

Д. А. Редчин¹✉, Д. С. Строганов¹, А. А. Шаронов¹

Разработка программного обеспечения для автономного комплекса управления и посадки беспилотного летательного аппарата при дистанционном видеомониторинге инженерных и природных объектов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: redchindaniil@gmail.com

Аннотация. В статье рассмотрен процесс разработки программного обеспечения для автономного комплекса управления и посадки беспилотных летательных аппаратов (БПЛА), предназначенных для дистанционного видеомониторинга инженерных и природных объектов. Описаны применяемые технологии, архитектура системы, подходы к реализации взаимодействия компонентов через брокер сообщений RabbitMQ, а также возможности интеграции бортовых компьютеров на основе платформы Linux (например, Raspberry Pi) с DJI Onboard SDK. Разрабатываются методы симуляционного тестирования разработанного ПО, позволяющие осуществлять предварительную проверку функционала без реальных полетов. Разработанная система ориентирована на эффективное управление группой беспилотников и безопасное выполнение автономных посадок.

Ключевые слова: беспилотный летательный аппарат, DJI Onboard SDK, видеомониторинг, RabbitMQ, автономная посадка, симулятор дронов

D. A. Redchin¹✉, D. S. Stroganov¹, A. A. Sharapov¹

Development of software for an autonomous control and landing complex for an unmanned aerial vehicle during remote video monitoring of engineering and natural objects

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: redchindaniil@gmail.com

Abstract. The article discusses the software development process for an autonomous control and landing system for unmanned aerial vehicles (UAVs), intended for remote video monitoring of engineering and natural objects. It describes the technologies used, system architecture, methods of implementing component interaction via RabbitMQ message broker, and integration possibilities with onboard computers based on Linux platforms (e.g., Raspberry Pi) using DJI Onboard SDK. Simulation testing methods of the developed software are also presented, enabling preliminary verification of functionality without actual flights. The developed system is aimed at efficient management of UAV groups and safe execution of autonomous landings.

Keywords: unmanned aerial vehicle, DJI Onboard SDK, video monitoring, RabbitMQ, autonomous landing, drone simulator

Введение

Современные задачи инженерного и природоохранного мониторинга требуют применения беспилотных летательных аппаратов (БПЛА) [6], способных

автономно выполнять задачи видеосъемки и анализа данных без постоянного вмешательства оператора. Примерами таких систем служат программное обеспечение SmartDrones [1], используемое для автономного видеомониторинга с применением нейросетевых технологий на мобильных устройствах, а также DJI Edge [2] – система, управляющая множеством дронов с единой стационарной точки.

Однако существующие решения зачастую ограничены в функциональности автономной посадки и требуют доработки для работы с наземными комплексами управления. В рамках данной статьи рассматривается разработка системы, объединяющей функциональность таких решений, с расширением возможностей по автономной посадке и централизованному управлению.

Методы и материалы

Архитектура системы разделена на две части – веб приложение для алгоритма формирования полетного задания и предоставлением интерфейса управления оператору, и бэкенд часть для взаимодействия с веб приложением и напрямую с док-станциями и дронами. Системы общаются между собой через брокер сообщений RabbitMQ. Взаимодействие с картой реализовано с помощью open source библиотекой Leaflet [9]. Файловое хранилище для загрузки в него медиа из дронов и просмотра оператором – Minio [10]. Для веб-приложения был выбран интерфейс, предоставляющий оператору инструмент для задания зон мониторинга и управления дронами (рис. 1).

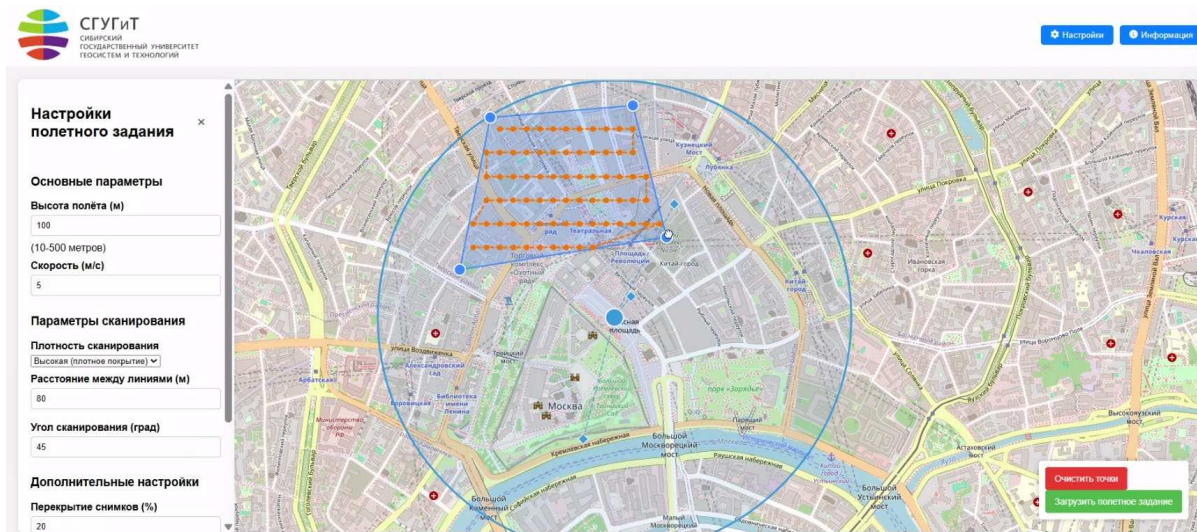


Рис. 1. Вид прототипа интерфейса оператора

Полетное задание формируется с помощью настроек на левой панели интерфейса. Сейчас представлены базовые настройки для задания маршрута дрона, такие как высота полета и скорость. И более продвинутые параметры сканирования, отвечающие за плотность покрытия выделенной зоны пролетающим дроном. Так на рисунке 1 выбрана высокая плотность сканирования и расстояние

между линиями 80 метров. Эти настройки корректируют построение маршрута из оранжевых точек на карте, таким образом, чтобы, пролетая по точкам, дрон покрывал 90 % выделенной синей области, при этом вертикальное расстояние между линиями корректируется второй настройкой и составляет 80 метров.

Также при нажатии на док-станцию появится возможность просмотреть привязанные к ней дроны и медиа данные (рис. 2).

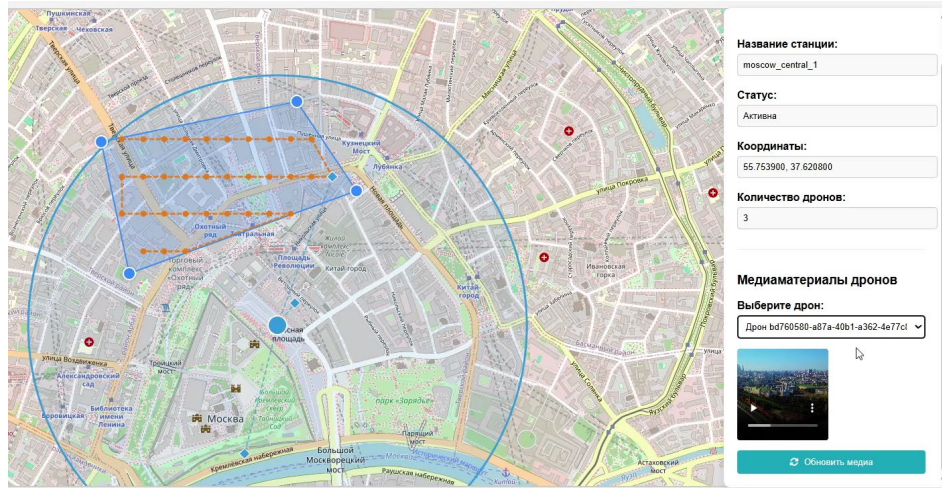


Рис. 2. Вкладка просмотра текущей док станции и дронов

Для реализации бэкенда выбрана экосистема Rust, обеспечивающая высокую надежность и производительность. Взаимодействие компонентов системы реализовано через gRPC и RabbitMQ (возможно добавление новых протоколов в процессе разработки). Бортовые компьютеры дронов будут работать на базе платформы Linux (к примеру, платы Raspberry Pi), подключаться к сети через интернет или специальные пультах дальнего действия, и использовать DJI Onboard SDK для управления дроном и сбора данных.

В качестве примера приведен основной gRPC [8] protobuf сервис «DroneStreamRequest» (рис 3). Бортовой компьютер дрона (далее, упрощенно, - «дрон») устанавливает потоковое общение по gRPC с управляющей единицей и обменивается заранее заданным набором сообщений.

```

163 }
164
165 // Bi-directional streaming message types for drone <=> controller communication
166 message DroneStreamRequest {
167   oneof payload {
168     DroneAuthRequest auth = 1; // First message from drone should be authentication
169     Telemetry telemetry = 2; // Telemetry update from drone
170     ImageData image = 3; // Camera image frame from drone (could be sent periodically or on request)
171     MissionStatus mission_update = 4; // Mission status update from drone (e.g., mission completed or progress)
172   }
173 }
174
175 // Service that drones use to connect to the Controller
176 service DroneService {
177   // Drones connect and maintain a bidirectional stream with the controller.
178   // The drone sends a stream of DroneStreamRequest messages (starting with auth, followed by telemetry/images),
179   // and the server responds with a stream of ControllerStreamResponse messages (auth response and commands).
180   rpc ConnectDrone(stream DroneStreamRequest) returns (stream ControllerStreamResponse);
181 }
182
183
184
  
```

Рис 3. Технические детали протокола общения

Также в разработке симулятор дронов, который позволяет проводить предварительное тестирование системы без реальных полетов. Симулятор упрощенно моделирует реальные физические характеристики устройств и использует такие же протоколы для взаимодействия с основным приложением, как и настоящее устройство.

Результаты

В ходе разработки создано программное обеспечение, обеспечивающее:

1. Задание и контроль автономных полетных заданий (веб-приложение).
2. Визуализацию и мониторинг состояния дронов в реальном времени (веб-приложение).
3. Возможность просматривать потокового просмотра медиа с дрона (веб-приложение).
4. Учет, управление, авторизация дронов («бэкенд»).
5. Симуляцию полетов для тестирования (инструменты тестирования).

Разработка еще не завершена, «веб-приложение» и «бэкенд» на данный момент еще интегрируются друг с другом. Также производится активное тестирование системы во время разработки.

Обсуждение

Выбранный стек технологий обеспечивает легкость и надежность разработки. Часть ПО, с повышенными требованиями к надежности – бэкенд и ПО бортовых компьютеров (в процессе разработки) пишется на Rust, который предоставляет инструменты, обеспечивающие надежность, безопасность, и производительность ПО. В качестве файлового хранилища используется Minio, обеспечивающее безопасность медиа данных. Использование симулятора позволяет снизить риски и затраты на этапах разработки и тестирования, и повысить скорость разработки. Реализация программного комплекса на основе Linux и DJI SDK делает систему доступной для широкого круга пользователей и разработчиков.

Заключение

Разработанный программный комплекс обеспечит эффективное управление и посадку БПЛА [7] при автономном видеомониторинге. Дальнейшие исследования планируется направить на совершенствование алгоритмов автономной посадки и интеграцию продвинутых методов компьютерного зрения и анализа видеоданных на борту беспилотных аппаратов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СГУГиТ и МЧС НСО испытали дроны с новым ПО SmartDrones // Пресс-служба СГУГиТ. URL: <https://sgugit.ru/news/sgugit-i-mchs-nso-ispytali-drony-s-novym-po-smartdrones/> (дата обращения: 21.04.2025).
2. DJI Dock – Автоматизированные ангары для беспилотников // DJI Enterprise. URL: <https://enterprise.dji.com/ru/dock> (дата обращения: 21.04.2025).

3. DJI Onboard SDK: Main Page // DJI Developer. URL: <https://developer.dji.com/onboard-api-reference/index.html> (дата обращения: 21.04.2025).
4. DJI Onboard SDK: File List // DJI Developer. URL: <https://developer.dji.com/onboard-api-reference/files.html> (дата обращения: 21.04.2025).
5. RabbitMQ: что это такое и как работает // Skillfactory Media. URL: <https://blog.skillfactory.ru/rabbitmq-cto-eto-takoe-i-kak-rabotaet/> (дата обращения: 21.04.2025).
6. Автономные беспилотные летательные аппараты в точных системах агропроизводства // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/avtonomnye-bespilotnye-letatelnye-apparaty-v-tochnyh-sistemah-agroproduzvodstva> (дата обращения: 21.04.2025).
7. Беспилотный летательный аппарат // Википедия. URL: https://ru.wikipedia.org/wiki/Беспилотный_летательный_аппарат (дата обращения: 21.04.2025).
8. gRPC // Википедия. URL: <https://ru.wikipedia.org/wiki/GRPC> (дата обращения: 21.04.2025).
9. Leaflet // Документация. URL: <https://leafletjs.com/reference.html> (дата обращения: 21.04.2025).
10. MiniO // Документация. URL: <https://min.io/docs/minio/kubernetes/upstream/index.html> (дата обращения: 21.04.2025).
11. Программное обеспечение для беспилотников // Радар ммс. URL: <https://radar-mms.com/product/bespilotnye-aviatsionnye-sistemy/programmnoe-obespechenie/> (дата обращения: 21.04.2025).
12. DJI FlightHub: система управления дронами // AEROMOTUS. URL: <https://aeromotus.ru/programmnoe-obespechenie-dlya-bpla/> (дата обращения: 21.04.2025).
13. UgCS – планирование и контроль съемки с БПЛА // Geodevice. URL: <https://geodevice.kz/software/gpr-software/ugcs/> (дата обращения: 21.04.2025).
14. ArcGIS Reality: программное обеспечение для картографирования данных БПЛА // Esri. URL: <https://www.esri.com/ru-ru/arcgis/products/arcgis-reality/platform/drone-mapping> (дата обращения: 21.04.2025).
15. Комплекс управления БПЛА для дистанционного зондирования Земли // UAV Siberia. URL: <https://uav-siberia.com/news/kompleks-upravleniya-bespilotnymi-letatelnyimi-apparatami-dlya-distantsionnogo-zondirovaniya-zemli/> (дата обращения: 21.04.2025).
16. Система автоматической посадки (AMLS) // Clover. URL: <https://clover.coex.tech/ru/amls.html> (дата обращения: 21.04.2025).
17. Автономный дронопорт: непрерывные полеты БПЛА // UAVProf. URL: <https://uav-prof.com/solutions/avtonomnyi-dronoport/> (дата обращения: 21.04.2025).
18. Интеллектуальная система управления автономным БПЛА // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/intellektualnaya-sistema-upravleniya-avtonomnym-bespilotnym-letatelnyim-apparatom> (дата обращения: 21.04.2025).
19. Модель системы автоматического управления беспилотным летательным аппаратом // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/model-sistemy-avtomaticheskogo-upravleniya-bespilotnym-letatelnyim-apparatom> (дата обращения: 21.04.2025).
20. Системы управления беспилотными летательными аппаратами // Военмех. URL: <https://voenmeh.ru/product/sistemy-upravleniya-bespilotnymi-letatelnyimi-apparatami/> (дата обращения: 21.04.2025).

© Д. А. Редчин, Д. С. Строганов, А. А. Шаранов, 2025