

Техническое задание на разработку программного обеспечения для автоматического патрулирования периметра охраняемой зоны с использованием БПЛА

1. Введение

В рамках организации автоматизированной охраны периметра прибрежной промышленной зоны требуется разработать программное обеспечение для управления патрульным беспилотным летательным аппаратом (БПЛА). Программа предназначена для выполнения автоматического маршрута патрулирования с взлетом с базовой станции, облетом ключевых точек наблюдения на береговой линии и ожиданием дальнейших команд в контрольной точке.

Основные термины:

- **БПЛА (Беспилотный летательный аппарат)** - дрон, оборудованный для патрулирования и наблюдения
- **Базовая станция** - точка взлета и посадки БПЛА
- **Контрольная точка** - точка патрулирования, где дрон ожидает команды
- **Миссия** - заранее определенная последовательность точек маршрута патрулирования
- **SITL (Software In The Loop)** - программная симуляция полетного контроллера для тестирования
- **MAVLink** - протокол обмена данными между БПЛА и наземной станцией управления
- **GUIDED режим** - режим ручного управления с автоматической стабилизацией

2. Основание для разработки

ТЗ составляется на основании потребности объекта "Бухта Казачья" в автоматизации охраны периметра прибрежной территории. Заказчиком выступает Министерство Обороны, исполнителем - команда разработчиков систем управления БПЛА.

Основные задачи разработки:

- Создание программного модуля для автоматического патрулирования периметра
- Реализация маршрута облета ключевых точек наблюдения
- Обеспечение стабильного позиционирования в контрольных точках
- Создание отказоустойчивой архитектуры с обработкой возможных сбоев
- Интеграция с системой управления для получения команд возврата

3. Назначение и цели

Назначение: Обеспечить автоматическое патрулирование периметра охраняемой зоны с возможностью оперативного реагирования на команды оператора.

Цели разработки:

- **Автоматизация патрулирования:** Выполнение маршрута от базы до контрольных точек без ручного управления
- **Надежность навигации:** Точное прохождение маршрута через заданные координаты
- **Стабильное позиционирование:** Удержание позиции в точках наблюдения
- **Готовность к реагированию:** Ожидание команд в контрольной точке для дальнейших действий
- **Гибкость маршрута:** Возможность настройки координат патрулирования

4. Требования к функциональной части

4.1. Управление полетом

- **Автоматическое подключение:** Установление соединения с автопилотом по TCP/IP (127.0.0.1:14550)
- **Получение текущей позиции:** Определение координат базовой станции из GLOBAL_POSITION_INT
- **Создание маршрута патрулирования:** Формирование миссии с точками наблюдения
- **Загрузка миссии:** Передача маршрута в автопилот по протоколу MAVLink

4.2. Структура маршрута патрулирования

Маршрут состоит из следующих точек:

1. **Точка 0 (Базовая станция):**
 - Текущая позиция дрона (получается автоматически)
 - Высота: 30 метров
 - Устанавливается как текущая точка миссии (current=1)
2. **Точка 1 (Береговая наблюдательная точка):**
 - Смещение: 400 метров на север, 300 метров на восток от базы
 - Высота: 30 метров
 - Назначение: Контроль береговой линии
3. **Точка 2 (Контрольная точка патрулирования):**
 - Смещение: 50 метров на север, 1350 метров на восток от базы
 - Высота: 30 метров
 - Назначение: Позиция ожидания команд и наблюдения за акваторией

4.3. Работа с координатами

- **Геодезическая система:** WGS84 (стандарт GPS)
- **Точность координат:** 7 знаков после запятой (± 1.11 см)
- **Формат передачи:** degE7 (градусы $\times 10^7$) для MISSION_ITEM_INT
- **Расчет смещений:** Использование библиотеки rproj для точных геодезических вычислений

- **Система координат миссии:** MAV_FRAME_GLOBAL_RELATIVE_ALT (относительная высота над землёй)

4.4. Параметры точек маршрута

- **Тип команды:** MAV_CMD_NAV_WAYPOINT (полет к точке)
- **Высота полета:** 30 метров (относительная)
- **Режим продолжения:** autocontinue=1 (автоматический переход между точками)
- **Радиус достижения:** По умолчанию (определяется автопилотом)
- **Параметры ожидания:** param1-4 = 0 (без задержек и дополнительных действий)

5. Требования к отказоустойчивости

5.1. Обработка ошибок связи

- **Контроль подключения:** Ожидание HEARTBEAT при установлении соединения
- **Таймауты операций:** Защита от зависания при ожидании сообщений
- **Диагностика сбоев:** Информативные сообщения об ошибках для оператора
- **Корректное завершение:** Безопасное завершение при потере связи

5.2. Контроль выполнения операций

- **Подтверждение загрузки:** Ожидание MISSION_ACK после загрузки миссии
- **Проверка последовательности:** Верификация номеров запрашиваемых точек (MISSION_REQUEST_INT)
- **Валидация координат:** Проверка корректности вычисленных координат
- **Протоколирование:** Вывод информации о каждом этапе выполнения

5.3. Безопасность полета

- **Валидация входных данных:** Проверка корректности базовых координат
- **Контроль высоты:** Проверка допустимости заданной высоты полета
- **Обработка исключений:** Перехват всех возможных ошибок на каждом этапе
- **Безопасное завершение:** Корректное закрытие соединения при любых обстоятельствах

6. Требования к производительности

6.1. Время выполнения операций

- **Подключение:** Установление соединения не более 10 секунд
- **Получение позиции:** Чтение координат не более 5 секунд
- **Расчет маршрута:** Вычисление координат всех точек не более 1 секунды
- **Загрузка миссии:** Передача 3 точек маршрута не более 10 секунд

6.2. Надежность операций

- **Успешность подключения:** 99% успешных подключений к SITL
- **Точность координат:** Погрешность не более 1 метра при расчете смещений
- **Успешность загрузки:** 95% успешных загрузок миссии в автопилот

- **Стабильность связи:** Устойчивое соединение на всех этапах выполнения

6.3. Точность навигации

- **Точность геодезических расчетов:** Использование эллипсоида WGS84
- **Учет кривизны Земли:** Корректные вычисления для расстояний до 2 км
- **Точность азимута:** Расчет направления с учетом географических координат
- **Формат координат:** Целочисленное представление degE7 без потери точности

7. Требования к интерфейсам

7.1. Программные интерфейсы

- **MAVLink протокол:** Использование стандартных MAVLink сообщений версии 2.0
- **TCP/IP соединение:** Подключение к SITL через TCP порт 14550 (127.0.0.1:14550)
- **Python API:** Использование библиотек pymavlink и pyproj
- **Структура миссии:** Стандартный формат MAVLink_mission_item_int_message

7.2. Форматы данных

- **Координаты GPS:**
 - Внутреннее представление: градусы (float) с точностью до 7 знаков
 - Передача в автопилот: degE7 (int) = градусы $\times 10^7$
- **Высота:** В метрах, относительная (relative_alt)
- **Смещения:** В метрах (север/восток) от базовой точки
- **Азимут:** В градусах (0° = север, 90° = восток)
- **Расстояние:** В метрах по поверхности эллипсоида WGS84

7.3. Конфигурационные параметры

Параметры патрулирования задаются в функции `build_mission()`:

```
# Точка 1: Береговая наблюдательная точка
north_m = 400.0    # Смещение на север (метры)
east_m = 300.0     # Смещение на восток (метры)
alt_m = 30.0       # Высота полета (метры)

# Точка 2: Контрольная точка патрулирования
north_m = 50.0     # Смещение на север (метры)
east_m = 1350.0    # Смещение на восток (метры)
alt_m = 30.0       # Высота полета (метры)
```

7.4. Форматы вывода

- **Консольный вывод:**
 - Подключение: "Подключено к системе X, компонент Y"
 - Текущие координаты: "Текущие координаты: lat=XX.XXXXXXXX, lon=XX.XXXXXXXX"
 - Загрузка миссии: "Загружаем миссию из N точек"
 - Отправка точек: "Отправка пункта миссии seq=N"
 - Подтверждение: "MISSION_ACK: {ack}"

8. Требования к тестированию

8.1. Модульное тестирование

- **Тестирование подключения:** Проверка установления соединения и получения HEARTBEAT
- **Тестирование получения позиции:** Проверка корректности чтения GLOBAL_POSITION_INT
- **Тестирование расчетов:** Проверка точности геодезических вычислений (rproj)
- **Тестирование формирования миссии:** Проверка создания корректной структуры MAVWPLoader
- **Тестирование загрузки:** Проверка протокола загрузки миссии (CLEAR → COUNT → REQUEST → ITEM → ACK)

8.2. Интеграционное тестирование

- **Полный цикл в SITL:** Выполнение всех этапов в симуляторе ArduPilot
- **Тестирование с реальными координатами:** Использование координат тестового полигона
- **Тестирование различных смещений:** Проверка корректности для разных расстояний (100м - 2000м)
- **Тестирование на разных высотах:** Проверка работы для высот 10м - 100м

8.3. Функциональное тестирование

- **Тестовый маршрут патрулирования:** Выполнение миссии с заданными координатами
- **Проверка точности:** Контроль достижения заданных точек с допустимой погрешностью
- **Тестирование на карте:** Визуальная проверка маршрута в Mission Planner
- **Проверка последовательности:** Контроль правильного порядка прохождения точек

8.4. Критерии приемки

1. **Успешное подключение:** Установление соединения с SITL за ≤ 10 секунд
2. **Корректное определение базы:** Получение координат текущей позиции
3. **Точность расчетов:** Погрешность геодезических вычислений ≤ 1 метр
4. **Успешное формирование миссии:** Создание 3 корректных точек маршрута
5. **Успешная загрузка:** Загрузка миссии в автопилот с получением MISSION_ACK
6. **Корректность координат:** Визуальная проверка маршрута на карте в Mission Planner
7. **Обработка ошибок:** Корректная реакция на все типы ошибок

9. Требования к документации

9.1. Техническая документация

- **Архитектура системы:** Описание структуры программы и взаимодействия компонентов
- **Алгоритм работы:** Блок-схема последовательности операций

- **Геодезические расчеты:** Описание метода вычисления координат по смещениям
- **Протокол MAVLink:** Описание используемых сообщений и их последовательности

9.2. Пользовательская документация

- **Руководство по установке:** Установка Python, pymavlink, pyproj
- **Руководство по настройке:** Настройка SITL ArduPilot
- **Руководство по запуску:** Последовательность действий для выполнения миссии
- **Настройка параметров:** Описание изменения координат и высоты патрулирования
- **Интеграция с системой управления:** Руководство по подключению модуля возврата

9.3. Документация разработчика

- **API функций:** Описание всех функций с параметрами и возвращаемыми значениями
- **Структуры данных:** Описание MAVLink_mission_item_int_message
- **Примеры использования:** Код примеров для различных сценариев
- **Расширение функциональности:** Инструкции по добавлению новых точек маршрута
- **Отладка:** Методы диагностики проблем с координатами и связью

10. Требования к безопасности

10.1. Безопасность полетов

- **Валидация координат:** Проверка корректности базовых координат перед расчетами
- **Контроль высоты:** Проверка допустимости заданной высоты (10-100м)
- **Проверка смещений:** Контроль разумности расстояний (не более 5 км)
- **Безопасные параметры:** Использование относительной высоты для избежания столкновений

10.2. Надежность операций

- **Подтверждение критических команд:** Ожидание MISSION_ACK при загрузке
- **Контроль последовательности:** Проверка seq номеров при загрузке точек
- **Таймауты операций:** Защита от зависания при ожидании сообщений
- **Обработка исключений:** Перехват всех возможных ошибок

10.3. Защита от сбоев

- **Валидация данных:** Проверка всех входных параметров
- **Обработка ошибок связи:** Корректная реакция на потерю соединения
- **Безопасное завершение:** Гарантированное закрытие соединения при любых ошибках
- **Информирование оператора:** Четкие сообщения об ошибках для принятия решений

Приложение А: Конфигурация тестового маршрута

Базовая станция (Home):

Координаты определяются автоматически из текущей позиции дрона в SITL

Маршрут патрулирования (3 точки):

Точка 0 (Базовая станция):

- Координаты: Текущая позиция дрона
- Высота: 30 м
- Тип: MAV_CMD_NAV_WAYPOINT
- current: 1 (текущая точка миссии)

Точка 1 (Береговая наблюдательная):

- Смещение от базы: 400 м север, 300 м восток
- Высота: 30 м
- Тип: MAV_CMD_NAV_WAYPOINT
- Назначение: Контроль береговой линии

Точка 2 (Контрольная точка патрулирования):

- Смещение от базы: 50 м север, 1350 м восток
- Высота: 30 м
- Тип: MAV_CMD_NAV_WAYPOINT
- Назначение: Наблюдение за акваторией, ожидание команд

Пример вычисления координат:

Для базовой точки: lat=56.461625°, lon=61.6037964°

Точка 1:

- Расстояние: $\sqrt{400^2 + 300^2} = 500$ м
- Азимут: $\arctan(300/400) = 36.87^\circ$
- Координаты: вычисляются через `geod.fwd()`

Точка 2:

- Расстояние: $\sqrt{50^2 + 1350^2} \approx 1351$ м
- Азимут: $\arctan(1350/50) = 87.87^\circ$ (почти на восток)
- Координаты: вычисляются через `geod.fwd()`

Приложение Б: Протокол загрузки миссии

Последовательность MAVLink сообщений:

1. **WAYPOINT_CLEAR_ALL:** Очистка старой миссии
2. **WAYPOINT_COUNT:** Отправка количества точек (3)

3. **MISSION_REQUEST_INT (seq=0):** Запрос точки 0 от автопилота
4. **MISSION_ITEM_INT (seq=0):** Отправка точки 0 (базовая станция)
5. **MISSION_REQUEST_INT (seq=1):** Запрос точки 1 от автопилота
6. **MISSION_ITEM_INT (seq=1):** Отправка точки 1 (береговая)
7. **MISSION_REQUEST_INT (seq=2):** Запрос точки 2 от автопилота
8. **MISSION_ITEM_INT (seq=2):** Отправка точки 2 (контрольная)
9. **MISSION_ACK:** Подтверждение успешной загрузки

Приложение В: Формат MISSION_ITEM_INT

```
MAVLink_mission_item_int_message(
    target_system,      # ID дрона (из master.target_system)
    target_component,   # ID автопилота (из master.target_component)
    seq,                # Порядковый номер точки (0, 1, 2)
    frame,              # MAV_FRAME_GLOBAL_RELATIVE_ALT
    command,            # MAV_CMD_NAV_WAYPOINT
    current,            # 1 для точки 0, 0 для остальных
    autocontinue,       # 1 (автоматическое продолжение)
    param1,             # 0 (не используется)
    param2,             # 0 (не используется)
    param3,             # 0 (не используется)
    param4,             # 0 (не используется)
    x,                 # Широта в degE7 (lat * 1e7)
    y,                 # Долгота в degE7 (lon * 1e7)
    z,                 # Высота в метрах (30.0)
)
```

Приложение Г: Ожидаемое поведение

Последовательность выполнения:

1. Запуск программы
2. Подключение к SITL (127.0.0.1:14550)
3. Ожидание HEARTBEAT сообщения
4. Вывод: "Подключено к системе X, компонент Y"
5. Получение текущей позиции из GLOBAL_POSITION_INT
6. Вывод: "Текущие координаты: lat=..., lon=..."
7. Расчет координат точек маршрута по смещениям
8. Формирование структуры миссии (3 точки)
9. Очистка старой миссии (WAYPOINT_CLEAR_ALL)
10. Отправка количества точек (WAYPOINT_COUNT)
11. Цикл загрузки: ожидание MISSION_REQUEST_INT → отправка MISSION_ITEM_INT
12. Получение MISSION_ACK
13. Завершение программы

Дальнейшие действия (выполняются другим кодом):

- Взлет дрона в режиме GUIDED
- Переключение в режим AUTO
- Выполнение миссии (полет по точкам 0 → 1 → 2)
- Ожидание команд в точке 2
- Возврат на базу по команде оператора

Приложение Д: Интеграция с системой управления

Данный модуль выполняет только **создание и загрузку миссии** патрулирования. Для полного цикла патрулирования требуется интеграция с дополнительными модулями:

Модуль взлета и запуска миссии:

- Arm двигателей
- Взлет в режиме GUIDED
- Переключение в режим AUTO
- Запуск выполнения миссии

Модуль мониторинга:

- Отслеживание прогресса миссии
- Мониторинг телеметрии (позиция, высота, батарея)
- Контроль достижения контрольной точки

Модуль возврата:

- Получение команды на возврат от оператора
- Переключение режима или загрузка новой миссии RTL
- Выполнение посадки на базе
- Disarm двигателей

Срок выполнения работ: 2 недели с момента утверждения ТЗ

Ответственный исполнитель: Отдел разработки систем управления БПЛА в лице Матофонова С.В.