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

Logo:

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: 5
10. Step-by-Step Implementation Plan: 6
11. Evaluation Criteria: 13
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:

Autonomous Sustainable Waste Detection Vehicle

---

2. Project Category:

AI in Robotics

---

3. Project Description:

This project presents the development of a very basic autonomous smart vehicle capable of driving continuously using obstacle-avoidance logic based on ultrasonic distance sensing, while applying artificial intelligence object detection to identify waste materials in the environment—specifically cups and plastic bottles. The robot navigates independently by controlling two DC motors, avoiding collisions using a ultrasound sensor.

When the onboard camera detects waste (bottles), the vehicle pauses its movement and triggers a notification by sending a WhatsApp message through an HTTP request, simulating a smart trash-collection alert system. After sending the notification and waiting briefly, the vehicle resumes its autonomous driving routine. The purpose of the system is to demonstrate how AI and robotics can be combined to improve waste monitoring in public spaces, promote sustainability awareness, and encourage hands-on application of AI-driven automation.

---

4. Keywords:

Autonomous Robot, Object Detection, Smart Waste Detection, Raspberry Pi

---

5. Target Audience:

Both Educators and Students

---

6. Project Objectives:

- Build and configure a fully autonomous robotic car using Raspberry Pi.

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.

- Implement ultrasonic-based obstacle avoidance with real-time movement control.
- Apply object detection with YOLO to identify bottles and cups.
- Integrate WhatsApp notifications via HTTP request.
- Understand the connection between AI recognition, robotics movement, and communication systems.
- Explore sustainability-focused applications of AI and automation.

---

7. Project Timeline:

With the provided resources, this project should take approximately 7-8 hours or less to be done calmly.

---

8. Resources Provided:

- Python codebase containing motor control, object detection, distance measurement, and WhatsApp alert logic
- Demo video of the robot detecting waste (bottles) and sending WhatsApp notifications
- Wiring diagram for motors, ultrasonic sensor, and camera

---

9. Tools and Technologies Used:

- Raspberry Pi 5
- Raspberry Pi Camera Module 4 v2
- Python programming language
- OpenCV for camera input and image processing
- Pre-trained Object Detection Model (YOLO11-Nano)
- Ultrasonic distance sensor
- Dual DC motor driver (L298N) + Wheels, etc.
- CallMeBot API for WhatsApp notifications via HTTP request
- (Optionally) Power bank to replace Raspberry Pi's power supply

Co-funded by the European Union

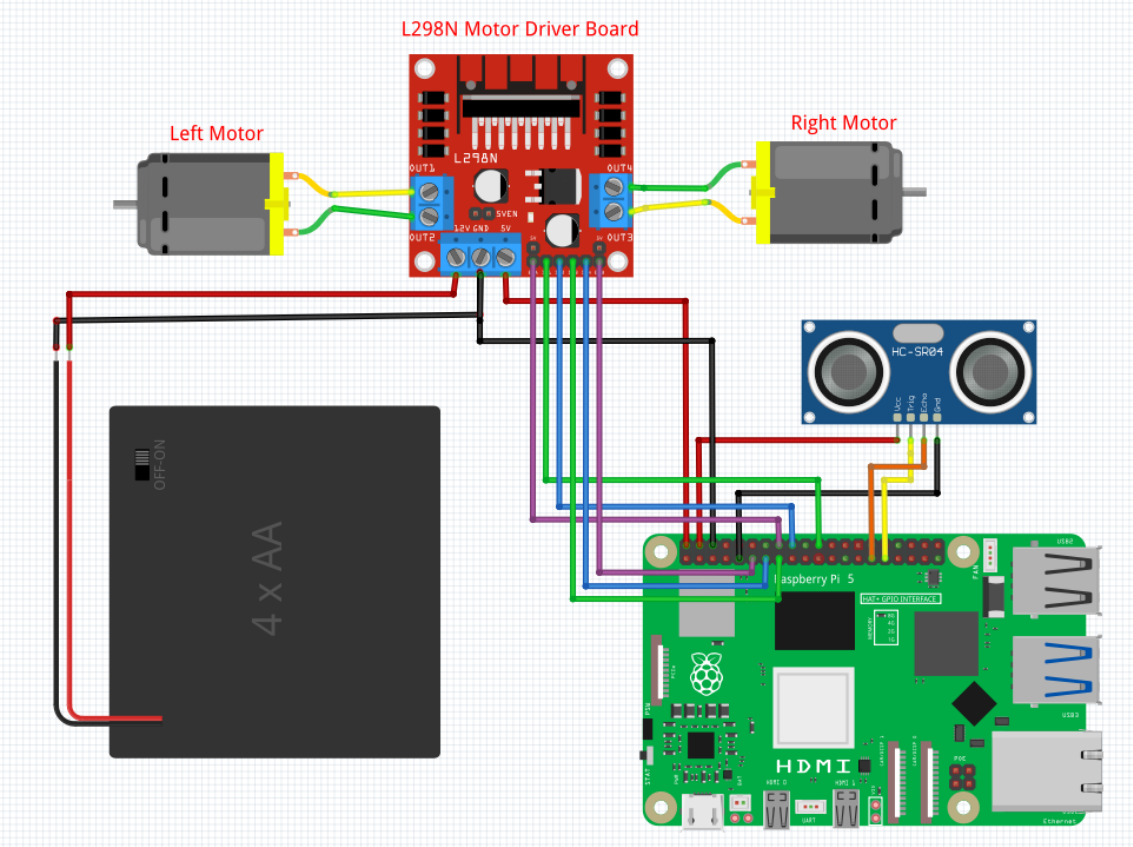
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

10. Step-by-Step Implementation Plan:

Provide a detailed step-by-step guide on how the project will be implemented. This will help users who wish to replicate the project or follow the process.

1. **Hardware Preparation:** Mount the motors on the car chassis and connect them to the motor driver. Connect Raspberry Pi, the camera module, motor driver and the ultrasonic sensor according to the wiring diagram. When connecting the motor driver, don't forget to remove all jumpers of the motor driver board.

   Optional: When testing, you can power Raspberry Pi with a PowerBank.



2. **Software Setup:** Install the necessary software on Raspberry Pi, including Raspbian OS and Python libraries including OpenCV, Picamera2, Ultralytics YOLO, Requests API, and GPIO libraries.

3. **CallMeBot API Setup:** Follow the setup guide of CallMeBot (https://www.callmebot.com/blog/free-api-whatsapp-messages/) and save your Api Key.

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.

4. **Execute the Code:** Copy the provided code bellow and save it on your Raspberry Pi. Before running the code, replace the value of “APIKEY” to your Api Key from step 3 and replace the value of “PHONE_NUMBER” to your phone number registered on step 3 (this value should contain your country code).

```
import time

import requests

import RPi.GPIO as GPIO

import cv2

from picamera2 import Picamera2

from ultralytics import YOLO



# ===== CONFIGURATION =====

APIKEY = "1111111"  # Replace with your actual API key

PHONE_NUMBER = "+1123456789"  # Your real phone number with country code (example for portugal: "+351911234567")



# -------------------- SENSOR PINS --------------------

TRIG = 6

ECHO = 5



GPIO.setmode(GPIO.BCM)

GPIO.setup(TRIG, GPIO.OUT)

GPIO.setup(ECHO, GPIO.IN)



# -------------------- MOTOR PINS --------------------

IN1 = 25

IN2 = 24

IN3 = 22

IN4 = 27
```

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.

```
ENA = 23

ENB = 17


GPIO.setup(IN1, GPIO.OUT)

GPIO.setup(IN2, GPIO.OUT)

GPIO.setup(IN3, GPIO.OUT)

GPIO.setup(IN4, GPIO.OUT)

GPIO.setup(ENA, GPIO.OUT)

GPIO.setup(ENB, GPIO.OUT)


pwmA = GPIO.PWM(ENA, 100)

pwmB = GPIO.PWM(ENB, 100)

pwmA.start(0)

pwmB.start(0)


# -------------------- MOTOR FUNCTIONS --------------------

def move_forward():

    GPIO.output(IN1, GPIO.HIGH)

    GPIO.output(IN2, GPIO.LOW)

    GPIO.output(IN3, GPIO.HIGH)

    GPIO.output(IN4, GPIO.LOW)

    pwmA.ChangeDutyCycle(60)

    pwmB.ChangeDutyCycle(60)

    print("Moving forward")


def stop_motors():

    GPIO.output(IN1, GPIO.LOW)

    GPIO.output(IN2, GPIO.LOW)
```

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.

```
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.LOW)
    pwmA.ChangeDutyCycle(0)
    pwmB.ChangeDutyCycle(0)
    print("Motors stopped")


def turn_right():
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    GPIO.output(IN3, GPIO.LOW)
    GPIO.output(IN4, GPIO.HIGH)
    pwmA.ChangeDutyCycle(60)
    pwmB.ChangeDutyCycle(60)
    print("Turning right")
    time.sleep(1.5)
    stop_motors()


# -------------------- DISTANCE SENSOR --------------------
def measure_distance():
    GPIO.output(TRIG, False)
    time.sleep(0.0002)
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)


    while GPIO.input(ECHO) == 0:
        pulse_start = time.time()
    while GPIO.input(ECHO) == 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.

```
        pulse_end = time.time()


    return ((pulse_end - pulse_start) * 34300) / 2


# -------------------- CAMERA + YOLO --------------------
picam2 = Picamera2()
picam2.preview_configuration.main.size = (640, 400)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.align()
picam2.configure("preview")
picam2.start()


model = YOLO("yolo11n.pt")


# cooldown tracking
last_detected = {"object": None, "time": 0}
DETECTION_COOLDOWN = 30  # seconds


# -------------------- SEND WHATSAPP NOTIFICATION --------------------
def send_notification(object_name):
    try:
        url                                                              =
f"https://api.callmebot.com/whatsapp.php?phone={PHONE_NUMBER}&text=Object+F
ound:+{object_name}&apikey={APIKEY}"
        requests.get(url, timeout=5)
        print(f"Notification sent: {object_name}")
    except Exception as e:
        print("Failed to send notification:", e)
```

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.

```
# -------------------- MAIN LOOP --------------------

try:

    move_forward()


    while True:

        # 1 – distance check

        distance = measure_distance()

        if distance < 50:

            stop_motors()

            print("Obstacle detected")

            turn_right()

            move_forward()

            continue


        # 2 – object detection

        frame = picam2.capture_array()

        results = model(frame)


        for result in results:

            for box in result.boxes:

                label = result.names[int(box.cls[0])].lower()


                if label in ("bottle", "cup"):

                    obj = label.upper()

                    now = time.time()


                    cooldown_active = (

                        last_detected["object"] == obj and
```

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 - last_detected["time"]) < DETECTION_COOLDOWN
                    )

                    if not cooldown_active:
                        print(f"Found: {obj}")
                        last_detected["object"] = obj
                        last_detected["time"] = now

                        stop_motors()
                        send_notification(obj)
                        print("Waiting 5 seconds...")
                        time.sleep(5)

                        turn_right()
                        move_forward()

                    break

except KeyboardInterrupt:
    print("Program terminated")

finally:
    stop_motors()
    GPIO.cleanup()
    cv2.destroyAllWindows()
    picam2.stop()
```

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. Evaluation Criteria:

Explain how the success of the project will be measured. This could include qualitative feedback from participants, quantitative performance indicators, or specific project outcomes.

- **System Functionality:** Ability to replicate the project, including Basic Autonomous Driving, Cup/Bottle Detection and WhatsApp Notification.
- **Creativity:** Suggestions for improvements (new features, new objects detected, etc.). If suggested and tried (with success or failure) should be more rewarded.
- **Optimizations:** Voluntary optimizations for the code with success or failure, especially on Autonomous Driving and Object Detection.

---

12. Languages Available:

English

---

13. Contributors:

- **Fátima Pais, Luís Soares** – Project Designers and Facilitators

---

14. Feedback and Continuous Improvement:

Describe how feedback will be collected from participants and how this feedback will be used to improve future projects or iterations of this project.

- **Post-Implementation Survey:** After deploying the project, feedback will be collected through surveys targeting participants involved in the testing and observation process.
- **Team Reflection Sessions:** The project team will organize internal review sessions to discuss challenges faced, what worked well, and areas that need refinement.
- **Technical Evaluation:** Observations regarding detection accuracy, system stability, and hardware durability will be documented for improving future versions.

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.

- **Iterative Updates:** Based on feedback and performance, updates to the software code and system design will be implemented to improve reliability and effectiveness.

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.