https://doi.org/10.31891/2307-5732-2023-317-1-214-219
УДК 621.396: 551.508

ЧИГІНЬ Василь
Національний університет «Львівська політехніка»
https://orcid.org/0000 -0003-1593-6832
e-mail: vchygin@gmail.com
ПАЗИНЮК Михайло
Національний університет «Львівська політехніка»
e-mail: pazyniuk.m@gmail.com
ТЕРЕНДІЙ Ольга
Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України
Національний університет «Львівська політехніка»
https://orcid.org/0000-0001-9429-852X
e-mail: ovterendiy@gmail.com
МЕНЬШИКОВ Олексій
Національний університет «Львівська політехніка»
e-mail: oleksii.menshykov.ki.2022@lpnu.ua

КЕРУВАННЯ РОБОТОЮ ВІДДАЛЕНОГО ПРИСТРОЮ З ВИКОРИСТАННЯМ PYTHON-СЕРВЕРА FLASK

Досліджена комп'ютерна модель керування віддаленим пристроєм з застосуванням дистанційних хмарних технологій за заздалегідь заданими сценаріями з робочого стола користувача. Для цього створена експериментальна установка, яка включає віддалений пристрій типу коптер, персональний комп'ютер з операційною системою Windows, бортовий комп'ютер Raspberry Pi 3 з операційною системою Raspbian Linux, відеокамеру Pi Camera V2, автопілот Pixhawk. Опрацьована послідовність з'єднання між клієнтом (користувачем веб-браузера) та сервером (Raspberry Pi 3 із Flask-системою) та виконання віддалених команд за допомогою HTTP-запитів. У ролі фреймворку обрано систему Flask, яка є однією з найпростіших і має вичерпну, незначного об'єму документацію. За отриманими результатами досліджень запропоновано модель керування віддаленим пристроєм з робочого стола персонального комп'ютера користувача через бортовий комп'ютер без використання стандартного пульта керування та оператора.

Ключові слова: комп'ютерна модель керування, фреймворк Flask, віддалений пристрій, бортовий комп'ютер, робочий стіл користувача.

CHYHIN Vasyl, PAZYNIUK Mykhailo
Lviv Polytechnic National University
TERENDII Olha
Pidstryhach Institute for Applied Problems of Mechanics and Mathematics
National Academy of Sciences of Ukraine
Lviv Polytechnic National University
MENSHIKOV Oleksii
Lviv Polytechnic National University

CONTROLLING THE OPERATION OF THE REMOTE DEVICE USING FLASK PYTHON SERVER

A computer model of controlling an unmanned aerial vehicle (UAV) using remote cloud technologies according to predetermined scenarios from the user's desktop was studied. For this, an experimental setup was created, which includes an unmanned aerial vehicle of the quadcopter type, a personal computer with the Windows operating system, a Raspberry Pi 3 on-board computer with the Raspbian Linux operating system, a Pi Camera V2 video camera, and a Pixhawk autopilot. Worked out connection sequence between client (web browser user) and server (Raspberry Pi 3 with Flask system) and execution of remote commands using HTTP requests. As a framework, the Flask system was chosen, which is one of the simplest and has comprehensive, small-volume documentation. According to the obtained research results, a model of controlling the UAV from the desktop of the user's personal computer through the on-board computer without using a standard control panel and operator is proposed. According to the obtained research results, a model of remote control of an unmanned aerial vehicle with an on-board computer of the Raspberry Pi type using remote cloud technologies and a control program according to predetermined scenarios is proposed. At the same time, the user works only with the desktop of a personal computer, and accesses through an external or internal network and a Flask-type server to the on-board computer of the UAV without using a standard control panel and operator.

The proposed experimental setup includes an unmanned aerial vehicle of the quadcopter type with a Q450 frame and D2212-920 kv engines, a personal computer with a Windows operating system, a Raspberry Pi 3 on-board computer with a Raspbian Linux operating system. It made it possible to realize the set goal of researching the process of controlling an unmanned aerial vehicle using a Flask-type server. It was found that the full time of passing the HTTP request through the browser window and receiving a response from the server about the successful completion of the task does not exceed one second.

In the future, it is planned to work out more complex processes of launching and controlling an unmanned aerial vehicle in flights according to predetermined scenarios, including the performance of a task such as neutralizing unauthorized aerial vehicles, as well as returning one's own UAV to the place of deployment. The proposed system of remote control with the Flask server can also be useful for receiving an immediate notification when detecting certain sounds from foreign unmanned aerial vehicles, detecting and measuring their flight parameters.

Keywords: computer control model, Flask framework, unmanned aerial vehicle, quadcopter, on-board computer, user desktop.

Постановка проблеми у загальному вигляді
та її зв'язок із важливими науковими чи практичними завданнями

Актуальність виконання даної роботи виникла через необхідність створення інтуїтивно зрозумілої процедури керування віддаленим пристроєм для користувача за допомогою його веб-браузера та іншого допоміжного програмного забезпечення. Для виконання цієї задачі обрано мову програмування Python як одну з найбільш пристосованих для подібних завдань через великий набір додаткових фреймворків (інфраструктур програмних рішень) та бібліотек, процес встановлення яких є зрозумілим та не викликає надмірної складності у користувачів. Цей набір компонент мови Python дозволяє нескладну підтримку при використанні, адже їх оновлення забезпечується авторами програмного забезпечення, а від користувачів вимагається лише коректувати відповідний файл із необхідними версіями та запускати відповідну команду.

Із відомих в літературі програмних веб-середовищ обрано фреймворк Flask, він є одним з найпростіших при написанні програмного забезпечення. Альтернативою є фреймворк Django, проте при виконанні даного завдання його функціонал та розмір виявились надто громіздкими. Фреймворк Flask має вичерпну, невеликого об'єму документацію, яка включає у себе інструкції для швидкого початку роботи, модифікуючи які можна отримати повністю працюючу програму за мінімальний період часу. Таким чином, цей фреймворк дозволяє спростити і пришвидшити процедуру керування віддаленим пристроєм.

Аналіз останніх досліджень і публікацій

У попередніх працях одного з авторів [1–5] досліджувалися процеси виявлення і вимірювання параметрів польоту віддалених пристроїв за допомогою наземних установок із звуковими і фото-приймачами. Проте запропоновані системи у вказаних роботах дозволяли виконувати лише польоти за наперед заданими GPS-координатами, але не дозволяли виконувати команди віддалено за допомогою взаємодії користувача і бортового мікрокомп'ютера. У праці [6] розглядається можливість побудови так званого "помічника фермера" на основі створення комп'ютерних засобів керування віддаленим пристроєм із застосуванням дистанційних хмарних обчислень. За отриманими результатами запропоновано модель керування віддаленим пристроєм з робочого стола персонального комп'ютера користувача через бортовий комп'ютер без використання стандартного пульта керування та оператора. Для моделювання дистанційного керування процесами використано командний bat-файл і типову програму зв'язку двох комп'ютерів типу VNC (Virtual Network Computing). В роботах [1–6] не розглядалася можливість взаємодії між клієнтом та сервером на основі веб-фреймворків. Враховуючи результати досліджень [4–6], у даній роботі використали досвід встановлення на віддалений пристрій додаткової техніки, у тому числі мікрокомп'ютера Raspberry Pi 3, а також камери Raspberry Pi V2.

У доступних українських публікаціях [7,8] не виявлено робіт, зв'язаних із використанням бортового комп'ютера для виконання запрограмованих команд віддаленим пристроєм, відпрацювання одного з обраних сценаріїв польоту чи інших дій типу знешкодження несанкціонованого віддаленого пристрою і т.п. В окремих зарубіжних роботах [9,10] висвітлено виконання окремих команд наперед запрограмованими системами, зокрема, при фотозахопленні об'єктів.

В інтернет виданнях [11, 12] виявлено ряд закритих програмних забезпечень, зв'язаних з відлагодженням передполітних режимів та автоматичним керуванням польотів за наперед заданою схемою. Однак, аналогів з використанням взаємодії між клієнтом та сервером на основі веб-фреймворків серед них не встановлено. Типові програми керування віддаленим пристроєм типу Mission Planner [13] і Q-GroundControl [14] призначені для виконання ряду строго визначених дій, зокрема, перевірки всіх сенсорів віддаленого пристрою перед пуском, заведення моторів, вильоту у вертикальному напрямку на певну задану висоту і виконання самого польоту у певному режимі (Gaided, Loiter тощо). Недоліком цих програм є обмежена кількість можливих сценаріїв, які можна відпрацювати на сервері, а також неможливість модифікувати програмне забезпечення через його закриту структуру. Не розглядається також можливість виконання команд віддалено при взаємодії сервера з користувачем.

У даній роботі у ролі бортового комп'ютера обрано Raspberry Pi 3 через сукупність таких задовільних характеристик, як потужність процесора, об'єм оперативної пам'яті, ціна за готовий виріб, розмір та наявність необхідних інтерфейсів під'єднання [15]. У моделі комп'ютера Broadcom BCM2837B0 SoC встановлено процесор ARMv8 Cortex-A53 з тактовою частотою 1.4 ГГц, графічний процесор Video Core IV® Multimedia з двома графічними ядрами, 1ГБ LPDDR2 SDRAM оперативної пам'яті та Wi-Fi модуль на 2.4 ГГц і 5 ГГц IEEE 802.11. Найближчими аналогами, які розглянуті у праці [16], є Raspberry Pi Zero W та Raspberry Pi 4. Перший характеризується значно меншим розміром та легшим встановленням на віддалений пристрій, проте менший об'єм ресурсів не дозволяє йому стабільно працювати з поставленими політними задачами. Новіша модель Raspberry Pi 4 надає значно більше потужності при майже такому ж форм-факторі та розмірах корпусу, але має вищу ціну та надлишкові ресурси, які не знадобляться для виконання поточних завдань для віддаленого пристрою.

Серед програмного забезпечення, яке може виконувати роль керування сервером, розглянули такі аналоги фреймворку Flask, як Django та NodeJS, кожен з яких має свої переваги та недоліки [17]. Оскільки головною метою створеної програми є швидке виконання невеликих команд чи їх комбінацій, перш за все до уваги бралась можливість фреймворку надавати інтерфейс взаємодії між ним та операційною системою, а також його простота та легкість встановлення й відлагодження у разі виникнення помилок. Перший аналог,

Django, як описано у статті [17], серед основних недоліків має більші системні вимоги для пуску та вищий рівень споживання ресурсів при роботі веб-сервера, що є критичним для мікрокомп'ютера Raspberry Pi 3. Аналог NodeJS також забезпечує подібний інтерфейс взаємодії користувача з системою, проте має меншу інтегрованість з портативною ОС Raspbian Linux, яка найкраще працює з Python-додатками та має завчасно налаштовані системні змінні для таких задач.

Серед подібних за призначенням веб-серверів також виділили фреймворк TurboGears [18]. Відмінною особливістю його є розширюваність – його можна розширити за допомогою всіх видів простих плагінів типу WSGI. Крім того, він підтримує горизонтальне розділення даних (шардінг) і його більше зосереджують на об'єктно-орієнтованій парадигмі, яка є суттєво іншим підходом до більшості фреймворків. У зв'язку з цим TurboGears також не розглядався у ролі фреймворка для керування польотами віддаленого пристрою, оскільки для завдань цієї роботи не було необхідності у створенні окремих класів та об'єктів, достатньо викликати потрібні функції та отримувати відповідь від сервера. Розглянуто також фреймворк Web2py [18], який зосереджений на забезпеченні повного стеку функцій та роботи з клієнтськими запитами. Головною перевагою Web2py є його повнофункціональна IDE (Integrated development environment), яка дозволяє змінювати веб-сайт з будь-якого веб-браузера після його розгортання. І хоча він здатний забезпечити досягання поставлених у наших дослідженнях цілей, рішення про відмову від роботи з Web2py виникло через дещо застарілий код та методи, на заміну яких із оновленням мови Python прийшли більш сучасні аналоги.

Обраний фреймворк Flask [19] є невеликим і доступним для користування, він написаний мовою Python, що пропонує корисні інструменти та функції для полегшення процесу створення веб-застосунків. Як вказано його творцями [19], він забезпечує гнучкість і є більш доступним фреймворком для нових дослідників, оскільки дозволяє створити веб-програму швидко, використовуючи лише один файл. Flask-застосунок використовує механізм шаблонів Jinja для динамічного створення HTML-сторінок з використанням знайомих понять у Python, таких як змінні, цикли, списки тощо. Дані шаблони дозволяють спрощено продемонструвати користувачеві результат виконання програми, адже автоматично генеруються на основі переданого з програми тексту.

Попри те, що Flask є власне веб-фреймворком, саме такий вид взаємодії з користувачем є найбільш спрощеним та не вимагає встановлення додаткових компонент, як це може бути у випадку із окремою віконною програмою [20]. Таку програму можна було би створити за допомогою додаткових бібліотек мови програмування Python, як-от Tkinter чи PyGame, проте, для їхнього пуску вимагалось би встановлення додаткових залежностей, чого не вимагається від стандартних браузерів. Через незалежність більшості сучасних браузерів від сімейства операційних систем, на які вони встановлюються, для даної роботи прийнято рішення користуватись веб-браузером Google Chrome та операційною системою Windows 10, які, згідно зі статистикою, користуються найбільшою популярністю. Аналогічними варіантами могли б стати ОС GNU/Linux або Apple macOS, які дозволяють користувачам виходити у глобальну мережу Інтернет або у локальну мережу зв'язаних між собою пристроїв.

Формулювання цілей статті

Метою роботи є відлагодження і дослідження системи, що включає бортовий мікроком'ютер у парі з віддаленим пристроєм, а також персональний комп'ютер з браузером Google Chrome для оптимізації взаємодії між клієнтом та сервером на основі веб-фреймворку. При цьому слід описати структуру системи та алгоритм віддаленого керування пристроями. Основними вимогами до програми є простота встановлення та можливість легкого використання кінцевими користувачами, а також невимогливість до ресурсів через специфіку обраного мікрокомп'ютера Raspberry Pi.

Методика дослідження і програма керування віддаленим пристроєм

Процес виконання програми керування віддаленим пристроєм задається користувачем з власного веб-клієнта (браузера) та не вимагає встановлення на власний комп'ютер додаткових компонент. У користувача є можливість обрати серед кількох доступних команд, що він планує виконати. Для прикладу розглянемо одну з команд програми керування call( ), яка викликає розміщений на Raspberry Pi сервері Python-скрипт. Користувач робить запит із веб-клієнта на адресу http://<адреса>:<порт>/call та очікує на відповідь від сервера про результат роботи. При цьому відправляється запит типу HTTP GET, який проходить відповідну перевірку на сервері та повертає статус-код 200 із додатковим повідомленням.

Основна робота функції call( ), написаної у Flask-застосунку, полягає у виклику вбудованим Python-модулем subprocess методу subprocess.call( ), аргументами до якого передаються шлях до бінарного файлу та шлях до програми керування віддаленим пристроєм, яку обрано для подальшого відпрацювання. Після успішного завершення викликаної методом subprocess.call( ) програми Flask-застосунок завершує виконання функції call( ) та за допомогою ключового слова return повертає відповідь користувачеві у вигляді HTTP Response. Останню користувач може спостерігати у вікні свого браузера. У ході виконання роботи використали частини зразків програм з веб-ресурсів DigitalOcean щодо Flask та DigitalOcean щодо модуля subprocess.

На рис. 1 зображена блок-схема запропонованої комп'ютерної моделі керування віддаленим пристроєм з використанням фреймворку Flask і сервера. Тут цифрами позначені наступні елементи

програми: 1 – Web–клієнт, тобто браузер, через який користувач керує пуском віддаленого пристрою, у нашому випадку використовувався Google Chrome версії 108.0.5359.125, 2 – HTTP-запит, який передається через інтернет-мережу до сервера та використовується для передавання команди пуску віддаленого пристрою, 3 – сервер на основі Raspberry Pi, 4 – опрацювання HTTP-запиту фреймворком Flask на сервері, тобто перевірка запиту на його коректність та вміст і, як наслідок, виконання відповідних до запиту команд, 5 – пуск однієї з обраних програм керування віддаленим пристроєм (пуск моторів, виліт на задану висоту і т.п.), 6 – повернення статус-коду та повідомлення користувачеві (наприклад, “200, Executed successfully”) про результат виконання програми керування віддаленим пристроєм (успішний пуск, помилка сервера і т.п.), HTTP-відгук, передається інтернет мережею назад до клієнта, 7 – очікування Web-клієнтом (браузером) на відповідь від сервера.

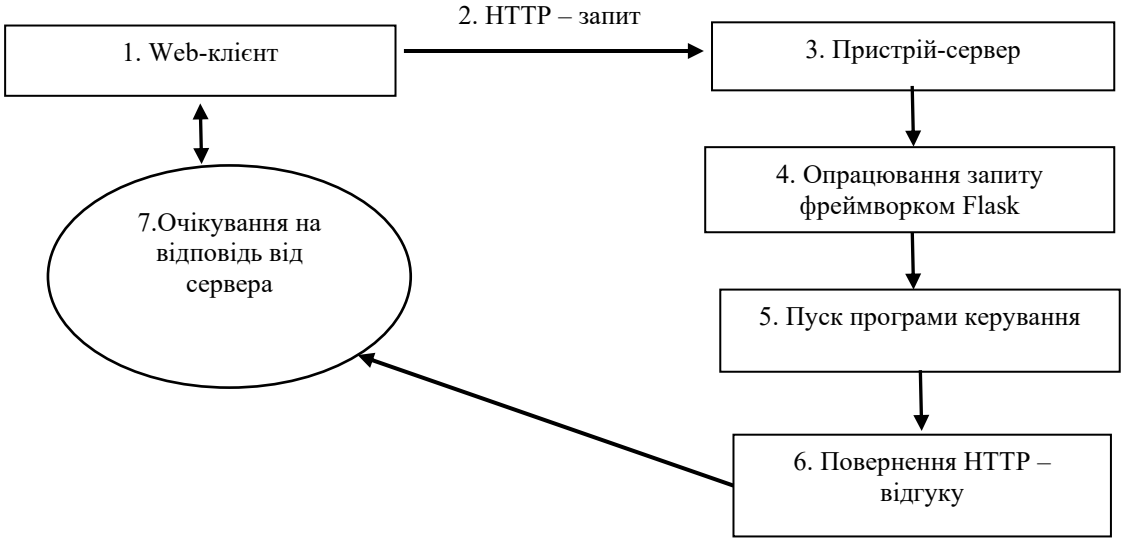

Рис. 1. Блок-схема комп’ютерної моделі керування пристроєм

Для дослідження роботи системи і програми керування віддаленим пристроєм створили експериментальну установку, яка включає пристрій типу квадрокоптер з рамою Q450 і двигунами D2212-920 kv, персональний комп’ютер з операційною системою Windows, бортовий комп’ютер Raspberry Pi 3 з операційною системою Raspbian Linux, відеокамеру Pi Camera V2, автопілот Pixhawk. Опрацьована послідовність з’єднання між клієнтом (користувачем веб-браузера) та сервером (Raspberry Pi 3 із Flask-системою) та виконання віддалених команд за допомогою HTTP-запитів.

У ролі Web-клієнта (користувача) служив браузер Google Chrome, встановлений на ноутбуці із операційною системою Windows 10, який розмістили на віддалі 100 метрів від стартової площадки віддаленого пристрою. Останній міг виконувати пуск моторів і польоти за заздалегідь заданими сценаріями [5]. У даній роботи виконували тільки пуск і зупинку моторів за допомогою з’єднання між клієнтом (користувачем веб-браузера) з робочого стола користувача та сервером (Raspberry Pi 3 із Flask-системою). При цьому HTTP-запити сформували у bat-файлі, вписуючи туди команди типу start C:"\\program files\\Google\\Chrome\\Application\\chrome.exe" http://192.168.0.178:5000/call. У даному випадку використали внутрішню мережу для зв’язку (192.168.0.178 – IP-адреса внутрішньої мережі).

Сервер складений мовою Python і включає Flask-систему із директивами типу наступних:

from flask import Flask – імпорт бібліотеки Flask,

import subprocess – імпорт бібліотеки subprocess,

app = Flask(__name__) – ініціалізація сервера,

@app.route('/call') – декоратор функції call,

subprocess.call(["/bin/bash", "Drone.sh"], stdout=output) – команда на пуск файлу Drone.sh,

return 'Executed successfully' – повертання користувачеві повідомлення, яке появляється на робочому столі у вікні браузера Google Chrome.

Аналіз результатів дослідження

При дослідженнях процес керування віддаленим пристроєм – коптером з рамою Q450 проходив наступним чином. Після пуску bat-файлу з командою типу start C:"\\program files\\Google\\Chrome\\Application\\chrome.exe" http://192.168.0.178:5000/call відкривається вікно браузера і надсилається HTTP-запит на внутрішню IP-адресу 192.168.0.178 бортового комп’ютера і на порт 5000. Функція call активізує команду на відкриття файлу drone.sh. При цьому пускаються мотори коптера і через деякий час вимикаються.

Користувач очікує не більше 1 секунди на відповідь із сервера. При успішному виконанні сервером команди пуску моторів, сповіщення типу 'Executed successfully' появляється у вікні браузера.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

За отриманими результатами досліджень запропоновано модель керування віддаленим пристроєм з бортовим комп'ютером типу Raspberry Pi з використанням дистанційних хмарних технологій і програми керування за заздалегідь заданими сценаріями. При цьому користувач працює тільки з робочим столом персонального комп'ютера і виходить через зовнішню чи внутрішню мережу та сервер типу Flask на бортовий комп'ютер віддаленого пристрою без використання стандартного пульта керування та оператора.

Запропонована експериментальна установка включає пристрій типу коптер з рамою Q450 і двигунами D2212-920 kv, персональний комп'ютер з операційною системою Windows, бортовий комп'ютер Raspberry Pi 3 з операційною системою Raspbian Linux. Вона дозволила реалізувати поставлену мету дослідження процесу керування віддаленим пристроєм за допомогою сервера типу Flask. Виявлено, що повний час проходження HTTP-запиту через вікно браузера і отримання відповіді від сервера про успішне виконання завдання не перевищує однієї секунди.

У перспективі планується опрацювати складніші процеси пуску і керування віддаленим пристроєм у польотах за наперед заданими сценаріями, в тому числі виконання завдання типу знешкодження несанкціонованих літальних апаратів, а також повертання власного віддаленого пристрою на місце дислокації. Запропонована система віддаленого керування з Flask-сервером може бути корисною також для отримання негайного сповіщення при виявленні певних звуків від чужих безпілотних літальних апаратів, виявленні і вимірюванні параметрів їх польоту.

Література

1. Чигінь В.І. Вдосконалення способу виявлення безпілотних літальних апаратів за результатами спектрального аналізу акустичних сигналів / В.І. Чигінь, М.М. Проценко, Ю.В. Шабатура, М.В. Бугайов // Військово-технічний збірник АСВ. – 2019. – № 20. – С. 58–63. DOI: https://doi.org/10.33577/2312-4458.20.2019.58-63

2. Чигінь В.І. Вимірювання координат безпілотних літальних об'єктів з використанням звукової і відеоапаратури / В.І. Чигінь, П.Я. Михайлишин // V Всеукраїнська наук.-техн. конф. у царині метрології «Technical Using of Measurement – 2019», м. Славське, 29 січ. – 2 лют. 2019 р. – С. 10–12.

3. Федишин Н. Дослідження звукової системи виявлення літальних об'єктів з використанням гармонік в акустичному сигналі / Н. Федишин, В.І. Чигінь // Міжнар. конф. молодих вчених та аспірантів "ІЕФ-2017. Інститут електронної фізики НАН України", м. Ужгород, 23–26 трав. 2017. – С. 13–15.

4. Чигінь В. Експериментальний безпілотний авіаційний комплекс для фотозахоплення / В.І. Чигінь, П.Я. Михайлишин // Вісник Хмельницького Національного університету. Технічні науки. – 2019. – № 2 (271). – С. 202–205.

5. Чигінь В. Експериментальні дослідження безпілотного авіаційного комплексу при фото захопленні / В. Чигінь, П Михайлишин // Вісник Хмельницького Національного університету. – 2020. – № 3(285). – Р. 170–174.

6. Чигінь В. Створення комп'ютерних засобів керування автономним літальним апаратом з застосуванням дистанційних хмарних обчислень / Василь Чигінь // Комп'ютерні системи та мережі. Кафедра ЕОМ НУЛП. – 2021. – № 3. – С. 106–113.

7. Salnik Yu.P., Matala I.V., Onishchenko V.A. The current state of equipping the Armed Forces of Ukraine with unmanned aviation complexes. The team sciences works of Kharkiv University. Air Forces, 2011, Issue 2 (28). P. 46–51.

8. Glotov V., Gunina A., Teleshchuk Yu. Analysis of the possibilities of using unmanned aerial vehicles for military purposes. Photogrammetry, geoinformation systems and cartography. 2017. V. 1 (33). P. 139–146.

9. Автоматичне відстеження об'єкта на Python. URL: https://robotos.in/uroki/avtomaticheskoe-otslezhivanie-ob-ekta-na-python

10. Ballon Finder. URL: https://www.youtube.com/watch?v=yRmXwRqPesY&feature=youtu.be.

11. DJI Mavic 2 pro active track 2.0 mode. URL: https://www.youtube.com/watch?v=qEmd5g2fMcE&feature=youtu.be

12. Experimental DJI system. URL: https: //vido.com.ua/article/12354/ekspierimientalnaia-sistiema-dji-pomoghaiet-priedotvrashchat-stolknovieniia-dronov-zalogh-biezopasnogho-vozdushnogho-dvizhieniia

13. Mission-planner. URL: http://www.ardupilot.su/wiki/arducopter/install-mission-planner.html (Accessed: 05 December 2021)

14. QGroundControl. URL: http://qgroundcontrol.com (Accessed: 05 December 2021)

15. Raspberry Pi Documentation. URL: https://www.raspberrypi.com/documentation/

16. Raspberry Pi Models Comparison. URL: https://socialcompare.com/en/comparison/raspberrypi-models-comparison

17. Flask vs Django vs NodeJS. URL: https://www.softwaretestinghelp.com/django-vs-flask-vs-node/

18. 5 Cool Alternatives To Django and Flask For Deploying Endpoints And FullStack. URL: https://towardsdatascience.com/5-cool-alternatives-to-django-and-flask-for-deploying-endpoints-5c99a066696

19. Flask documentation. URL: https://flask.palletsprojects.com/en/2.0.x/

20. Python GUI Programming with Tkinter. URL: https://realpython.com/python-gui-tkinter/

References

1. Chyhin V.I. Vdoskonalennia sposobu vyiavlennia bezpilotnykh litalnykh aparativ za rezultatamy spektralnoho analizu akustychnykh syhnaliv / V.I. Chyhin, M.M. Protsenko, Yu.V. Shabatura, M.V. Buhaiov // Viiskovo-tekhnichnyi zbirnyk ASV. – 2019. – № 20. – S. 58–63. DOI: https://doi.org/10.33577/2312-4458.20.2019.58-63

2. Chyhin V.I. Vymiriuvannia koordynat bezpilotnykh litalnykh obiektiv z vykorystanniam zvukovoi i videoaparatury / V.I. Chyhin, P.Ia. Mykhailyshyn // V Vseukrainska nauk.-tekhn. konf. u tsaryni metrolohii «Technical Using of Measurement – 2019», m. Slavske, 29 sich. – 2 liut. 2019 r. – S. 10–12.

3. Fedyshyn N. Doslidzhennia zvukovoi systemy vyiavlennia litalnykh obiektiv z vykorystanniam harmonik v akustychnomu syhnali / N. Fedyshyn, V.I. Chyhin // Mizhnar. konf. molodykh vchenykh ta aspirantiv "IEF-2017. Instytut elektronnoi fizyky NAN Ukrainy", m. Uzhhorod, 23–26 trav. 2017. – S. 13–15.

4. Chyhin V. Eksperymentalnyi bezpilotnyi aviatsiinyi kompleks dlia fotozakhop¬lennia / V.I. Chyhin, P.Ia. Mykhailyshyn // Visnyk Khmelnytskoho Natsionalnoho universytetu. Tekhnichni nauky. – 2019. – № 2 (271). – S. 202–205.

5. Chyhin V. Eksperymentalni doslidzhennia bezpilotnoho aviatsiinoho kompleksu pry foto zakhoplenni / V. Chyhin, P Mykhailyshyn // Visnyk Khmelnytskoho Natsionalnoho universytetu. – 2020. – № 3(285). – R. 170–174.

6. Chyhin V. Stvorennia kompiuternykh zasobiv keruvannia avtonomnym litalnym aparatom z zastosuvanniam dystantsiinykh khmarnykh obchyslen / Vasyl Chyhin // Kompiuterni systemy ta merezhi. Kafedra EOM NULP. – 2021. – № 3. – S. 106–113.

7. Salnik Yu.P., Matala I.V., Onishchenko V.A. The current state of equipping the Armed Forces of Ukraine with unmanned aviation complexes. The team sciences works of Kharkiv University. Air Forces, 2011, Issue 2 (28). P. 46–51.

8. Glotov V., Gunina A., Teleshchuk Yu. Analysis of the possibilities of using unmanned aerial vehicles for military purposes. Photogrammetry, geoinformation systems and cartography. 2017. V. 1 (33). R. 139–146.

9. Avtomatychne vidstezhennia obiekta na Python. URL: https://robotos.in/uroki/avtomaticheskoe-otslezhivanie-ob-ekta-na-python

10. Ballon Finder. URL: https://www.youtube.com/watch?v=yRmXwRqPesY&feature=youtu.be.

11. DJI Mavic 2 pro active track 2.0 mode. URL: https://www.youtube.com/watch?v=qEmd5g2fMcE&feature=youtu.be

12. Experimental DJI system. URL: https: //vido.com.ua/article/12354/ekspierimientalnaia-sistiema-dji-pomoghaiet-priedotvrashchat-stolknovieniia-dronov-zalogh-biezopasnogho-vozdushnogho-dvizhieniia

13. Mission-planner. URL: http://www.ardupilot.su/wiki/arducopter/install-mission-planner.html (Accessed: 05 December 2021)

14. QGroundControl . URL: http://qgroundcontrol.com (Accessed: 05 December 2021)

15. Raspberry Pi Documentation. URL: https://www.raspberrypi.com/documentation/

16. Raspberry Pi Models Comparison. URL: https://socialcompare.com/en/comparison/raspberrypi-models-comparison

17. Flask vs Django vs NodeJS. URL: https://www.softwaretestinghelp.com/django-vs-flask-vs-node/

18. 5 Cool Alternatives To Django and Flask For Deploying Endpoints And FullStack. URL: https://towardsdatascience.com/5-cool-alternatives-to-django-and-flask-for-deploying-endpoints-5c99a066696

19. Flask documentation. URL: https://flask.palletsprojects.com/en/2.0.x/

20. Python GUI Programming with Tkinter. URL: https://realpython.com/python-gui-tkinter/