AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Kaunas University of Technology, Lina Narbutaitė
Title of WP: WP3 – AI-SAVES@ProjectHub
Activity Reference: Project for AI-SAVES@ProjectHub

Logo: ktu kaunas university of technology 1922

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

AI-SAVES@PROJECTHUB PROJECT GREENBIN..... 4

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:..... 4
8. Resources Provided:..... 4
9. Tools and Technologies Used:..... 5
10. Step-by-Step Implementation Plan:..... 5
11. Evaluation Criteria:..... 9
12. Languages Available:..... 9
13. Contributors:..... 9
17. Feedback and Continuous Improvement:..... 10

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.

AI-SAVES@PROJECTHUB PROJECT GREENBIN

1. Project Title:

GreenBin – Intelligent Bin Fill Level Notifier.

---

2. Project Category:

- AI in Robotics
- **AI in IoT**
- AI for Education
- AI in VET
- AI Tools and Resources

---

3. Project Description:

This project aims to promote sustainability through the development of a smart waste bin using AI and sensor technology. GreenBin detects when the bin is full and provides visual alerts via LED and display, helping improve waste management efficiency and environmental awareness

.

---

4. Keywords:

Sensors, Smart Waste Management, Sustainability

---

5. Target Audience:

Identify the primary audience for this project. Choose one or more of the following:

- Educators
- Students
- **Both Educators and Students**
- VET Institutions

---

6. Project Objectives:

- Develop practical skills in AI and IoT sensor integration.
- Understand how AI can be applied to real-world environmental challenges.
- Gain hands-on experience with electronics, coding, and automation.
- Promote sustainable thinking through technology-driven waste management.

---

7. Project Timeline:

- Start Date: November 10, 2024
- End Date: January 10, 2025

---

8. Resources Provided:

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.

- Raspberry Pi and sensor kits (AI-SAVES kit)
- Python program and its explanation
- Brief project description (PDF))
- Lesson plan: "Introduction to AI KIT" (PDF)
- Project report template (DOCX)

---

9. Tools and Technologies Used:

- Raspberry Pi 4
- Sensor kits
- Python programming language
- Libraries :
    - import lgpio
    - import time
    - import board
    - import digitalio
    - from PIL import Image, ImageDraw, ImageFont
    - import adafruit_ssd1306
- GitHub for version control and project collaboration

---

10. Step-by-Step Implementation Plan:

1. Set up the Raspberry Pi: Configure the Raspberry Pi with the necessary software (Raspbian OS, Python).
2. Install Libraries: Install the required libraries, including sensor management (e.g., lgpio), display (e.g., adafruit_ssd1306), and potentially AI libraries (e.g., TensorFlow Lite if waste classification will be implemented).
3. Introduce Basic Concepts: Explain the fundamental concepts related to distance sensors, servo control, and (potentially) AI waste classification, and their application in smart waste bins.
4. Conduct Practical Sessions: Conduct practical sessions where you integrate the sensors, program the core logic (fullness detection), control the LED, servo motor, and integrate AI (if used).
5. Monitor Project Progress: Monitor the project's progress through weekly reviews.
6. Evaluate Understanding: Assess student understanding through project demonstrations where they showcase the functionality of their smart waste bin.

The program code

After making the circuit you are ready to start coding!

```
source env/bin/activate
```

With the terminal open in the environment we are working, you should now install adafruit_ssd1306 library. This library specifically provides functions and tools to

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.

interact with SSD1306 OLED displays via I2C or SPI communication. To install run the following command:

```
pip3 install adafruit-circuitpython-ssd1306
```

Open Geany and write the following code. Check the comments to fully understand each line:

```
import lgpio
import time
import board
import digitalio
from PIL import Image, ImageDraw, ImageFont
import adafruit_ssd1306

TRIGGER_PIN = 17
ECHO_PIN = 27
LED_PIN = 16
SERVO_PIN = 22
BUTTON_PIN = 23

h = lgpio.gpiochip_open(0)

# Output
lgpio.gpio_claim_output(h, TRIGGER_PIN)
lgpio.gpio_claim_output(h, LED_PIN)
lgpio.gpio_claim_output(h, SERVO_PIN)

# Input
lgpio.gpio_claim_input(h, ECHO_PIN)
lgpio.gpio_claim_input(h, BUTTON_PIN)


threshold = 5  # cm



# Initialize OLED display dimensions (128x64 pixels)
WIDTH = 128
HEIGHT = 64
# Set up I2C communication with the OLED display
# Uses the default I2C pins on the board (SCL and SDA)
i2c = board.I2C() # Initializes I2C communication
# Create an SSD1306 OLED display instance with the specified width,
height,and I2C interface
oled = adafruit_ssd1306.SSD1306_I2C(WIDTH, HEIGHT, i2c, addr=0x3C)
# Clear the OLED display (fills the screen with black)

def getfontsize(font, text):
# This calculates the bounding box of the text, returning its width and
height
    left, top, right, bottom = font.getbbox(text)
    return right - left, bottom - top


def update_display(text):
    oled.fill(0)
    oled.show()
# Create a new image object with "1-bit" color (black and white) for
drawing on the display
```

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.

```
    image = Image.new("1", (oled.width, oled.height))
# Obtain a drawing object from the image to draw shapes and text
    draw = ImageDraw.Draw(image)
# Load the default font to use for displaying text on the OLED
    font = ImageFont.load_default()

# Define a helper function to calculate the pixel size of the text
(width and height)

# Define the text to display on the OLED screen
    # text = "FULL   bin"
# Get the width and height of the text in pixels to help with centering
it on the display
    (font_width, font_height) = getfontsize(font, text)
# Draw only the text on the image, centered horizontally and vertically
on the OLED display
    draw.text(
            (oled.width // 2 - font_width // 2, oled.height // 2 -
font_height // 2),
# Center the text
        text, # Text to display
        font=font, # Use the default font
        fill=255 # White color for the text
    )
    oled.image(image)
    oled.show()
    time.sleep(1)

def set_servo_pulsewidth(h, pin, pulse_width_ms):
    lgpio.gpio_write(h, pin, 1)
    time.sleep(pulse_width_ms / 1000.0)
    lgpio.gpio_write(h, pin, 0)

def move_servo(angle):
    pulse_width_ms = 1 + (angle / 180.0)  # 1ms?2ms
    set_servo_pulsewidth(h, SERVO_PIN, pulse_width_ms)

def measure_distance():
    lgpio.gpio_write(h, TRIGGER_PIN, 1)
    time.sleep(0.00001)
    lgpio.gpio_write(h, TRIGGER_PIN, 0)

    start_time = time.time()
    while lgpio.gpio_read(h, ECHO_PIN) == 0:
        start_time = time.time()

    end_time = time.time()
    while lgpio.gpio_read(h, ECHO_PIN) == 1:
        end_time = time.time()

    pulse_duration = end_time - start_time
    distance = (pulse_duration * 34300) / 2
    return distance

try:
    while True:
        #  Servo and button
        if lgpio.gpio_read(h, BUTTON_PIN) == 0:
            move_servo(180)
```

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.

```
            time.sleep(0.5)
        else:
            move_servo(0)
            print("Open ")
            update_display("Open")
            time.sleep(1)
            oled.fill(0)
            oled.show()

        # Distance
        distance_cm = measure_distance()
        print(f"Distance: {distance_cm:.2f} cm")

        if distance_cm < threshold:
            print("FULL bin")
            update_display("FULL   bin")
            time.sleep(1)
            oled.fill(0)
            oled.show()
            lgpio.gpio_write(h, LED_PIN, 1)
        else:
            lgpio.gpio_write(h, LED_PIN, 0)

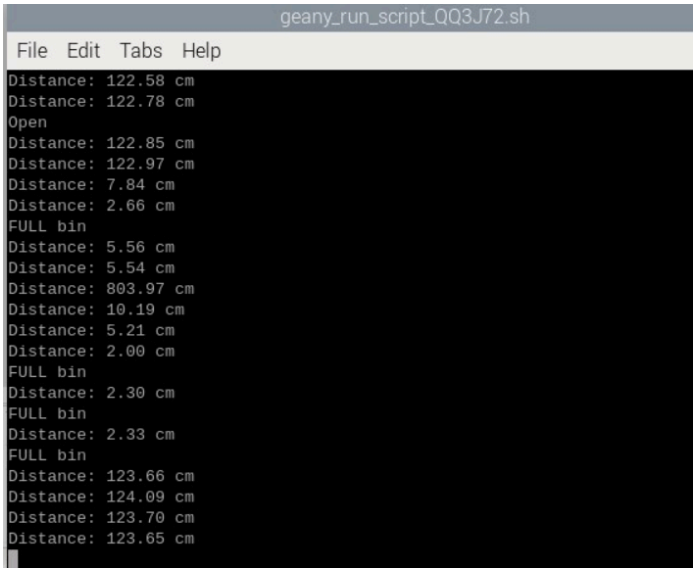
        time.sleep(0.3)

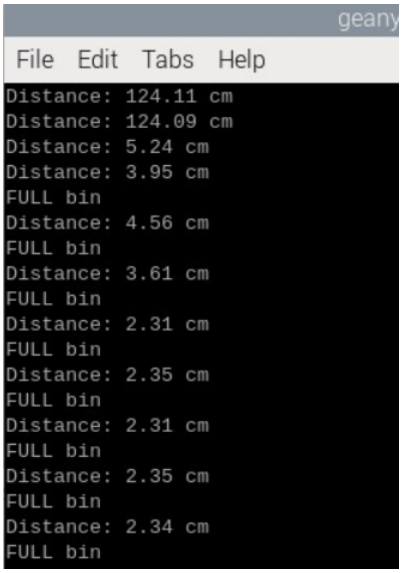
except KeyboardInterrupt:
    lgpio.gpio_write(h, LED_PIN, 0)
    lgpio.gpio_write(h, SERVO_PIN, 0)
    lgpio.gpiochip_close(h)
    print("GPIO clear")
```

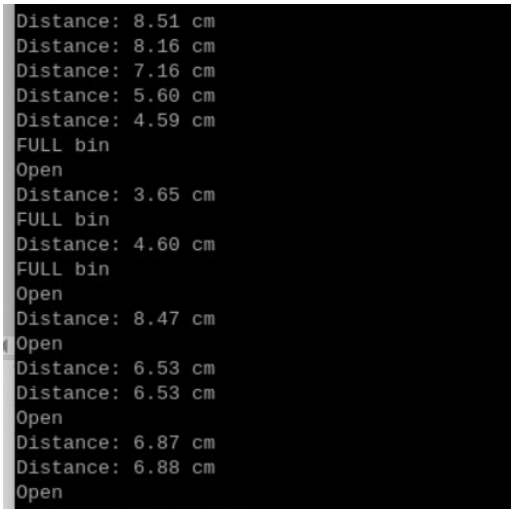
Save the program.
Now, you can run the program and see the result:

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.

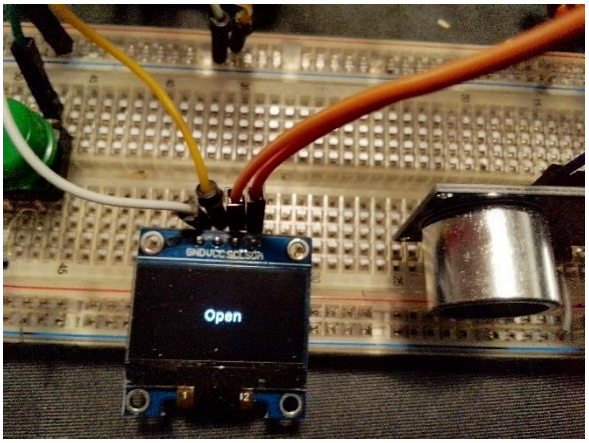






Funded by the European Union

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:

- **Student Feedback**: Gather feedback on the learning process through discussion.
- **Project Success Metrics:**
  - Successful and accurate fullness detection by the waste bin.
  - Reliable visual alerts via LED and display.
  - Proper servo motor functionality (if implemented for lid control).
  - (If applicable) Accurate waste classification by the AI model.
- **Skills Assessment:**
  - Measure student proficiency in:
  - Sensor integration and data interpretation.
  - Python programming for GPIO control and logic implementation.
  - (If applicable) AI model training and integration for waste classification.

12. Languages Available:

English, Lithuanian

13. Contributors:

Lina Narbutaitė

17. Feedback and Continuous Improvement:

- **Discussions**: At the end of each 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**: The feedback collected from surveys and project reviews will be analyzed to identify strengths and areas for improvement. This data-driven approach will inform future project planning, curriculum development, and resource allocation, ensuring that future GreenBin or similar projects are more effective and engaging.

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.

Sustainability Justification

GreenBin supports sustainability by improving waste management efficiency and reducing environmental impact through smart monitoring technology. The system uses AI and sensors to detect waste levels in real time, helping prevent overflowing bins and unnecessary collection trips. This leads to reduced fuel consumption, lower carbon emissions, and more efficient use of waste collection resources.

By providing visual alerts through LEDs and displays, GreenBin encourages timely waste disposal and promotes environmental awareness among users. The project also supports cleaner public spaces and healthier communities by reducing litter accumulation and improving sanitation.

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.