MCIT Modeling, control & information technologies

Remote Control System for a VTOL UAV Stand with Computer Vision Elements

https://doi.org/10.31713/MCIT.2025.115

Oleh Tsukanov
National University of Water and Environmental Engineering, Rivne, Ukraine
tsukanov_ak23@nuwm.edu.ua

Dmytro Reut
National University of Water and Environmental Engineering, Rivne, Ukraine
d.t.reut@nuwm.edu.ua

Abstract — **This work presents a remote control system for a vertical take-off and landing unmanned aerial vehicle (VTOL UAV) stand, implemented using the ESP32 microcontroller. The system provides control of actuators and aviation lights through a local web interface via a Wi-Fi connection. Additionally, a visualization subsystem was developed, including a camera with remote positioning capabilities and a human silhouette recognition algorithm using computer vision tools. The processed video stream is displayed on the web interface, with detected objects automatically highlighted by graphical bounding boxes. The proposed solution integrates hardware and software components, offering intuitive control and basic computer vision functions to enhance system functionality.**

Keywords — **UAV; remote control system; computer vision; web interface.**

INTRODUCTION

Modern unmanned aerial vehicles (UAVs) are increasingly used in various fields. This creates a need for solutions that allow the study of control algorithms, data processing, and integration with user interfaces.

Today, special attention is given to computer vision systems for real-time object detection and tracking. Combining a camera with remote positioning capabilities and human silhouette recognition enables enhanced monitoring, particularly through web interfaces.

PROBLEM SOLUTION

This work presents the development of a UAV stand based on the ESP32 microcontroller and an ARM Linux computer, which performs human silhouette detection on video frames and streams the results to the user in an intuitive and convenient format.

The stand [1] was additionally equipped with a microprocessor board based on the ESP32 microcontroller, which implements a local HTTP server. The user connects to the local server via a Wi-Fi access point created by the built-in Wi-Fi module of the ESP32. After connecting to the Wi-Fi access point, the user can either open a webpage in a browser using the IP address of the Wi-Fi module or scan a QR code to access the local HTTP server link. This opens a graphical web interface, from which the user can control the stand. To simplify user access to the UAV control system, a Captive Portal mechanism was implemented. After connecting to the Wi-Fi access point created by the ESP32, the user is automatically redirected to the local web interface without the need to manually enter the IP address. This approach improves usability, as the operator gains immediate access to the control interface.

The web interface was implemented as a web page using HTML and CSS. Within this interface, the user can control the ailerons through an interactive joystick (Fig. 1), which simulates the operation of an aircraft control yoke. Additionally, the interface includes buttons for switching the aircraft navigation lights on and off, buttons for activating or deactivating the vertical take-off motors, a reset button that returns all elements to their default position, and a slider for adjusting both the speed and direction of the thrust motor. The interface also provides dedicated controls for camera positioning, enabling the user to rotate the onboard camera via servo drives in both horizontal and vertical directions.

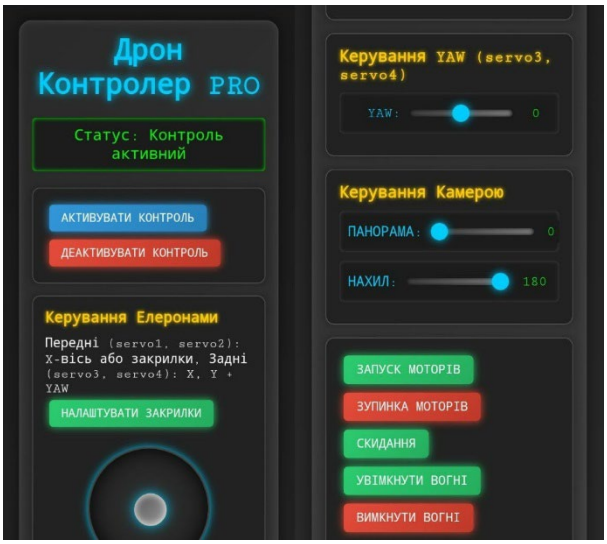

Fig. 1. User interface

Furthermore, the system was extended with an ARM-based Linux computer (Rockchip RK3188, 2 GB RAM) responsible for real-time video processing. A Python script, executed as a daemon, performs two primary tasks: (1) capturing and processing images from a USB camera, and (2) streaming the processed video over a local Flask-based HTTP server.[2] For image analysis, the system employs the OpenCV library with a

Histogram of Oriented Gradients (HOG) descriptor for pedestrian detection. [3]

Video Processing Workflow:

- The video stream from the onboard camera is processed on an ARM-based embedded Linux computer. The image-processing script is implemented in Python using the OpenCV library. The workflow consists of the following steps:
- Capturing frames from the USB camera.
- Downscaling the frames to optimize processing speed.
- Applying the HOG (Histogram of Oriented Gradients) descriptor for human detection [1].
- Drawing bounding boxes around detected objects.
- Encoding the processed frames into an MJPEG stream for transmission to the web interface.

As a result, the operator receives both control functionality and real-time visual feedback within a single interface. The script is executed in daemon mode to ensure continuous operation, while a Flask-based server streams the video via the /video_feed route.

In addition, servo-based camera positioning is integrated, enabling dynamic adjustment of the field of view and object tracking.

Figure 2 presents the structural diagram of the remote control system for the VTOL UAV stand, implemented on a controller based on the ESP32 microcontroller. The red, green, and white LEDs, serving as aviation navigation lights, are switched on through transistors upon command from the microcontroller.

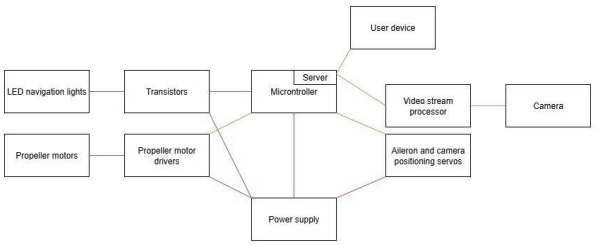

Fig. 2 Structural diagram

The current through the LEDs is limited by series-connected resistors. The vertical lift propeller drives are controlled via L298N motor driver ICs. PWM control signals for the servos of the ailerons and flap-ailerons are provided directly from the ESP32 microcontroller board. A camera mounted on a servo gimbal is also controlled by the ESP32, allowing remote positioning to adjust the viewing angle.

For the UAV stand (fig. 3), actuators were selected based on load:

- SG90 micro servo – camera gimbal. Torque ~1.8 kg·cm, speed 0.12 s/60°, supply voltage 4.8–6 V, weight 9 g.
- MG996R servo – wing ailerons. Torque 9.4 kg·cm, speed 0.14 s/60°, supply voltage 4.8–7.2 V, weight 55 g.
- MG90S servo – tail ailerons. Torque 2.2 kg·cm, speed 0.11 s/60°, supply voltage 4.8–6 V, weight 13 g.
- L298N motor driver controls propulsion motors, supply voltage 5–35 V, current 2 A/channel, PWM.

Fig. 3. VTOL UAV stand

CONCLUSION

The developed remote control system for the VTOL UAV stand, based on an ESP32 server using a local Wi-Fi network and a web interface, demonstrates stable operation provided the user remains within sufficient Wi-Fi signal range (near the UAV). The system is intuitive and easy to operate. Integration of motor drivers ensures smooth speed regulation of the propellers.

Additionally, the video stream from the UAV's onboard camera is transmitted via a ARM-based embedded Linux device connected to the system. The video feed is processed and displayed in real time within the same web interface used for control, providing the operator with both telemetry and visual feedback in a single environment.

REFERENCES

[1] Tsukanov O.S., Reut D.T. СИСТЕМА ДИСТАНЦІЙНОГО КЕРУВАННЯ ДРОНОМ ЧЕРЕЗ ВЕБ-СЕРВЕР НА ESP32 [Remote Drone Control System via ESP32 Web Server]. Zbirnyk tez dopovidei Vseukrainskoi naukovo-praktychnoi konferentsii zdobuvachiv vyshchoi osvity ta molodykh vchenykh "VODA. ZEMLYA. ENERGETYKA", Rivne, 15 May 2025. Rivne: NUWEE, 2025. [in Ukrainian]

[2] Pallets, "Flask," GitHub repository, https://github.com/pallets/flask

[3] OpenCV, "HOG Descriptor and Object Detection," Documentation, [Online]. Available: https://docs.opencv.org/4.x/d5/d33/structcv_1_1HOGDescriptor.html