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: 2
3. Project Description: 2
5. Target Audience: 2
6. Pre-Requisite knowledge: 2
7. Project Objectives: 3
8. Project Timeline: 3
9. Resources Provided: 3
10. Tools and Technologies Used: 4
11. Step-by-Step Implementation Plan: 6
12. Evaluation Criteria: 17
13. Languages Available: 17
14. Contributors: 17

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:

Motion Detection System with Smart Lighting

---

2. Project Category:

- AI in IoT

---

3. Project Description:

This project involves developing an intelligent motion detection system using a Raspberry Pi that automatically controls LED strip lighting based on movement detection and ambient light conditions. The system uses an ultrasonic sensor to detect nearby movement and only activates lighting when it's dark, providing an energy-efficient automated lighting solution. This smart system contributes to energy conservation and enhances safety in various environments such as hallways, stairways, and entrances.

---

4. Keywords:

Motion Detection, Smart Lighting, Raspberry Pi, IoT, Ultrasonic Sensor, Energy Efficiency, Automated Systems, Environmental Sustainability

---

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

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.

- Basic experience with SPI communication protocol
- Understanding of sensor interfacing and signal processing
- Familiarity with LED strip technology and control methods

---

7. Project Objectives:

- Introduce students to advanced IoT sensor integration and automation
- Teach real-time data acquisition from multiple sensors simultaneously
- Develop understanding of energy-efficient smart lighting systems
- Promote environmental sustainability through intelligent automation
- Enhance hands-on skills in electronics, Python programming, and system integration
- Encourage creation of practical, scalable solutions for real-world applications
- Foster understanding of motion detection algorithms and ambient light sensing

---

8. Project Timeline:

---

9. Resources Provided:

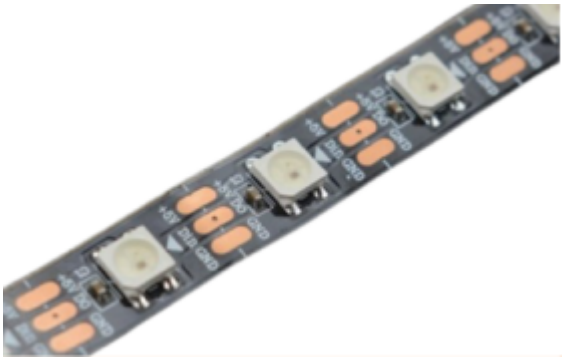
- Raspberry Pi 5 with SPI enabled
- HC-SR04 Ultrasonic Distance Sensor
- LDR (Light Dependent Resistor) sensor
- NeoPixel LED Strip (WS2812B compatible)
- Breadboard and jumper wires
- External power supply for LED strip
- Python code templates
- Comprehensive step-by-step instructional guide
- Wiring diagrams and troubleshooting 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.

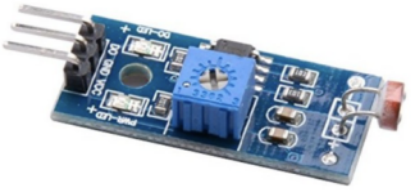
10. Tools and Technologies Used:

10.1 Hardware:

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

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.

HC-SR04P - ULTRASONIC SENSOR 3V~5V	
LDR Sensor Module with Potentiometer	

10.2 Software:

- **Operating System:** Raspberry Pi OS with SPI enabled
- **Programming Language:** Python 3.7+
- **Programming Environment:** Python 3.7+ with Geany or Python IDE

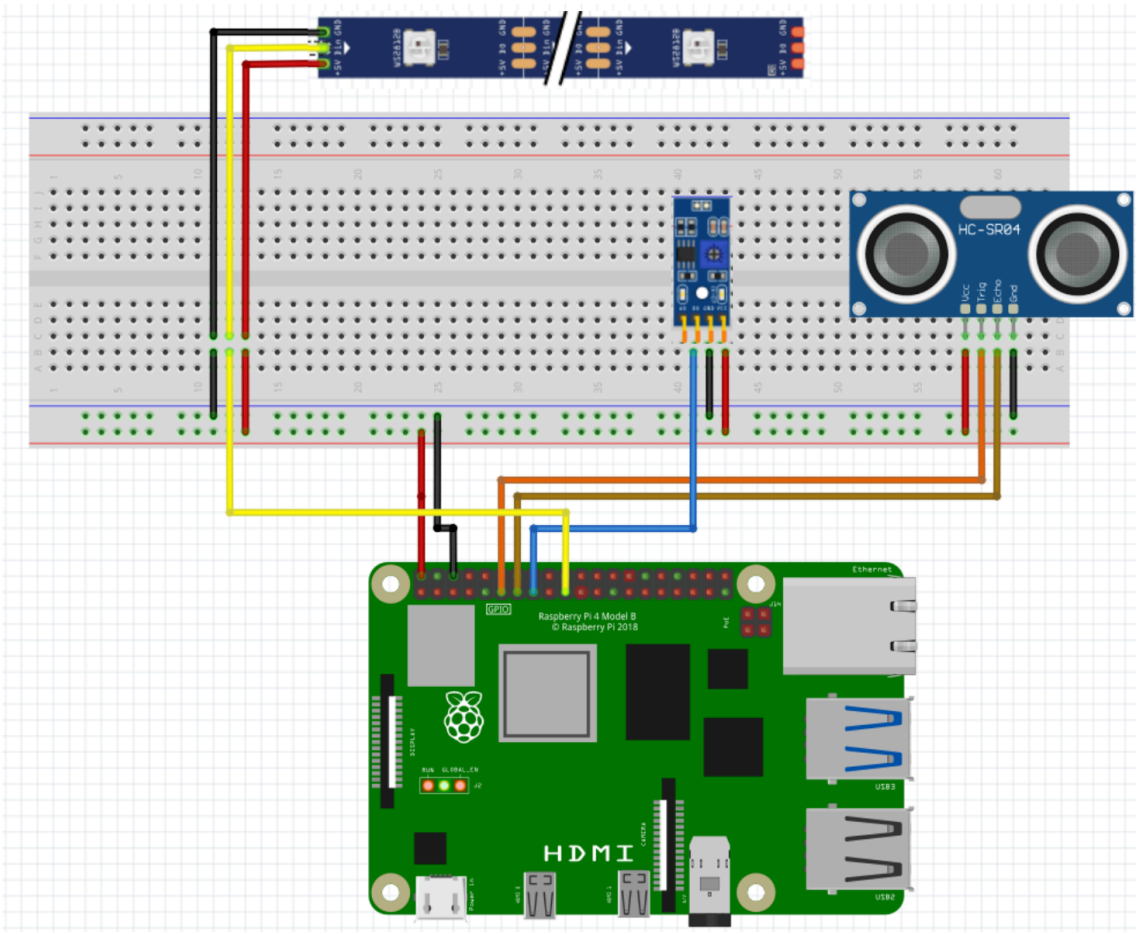
Libraries: RPi.GPIO, time, board, neopixel

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.

11. Step-by-Step Implementation Plan:

11.1 Hardware Setup



Circuit Connections

HC-SR04 Ultrasonic Sensor:

- VCC ⟶ 5V (Pin 2)
- GND ⟶ Ground (Pin 6)
- Trigger ⟶ GPIO 17 (Pin 11)
- Echo ⟶ GPIO 27 (Pin 13)

LDR Sensor:

- One leg ⟶ 3.3V (Pin 1)

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.

- Other leg ⟶ GPIO 22 (Pin 15) and 10kΩ resistor to Ground (Pin 6)

NeoPixel LED Strip:

- VCC ⟶ 5V External Power Supply
- GND ⟶ Ground (Pin 6) and External Power Supply Ground
- Data ⟶ GPIO 10 (MOSI/Pin 19)

Ground Connection: Use Pin 6 (Ground) as common ground for all components

Wiring Steps

1. Connect HC-SR04 sensor: 5V power, Ground, Trigger to GPIO 17, Echo to GPIO 27
2. Set up LDR circuit
3. Connect NeoPixel LED strip data line to GPIO 10 (MOSI)
4. Ensure all ground connections are properly connected to common ground rail

⚠️ Important: Before powering on the system, carefully verify all connections. Incorrect wiring can damage components or cause system instability.

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.

11.2 Software Setup

Now that the hardware is ready, it's time to implement the light tracking control system.

1. Create Virtual Environment

```
# Create project directory
mkdir motion_detection_system
cd motion_detection_system

# Create virtual environment
python3 -m venv venv

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

2. Install Dependencies

```
# Install requirements
pip install -r requirements.txt
```

3. Add the Code

- Copy the motion detection system Python code to `main.py`
- Make sure the file is in your project directory

4. Run the System

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

# Run the motion detection system
python main.py
```

5. Stop the System

- Press Ctrl+C to stop the program safely
- All components will be properly cleaned up automatically

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.

System Behavior:

LED Behavior:

- **White LEDs:** Activated when motion detected in dark conditions
- **All OFF:** No motion detected or bright ambient light
- **Automatic Timeout:** LEDs turn off after 5 seconds
- **Smart Activation:** Only responds to motion when ambient light is low

Console Output:

- Real-time distance measurements
- Motion detection status
- Ambient light level readings
- System status messages

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 board
import lgpio
import neopixel_spi as neopixel

# GPIO pin definitions
TRIGGER_PIN = 17     # GPIO pin for ultrasonic sensor trigger
ECHO_PIN = 27        # GPIO pin for ultrasonic sensor echo
LDR_PIN = 22         # GPIO pin for LDR (Light Dependent Resistor) sensor

# NeoPixel LED strip configuration
NUM_PIXELS = 12                      # Number of LEDs in the strip
PIXEL_ORDER = neopixel.GRB           # Color order (Green, Red, Blue)
LED_COLOR = 0xFFFFFF                 # White color for motion detection
LED_BRIGHTNESS = 0.5                 # LED brightness (0.0 to 1.0)

# Motion detection parameters
MOTION_THRESHOLD = 50.0              # Distance threshold in cm to detect motion
LIGHT_ON_DURATION = 5.0                # How long to keep lights on after motion
(seconds)
MEASUREMENT_INTERVAL  =  0.1               # Time between distance measurements
(seconds)

# Initialize GPIO chip
h = lgpio.gpiochip_open(0)
lgpio.gpio_claim_output(h, TRIGGER_PIN)     # Set trigger pin as output
lgpio.gpio_claim_input(h, ECHO_PIN)         # Set echo pin as input
lgpio.gpio_claim_input(h, LDR_PIN)          # Set LDR pin as input

# Initialize NeoPixel LED strip
spi = board.SPI()
pixels = neopixel.NeoPixel_SPI(
    spi,
    NUM_PIXELS,
    brightness=LED_BRIGHTNESS,
    pixel_order=PIXEL_ORDER,
    auto_write=False
)

# Global variables for motion tracking
previous_distance = None
lights_on_until = 0  # Timestamp when lights should turn off
```

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.

```python
def measure_distance():
    """
    Measure distance using the ultrasonic sensor.

    Returns:
        float: Distance in centimeters, or None if measurement fails
    """
    try:
        # Send 10-microsecond pulse to trigger pin
        lgpio.gpio_write(h, TRIGGER_PIN, 1)
        time.sleep(0.00001)  # 10 microseconds
        lgpio.gpio_write(h, TRIGGER_PIN, 0)

        # Wait for echo to start (pin goes HIGH)
        start_time = time.time()
        timeout = start_time + 0.1  # 100ms timeout

        while lgpio.gpio_read(h, ECHO_PIN) == 0:
            start_time = time.time()
            if start_time > timeout:
                return None  # Timeout occurred

        # Wait for echo to end (pin goes LOW)
        end_time = time.time()
        while lgpio.gpio_read(h, ECHO_PIN) == 1:
            end_time = time.time()
            if end_time > timeout:
                return None  # Timeout occurred

        # Calculate distance based on time of flight
        pulse_duration = end_time - start_time
        distance = (pulse_duration * 34300) / 2  # Speed of sound: 343 m/s

        return distance

    except Exception as e:
        print(f"Error measuring distance: {e}")
        return None

def is_dark():
    """
    Check if ambient light level is low using LDR sensor.
```

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.

```
    Returns:
        bool: True if it's dark (LDR reads 1), False if there's light (LDR
reads 0)
    """
    try:
        return lgpio.gpio_read(h, LDR_PIN) == 1
    except Exception as e:
        print(f"Error reading LDR sensor: {e}")
        return False

def detect_motion(current_distance):
    """
    Detect motion by comparing current distance with previous measurement.

    Args:
        current_distance (float): Current distance measurement in cm

    Returns:
        bool: True if motion is detected, False otherwise
    """
    global previous_distance

    if previous_distance is None:
        previous_distance = current_distance
        return False

    # Calculate the change in distance
    distance_change = abs(current_distance - previous_distance)

    # Update previous distance for next comparison
    previous_distance = current_distance

    # Motion detected if distance change exceeds threshold
    return distance_change > 5.0  # 5cm change threshold

def turn_on_lights():
    """
    Turn on all LEDs in the strip with the specified color.
    """
    global lights_on_until
    # Set all pixels to the motion detection color
    for i in range(NUM_PIXELS):
```

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.

```
        pixels[i] = LED_COLOR

    pixels.show()

    # Set the time when lights should turn off
    lights_on_until = time.time() + LIGHT_ON_DURATION
    print("Motion detected in dark environment - Lights ON")

def turn_off_lights():
    """
    Turn off all LEDs in the strip.
    """
    # Set all pixels to black (off)
    pixels.fill(0x000000)
    pixels.show()

    print("Lights OFF")

def should_lights_be_on():
    """
    Check if lights should currently be on based on timer.
    Returns:
        bool: True if lights should be on, False otherwise
    """
    return time.time() < lights_on_until

def main():
    """
    Main program loop for motion detection system.
    """
    global lights_on_until

    print("Motion Detection System Starting...")
    print(f"Motion threshold: {MOTION_THRESHOLD} cm")
    print(f"Light duration: {LIGHT_ON_DURATION} seconds")
    print("Press Ctrl+C to exit")

    try:
        # Initialize with lights off
        turn_off_lights()

        while True:
            # Measure current distance
```

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.

```
            distance = measure_distance()

            if distance is not None:
                # Check if object is within motion threshold
                object_detected = distance <= MOTION_THRESHOLD

                # Check ambient light level
                dark_environment = is_dark()

                # Print status information
                        print(f"Distance: {distance:.1f}cm | Object detected:
{object_detected} | Dark: {dark_environment}")

                # Turn on lights if object detected in dark environment
                if object_detected and dark_environment:
                    turn_on_lights()

                # Turn off lights if timer expired (mandatory after timeout)
                if not should_lights_be_on() and lights_on_until > 0:
                    turn_off_lights()
                    lights_on_until = 0

            else:
                print("Distance measurement failed")
                # Turn off lights if timer expired even with failed measurement
                if not should_lights_be_on() and lights_on_until > 0:
                    turn_off_lights()
                    lights_on_until = 0

            # Wait before next measurement
            time.sleep(MEASUREMENT_INTERVAL)

    except KeyboardInterrupt:
        print("\nShutting down motion detection system...")

    finally:
        # Cleanup: turn off lights and close GPIO
        turn_off_lights()
        lgpio.gpiochip_close(h)
        print("System shutdown complete!")

if __name__ == "__main__":
    main()
```

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.

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

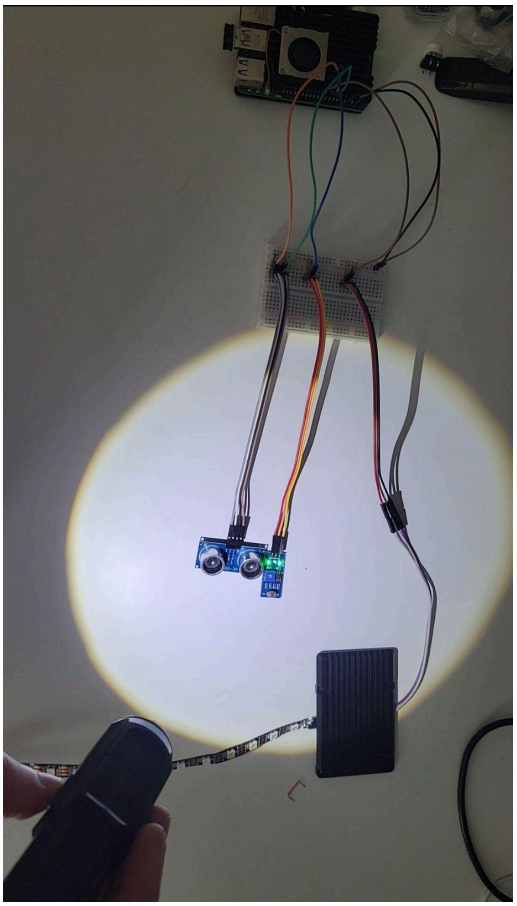
```
Distance: 8.9cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 8.9cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 9.2cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 9.2cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 7.8cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 7.8cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 9.5cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 13.3cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 16.0cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
Distance: 169.6cm | Object detected: False | Dark: True
Distance: 169.6cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.8cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
```

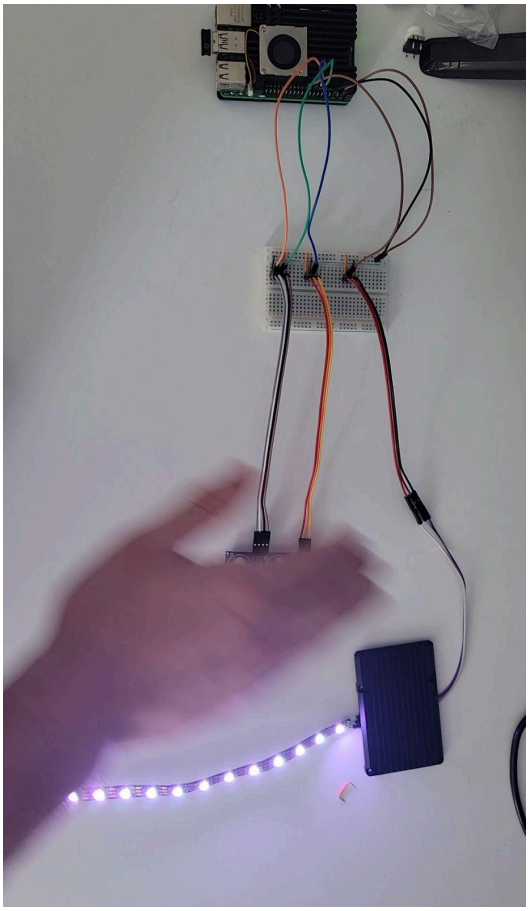
```
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.8cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 28.2cm | Object detected: True | Dark: False
Distance: 22.1cm | Object detected: True | Dark: False
Distance: 16.0cm | Object detected: True | Dark: False
Distance: 10.5cm | Object detected: True | Dark: False
Distance: 9.2cm | Object detected: True | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.8cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 169.7cm | Object detected: False | Dark: False
Distance: 170.1cm | Object detected: False | Dark: False
Distance: 8.5cm | Object detected: True | Dark: False
Distance: 10.5cm | Object detected: True | Dark: False
Distance: 16.3cm | Object detected: True | Dark: False
Distance: 24.7cm | Object detected: True | Dark: False
Distance: 41.3cm | Object detected: True | Dark: True
Motion detected in dark environment - Lights ON
```

```
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.8cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Lights OFF
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.2cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 170.1cm | Object detected: False | Dark: True
Distance: 169.7cm | Object detected: False | Dark: True
Distance: 169.8cm | Object detected: False | Dark: True
```

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.

12. Evaluation Criteria:

- **Hardware Setup (30%)**: Proper wiring and stable connection of Raspberry Pi, HC-SR04 ultrasonic sensor, LDR sensor, and NeoPixel LED strip; correct use of external power supply; all ground references aligned; no loose or incorrect connections.
- **Code Functionality (25%)**: System runs without errors; LEDs activate only under correct conditions (motion detected + low ambient light); sensors respond accurately; console outputs are informative and real-time.
- **System Understanding (20%)**: Clear explanation of how ultrasonic and LDR sensors work together to drive LED behavior; understanding of SPI communication; appropriate use of power and GPIO resources.
- **Energy Efficiency & Enhancement (15%)**: Demonstrates understanding of energy-saving through conditional lighting; includes ideas for improvement such as mobile app control, data logging, or solar power integration.
- **User Interaction & Safety (10%)**: System handles edge cases (e.g., false positives, sudden light changes); safe startup/shutdown procedures; proper component handling and safe power distribution.

---

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.