AI & SUSTAINABILITY
IN VET EDUCATION

AI-VET MOOC PROJECT SUBMISSION TEMPLATE

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 & SUSTAINABILITY IN VET EDUCATION
ERASMUS 2023-1-LT01-KA220-VET-000155506

1. Project Title:

"Regulation of Crossroad traffic"

2. Project Category:

- AI for Sustainable Development
- Connected Systems and Smart Data (AI + IoT)
- AI for Industry 4.0
- Social Impact and Inclusive AI
- Computer Vision and Image Recognition
- Predictive Analytics and Decision Support
- **Intelligent Automation and Control Systems**
- Smart Environments and Monitoring
- Other:_____

3. Project Description:

This project represents a traffic management system for an intersection that can automatically regulate the movement of vehicles and pedestrians. At the intersection, four traffic lights are installed to control vehicle movement and four traffic lights to control pedestrian movement.

A 7-segment display with a common cathode is also connected, which counts down the remaining time for the red and green lights. LED lights in three colors are used, which alternately switch in the sequence red–yellow–green–yellow–red. They are programmatically controlled to turn on and off in the following order: green for 10 seconds, yellow for 2 seconds, red for 16 seconds, yellow for 2 seconds, and so on.

The pedestrian traffic lights operate in the following sequence: red for 16 seconds and green for 10 seconds.

4. Keywords:

traffic light, intersection, traffic, regulation, 7-segment display

5. Target Audience:

- Students

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.

6. Project Objectives:

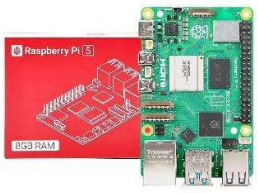
The aim of this project is to develop a smart traffic management system for an intersection that will automatically regulate the movement of vehicles and pedestrians. During the development of the project, basic knowledge is acquired in working with Raspberry Pi, programming in Python, and using various libraries. Additionally, it helps in understanding how different electronic components are connected and integrated, as well as how they function together within a single system. Through this exercise, the following skills are learned: Installing an operating system for Raspberry Pi Hardware connection of components Understanding the basic integration of hardware and software (Python, libraries) Demonstrating hardware-software interaction using Python

7. Project Timeline:

- Start Date: 15.3.2026
- End Date: 1.4.2026

8. Tools and Technologies Used:

1. Components of the AI-VET kit:

	Компонента	picture	Функција
1.	Raspberry Pi 5 – 8GB		Used as the central control unit (“brain”) of the system. After receiving results from the camera, it processes them and sends them to the display for visualization.

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.	Raspberry Pi metal case with fan		Provides protection and cooling for the device. The metal case protects the Raspberry Pi from dust, mechanical damage, and accidental contact, while the fan helps reduce the temperature during operation.
3.	LED lights – red, green, yellow		Used to build traffic lights for vehicles and pedestrians.
4.	7-segment display (common cathode)		Displays the remaining time for the red and green lights.
5.	Resistors – 330 Ω		Their role is to protect the LEDs from burning out.
6.	Breadboard and jumper wires		Jumper wires are used to connect components on the breadboard and to the microcontroller (Raspberry Pi).
7.	Micro HDMI to HDMI cable (1 m)		Used to connect the system to a display, transmitting video and audio signals.

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.

8.	27W USB Type-C power supply for Raspberry Pi 5		Power supply cable.
9.	Micro SDHC card – 64GB		Used for storing the operating system and data required for system operation.

2. AI-VET Assembly Manual with step-by-step instructions
3. Python code and libraries
4. Datasheet for the 7-segment display
5.

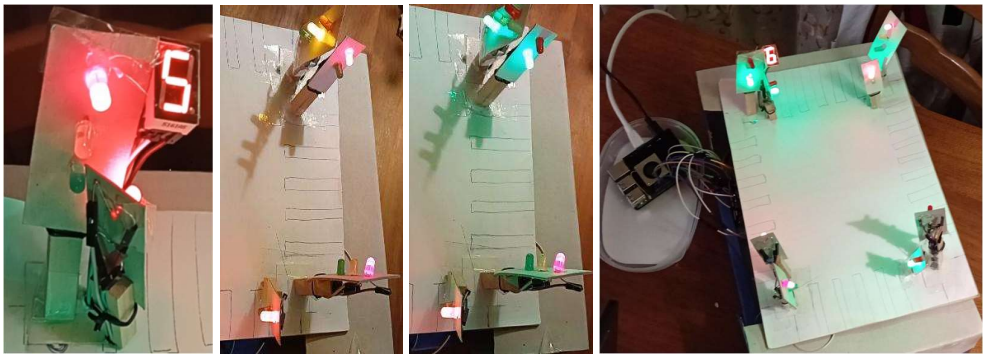
9. Step-by-Step Implementation Plan:

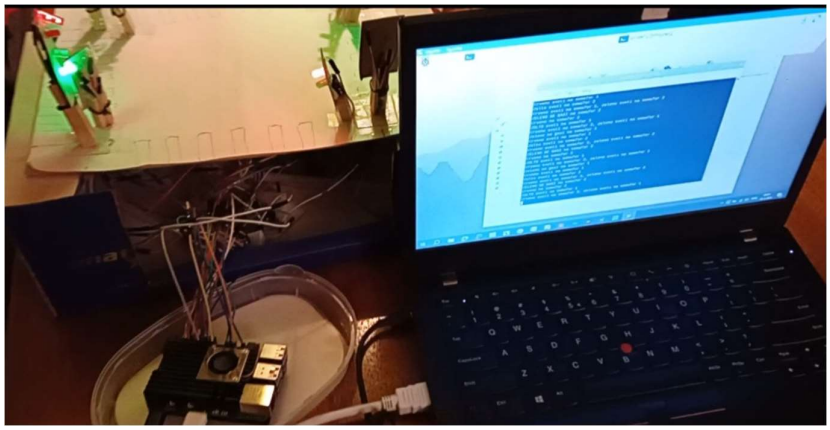
1. **Project plan / idea:** A modern intersection that will regulate traffic. We start by working on a single traffic light and then gradually add all traffic lights for vehicle movement, followed by the four traffic lights for pedestrian movement. Finally, we add a 7-segment display that shows the remaining time for the red and green lights.
2. **Preparation of hardware components:** Gather all necessary materials – Raspberry Pi 5, LEDs, resistors, 7-segment display, wires, breadboard, and power cables, followed by their connection.
3. **Assembly of Raspberry Pi 5 and case with cooling fan:** Assemble the Raspberry Pi 5 in a case with a temperature-controlled cooling fan.
4. **Installing Raspbian OS:** Install Raspbian OS on the SD card, configure network access via VNC or SSH, and update the system packages.
5. **Integration of hardware components:** Connect the LEDs, resistors, and the 7-segment display on the breadboard and link them with wires to the Raspberry Pi
6. **Programming in Python and using libraries**

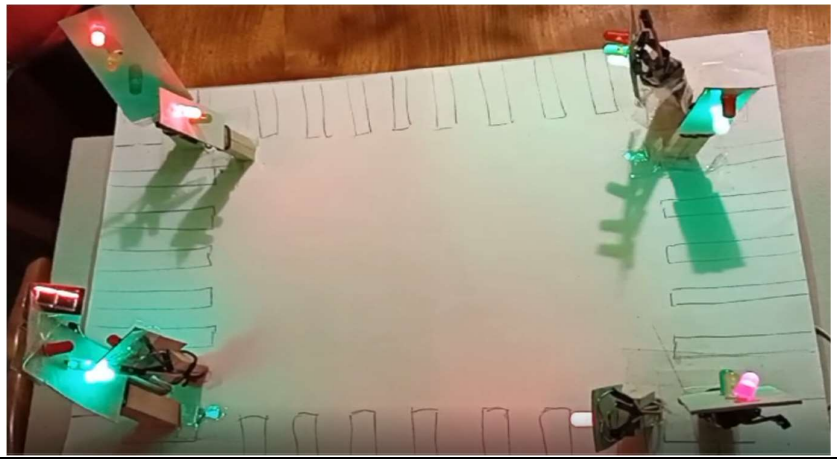
Funded by the European Union

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. Testing the project functionality







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- Code

```
from gpiozero import LED
import lgpio
import time

RED1 = 4
YELLOW1 = 3
GREEN1 = 23
YELLOW2 = 15
GREEN2 = 2
RED2 = 18
PR1 = 24
PG1= 25
PR2= 17
PG2= 14
D1=27
D2=10
D3=22
D4=9
D5=11
D6=5
D7=6
D8=13

segments = {
    'a': LED(22),
    'b': LED(27),
    'c': LED(5),
    'd': LED(6),
    'e': LED(13),
    'f': LED(10),
    'g': LED(9)
}

numbers = {
    0:['a','b','c','d','e','f'],
    1:['b','c'],
    2:['a','b','g','e','d'],
    3:['a','b','g','c','d'],
    4:['f','g','b','c'],
    5:['a','f','g','c','d'],
    6:['a','f','g','c','d','e'],
    7:['a','b','c'],
    8:['a','b','c','d','e','f','g'],
    9:['a','b','c','d','f','g']  }
```

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.

```
h = lgpio.gpiochip_open(0)
lgpio.gpio_claim_output(h, RED1)
lgpio.gpio_claim_output(h, YELLOW1)
lgpio.gpio_claim_output(h, GREEN1)
lgpio.gpio_claim_output(h, RED2)
lgpio.gpio_claim_output(h, GREEN2)
lgpio.gpio_claim_output(h, YELLOW2)
lgpio.gpio_claim_output(h, PR1)
lgpio.gpio_claim_output(h, PG1)
lgpio.gpio_claim_output(h, PR2)
lgpio.gpio_claim_output(h, PG2)

def display_number(num):
    for seg in segments.values():
        seg.off()

    for seg in numbers[num]:
        segments[seg].on()

try:
    while True:
        print("The red is on at traffic light 1, and the green is on at traffic light 2")
        lgpio.gpio_write(h, RED2, 0)
        lgpio.gpio_write(h, GREEN1, 0)
        lgpio.gpio_write(h, YELLOW1, 0)
        lgpio.gpio_write(h, YELLOW2, 0)
        lgpio.gpio_write(h, RED1, 1)
        lgpio.gpio_write(h, GREEN2, 1)
        lgpio.gpio_write(h, PG1, 0)
        lgpio.gpio_write(h, PR1, 1)
        lgpio.gpio_write(h, PG2, 1)
        lgpio.gpio_write(h, PR2, 0)

        for i in range(9, -1, -1):
            display_number(i)
            time.sleep(1)

        print("The green light is turning off at traffic light 2")
        lgpio.gpio_write(h, RED1, 1)
        lgpio.gpio_write(h, GREEN1, 0)
        lgpio.gpio_write(h, YELLOW2, 1)
        lgpio.gpio_write(h, YELLOW1, 0)
```

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.

```
        lgpio.gpio_write(h, GREEN2, 0)
        lgpio.gpio_write(h, PG1, 0)
        lgpio.gpio_write(h, PR1, 1)
        lgpio.gpio_write(h, PG2, 0)
        lgpio.gpio_write(h, PR2, 1)

        for seg in segments.values():
            seg.off()

        time.sleep(2)

        print("The red light is on at traffic light 2")
        lgpio.gpio_write(h, RED2, 1)
        lgpio.gpio_write(h, GREEN1, 0)
        lgpio.gpio_write(h, YELLOW1, 0)
        lgpio.gpio_write(h, YELLOW2, 0)
        lgpio.gpio_write(h, RED1, 1)
        lgpio.gpio_write(h, GREEN2, 0)
        lgpio.gpio_write(h, PG1, 0)
        lgpio.gpio_write(h, PR1, 1)
        lgpio.gpio_write(h, PG2, 0)
        lgpio.gpio_write(h, PR2, 1)
        time.sleep(2)

        print("The yellow light is on at traffic light 1")
        lgpio.gpio_write(h, RED2, 1)
        lgpio.gpio_write(h, GREEN1, 0)
        lgpio.gpio_write(h, YELLOW1, 1)
        lgpio.gpio_write(h, YELLOW2, 0)
        lgpio.gpio_write(h, RED1, 1)
        lgpio.gpio_write(h, GREEN2, 0)
        lgpio.gpio_write(h, PG1, 0)
        lgpio.gpio_write(h, PR1, 1)
        lgpio.gpio_write(h, PG2, 0)
        lgpio.gpio_write(h, PR2, 1)
        time.sleep(2)

        print("The red is on at traffic light 2, and the green is on at traffic light 1")
        lgpio.gpio_write(h, RED2, 1)
        lgpio.gpio_write(h, GREEN1, 1)
        lgpio.gpio_write(h, YELLOW1, 0)
        lgpio.gpio_write(h, YELLOW2, 0)
```

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.

```
    lgpio.gpio_write(h, RED1, 0)
    lgpio.gpio_write(h, GREEN2, 0)
    lgpio.gpio_write(h, PG1, 1)
    lgpio.gpio_write(h, PR1, 0)
    lgpio.gpio_write(h, PG2, 0)
    lgpio.gpio_write(h, PR2, 1)

    for i in range(9, -1, -1):
        display_number(i)
        time.sleep(1)

    print("The green light is turning off at traffic light 1")
    lgpio.gpio_write(h, RED2, 1)
    lgpio.gpio_write(h, GREEN1, 0)
    lgpio.gpio_write(h, YELLOW1, 1)
    lgpio.gpio_write(h, YELLOW2, 0)
    lgpio.gpio_write(h, RED1, 0)
    lgpio.gpio_write(h, GREEN2, 0)
    lgpio.gpio_write(h, PG1, 0)
    lgpio.gpio_write(h, PR1, 1)
    lgpio.gpio_write(h, PG2, 0)
    lgpio.gpio_write(h, PR2, 1)

    for seg in segments.values():
        seg.off()

    time.sleep(2)

    print("The red light is on at traffic light 1")
    lgpio.gpio_write(h, RED2, 1)
    lgpio.gpio_write(h, GREEN1, 0)
    lgpio.gpio_write(h, YELLOW1, 0)
    lgpio.gpio_write(h, YELLOW2, 0)
    lgpio.gpio_write(h, RED1, 1)
    lgpio.gpio_write(h, GREEN2, 0)
    lgpio.gpio_write(h, PG1, 0)
    lgpio.gpio_write(h, PR1, 1)
    lgpio.gpio_write(h, PG2, 0)
    lgpio.gpio_write(h, PR2, 1)
    time.sleep(2)

```

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.

```
            print("The yellow light is on at traffic light 2")
            lgpio.gpio_write(h, RED2, 1)
            lgpio.gpio_write(h, GREEN1, 0)
            lgpio.gpio_write(h, YELLOW1, 0)
            lgpio.gpio_write(h, YELLOW2, 1)
            lgpio.gpio_write(h, RED1, 1)
            lgpio.gpio_write(h, GREEN2, 0)
            lgpio.gpio_write(h, PG1, 0)
            lgpio.gpio_write(h, PR1, 1)
            lgpio.gpio_write(h, PG2, 0)
            lgpio.gpio_write(h, PR2, 1)
            time.sleep(2)

    except KeyboardInterrupt:
        print("Stopping LED blinking...")
        lgpio.gpio_write(h, RED2, 0)
        lgpio.gpio_write(h, GREEN1, 0)
        lgpio.gpio_write(h, YELLOW1, 0)
        lgpio.gpio_write(h, RED1, 0)
        lgpio.gpio_write(h, GREEN2, 0)
        lgpio.gpio_write(h, YELLOW2, 0)
        lgpio.gpio_write(h, PG1, 0)
        lgpio.gpio_write(h, PR1, 0)
        lgpio.gpio_write(h, PG2, 0)
        lgpio.gpio_write(h, PR2, 0)
        lgpio.gpiochip_close(h)
        for seg in segments.values():
            seg.off()

```

11. Evaluation Criteria:

- **Student Feedback:** Feedback will be collected from students through questionnaires to assess how well they understood the process of programming the traffic light, the operating logic, and the connection of hardware components (LED diodes, Raspberry Pi/Arduino). It will also be analyzed how interesting and useful they found the learning process.
- **Project Success Metrics:** Success will be measured based on the correct and stable functioning of the traffic light system, including accurate switching of lights (red, yellow, green) according to the set time intervals and logic. Additionally, it

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.

will be checked whether the system works without errors during long-term testing and whether all pins and LED diodes are activated correctly.

- **Skills Assessment:** The level of students' knowledge and skills in programming (Python or C++) will be evaluated, as well as their ability to connect and control electronic components. Special emphasis will be placed on understanding the logic of automated systems, timing functions, and working with GPIO pins.

The success of the project **"Regulation of Crossroad Traffic"** will be evaluated through a combination of qualitative feedback, measurable indicators, and achieved outcomes.

12-Credits

Ana Nikoloska, III-4 – Worked independently on the project, including the idea development, research, assembly, installation, hardware design, connections, and software programming. **Professor Gordana Naumovska** – Provided guidance and support throughout the development of the project.

13- Video presentation link (max. 5 minutes)

https://drive.google.com/file/d/13fSXODzDGP7Hw23YBkBZC-H5RMIYOgQc/view?usp=drive_link

14- Declarations

I declare that I have read and agree with the following statements:

1- The project is original and developed by the listed team members.
2- All technologies and sources are properly cited.
3- The project was developed using components from the AI-VET kit.
4- We authorize the dissemination of the project for educational and promotional purposes (non-exclusive rights).

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.