

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## на разработку программного обеспечения наземной станции управления (GCS) для автономного БПЛА

---

### Основание для разработки

ТЗ составляется на основании постановки задачи от предприятия-изготовителя квадрокоптеров о необходимости создания наземной станции управления БПЛА с прошивкой Ardupilot с минимально необходимым набором функций по управлению, простым и дружелюбным графическим интерфейсом пользователя.

- Заказчик: АО "Metaferrum", разработчик дронов.
- Исполнитель: Сафронов Юрий Анатольевич
- Задачи, лежащие в основе разработки:
  1. Создание простого и понятного программного интерфейса для управления БПЛА через протокол MAVLink.
  2. Разработка удобного инструмента визуального планирования миссий на спутниковых картах.
  3. Автоматизация типовых операций управления дроном (взлет, посадка, смена режимов полета).
  4. Обеспечение возможности работы с офлайн-картами.

### Назначение и цели

- Для чего ведется разработка: Создание специализированного ПО "Mini GCS" для оперативного управления автономными БПЛА при выполнении облетов территорий, мониторинга и других задач
- Какие проблемы решает:
  1. Упрощение процесса планирования миссий через интуитивный графический интерфейс.
  2. Снижение порога вхождения для операторов БПЛА.
  3. Минимизация ошибок при вводе координат путевых точек.
  4. Обеспечение единого интерфейса для основных операций управления.
- Что будет автоматизировать:

1. Установку соединения с автопилотом.
  2. Отправку команд управления (арм/дизарм, режимы полета).
  3. Формирование и загрузку маршрутов в автопилот.
  4. Визуализацию текущей позиции и планируемого пути.
- 

## Требования к функциональной части

### Необходимые протоколы

- MAVLink v2.0 - основной протокол связи с автопилотом
- TCP/IP - транспортный протокол для соединения с автопилотом
- Поддержка строк подключения формата: `tcp:IP_адрес:порт`

### Определение режимов полета

- GUIDED - режим ручного управления через GCS
- AUTO - автономное выполнение загруженной миссии
- LAND - автоматическая посадка
- Режимы должны устанавливаться через соответствующие кнопки интерфейса
- Команды: ARM, TAKEOFF, DISARM, Установить HOME, Загрузить точки маршрута

### Определение необходимых сенсоров *(со стороны автопилота)*

- GPS/ГЛОНАСС приемник для позиционирования
- Барометр для определения высоты
- IMU (инерциальный измерительный блок)
- *Примечание: GCS получает данные от сенсоров через телеметрию MAVLink*

### Определение какую телеметрию обрабатывать и для чего

- GPS-позиция - для отображения на карте
- Высота - для контроля состояния полета
- Статус полета (режимы полета, арм/дизарм) - для индикации состояния
- Батарея - для мониторинга энергосистемы

- Скорость - для контроля выполнения миссии

## Определение какую полезную нагрузку будет нести устройство

- *Не определено в этом задании, GCS не накладывает ограничений, но дополнительные команды надо будет реализовать в официальном дополнении к ТЗ.*

## Определение с алгоритмами навигации

- Основной алгоритм: следование по путевым точкам (waypoint navigation)
- Поддержка взлета на заданную относительную высоту
- Автоматическая посадка в текущей позиции
- Возможность задания домашней точки (Home Position)

## Определение что делать в случае ЧС

- Кнопка экстренной посадки (LAND)
- Кнопка дизарма (DISARM) - только при нахождении на земле
- Возможность быстрой смены режима на GUIDED для ручного управления

## Какая наземная станция будет использоваться

- Mini GCS v1.0.0 - разрабатываемая система
- Требования к аппаратной части станции:
  - ПК/ноутбук с ОС Windows/Linux
  - Ethernet/Wi-Fi соединение с автопилотом
  - Минимум 2 ГБ оперативной памяти
  - 100 МБ свободного места на диске

## Какие логи необходимо сохранять

- Журнал всех отправленных команд с временными метками
- История изменений маршрута
- Сообщения об ошибках соединения
- Координаты всех путевых точек
- Параметры выполненной миссии

---

## Требования к необходимой отказоустойчивости

Отказоустойчивость - способность системы сохранять работоспособность при возникновении ошибок или сбоев. Для GCS включает устойчивость к потере связи, некорректным данным и ошибкам оператора.

### Поведение при потере связи

1. Визуальная индикация потери связи (изменение цвета/иконки кнопки подключения)
2. Сохранение текущего состояния интерфейса
3. Автоматические попытки переподключения с заданным интервалом (опционально)
4. Предупреждение оператора о потере телеметрии

### Что такое fail-safe и зачем нужны сценарии

Fail-safe - механизмы, обеспечивающие безопасное состояние системы при сбоях. Для GCS:

- Сценарий 1: При потере связи - запрет отправки новых команд до восстановления соединения
- Сценарий 2: При ошибке загрузки маршрута - сохранение предыдущего валидного состояния
- Сценарий 3: При некорректных координатах - отказ от добавления точки с выводом ошибки

### Установка ограничения по времени реакции

- Отклик на действия пользователя:  $\leq 100$  мс
- Обновление позиции на карте:  $\leq 500$  мс
- Обработка телеметрии:  $\leq 100$  мс на сообщение
- Загрузка маршрута:  $\leq 5$  секунд для 100 точек

### Защита от шумов и некорректных команд

1. Валидация входных данных (координаты в допустимых диапазонах)

2. Проверка состояния перед отправкой команд (например, нельзя взлететь без арма)
  3. Подтверждение критических команд (дизарм, смена режима)
  4. Фильтрация дублирующихся команд
  5. Ограничение скорости отправки команд (rate limiting)
- 

## Требования к производительности

### Частоты обновления

- Обновление позиции на карте: 2-5 Гц (при наличии телеметрии)
- Обновление статус-бара: 1 Гц
- Обработка пользовательского ввода: немедленно

### Задержка в контуре управления

- Задержка от нажатия кнопки до отправки команды:  $\leq 50$  мс
- Отображение подтверждения команды:  $\leq 200$  мс

### Пропускная способность каналов связи

- Минимальная: 56 кбит/с (для базового управления)
- Рекомендуемая: 256 кбит/с (для стабильной работы с телеметрией)
- Максимальная: 1 Мбит/с (для будущего расширения функционала)

### Требования к ресурсам

- Процессор (CPU): Минимум 2 ядра, 1.5 ГГц
  - Оперативная память (RAM): Минимум 512 МБ, рекомендуется 1 ГБ
  - Дисковое пространство: 100 МБ для программы + место для офлайн-карт
  - GPU: Интегрированная графическая карта с поддержкой OpenGL 2.0+
- 

## Требования к интерфейсам

API модуля управления - набор функций и методов для взаимодействия с автопилотом. Требования:

- Четкая документация каждого метода
- Обработка ошибок с понятными кодами
- Потребность безопасности при одновременном доступе
- Расширяемость для добавления новых команд

## Протоколы обмена

- Внешний: MAVLink поверх TCP
- Внутренний: Собственный API между модулями GCS (Python-модули)

## Формат телеметрии

- MAVLink messages (стандартные типы сообщений)
- Структурированные данные в формате Python-словарей для внутренней обработки
- JSON-формат для сохранения логов

## Определение интерфейсов

- Сенсоры: Через MAVLink телеметрию (параметры, RAW\_IMU, GPS\_RAW\_INT и т.д.)
- Наземная станция: GUI на основе CustomTkinter, взаимодействие через события
- Облако: Не предусмотрено в текущей версии (возможность расширения)

---

## Требования к тестированию

### Что такое стендовое испытание и зачем нужно

Стендовое испытание - тестирование GCS без реального БПЛА, с использованием симулятора. Необходимо для:

- Проверки логики работы интерфейса
- Отладки взаимодействия с MAVLink
- Тестирования сценариев отказа

## Симуляция

- Использование ArduPilot SITL (Software In The Loop)
- Эмуляция различных сценариев полета
- Тестирование реакции на потерю связи
- Проверка обработки некорректных данных

## Летные испытания

1. Базовый функционал:
  - Подключение к реальному автопилоту
  - Проверка команд ARM/DISARM
  - Тестирование взлета и посадки
2. Планирование миссий:
  - Создание и загрузка простых маршрутов
  - Выполнение автономного полета
  - Проверка точности следования по точкам
3. Граничные условия:
  - Работа при слабом сигнале
  - Быстрая смена режимов
  - Экстренное прерывание миссии

## Определение критериев успешности

1. Функциональные:
    - 100% успешное выполнение запланированных тест-кейсов
    - Отсутствие критических ошибок (краши программы)
  2. Производительность:
    - Отклик интерфейса в пределах требований
    - Стабильная работа при 4+ часах непрерывной работы
  3. Надежность:
    - Успешное восстановление после потери связи
    - Корректная обработка некорректного ввода
- 

## Требования к документации

### Руководство оператора

1. Установка и запуск:
  - Системные требования
  - Процесс установки зависимостей
  - Настройка соединения с автопилотом
2. Базовое управление:
  - Описание интерфейса
  - Подключение к дрону
  - Основные команды управления
3. Планирование миссий:
  - Добавление путевых точек
  - Редактирование маршрута
  - Загрузка миссии в автопилот
4. Устранение неисправностей:
  - Частые проблемы и решения
  - Диагностика подключения
  - Восстановление после сбоев

## Документация на API

- Описание всех публичных функций в модулях `flight_control.py` и `mission_control.py`
- Примеры использования каждого метода
- Описание форматов входных/выходных данных
- Коды ошибок и их значения

## Схемы взаимодействия

1. Блок-схема архитектуры GCS
2. Диаграмма последовательности для типового сценария (создание миссии → выполнение)
3. Схема соединения оборудования (GCS ↔ Радиомодем ↔ БПЛА)
4. Диаграмма состояний автопилота и их отображения в GCS

## ТЕРМИНЫ И СОКРАЩЕНИЯ

### Основные термины



БПЛА (Беспилотный летательный аппарат) - летательный аппарат без экипажа на борту, управляемый автоматически или оператором с наземной станции.

GCS (Ground Control Station) - наземная станция управления, программно-аппаратный комплекс для управления БПЛА, приема телеметрии и планирования миссий.

Автопилот - аппаратно-программный комплекс на борту БПЛА, обеспечивающий стабилизацию полета и выполнение автономных миссий (в данном контексте - ArduPilot).

Миссия - заранее определенный план полета БПЛА, включающий последовательность путевых точек и команд.

Путевая точка (Waypoint, WP) - географическая координата (широта, долгота, высота), через которую должен пролететь БПЛА при выполнении миссии.

Домашняя точка (Home Position, HP) - начальная точка взлета или базовая позиция БПЛА, используемая для возврата в случае потери связи или низкого заряда батареи.

Телеметрия - данные о состоянии БПЛА (позиция, скорость, заряд батареи и др.), передаваемые на наземную станцию в реальном времени.

## Технические термины

MAVLink (Micro Air Vehicle Link) - протокол обмена сообщениями между БПЛА и наземной станцией, используемый для передачи команд и телеметрии.

SITL (Software In The Loop) - программная среда имитации полета БПЛА для тестирования без реального оборудования.

Fail-safe - механизм безопасности, обеспечивающий переход системы в безопасное состояние при возникновении сбоев.

API (Application Programming Interface) - набор функций и методов для взаимодействия программных модулей между собой.

GUI (Graphical User Interface) - графический пользовательский интерфейс.

TCP/IP (Transmission Control Protocol/Internet Protocol) - стек сетевых протоколов для передачи данных.

## Режимы полета (Flight Modes)

GUIDED - режим, в котором БПЛА управляется оператором через GCS или следует заданным целевым точкам.

AUTO - полностью автономный режим выполнения загруженной миссии по путевым точкам.

LAND - режим автоматической посадки в текущей позиции или заданной точке.

STABILIZE - режим стабилизации, в котором автопилот поддерживает горизонтальное положение, а оператор управляет креном и тангажом.

LOITER - режим удержания позиции (автоматическое зависание в точке).

## Сокращения

IMU (Inertial Measurement Unit) - инерциальный измерительный блок, совокупность датчиков (акселерометр, гироскоп).

GPS (Global Positioning System) - глобальная система позиционирования.

RTK (Real Time Kinematic) - метод спутниковой навигации с сантиметровой точностью.

CPU (Central Processing Unit) - центральный процессор.

RAM (Random Access Memory) - оперативная память.

GPU (Graphics Processing Unit) - графический процессор.

UAV (Unmanned Aerial Vehicle) - беспилотный летательный аппарат (английский аналог БПЛА).

VTOL (Vertical Take-Off and Landing) - летательный аппарат вертикального взлета и посадки.

FOV (Field Of View) - поле зрения камеры или датчика.

BVLOS (Beyond Visual Line Of Sight) - полет вне прямой видимости оператора.

## Команды управления

ARM - команда активации двигателей и систем БПЛА (взведение).

DISARM - команда деактивации двигателей и систем (снятие с взведения).

TAKEOFF - команда взлета на заданную высоту.

RTL (Return To Launch) - возврат к точке взлета (не реализовано в текущей версии, но относится к типовым командам).

GEO FENCE - геозона, виртуальный периметр для ограничения полета БПЛА.

## Форматы данных

JSON (JavaScript Object Notation) - текстовый формат обмена данными.

CSV (Comma-Separated Values) - формат представления табличных данных.

DB (DataBase) - база данных, в контексте GCS - база офлайн-тайлов карт.

Tile - тайл, фрагмент карты определенного масштаба и координат.

Zoom Level - уровень масштабирования карты.

---

Подпись разработчика: \_\_\_\_\_ /Сафронов Ю.А./

Дата: [Дата составления]

Подпись заказчика: \_\_\_\_\_

Дата: [Дата согласования]