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: Trabalhar com o Ecrã OLED

---

2. Categoria do Projeto:

Fundamentos da Microeletrónica

---

3. Descrição do Projeto:

Neste projeto, os alunos irão explorar como apresentar informação usando um ecrã OLED de 0,95 polegadas ligado a um Raspberry Pi 5. Usando comunicação I^2C, controlam o ecrã via Python e visualizam texto ou imagens, como leituras de sensores ou logótipos. Este projeto desenvolve competências em visualização de dados, inicialização de ecrãs e protocolos de comunicação por hardware — aspetos vitais da computação física em dispositivos inteligentes.

---

4. Palavras-chave:

OLED, SSD1306, Raspberry Pi, Python, I2C, Ecrã, Renderização de Imagem

---

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 um ecrã OLED I²C com um Raspberry Pi.
- Mostrar e atualizar texto em tempo real usando Python.
- Carregar e renderizar imagens usando PIL no SSD1306 OLED.
- Compreender a utilização do protocolo I²C na comunicação sensor/ecrã.

---

7. Cronologia do Projeto:

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

---

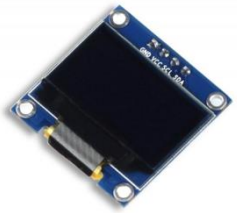
8. Recursos Fornecidos:

- Kit AI-VET (Raspberry Pi 5, LED, Breadboard, Jumpers, Ecrã OLED)

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	
Ecrã OLED i2c de 0,95 polegadas Azul/Amarelo SSD1306	

---

9. Ferramentas e Tecnologias Utilizadas:

- Raspberry Pi OS
- Python 3 (ambiente virtual)
- `Biblioteca adafruit-circuitpython-ssd1306`
- PIL (Biblioteca de Imagem em Python)
- 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 Raspberry Pi 5: Configuração, Configuração e Hello World em Python."** Se está a começar por aqui, ou precisa de uma atualização, começará por configurar o seu Raspberry Pi para comunicação I²C e desenvolvimento em Python.

Neste passo, irá:

- Instale o Raspberry Pi OS usando a **ferramenta Raspberry Pi Imager**.
- Configure definições do sistema como **Wi-Fi**, **disposição do teclado** e **fuso horário**.
- Ative as interfaces principais — especialmente **I²C**, mas também **SPI**, **1-Wire** e **VNC** — através do menu **de Configuração do Raspberry Pi**.
- Use o comando terminal `ifconfig` para identificar o endereço IP do Raspberry Pi e ligue-se remotamente via **TigerVNC Viewer**.
- Instalar os pacotes Python necessários, especialmente o driver de SSD1306 Adafruit, usando. Esta biblioteca fornece especificamente funções e ferramentas para interagir com SSD1306 ecrãs OLED através de comunicação I2C ou SPI. Para instalar, execute o seguinte comando::

```
pip3 install adafruit-circuitpython-ssd1306
```

Embora este projeto não exija controlo GPIO de baixo nível, apresenta-lhe comunicação **I²C**, renderização de ecrã e uso de bibliotecas externas em Python para interfaces embutidas.

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 o **módulo de ecrã OLED** ao Raspberry Pi usando o protocolo de comunicação I²C. I²C (Circuito Inter-Integrado) é um protocolo comum usado para ligar sensores e ecrãs com apenas duas linhas de dados.

Ecrã OLED (Saída Visual):

O ecrã OLED recebe texto ou imagens do Raspberry Pi e apresenta-as visualmente. Utiliza a **interface I²C**, que simplifica a cablagem ao partilhar duas linhas de dados entre vários dispositivos.

Ligações:

- **VCC → Raspberry Pi 5V (pino 2)**
  Fornece energia ao ecrã OLED.

- **GNDRaspberry Pi GND (pino 6) →**
  Liga a terra do ecrã à terra da Raspberry Pi.

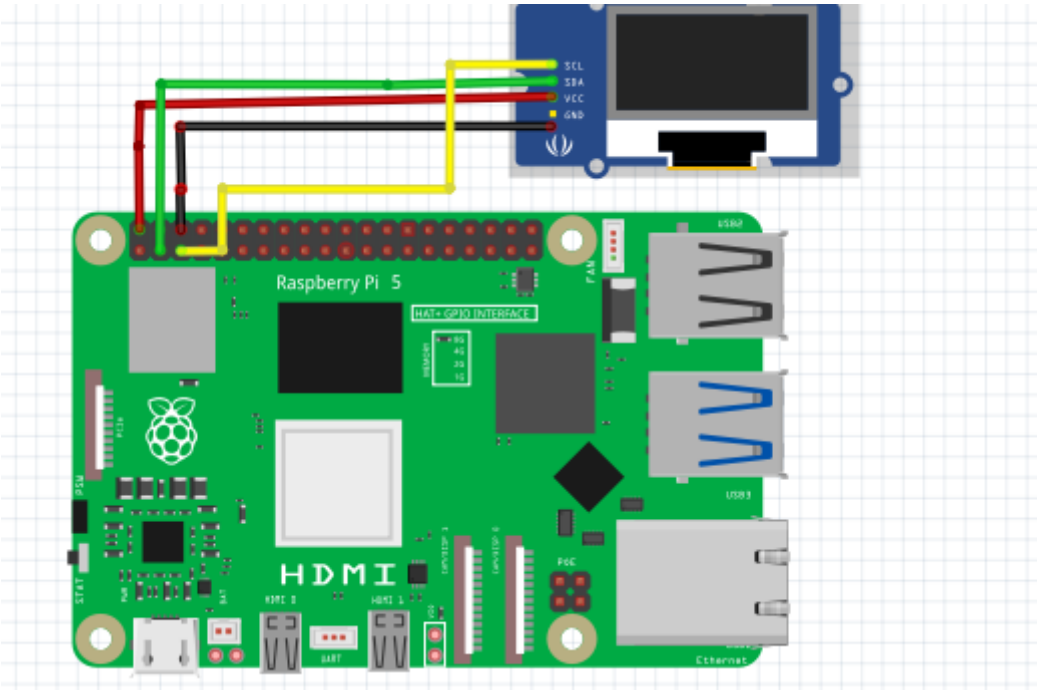
- **SDA (Linha de Dados Serial) GPIO 2 (pino 3) →**
  Transmite dados do Raspberry Pi para o ecrã OLED.

- **SCL (Linha de Relógio Serial) GPIO 3 (pino 5) →**
  Envia sinais de temporização usados para sincronizar a comunicação.

Estas quatro ligações são tudo o que precisas para comunicar com o ecrã OLED. Uma vez ligado, o Raspberry Pi envia dados de píxeis através do barramento I²C para controlar o que é exibido.

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

Com o ecrã OLED ligado, agora vais escrever um programa Python que inicializa o ecrã e mostra texto centrado. Este programa ajuda-te a perceber como enviar a saída visual através do I²C usando uma biblioteca gráfica simples.

Abra o Geany e introduza o seguinte código:

```
Tabuleiro de importação
Import Digitalio
a partir de PIL importar Image, ImageDraw, ImageFont
```

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.

```
montante adafruit_ssd1306
# Inicializar dimensões do ecrã OLED (128x64 pixels)
LARGURA = 128
ALTURA = 64
# Configurar comunicação I2C com o ecrã OLED
# Usa os pinos I2C padrão na placa (SCL e SDA)
i2c = tabuleiro. I2C() # Inicializa comunicação I2C
# Criar uma instância de ecrã OLED SSD1306 com a largura, altura e interface
I2C especificadas
OLED = adafruit_ssd1306. SSD1306_I2C(LARGURA, ALTURA, i2c, addr=0x3C)
# Limpar o ecrã OLED (preenche o ecrã de preto)
oled.fill(0)
oled.show()
# Criar um novo objeto de imagem com cor "1-bit" (preto e branco) para
desenhar no ecrã
imagem = Image.new("1", (oled.width, oled.height))
# Obtenha um objeto de desenho a partir da imagem para desenhar formas e texto
draw = ImageDraw.Draw(image)
# Carregar a fonte padrão para mostrar texto no OLED
fonte = ImageFont.load_default()
# Defina uma função auxiliar para calcular o tamanho do pixel do texto
(largura e altura)
Definitivamente getfontsize(fonte, texto):
    # Isto calcula a caixa delimitadora do texto, devolvendo a sua largura e
altura
    esquerda, cima, direita, baixo = font.getbbox(texto)
    voltar direita - esquerda, baixo - cima
# Defina o texto a mostrar no ecrã OLED
text="AI-VET"
# Obtenha a largura e altura do texto em pixels para ajudar a centrá-lo no
ecrã
(font_width, font_height) = getfontsize(fonte, texto)
# Desenhe apenas o texto na imagem, centrado horizontal e verticalmente no
ecrã OLED
draw.text(
    (oled.width // 2 - font_width // 2, oled.height // 2 - font_height // 2),
# Centrar o texto
    texto, # Texto para mostrar
    font=fonte, # Usar a fonte padrão
    preenchimento=255 # Cor branca para o texto
)
# Enviar a imagem (que agora contém apenas o texto) para o ecrã OLED
oled.imagem(imagem)
# Mostrar a imagem no ecrã OLED
oled.show()
```

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.

Guardar o ficheiro como **`OLED.py`**

Decomposição do Código:

Configuração e Importações:

O script começa por importar:

- placa e `digitalio` para gerir o mapeamento de pinos de hardware.
- `Módulos PIL` (Python Imaging Library) para renderização de imagens e texto.
- `adafruit_ssd1306` para interagir com o ecrã OLED.

Inicialização do Ecrã:

- A largura e altura do ecrã são definidas como 128x64.
- Os pinos I^2C padrão do Raspberry Pi são usados para configurar a comunicação.
- Um objeto OLED é criado, limpo e preparado para desenho.

Desenhar o Texto:

- Uma imagem de 1 bit (preto e branco) é criada na memória.
- A cadeia `"AI-VET"` é preparada para exibição usando a fonte predefinida.
- Uma função auxiliar calcula o tamanho do texto para que possa ser centrado.
- O texto é desenhado diretamente para a tela da imagem.

Renderização do Resultado:

- A imagem é transferida para o ecrã OLED.
- A função `oled.show()` torna o conteúdo visível no ecrã.

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

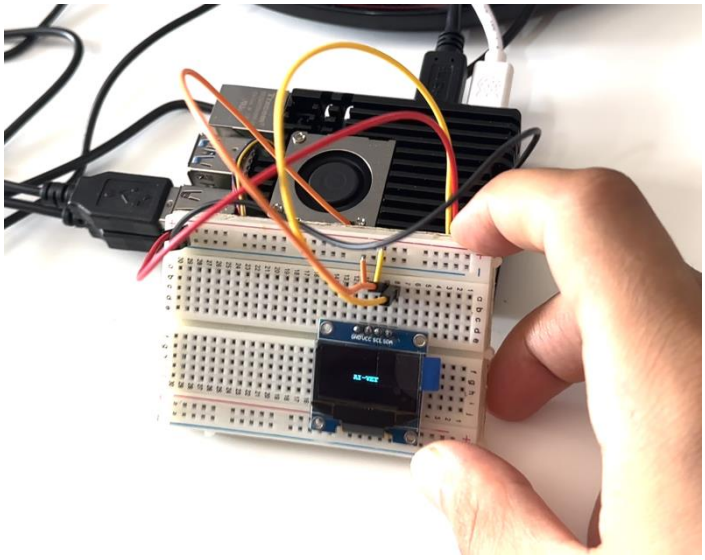
Com o código escrito e guardado, é altura de testar a configuração e observar como responde a objetos próximos.

Para executar o script:

Guarda o guião como `OLED.py`

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

Python3 oled.py



Passo 5: Estenda o programa para alternar entre texto e imagem

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.

Agora que mostraste texto estático, vamos mais longe. Neste passo, irá criar um programa dinâmico que alterna entre dois ecrãs no OLED: um com o texto `"AI-VET"` e outro com o logótipo da AI-VET.

Este exercício apresenta:

- Funções reutilizáveis para saída OLED.
- Gestão de imagem com PIL.
- Comportamento de looping para ecrãs dinâmicos.

Antes de começar:

- **Descarregue** o logótipo da AI-VET de ai-vet.eu
- **Clique com o botão direito** no logótipo e escolha "Guardar Imagem Como"
- **Guarda** como `logo.png` na mesma pasta do teu script Python

Abra o Geany e introduza o seguinte código:

```
Tempo de importação
Tabuleiro de importação
Import Digitalio
a partir de PIL importar Image, ImageDraw, ImageFont
montante adafruit_ssd1306
# Inicializar dimensões do ecrã OLED (128x64 pixels)
LARGURA = 128
ALTURA = 64
# Configurar comunicação I2C com o ecrã OLED
i2c = tabuleiro. I2C() # Inicializa comunicação I2C
# Criar uma instância de ecrã OLED SSD1306 com a largura, altura e interface
I2C especificadas
OLED = adafruit_ssd1306. SSD1306_I2C(LARGURA, ALTURA, i2c, addr=0x3C)
# Função auxiliar para mostrar texto
Def display_text(texto):
    # Criar um novo objeto de imagem com cor "1-bit" (preto e branco) para
desenhar no ecrã
 imagem = Image.new("1", (oled.width, oled.height))
    draw = ImageDraw.Draw(image)
    # Carregar a fonte padrão para mostrar texto no OLED
    fonte = ImageFont.load_default()
    # Calcular o tamanho do pixel do texto (largura e altura)
    font_width, font_height = font.getsize(texto)
```

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.

```
    # Desenhe o texto na imagem, centrado horizontal e verticalmente no ecrã
OLED
    draw.text(
        (oled.width // 2 - font_width // 2, oled.height // 2 - font_height //
2),  # Centrar o texto
        texto, # Texto para mostrar
        font=fonte, # Usar a fonte padrão
        preenchimento=255 # Cor branca para o texto
    )
    # Enviar a imagem (que agora contém apenas o texto) para o ecrã OLED
    oled.imagem(imagem)
    oled.show()
# Função auxiliar para mostrar uma imagem
Definitivamente display_image(image_path):
    # Carregar uma imagem a partir do ficheiro (substituir pelo caminho
correto para a sua imagem)
    image = Image.open(image_path).convert("1")  # Converter para cor de 1 bit
    # Obtém o tamanho original da imagem
    original_width, original_height = image.size
    # Calcular o fator de escala para se ajustar à imagem dentro do ecrã
(128x64) sem distorção
    scaling_factor = min(LARGURA / original_width, ALTURA / original_height)
    # Calcular o novo tamanho, mantendo a proporção de aspeto
    new_width = int(original_width * scaling_factor)
    new_height = int(original_height * scaling_factor)
    # Redimensiona a imagem com a nova largura e altura
    imagem = imagem.redimensionamento((new_width, new_height))
    # Crie uma imagem em branco do tamanho do ecrã OLED (128x64 píxeis)
    final_image = Image.new("1", (oled.width, oled.height))
    # Calcular a posição para centralizar a imagem no ecrã
    x_position = (oled.width - new_width) // 2
    y_position = (altura oleada - new_height) // 2
    # Cola a imagem redimensionada na tela em branco, centrada
    final_image.paste(image, (x_position, y_position))
    # Enviar a imagem final para o ecrã OLED
    oled.imagem(final_image)
    oled.show()
# Loop principal para alternar entre texto e imagem
Experimente:
    embora Verdadeiro:
        # Mostrar o texto durante 2 segundos
        display_text("AI-VET")
        tempo.sono(2)
        # Mostrar a imagem durante 2 segundos
 display_image("logo.png"") # Substituir pelo caminho correto do ficheiro de
imagem
        tempo.sono(2)
exceto o KeyboardInterrupt:
    # Limpar o ecrã antes de sair
    oled.fill(0)
    oled.show()
```

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("Programa encerrado.")
```

Passo 6: Execute o programa e observe o ecrã alternado

Com tudo pronto, está na hora de testar o seu script melhorado.

Para executar o programa:

1. **Guarda** o teu ficheiro como `oled_alternate.py`
2. Certifica-te de **que o logo.png está na mesma pasta.**
3. **Prima F5** em **Geany** ou executa o script via terminal:

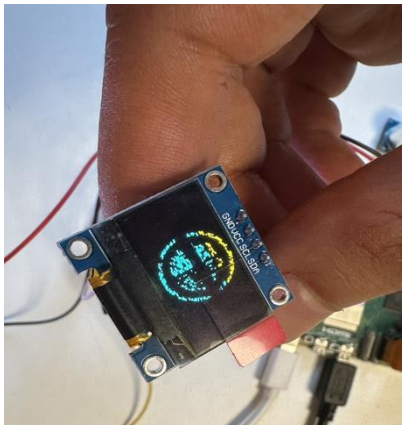
```
Python3 oled_alternate.py
```

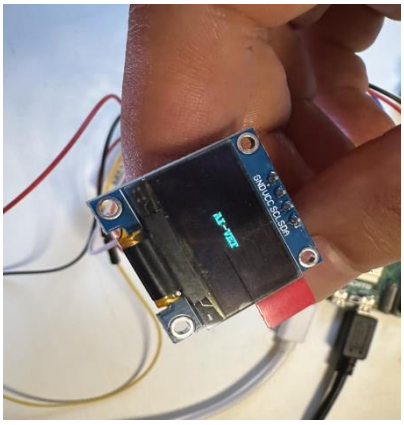
Deves ver o OLED:

- Primeiro, mostre o texto "AI-VET" centrado no ecrã
- Depois muda para a imagem do logótipo da AI-VET
- Continue a alternar a cada 2 segundos

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.





---

11. Critérios de Avaliação:

- **Precisão do Circuito:** OLED está corretamente ligado a pinos I^2C (SDA, SCL).
- **Inicialização do Ecrã:** O ecrã é detetado e limpo com sucesso pelo script.
- **Renderização de Texto:** O texto está devidamente centrado e legível no ecrã.
- **Gestão de Imagens:** O programa carrega e redimensiona as imagens com sucesso sem distorção.
- **Estabilidade do script:** Sem erros de sintaxe ou runtime; o script sai limpo em Ctrl + C.

---

12. Línguas Disponíveis:

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

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.

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.