AI & SUSTAINABILITY
IN VET EDUCATION

Team Name: Aerix

Institution: SETUGS “Mihajlo Pupin”

Country: North Macedonia

Mentor/ Teacher: Tamara Borovska

Team members: Martin Damkovski, Nemanja Rajkovikj and Leonid Simonovski

AI-VET MOOC - AI-VET MOOC - TEMPLATE FOR SUBMISSION OF THE PROJEC

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

1. Project Title:

Artificial Intelligence for the Smart Environment and Monitoring Industry

2. Project Category

- Artificial Intelligence for Industry 4.0
- Smart Environments and Monitoring

3. Project Description:

The goal of the AI project is to develop an intelligent system that, through the analysis of the frequency of sound produced by bees, will allow monitoring their behavior and condition in the hive. With the help of sensors and artificial intelligence, the system will recognize various states such as calmness or aggressiveness of bees, the occurrence of conflict, opening of the hive, as well as potential attacks by wasps or other threats.

We gained knowledge on how to use audio sensors (microphones) and microcontrollers to collect data from beehives, as well as how to connect and configure them. The project enables an understanding of frequency analysis and the recognition of different sound patterns in bees, related to their behavior.

We learned how to apply basic AI algorithms for data classification based on sound. A functional system will be created that detects changes and sends notifications. Through the analysis of real data, we learned how to draw conclusions and solve problems. We gained programming skills in Raspberry Pi. The final result will be a prototype of the BeeAI system that can be tested in real conditions and provide useful information about the state of the bees.

Funded by the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

4. Hardware Components

Components of the AI-VET kit:

- ESP32-S3 , Rasberry Pi 5 (8G) ,
- Audio Sensor , Light Sensor ,
- OLED Display ,
- 64G Memory Card ,
- Rasberry Pi Cooler ,
- HDMI Cable ,
- Jumper Wires ,
- Breadboard

5. Software apps and Online Services

- Arduino IDE ,
- Raspberry Pi Imager,
- Thonny IDE,
- Raspberry Pi OS

IT and Cloud Services:

MQTT protocol for fast and efficient communication between devices and sending data in real time. Additionally, Msquitt is used as an MQTT broker to receive, manage, and redirect data to the rest of the system.

Additional tools (optional):
GitHub as a repository for storing code

Funded by the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

6. Step-by-step implementation plan:

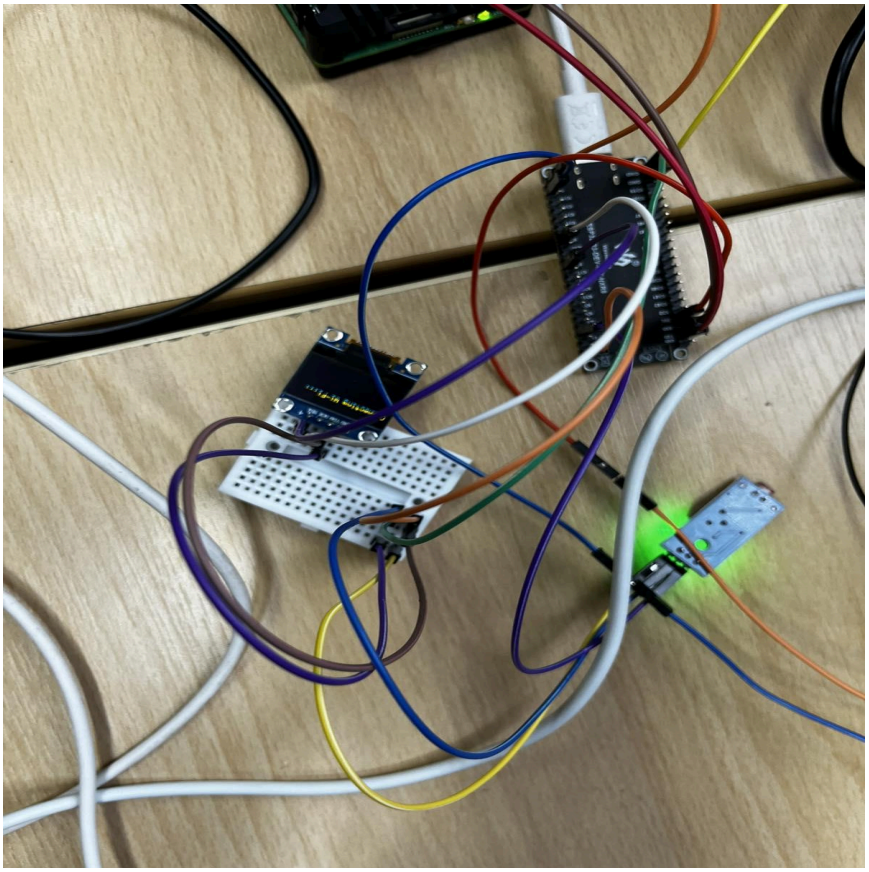
ESP32 grounding is tied to ground on the Rasberry Pi, TXD and RXD pins tied to GPIO 15 (TX) and 14 (RX). The light sensor is powered by the 5V from ESP32, and the ground is also obtained from the ESP32, the D0 pin of the light sensor is tied to the GPIO 4 of the ESP32. The power supply of the OLED display is tied to the 5V of the ESP32, and the ground is tied to the ground of the ESP32, the SDA pin of the display is tied to the GPIO 8 of the ESP32 and the SCK pin of the display of the GPIO 9 on the ESP32.
The microphone power supply is connected to 5V on the ESP32, the microphone ground is also on the ground on the ESP32 and the INPUT pin of the microphone is connected to the GPIO 1 of the ESP32

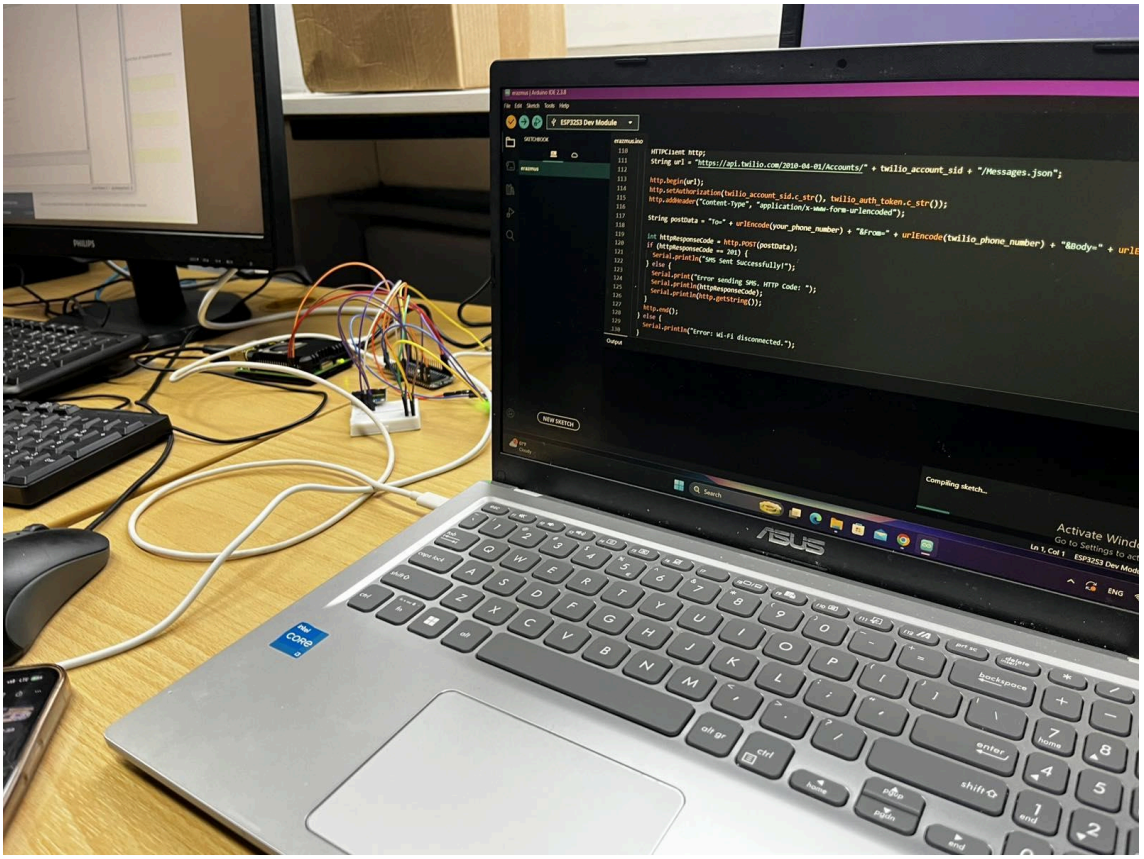
Concept & Planning

Apply protection against threats faced by beekeeping, such as wasp attacks. This design solves that problem by having smart monitoring that monitors specific frequencies that correspond to the frequencies that insects kill when buzzing, such as wasps and hornets, and gives a signal in the form of a message that there is some danger to the boxes where the bees are kept.
The success of a project is measured through practical results and the functioning of the system. We focused on whether the device was properly collecting data from the sensors and whether that data was being successfully transmitted through MQTT and Msquitt. It also monitors whether the system is running smoothly without interruptions, as well as whether the data is displayed accurately and in real time. In addition, checks are being made to see if we can detect changes in bee behavior based on the information obtained. In the end, the project is considered successful because the system works, goals are achieved, and practical results are obtained.

Funded by
the European Union

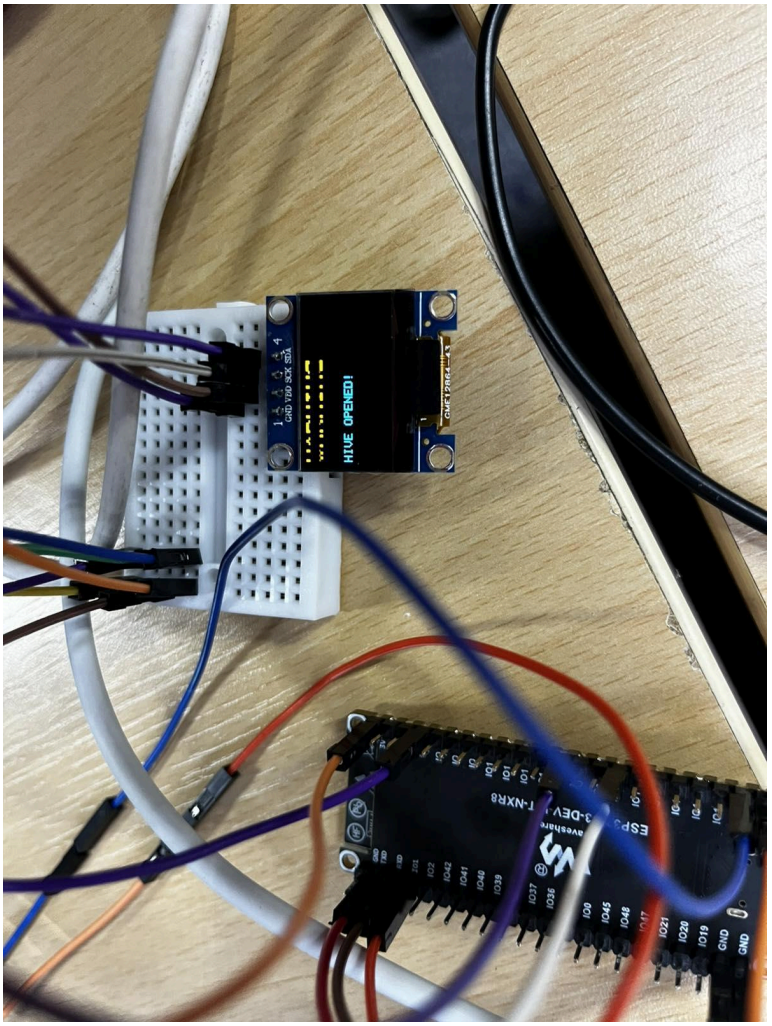
Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.





Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.



Testing & Validation
Testing and validation of the project itself has been carried out..

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

7 - Code

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <UrlEncode.h>
 // PINS
#define LIGHT_SENSOR_PIN 4
#define MIC_PIN 1
#define OLED_SDA 8
#define OLED_SCK 9
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
const char* ssid = "Leo's cookie";
const char* password = "leokolach2";
String twilio_account_sid = "AC18be67845d93797fb723258fe0b8642f";
String twilio_auth_token = "a69eb1976146882e9fdf84602c265505";
String twilio_phone_number = "+17276064890";
String your_phone_number = "+38972256804";
const int sampleWindow = 200; // HIGTHENED to 200ms for precise measurements of the
herz
int dynamicCenter = 2048;
bool isHiveOpen = false;
bool smsSentOpen = false;
bool smsSentAttack = false;
void setup() {
  Serial.begin(115200);
  pinMode(LIGHT_SENSOR_PIN, INPUT);

  Wire.begin(OLED_SDA, OLED_SCK);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("OLED greska!"));
for(;;);
}
```

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

```
  display.setCursor(0, 10);
  display.println("Startuvanje...");
  display.display();
  WiFi.begin(ssid, password);
  int wifiAttempts = 0;
  Serial.print("Konektiranje Wi-Fi"); //"WIFI connection
  while (WiFi.status() != WL_CONNECTED && wifiAttempts < 10) {
    delay(500);
    Serial.print(".");
    wifiAttempts++;
  }
  Serial.println();

  display.clearDisplay();
  display.setCursor(0, 0);
  if(WiFi.status() == WL_CONNECTED) {
    display.println("Wi-Fi OK! Spremen.");
  } else {
    display.println("NEMA Wi-Fi! Raboti lokalno."); // WIFI off, it works remotely
  }
  display.display();
  delay(2000);
}
void loop() {
  // 1. light check up if there is bees inside
  int lightLevel = digitalRead(LIGHT_SENSOR_PIN);
  if (lightLevel == HIGH) {
    isHiveOpen = true;
    if (!smsSentOpen) {
sendSMS("ALARM: Sandakot e OTVOREN!"); // "Alarm - The basket of bees is open"
      smsSentOpen = true;
    }
  } else {
    isHiveOpen = false;
```

Funded by the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

```
// 2. Microphone read
unsigned long startMillis = millis();
  int signalMax = 0;
int signalMin = 4095;
  int crossings = 0;
  bool isAbove = false;
  // Initial status
  int firstVal = analogRead(MIC_PIN);
  if (firstVal > dynamicCenter) isAbove = true;
  while (millis() - startMillis < sampleWindow) {
    int audioVal = analogRead(MIC_PIN);
if (audioVal > signalMax) signalMax = audioVal;
    if (audioVal < signalMin) signalMin = audioVal;
// We filter the sound (herz) to jump the fors 30 units and it eliminates the forst 10 -40 herz
od false breathing or moving sounds. We are looking for The //humming sound
  if (audioVal > dynamicCenter + 30) {
      if (!isAbove) {
        crossings++;
        isAbove = true;
      }
    } else if (audioVal < dynamicCenter - 30) {
      if (isAbove) {
        crossings++;
        isAbove = false;
      }
    }
  }
// 3. Sound calculations
  int amplitude = signalMax - signalMin;
  float frequency = (crossings / 2.0) * (1000.0 / sampleWindow);
if (amplitude > 80) {
    dynamicCenter = (signalMax + signalMin) / 2;
  }
// 4. Printing on the monitor
Serial.print("Amp: ");
```

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

```
Serial.println(" Hz");
  display.clearDisplay();
  display.setCursor(0, 0);
  display.setTextSize(1);
//Testing and sending sms to alert
if (isHiveOpen) {
    display.setTextSize(2);
    display.println("ALARM!");
    display.setTextSize(1);
    display.println("Sandakot e OTVOREN!"); //"the basket is open"
  }
  else if (amplitude > 150) {
    if (frequency >= 250 && frequency <= 600) {
      display.setTextSize(2);
      display.println("OPASNOST!"); //"DANGER"
      display.setTextSize(1);
      display.print("Freq: "); display.print(frequency); display.println(" Hz");

      if (!smsSentAttack) {
        sendSMS("ALARM: Detektirana e visoka frekvencija (Osi/Napad)!"); //"Detection of high
frequency"
        smsSentAttack = true;
      }
    }
    else if (frequency >= 100 && frequency < 250) {
      display.setTextSize(2);
      display.println("NORMALNI"); //NORMAL
      display.setTextSize(1);
      display.print("Freq: ");
display.print(frequency);
 display.println(" Hz");
      smsSentAttack = false;
    }
    else {
      display.println("Nizok sum/zvuk..."); //LOW SOUND
```

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

```
}
  else {
    display.println("Tivko e...");
  }

  display.display();
}

void sendSMS(String message) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

//Sending sms
void sendSMS(String message) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
String url = "https://api.twilio.com/2010-04-01/Accounts/" + twilio_account_sid +
"/Messages.json";

    http.begin(url);
    http.setAuthorization(twilio_account_sid.c_str(), twilio_auth_token.c_str());
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
 String postData = "To=" + urlEncode(your_phone_number) + "&From=" +
urlEncode(twilio_phone_number) + "&Body=" + urlEncode(message);

    int httpResponseCode = http.POST(postData);
    if (httpResponseCode == 201) {
      Serial.println("SMS ispraten uspesno!");
    } else {
      Serial.print("Greska pri prakanje SMS. Kod: ");
      Serial.println(httpResponseCode);
    }
    }
    http.end();
  } else {
```

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.

8. Link of the video presentation

https://drive.google.com/file/d/15vC-k6c0ur4CINduqXnEwdSmhw3XX2RO/view?usp=sharing

Funded by
the European Union

Овој проект е финансиран со поддршка од Европската комисија. Оваа комуникација ги одразува само ставовите на авторот, а Комисијата не може да се смета за одговорна за каква било употреба што може да се направи на информациите содржани во неа.