

РЕАЛІЗАЦІЯ КАНАЛУ ЗВ'ЯЗКУ МІЖ RASPBERRY PI ТА ANDROID ДЛЯ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ У РЕАЛЬНОМУ ЧАСІ

Тези присвячено аналізу інтеграції мобільних і вбудованих платформ для створення апаратно-програмного комплексу реального часу детектування об'єктів і потенційних загроз. Основна увага зосереджена на забезпеченні стабільного та енергоефективного каналу зв'язку між компонентами системи.

Ключові слова: *роботизована технічна система, FPV-дрон, Raspberry Pi 4, фреймворк TensorFlow Lite, детектування об'єктів та перешкод.*

З розвитком цифрових технологій, передусім у сфері відеопередачі, а також завдяки мініатюризації апаратних компонентів стало можливим створювати легкі мобільні комплекси з підтримкою високоякісного відеозв'язку, значною дальністю польоту та підвищеною стабільністю каналу. Це сприяло активному поширенню FPV-дронів у цивільному секторі – зокрема для інспекцій промислових об'єктів та інфраструктури, сільськогосподарського й екологічного моніторингу, рятувальних операцій, а також у воєнній сфері. Дедалі частіше роботизовані технічні системи (РТС) використовуються в розважальних цілях (напр., кінозйомки та телевізійного виробництва), проте навіть в такому випадку потребують спеціального налаштування [1]. Все більше стає помітним втручання РТС у сфери, де людська діяльність може бути неефективною або надто тривалою. Актуальність детектування перешкод спричинена необхідністю зменшення людського фактору під час процесів ідентифікації об'єктів та уникнення можливих загроз апаратної складової. Основним завданням є забезпечення безперебійного зв'язку між мікроконтролером та оператором, і як результат – фотозвіт з вказаним рівнем безпеки навколишнього середовища.

Необхідність у РТС для детектування пояснюється тим, що під час польоту FPV-дрон може зіштовхнутися з непередбачуваними об'єктами – як природними (дерева, птахи), так і техногенними (лінії електропередач, будівельні конструкції), які важко заздалегідь виявити візуально. У таких умовах автоматичне фіксування та аналіз перешкод дозволяє своєчасно змінити маршрут польоту або попередити оператора про потенційну небезпеку. Це особливо важливо в сценаріях автономного або напівавтономного управління, де зволікання може спричинити втрату цінного обладнання або створити небезпеку для людей.

При виборі платформи враховувалась потреба в моніторингу масштабних територій та швидкому реагуванні на об'єкт. Сферою застосування виступає FPV-дрон як окрема рухома багатоядерна комп'ютерна система з можливістю розміщення на ній елементів детектування та обробки інформації з подальшим передаванням на мобільний пристрій. DJI Phantom 4 є оптимальним вибором, адже забезпечує стабільність польоту та дальність польоту (до 5 км), що є достатнім для моніторингу територій середнього масштабу [2]. Батарея розміщується під корпусом, що покращує аеродинамічні властивості дрона. Орієнтована вантажопідйомність становить 200–300 гр., чого достатньо для перевезення апаратних компонентів (рис. 1). Саме кріплення має бути збалансованим, щоб не змішувати центр мас, а габарити компонентів дозволяють їх розташування відносно центру.

Основним апаратним компонентом є мікроконтролер Raspberry Pi 4 Model B, який виконує обробку даних. Його потужності достатньо для реалізації детектування об'єктів за допомогою фреймворку TensorFlow Lite (TFLite) по фото або відео з низькою частотою кадрів (FPS). Окрім цього, він забезпечує підтримку камер, периферійних пристроїв, а також оснащений WiFi-модулем, що дозволяє передавати дані на Android без додаткових затримок. Центральний процесор обробляє вхідні дані (фото- або відеопотік) та виконує алгоритми машинного навчання з використанням відповідного фреймворку. Значний обсяг оперативної пам'яті гарантує підтримку стабільної роботи операційної системи, службових процесів та дає змогу обробки проміжних результатів у межах моделі глибинного навчання.



Рис. 1. Підбір компонентів з урахуванням їх ваги

Графічний процесор прискорює обробку графічних даних, зокрема у декодуванні відеопотоку, який передається до основного обчислювального модуля. Wi-Fi та Ethernet дозволяють здійснювати двосторонню комунікацію із зовнішніми пристроями, зокрема Android-системами, для передачі інформації про виявлені об'єкти чи отримання керуючих сигналів.

Оскільки запуск фреймворку потребує значних обчислювальних ресурсів, плата не застрахована від перегріву. Отже потрібно охолоджувати центральний процесор для забезпечення тривалої роботи мікроконтролера. Обрано алюмінієвий корпус зі вбудованими двома кулерами для запобігання перегріву та ушкодженням від можливих перешкод на шляху РТС.

Архітектура програмної частини комплексу є багаторівневою. Вона складається з блоку локальної обробки даних на Raspberry Pi 4, блоку взаємодії з камерою та мобільного застосунку на Android.

Вибір фреймворку TensorFlow Lite обумовлений використанням мікроконтролера, ресурси якого обмежені. Їх недостатньо для використання в комплексі з повнофункціональною версією TensorFlow. TFLite є оптимізованою версією TensorFlow, розробленою спеціально для запуску попередньо навчених моделей на мобільних та вбудованих пристроях із обмеженими ресурсами, таких як Raspberry Pi або мобільні платформи. Окремо слід зазначити можливість інтеграції бібліотеки TFLite в Android-застосунок, що забезпечує швидке виконання алгоритмів машинного навчання з низькою затримкою та енергоефективністю. TFLite працює зі спеціалізованими моделями MobileNet, YOLO або SSD та дозволяє розпізнавати та детектувати об'єкти в реальному часі. Обов'язковим є підключення бібліотеки OpenCV [3], для використання функцій захоплення відео з камери та обробки зображень. Можливі конфлікти під час встановлення бібліотек вирішуються за допомогою virtualenv – інструменту для створення ізолюваного Python-середовища. Операційна система (ОС) – Raspberry Pi OS забезпечує повну сумісність із Python, OpenCV та TensorFlow Lite. ОС Android має бути не нижче 9-ї версії, адже вона дозволяє використання протоколів HTTP та WebSocket. Оброблені зображення передаються у форматі JSON-повідомлень.

На рис. 2 зображено процес детектування перешкод. Приклад демонструє ефективність фреймворку TensorFlow для фіксування та розпізнавання об'єктів.



Рис. 2. Приклад детектування перешкод (дерев) [4]

На етапі обробки кожен кадр отримує пріоритет залежно від кольору:

- зелений колір вказує на відсутність небезпеки та можливість безперешкодного перетину сектору;
- жовтий колір вказує на наявність ймовірної небезпеки та можливість пошкодження дрону;
- червоний колір повідомляє про небезпечну ділянку та необхідність негайно покинути територію через загрозу втрати РТС.

На рис. 3 наведено результат детектування перешкод на прикладі густини лісу: чим щільніше розташовані дерева, тим важче продовжувати політ.



Рис. 3. Приклад результату обробки

Таким чином, оператор отримує оброблене зображення та на основі нього має можливість координувати траєкторію FPV-дрону відповідно до прийнятого рішення. Цей підхід дозволяє пришвидшити процес реагування на перешкоди, а в разі помилки програмного забезпечення керування переходить напругу до оператора. Зважаючи на це, доцільно підсумувати загальний процес передачі інформації та детектування небезпеки (рис. 4).

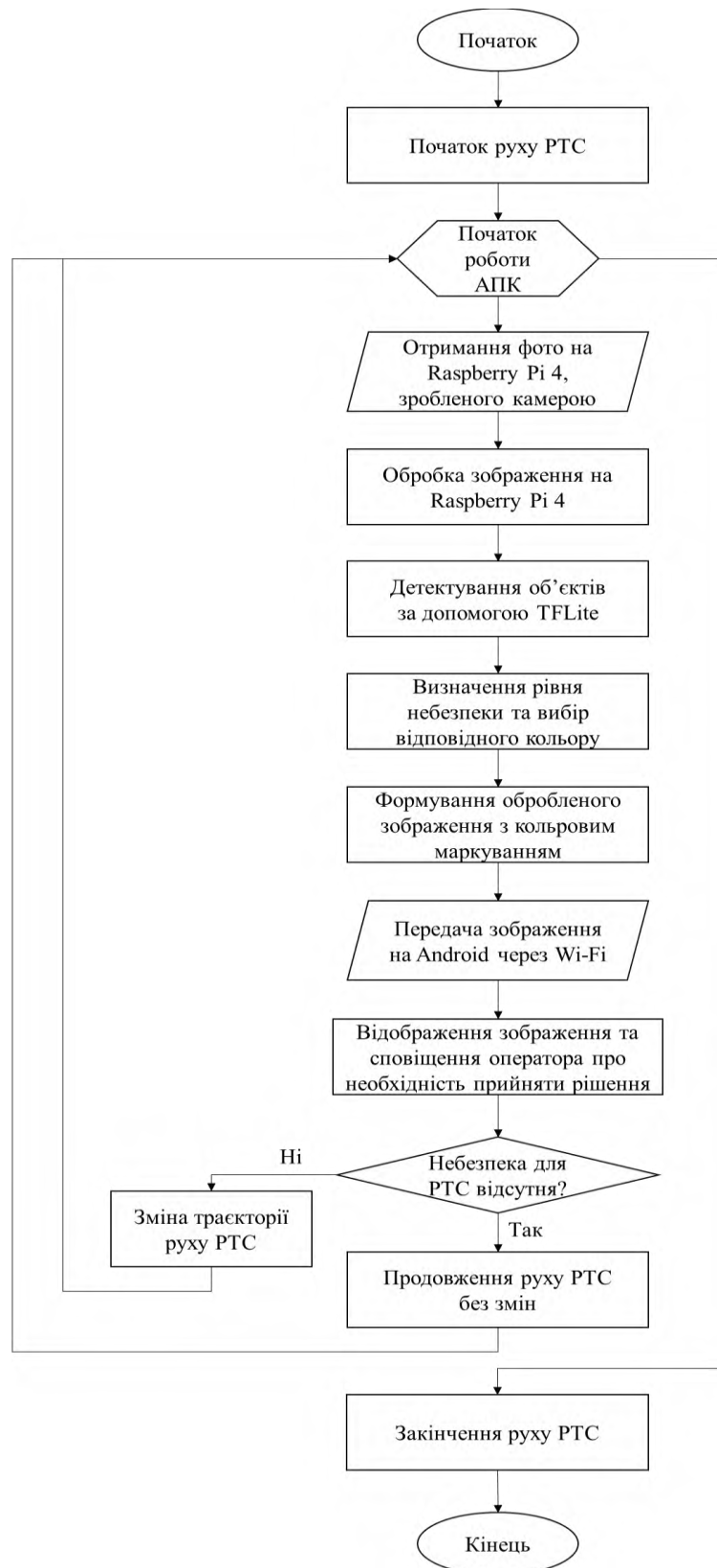


Рис. 4. Алгоритм роботи апаратно-програмного комплексу

Запропонований підхід ґрунтується на поєднанні мобільних обчислювальних платформ та сучасних алгоритмів машинного навчання. Це рішення забезпечує високу швидкість обробки даних (фото- та відеопотоку) та стабільний зв'язок між FPV-дроном і оператором. Використання Raspberry Pi 4 у поєднанні з Android-пристроями через бездротовий канал дозволяє створити компактний, гнучкий й енергоефективний комплекс, здатний адаптуватися до різних умов. Додаткові компоненти, такі як кулери та алюмінієвий корпус, підвищують функціональність мікроконтролера, забезпечуючи його надійну роботу при будь-яких умовах навколишнього середовища. Застосування фреймворку TensorFlow Lite дозволяє ефективно виконувати завдання детектування без перевантаження апаратних ресурсів.

Реалізація такого рішення не лише відповідає сучасним вимогам до безпеки та оперативності, а й відкриває нові перспективи для розвитку автономних систем контролю з використанням роботизованих технічних систем у цивільному та військовому секторах.

References

1. Siddiqui W. Design and Fabrication of FPV Racing Drone. *Journal of Mechanical and Construction Engineering (JMCE)*. 2024. Vol. 4, No. 1. P. 1–8. DOI: 10.54060/a2zjournals.jmce.38.
2. Support for Phantom 4. Retrieved from: <https://www.dji.com/support/product/phantom-4> (Last accessed: 17.05.2025).
3. How to Run TensorFlow Lite Object Detection Models on the Raspberry Pi (with Optional Coral USB Accelerator). Retrieved from: https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/deploy_guides/Raspberry_Pi_Guide.md (Last accessed: 12.05.2025).
4. How to detect a single tree from a drone imagery of a dense forest? Nothing is simpler than custom object detection using TensorFlow. Retrieved from: <https://blog.skygate.io/how-to-detect-a-single-tree-from-a-drone-imagery-of-a-dense-forest-de190f64cdb7> (Last accessed: 17.05.2025)

DOI 10.34132/mspc2025.01.06.08

*Illia Bondarenko,
Kateryna Obukhova*

IMPLEMENTATION OF A COMMUNICATION CHANNEL BETWEEN RASPBERRY PI AND ANDROID FOR REAL-TIME OBJECT DETECTION

The paper are devoted to the integration of mobile and embedded platforms for the development of a hardware-software complex for objects and threats detection in real-time. Special attention is given to ensuring a stable and energy-efficient communication channel between system components.

Keywords: robotic technical system, FPV drone, Raspberry Pi 4, TensorFlow Lite framework, object and obstacle detection.

Відомості про авторів / Information about the Authors

Ілля Бондаренко, бакалаврант кафедри комп'ютерної інженерії Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: ilyabondarenko66@gmail.com, orcid: <https://orcid.org/0009-0000-9345-0286>.

Illia Bondarenko, BSc Student of the Computer Engineering Department at the Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. Email: ilyabondarenko66@gmail.com, orcid: <https://orcid.org/0009-0000-9345-0286>.

Катерина Обухова, ст. викладач кафедри комп'ютерної інженерії Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: kateryna.obukhova@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-8793-7055>.

Kateryna Obukhova, Senior Lecturer of the Computer Engineering Department at the Petro Mohyla Black Sea National University, Mykolaiv, Ukraine. E-mail: kateryna.obukhova@chmnu.edu.ua, orcid: <https://orcid.org/0000-0001-8793-7055>.

© I. Бондаренко, К. Обухова
20.05.2025..

Стаття отримана редакцією