

Техническое задание на разработку автономного беспилотного летательного аппарата (БПЛА) для мониторинга пожарной безопасности в крупногабаритных помещениях (ангарах)

1. Основание для разработки:

ТЗ составляется на основании постановки задачи от предприятия тяжелой промышленности о необходимости создания автономного БПЛА (или роя БПЛА) для непрерывного и круглосуточного мониторинга пожарной безопасности в крупногабаритных помещениях (ангарах) заказчика с применением для детекции очага возгорания (пламени, дыма) компьютерного зрения.

- Заказчик: АО “Петрозаводскмаш”.
- Исполнитель: АО Центр пожарной робототехники “ЭФЭР”.
- Задачи, лежащие в основе разработки:
 1. Повышение эффективности и скорости обнаружения очагов возгорания (пожаров) и задымления в ангарах, например, таких, как: промышленные и товарные склады, промышленные цеха, самолетные и железнодорожные ангары и т.п.
 2. Автоматизация процесса патрулирования для сокращения затрат на человеческие ресурсы и обеспечение непрерывного и круглосуточного мониторинга с применением роя БПЛА.
 3. Автоматизация процесса детектирования пожара (пламени, дыма) с помощью компьютерного зрения.

4. Автоматическая передача сигнала тревоги в пожарную сеть предприятия, а также видеофиксация и передача локальных координат пожара.
5. Снижение рисков для жизни и здоровья персонала за счет исключения его нахождения в потенциально опасной зоне при осмотре ангаров.
6. Автоматизация процесса подзарядки аккумуляторных батарей БПЛА.

2. Назначение и цели:

- Назначение ([для чего ведется разработка](#)): для обеспечения ангаров заказчика автономным роем БПЛА, способных осуществлять непрерывный круглосуточный мониторинг пожарной безопасности при помощи компьютерного зрения в замкнутом пространстве ангаров с автоматической передачей сигнала тревоги в случае обнаружения пожара в пожарную сеть предприятия, а также видеофиксацией и передачей локальных координат пожара.
- Цели ([какие проблемы решает](#)):
 1. Обеспечение непрерывного и круглосуточного 3D-мониторинга внутреннего пространства ангаров по заданному маршруту.
 2. Необходимость быстрого обнаружения очагов возгорания и задымления в труднодоступных и обширных зонах ангаров.
 3. Невозможность обнаружения или задержки в обнаружении пожара традиционными стационарными датчиками из-за большой высоты потолков и объема помещения, наличия крупногабаритных препятствий или скрытых зон для визуального обнаружения пожара.
 4. Обеспечение визуального подтверждения и точного позиционирования пожара (пламени, дыма) для службы быстрого реагирования с возможностью передачи сигнала тревоги в пожарную сеть предприятия.
 5. Снижение производственного травматизма и смертности, связанных с пожарной безопасностью на производстве.
- Что будет автоматизировано:
 1. Процесс патрулирования внутреннего пространства ангаров роем БПЛА.

2. Обнаружение признаков пожара (пламя, дым) в видеопотоке камеры (тепловизора) с помощью компьютерного зрения.
3. Передача сигнала тревоги в пожарную сеть предприятия, а также видеофиксация и передача локальных координат пожара.
4. Процесс подзарядки аккумуляторных батарей БПЛА.

3. Требование к функциональной части:

- Определение задачи СВ:

По кадрам видеопотока необходимо распознавать 2 класса объектов: пламя и дым с **уверенностью > 0.55**. Для задачи детекции необходимо использовать модель YOLO версии 11 нано - YOLO 11n. Для этого потребуется использование компаньон-вычислителя на борту - одноплатного компьютера, например, Raspberry Pi 5. Для ускорения процесса распознавания нейросетью, предполагается применить модуль EDGE TPU (тензорный процессор) в составе вычислителя. Обученная модель YOLO 11n должна быть конвертирована в формат TensorFlow Lite для работы с EDGE TPU. Первичная (начальная) конвертация и обновление моделей должны выполняться средствами исполнителя.

- Необходимые протоколы:

1. MAVLink – для связи между полетным контроллером, наземной станцией управления и бортовыми системами БПЛА.
2. Wi-Fi (IEEE 802.11) – для связи с пожарной сетью предприятия: HTTP - для реализации REST API, RTSP - для передачи видео.
3. Резервный: протоколы RC-передатчиков (например, CRSF, SBUS).

- Определение режимов полета (режимов работы):

1. Начало полета: ARM и взлет на заданную высоту с зарядной станции (базы) по команде “ВЗЛЕТ” от оператора (или от автоматизированной системы управления роем БПЛА) с последующим переходом в режим патрулирования.
2. Патрулирование (основной): автономный циклический полет по заранее заданному маршруту.

3. Режим обнаружения пожара: автоматическая активация при обнаружении пламени/дыма, выход из режима патрулирования, **автономное приближение к детектируемой цели**, стабилизация, наведение камер на цель, передача сообщения об обнаружении пожара в пожарную сеть предприятия и на наземную станцию управления оператору, ожидание команды "ОТБОЙ ТРЕВОГИ" для перехода в основной режим патрулирования.
4. Режим возврата на базу: автоматическая активация при заранее определенном низком заряде аккумуляторной батареи для следования на зарядную станцию.
5. Режим подготовки к зарядке: контролируемая посадка на зарядную станцию с использованием навигации по датчикам.
6. Режим зарядки: собственно зарядка аккумуляторной батареи.
7. Режим окончания зарядки: аккумуляторная батарея заряжена полностью, готовность к началу полета по команде (см. п.1).

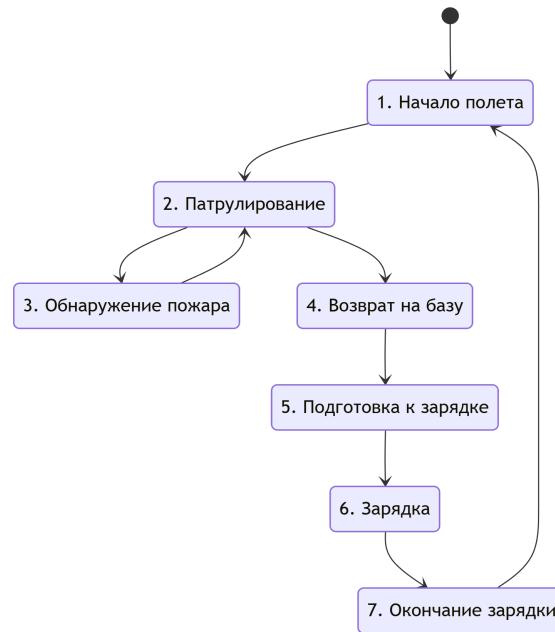


Диаграмма режимов работы БПЛА.

- Определение необходимых сенсоров:
 1. Камера дневного видения (HD/4K): для обнаружения пожаров и FPV.
 2. Тепловизор (неохлаждаемый микроболометрический): для обнаружения/подтверждения пожаров.
 3. Лидар (LIDAR): для построения карты помещения, навигации и обхода препятствий.
 4. Датчики на полетном контроллере: IMU, барометр, дальномеры (вниз/вперед/вбок): для стабилизации полета, поддержания высоты, точной навигации и посадки на зарядную станцию.
- Определение, какую телеметрию обрабатывать и для чего:

1. Видеопоток и тепловизионные данные: для обнаружения нейросетью пламени и дыма, визуальный контроль и подтверждение тревоги дежурным диспетчером, FPV оператора. На наземную станцию управления и в пожарную сеть предприятия передается видеопоток уже обработанный нейросетью YOLO (с наложенными bounding boxes). При обнаружении пожара, в пожарную сеть передается сигнал тревоги с передачей параметров детекции пожара нейросетью.
 2. Данные с лидара: для построения карты помещения, навигации и обхода препятствий.
 3. Датчики (IMU, барометр, дальномеры): фильтрация данных.
 4. Полетная телеметрия (высота, скорость, ориентация, заряд батареи, тип съемки видео/тепловизор): для управления и мониторинга состояния БПЛА.
 5. Координаты БПЛА в системе координат ангара: для определения местоположения БПЛА и очага возгорания.
- Определение, какую полезную нагрузку будет нести устройство:
 1. Модуль камеры (дневная + тепловизор) на сервоприводе / гимбale.
 2. Модуль одноплатного вычислителя (Raspberry Pi 5).
 3. Модуль LIDAR.
 4. Модуль ускорения нейронных сетей (EDGE TPU): Google USB Accelerator (Coral).
 5. Проблесковый маячок красного цвета.
 6. Сирена/звуковой излучатель.
 7. Механизм-сцепка с автоматической зарядной станцией.
 - Определение с алгоритмами навигации:
 1. Навигация по заданным via-points в системе координат ангара.
 2. SLAM (Simultaneous Localization and Mapping) на основе данных LIDAR, для построения карты и точного позиционирования в реальном времени.
 3. Алгоритм обхода статических и динамических препятствий.
 4. Алгоритмы фильтрации данных с датчиков.
 - Определение, что делать в случае обнаружения пожара:

При обнаружении пожара (пламени/дыма): выход из режима патрулирования, **автономное приближение к детектируемой цели**, стабилизация, наведение камер на цель, передача сообщения об обнаружении пожара в пожарную сеть предприятия и на наземную станцию управления оператору, ожидание команды "ОТБОЙ ТРЕВОГИ" для перехода в основной режим патрулирования.

- Какая наземная станция управления будет использоваться: Mission Planner и/или QGroundControl, ПО собственной разработки исполнителя.
- Какие логи необходимо сохранять:
 1. Полные полетные логи (полетный контроллер).
 2. Логи работы ROS-нод (вычислитель Raspberry Pi).
 3. Логи детекции нейросети (время, координаты, уверенность срабатывания, класс объекта).
 4. Логи всех передаваемых и принимаемых сообщений с пожарной сетью предприятия.
 5. Логи всех передаваемых и принимаемых сообщений с наземной станцией управления (оператором).

4. Требования к необходимой отказоустойчивости:

- Отказоустойчивость - это способность системы сохранять работоспособность при возникновении отказов отдельных ее компонентов.
- Общие положения: БПЛА должен иметь резервирование критических узлов: например, резервные гироскоп/акселерометр, резервные каналы связи с оператором.
- В целях повышения отказоустойчивости, предусмотрен Резервный канал управления через RC-пульт по протоколам (**CRSF / SBUS ?**). Это "последняя линия обороны" оператора.
- При получении сигнала с RC-пульта полетный контроллер немедленно переходит в ручной режим, отменяя все автономные команды от GCS и RPi. Это "аварийный переключатель" оператора.
- При наступлении любой потери из списка ниже, **обязательно** должна происходить отправка сообщения в пожарную сеть предприятия и на наземную станцию управления оператору!

- При потере связи по Wi-Fi с пожарной сетью предприятия остается возможной передача сообщения об этом сбое по радиоканалу на наземную станцию управления оператору. И наоборот, при потере связи по радиоканалу остается возможной передача сообщения об этом сбое по Wi-Fi в пожарную сеть предприятия. Отправка сообщений в этих случаях выполняется программными средствами на стороне вычислителя.
- При потере основного вычислителя (а следовательно и модуля Wi-Fi), сообщение об этом сбое можно передать только по радиоканалу на наземную станцию управления оператору, но сделать это возможно только программными средствами на стороне полетного контроллера ([например, на Pixhawk-Ardupilot скриптами LUA ?](#)).
- При потере полетного контроллера (а следовательно и связи по радиоканалу), сообщение об этом сбое можно передать только по Wi-Fi в пожарную сеть предприятия. Отправка сообщения в этом случае выполняется программными средствами на стороне вычислителя.
- Поведение при потере связи:
 1. При выполнении режима патрулирования и потери связи по Wi-Fi с пожарной сетью предприятия более чем на FIRE_NETWORK_TIMEOUT секунд: немедленное прекращение режима патрулирования и переход в режим возврата на базу для посадки на зарядную станцию.
 2. При выполнении режима патрулирования и потери связи по радиоканалу с наземной станцией управления более чем на GCS_NETWORK_TIMEOUT секунд: немедленное прекращение режима патрулирования и переход в режим возврата на базу для посадки на зарядную станцию.
- [Что такое fail-safe и зачем нужны сценарии: Fail-safe – это заранее запрограммированные безопасные действия системы при возникновении сбоя.](#) Сценарии:
 1. Сценарий-0. Потеря полетного контроллера - это полный 0! При потере полетного контроллера (полном его отказе) система лишается возможности управлять полетом. Сообщение об этом критическом отказе может быть передано только через Wi-Fi в пожарную сеть предприятия программными средствами на стороне вычислителя (RPi 5), при условии сохранения его работоспособности.
 2. Сценарий-1. Потеря основного вычислителя (а следовательно и датчика LIDAR): полетный контроллер переводит БПЛА в режим стабилизации и ожидает команду оператора в течение OPERATOR_CMD_TIMEOUT секунд. Если за это время команда не поступила, БПЛА, используя данные барометра и дальномеров, производит медленное контролируемое снижение для посадки на текущем месте, передавая непрерывные

предупреждающие сигналы о выполнении аварийной посадки проблесковым маячком и сиреной (или звуковым излучателем). Про передачу сообщения по радиоканалу на наземную станцию управления оператору см. выше.

3. Сценарий-2. Потеря датчика LIDAR: действия аналогичны Сценарию-1.
 4. Сценарий-3. Критический разряд батареи: Немедленная посадка на ближайшую безопасную площадку, определенную по карте. Если безопасная площадка отсутствует или недостижима, то выполняются действия аналогично Сценарию-1, только без ожидания команд от оператора.
 5. Сценарий-4. Отказ EDGE TPU: Детекция должна продолжаться на центральном процессоре Raspberry Pi 5 с пониженным FPS. После очередного ухода на зарядную станцию новый взлет на патрулирование данному БПЛА должен быть запрещен.
- После завершения fail-safe сценария или сценария при потере связи, БПЛА должен быть отправлен на анализ данных “черного ящика” - логов в самом БПЛА, логов в пожарной сети предприятия и в наземной станции управления. Далее БПЛА должен пройти проверку и/или ремонт в сервисном центре исполнителя или его агента, тестовые испытания, полеты, прежде чем снова может быть допущен на объект заказчика.
 - Установка ограничения по времени реакции:

На критические события (потеря связи, скачок в данных датчиков) система управления должна реагировать моментально – **задержка реакции контроллера не должна превышать порядка 10 миллисекунд**. Это означает наличие высокой тактовой частоты полетного контроллера и эффективных алгоритмы фильтрации помех.

- Защита от шумов и некорректных команд:
 1. Предусмотреть аппаратное и программное фильтрование шумов (например, фильтр Калмана для IMU), проверку границ входных команд (чтобы предотвращать выходы сигналов за допустимый диапазон) и алгоритмы обнаружения ложных данных датчиков.
 2. Валидация всех входящих команд от пожарной сети (checksum, аутентификация источника).
 3. Игнорирование команд, которые могут вывести БПЛА за пределы безопасной зоны (геозона).

5. Требования к производительности:

- Требования к ресурсам (Процессор CPU, RAM оперативная память):
 1. Полетный контроллер должен базироваться на 32-битном микропроцессоре (ARM Cortex-M4/M7 или аналогичный) с тактовой частотой не менее 100 МГц. Объем оперативной памяти (RAM) – достаточный для хранения рабочих данных и логов (от 256 КБ), энергонезависимой памяти (Flash/EEPROM) – не менее нескольких мегабайт для прошивки и логирования.
 2. Вычислитель (Raspberry Pi 5): Минимум 8 ГБ ОЗУ. Вычислительной мощности должно хватать для параллельной работы ROS 2, SLAM-алгоритмов и нейросети YOLO 11n (в формате TensorFlow Lite) с подключенным EDGE TPU.
- Частоты обновления:
 1. IMU: 200-500 Гц.
 2. Лидар: ≥ 10 Гц.
 3. Цикл управления полетом: ≥ 100 Гц.
 4. Цикл обработки видео нейросетью (с использованием EDGE TPU): ≥ 15-20 FPS.
- Задержка в контуре управления: общая задержка от сенсора до исполнительного механизма не более 50 мс.
- Время от детекции пожара до передачи тревожного сигнала – не более 2 секунд.
- Время отклика на команду переключения камеры – не более 1 секунды.
- Пропускная способность каналов связи:
 1. Wi-Fi канал: не менее 10 Мбит/с для стабильной передачи видеопотока (с наложенными детекциями) и REST API.
 2. Канал телеметрии должен обеспечивать передачу всех необходимых данных (MAVLink-сообщений) без потерь. Рекомендуемая скорость радиосвязи – не менее 57600 бод (лучше 115200) или цифровой канал ≥1 Мбит/с. Наличие механизмов обнаружения и коррекции ошибок в протоколе (например, контрольные суммы в MAVLink).
-

6. Требования к каналам связи:

КАНАЛ СВЯЗИ WI-FI (ЗАКАЗЧИКА)

Назначение: Обеспечение интеграции дрона с пожарной системой заказчика для оперативного оповещения о ЧС, визуального контроля и удаленного управления режимами съемки.

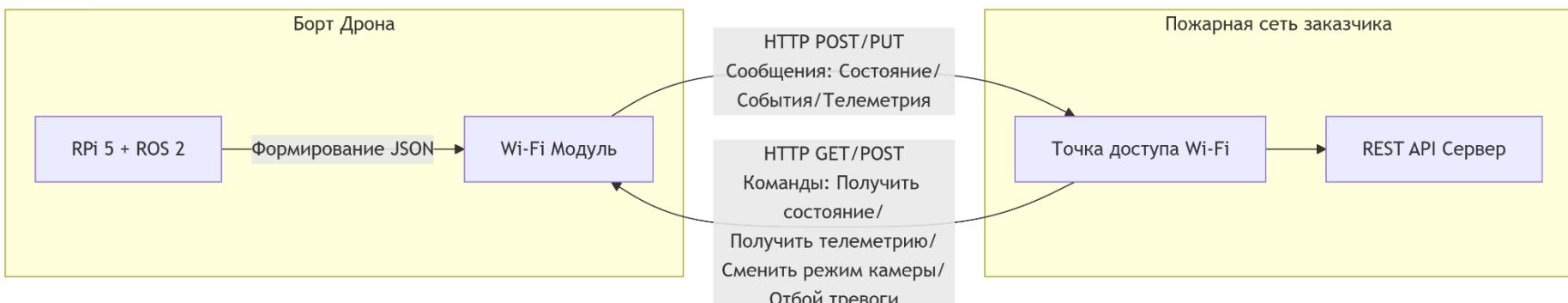
1. СХЕМА ПЕРЕДАЧИ ВИДЕОПОТОКА



Характеристики видеопотока:

- Кодек: H.264
- Нагрузка на сеть: 2-8 Мбит/с
- Задержка: 100-300 мс
- Назначение: Визуальный контроль и подтверждение тревоги дежурным диспетчером

2. СХЕМА ВЗАИМОДЕЙСТВИЯ REST API



Типы передаваемых данных:

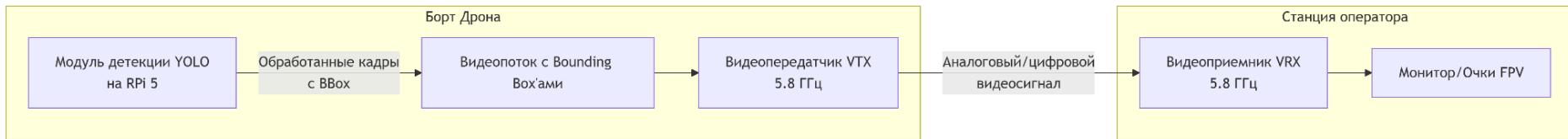
Направление	Тип данных	Примеры
Дрон → Сеть	Состояние/Телеметрия	Статус, заряд батареи, координаты
Дрон → Сеть	События	Детекция пожара, сбои системы
Сеть → Дрон	Команды	Запрос статуса, смена камеры, отбой тревоги

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ КАНАЛА WI-FI

- Скорость передачи: ≥ 10 Мбит/с
 - Протоколы приложения:
 - RTSP/HTTP для видеопотока
 - HTTP/REST API для служебных сообщений
 - Формат данных:
 - Видео: H.264
 - Данные: JSON
 - Нагрузка на канал:
 - Постоянная: видеопоток (2-8 Мбит/с)
 - Периодическая: телеметрия (10-50 кбит/с)
 - Событийная: тревоги и команды (10-100 кбит/с)
 - Задержка передачи:
 - Видеопоток: 100-300 мс
 - Данные: < 100 мс
 - Надежность: Подтверждение доставки для критических сообщений
-

КАНАЛЫ СВЯЗИ ОПЕРАТОРА

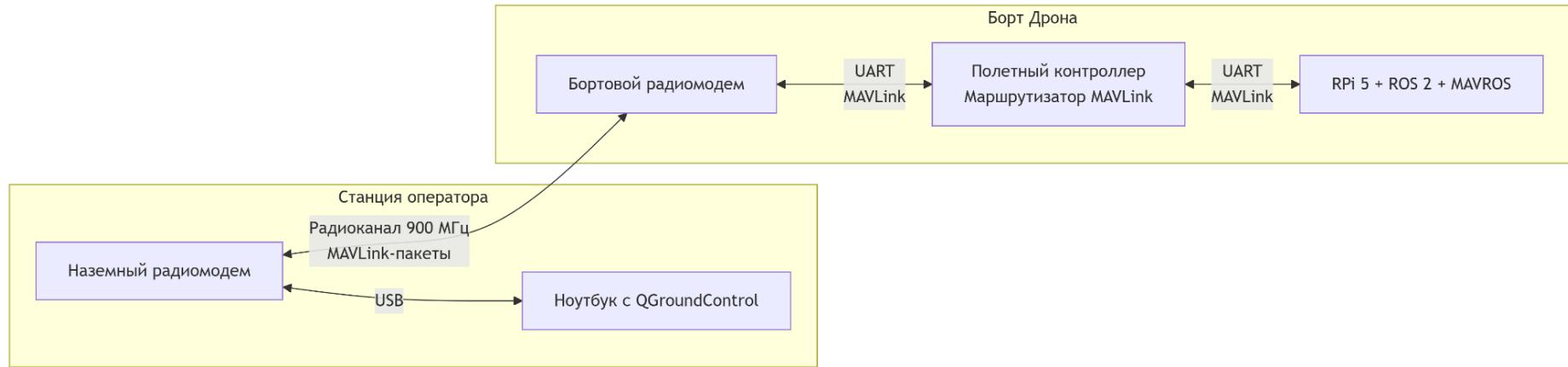
1. FPV-ВИДЕОКАНАЛ (5.8 ГГц)



Характеристики:

- Технология: Аналоговый (FatShark) или цифровой (DJI FPV, Walksnail) видеопередатчик
- Диапазон: 5.8 ГГц
- Задержка: < 30 мс (аналог), < 60 мс (цифровой)
- Качество: 720р-1080р
- Назначение: Прямое визуальное пилотирование, наблюдение в реальном времени
- Данные: Видеопоток с наложенными рамками от YOLO

2. ОСНОВНОЙ КАНАЛ УПРАВЛЕНИЯ (900 МГц / 2.4 ГГц)



Характеристики:

- Протокол: MAVLink поверх радиосвязи
- Диапазон: 900 МГц (лучшая проникающая способность в ангаре) или 2.4 ГГц
- Пропускная способность: 1-4 Мбит/с
- Задержка: 50-200 мс
- Назначение: Загрузка миссий, телеметрия, команды управления
- Данные:
 - От дрона: Полная телеметрия, статус систем (в т.ч. ROS), данные детекции YOLO
 - К дрону: Команды управления (в т.ч. полезной нагрузкой), точки маршрута, параметры, прошивка

3. РЕЗЕРВНЫЙ КАНАЛ УПРАВЛЕНИЯ (RC-ПУЛЬТ)



Характеристики:

- Протокол: SBUS или CRSF (современные цифровые протоколы)
- Диапазон: 2.4 ГГц
- Задержка: < 20 мс (экстремально низкая)
- Каналы: 8-16 каналов управления
- Назначение: Аварийное прямое управление, ручной взлет/посадка
- Приоритет: Наивысший - отменяет любые автономные команды
- Данные: Прямые команды: газ, крен, тангаж, рыскание, переключатели режимов

Сводная таблица характеристик:

Канал	Технология	Задержка	Приоритет	Назначение
RC-Пульт	2.4 ГГц, SBUS/CRSF	< 20 мс	ВЫСШИЙ	Аварийное прямое управление
FPV-Видео	5.8 ГГц, VTX/VRX	< 30-60 мс	-	Визуальное пилотирование
Основной MAVLink	900 МГц, радиомодем	50-200 мс	Средний	Автономное управление, телеметрия

Эти три канала работают параллельно и независимо, обеспечивая полное резервирование управления и наблюдения.

7. Требования к интерфейсам:

- Что такое API модуля управления и какие требования бывают: Это программный интерфейс для взаимодействия между компонентами.
 - Требования: Четко определенные топики, сервисы и сообщения ROS 2 для обмена данными между модулем навигации, модулем детекции, модулем управления камерами и модулем связи.
 - REST API, JSON - для взаимодействия с пожарной сетью предприятия.
- Протоколы обмена:
 - Внутрибортовые: MAVLink (через UART/SPI), ROS 2 ([через DDS](#)), взаимодействие между ROS 2 и полетным контроллером должно осуществляться через MAVROS или его аналог для ROS 2.

2. Внешние: Wi-Fi (TCP/UDP) для связи с пожарной сетью предприятия. MAVLink (через радиомодем) для связи с наземной станцией управления. RC-каналы пульта управления (**CRSF / SBUS ?**).
- Формат телеметрии:
 1. Полетная телеметрия: стандартные сообщения MAVLink.
 2. Данные для пожарной сети предприятия: JSON-сообщения, включающие тип сообщения (состояние, события, команды), ID БПЛА, координаты, уровень заряда, данные детекции, текущий режим камеры Видео/Тепловизор т.д.
 - Определение интерфейсов для сенсоров, наземной станции, облака:
 1. Сенсоры: Подключение через USB 3.0 (камеры, EDGE TPU), UART/SPI/I2C (лидар, полетный контроллер).
 2. Наземная станция (Mission Planner/QGC): Связь по радиоканалу (например, 900 МГц) с ретрансляцией MAVLink.
 3. Пожарная сеть предприятия: сетевой REST API или MQTT-брюкер по Wi-Fi. Должен предоставить: команду смена режима камеры Видео/Тепловизор, команду "ОТБОЙ ТРЕВОГИ".

8. Требования к тестированию:

- Что такое стендовое испытание и зачем нужно: Тестирование систем БПЛА на стенде (без полета) для проверки взаимодействия компонентов, логики и ПО.
 1. План: Стендовая отладка ROS-нод, проверка детекции нейросети на записанном видео, имитация передачи сигналов, проверка работы EDGE TPU и переключения режимов камер.
- Симуляция:
 1. Проведение испытаний в симуляторе (Gazebo + ROS 2) для отладки навигации, SLAM и поведения в ЧС.
- Летные испытания:
 1. Этап 1: Испытания в безопасном помещении для проверки стабильности полета и навигации.
 2. Этап 2: Испытания в целевом ангаре (или его макете) для проверки всего функционала, включая детекцию пожара (на имитаторах огня/дыма), проверку FPS детекции и отработку команд переключения камер.
- Определение критериев успешности:

1. Успешное выполнение 10 циклов патрулирования подряд без сбоев.
2. Детекция тестовых очагов возгорания/дыма с вероятностью > 98% (**с уверенностью > 0.55**).
3. Успешное выполнение 5 циклов "зарядка по низкому заряду батареи" подряд.
4. Корректная передача 100% тревожных сообщений в тестовую пожарную сеть.
5. Средний FPS детекции с использованием EDGE TPU не ниже 15 кадров/с.
6. Время от детекции пожара до передачи тревожного сигнала – не более 2 секунд.
7. Время отклика на команду переключения камеры – не более 1 секунды.
8. Время реакции на потерю связи – не более 5 секунд.

9. Требования к документации:

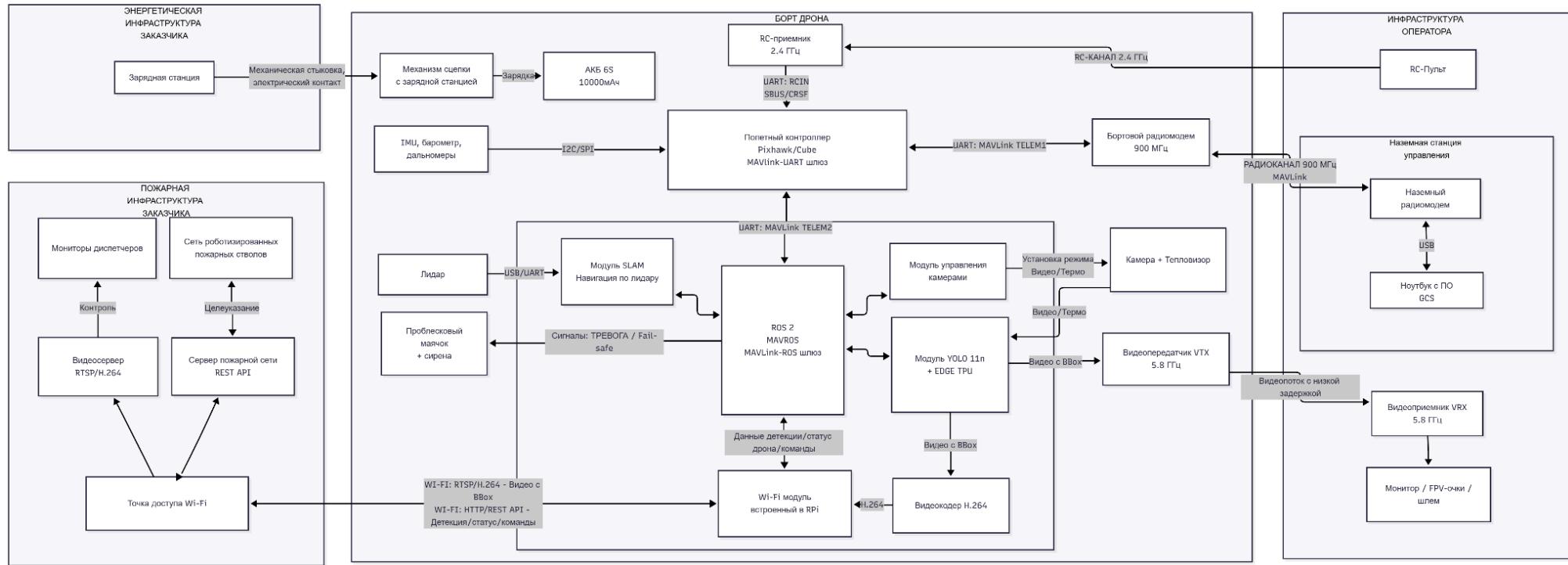
- Руководство оператора: Инструкция по подготовке к полету, настройке маршрута, запуску, мониторингу и переключению режимов камер.
- Альбом инструкций для оператора на реагирование при “Fail-safe”.
- Документация на API: Описание форматов сообщений JSON/MQTT для интеграции с пожарной сетью заказчика, включая команды управления.
- Схемы взаимодействия: Блок-схемы, отображающие взаимодействие между ROS-нодами, полетным контроллером и внешними системами, каналы передачи данных с потоками данных.
- Инструкция по компиляции и обновлению моделей ИИ: Описание процесса конвертации моделей YOLO в формат TensorFlow Lite для развертывания на EDGE TPU.

10. Требования к летной платформе и энергетике:

- Тип БПЛА: Квадрокоптер.
- Диагональ рамы: 450-550 мм (класс 8-10 дюймов).
- Полная взлетная масса (MTOW): ≤ 4.5 кг.
- Вес полезной нагрузки: ≥ 1.0 кг.

- Энергосистема:
 - Тип АКБ: LiPo.
 - Вольтаж: 6S (22.2 В).
 - Емкость: 10000 мАч (рекомендуемая).
 - Летные характеристики:
 - Тяговооруженность: $\geq 2:1$ (требуется для точного позиционирования и маневренности в помещении, а не для борьбы с ветром).
 - Время непрерывного циклического патрулирования в помещении: ≥ 35 минут.
 - Рабочий режим полета: Крейсерский, 50-70% от максимальной мощности двигателей.
 - Автономная работа (с учетом времени зарядки): Определяется количеством БПЛА в рое и конфигурацией зарядных станций.
-

Приложение А. Диаграмма взаимодействия систем



РАСШИФРОВКА ДИАГРАММЫ:

I. БОРТ ДРОНА

A. Система управления полетом:

- Полетный контроллер (Pixhawk/Cube) - основной мозг управления полетом
- Даталоггер - запись полетных логов

Б. Вычислительный модуль (RPi 5 + ПО):

- MAVROS - шлюз между ROS и MAVLink
- YOLO 11n + EDGE TPU - детекция пламени/дыма
- SLAM модуль - навигация по данным лидара
- Модуль управления камерами - переключение режимов Видео/Тепловизор

В. Сенсоры:

- Камера + Тепловизор - получение видеопотока
- Лидар - построение карты, обход препятствий
- IMU, барометр, дальномеры - стабилизация, навигация

Г. Коммуникационные модули:

- Видеопередатчик VTX (5.8 ГГц) - FPV-видео для оператора
- Бортовой радиомодем (900 МГц) - основной канал MAVLink
- RC-приемник (2.4 ГГц) - резервное управление
- Wi-Fi модуль (встроенный в RPi) - связь с пожарной сетью

Д. Вспомогательные системы:

- АКБ 6S 10000мАч - питание
- Проблесковый маячок + сирена - светозвуковая сигнализация
- Механизм сцепки - автономная зарядка

II. ИНФРАСТРУКТУРА ОПЕРАТОРА

А. Наземная станция управления:

- Ноутбук с ПО GCS - управление полетом, мониторинг
- Наземный радиомодем (900 МГц) - прием/передача MAVLink
- Видеоприемник VRX (5.8 ГГц) - прием FPV-видео
- Монитор/FPV-Очки - отображение видео

Б. Резервное управление:

- RC-пульт (2.4 ГГц) - прямое аварийное управление

III. ПОЖАРНАЯ ИНФРАСТРУКТУРА ЗАКАЗЧИКА

А. Сетевая инфраструктура:

- Точка доступа Wi-Fi - подключение дронов по Wi-Fi
- Сервер пожарной сети - обработка данных, REST API

Б. Системы отображения и реагирования:

- Мониторы диспетчеров - визуальный контроль
- Роботизированные пожарные стволы - автоматическое тушение
- Видеосервер - прием видеопотока и архивирование

В. Инфраструктура обслуживания:

- Зарядная станция - автономная подзарядка дронов

Приложение Б. Форматы JSON-сообщений

Примечание: данный раздел описывает только пакеты данных, передаваемые по каналу Wi-Fi в рамках текущего взаимодействия с пожарной сетью предприятия. Взаимодействие с наземной станцией управления (GCS) осуществляется по протоколу MAVLink через радиоканал и в настоящем приложении не описывается.

Протокол REST API для полноценного управления дроном из пожарной сети может быть расширен по взаимной договоренности между исполнителем и заказчиком, что должно быть оформлено официальным дополнением к ТЗ.

Б.1. Общая структура пакета

Все сообщения в системе имеют следующую структуру JSON-объекта:

```
json
{
    "msg_dst": "<адрес_получателя>",
    "msg_src": "<адрес_отправителя>",
    "msg_type": "<тип_пакета>",
    "timestamp": <unix_time>,
    "payload": { ... },
    "crc": "<контрольная_сумма_CRC>"
}
```

Описание полей с допустимыми значениями:

Поле	Тип	Допустимые значения	Описание
msg_dst	string	fire_station ground_station drone_01 drone_02 ... broadcast	Мнемонический адрес системы-получателя. broadcast – широковещательное сообщение.
msg_src	string	fire_station ground_station drone_01 drone_02 ...	Мнемонический адрес системы-отправителя.
msg_type	string	status event command	Тип пакета.
timestamp	integer	Целое число, секунды с 00:00:00 01.01.1970 UTC (Unix time)	Время формирования пакета данных, события и т.п. Пример: 1733651422

(соответствует
2025-12-08T14:30:22Z).

payload	object	Зависит от <code>msg_type</code> и конкретного события/команды (см. разделы Б.2.1–Б.2.3)	Полезная нагрузка пакета.
crc	string	Две шестнадцатеричные цифры (0–9, A–F), например "2A"	Контрольная сумма, рассчитанная по алгоритму CRC NMEA 0183 (см. раздел Б.4).

Б.2. Типы пакетов и их содержимое

Б.2.1. Пакеты типа `status` (статус)

Используются для регулярной периодической или ответной передачи состояния системы.

Структура payload для status:

```
json
{
    "system_state": "<состояние_системы>",
    "battery_level": <уровень_заряда_в_процентах>,
    "battery_voltage": <напряжение_в_вольтах>,
    "speed": {
        "vx": <скорость_X_м/с>,
        "vy": <скорость_Y_м/с>,
        "vz": <скорость_Z_м/с>
    },
    "position": {
        "x": <координата_X_метры>,
        "y": <координата_Y_метры>,
        "z": <координата_Z_метры>
    },
    "video_mode": "<режим_камеры>",
    "connection_status": {
        "wifi": <логическое_значение>,
        "mavlink": <логическое_значение>,
    }
}
```

```
    "rc": <логическое_значение>
}

}
```

Допустимые значения:

- system_state: STANDBY | PATROLLING | FIRE_DETECTED | RTL | LANDING | CHARGING | FAILSAFE
- video_mode: VIDEO | THERMAL
- battery_level: 0–100
- battery_voltage: число с плавающей точкой
- speed.vx, speed.vy, speed.vz: числа с плавающей точкой
- position.x, position.y, position.z: числа с плавающей точкой
- connection_status.wifi, connection_status.mavlink, connection_status.rc: true | false

Пример:

```
json
{
    "msg_dst": "fire_station",
    "msg_src": "drone_01",
    "msg_type": "status",
```

```
"timestamp": 1733651422,  
"payload": {  
    "system_state": "PATROLLING",  
    "battery_level": 85,  
    "battery_voltage": 24.2,  
    "speed": { "vx": 0.5, "vy": 0.3, "vz": 0.0 },  
    "position": { "x": 12.5, "y": 8.2, "z": 3.0 },  
    "video_mode": "VIDEO",  
    "connection_status": { "wifi": true, "mavlink": true, "rc": false }  
},  
"crc": "3F"  
}
```

Б.2.2. Пакеты типа event (событие)

Б.2.2.1. Событие FIRE_DETECTED

Назначение: Уведомление об обнаружении пожара.

```
json  
{  
    "msg_dst": "fire_station",
```

```
"msg_src": "drone_01",
"msg_type": "event",
"timestamp": 1733651535,
"payload": {
    "event": "FIRE_DETECTED",
    "confidence": 0.92,
    "class": "flame",
    "drone_position": {
        "x": 12.5,
        "y": 8.2,
        "z": 3.0
    },
    "fire_position": {
        "x": 15.7,
        "y": 12.1,
        "z": 1.5
    },
    "bbox": {
        "x_min": 300,
        "y_min": 200,
        "x_max": 340,
        "y_max": 240
    },
    "video_stream_url": "rtsp://192.168.1.100:8554/guardian_stream",
    "video_mode": "VIDEO"
},
"crc": "5C"
}
```

Описание полей payload:

Поле	Тип	Описание
event	string	FIRE_DETECTED
confidence	float	Уверенность модели (0.0–1.0).
class	string	Класс объекта: flame smoke.
drone_position	object	Координаты БПЛА в момент детекции.
fire_position	object	Координаты очага пожара.
bbox	object	Координаты bounding box в пикселях.
video_stream_url	string	URL RTSP-потока с детекцией.
video_mode	string	Режим камеры: VIDEO THERMAL.

Б.2.2.2. Событие SYSTEM_WARNING

Назначение: Предупреждение о не критическом отказе компонента, потере связи или не критическом разряде батареи, не требующем оперативного вмешательства оператора.

```
json
{
  "msg_dst": "fire_station",
  "msg_src": "drone_01",
  "msg_type": "event",
  "timestamp": 1733651710,
  "payload": {
    "event": "SYSTEM_WARNING",
    "component": "RADIO_LOST",
    "message": "MAVLink connection to GCS lost for 5 seconds, initiating RTL",
    "system_state": "PATROLLING",
    "recommended_action": "Check radio link and GCS"
  },
  "crc": "7A"
}
```

Описание полей payload:

Поле	Тип	Описание
event	string	SYSTEM_WARNING

component	string	Идентификатор компонента/условия: EDGE_TPU BATTERY_LOW WIFI_LOST RADIO_LOST, ...
message	string	Текстовое описание предупреждения.
system_state	string	Текущее состояние системы (например, PATROLLING).
recommended_action	string	Рекомендуемое действие для обслуживания или мониторинга.

Б.2.2.3. Событие SYSTEM_FAILURE

Назначение: Уведомление о критическом отказе компонента, требующем немедленного перехода в режим fail-safe.

```
json
{
  "msg_dst": "fire_station",
  "msg_src": "drone_01",
  "msg_type": "event",
```

```
"timestamp": 1733651800,  
"payload": {  
    "event": "SYSTEM_FAILURE",  
    "component": "LIDAR",  
    "message": "LIDAR sensor failure, initiating emergency landing",  
    "system_state": "FAILSAFE",  
    "failover_action": "EMERGENCY_LANDING"  
},  
"crc": "8F"  
}
```

Описание полей payload:

Поле	Тип	Описание
event	string	SYSTEM_FAILURE
component	string	Идентификатор компонента: FLIGHT_CONTROLLER MAIN_COMPUTER LIDAR BATTERY_CRITICAL_LOW, ...
message	string	Текстовое описание отказа.

system_state	string	FAILSAFE
failover_action	string	Автоматическое действие: EMERGENCY_LANDING RTL HOLD.

Б.2.3. Пакеты типа command (команда)

Доступные команды (от пожарной сети к БПЛА):

1. FIRE_ALARM_CANCEL – отбой пожарной тревоги.
2. SWITCH_CAMERA_MODE – смена режима камеры (видео/тепловизор).
3. REQUEST_STATUS – запрос текущего статуса.

Описание полей payload для всех команд:

Поле	Тип	Обязательно	Описание
command	string	Да	Команда.

<code>param</code>	<code>mixed</code>	Да	Параметр команды (зависит от <code>command</code>).
<code>operator_id</code>	<code>string</code>	Да	Идентификатор диспетчера/оператора.

Б.2.3.1. Команда FIRE_ALARM_CANCEL

Назначение: Отмена состояния тревоги с указанием следующего режима работы.

```
json
{
  "msg_dst": "drone_01",
  "msg_src": "fire_station",
  "msg_type": "command",
  "timestamp": 1733651580,
  "payload": {
    "command": "FIRE_ALARM_CANCEL",
    "param": "PATROLLING",
    "operator_id": "disp_01"
  },
  "crc": "8B"
}
```

Допустимые значения param:

- PATROLLING – возобновить патрулирование.
 - RTL – вернуться на базу для зарядки.
 - LAND – выполнить посадку на текущем месте.
-

Б.2.3.2. Команда SWITCH_CAMERA_MODE

Назначение: Переключение режимов камеры.

```
json
{
  "msg_dst": "drone_01",
  "msg_src": "fire_station",
  "msg_type": "command",
  "timestamp": 1733651460,
  "payload": {
    "command": "SWITCH_CAMERA_MODE",
    "param": "THERMAL",
    "operator_id": "disp_02"
  },
  "crc": "9D"
}
```

Допустимые значения param:

- VIDEO – дневная камера.

- THERMAL – тепловизор.
-

Б.2.3.3. Команда REQUEST_STATUS

Назначение: Запрос статуса с управлением периодичностью отправки.

```
json
{
  "msg_dst": "drone_01",
  "msg_src": "fire_station",
  "msg_type": "command",
  "timestamp": 1733651340,
  "payload": {
    "command": "REQUEST_STATUS",
    "param": null,
    "operator_id": "disp_03"
  },
  "crc": "A1"
}
```

Допустимые значения param:

- null – однократный ответ с текущим статусом.
- 0 – отменить периодическую отправку статуса.
- >0 (целое число) – установить интервал отправки статуса в секундах.

Подводные камни (учтены в данном ТЗ):

- Исключена ошибка переписи требований – все ключевые функции заказчика детализированы.
- ТЗ служит рамками для изменения целей заказчиком – любые изменения должны вноситься официальными дополнениями к ТЗ.
- Исключены расплывчатые формулировки – требования конкретны и измеримы (время реакции, вероятность детекции, FPS).
- Заложено тестирование на уровне требований – раздел 7 напрямую вытекает из функциональных требований.

Подведение итогов, финальная сборка ТЗ:

- Оформление итогового документа должно быть выполнено в соответствии со стандартом [Указать ГОСТ или внутренний стандарт предприятия, например, ГОСТ 34.602-89].