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: Automação de Janelas Controlada por Luz para Eficiência Energética

---

2. Categoria do Projeto:

Fundamentos da Microeletrónica

---

3. Descrição do Projeto:

Este projeto combina a deteção de luz ambiente com a atuação automática para abrir ou fechar uma janela com base na intensidade da luz em tempo real. Utilizando um sensor LDR (Light Dependent Resistor) e um servomotor ligado a um Raspberry Pi 5, os aprendizes irão construir um sistema responsivo que apoia a eficiência energética. Quando há luz, o servo abre a janela, promovendo ventilação e iluminação natural. Quando falta luz, a janela fecha-se, ajudando a conservar o calor ou a reduzir o fluxo de ar frio. Esta atividade combina objetivos de sustentabilidade com princípios fundamentais da IoT.

---

4. Palavras-chave:

Raspberry Pi, Python, LDR, Servomotor, GPIO, Casa Inteligente, Eficiência Energética

---

5. Público-alvo:

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.

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

---

6. Objetivos do Projeto:

- Interface um sensor LDR e um servomotor com um Raspberry Pi 5
- Construir um sistema de controlo servo ativado por luz em tempo real
- Compreenda os princípios da PWM (Modulação de Largura de Pulso)
- Promover a automação sustentável usando luz natural e ar

---

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 LDR, Módulo Microservo Motor)

MATERIAL	
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.

Fonte de alimentação USB Tipo-C de 27W para Raspberry Pi 5	
Cabos Flat Jumper F/F	
Breadboard	
FIO EM FILA ÚNICA PARA PLACAS DE ENSAIO	
Módulo de Sensor LDR com potenciómetro	
SG90 1,6Kg 180º Microservo (IM4)	

---

9. Ferramentas e Tecnologias Utilizadas:

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

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.

10. Plano de Implementação Passo a Passo:

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 Olá Mundo 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.
- Faça uma atualização completa do sistema.
- Instalar a `biblioteca` lgpio

Para melhores resultados, conclua previamente os projetos fundamentais relevantes:

- "Primeiros Passos com o Kit AI-VET: Deteção de Níveis de Luz com um Sensor LDR e Raspberry Pi 5"
- "Primeiros Passos com o Kit AI-VET: Controlar um Servomotor com Raspberry Pi 5"

Estes guias fornecem instruções passo a passo para ligar e testar cada componente individualmente antes de os combinar neste projeto.

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 2: Construir o Circuito

Neste passo, vais ligar os componentes ao Raspberry Pi. O sistema utiliza um sensor de Resistência Dependente da Luz (LDR) para detetar luz ambiente e um servomotor para abrir ou fechar uma janela com base no nível de luz.

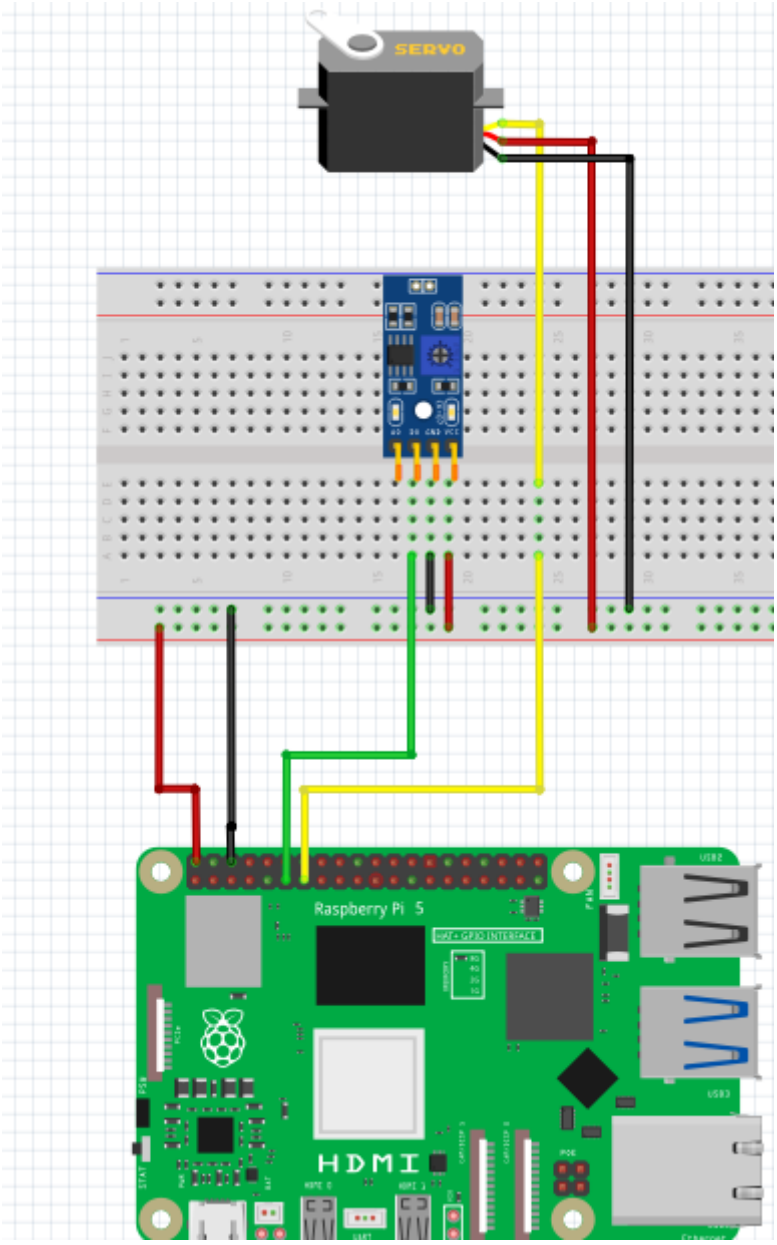
- **Sensor LDR (Deteção de Luz):** Envia um sinal digital indicando se a luz é detetada ou não.
- **Servomotor (Controlo de Movimento):** Move-se para um ângulo específico dependendo das condições de luz (0° para fechado, 180° para aberto).

Ligações:

- **Sensor de LDR:**
  - VCC → Raspberry Pi 5V (pino 2)
  - GND → Raspberry Pi GND (pino 6)
  - SAÍDA → GPIO 17 (pino 11)
- **Servomotor (SG90):**
  - Vermelho (VCC) → 5V (pino 4)
  - Castanho (GND) → GND (pino 9)
  - Laranja (Sinal) → 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 controlar a janela

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.

Com os sensores ligados, vais agora escrever um programa em Python que combina estas leituras de sensores num pequeno sistema inteligente que imprime uma mensagem no terminal com base no resultado.

Abra o Geany e introduza o seguinte código:

```
Importação da LGPIO
Tempo de importação
# Pinos GPIO
SERVO_PIN = pino 27  # para o servomotor
LDR_PIN = pino # 17    para o sensor LDR
# Inicializar o chip GPIO (o chip GPIO padrão é normalmente "/dev/gpiochip0")
h = lgpio.gpiochip_open(0)
# Reclamar os pinos GPIO
lgpio.gpio_claim_output(h, SERVO_PIN)  # Servo como saída
lgpio.gpio_claim_input(h, LDR_PIN)     # LDR como entrada
# Função para gerar PWM para servomotor
Def set_servo_pulsewidth(h, pin, pulse_width):
    """
    Envia um sinal alto para o pino servo para a largura do pulso (em
milissegundos).
    Isto cria um sinal PWM mantendo o pino ALTO durante um determinado período.
    """
 lgpio.gpio_write(h, pino 1) # Definir pino ALTO (início do pulso)
    time.sleep(pulse_width / 1000)            # Esperar pela duração da largura
do pulso
    lgpio.gpio_write(h, pino, 0)                  # Definir pino BAIXO (fim do
pulso)
# Função para mover o servo para um ângulo especificado
def move_servo(ângulo):
    """
    Converter o ângulo dado (entre 0 e 180 graus) numa largura de pulso
correspondente
    e move o servo para essa posição.
    """
    pulse_width = 1 + (ângulo / 180,0)          # Calcular a largura do pulso
(1ms a 2ms)
    set_servo_pulsewidth(h, SERVO_PIN, pulse_width)  # Define a largura do pulso
para mover o servo
# Loop principal para controlar o servo com base na entrada do sensor LDR
Experimente:
    embora Verdadeiro:
        # Leia o estado atual do sensor de LDR
        light_state = lgpio.gpio_read(h, LDR_PIN)
        # Se for detetada luz (o sensor LDR devolve 0), mova o servo para 180
graus
        se light_state == 0:
```

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.

```
            print ("Luz detetada! Movendo o servo para 180 graus.")
            move_servo(180)
        # Se não for detetada luz (o sensor LDR retorna 1), mova o servo para 0
graus
        Caso mais:
            imprimir("Nenhuma luz detetada! Movendo servo para 0 graus.")
            move_servo(0)
        # Adicionar um pequeno atraso para evitar leituras contínuas e permitir
tempo para alterações
        tempo.sono(1)
exceto o KeyboardInterrupt:
    # Lidar com a interrupção do utilizador (Ctrl + C) para parar o programa
    print("\nParar o controlo servo e sair do programa.")
    # Limpar e fechar os recursos do GPIO antes de sair do programa
    lgpio.gpio_write(h, SERVO_PIN, 0)      # Desliga o SERVO_PIN antes de sair
    lgpio.gpiochip_close(h)

    print("GPIO limpo!")
```

Guarda o ficheiro como `light_controlled_window.py`

Decomposição do Código:

Configuração e Importações:
Importa `o lgpio` para interação com GPIO e `tempo` para atrasos.

Definição e Inicialização dos Pinos:
Define GPIO 17 para o sensor LDR e GPIO 27 para o servo. Reclama estes pinos para entrada/saída.

Controlo PWM para
função personalizada `Servo:A set_servo_pulsewidth` imita PWM mantendo o GPIO alto durante um período calculado.

Lógica de Controlo de Janelas:

- Se **for detetada luz**, o servo roda para 180° (abre a janela)
- Se **não houver luz**, o servo volta a 0° (fecha a janela)

Loop e Interrupção:
O script monitoriza continuamente o nível de luz e ajusta o servo a cada segundo. Se Ctrl + C for pressionado, o script sai e limpa o 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.

Passo 4: Execute o Programa e Observe a Saída

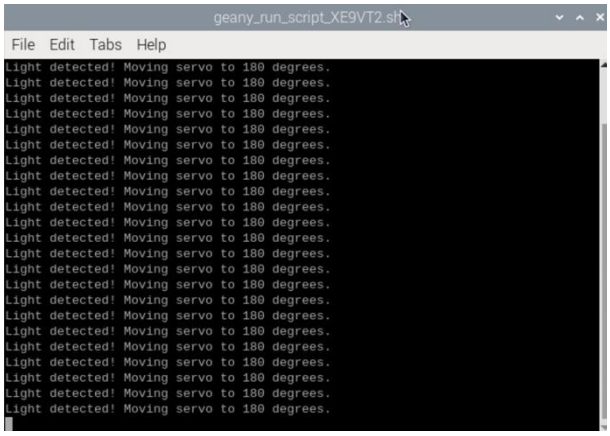
Com o código escrito e guardado, é altura de testar a configuração e observar como responde a objetos próximos.

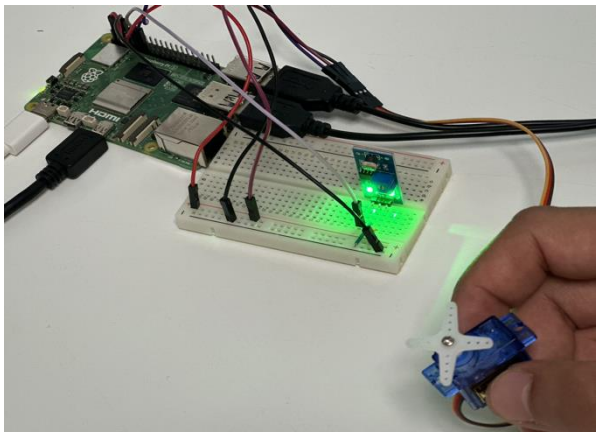
Para executar o script:

Guarda o guião como **light_controlled_window.py**

Pressione F5 em **Geany** ou **abra** o **terminal** e **execute**:

Python3 light_controlled_window.py





---

11. Critérios de Avaliação:

- **Integração do Sensor LDR**: O sensor de luz está corretamente ligado, configurado e capaz de distinguir entre condições de luz e de escuridão.
- **Funcionalidade do Servomotor**: O servo responde com precisão a alterações de luz, rodando entre 0° e 180° com base na entrada do sensor.

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.

- **Desempenho do Código**: O script corre sem erros de sintaxe ou de execução, executa a lógica correta e sai de forma elegante quando interrompido.
- **Relevância da Eficiência Energética**: O aprendiz compreende como a automação contribui para a sustentabilidade ao reduzir a dependência da iluminação artificial ou do controlo climático.
- **Testes do Sistema:** O sistema foi testado em condições reais com entradas de luz intensa e baixa, demonstrando o comportamento correto.

---

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.

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.