

**О. І. КУЧЕРЕНКО**аспірант кафедри інженерії програмного забезпечення  
Державний університет «Житомирська політехніка»  
ORCID: 0009-0000-1488-5645**Т. А. ВАКАЛЮК**доктор педагогічних наук, професор,  
завідувач кафедри інженерії програмного забезпечення  
Державний університет «Житомирська політехніка»  
ORCID: 0000-0001-6825-4697

## ОГЛЯД ТЕХНІЧНИХ ТА ПРОГРАМНИХ ЗАСОБІВ КЕРУВАННЯ БПЛА

Стаття присвячена огляду технічних та програмних засобів керування безпілотними літальними апаратами (БПЛА), що є актуальною темою у сучасній робототехніці та авіаційній індустрії. Розвиток безпілотних літальних апаратів породжує попит на комплексні технології та програмне забезпечення, які забезпечують ефективне управління та навігацію цими системами. У статті розглядаються основні аспекти технічних засобів керування БПЛА, включаючи контролери польоту, сенсори, системи стабілізації, комунікаційні системи, апаратне та програмне забезпечення для управління рухом та збору даних. Описано характеристики апаратних засобів, що базуються на архітектурах FPGA, ARM, Atmel та Raspberry Pi. Проведено аналіз доступних програмних засобів керування БПЛА, а саме: ArduPilot, Multiwii, AutoQuad, LibrePilot, AuterionOS та розробки від Dronecode Community. В аналізі визначено сумісність програмних засобів з польотними контролерами. Окремо розглянуто системи управління високого рівня, які дозволяють розробникам створювати власні додатки та інтеграції для різноманітних завдань та додаткових функцій безпілотних літальних апаратів. Виконано порівняння систем управління високого рівня між собою за наступними критеріями: модульність структури, підтримка БПЛА різної конструкції, підтримка акроконтролю повітряного судна, підтримка багатозадачності, підтримка багатьох польотних платформ, плагіно-орієнтованість архітектури, використане проміжне програмне забезпечення, відкрите програмне забезпечення. Приведена у роботі порівняльна інформація програмних та апаратних засобів та систем високого рівня покликана спростити задачу остаточного вибору засобів керування БПЛА. Ця потреба виникає через зростання числа галузей, де використовуються безпілотні літальні апарати. Під час проведення дослідження виявлено існування великої кількості як технічних, так і програмних засобів керування БПЛА. Для проведення огляду були взяті найбільш відомі з них.

**Ключові слова:** дрон, БПЛА, технічні засоби, програмні засоби, управління, контролер польоту, автопілот, планування маршрутів.

**О. І. KUCHERENKO**Postgraduate Student at the Software Engineering Department  
Zhytomyr Polytechnic State University  
ORCID: 0009-0000-1488-5645**T. A. VAKALIUK**Doctor of Pedagogical Sciences, Professor,  
Head of the Software Engineering Department  
Zhytomyr Polytechnic State University  
ORCID: 0000-0001-6825-4697

## OVERVIEW OF TECHNICAL AND SOFTWARE MEANS OF UAV CONTROL

The article is devoted to the review of technical and software control of unmanned aerial vehicles (UAVs), which is an actual topic in modern robotics and the aviation industry. The development of unmanned aerial vehicles creates a demand for complex technologies and software that provide effective control and navigation of these systems. The article examines the main aspects of the technical means of controlling UAVs, including flight controllers, sensors, stabilization systems, communication systems, hardware and software for motion control and data acquisition. The characteristics of hardware based on FPGA, ARM, Atmel and Raspberry Pi architectures are described. An analysis of available UAV control software, namely: ArduPilot, Multiwii, AutoQuad, LibrePilot, AuterionOS and developments from the Dronecode Community, was carried out. The analysis determined the compatibility of software with flight controllers. High-level control systems that allow developers to create their own applications and integrations for various tasks and additional functions of unmanned aerial vehicles are separately considered. A comparison of high-level control systems was made

*among themselves according to the following criteria: modularity of the structure, support for UAVs of multiple frames, support for aircraft rate output, multi-agency support, support for many flight platforms, plugin-oriented architecture, used middleware, open source software. The comparative information of software and hardware tools and high-level systems presented in the work is designed to simplify the task of final selection of UAV control tools. This need arises due to the growing number of industries where unmanned aerial vehicles are used. During the research, the existence of a large number of both technical and software control tools for UAVs was revealed. The most famous of them were taken for the review.*

**Key words:** drone, UAV, technical means, software, control, flight controller, autopilot, route planning.

### Постановка проблеми

У цей час існує велика кількість безпілотних літальних апаратів (БПЛА), які застосовуються у різних сферах, включаючи аерофотозйомку, геодезію, агрономію, пошук і рятувальні роботи тощо. Кожна із сфер застосування має свої особливості до побудови маршрутів руху БПЛА. Невід'ємною частиною механізму керування дронами є зв'язок програмної та апаратної складових. Найважливіші аспекти, що потребують уваги в цьому контексті, включають розробку ефективних програмних платформ для керування дронами, реалізацію надійних систем автопілоту, розвиток технологій обробки сигналів для забезпечення точності навігації та стабільності польоту, а також створення високоефективних методів управління БПЛА в умовах обмежених потужностей обробки даних та зв'язку.

Наразі існує велика кількість розробок програмних та апаратних комплексів від різних виробників. Кожна з них має свої переваги та недоліки.

Окрім низькорівневих систем, важливо звернути увагу на системи управління високого рівня. Вони покликані спростити вирішення як базових, так і більш складних шаблонних задач при керуванні БПЛА. Тому дослідження програмних та апаратних засобів керування дронами є необхідною умовою вибору оптимальних рішень, оскільки конкретні галузі застосування БПЛА вимагають врахування своїх особливостей.

### Аналіз останніх досліджень і публікацій

Відомості щодо програмних та технічних засобів керування дронами у більшості випадків представлена на сайтах виробників, проте існують публікації, в яких представлено порівняння характеристик різних засобів. Цій темі свою увагу присвятили такі науковці: М. Бронз (M. Bronz) [9], П. Бріссе (P. Brisset) [8], Томаш Бака (Tomas Baca) [19], Матоус Врба (Matous Vrba) [19], Д. Л. Габріель (D. L. Gabriel) [6], М. Гопраз (M. Gorraz) [8], П. Д. Гровс (P. D. Groves) [3], Л. Майер (L. Meier) [5, 7], І. Маза (I. Maza) [4], Й. Меєр (J. Meyer) [6], Мартін Моліна (Martin Molina) [18], С. В. Нававі (S. W. Nawawi) [2], Роберт Пенічка (Robert Penicka) [19], Матеї Петрлік (Matej Petrlik) [19], Ф. дю Плессіс (F. du Plessis) [6], С. Сабікан (S. Sabikan) [2], Карлос Сампедро (Carlos Sampedro) [18], Д. Скарлатті (D. Scarlatti) [5, 7], Рамон А Суарез Фернандес (Ramon A Suarez Fernandez) [18], Мартін Саска (Martin Saska) [19], Дж. Тайлер (J. Tyler) [8], П. Танскаанен (P. Tanskanen) [5, 7], П. С. Уард (P. S. Huard) [8], Ф. Фраундорфер (F. Fraundorfer) [5, 7], Л. Хенг (L. Heng) [5, 7], Даніель Херт (Daniel Hert) [19], Хосе Луїс Санчес-Лопес (Jose Luis Sanchez-Lopez) [18], Ч. Янь (Z. Yan) [1], Ч. Чженьці (X. Chengqi) [1], Войтех Шпурні (Vojtech Sprny) [19] та інші.

### Формулювання мети дослідження

Метою роботи є огляд існуючих технічних та програмних засобів керування БПЛА, що дозволяють програмно задавати їх маршрути руху.

### Викладення основного матеріалу дослідження

Перш ніж перейти до огляду безпосередньо засобів керування БПЛА, розглянемо базові відомості про будову системи автопілота. Основними компонентами системи автопілота є:

- контролер польотів (Flight Controller);
- рухова система (Propulsion System);
- сенсори (Sensors);
- комунікаційні системи (Communication Systems) [1].

**Контролер польотів** – це основна апаратна плата для роботи системи БПЛА [2]. Вона керує двигунами, взаємодіє з внутрішніми або зовнішніми датчиками, реалізує оцінку положення, керування, навігацію та підтримує зв'язок з наземними системами керування або сусідніми БПЛА. Її продуктивність залежить від вбудованого пристрою, який використовується. Для побудови контролера польоту використовуються модулі ARM, Atmel, Arduino та інші. Більшість польотних контролерів використовують 32-розрядний процесор і лише деякі 8-розрядний. Контролер польоту знаходиться посередині та взаємодіє з іншими блоками через стандартні комунікаційні інтерфейси, наприклад: мережа контролера (CAN), широтно-імпульсна модуляція (PWM), універсальний асинхронний приймач/передавач (UART) тощо.

**Рухова система** – система, що приводить БПЛА у рух. Для багатороторних БПЛА часто для приводу гвинтів використовуються безщіточні двигуни постійного струму (BLDC), керовані електронним контролером швидкості (ESC), але в деяких випадках, що трапляється рідше, також щіткові двигуни, щоб заощадити витрати на ESC [6].

Інший варіант безпілотних літальних апаратів із фіксованим крилом використовує гвинт, як багатороторний безпілотний літальний апарат для вертикального зльоту та посадки (VTOL). VTOL літає як дрон із фіксованим крилом, коли він знаходиться в повітрі завдяки фіксованому крилу, але для позиціонування критичних польотів, зльоту та посадки він використовує пропелери, як мультиротор.

**Сенсори.** БПЛА використовують датчики для виявлення змін в оточенні, що дозволяє їм збирати важливі дані про об'єкт, який вони перевіряють, і краще маневрувати. Для реєстрації змін і збору різноманітної інформації в БПЛА використовуються різні типи датчиків. Загальні датчики: 1) Інерціальний вимірювальний блок (IMU): це основний компонент інерціальних систем навігації та маневрування в системі БПЛА. Він використовує комбінацію акселерометрів і гіроскопів для точної оцінки положення БПЛА, включаючи крен і тангаж. Крім того, він використовує магнітометри, доповнення до акселерометра та гіроскопа, для вимірювання курсу (тобто повороту), а також для вимірювання та звітування питомої сили дрона, кутової швидкості та магнітного поля, що оточує дрон. 2) Барометр: використовується для вимірювання атмосферного тиску та обчислення висоти дрона. Він може виявляти рух у кілька сантиметрів. 3) GNSS: Глобальна навігаційна супутникова система [3], вимірює місцезнаходження дрона, обчислюючи його відстань від попередньо визначених супутників, таких як Global Positioning System (GPS), GLONASS, Galileo та BeiDuo. Зазвичай датчики, такі як IMU та барометри, поєднуються з GNSS, щоб підвищити точність показників дрона.

**Комунікаційні системи.** Для взаємодії з безпілотниками використовуються два різних типи зв'язку: 1) Наземна станція управління (Ground Control Station, або GCS) – це програмне забезпечення, яке працює на обчислювальній пристрої (ПК, планшет тощо) [4]. GCS використовується для бездротового зв'язку з дроном, щоб стежити за тим, куди він летить, встановлювати маршрутні точки або виконувати нові команди. Щоб надсилати й отримувати такі дані, до обчислювального блоку необхідно приєднати телеметричний апаратний радіоблок для виконання операцій керування. Він реалізує протокол послідовного з'єднання MAVlink [5]. Прикладами програмного забезпечення GCS є: QGroundControl, Mission Planner, APM Planner. 2) Передавач радіокерування (R/C): він використовується для мінімального керування рухом дрона (дросьель) і орієнтацією (нахил, крен і поворот). Команди керування перетворюються в сигнали PWM (Pulse Width Modulation) або імпульсно-позиційної модуляції (Pulse Position Modulation, або PPM) і передаються контролеру польоту, який використовує їх для управління двигунами дрона.

Розглянемо різні **апаратні платформи** (Hardware Platforms) з групуванням відповідно до використовуваного мікроконтролера або чіпа (FPGA, ARM, Atmel, Raspberry Pi).

#### Платформи на основі FPGA

1) Phenix Pro – платформа, побудована на реконфігурованій системі SoC (System on a Chip), розробленій компанією RobSense Tech (Ханчжоу, Китай). Контролер польоту оснащений операційною системою ROS (Linux-based Robot Operating System). Платформа підтримує понад 20 інтерфейсів, включаючи бортові датчики, радар mmWave, лідар, тепловізійну камеру, ultra-vision HD-відео трансивер тощо. Крім того, його апаратне прискорення забезпечує можливості використання комп'ютерного зору та додатки глибокої нейронної мережі. Платформа Phenix Pro працює під управлінням операційної системи реального часу PhenOS на базі FreeRTOS. Схема цієї платформи закрита, однак програмне забезпечення проекту відкрите під ліцензією GNU General Public License v3.

2) Octagonal Pilot on Chip (OcPoC) – платформа OcPoC була розроблена компанією Aerotenna, що заснована в 2015 році (Канзаський університет, США). OcPoC розширює свої можливості введення та виведення за рахунок повністю програмованих контактів PWM, PPM та GPIO для інтеграції з великою кількістю різних датчиків. Платформа також містить багато інших стандартизованих роз'ємів для периферійних пристроїв, таких як GPS, зв'язок камери CSI та SD-карта. Апаратна платформа OcPoC працює на програмній платформі ArduPilot і реалізує одночасну обробку даних з датчиків у реальному часі. Схема цієї платформи закрита.

#### Платформи на основі ARM архітектури

1) PIXHAWK/PX4 – це система керування польотом на основі комп'ютерного бачення, що розроблена Computer Vision and Geometry Lab of ETH Zurich та Autonomous Systems Lab [7]. Платформа складається з контролера PX4-Flight Management Unit (FMU) і PX4-IO, інтегрованих на одній платі з додатковими функціями введення-виведення, пам'яті та іншими функціями. Автори платформи тісно співпрацюють з проектом DroneCode Linux Foundation.

2) PIXHAWK 2 – платформа походить з Pixhawk Hardware Project і створена груповими зусиллями команд PX4 і ArduPilot. Представляє собою невеликий куб, має потрібне резервування IMU та до 3 модулів GPS. Усі підключення (введення-виведення) до плати здійснюються через один роз'єм DF17. Материнська плата має інтерфейс до Intel Edison.

3) Pararaz – це перший і найстаріший проект апаратного та програмного забезпечення дронів з відкритим кодом. Він розробляється в Ecole Nationale de l'Aviation Civil (ENAC) UAV Lab з 2003 року [8]. Проект охоплює системи автопілотування та програмне забезпечення наземних станцій для мультикоптерів/мультироторів, літаків з нерухомим крилом, гелікоптерів та гібридних літаків [9]. У березні 2017 року ENAC Lab випустила новий автопілот під назвою Chimera, який базується на новітньому мікроконтролерному блоці (MCU) STM32F7.

4) CC3D & Atom – це два контролери польоту з однаковими функціями, але різного розміру. Вони розроблені компанією OpenPilot, яка пізніше стала LibrePilot. Контролери польоту CC3D і Atom мають усі типи апаратного забезпечення стабілізації, що працює під керуванням мікропрограми OpenPilot/LibraPilot. За допомогою OpenPilot/LibraPilot їх можна налаштувати для польотів на будь-якому планері – від літака з нерухомим крилом до октокоптера.

#### Платформи на основі Atmel

1) ArduPilot Mega (APM) – це система автопілота на базі Arduino Mega, розроблена спільнотою DIY Drones як оновлення системи керування польотом ArduPilot. Автопілот здатний керувати автономними мультикоптерами, літаками з нерухомим крилом, традиційними гелікоптерами, наземними машинами тощо. На рис. 1 показано блок ArduPilot Mega (APM).



Рис. 1. ArduPilot Mega (APM) [10].

2) FlyMaple – це плата контролера квадрокоптера, заснована на проєкті Maple. Дизайн FlyMaple базується на процесорі ARM у стилі Arduino. FlyMaple призначений для роботи на балансуючих роботах, мобільних платформах, гелікоптерах і квадрокоптерах, що потребують IMU і високопродуктивних контролерів реального часу.

#### Платформи на основі Raspberry Pi

1) Erle-Brain 3 – це відкрита пілотна програма для БПЛА на базі Linux, розроблена Erle Robotics. Платформа поєднує в собі вбудований комп'ютер Linux (Raspberry Pi) і дочірню плату (PXFmini), що містить кілька датчиків, систему введення-виведення та силову електроніку. PXFmini – це відкрита апаратна система автопілота для створення роботів і дронів, призначених для сімейства Raspberry Pi. Система створена на основі технологій Dronescode Foundation.

2) Navio2 – це Raspberry Pi автопілот, що використовує програмну платформу ArduPilot і підтримує всі її функції. Автопілот включає в себе вбудований GPS, компас, гіроскоп, акселерометр та інші сенсори, а також має можливість підключення до електронних контролерів швидкості (ESC). Це робить Navio2 популярним вибором для створення UAV з використанням Raspberry Pi.

Наступним етапом розглянемо різні *програмні платформи* (Software Platforms).

**ArduPilot** – це широко використовуване, повнофункціональне та надійне програмне забезпечення для автопілота. Платформа здатна керувати будь-якою транспортною системою, від звичайних літаків, мультироторів і гелікоптерів до човнів і навіть підводних човнів [10]. Програмне забезпечення спочатку було розроблено для 8-розрядних мікроконтролерів на базі ARM для роботи на власній платі ArduPilot, яка була замінена ArduPilot Mega (APM) і оптимізована для використання з 32-розрядними MCU на основі ARM архітектури. Однак даний автопілот може працювати під Linux, що дозволяє запуск на одноплатних комп'ютерах та ПК.

**Multiwii** – це програмне забезпечення контролера польоту, розроблене для платформ Arduino та засноване на датчиках від Nintendo Wii, але його можна перенести на інші датчики та платформи. Він підтримує від двох до восьми пропелерів (наприклад, трикоптер, квадрокоптер або гексакоптер) [12; 13].

**AutoQuad** – це проєкт, що розробляє апаратне забезпечення з відкритим вихідним кодом на основі ESC і контролери польоту на основі програмного забезпечення з відкритим кодом [14]. Контролер польоту розроблявся протягом більш ніж 6 років [15]. Прошивка написана для мікроконтролера серії STM32F4 з процесором CORTEX M4 і модулем з плаваючою комою (FPU). Він підтримує до 14 BLDC і сумісний з QGroundControl.

**LibrePilot** – проєкт зосереджений на дослідженні та розробці програмного та апаратного забезпечення для використання в різноманітних додатках, включаючи керування та стабілізацію транспортних засобів, безпілотних автономних транспортних засобах та робототехніки [16]. Проєкт був побудований на основі проєкту OpenPilot. Платформа LibrePilot працює на різних платах польотних контролерів із закритим кодом на основі ARM.

**Dronescode Community** – це неприбуткова організація, метою якої є розробка дешевого, надійного та кращого програмного забезпечення [17]. База відкритого вихідного коду включає зв'язок, апаратне та програмне забезпечення, моделювання. Більше 1200 розробників працює над кодом для цього проєкту, напрацювання

застосовуються в багатьох комерційних і безкоштовних продуктах. Dronescape Community надає програмне забезпечення для OSH PIXHAWK, FlyMaple, Erle Brain 2 тощо.

**AuterionOS** – це повноцінна бортова програмна платформа, що складається з контролера польоту та операційної системи, яка працює на комп'ютері місії. Це дозволяє роботам виконувати розширені операції, включаючи певні автономні дії, забезпечуючи безпечне виконання місії, встановлюючи зв'язок з іншим програмним і апаратним забезпеченням. Зв'язок забезпечується за допомогою підключення LTE для передачі даних користувачеві та хмарі у режимі реального часу. Крім того, платформа дозволяє встановлювати сторонні програми для отримання додаткових функцій.

Окрім низькорівневих систем відзначимо також *системи управління високого рівня*, які відповідають за високорівневе керування або навігацію БПЛА з метою досягнення конкретної мети чи завдання. Головні завдання таких систем: керування ресурсами, інтеграція з нижчими рівнями, обробка виняткових ситуацій та ін. Розглянемо коротко їх основні властивості.

**Aerostack** – це програмна система, що допомагає розробникам проектувати та створювати повну архітектуру управління повітряними робототехнічними системами, інтегруючи численні гетерогенні обчислювальні рішення (наприклад, алгоритми комп'ютерного зору, контролери руху, алгоритми планування руху тощо). Aerostack був розроблений з використанням операційної системи ROS [18].

Подальшим розвитком системи Aerostack стала система Aerostack2. Система Aerostack2 спрямована на вирішення проблеми відсутності стандартизації у галузі систем автономного керування, створена на основі проміжного програмного забезпечення ROS2, має модульну архітектуру програмного забезпечення та орієнтацію на керування групами роботів. До переваг можна віднести те, що система є незалежною від платформи, забезпечує створення логічного рівня для визначення завдань, повторне використання компонентів і підсистем для повітряної робототехніки та можливість розробки повної архітектури керування. Авторами було проведено тести в симуляціях та реальних польотах із кількома різнорідними роями. Aerostack2 є системою з відкритим вихідним кодом [27].

**AerialCore** – це повітряна система, створена з використанням ROS Noetic і призначена для виконання повністю на борту БПЛА. Її можна розгорнути на будь-якому багатороторному транспортному засобі, враховуючи, що він оснащений контролером польоту, сумісним з PX4, як у приміщенні, так і на вулиці. Він підтримує експерименти з декількома роботами, використовуючи мережевий зв'язок Nimbro, і забезпечує як швидкий політ, так і надійне керування [19].

**Agilicious** – це спільно розроблена апаратно-програмна система, призначена для автономного та гнучкого польоту на квадродорі. Апаратне забезпечення та вихідний код є повністю відкритими. Система Agilicious підтримує контролери як на основі моделі, так і на основі нейронної мережі. Апаратне забезпечення має високе співвідношення тяги до ваги та крутного моменту до інерції для маневреності, бортові датчики зору, GPU-accelerated засоби обчислень для роботи нейронної мережі у реальному часі та універсальний програмний стек [20].

**Система від KumarRobotics** дозволяє квадродору автономно переміщатися в захищених середовищах, де немає GPS. Система являє собою стек, що складається з набору модулів, які працюють разом, щоб забезпечити швидку автономну навігацію літального робота в невідомому середовищі. Система була розроблена таким чином, що всі вимірювання та обчислення відбуваються на борту робота. Після того, як робота було запущено, не потрібно залучення оператора для навігації до мети [21].

**CrazyChoir** – це набір інструментів ROS 2, який дозволяє користувачам запускати симуляції та експерименти на зграях наноквадродорів Crazyflie [22].

**UAV Abstraction Layer (UAL)** – це рівень програмного забезпечення для абстрагування користувачів безпілотних літальних апаратів від конкретного апаратного забезпечення платформи та інтерфейсів автопілота. Її головна мета – спростити розробку та тестування алгоритмів вищого рівня в повітряній робототехніці шляхом стандартизації та спрощення інтерфейсів з безпілотними літальними апаратами. UAL підтримує роботу з автопілотами PX4 і DJI. Крім того, UAL може бездоганно працювати з симульованими або реальними платформами та надає виклики для виконання стандартних команд, таких як зліт та посадка, а також управління швидкістю [23].

**XTDrone** – це платформа моделювання БПЛА на основі PX4, ROS і Gazebo. XTDrone підтримує мультиротори (включно з квадродорами та гексароторами), нерухомі крила, VTOL та інші безпілотні системи (наприклад, UGV, USV та роботизовану зброю). Після тестування та налагодження на платформі моделювання, алгоритм зручно розгорнути на реальних БПЛА [24].

**RotorS** – це модульна структура моделювання мікроповітряних апаратів (Micro Aerial Vehicle, або MAV). Симулятор був розроблений за модульним принципом, щоб різні контролери та оцінювачі стану могли використовуватися взаємозамінно, тоді як включення нових MAV зводиться до кількох кроків. Необхідні контролери можна адаптувати, просто змінивши файл параметрів. Усі компоненти були розроблені таким чином, щоб бути аналогічними їхнім реальним аналогам. Це дозволяє використовувати ті самі контролери та оцінювачі стану, включаючи їхні параметри, у моделюванні, що й у реальному MAV [25].

**Generalized Autonomy Aviation System (GAAS)** – це програма з відкритим вихідним кодом, розроблена для повністю автономних VTOL і дронів. GAAS забезпечує повністю автономну польотну платформу на основі лідара, релокалізації з використанням HD-карти, планування траєкторії та інших модулів для повітряних суден. На відміну від технології автопілота, яка раніше була доступна лише для безпілотних літальних апаратів споживчого класу, GAAS спрямована на надійний повністю автономний політ для людей. Весь фреймворк слабо пов'язаний, тому є можливість налаштувати власні модулі та легко додати їх до GAAS [26].

У таблиці 1 наведено порівняння основних характеристик розглянутих систем управління високого рівня.

Таблиця 1

### Порівняння систем управління високого рівня

Система	Open Source	Модульність	OC	Multi-frame	Rate output	Багато-агентна	Мульти-платформенна	Плагіно-орієнтована
Aerostack2	+	+	ROS 2	+	+	+	+	+
AerialCore	+	+	ROS	+	+	+	-	+
Agilicious	+	+	ROS		+	-	-	-
KumarRobotics	+	-	ROS		+	-	+	-
CrazyChoir	+	-	ROS 2		+	+	-	-
UAL	+	-	ROS	+	-	-	+	-
XTDrone	+	+	ROS		+	-	-	-
RotorS	+	+	ROS		+	-	-	-
GAAS	+	+	ROS		-	-	-	-

З проаналізованих систем керування високого рівня було виявлено, що (1) усі вони є відкритими; (2) шість мають модульну структуру; (3) лише дві із систем, CrazyChoir та Aerostack2, використовують ROS 2 у якості проміжного програмного забезпечення; (4) три системи підтримують дрони різних конструкцій (Multiframe); (5) сім систем підтримують акроконтроль повітряного судна (Rate output); (6) три системи мають багатоагентний підхід; (7) три системи підтримують більше однієї іншої польотної платформи; і (8) лише дві системи мають архітектуру, орієнтовану на використання плагінів.

### Висновки

Отже, було розглянуто основні логічні та конструктивні блоки для побудови БПЛА, описали складові компоненти системи автопілота, розглянули 16 різних апаратних і програмних платформ керування польотом, що можна використовувати для академічних досліджень. Дослідили можливості обробки даних апаратними засобами, композицію датчиків, інтерфейси, порівняли різні функції платформ БПЛА. Також виконали огляд дев'яти систем управління високого рівня та привели порівняння їх основних характеристик. Важливо зазначити, що під час дослідження було виявлено брак документації щодо енергоспоживання розглянутого апаратного забезпечення.

Перспективами подальших досліджень можуть бути у проведенні огляду інших програмних та технічних засобів, а також нещодавно створених проектів та платформ, велику кількість яких було виявлено під час дослідження, та аналіз енергоспоживання комбінацій програмно-апаратних платформ разом із розрахунками відносної вартості апаратних платформ.

### Список використаної літератури

- Chengqi X., Cen Q., Yan Z. Design and research of human-computer interaction interface in autopilot system of aircrafts. *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*. 2009. С. 1498–1501. <https://doi.org/10.1109/CAIDCD.2009.5374997>.
- Sabikan S., Nawawi S. W. Open-Source Project (OSPs) Platform for Outdoor Quadcopter. *Journal of Advanced Research Design*. 2016. Т. 24, № 1.
- Groves P. D. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition. Artech House, 2013.
- D. Perez et al. A ground control station for a multi-UAV surveillance system: design and validation in field experiments. *Journal of Intelligent & Robotic Systems*. 2013. Т. 69. С. 119–130. <https://doi.org/10.1007/s10846-012-9759-5>.
- L. Meier et al. PIXHAWK: A system for autonomous flight using onboard computer vision. *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 9–13 May 2011. 2011. Pp. 2992-2997. <https://doi.org/10.1109/icra.2011.5980229>.
- Gabriel D. L., Meyer J., du Plessis F. Brushless DC motor characterisation and selection for a fixed wing UAV. *AFRICON 2011*, Victoria Falls, Livingstone, Zambia, 13–15 sept. 2011 p. 2011. <https://doi.org/10.1109/afcon.2011.6072087>.
- L. Meier et al. PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*. 2012. Т. 33, № 1-2. Pp. 21–39. <https://doi.org/10.1007/s10514-012-9281-4>.

8. P. Brisset et al. The paparazzi solution. 2006. MAV 2006, 2nd US-European competition and workshop on micro air vehicles.
9. E. Baskaya et al. Flexible open architecture for UASs integration into the airspace: Paparazzi autopilot system. 2016 IEEE/AIAA 35th *Digital Avionics Systems Conference (DASC)*, Sacramento, CA, USA, 25–29 sept. 2016. 2016. <https://doi.org/10.1109/dasc.2016.7778016>.
10. Ardupilot Mega. *Ardupilot*. URL: <https://www.ardupilot.co.uk> (date of access: 23.04.2024).
11. Use-Cases and Applications – Copter documentation. *ArduPilot*. URL: <https://ardupilot.org/copter/docs/common-use-cases-and-applications.html> (date of access: 23.04.2024).
12. Google Code Archive – MultiWii flight controller source code. *Google Code*. URL: <https://code.google.com/archive/p/multiwii> (date of access: 23.04.2024).
13. MultiWii home web page. *MultiWii*. URL: <http://www.multiwii.com> (date of access: 23.04.2024).
14. AutoQuad Flight Control Firmware. *GitHub*. URL: [https://github.com/mpaperno/aq\\_flight\\_control](https://github.com/mpaperno/aq_flight_control) (дата звернення: 23.04.2024).
15. AutoQuad | Autonomous Multi Rotor Vehicle Controller. *AutoQuad*. URL: <http://autoquad.org> (date of access: 23.04.2024).
16. The LibrePilot open source project. *LibrePilot*. URL: <https://www.librepilot.org> (date of access: 23.04.2024).
17. The Dronecode Foundation. *Dronecode Foundation*. URL: <https://www.dronecode.org/> (дата звернення: 23.04.2024).
18. J. L. Sanchez-Lopez et al. AEROSTACK: An architecture and open-source software framework for aerial robotics. 2016 *International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, 7–10 june 2016. 2016. <https://doi.org/10.1109/icuas.2016.7502591>.
19. T. Baca et al. The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*. 2021. T. 102, № 1. <https://doi.org/10.1007/s10846-021-01383-5>.
20. P. Foehn et al. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Science Robotics*. 2022. T. 7, № 67. <https://doi.org/10.1126/scirobotics.abl6259>.
21. K. Mohta et al. Fast, autonomous flight in GPS-denied and cluttered environments. *Journal of Field Robotics*. 2017. T. 35, № 1. Pp. 101–120. <https://doi.org/10.1002/rob.21774>.
22. Pichierri L., Testa A., Notarstefano G. CrazyChoir: Flying Swarms of Crazyflie Quadrotors in ROS 2. *IEEE Robotics and Automation Letters*. 2023. Pp. 1–8. <https://doi.org/10.1109/lra.2023.3286814>.
23. F. Real et al. Unmanned aerial vehicle abstraction layer: An abstraction layer to operate unmanned aerial vehicles. *International Journal of Advanced Robotic Systems*. 2020. T. 17, № 4. Article no 172988142092501. <https://doi.org/10.1177/1729881420925011>.
24. K. Xiao et al. XTDrone: A Customizable Multi-rotor UAVs Simulation Platform. 2020 *4th International Conference on Robotics and Automation Sciences (ICRAS)*, Wuhan, China, 12–14 june 2020. 2020. <https://doi.org/10.1109/icras49812.2020.9134922>.
25. F. Furrer et al. RotorS—A Modular Gazebo MAV Simulator. *Studies in Computational Intelligence*. Cham, 2016. Pp. 595–625. [https://doi.org/10.1007/978-3-319-26054-9\\_23](https://doi.org/10.1007/978-3-319-26054-9_23).
26. GitHub – generalized-intelligence/GAAS: GAAS is an open-source program designed for fully autonomous VTOL (a.k.a flying cars) and drones. *GitHub*. URL: <https://github.com/generalized-intelligence/GAAS> (date of access: 23.04.2024).
27. M. Fernandez-Cortizas et al. Aerostack2: A Software Framework for Developing Multi-robot Aerial Systems. 2023.