AI & SUSTAINABILITY IN VET EDUCATION

Nome dell'autore/i: Sucessos Criativos, Lda | P. Porto

Titolo del WP: WP3 – AI-SAVES@ProjectHub

Riferimento dell'attività: 3.2 – Modello di progetto per 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.

Indice dei contenuti

1. Titolo del progetto: 2

2. Categoria del progetto: 2

3. Descrizione del progetto: 2

5. Pubblico target: 2

6. Conoscenze preliminari: 2

7. Obiettivi del progetto: 3

8. Tempistica del progetto: 3

9. Risorse fornite: 3

10. Strumenti e tecnologie utilizzati: 4

11. Piano di implementazione passo dopo passo: 6

12. Criteri di valutazione: 13

13. Lingue disponibili: 13

14. Collaboratori: 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. **Titolo del progetto:**

Sistema di tracciamento della luce con servomotore e Raspberry Pi

---

2. **Categoria del progetto:**
   - IA nell'IoT

---

3. **Descrizione del progetto:**

Questo progetto mira a sviluppare un sistema di tracciamento della luce utilizzando un Raspberry Pi, sensori LDR (Light Dependent Resistor) e un servomotore. Il sistema rileva l'intensità della luce da due direzioni e regola automaticamente il servomotore per seguire la fonte luminosa più intensa. Questo progetto dimostra come sensori e attuatori possano essere combinati per creare sistemi IoT reattivi, capaci di interagire con l'ambiente e prendere decisioni in modo automatizzato.

---

4. **Parole chiave:**
   Tracciamento della luce, Raspberry Pi, IoT, Sensori LDR, Servomotore, Automazione

---

5. **Pubblico target:**
   Sia docenti che studenti

---

6. **Conoscenze preliminari:**

- Comprensione di base della legge di Ohm e dei circuiti elettrici
- Introduzione alla programmazione Python (variabili, cicli, funzioni)
- Conoscenza del controllo GPIO del Raspberry Pi
- Comprensione di base dei sensori e dei servomotori

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. **Obiettivi del progetto:**
   - Introdurre gli studenti al rilevamento della luce tramite sensori LDR
   - Insegnare il controllo dei servomotori con Raspberry Pi
   - Dimostrare decisioni in tempo reale basate sui dati dei sensori
   - Sviluppare competenze pratiche in elettronica e programmazione Python
   - Incoraggiare lo sviluppo di sistemi IoT reattivi

---

8. **Tempistica del progetto:**
   Tempistica flessibile in base al calendario della classe o del workshop

---

9. **Risorse fornite:**
   - Raspberry Pi
   - Servomotore (180 gradi)
   - 2 sensori LDR
   - Breadboard e cavi jumper
   - Alimentatore esterno per il servomotore
   - Modelli di codice Python
   - Guida didattica passo-passo

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. Strumenti e tecnologie utilizzati:

10.1 Hardware:

Raspberry Pi, servomotore, 2 sensori LDR, breadboard, cavi jumper

10.2 Software:

- Raspbian OS, Python 3
- Sistema operativo: Raspberry Pi OS
- Ambiente di programmazione: Geany o IDE Python
- Librerie Python: lgpio, time

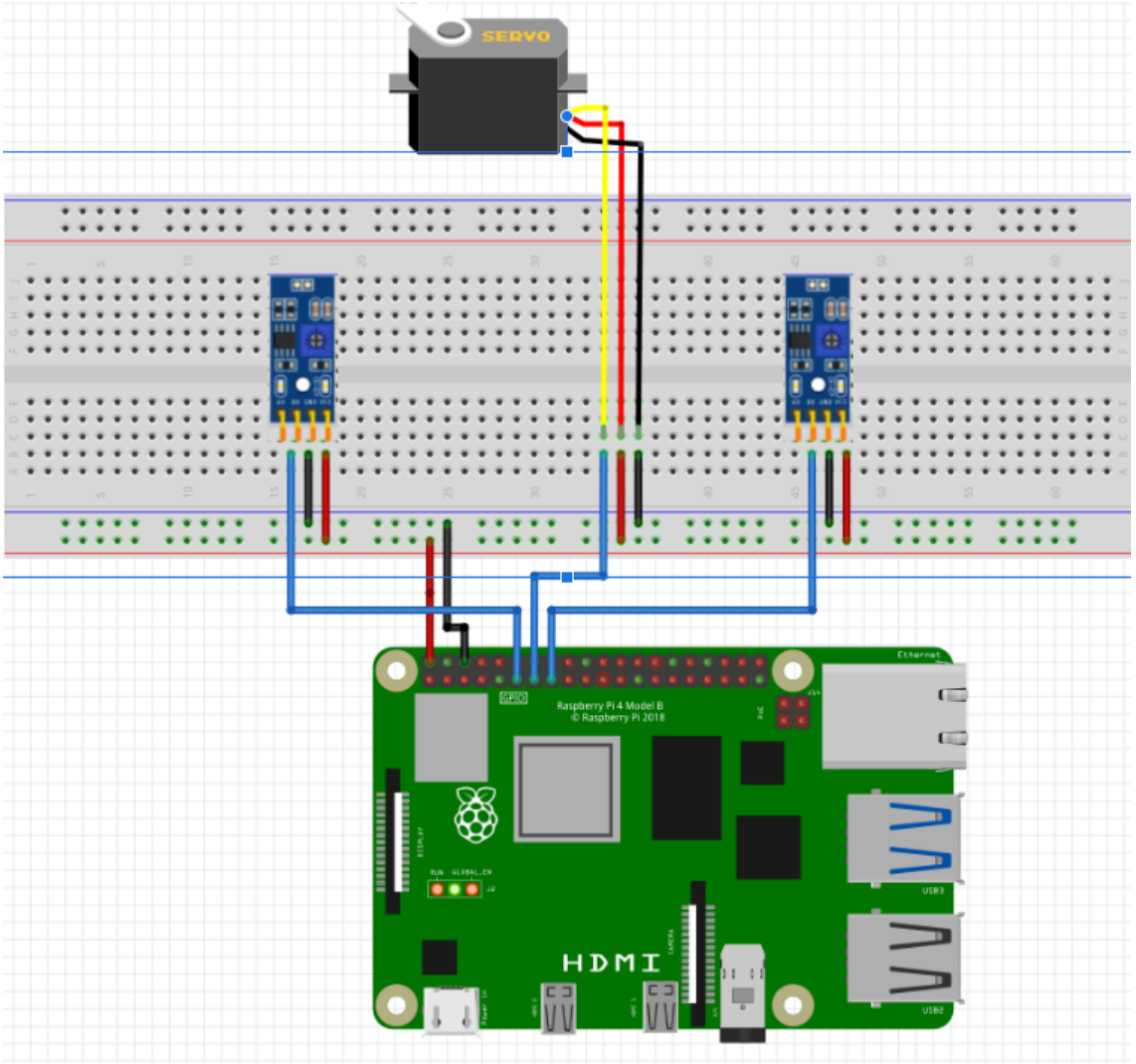
Librerie: Adafruit_DHT, RPi.GPIO

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.

11. Piano di implementazione passo dopo passo:

11.1 Configurazione hardware



Collegamenti del circuito

- Servomotore:
  - Signal ⟶ GPIO 27 (Pin 13)
  - Power ⟶ 5V (Pin 2 or Pin 4) (alimentazione esterna consigliata)
  - Massa ⟶ Massa (Pin 6)
- LDR Sensori:
  - Sinistro LDR ⟶ GPIO 17 (Pin 11)
  - Destro LDR ⟶ GPIO 22 (Pin 15)

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.

- Connessione di massa : Utilizzare una massa comune (Pin 6, 9, 14, or 20) per tutti i componenti.
- Configurazione del circuito LDR: sensori LDR sono collegati direttamente ai pin GPIO:
    - LDR collegato tra il pin GPIO e GND
    - Quando la luce colpisce l'LDR: il GPIO legge LOW (0)
    - Quando non c'è luce: il GPIO legge HIGH (1)

Passaggi di cablaggio:

1. Collegare il filo del segnale del servo a **GPIO 27 (Pin 13)**.
2. Collegare il filo di alimentazione del servo a **5V**.
3. Si consiglia di utilizzare un'**alimentazione esterna a 5V** per garantire prestazioni stabili del servo.
4. Collegare il filo di massa del servo al **GND del Raspberry Pi**.
5. Collegare il sensore LDR sinistro a **GPIO 17 (Pin 11)**.
6. Collegare il sensore LDR destro a **GPIO 22 (Pin 15)**.
7. Collegare entrambi i sensori LDR alla **massa comune (GND)**.
8. 🎬 Assicurarsi che tutti i componenti condividano lo **stesso riferimento di massa (GND)**..

⚠️**Importante:** una volta assemblato il circuito, controlla attentamente tutti i collegamenti prima di procedere. Assicurati che ogni filo sia correttamente collegato al pin giusto. Un cablaggio errato può causare comportamenti instabili, malfunzionamenti dei sensori o danni al Raspberry Pi e al servomotore.

11.2 Configurazione del software

Ora che l'hardware è pronto, è il momento di configurare il codice che farà funzionare il tuo sistema di monitoraggio della luce.

1. Creare un ambiente virtuale

```
# Create project directory
mkdir light_tracker
cd light_tracker

# Create virtual environment
python3 -m venv venv

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

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.

2. Installare le dipendenze

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

3. Aggiungere il codice

- Copia il codice Python in `light_tracker.py`
- Assicurati che il file sia nella directory del progetto

4. Avviare il sistema

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

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

5. **Arrestare il sistema**
   - Premi **Ctrl+C** per fermare il programma in modo sicuro
   - Il servomotore si fermerà e le risorse GPIO verranno rilasciate automaticamente.

Comportamento del sistema

- Posizione centrale (90°): entrambi i sensori rilevano luce o non c'è preferenza direzionale
- Posizione sinistra (45°): solo il sensore LDR sinistro rileva luce
- Posizione destra (135°): solo il sensore LDR destro rileva luce
- Il sistema controlla continuamente i sensori con un ritardo di 1 secondo tra gli aggiornamenti

Il tuo sistema di monitoraggio della luce è ora pronto per essere utilizzato!
CODICE

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 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
    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
```

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.

```
    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...")

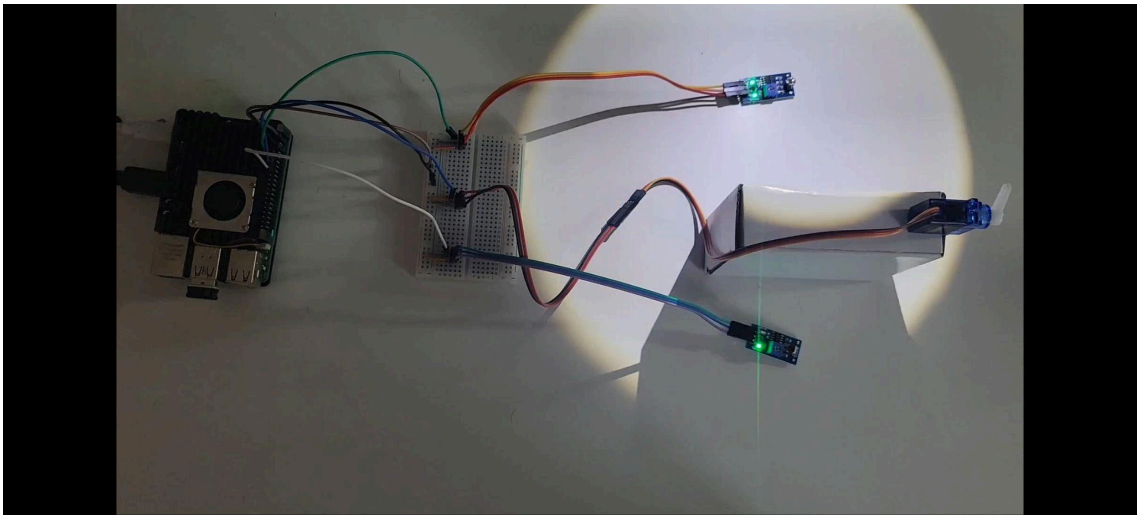
finally:
    # Stop servo signal
    lgpio.tx_servo(h, SERVO_PIN, 0)
    lgpio.gpiochip_close(h)
```

Ora puoi eseguire il programma e vedere il risultato:

Co-funded by the European Union

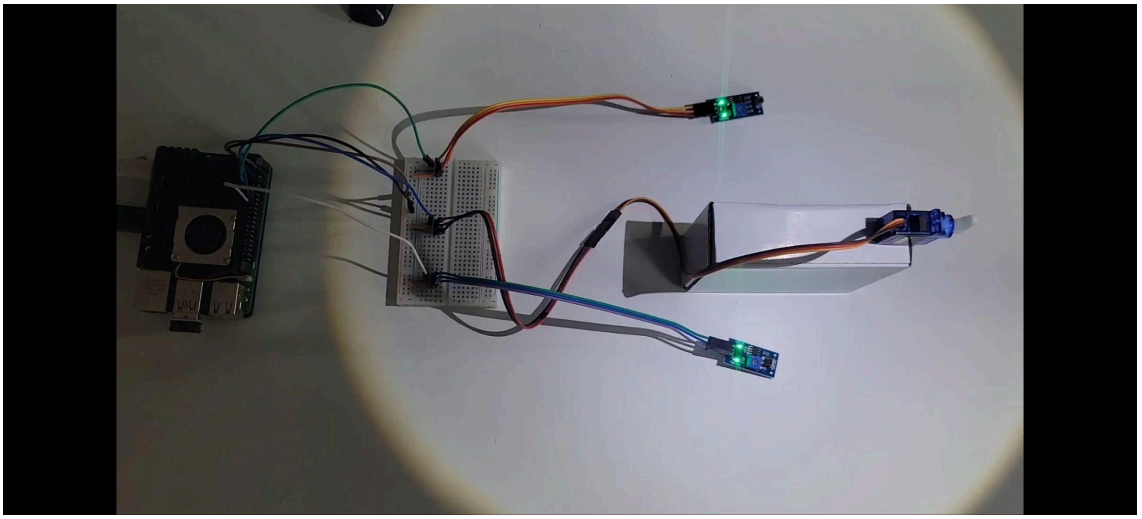
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

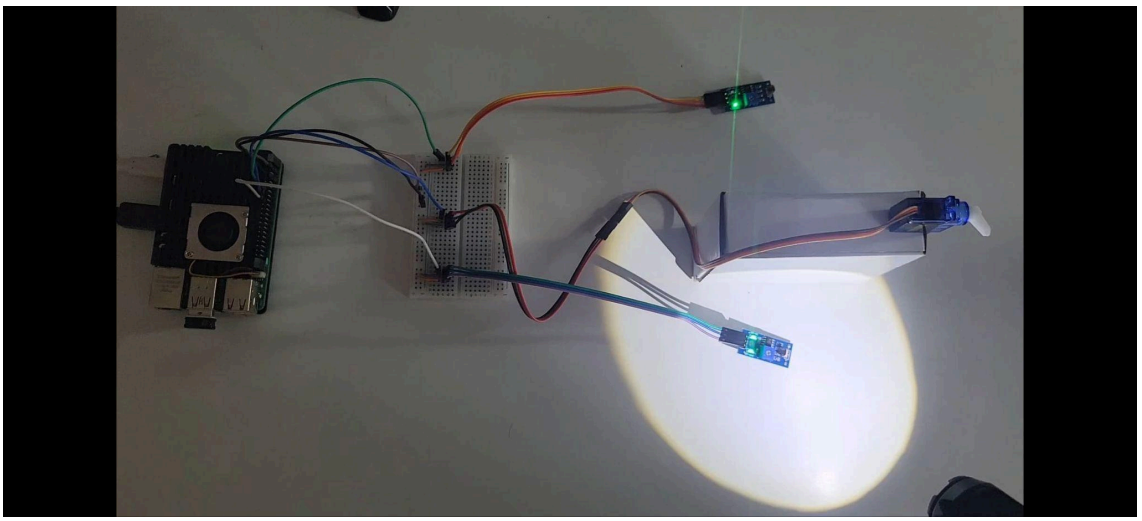
```
File  Edit  Tabs  Help
GPIO cleaned up and resources released.
(env) pi@raspberrypi-2:~ $ python placa.py
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in straight position.
No light detected on either side! Keeping servo in 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. Criteri di valutazione:

- **Configurazione hardware (35%)**: cablaggio corretto dei sensori e del servomotore; circuito stabile e funzionante.
- **Funzionalità del codice (30%)**: il codice funziona correttamente e il servomotore risponde alla rilevazione della luce.
- **Comprensione del sistema (20%)**: spiegazione chiara di come i sensori controllano il posizionamento del servomotore.
- **Innovazione e miglioramenti (15%)**: miglioramenti creativi, come un tracciamento più preciso o sensori aggiuntivi.

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. Lingue disponibili:

---

14. Collaboratori:

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.