AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Sucessos Criativos, Lda | P. Porto

Title of WP: WP3 – AI-SAVES@ProjectHub

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

European Vocational Training Association

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: 2

2. Project Category: 3

3. Project Description: 3

5. Target Audience: 3

6. Pre-Requisite knowledge: 3

7. Project Objectives: 3

8. Project Timeline: 4

9. Resources Provided: 4

10. Tools and Technologies Used: 5

11. Step-by-Step Implementation Plan: 7

12. Evaluation Criteria: 13

13. Languages Available: 13

14. Contributors: 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:

Light Tracking Servo System with Raspberry Pi

---

2. Project Category:

- AI in IoT

---

3. Project Description:

This project aims to develop a light tracking system using a Raspberry Pi, LDR (Light Dependent Resistor) sensors, and a servo motor. The system detects light intensity from two directions and automatically adjusts the servo motor to follow the brightest light source. This project demonstrates how sensors and actuators can be combined to create responsive IoT systems capable of environmental interaction and automated decision-making.

---

4. Keywords:

Light Tracking, Raspberry Pi, IoT, LDR Sensors, Servo Motor, Automation

---

5. Target Audience:

Both Educators and Students

---

6. Pre-Requisite knowledge:

- Basic understanding of Ohm's Law and electrical circuits
- Introduction to Python programming (variables, loops, functions)
- Knowledge of Raspberry Pi GPIO control
- Basic understanding of sensors and servo motors

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.

7. Project Objectives:

- Introduce students to light detection using LDR sensors.
- Teach how to control servo motors with Raspberry Pi.
- Demonstrate real-time decision-making based on sensor input.
- Develop hands-on skills in electronics and Python programming.
- Encourage the development of responsive IoT systems.

---

8. Project Timeline:

Flexible timeline depending on classroom or workshop schedule.

---

9. Resources Provided:

- Raspberry Pi
- Servo motor (180-degree)
- 2 LDR sensors
- Breadboard and jumper wires
- External power supply for servo
- Python code templates
- Step-by-step instructional guide

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. Tools and Technologies Used:

10.1 Hardware: Raspberry Pi, Servo Motor, 2x LDR sensors, Breadboard, Jumper wires.

10.2 Software: Raspbian OS, Python 3

- Operating System: Raspberry Pi OS
- Programming Environment: Geany or Python IDE
- Python Libraries: lgpio, time

Libraries: Adafruit_DHT, RPi.GPIO

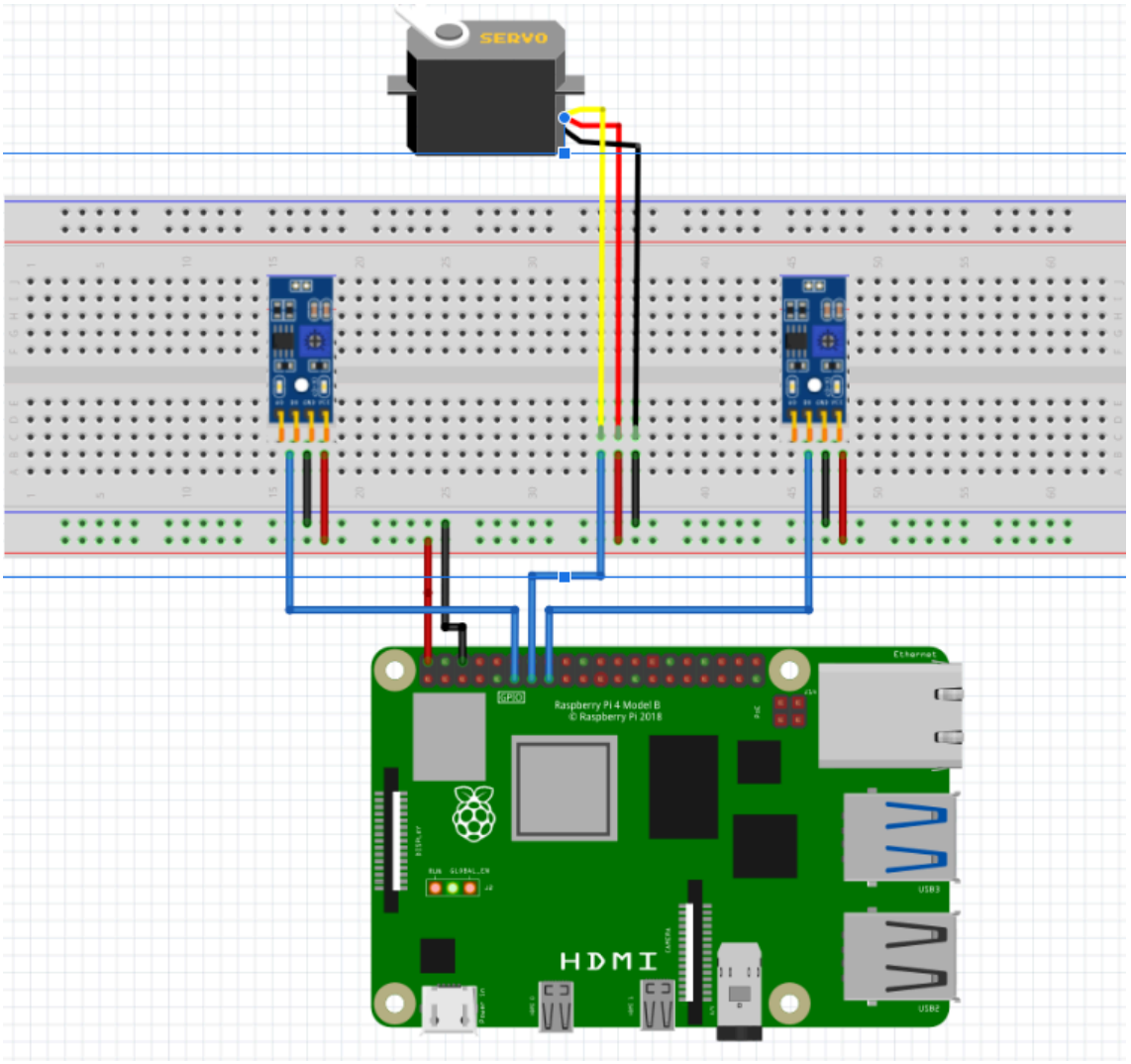
---

11. Step-by-Step Implementation Plan:

11.1 Hardware Setup

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.



Circuit Connections

- Servo Motor:
  - Signal ⟶ GPIO 27 (Pin 13)
  - Power ⟶ 5V (Pin 2 or Pin 4) (external power supply recommended)
  - Ground ⟶ Ground (Pin 6)
- LDR Sensors:
  - Left LDR ⟶ GPIO 17 (Pin 11)
  - Right LDR ⟶ GPIO 22 (Pin 15)
- Ground Connection: Use a common Ground (Pin 6, 9, 14, or 20) for all components.
- LDR Circuit Setup: The LDR sensors are connected directly to GPIO pins:
  - LDR connected between GPIO pin and GND

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.

- When light hits the LDR: GPIO reads LOW (0)
- When there is no light: GPIO reads HIGH (1)

Wiring Steps

1. Connect the servo signal wire to GPIO 27 (Pin 13).
2. Connect the servo power wire to 5V.
3. It is recommended to use an external 5V power supply for stable servo performance.
4. Connect the servo ground wire to Raspberry Pi Ground.
5. Connect the left LDR sensor to GPIO 17 (Pin 11).
6. Connect the right LDR sensor to GPIO 22 (Pin 15).
7. Connect both LDR sensors to the common ground.
8. Make sure all components share the same ground reference.

⚠️ Important: Once you have assembled the circuit, double-check all connections before proceeding. Make sure each wire is properly connected to the correct pin. Incorrect wiring can cause unstable behavior, sensor faults, or damage to the Raspberry Pi and servo motor.

11.2 Software Setup

Now that the hardware is ready, it's time to set up the code that will bring your temperature monitoring system to life.

1. Create Virtual Environment

```
# Create project directory
mkdir light_tracker
cd light_tracker

# Create virtual environment
python3 -m venv venv

# Activate virtual environment
source venv/bin/actívate
```

2. Install Dependencies

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.

```
# Install requirements
pip install lgpio
```

3. Add the Code

- Copy the Python code to light_tracker.py
- Make sure the file is in your project directory

4. Run the System

```
bash
# Make sure virtual environment is activated
source venv/bin/activate

# Run the monitoring system
python light_tracker.py
```

5. Stop the System

- Press Ctrl+C to stop the program safely
- The servo will stop and GPIO resources will be released automatically

System Behavior

- Center position (90°): both sensors detect light or no directional preference
- Left position (45°): only the left LDR detects light
- Right position (135°): only the right LDR detects light
- The system checks the sensors continuously with a 1-second delay between updates

Your temperature monitoring system is now ready to use!

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

```
import time
import lgpio

# GPIO pin definitions
SERVO_PIN = 27
LDR_PIN_LEFT = 17
LDR_PIN_RIGHT = 22

# Servo positions
CENTER_POSITION = 90
LEFT_POSITION = 45
RIGHT_POSITION = 135

# Timing
RESPONSE_DELAY = 1  # seconds

# Open GPIO chip
h = lgpio.gpiochip_open(0)

# Claim LDR pins as inputs
lgpio.gpio_claim_input(h, LDR_PIN_LEFT)
lgpio.gpio_claim_input(h, LDR_PIN_RIGHT)

def set_servo_angle(angle):
    """
    Move servo to desired angle using PWM pulse width.
    0° -> 1ms
    90° -> 1.5ms
```

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.

```
    180° -> 2ms
    """
    pulse_width_ms = 1 + (angle / 180.0)
    lgpio.tx_servo(h, SERVO_PIN, pulse_width_ms * 1000)

try:
    # Start centered
    set_servo_angle(CENTER_POSITION)
    current_angle = CENTER_POSITION

    while True:
        # Read LDR sensor values
        light_left = lgpio.gpio_read(h, LDR_PIN_LEFT)
        light_right = lgpio.gpio_read(h, LDR_PIN_RIGHT)

        # LDR logic:
        # LOW (0) = light detected
        # HIGH (1) = no light detected

        if light_left == 0 and light_right == 1:
            current_angle = LEFT_POSITION
        elif light_right == 0 and light_left == 1:
            current_angle = RIGHT_POSITION
        else:
            current_angle = CENTER_POSITION

        set_servo_angle(current_angle)

          print(f"Left LDR: {light_left}, Right LDR: {light_right},
Servo angle: {current_angle}")

        time.sleep(RESPONSE_DELAY)

except KeyboardInterrupt:
    print("Shutting down light tracking system...")

finally:
```

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.

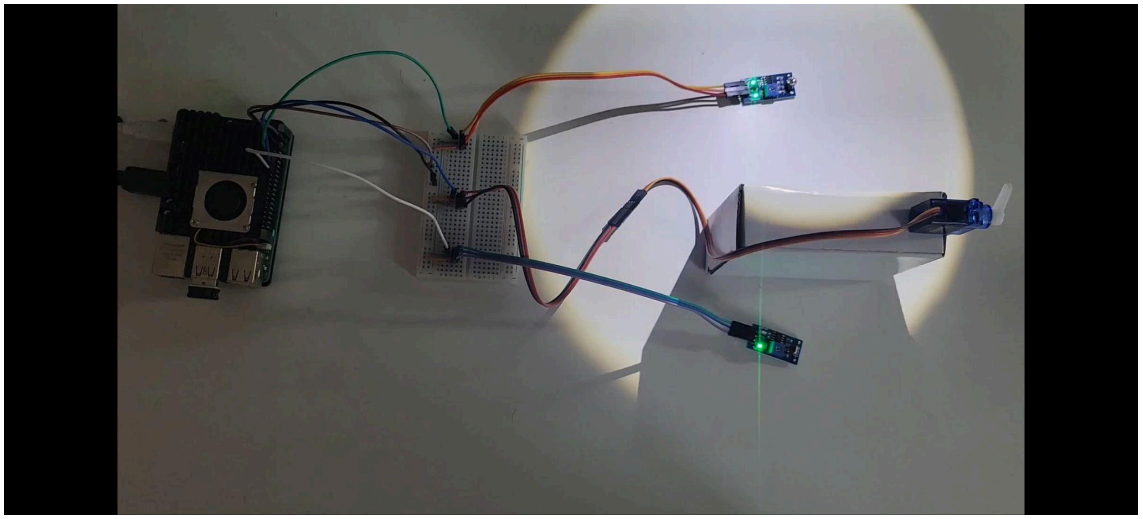
```
    # Stop servo signal
    lgpio.tx_servo(h, SERVO_PIN, 0)
    lgpio.gpiochip_close(h)
```

Now, you can run the program and see the result:

Co-funded by the European Union

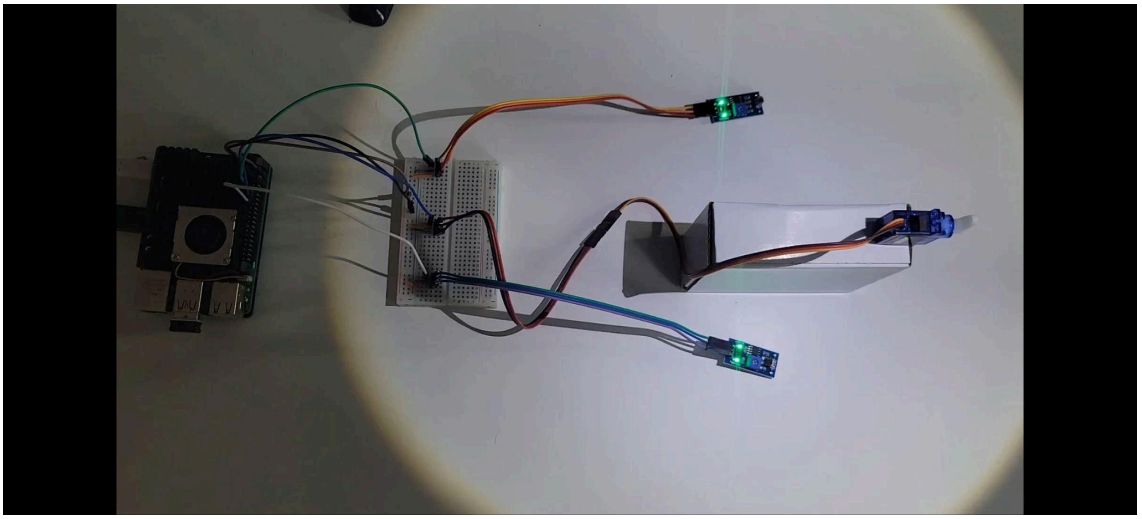
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

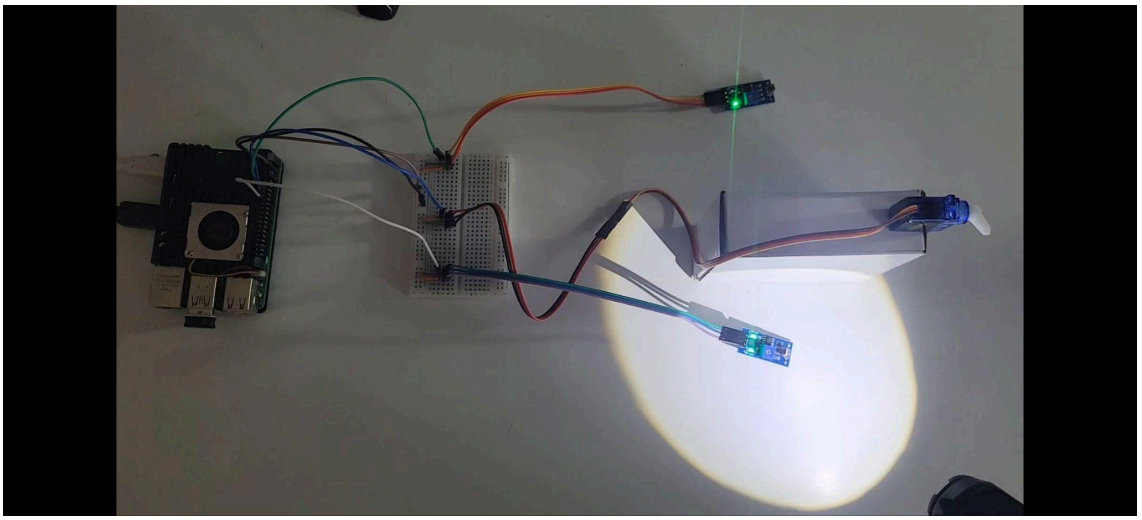
```
File  Edit  Tabs  Help
GPIO cleaned up and resources released.
(env) pi@raspberrypi-2:~ $ python placa.py
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in straight position.
Light detected on the right! Moving servo towards the right.
Light detected on the right! Moving servo towards the right.
Light detected on both sides! Moving servo to straight position.
Light detected on the left! Moving servo towards the left.
Light detected on the left! Moving servo towards the left.
Light detected on the left! Moving servo towards the left.
Light detected on the left! Moving servo towards the left.
Light detected on the right! Moving servo towards the right.
Light detected on the right! Moving servo towards the right.
Light detected on the left! Moving servo towards the left.
Light detected on the left! Moving servo towards the left.
Light detected on both sides! Moving servo to straight position.
```



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.





12. Evaluation Criteria:

EVALUATION CRITERIA

Hardware Setup (35%): Correct wiring of sensors and servo motor; stable and functional circuit.

Code Functionality (30%): Code runs correctly and servo responds to light detection.

System Understanding (20%): Clear explanation of how sensors control servo positioning.

Innovation & Improvement (15%): Creative enhancements such as improved tracking or additional sensors.

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.

13. Languages Available:

14. Contributors:

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.