Depois espera para receber o eco de regresso. O tempo entre o envio e a receção do sinal é usado para calcular a distância.

Se o eco for recebido rapidamente → o objeto está próximo

Se demorar mais → o objeto está mais longe

A fórmula utilizada é:

distância = (duração × velocidade do som) / 2

Onde a velocidade do som é de 34300 cm/s.

Loop Principal: O programa mede e imprime continuamente a distância a cada 0,3 segundos. Isto torna-o responsivo a alterações em tempo real na proximidade dos objetos.

Gestão de Interrupções: Quando o utilizador pressiona Ctrl + C, o loop para de forma elegante e o chip GPIO é fechado em segurança usando lgpio.gpiochip_close().

Passo 4: Execute o Programa e Observe a Saída

Com o código escrito e guardado, é altura de testar a configuração e observar como responde a objetos próximos.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Para executar o script:

Guarde o ficheiro como UltraSound.py

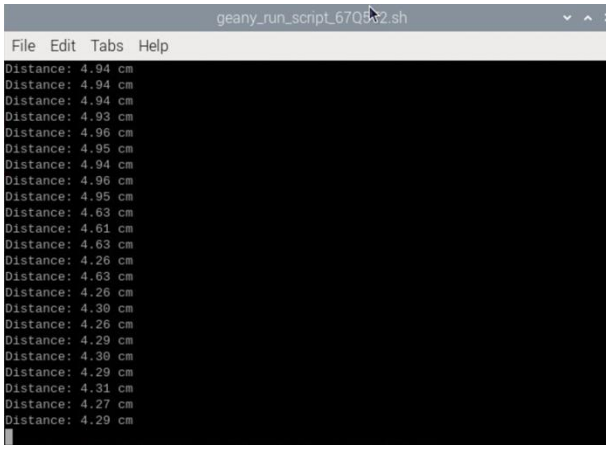
Pressione F5 em Geany para **executar** o código, ou execute-o pelo **terminal escrevendo**:

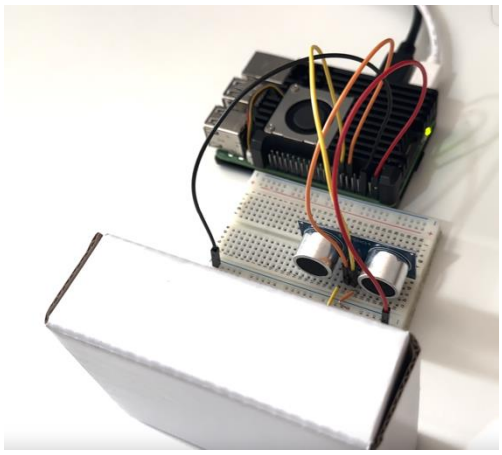
Python3 UltraSound.py

À medida que o script é executado, ele mede e imprime a distância entre o sensor e qualquer objeto colocado à sua frente, atualizando-se a cada 0,3 segundos.

Objeto à frente do sensor → impressão a uma distância menor

Objeto distante → imprime valor maior





---

11. Critérios de Avaliação:

-
- **Ligação ao Sensor**: O sensor ultrassónico está devidamente ligado e funcional.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- **Lógica Digital**: O script deteta com sucesso a proximidade de objetos com base no tempo do eco.
- **Funcionalidade do Código**: Sem erros; o script corre suavemente e sai limpo.
- **Testes Práticos**: O utilizador testou o sensor com objetos a diferentes distâncias e observou os resultados esperados.

---

12. Línguas Disponíveis:

Inglês, **Português** (traduções em progresso)

---

13. Colaboradores:

Fátima Pais, Luís Soares – Designers e Facilitadores de Projetos

---

17. Feedback e Melhoria Contínua:

- **Inquéritos**: No final deste marco, alunos e educadores irão preencher inquéritos de feedback para avaliar a eficácia do projeto.
- **Revisões de Projetos**: Após a conclusão do projeto, será realizada uma reunião final de revisão para discutir potenciais melhorias para futuros projetos baseados em IA.
- **Análise de Dados**: Os dados de feedback serão analisados para identificar pontos fortes e áreas a melhorar, informando projetos futuros.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.