AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Sucessos Criativos, Lda

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

1. Project Title: 4

2. Project Category: 4

3. Project Description: 4

4. Keywords: 4

5. Target Audience: 4

6. Project Objectives: 4

7. Project Timeline: 5

8. Resources Provided: 5

9. Tools and Technologies Used: 6

10. Step-by-Step Implementation Plan: 7

Step 1: Prepare the Raspberry Pi for Hardware Interaction 7

Step 2:Prepare the Strip 8

Step 3: Build the Circuit 9

Step 4: Write the Python Program to Control the LEDs 10

Step 5: Run the Program and Observe the Output 11

11. Evaluation Criteria: 12

12. Languages Available: 13

13. Contributors: 13

14. Feedback and Continuous Improvement: 13

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Project Title:

First Steps with AI-VET Kit: Controlling NeoPixel Strip

---

2. Project Category:

Microelectronics Fundamentals

---

3. Project Description:

This project demonstrates how to control a WS2812 NeoPixel LED strip using a Raspberry Pi 5. Learners will explore how to create visual lighting effects using SPI communication and Python code. The project introduces programmable LED control, RGB color manipulation, and animated light patterns—perfect for interactive or decorative applications.

---

4. Keywords:

NeoPixel, WS2812, Raspberry Pi, RGB LEDs, SPI, Python, Light Animation

---

5. Target Audience:

Educators, learners, and makers with or without prior experience who are beginning their journey into microelectronics and physical computing.

---

6. Project Objectives:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Wire a WS2812 NeoPixel LED strip to the Raspberry Pi using SPI
- Install and use the `adafruit-circuitpython-neopixel-spi` Python library
- Generate dynamic lighting effects using loops and random colors
- Understand how SPI works in the context of RGB LED control

---

7. Project Timeline:

The estimated duration is approximately **2.5 to 3 hours**, including environment setup and physical assembly.

---

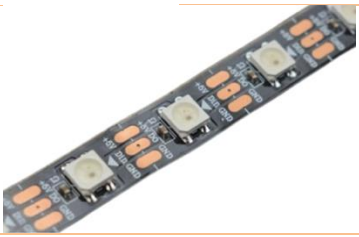
8. Resources Provided:

- AI-VET Kit (Raspberry Pi 5, LED, Breadboard, Jumpers, WS2812 RGB LED Strip)

MATERIAL	
Raspberry Pi 5	

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

27W USB Type-C Power Supply for Raspberry Pi 5	
Flat Jumper Wires F/F	
Breadboard	
SINLGE-FILE WIRE FOR BREADBOARDS	
WS2812 RGB Programmable LED Strip 15x LED	

---

9. Tools and Technologies Used:

- Raspberry Pi OS
- Python 3
- `adafruit-circuitpython-neopixel-spi` library
- Geany IDE
- Terminal (bash)

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

10. Step-by-Step Implementation Plan:

Step 1: Prepare the Raspberry Pi for Hardware Interaction

This project builds directly on the tasks completed in **"Getting Started with Raspberry Pi 5: Setup, Configuration, and Hello World in Python."** If you're starting here, or need a refresher, begin by preparing your Raspberry Pi for hardware-level communication using Python.

In this step, you will:

- **Install Raspberry Pi OS** using the Raspberry Pi Imager tool
- **Configure system settings** such as Wi-Fi, keyboard layout, and time zone
- **Enable key interfaces** via the Raspberry Pi Configuration menu:
    - **SPI** (required for NeoPixel control)
    - **I²C**, 1-Wire, and VNC (optional but useful for other projects)
- Use the `ifconfig` command in the terminal to **identify your Raspberry Pi's IP address** and connect remotely via **TigerVNC Viewer** if needed

Install Required Python Libraries

You'll now install the libraries used to interact with SPI-based NeoPixel strips and, optionally, OLED displays:

- **To control NeoPixel LEDs**, install the Adafruit SPI-compatible driver:

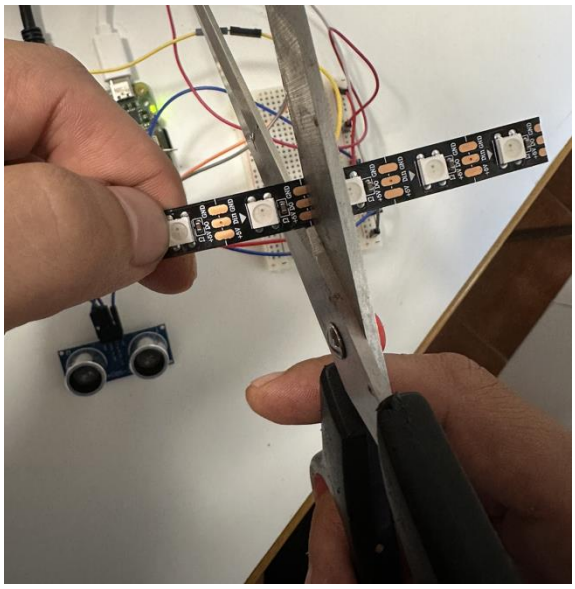
```
pip3 install adafruit-circuitpython-neopixel-spi
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 2:Prepare the Strip

Its possible to customize the length by cutting it to have the exact number of LEDs you need. It's important to cut the strip carefully at the designated cutting lines, which are marked between every LED:These lines indicate safe cutting points, ensuring you don't damage the internal wiring:



After cutting, you'll need to solder the necessary connectors to the exposed pads (5V, GND, and Data In/Out). Make sure to match the orientation of the connectors as shown in the picture, ensuring proper connection and reliable operation:



The final result is shown on the following image:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Step 3: Build the Circuit

In this step, you'll connect your NeoPixel LED strip to the Raspberry Pi using SPI. The WS2812 strip requires accurate timing for communication, making SPI an excellent interface.

NeoPixel Strip Connections:

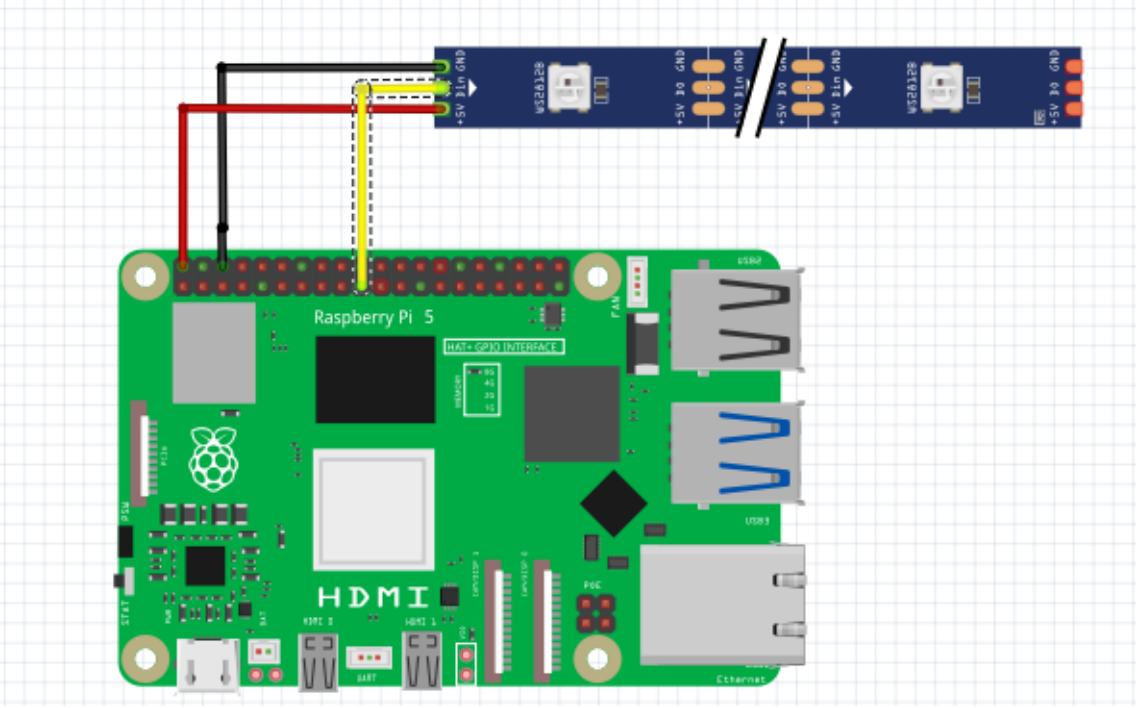
- **5V → Raspberry Pi 5V (pin 2)**
  Powers the LED strip

- **GND → Raspberry Pi GND (pin 6)**
  Establishes a common ground

- **Data In → GPIO 10 (pin 19)**
  The SPI MOSI (Master Out, Slave In) pin sends RGB data to the LED strip

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Step 4: Write the Python Program to Control the LEDs

With the wiring complete, it's time to write your first lighting animation using SPI.

Open Geany and enter the following code:

```
import board
import neopixel_spi as neopixel
import time
import random  # Import random to generate random colors
# Define the number of LEDs in the NeoPixel strip
NUM_PIXELS = 6
# Define the order of colors for the NeoPixel (Green, Red, Blue)
PIXEL_ORDER = neopixel.GRB
# Define a delay between color changes
DELAY = 0.1
# Initialize the SPI communication for controlling NeoPixel LEDs
spi = board.SPI()
# Create a NeoPixel object for controlling the LEDs via SPI
pixels = neopixel.NeoPixel_SPI(spi,
                               NUM_PIXELS,
                               pixel_order=PIXEL_ORDER,
                               auto_write=False)  # Set auto_write to False
for manual updates
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
# Main loop to change the colors of the LEDs
while True:
    # Generate a random color for each iteration
    color = (random.randint(0, 255), random.randint(0, 255), random.randint(0,
255))  # Random RGB color

    # Iterate through each pixel and set it to the randomly generated color
    for i in range(NUM_PIXELS):
        pixels[i] = color  # Assign the random color to the current pixel
        pixels.show()  # Update the LEDs to display the new color
        time.sleep(DELAY)  # Delay between each LED change

        # Clear the LEDs by setting them to 0 (off) after the delay
        pixels.fill(0)
```

Save the file as **`neopixel_random.py`**

Code Breakdown:

Setup and Imports

Imports the `neopixel_spi` library and Python's `random` and `time` modules for lighting logic and animation.

Configuration

- `NUM_PIXELS` defines how many LEDs are active
- The pixel order is set to GRB (common for WS2812 strips)

Pixel Control

- A random RGB color is generated
- Each pixel is set to that color with a short delay
- After displaying the animation, the strip is cleared

Step 5: Run the Program and Observe the Output

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

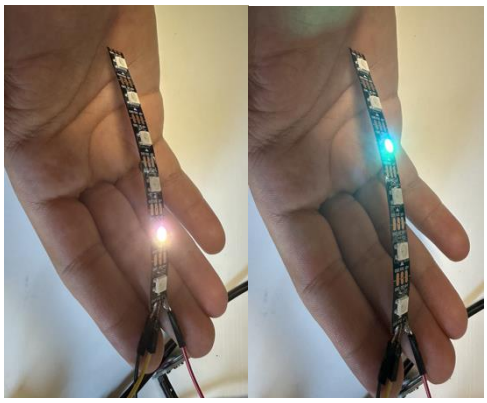
Let's light things up!

To **run** the script:

1. **Save** the program as `neopixel_random.py`
2. In the **terminal**, **type**:

```
python3 neopixel_random.py
```

You should see your strip displaying random colors, one LED at a time, in a loop.



---

11. Evaluation Criteria:

- **Hardware Setup**: NeoPixel strip is properly soldered and wired
- **SPI Communication**: Raspberry Pi SPI is enabled and the strip receives data
- **Visual Output**: The LED strip displays dynamic color changes as programmed

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- **Code Execution**: Python script runs without errors and updates LEDs in real-time
- **Creative Exploration**: Learner modified or extended code for personalized lighting effects

---

12. Languages Available:

English, **Portuguese** (translations in progress)

---

13. Contributors:

Fátima Pais, Luís Soares – Project Designers and Facilitators

---

14. Feedback and Continuous Improvement:

- **Surveys**: At the end of this milestone, students and educators will complete feedback surveys to assess the effectiveness of the project.
- **Project Reviews**: After project completion, a final review meeting will be held to discuss potential improvements for future AI-based projects.
- **Data Analysis**: Feedback data will be analyzed to identify strengths and areas for improvement, informing future projects.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.