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: 5

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: Build the Circuit 7

Step 3: Write the Python Program to Control the Window 9

Step 4: Run the Program and Observe The Output 12

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: Light-Controlled Window Automation For Energy Efficiency

---

2. Project Category:

Microelectronics Fundamentals

---

3. Project Description:

This project combines ambient light sensing with automated actuation to open or close a window based on real-time light intensity. Using an LDR (Light Dependent Resistor) sensor and a servo motor connected to a Raspberry Pi 5, learners will build a responsive system that supports energy efficiency. When light is present, the servo opens the window, promoting ventilation and natural lighting. When light is absent, the window closes, helping conserve heat or reduce cold airflow. This activity blends sustainability goals with foundational IoT principles.

---

4. Keywords:

Raspberry Pi, Python, LDR, Servo Motor, GPIO, Smart Home, Energy Efficiency

---

5. Target Audience:

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

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.

6. Project Objectives:

- Interface an LDR sensor and servo motor with a Raspberry Pi 5
- Build a real-time light-activated servo control system
- Understand PWM (Pulse Width Modulation) principles
- Promote sustainable automation using natural light and air

---

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, LDR Sensor, ModuleMicroservo Motor)

MATERIAL	
Raspberry Pi 5	
27W USB Type-C Power Supply for 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.

Flat Jumper Wires F/F	
Breadboard	
SINGLE-FILE WIRE FOR BREADBOARDS	
LDR Sensor Module with Potentiometer	
Microservo SG90 1.6Kg 180º (IM4)	

---

9. Tools and Technologies Used:

- Raspberry Pi OS
- Python 3 (in virtual environment)
- `lgpio` 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” and “Blinking an LED”. If you're starting here, or need a refresher, you’ll first go through the essential setup steps to prepare your Raspberry Pi to control physical components.

In this step, you will:

- Install Raspberry Pi OS using the Raspberry Pi Imager tool.
- Configure basic system settings such as Wi-Fi, keyboard layout, and time zone.
- Enable key interfaces including SPI, I2C, 1-Wire, and VNC using the Raspberry Pi Configuration menu.
- Identify the Raspberry Pi’s IP address using `ifconfig` and connect remotely via TigerVNC Viewer.
- Perform a full system update.
- Install the `lgpio` library

For best results, complete the relevant foundational projects beforehand:

- "First Steps with AI-VET Kit: Detecting Light Levels with an LDR Sensor and Raspberry Pi 5"
- "First Steps with AI-VET Kit: Controlling a Servo Motor with Raspberry Pi 5"

These guides provide step-by-step instructions for connecting and testing each component individually before combining them in this project.

Step 2: Build the Circuit

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.

In this step, you'll wire the components to the Raspberry Pi. The system uses a Light Dependent Resistor (LDR) sensor to detect ambient light and a servo motor to open or close a window based on the light level.

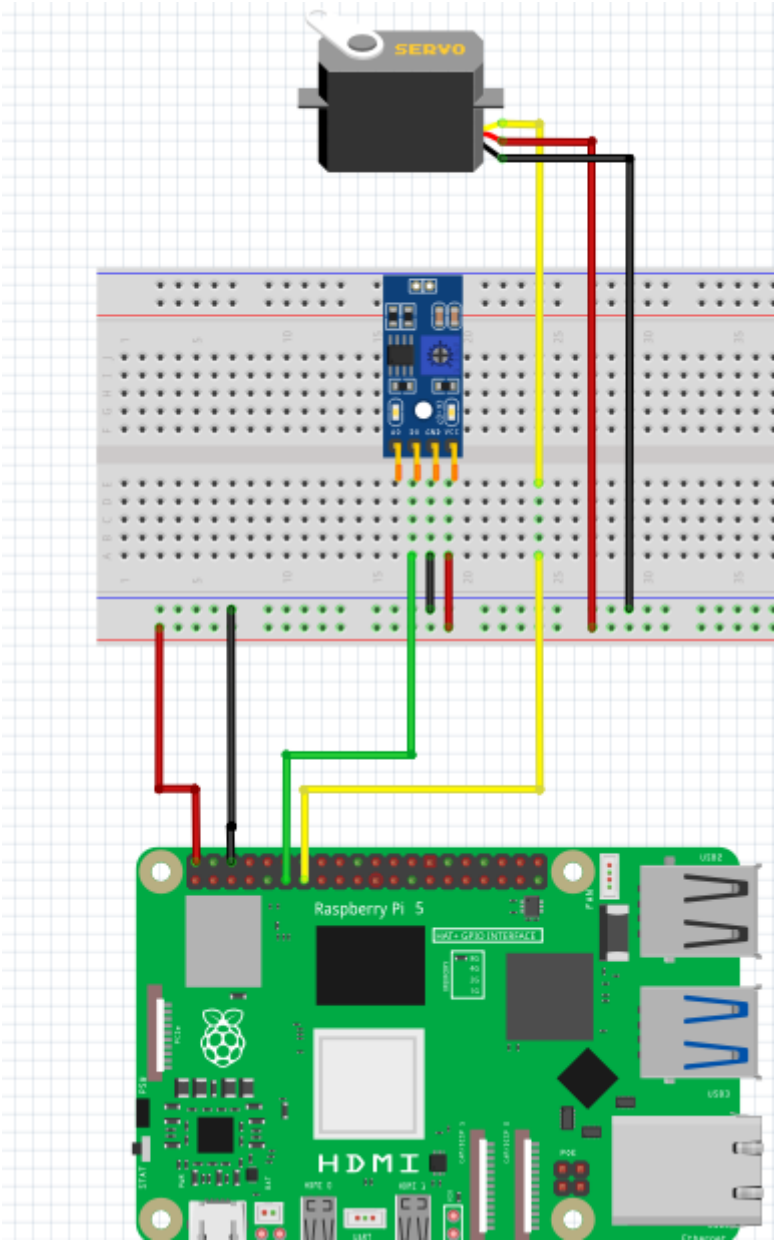
- **LDR Sensor (Light Detection):** Sends a digital signal indicating whether light is detected or not.
- **Servo Motor (Movement Control):** Moves to a specific angle depending on light conditions (0° for closed, 180° for open).

Connections:

- **LDR Sensor:**
  - VCC → Raspberry Pi 5V (pin 2)
  - GND → Raspberry Pi GND (pin 6)
  - OUT → GPIO 17 (pin 11)
- **Servo Motor (SG90):**
  - Red (VCC) → 5V (pin 4)
  - Brown (GND) → GND (pin 9)
  - Orange (Signal) → GPIO 27 (pin 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.



Step 3: Write the Python Program to Control the Window

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.

With the sensors connected, you'll now write a Python program that combines these sensor readings into one small smart system that prints a message in the terminal based on the result.

Open Geany and enter the following code:

```
import lgpio
import time
# GPIO pins
SERVO_PIN = 27  # Pin for the servo motor
LDR_PIN = 17    # Pin for the LDR sensor
# Initialize GPIO chip (the default GPIO chip is usually "/dev/gpiochip0")
h = lgpio.gpiochip_open(0)
# Claim the GPIO pins
lgpio.gpio_claim_output(h, SERVO_PIN)  # Servo as output
lgpio.gpio_claim_input(h, LDR_PIN)     # LDR as input
# Function to generate PWM for servo motor
def set_servo_pulsewidth(h, pin, pulse_width):
    """
    Sends a high signal to the servo pin for the pulse width (in milliseconds).
    This creates a PWM signal by holding the pin HIGH for a specific duration.
    """
    lgpio.gpio_write(h, pin, 1)               # Set pin HIGH (start of pulse)
    time.sleep(pulse_width / 1000)           # Wait for the pulse width duration
    lgpio.gpio_write(h, pin, 0)               # Set pin LOW (end of pulse)
# Function to move the servo to a specified angle
def move_servo(angle):
    """
    Convert the given angle (between 0 and 180 degrees) into a corresponding
pulse width
    and move the servo to that position.
    """
    pulse_width = 1 + (angle / 180.0)          # Calculate pulse width (1ms to
2ms)
    set_servo_pulsewidth(h, SERVO_PIN, pulse_width)  # Set the pulse width to
move the servo
# Main loop to control the servo based on LDR sensor input
try:
    while True:
        # Read the current state of the LDR sensor
        light_state = lgpio.gpio_read(h, LDR_PIN)
        # If light is detected (LDR sensor returns 0), move the servo to 180
degrees
        if light_state == 0:
            print("Light detected! Moving servo to 180 degrees.")
            move_servo(180)
```

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.

```
        # If no light is detected (LDR sensor returns 1), move the servo to 0
degrees
        else:
            print("No light detected! Moving servo to 0 degrees.")
            move_servo(0)
        # Add a small delay to prevent continuous readings and allow time for
changes
        time.sleep(1)
except KeyboardInterrupt:
    # Handle user interruption (Ctrl + C) to stop the program
    print("\nStopping servo control and exiting the program.")
    # Clean up and close the GPIO resources before exiting the program
    lgpio.gpio_write(h, SERVO_PIN, 0)      # Turn off the SERVO_PIN before exit
    lgpio.gpiochip_close(h)

    print("GPIO cleaned up!")
```

Save the file as `light_controlled_window.py`

Code Breakdown:

Setup and Imports:
Imports `lgpio` for GPIO interaction and `time` for delays.

Pin Definition and Initialization:
Defines GPIO 17 for the LDR sensor and GPIO 27 for the servo. Claims these pins for input/output.

PWM Control for Servo:
A custom function `set_servo_pulsewidth` mimics PWM by holding the GPIO high for a calculated duration.

Window Control Logic:

- If **light is detected**, the servo rotates to 180° (opens the window)
- If **no light**, the servo returns to 0° (closes the window)

Loop and Interruption:
The script continuously monitors the light level and adjusts the servo every second. If Ctrl + C is pressed, the script exits and cleans up GPIO.

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: Run the Program and Observe The Output

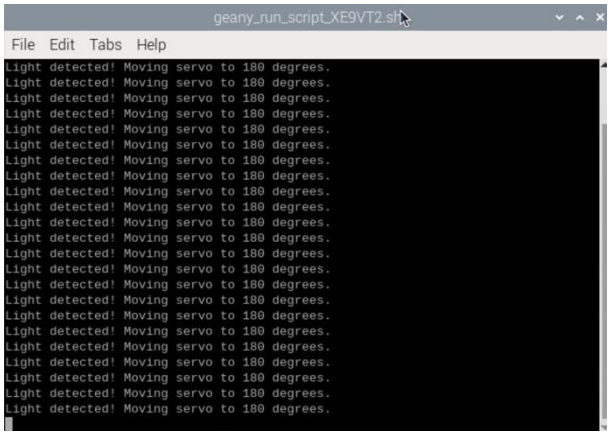
With the code written and saved, it's time to test the setup and observe how it responds to nearby objects.

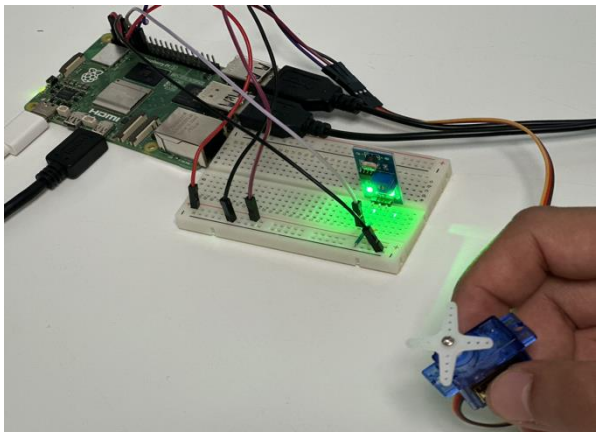
To run the script:

Save the script as **light_controlled_window.py**

Press F5 in **Geany** or **open** the **terminal** and **run**:

python3 light_controlled_window.py





---

11. Evaluation Criteria:

- **LDR Sensor Integration**: The light sensor is correctly connected, configured, and able to distinguish between light and dark conditions.
- **Servo Motor Functionality**: The servo responds accurately to changes in light, rotating between 0° and 180° based on sensor input.

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 Performance**: The script runs without syntax or runtime errors, performs the correct logic, and exits gracefully when interrupted.
- **Energy Efficiency Relevance**: Learner understands how the automation contributes to sustainability by reducing reliance on artificial lighting or climate control.
- **System Testing:** The system was tested under real conditions with both high and low light inputs, demonstrating correct behavior.

---

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.

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.