



ПрофМир

Институт непрерывного образования
и профессионального развития

ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

“Python-разработка для БПЛА: паттерны проектирования, API-интерфейсы и библиотеки для оптимизации решений”

Преподаватель курса
Максимов Егор Васильевич

Выполнил
Атасян Сергей Мгерович

№ БПЛА_256-2

СОДЕЙСТВИЕ | **Федеральный**
ЗАНЯТОСТИ | **проект**

Москва 2024 г.

Проект по теме

«Создание полноценного Web-сайта и публикация его на GitHub»

Содержание:

1. Описание проекта, его основные характеристики, цели и область применения
2. Применяемые паттерны проектирования, обоснование их выбора
3. API-интерфейсы для обеспечения взаимодействия программного решения с внешними системами или сервисами
4. Библиотеки для оптимизации решений в проекте, объяснение их применения
5. Разработанная система тестирования ключевых компонентов проекта, включая отладку кода и обработку возможных ошибок
6. Анализ производительности проекта
7. Стратегии управления ресурсами и энергопотреблением для повышения эффективности работы программы
8. Документация к коду проекта
9. Демонстрация ключевой функциональности проекта

Описание проекта, его основные характеристики, цели и область применения

Актуальность темы:

В последнее десятилетие беспилотные летательные аппараты (БПЛА) нашли широкое применение в различных сферах жизни. Использование дронов становится всё более востребованным в сельском хозяйстве, логистике, мониторинге окружающей среды, в спасательных операциях, а так же в военных операциях и гражданской авиации. Это обуславливает необходимость разработки и постоянного усовершенствования качественного программного обеспечения для управления такими устройствами, что позволит повысить их эффективность, безопасность и автономность.

Программирование БПЛА — это перспективная область, требующая особых знаний и навыков. Она стала особенно популярной и актуальной темой на сегодняшний день.

Описание проекта, его основные характеристики, цели и область применения

Цель и задачи исследования:

Целью данной работы является разработка программного обеспечения для управления БПЛА с использованием современных алгоритмов автономного полета. В рамках этой цели решаются следующие задачи:

1. Проектирование архитектуры системы управления БПЛА.
2. Реализация программных модулей, обеспечивающих управление полетом, обработку данных с сенсоров и принятие решений.
3. Интеграция внешних API для получения данных и управления БПЛА.
4. Оптимизация и тестирование программного обеспечения с целью повышения его эффективности и надежности.

Описание проекта, его основные характеристики, цели и область применения

Обзор архитектуры системы:

Архитектура системы управления БПЛА представляет собой совокупность аппаратных и программных компонентов, взаимодействие которых позволяет осуществлять контроль над полетом дрона.

Основные компоненты системы:

- Сенсоры
- Исполнительные механизмы
- Блок управления
- Программное обеспечение



Описание проекта, его основные характеристики, цели и область применения

Определение целей проекта:

Цели проекта включают в себя разработку автономной системы управления БПЛА, способной выполнять задачи навигации, избегания препятствий и стабилизации полета. Важными аспектами являются обеспечение точности управления, устойчивости к внешним воздействиям и минимизация энергопотребления.

Область применения:

Разработанное программное обеспечение может быть использовано в различных отраслях. Например, наблюдение за движением автомобилей каршеринга, в сельском хозяйстве для мониторинга состояния посевов, в логистике для доставки малогабаритных грузов, в охранных системах для патрулирования территорий, а также в поисково-спасательных операциях.

Применяемые паттерны проектирования, обоснование их выбора

Применение паттерна MVC:

Паттерн **Model-View-Controller** (MVC) разделяет программное обеспечение на три компонента: модель, представление и контроллер. В контексте управления БПЛА:

- **Model (Модель):** Хранит данные о текущем состоянии БПЛА, такие как скорость, высота, координаты, уровень заряда батареи.
- **View (Представление):** Обеспечивает визуализацию данных и выводит информацию о состоянии дрона на экран пользователя. Например, отображение маршрута полета на карте.
- **Controller (Контроллер):** Управляет логикой взаимодействия между моделью и представлением, обрабатывает команды пользователя и обновляет данные в модели.

Применяемые паттерны проектирования, обоснование их выбора

Применение паттерна Observer:

Паттерн Наблюдатель (**Observer**) используется для реализации автоматического обновления системы при изменении данных. В данном проекте он может применяться для обработки данных с сенсоров в реальном времени. Когда сенсоры фиксируют изменение окружающей среды (например, обнаружение препятствия), это изменение автоматически передается системе управления, которая принимает соответствующие меры.

Применение паттерна State:

Паттерн Состояние (**State**) позволяет организовать различные режимы работы БПЛА, такие как взлет, круиз и посадка. Каждый режим (состояние) имеет свои особенности и логику, которые реализованы в отдельных классах.

API-интерфейсы для обеспечения взаимодействия программного решения с внешними системами или сервисами

Интеграция с внешними системами:

Для повышения функциональности БПЛА использован ряд внешних API.

API для получения данных о погодных условиях позволило адаптировать полет БПЛА к изменяющимся погодным условиям, и API для обмена данными с наземными станциями управления.

Библиотеки для оптимизации решений в вашем проекте, объяснение их применения

Использование библиотеки для обработки сигналов:

Для обработки сигналов с сенсоров можно использовать библиотеку для фильтрации данных, таких как **фильтр Калмана**. Это позволяет сгладить шумы и получить более точные данные для принятия решений.

Оптимизация вычислительных процессов:

Для оптимизации вычислительных процессов можно использовать многопоточность или параллельные вычисления. Например, библиотека `multiprocessing` в Python может быть использована для обработки данных с сенсоров в реальном времени.

Разработанная система тестирования ключевых компонентов проекта, включая отладку кода и обработку возможных ошибок

Разработка системы тестирования:

Для проверки правильности работы отдельных модулей и функций использовался модуль unittest в Python. Тестирование корректности работы алгоритмов навигации, модулей обработки данных с сенсоров.

Обработка ошибок и отладка:

Для обработки ошибок и отладки кода можно использовать встроенные механизмы обработки исключений. Это позволяет избежать некорректного поведения программы в случае непредвиденных ситуаций, например, при потере связи с GPS.

Анализ производительности проекта

Анализ производительности системы:

Оценка производительности включает в себя тестирование времени отклика системы на команды управления, точности навигации, скорости обработки данных с сенсоров и других критически важных параметров.

Использовались встроенные средства Python, такие как `timeit`, для измерения времени выполнения отдельных функций.

Влияние оптимизаций:

После внедрения оптимизаций проведен повторный анализ производительности и сравнились результаты с исходными данными. Было оценено, насколько сократилось время отклика системы после внедрения многопоточности.

Стратегии управления ресурсами и энергопотреблением для повышения эффективности работы программы

Оптимизация энергопотребления:

Одной из задач программного обеспечения для БПЛА является минимизация энергопотребления. Это было достигнуто за счет оптимизации алгоритмов управления, уменьшения частоты обработки данных, использования энергосберегающих режимов для оборудования.

Управление ресурсами:

Ресурсы, такие как процессорное время и память, должны использоваться эффективно. Это было достигнуто путем рационального распределения задач, использования ленивой загрузки данных и кэширования часто используемых данных.

Документация к коду проекта

Структурированная документация:

Каждая часть кода сопровождается подробной документацией, описывающей её назначение, входные и выходные данные, а также принципы работы. Это облегчает дальнейшую поддержку и развитие проекта.

Комментарии и инструкции:

Код содержит комментарии, объясняющие работу сложных алгоритмов или нетривиальных решений. Так же подготовлены инструкции для развертывания и запуска ПО на реальном устройстве.

Демонстрация ключевой функциональности проекта

```
State: Disarmed  
releaseApiControl was successful  
Vehicle is disarmed  
Vehicle is already armed  
Collision Count:0  
requestApiControl was successful  
Vehicle is armed
```





ПрофМир

Институт непрерывного образования
и профессионального развития

Благодарю за внимание!

СОДЕЙСТВИЕ | Федеральный
ЗАНЯТОСТИ | проект

info@eduom.ru