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: 6

Step 1: Prepare the Raspberry Pi for Hardware Interaction 7

Step 2: Connect the LDR Sensor 7

Step 3: Write the Python Program to Detect Light 8

Step 4: Run the Program and Observe Light Detection 10

Step 5: Steps to Adjust the Potentiometer 11

11. Evaluation Criteria: 12

12. Languages Available: 12

13. Contributors: 12

17. 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: Detecting Light Levels with an LDR Sensor and Raspberry Pi 5

---

2. Project Category:

Microelectronics Fundamentals

---

3. Project Description:

In this project, participants will learn how to detect the presence or absence of light using an LDR (Light Dependent Resistor) directly connected to a GPIO pin. This project follows the method where a basic GPIO read operation is used to determine whether the LDR is exposed to light or in darkness. This forms the basis for understanding digital sensor input and reactive behaviors.

---

4. Keywords:

Raspberry Pi, LDR Sensor, GPIO Input, Light Detection, Digital Sensing, lgpio

---

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.

- Build a digital input circuit using an LDR and connect it to a GPIO pin.
- Use the `lgpio` library to detect whether the sensor is exposed to light.
- Write a Python script to monitor and print light status to the terminal.
- Understand the logic behind digital sensor input using LDRs.

---

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)

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	

---

9. Tools and Technologies Used:

- Raspberry Pi OS
- Python 3 (in virtual environment)
- `lgpio` library
- Geany IDE
- Terminal (bash)

---

10. Step-by-Step Implementation Plan:

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 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

Step 2: Connect the LDR Sensor

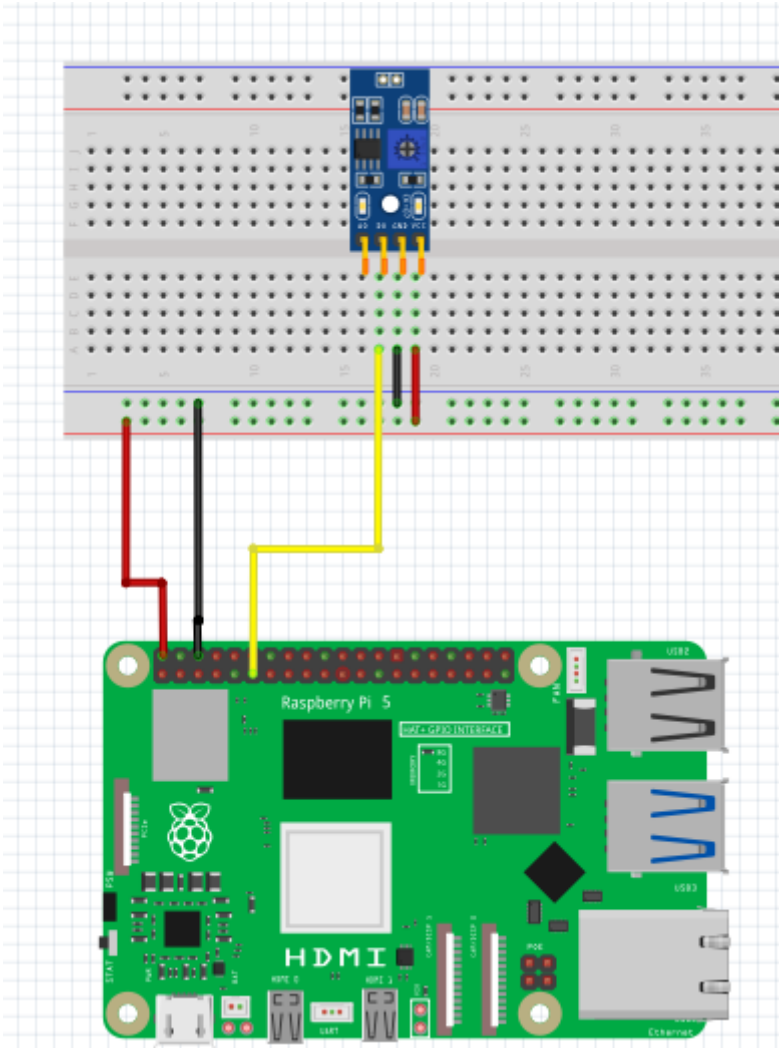
In this step, you will connect a digital LDR sensor module to the Raspberry Pi to detect changes in light. The module has three pins: **VCC**, **GND**, and **OUT**.

Make the following connections:

- Connect the **VCC** pin of the sensor to the Raspberry Pi’s **5V pin (physical pin 2)**
- Connect the **GND** pin of the sensor to a **GND pin (physical pin 6)**
- Connect the **OUT** pin to **GPIO17 (physical pin 11)**

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 Detect Light

With the sensor connected, you'll now write a Python program that checks whether light is present and prints a message in the terminal based on the result.

Open Geany and enter the following code:

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.

```
import lgpio
from time import sleep
# Define the GPIO pin connected to the LDR sensor
DO_PIN = 17
# Initialize GPIO chip (the default GPIO chip is "/dev/gpiochip0")
h = lgpio.gpiochip_open(0)
# Set GPIO 17 as an input to read data from the LDR sensor
lgpio.gpio_claim_input(h, DO_PIN)
try:
    while True:
        # Read the current state of the LDR sensor
        light_state = lgpio.gpio_read(h, DO_PIN)
        # Print whether light is detected or not based on the sensor reading
        if light_state == 0:
            print("The light is present!")
        else:
            print("The light is NOT present!")
        # Add a small delay to prevent continuous readings and CPU overloading
        sleep(0.1)
except KeyboardInterrupt:
    # Handle program exit when the user presses Ctrl + C
    print("\nExiting the program.")
    lgpio.gpiochip_close(h)
```

Code Breakdown:

Setup and Imports:

The script begins by importing the lgpio library to interact with the GPIO pins, and sleep from the time module to create short pauses between readings.

Pin Definition and Initialization:

It defines GPIO 17 as the input pin connected to the LDR sensor's digital OUT pin. The GPIO chip is opened, and the pin is claimed for input so the Raspberry Pi can receive data from the sensor.

Reading the Sensor:

In the loop, lgpio.gpio_read() checks the signal from the LDR:

- If the result is 0, light is detected.
- If the result is 1, the sensor is in darkness.

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.

This allows the program to interpret light conditions in real time.

Main Loop:

The script runs continuously, checking the sensor's state and printing a message every 0.1 seconds. This fast refresh rate helps detect changes quickly while avoiding CPU overload.

Interrupt Handling:

If the user stops the program using Ctrl + C, the script prints an exit message and safely closes the GPIO chip connection.

Step 4: Run the Program and Observe Light Detection

With the code ready, it's time to test your setup and see how it responds to different lighting conditions.

To run the script:

- Save the file as ldr_light_check.py
- Press F5 in Geany, or run it in the terminal:

  python ldr_light_check.py

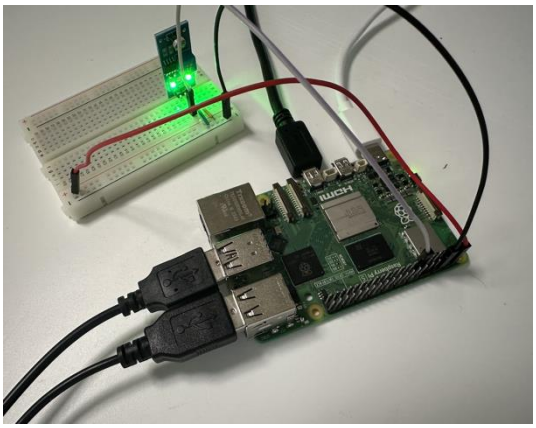


Figure 54- Raspberry Pi connected to an LDR sensor detecting light.

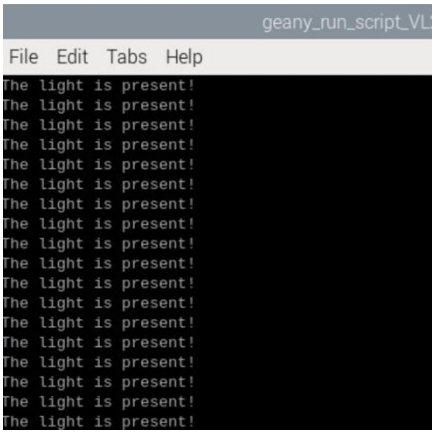


Figure 55- Output showing "The light is present!" when light is detected.

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.

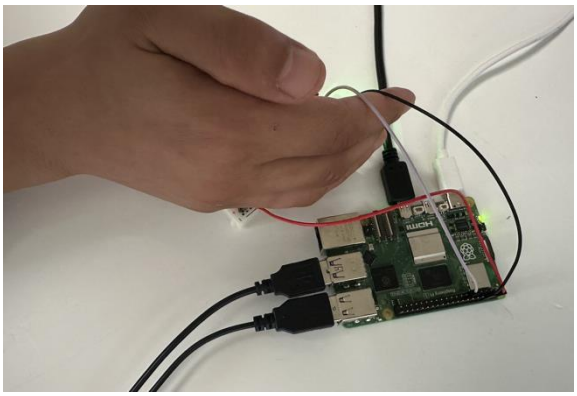


Figure 56- Hand covering the sensor, causing it to detect no light.

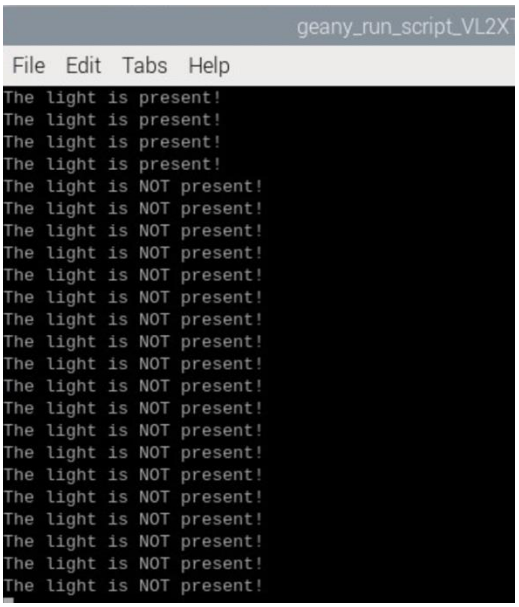


Figure 57- Output showing "The light is NOT present!" when no light is detected

---

Step 5: Steps to Adjust the Potentiometer

The sensor module includes a potentiometer (the blue component with a screw), which is used to adjust the sensitivity or threshold of the sensor.

1. Identify the Adjustment Screw:
   The potentiometer has a small screw or dial on top. This is the adjustment point.

2. Tools Required:
   Use a small screwdriver to make adjustments.

3. Making Adjustments:
   Clockwise Rotation (Right Turn):
   Turning the screw clockwise increases the resistance, which usually increases the threshold or reduces the sensitivity of the sensor.
   Counterclockwise Rotation (Left Turn):

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.

Turning the screw counterclockwise decreases the resistance, which usually decreases the threshold and increases the sensitivity of the sensor.

4. Testing and Calibration:
    - After making adjustments, test the sensor under the conditions it is designed to detect.
    - Fine-tune the screw until the sensor responds accurately to the desired

---

11. Evaluation Criteria:

- **Sensor Connection:** LDR is properly wired and responds to light changes.
- **Digital Logic:** The script detects changes in light and reacts correctly.
- **Code Functionality:** No errors; script runs and exits cleanly.
- **Hands-On Testing:** User tested the sensor under both light and dark conditions.

---

12. Languages Available:

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

---

13. 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.

Fátima Pais, Luís Soares – Project Designers and Facilitators

---

17. 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.