AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Politeknika Txorierri

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: Parking management system

Logo:

POLITEKNIKA TXORIERRI

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

Contenido

AI-SAVES@PROJECTHUB PROJECT SUBMISSION TEMPLATE 4

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: *Energy efficiency through IOT: Smart parking*

2. Project Category: IOT

3. Project Description:

This project aims to design and implement a smart parking system that automatically monitors vehicle access, detects parking space occupancy, and provides real-time information about availability through an IoT infrastructure. The system improves traffic management and user experience by offering up-to-date data on parking capacity while reducing congestion and unnecessary vehicle circulation.

The system uses multiple ESP32 microcontrollers equipped with distance sensors to monitor the occupancy status of individual parking spaces. Each ESP32 is responsible for a single parking spot and continuously reports whether the space is occupied or free based on distance measurements. In addition, a separate ESP32 placed at the parking entrance monitors vehicle entry and exit, allowing the system to maintain an accurate count of vehicles currently inside the parking area. All status updates are transmitted via MQTT to the central controller.

To provide a global and coherent view of the parking area, a Raspberry Pi acts as the main controller and MQTT broker. It aggregates data received from all ESP32 nodes, calculates the number of occupied and available spaces, and determines whether parking capacity has been reached. This information is then published to a visualization dashboard using Node-RED, enabling users and administrators to quickly assess parking availability. If the parking area becomes full, the system can trigger alerts or visual indicators to inform drivers and prevent further access.

From a sustainability perspective, the smart parking system helps reduce fuel consumption and emissions by minimizing the time vehicles spend searching for available spaces. This contributes to more efficient urban mobility and supports smart-city principles focused on resource optimization and environmental responsibility.

The system working architecture is the following:

1. Detect vehicle entry and exit at the parking access point.
2. Monitor the occupancy status of each parking space.
3. Update the total number of vehicles inside the parking area.

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. Aggregate parking space data in the central controller.
5. Calculate available and occupied spaces.
6. Publish real-time parking information to the Node-RED dashboard.

4. Keywords: IOT, Vocational Education, Automation, Sustainability, Enviromental care, Hands-on Learning

5. Target Audience: Both educators and Students

6. Project Objectives:

- Develop practical knowledge in IOT.
- Understand how to design and implement an smart watering system.
- Gain hands-on experience in programming microcomputers and microcontrollers.
- Understand system-level design: connecting sensors, controlling actuators, and managing data flow.
- Improve teamwork, documentation, and presentation skills.
- Explore sustainability through energy-efficient design choices

7. Project Timeline:

Task	Estimated Hours
Raspberry and ESP32 hardware and programming Node-RED installation and configuration	13 hours
Practice and validation	4 hours
Documentation and presentation preparation	3 hours
Total	**20 hours**

8. Resources Provided:

- Raspberry Pi, ESP32 and sensor kits (AI-SAVES Kit)

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.

- Programming guides
- Needed software and hardware

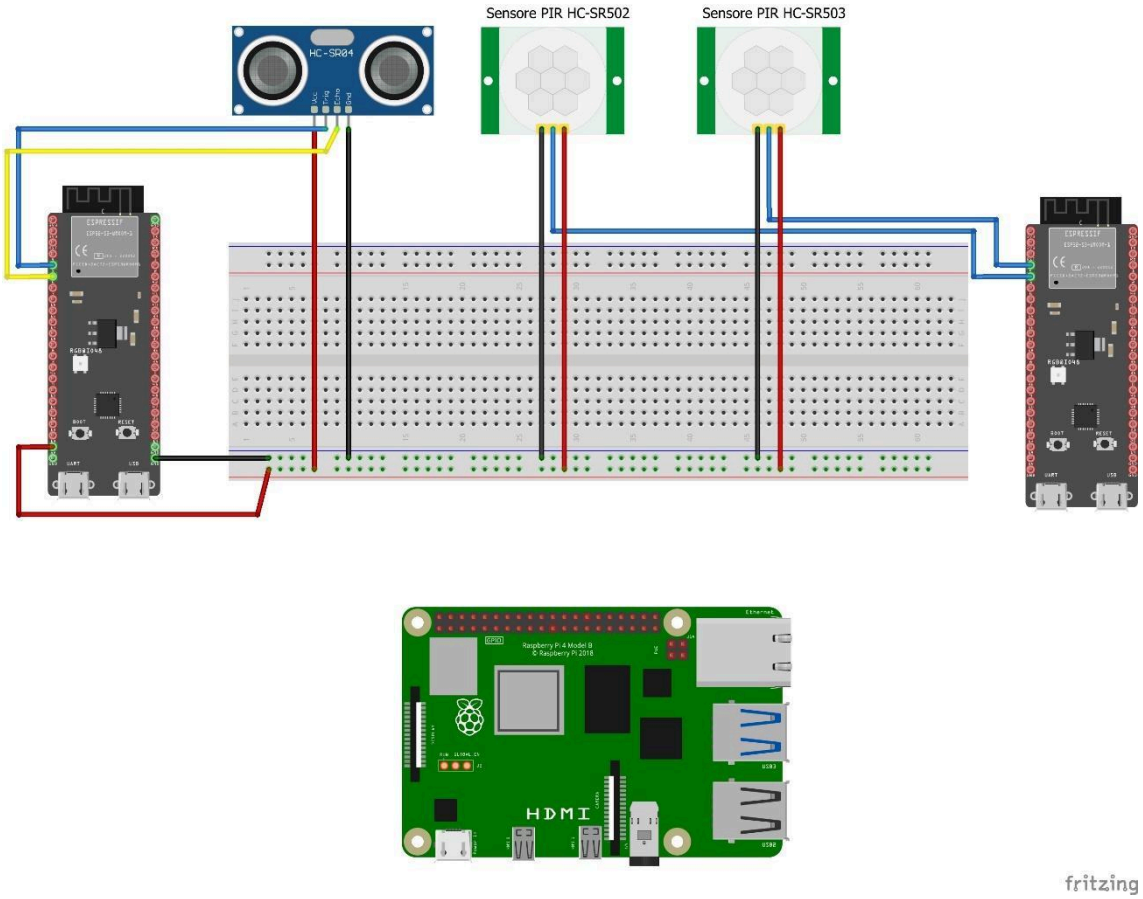
9. Tools and Technologies Used:

- Raspberry PI 4 and ESP32
- Hardware: ultrasonic sensor, PIR sensor,
- Python and C/C++ programming languages, with Geay and IDE programming tools.
- Node-RED application and MQTT protocol
- GitHub for version control and project collaboration
- Fritzing for circuit design and testing

10. Step-by-Step Implementation Plan:

1. Set up the Raspberry Pi with the necessary software (Raspbian OS, Python).
2. Conduct practical sessions where students build and program the IOT driven domotic system.

a. Assemble circuit according to the ESP32 pin mapping.



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.

b. Load the needed code (you can use the two provided .ino files to work locally) in the ESP32 and check locally it works correctly; the object distance measurement (to decide if there is a car in a spot) and presence detection (to decide if a car is entering or leaving the parking) must be checked.

c. Activate Python virtual environment in the Raspberry Pi:

cd project_folder_path

python -m venv .venv

source .venv/bin/activate

d. Install required libraries:

pip install paho-mqtt

e. Run the needed/provided script using Geany

python parking-control.py

f. Load the needed/provided file in the ESP32 using IDE

in-out.ino and *occupied.ino*

g. Be sure Raspberry PI and ESP32 are connected to the same wifi network SSID

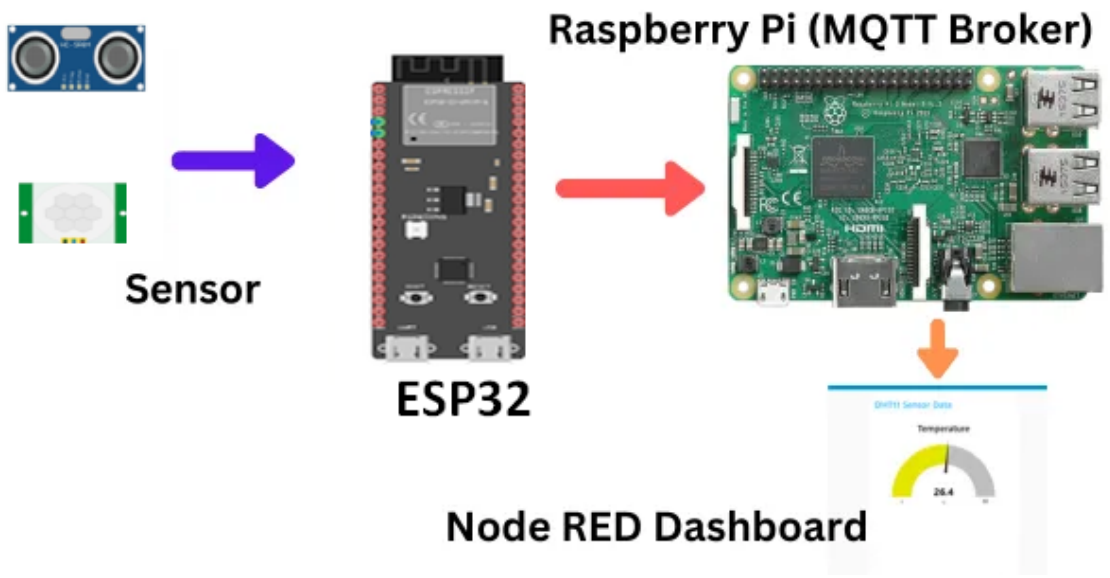
h. Install Nodered in the Raspberry and import the provided JSON file to create the dashboard.

dashboard.json

i. Use the different inputs/outputs to check the system is running properly and check data are correctly shown in the Node-RED dahsboard. This is the program flow

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.



3. Evaluate student understanding through project demonstrations.

11. Evaluation Criteria:

- **Hardware implementation (25%):** Properly connected circuit, no loose wires, correct sensor placement.
- **Code functionality (30%):** Script runs without errors, correctly detects water level and activates actuators, correctly detects animals (optional), including MQTT funcionality.
- **Nodered functionality (15%):** The dashboard shows provided data from Raspberri.
- **Understanding & Explanation (20%):** Ability to explain how the system works and its real-world applications.
- **Creativity & Optimization (10%):** Suggestions for improvements, such as adding additional automation features.

12. Languages Available: English

13. Contributors:

- **Jokin Goioaga (Project Lead)** – Lead educator responsible for project development and coordination.
- **Aitor Rey (Technical Expert)** – Provided expertise in IOT.

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.

14. Feedback and Continuous Improvement:

- **Surveys**: 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**: 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.