



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт	ЭнМИ
Кафедра	РМДиПМ

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

Направление	15.03.06 Мехатроника и робототехника
	(код и наименование)

Образовательная программа	Компьютерные технологии управления в робототехнике и мехатронике
---------------------------	--

Форма обучения	очная
	(очная/очно-заочная/заочная)

Тема:	Разработка и программирование автономного квадрокоптера
-------	---

Студент	С-126-19	Котов Д.Д.
	группа	подпись
		фамилия и инициалы

Руководитель ВКР	к.т.н.	доцент	Капустина О.М.
	уч. степень	должность	подпись
			фамилия и инициалы

организация

«Работа допущена к защите»

Заведующий кафедрой	д.т.н.	профессор	Меркурьев И.В.
	уч. степень	звание	подпись
			фамилия и инициалы

Дата

Москва, 2023



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт	<u>ЭнМИ</u>
Кафедра	<u>РМДиПМ</u>

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
(БАКАЛАВРСКУЮ РАБОТУ)**

Направление	<u>15.03.06 Мехатроника и робототехника</u>
	<small>(код и наименование)</small>

Образовательная программа	<u>Компьютерные технологии управления в робототехнике и мехатронике</u>
----------------------------------	---

Форма обучения	<u>очная</u>
	<small>(очная/очно-заочная/заочная)</small>

Тема:	<u>Разработка и программирование автономного квадрокоптера</u>
--------------	--

Студент	<u>С-126-19</u>	<u>Котов Д.Д.</u>
	<small>группа</small>	<small>подпись</small>
		<small>фамилия и инициалы</small>

Руководитель ВКР	<u>к.т.н.</u>	<u>доцент</u>	<u>Капустина О.М.</u>
	<small>уч. степень</small>	<small>должность</small>	<small>подпись</small>
			<small>фамилия и инициалы</small>

	<small>организация</small>			
Заведующий кафедрой	<u>д.т.н.</u>	<u>профессор</u>	<u>Меркурьев И.В.</u>	
	<small>уч. степень</small>	<small>звание</small>	<small>подпись</small>	<small>фамилия и инициалы</small>

Место выполнения работы	<u>ФГБОУ ВО «НИУ «МЭИ»</u>
--------------------------------	----------------------------

СОДЕРЖАНИЕ РАЗДЕЛОВ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

1. Постановка задачи

2. Описание используемых систем координат и краткое описание принципа работы квадрокоптера

3. Подбор и обоснование выбора комплектующих

4. Создание 3D модели, изготовление деталей и сборка прототипа

5. Настройка прототипа и полетного контроллера

6. Описание используемых инструментов для программирования

7. Описание алгоритма управления

8. Тестирование алгоритма в симуляторе Gazebo

9. Перенос программы и тестирование в реальности

ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА

Количество листов _____

Количество слайдов в презентации _____

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. K. P. Valavanis, G. J. Vachtsevanos. Handbook of Unmanned Aerial Vehicles (UAVs). - Springer, 2014

2. Т. Килби. Make: дроны с нуля: Пер. с англ./ Т. Килби, Б. Килби. - СПб.: БХВ-Петербург, 2016. -192 с.: ил.

3. YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, TaeHoon Lim. ROS robot programming. - ROBOTIS Co.,Ltd, 2017

Примечания:

1. Задание брошюруется вместе с выпускной работой после титульного листа (страницы задания имеют номера 2, 3).
2. Отзыв руководителя, рецензия(и), отчет о проверке на объем заимствований и согласие студента на размещение работы в открытом доступе вкладываются в конверт (файловую папку) под обложкой работы.

АННОТАЦИЯ

В рамках работы рассматривается разработка прототипа БПЛА мультироторного типа, с возможностью автономного полета, на основе имеющихся электронных и электро-механических компонентов. Подобраны комплектующие и создана аппаратная конфигурация, собрана физическая модель БПЛА. Обоснован выбор программной архитектуры и инструментов разработки ПО. Произведены тестовые вылеты, подтверждающие работоспособность собранного прототипа.

ABSTRACT

Within the framework of the work, the development of a prototype of a multirotor type UAV, with the possibility of autonomous flight, based on existing electronic and electro-mechanical components is being considered. Components were selected and a hardware configuration was created, a physical model of the UAV was assembled. The choice of software architecture and software development tools is justified. Test flights were carried out, confirming the operability of the assembled prototype.

ВВЕДЕНИЕ	6
1.1. Задачи работы	11
1.2. Краткое описание систем координат квадрокоптера	11
1.3. Краткое описание принципа работы квадрокоптера	14
Глава 2: Подбор комплектующих	16
2.1. Рама и плата питания.....	17
2.2. Винтомоторная группа	18
2.2.1. Двигатели	19
2.2.2. Регуляторы оборотов	19
2.2.3. Пропеллеры	20
2.2.4. Расчет подъемной тяги	21
2.3. Аккумулятор	22
2.4. Полетный контроллер	23
2.5. Бортовой компьютер	24
2.6. Пульт и передатчик	26
2.7. Модуль GPS	26
2.8. Силовые модули	27
2.9. Виброгаситель.....	28
Глава 3: Создание 3D-модели и сборка	29
3.1. Создание 3D-модели.....	29
3.2. Сборка прототипа	32
Глава 4: Настройка прототипа	34
4.1. Первичная настройка.....	34
4.2. ПИД-регулятор	37
4.3. Настройка полуавтономных режимов полета.....	39
Глава 5: Разработка ПО.....	45
5.1. Структура управления.....	45
5.1.1. Автопилот ArduCopter	46
5.1.2. Фреймворк Robot Operating System	47
5.1.3. Протокол MAVLink	48
5.1.4. Пакет MAVROS	50
5.2. Алгоритм управления автономным полетом квадрокоптера	51
5.3. Тестирование работы алгоритма в симуляторе Gazebo.....	52
5.4. Перенос ПО на бортовой компьютер и тестирование прототипа.....	55
ЗАКЛЮЧЕНИЕ	58
Библиографический список	59
Приложение А	63

ВВЕДЕНИЕ

Беспилотный летательный аппарат (БПЛА) - это летательный аппарат без экипажа на борту. За последние 10 лет БПЛА мультироторного типа приобрели огромную популярность. Эти аппараты используются для видео- и фото-съемки, орошения сельскохозяйственных угодий и других задач, связанных с облетом определенной территории. Управляет дроном в большинстве случаев либо оператор, либо наземная станция, с которой поддерживается постоянная связь. Это создает лишние траты денежных средств и времени. Решением данной проблемы является разработка автономного дрона с бортовым компьютером.

Рынок автономных БПЛА

Классифицировать БПЛА можно по огромному количеству признаков, например, по конструктивной схеме БПЛА, лётно-эксплуатационным и взлетно-посадочным характеристикам, по степени автономности и т.д. Будем рассматривать только БПЛА мультироторного типа с 4-мя винтами - квадрокоптер.

Существует несколько уровней автономности БПЛА, приведенные на Рис.0-1. Для обеспечения автономности в состав робототехнической системы должны входить исполнительные органы, датчики, системы управления и навигации вместе с соответствующими алгоритмами обработки данных, планирования и выполнения действий, управления целями.

Уровень автономии	0	1	2	3	4	5
Вовлеченность человека						
Вовлеченность дрона						
Степень автономности	Нет	Низкая	Частичная	Условная	Высокая	Полная
Описание	Управление дроном полностью осуществляется оператором.	Пилот практически полностью контролирует процесс работы. Дрон отвечает только за "жизненно" важные для его работы функции	Оператор контролирует безопасность дрона. Дрон может удерживать положение итд.	Оператор выступает в качестве "резервной системы"	Оператор практически не контролирует работу дрона.	Дрон полностью выполняет свои функции без оператора.

Рис. 0-1 Степени автономности БПЛА

На данный момент существует огромное количество квадрокоптеров совершенного различного размера и форм-фактора. Но среди всех них можно выделить тех, чей уровень автономности больше или равен 3. Это позволяет таким дронам выполнять полетные миссии без участия человека.

Среди современных автономных квадрокоптеров можно выделить следующие образцы:

- «Геоскан Gemini» от компании ГЕОСКАН [1] (Рис.0-2)



Рис. 0-2. квадрокоптер Геоскан Gemini

Этот квадрокоптер предназначен для выполнения аэросъемочных работ с получением высокоточных пространственных данных, например, для

получения цифровой модели местности или рельефа, ортофотоплана местности или для картографических работ. Аппаратная база этого квадрокоптера к сожалению не позволяет устанавливать дополнительную полезную нагрузку или датчики.

- «Matrice 300» от компании DJI [2] (Рис.0-3)



Рис. 0-3. квадрокоптер Matrice 300

Это один из самых популярных квадрокоптеров на данный момент, который используется для поисково-спасательных операций, аэросъемочных работ, инспекций ЛЭП и трубопроводов. Matrice 300 может выполнять миссию автономно с полезной нагрузкой до 2.7 кг в течении 50 минут. Программное обеспечение позволяет добиться точности позиционирования 1 см, при помощи использования метода определения относительных координат RTK[3] и алгоритмов навигации по изображению с камер. Также дрон оснащен вспомогательным освещением и системой предотвращения столкновений[4]. Коптер поддерживает только официальные модули полезной нагрузки от компании DJI.

- Пеликан от COEX[5] (Рис.0-4)



Рис. 0-4. квадрокоптер Пеликан от COEX

Дрон оснащен автоматической системой зарядки. Если во время полета заряд аккумулятора снизится до установленного порогового значения, дрон приземлится на зарядную станцию и самостоятельно выполнит заряд аккумулятора. После этого дрон продолжит миссию в стандартном режиме. Также рама данного квадрокоптера позволяет нести полезную нагрузку, например, дополнительные камеры или мелкие грузы.

Все вышеперечисленные дроны объединяет одна особенность - на их борту установлен микрокомпьютер, позволяющий выполнять высокоуровневые задачи, например, планирование и управление действиями, построение маршрута или управление полезной нагрузкой. Например, бортовой компьютер используется для планирования маршрута с учетом препятствий, сбора и обработки данных с датчиков, выступающих в роли полезной нагрузки, построения 3D карт местности или рельефа и многих других задач.

Также вышеперечисленные квадрокоптеры не позволяют использовать собственноручно изготовленные модули и программное обеспечение для них непосредственно с установленным на борту микрокомпьютером. Если пользователю необходим нестандартный модуль, его придется заказывать у компании изготовителя, что обычно требует огромных денежных средств, либо вовсе неосуществимо. Таким образом теряется универсальность. Решить данную проблему можно с помощью установки универсального бортового

компьютера, который был бы открыт для индивидуальных модулей, а также выполнял задачи высокоуровневого управления квадрокоптером.

Актуальность темы исследования.

Задача разработки автономных коптеров становится все более актуальной с каждым годом, но при этом автономность все еще остается открытым вопросом. Люди все в большей степени стараются автоматизировать некоторые процессы, опасные или трудновыполнимые для человека, а также усовершенствовать и оптимизировать их посредством БПЛА. Также недавно появился опыт применения квадрокоптеров для дезинфекции улиц[6]. Кроме того, согласно инфографике[7], приведенной на Рис.0-5. рынок БПЛА стремительно растет и развивается.

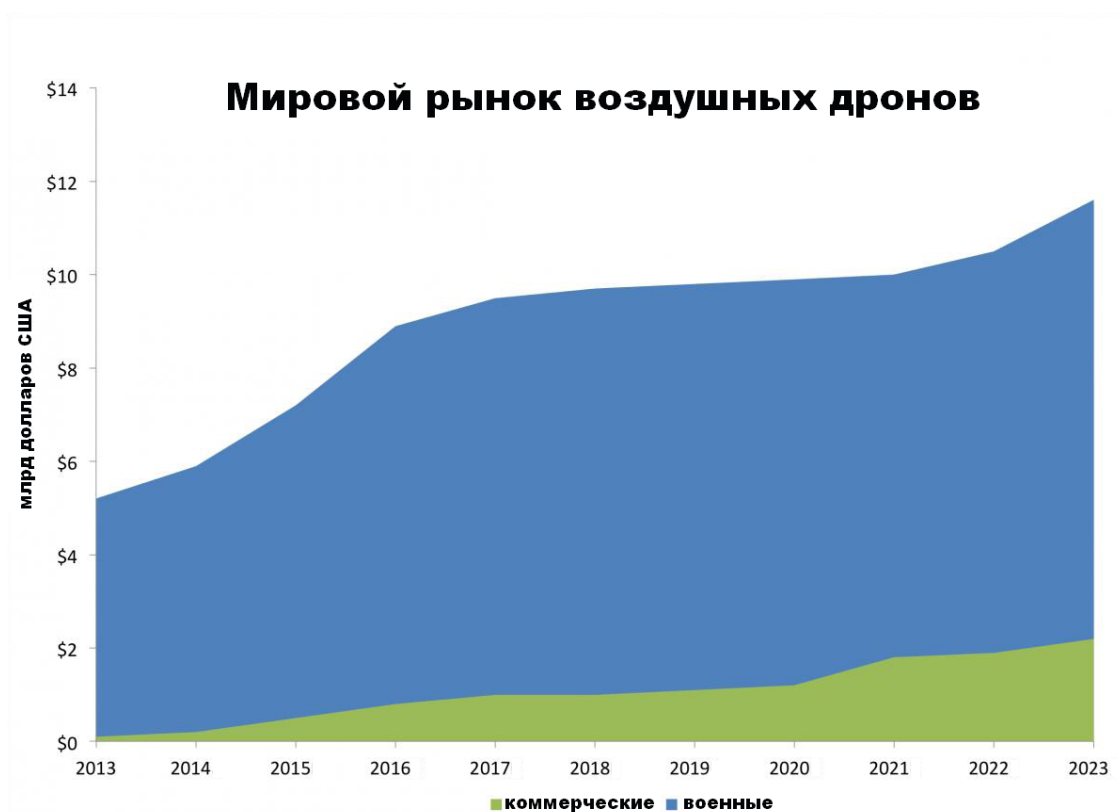


Рис. 0-5. прогноз рынка дронов в 2022

Новизна данного проекта заключается в разработке функционального прототипа и ПО к нему, таким образом, чтобы дрон мог автономно выполнять полетные задачи, а также обеспечивал возможность дальнейшего расширения функционала аппарата.

Глава 1: Постановка задачи

1.1. Задачи работы

В рамках работы рассматривается разработка БПЛА мультироторного типа на основе имеющихся электронных компонентов и механических деталей, а также разработка ПО для обеспечения автономного полета и обеспечивающего возможность управления полезной нагрузкой и расширения функционала дрона.

Таким образом для выполнения работы требуется:

- Описание принципа работы квадрокоптера и систем координат;
- Выбор оптимального набора компонентов, совместимых друг с другом и обеспечивающих возможность автономного полета;
- Создание 3d-модели и сборка;
- Настройка прототипа;
- Разработка алгоритма управления автономным полетом, его реализация и тестирование.

1.2. Краткое описание систем координат квадрокоптера

При работе БПЛА использует несколько систем координат, сокращенно именуемые СК. СК, расположенная в точке взлета, СК БПЛА, связанная СК и промежуточные СК[8]. Рассмотрим их более подробно.

При движении коптера оси СК в точке взлета остаются неподвижными и жестко связанными с Землей. Принято, что ось $X_{НСК}$ данной системы координат направлена на Север, ось $Y_{НСК}$ на Восток, а ось $Z_{НСК}$ направлена в центр Земли. Для краткости назовем данную систему координат НСК (от Неподвижная Система Координат). Данная СК изображена на Рис.1-1.

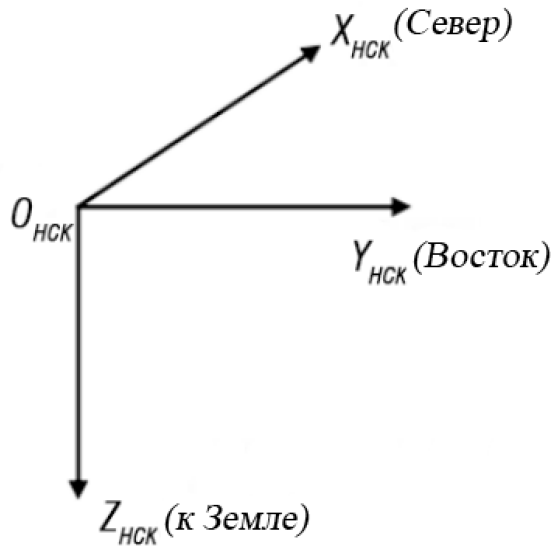


Рис. 1-1. Система координат, связанная с Землей

Центр отсчета СК БПЛА совпадает с центром тяжести БПЛА. Для краткости будем называть эту систему координат СК-1. Эта система координат получается путем параллельного переноса НСК и является системой Кенига. Связь между СК-1 и НСК представлена на Рис.1-2.

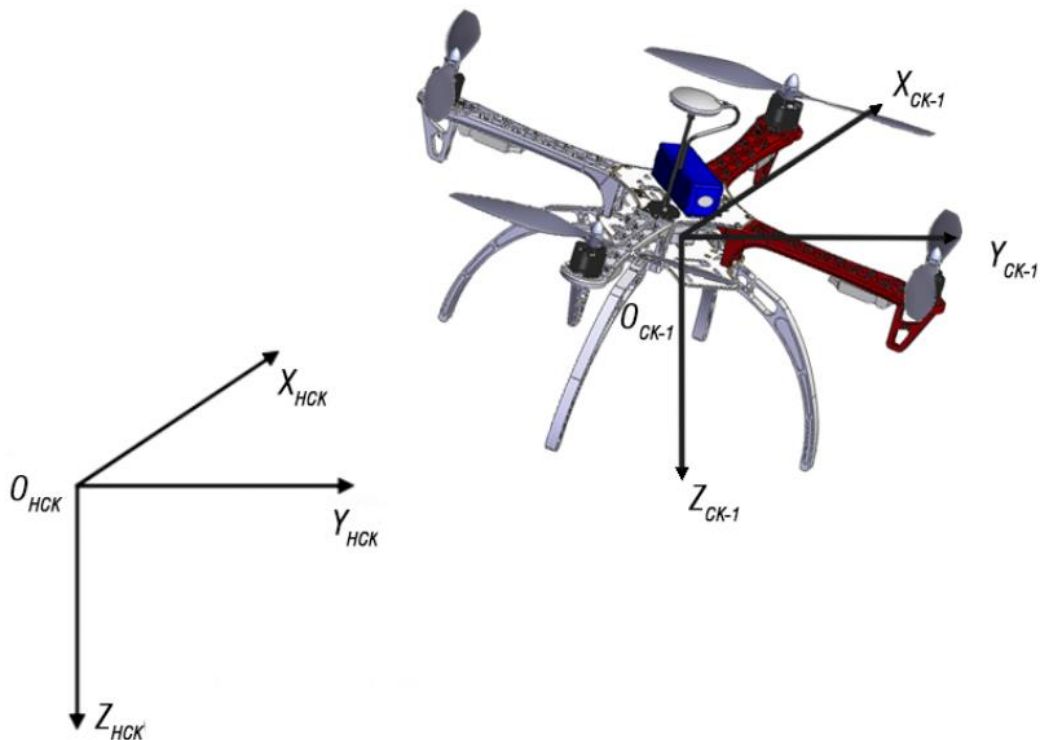


Рис. 1-2. Кенигова система координат для БПЛА.

Связанная система координат жестко связана с БПЛА. Она получается путем вращения вокруг осей промежуточных систем координат.

Сначала вращением вокруг оси $Z_{СК-1}$ СК-1 на угол ψ получается СК-2 (Рис. 1-2.).

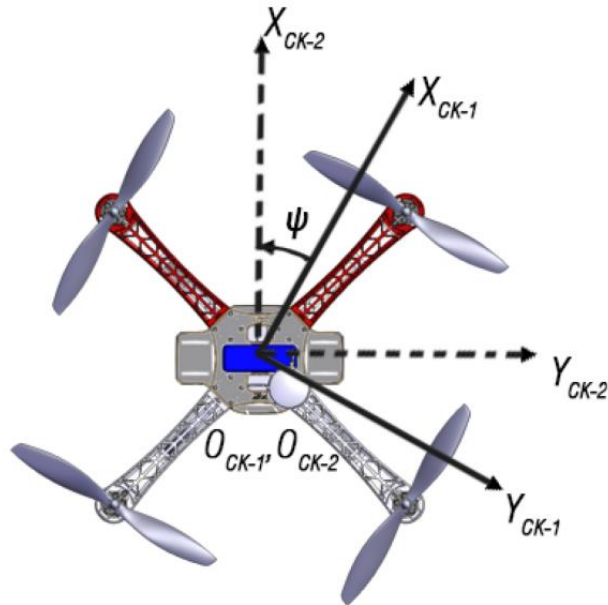


Рис. 1-2. Промежуточная система координат СК-2

Затем - вокруг оси $Y_{СК-2}$ СК-2 на угол θ получается СК-3 (Рис. 1-3.).

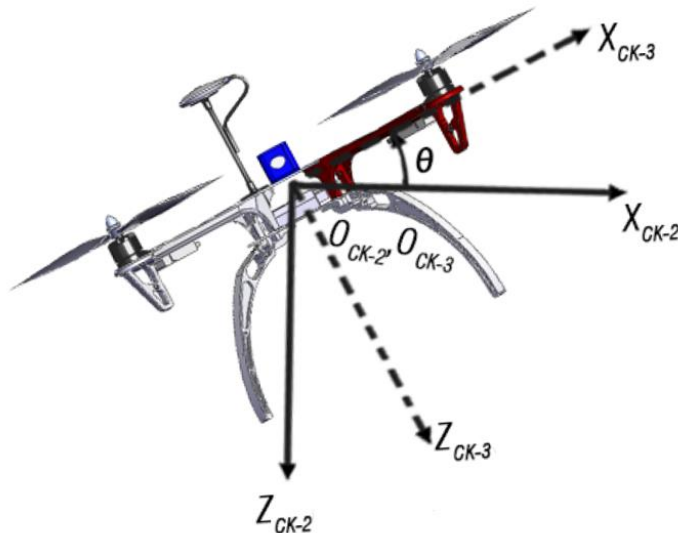


Рис. 1-3. Промежуточная система координат СК-3

И наконец - вокруг оси $X_{СК-3}$ на угол φ получается связанная система координат БПЛА - ССК (Рис. 1-4.).

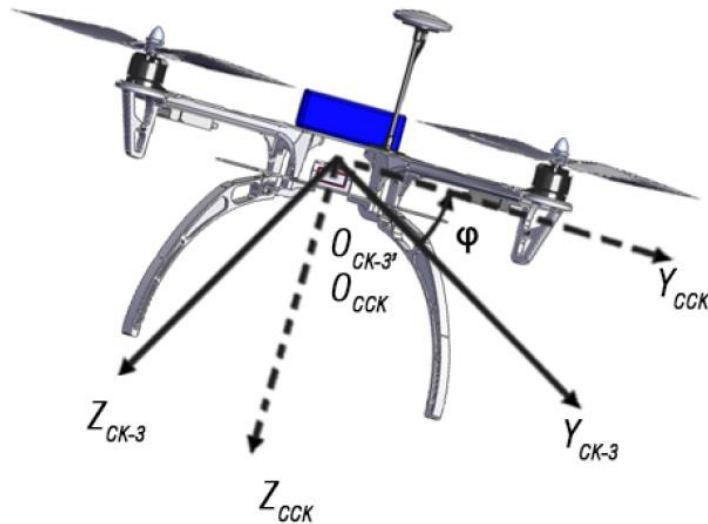


Рис. 1-4. Связанная система координат квадрокоптера

Эти углы образуют углы Эйлера-Крылова. Они задают ориентацию квадрокоптера в пространстве.

1.3. Краткое описание принципа работы квадрокоптера

Принцип работы квадрокоптера заключается в том, что изменением скорости вращения каждого из пропеллеров можно изменять создаваемую тягу. Это позволяет управлять положением и ориентацией квадрокоптера в пространстве. Создаваемые силы(F) и моменты(M) показаны на Рис.1-5. Обозначения f , b , r , l - соответственно перед, зад, право, лево. Это необходимо для правильного обозначения сил и моментов, например, F_f - сила создаваемая передним винтом итд.

Поворот по крену - это поворот аппарата вокруг его продольной оси, той, которая проходит от носа до хвоста, на определенный угол. Поворот по тангажу - это поворот вокруг его поперечной оси. Поворот по курсу или рысканью - поворот вокруг вертикальной оси. Например, при одновременном увеличении угловой скорости всех пропеллеров, коптер будет совершать движение вверх относительно геометрической оси Z . При увеличении скорости вращения левого и уменьшении правого винтов коптер совершает поворот в сторону «положительного крена». Аналогично с «положительным

тангажем». Для положительного поворота по курсу или по углу рысканья, необходимо увеличить скорость вращения правого и левого винтов, при одновременном уменьшении скорости переднего и заднего винтов.

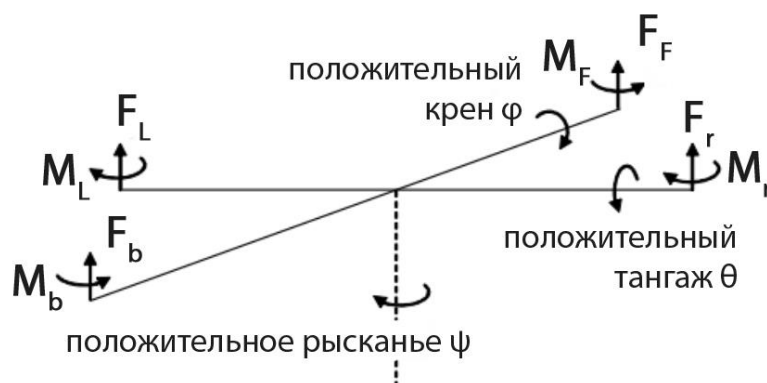


Рис. 1-5. Силы и моменты, создаваемые двигателями квадрокоптера

Среди основных частей квадрокоптера можно выделить раму, двигатели, регуляторы скорости и полетный контроллер. Речь о всех этих компонентах пойдет далее. Полетный контроллер это «мозги» дрона. Он строит управление для двигателей, обрабатывает информацию с датчиков для определения ориентации и положения в пространстве, принимает и обрабатывает сигналы исходящие от оператора. Кроме того, к функциям полетного контроллера относятся: стабилизация аппарата в воздухе; удержание высоты при помощи барометрического высотомера или иных датчиков и позиции при помощи GPS/GLONASS; автоматический полёт по заданным заранее точкам; передача на землю текущих параметров полёта; обеспечение безопасности полёта. Все алгоритмы управления реализуются программно и записаны в прошивке полетного контроллера.

Глава 2: Подбор комплектующих

Для обеспечения необходимой степени автономности и возможности расширения функционала не хватит одного лишь полетного контроллера, который управляет двигателями и занимается расчетом управления. Чтобы повысить вычислительную мощность дрона необходимо разработать более сложную аппаратную и программную архитектуру.

Для решения этой проблемы было принято решение использовать трехуровневую архитектуру, состоящую из бортового компьютера, полетного контроллера и исполнительных устройств квадрокоптера. Микрокомпьютер выполняет роль контроллера верхнего уровня, полетный контроллер - нижнего(контроллера реального времени), двигатели и датчики, контактирующие непосредственно с полетным контроллером - полевого уровня. Преимущество такого решения будет рассмотрено позже.

В процессе разработки было принято решение не разрабатывать с нуля каждый из компонентов, а использовать готовую электронику и остальные составляющие. Такое решение ускорит разработку, а также позволит создать более гибкую и универсальную систему.

Исходя из поставленных целей, можно составить список необходимых деталей. Также необходимо учесть факторы популярности, а следовательно обратной совместимости, качества, стоимости и надежности. Приведем список необходимых составляющих:

- Рама и плата питания
- Аккумулятор
- BLDC двигатели, регуляторы скорости(ESC) и винты
- Полетный контроллер
- Бортовой компьютер
- Пульт и передатчик (для ручного управления)
- Модуль GPS
- Силовые модули для питания логической электроники

В результате определения необходимых деталей была составлена аппаратная конфигурация, приведенная на Рис. 2-1.

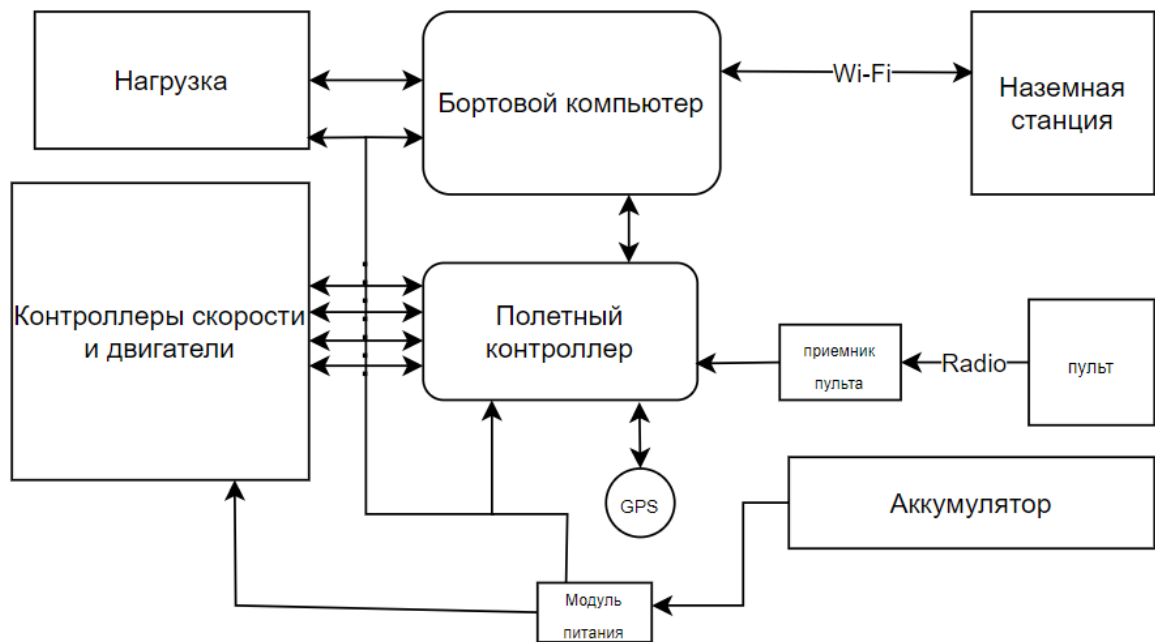


Рис. 2-1. Аппаратная архитектура прототипа

Рассмотрим каждую из составляющих более подробно.

2.1. Рама и плата питания

Одной из самых стандартных и популярных рам является DJI 450, приведенная на Рис. 2-2.



Рис. 2-2. Рама DJI F450

Данная рама имеет следующие характеристики:

- Расстояние по диагонали: 450 мм
- Масса: 282 грамма
- Полетный вес: ≤ 1600 граммов

Рама состоит из 4-х лучей для крепления моторов, 4-х стоек для устойчивой стоянки, верхней и нижней пластины.

Дешевизна и огромная популярность делают эту раму легкодоступной, а надежность и размеры позволяют дрону летать практически в любой среде. Кроме того, конструкция позволяет в случае поломки заменить только испорченный луч, каждый из которых крепится к верхней пластине 4-мя винтами 2.5х6 мм и 2-мя такими же винтами к нижней пластине.

Еще одним преимуществом является то, что плата питания разведена на нижней пластине рамы. Это позволяет сократить количество проводов, уменьшает вес несущей конструкции и упрощает сборку.

2.2. Винтомоторная группа

Винто-моторная группа (ВМГ) квадрокоптера состоит из 4-х бесколлекторных синхронных двигателей (BLDC), 4-х регуляторов контроля скорости (ESC) и 4-х пропеллеров. Двигатели создают вращающий момент, пропеллеры создают тягу, а ESC необходимы для управления двигателями. Так как принцип работы BLDC основан на переменном токе, необходимо преобразовать постоянный ток в трехфазный переменный. Именно этим и занимаются ESC, они строят управление на основе поступающего от полетного контроллера сигнала, и подают на каждую из обмоток двигателя соответствующий ток.

Определим составляющие ВМГ, которые будут совместимы с рамой и с друг другом[9]:

- Бесколлекторный синхронный двигатель Ready to Sky 2212 920KV
- ESC (electronic speed controller) фирмы Ready to sky 30A

- Комплект самозатягивающихся нейлоновых пропеллеров длиной 10 дюймов с шагом винта 4.5 дюйма.

2.2.1. Двигатели

Выбранный двигатель приведен на Рис.2-3.



Рис. 2-3. Бесколлекторный двигатель ReadyToSky 2212 920KV

Он обладает следующими характеристиками:

- Рабочее напряжение - 7.4...11.1В
- Максимальный ток - 12А
- Число оборотов на вольт(KV) - 920
- Вес (с гайкой) - 56 г

Характеристика KV или RPM/V означает количество оборотов, на которое увеличится скорость вращения двигателя при увеличении подачи питающего напряжения на 1 Вольт. Конкретное значение KV зависит от конструкции двигателя, включая количество обмоток и магнитов. Значение KV может использоваться для выбора подходящего винта для модели или для настройки ESC.

2.2.2. Регуляторы оборотов

Регуляторы оборотов (ESC) необходимы для управления скоростью вращения моторов квадрокоптера. Регуляторы Ready to sky 30A приведены на Рис.2-4.



Рис. 2-4. Электронный регулятор скорости(ESC) Ready To Sky 30 А
Характеристики ESC:

- Размер: 45х24х8мм
- Масса: 25г
- Пиковый ток: 40А

Ток, на который рассчитаны регуляторы, превышает пиковый ток потребления двигателей более чем вдвое. Этот запас позволит не сгореть ESC в случае непредвиденной ситуации, а если регулятор выходит из строя, то, как правило, выходит из строя и двигатель. Также еще одной причиной выбора такого регулятора является их огромная популярность и дешевизна.

2.2.3. Пропеллеры

Пара правого и левого винта размером 10 дюймов с шагом 4.5 дюйма приведена на Рис. 2-5.



Рис. 2-5. Пара винтов 10*4.5

Эти самозатягивающиеся винты. Такая особенность делает их легкими и быстроснимаемыми. Масса каждого из винтов составляет 12 грамм.

2.2.4. Расчет подъемной тяги

Существует эмпирический вывод[10], который гласит, что для получения необходимой маневренности и управляемости квадрокоптера максимальная тяга, создаваемая винтами, должна быть минимум вдвое больше общего веса БПЛА.

Максимальная полетная масса, на которую рассчитана рама, составляет 1600 грамм, от этой массы и будем отталкиваться. Каждый двигатель должен создавать тягу минимум 400 грамм просто для удержания собственного веса.

Определим подъемную силу, создаваемую каждым винтом. Связь между используемой винтомоторной группой, напряжением и силой тяги приведена в таблице 2-1.

Таблица 2-1

Значения силы тяги для BLDC Ready to Sky 2212 920 KV

Двигатель	Напряжение, В	Пропеллер, длина*шаг, дюйм	Мощность работы двигателя, %	Тяга, г
RTS 2212/920KV	11.1	10*4.5	50	290
			65	460
			75	590
			85	730
			100	860

Для выбранной винтомоторной группы видим, что максимальная сила тяги равняется 860 г., что превышает необходимые 400 г. в 2.15 раз. Можно сделать вывод, что выбранные комплектующие обеспечат квадрокоптер необходимой маневренностью и управляемостью.

2.3. Аккумулятор

В качестве источника энергии был использован литий-полимерный аккумулятор (Li-Po). Преимущества аккумуляторов таких типов:

- Высокая токоотдача
- Большая емкость
- Малый вес и габариты относительно других видов аккумуляторов

Однако данный тип аккумуляторов обладает рядом минусов, а именно:

- Непереносимость переразряда или перезаряда
- Чувствительность к температуре
- Быстрое старение
- Взрывоопасность, легкая возгораемость при повреждениях

Для квадрокоптера важна минимальная масса при максимальной токоотдаче и емкости. Li-Po один из немногих типов аккумуляторов, которые удовлетворяют данным условиям.

В сборке используется аккумулятор со следующими характеристиками:

- Емкость: 5000 mAh
- Номинальное напряжение секции: 3.7 В
- Конфигурация: 3S (3 последовательных соединения секций)
- Токоотдача: 50C
- Масса: 380 г.

Внешний вид батареи показан на Рис.2-6.



Рис. 2-6. Аккумулятор Li-Po 3S 5000mAh 50C

Параметр токоотдачи является одним из самых важных показателей при выборе аккумулятора. Он показывает предельно допустимый ток разряда. Определим токоотдачу выбранного аккумулятора в более удобных для восприятия Амперах. Предельный выходной ток получается путем умножения емкости батареи на параметр токоотдачи[11]. В результате получаем: $5000 * 50 = 250 \text{ А}$. Это значение намного больше, чем суммарный максимальный потребляемый двигателями ток в 48 А, следовательно аккумулятор подходит.

2.4. Полетный контроллер

Полетный контроллер - это микросхема, оснащенная инерциальным измерительным блоком(IMU) и предназначенная для управления двигателями дрона. Измерительный блок включает в себя датчики линейного ускорения (акселерометр), угловой скорости (гироскоп), барометр и магнитометр. Основной задачей датчиков на полетном контроллере является непрерывное получение навигационных данных с блока IMU и внешней периферии для дальнейших математических расчетов микропроцессором, который устанавливает и корректирует положение беспилотника в соответствии с заданным управлением.

Также необходимо, чтобы полетный контроллер имел возможность подключать периферийные устройства, такие как модуль GPS и телеметрию.

В качестве полетного контроллера был выбран APM 2.8, поддерживающий прошивку ArduCopter 3.4.1, которая разработана для мультироторных БПЛА. Внешний вид контроллера представлен на Рис.2-7.

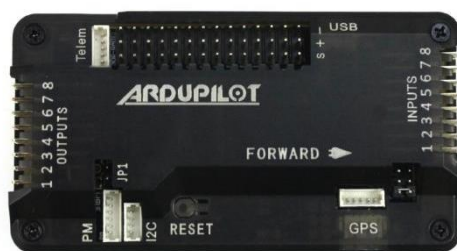


Рис. 2-7. Полетный контроллер APM2.8

Этот полетный контроллер имеет разъемы для подключения дополнительных датчиков и электроники, а также сочетает в себе простоту использования. Помимо этого АРМ является платформой с открытым исходным кодом и огромным сообществом. Все это упрощает работу над проектом и делает его более универсальным и гибким.

АРМ 2.8 имеет следующие характеристики:

- Акселерометр/гироскоп: Invensense 6 DoF MPU-6000
- Высокоточный барометр: MS5611-01BA03
- Память для автоматического логирования: 4MB
- Главный чип: Atmel ATMEGA2560
- Размеры основного блока: 70 x 44 x 14 мм
- Вес: 34 г

Прошивка ArduCopter обладает огромным количеством полезных функций и параметров для настройки. Все это делает эту платформу очень гибкой и широко-настраиваемой. Также ПО данного полетного контроллера поддерживает протокол передачи данных MAVLink, предназначенный для обмена данными между наземной станцией и микролетательными аппаратами. Это означает, что данный полетный контроллер может использоваться для автономных миссий, без участия человека, а также, что с АРМ можно установить связь с ROS через пакет MAVROS. Речь об этом пойдет далее.

2.5. Бортовой компьютер

Для реализации автономного полета было принято решение использовать бортовой компьютер. Суть данного подхода заключается в том, что маршрут и параметры полета сохраняются на бортовом компьютере. Во время полета бортовой компьютер постепенно передает необходимые данные полетному контроллеру, который на основе полученной информации строит управление. Это расширяет спектр возможных применений квадрокоптера, а также обеспечивает большую автономность. Кроме того, наличие бортового компьютера позволит совершать полет без связи с наземной станцией.

Одним из наиболее популярных и универсальных микрокомпьютеров является Raspberry Pi 3b+(далее Rpi), вид которого приведен на Рис.2-8. Данная плата поддерживает Ubuntu и ROS, обладает достаточной мощностью для поддержки функционирования автономного полета БПЛА, а также для обработки данных, получаемых с полезной нагрузки и управления ей.

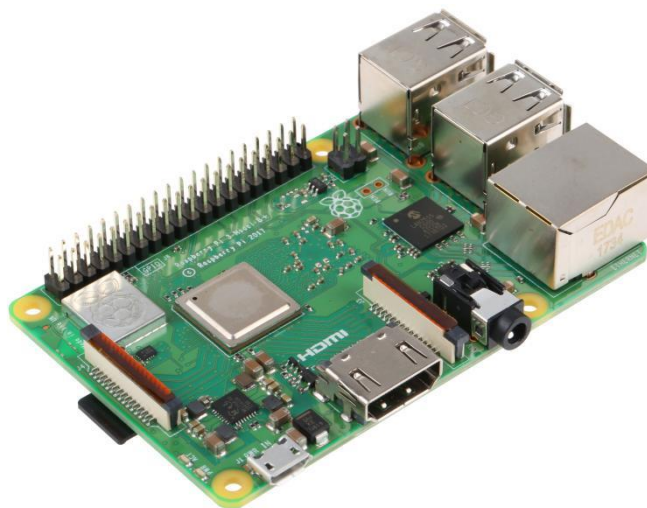


Рис. 2-8. Одноплатный микрокомпьютер Raspberry Pi 3

Краткая сводка о технических характеристиках:

- Процессор: Cortex-A53, 4 ядра
- Оперативная память: 1 ГБ, LPDDR2 SDRAM
- Интегрированная графика: VideoCore IV 3D
- Wi-Fi: IEEE 802.11 b/g/n 2.4GHz Wireless LAN (WLAN)
- Масса: 45 г.

Универсальность использования Raspberry Pi 3 заключается в наличии широких коммуникационных возможностей, обеспечиваемыми модулями Wi-Fi, и Bluetooth, 40-контактной линией ввода общего назначения(GPIO), четырех портов USB 2.0. Raspberry Pi 3 предназначена для универсального использования во многих встраиваемых системах, также этот микрокомпьютер обладает открытым исходным кодом и огромным сообществом разработчиков и пользователей, что делает разработку проекта проще и доступнее.

2.6. Пульт и передатчик

Для ручного управления был подобран пульт FS-i6 от компании FlySky и приемник FS-IA6B от той же компании, которые приведены на Рис.2-9.



Рис. 2-9. Пульт FS-i6 и приемник FS-IA6B

Наличие пульта и приемника обеспечивает возможность перехода на ручное управление в нештатной ситуации, что повышает безопасность использования дрона.

2.7. Модуль GPS

Для полетов на открытой местности целесообразно использовать GPS модуль. Это позволяет осуществлять более точное следование заданной траектории. Одним из наиболее популярных GPS модулей является Ublox M8N, приведенный на Рис.2-10.



Рис. 2-10. Внешний модуль GPS(компас) Ublox M8N

2.8. Силовые модули

Один из силовых модулей необходим для корректной работы полетного контроллера и двигателей. Он разделяет питание, подаваемое на полетный контроллер и подключенную к нему одометрию, и подаваемое к регуляторам оборотов, а также позволяет полетному контроллеру отслеживать уровень напряжения аккумулятора.

JMT APM 2.6, приведенный на Рис.2-11., является рекомендованным для серии полетных контроллеров APM. Характеристики этого модуля приведены ниже:

- Максимальное входное напряжение: 25 в
- Выдаваемое напряжение: 5В
- Выдаваемый ток: 2А



Рис. 2-11. Модуль питания JMT APM 2.6.

Для питания бортового компьютера необходимо использовать отдельный модуль питания, так как Raspberry Pi 3 может использовать 2-3 А при работе. Был выбран модуль питания UBEC - регулятор внешнего импульсного источника питания, изображенный на Рис. 2-12.



Рис. 2-12. Модуль питания для Raspberry.

Данный модуль принимает 5-23 В и выдает 5 В с током 3А, этого вполне хватает для полноценной работы микрокомпьютера.

2.9. Виброгаситель

Микроэлектромеханические системы крайне чувствительны к колебаниям, поэтому полетный контроллер необходимо зафиксировать на виброгасящей платформе, выполненной из двух пластин стекловолокна и 4 резиновых прокладок. Внешний вид представлен на Рис.2-13.



Рис. 2-13. Виброгасящая платформа.

Глава 3: Создание 3D-модели и сборка

3.1. Создание 3D-модели

По блок-схеме аппаратной конфигурации построим 3D-модель для дальнейшей разработки индивидуальных деталей, предназначенных для крепления электроники. Перед проектированием были выдвинуты функциональные требования к креплениям и другим индивидуальным деталям. Они должны быть легкими и прочными. Основным способом достижения этой цели - уменьшение веса, поэтому по возможности необходимо изготовить детали из пластика. При этом они должны вмещать как можно больше электроники на себе. Это необходимо для уменьшения момента инерции всего дрона. Достичь этого можно путем изготовления элементов сложной формы, поэтому рациональным решением является использование технологий 3D-печати.

Имеет смысл располагать все детали как можно ближе к геометрическому центру квадрокоптера, так как это уменьшает момент инерции, а следовательно экономит энергию[12].

Так как аккумулятор является самой массивной частью коптера, расположим его между верхней и нижней пластинами, практически в центре рамы. Обычно батарея устанавливается прямо на верхней или нижней пластине. Для удержания аккумулятора обычно используются специальные ленты с «липучками», но так как его расположение не совсем стандартное, было принято решение спроектировать специальный кейс, приведенный на Рис.3-1., удерживающий батарею на месте и обеспечивающий неподвижность относительно рамы и безопасность при полете. Крепится кейс будет с помощью лент с липучками, они обеспечивают надежное закрепление кейса на месте.

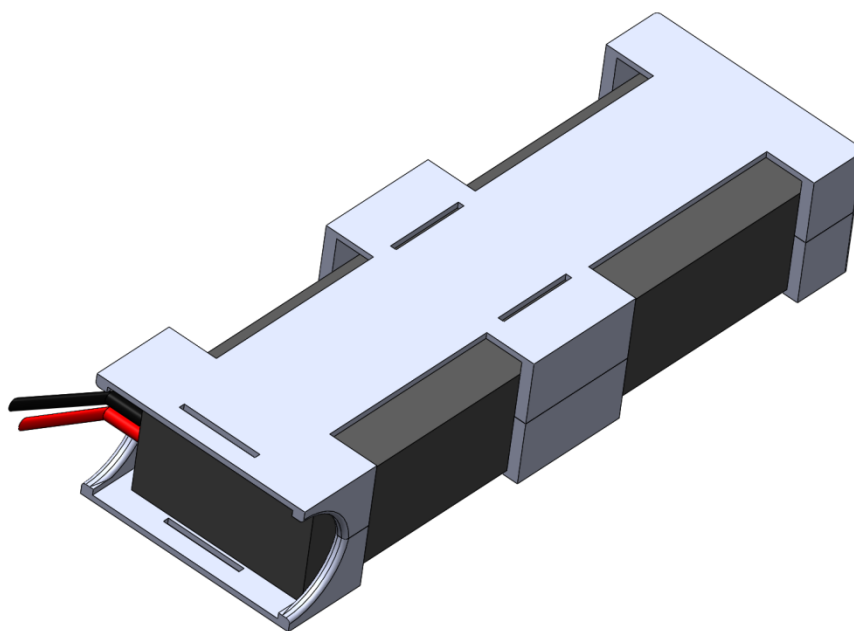


Рис. 3-1. Кейс для аккумулятора.

Так как обычно вся полезная нагрузка располагается снизу, имеет смысл там же разместить бортовой компьютер. Это позволит избавиться от лишних проводов, сделать конструкцию более компактной и в таком состоянии сохраняется «наглядность» подключения. Для закрепления Rpi была спроектирована пластина, крепящаяся к нижней части квадрокоптера. Ее вид приведен на Рис.3-2. Для соединения крепления, Raspberry Pi и квадрокоптера использовались латунные стойки 6*2.5 мм.

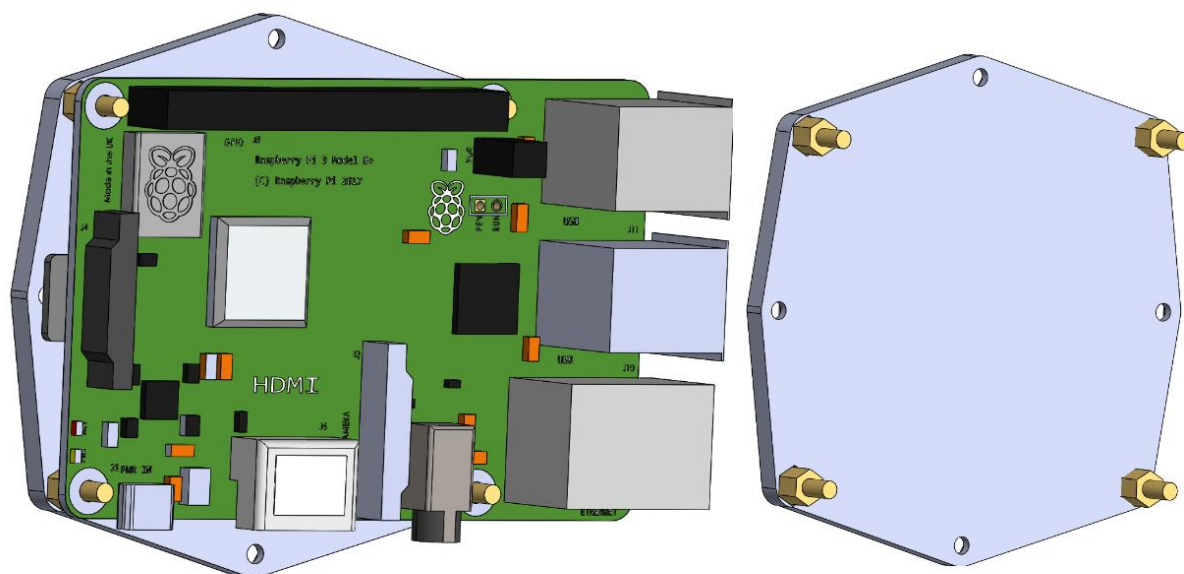


Рис. 3-2. Крепление для Raspberry Pi. Вид с микрокомпьютером и без.

Оставшуюся электронику можно разместить только на верхней пластине. Для закрепления всех деталей нет смысла проектировать отдельную деталь, так как все комплектующие надежно крепятся на раме без дополнительных элементов конструкции. Полетный контроллер крепится к виброгасителю, приемник с помощью стяжек крепится к переднему левому лучу рамы. Использование стяжек обусловлено тем, что они очень легкие и обеспечивают необходимую надежность крепления. Модуль GPS закрепляется с помощью болтов на специальном луче, который отдаляет компас от магнитных полей всей электроники коптера. Это сделано, потому что GPS крайне чувствителен к магнитным полям[12].

Итоговая 3D модель представлена на Рис.3-3.

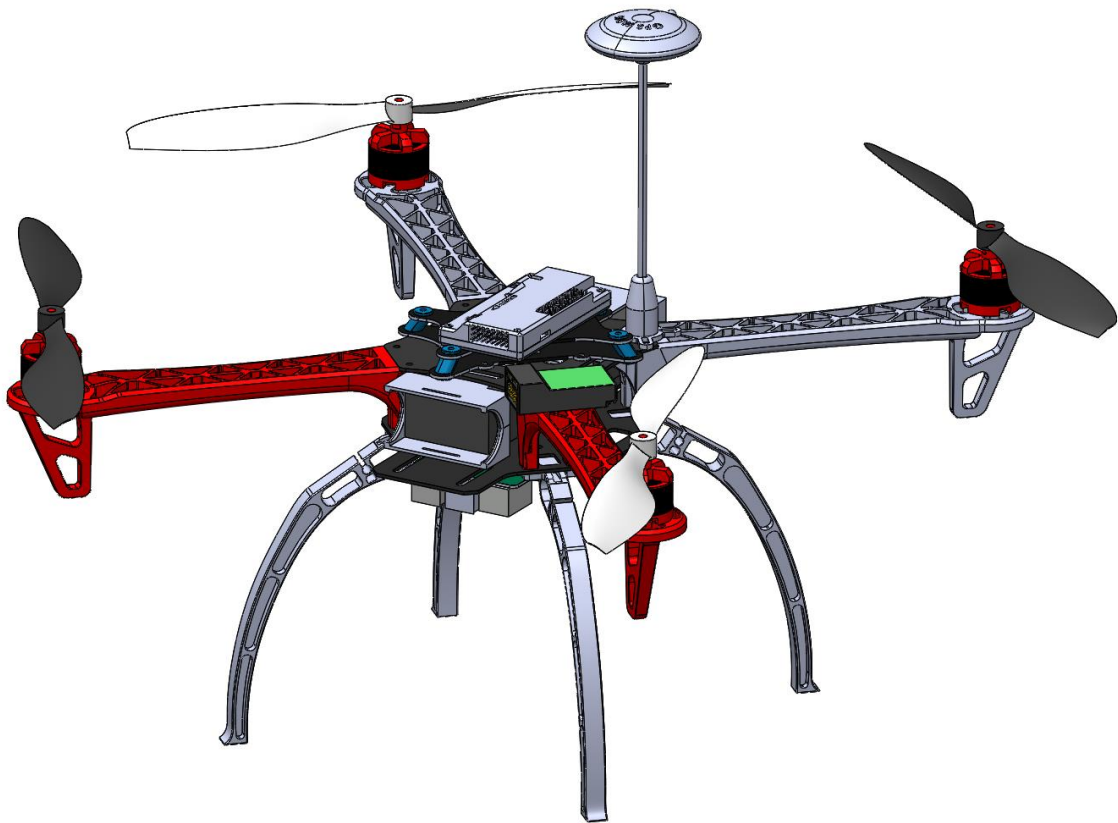


Рис. 3-3. 3D модель квадрокоптера в сборке.

3.2. Сборка прототипа

Для начала необходимо изготовить индивидуальные детали. Они изготавливались на 3D-принтере методом FDM печати из материала PLA[13].

После процесса пайки, сборки и компоновки элементов получаем следующий результат. Фотографии собранного прототипа приведены на Рис. 3-4. и Рис.3-5.



Рис. 3-4. Собранный прототип квадрокоптера, вид сбоку.

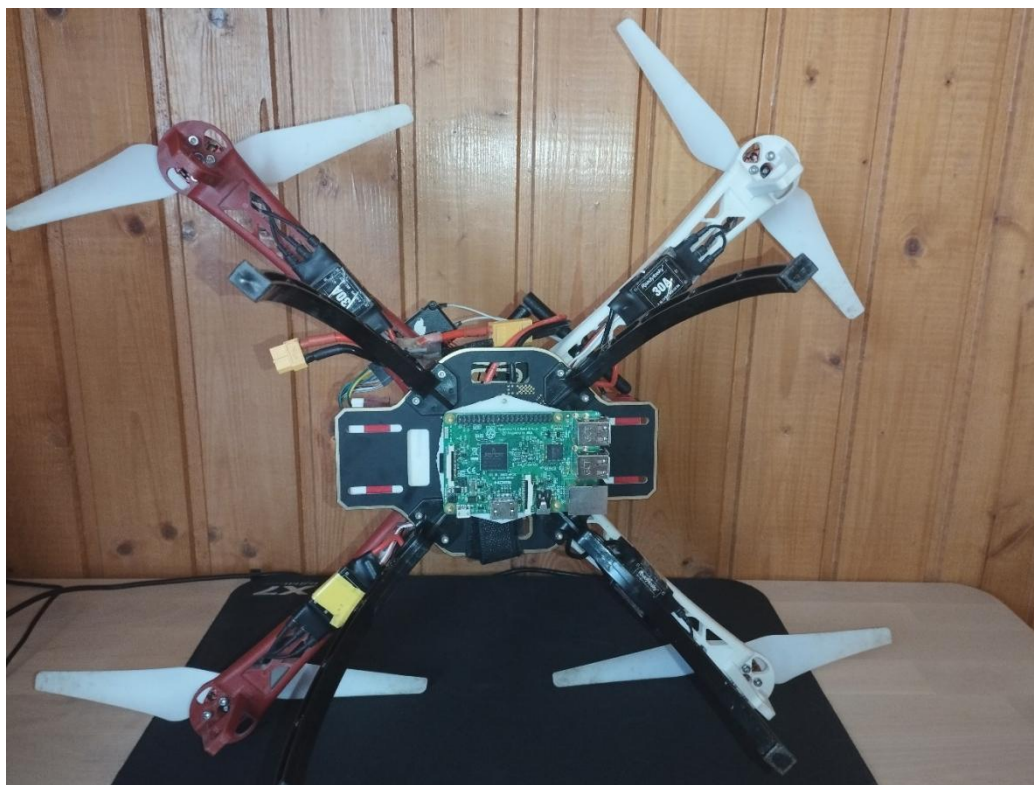


Рис. 3-5.Собранный прототип квадрокоптера, вид снизу.

Глава 4: Настройка прототипа

4.1. Первичная настройка

Первым шагом к настройке полетного контроллера является установка приложения Mission Planner версии 1.3.45, которое является универсальной утилитой для настройки полетного контроллера и планирования автономных миссий, с официального сайта Ardupilot. Это наиболее стабильная версия приложения для полетного контроллера APM2.8. Вся настройка полетного контроллера будет происходить через это приложение. На Рис.4-1. показано рабочее окно приложения. Mission Planner также называют наземной станцией. Обычно наземная станция - это приложение, которое запущено на компьютере, находящемся на поверхности Земли, и взаимодействует с БПЛА по беспроводной связи. Наземная станция выводит информацию о дроне в реальном времени, а также может использоваться для управления БПЛА в полете.

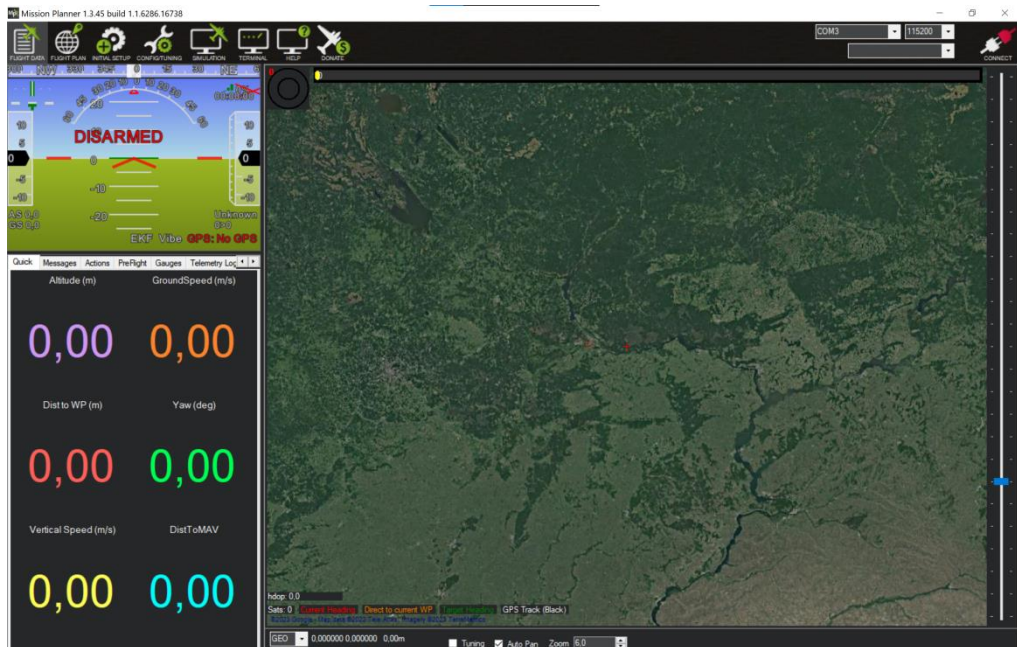


Рис. 4-1. Рабочее окно программы Mission Planner.

После этого необходимо подключить контроллер к компьютеру и загрузить прошивку ArduCopter 3.2.1 - последнюю версию прошивки для

мультикоптеров, поддерживаемая контроллерами серии APM. Далее производится первичная настройка[14]. Она состоит из следующих шагов:

1. Выбор конфигурации БПЛА

Параметры конфигурации рамы должны быть установлены таким образом, чтобы они соответствовали реально используемой модели. За это отвечают параметры FRAME_CLASS и FRAME_TYPE, которые в нашем случае устанавливаются как Quad и QuadX соответственно.

2. Калибровка акселерометра

Добиться идеально точной установки полетного контроллера на раме квадрокоптера невозможно, всегда будет небольшое отклонение в одну из сторон. Для корректного определения ускорений по каждой из осей независимо от расположения устройства необходимо провести калибровку[15]. Во время калибровки полетный контроллер определяет смещения и программно компенсирует их, чтобы оси измерений акселерометра были ортогональны. После проведения калибровки удалось добиться корректности показаний.

3. Калибровка GPS

Калибровка компаса необходима для корректного определения истинного севера, а также магнитных смещений и отклонений. Так модуль GPS будет показывать позицию намного точнее, чем без калибровки. Это очень важный шаг, так как точность полета крайне важна.

4. Калибровка ESC

Калибровка ESC нужна для правильной работы двигателей и в общем случае она зависит от используемой марки регуляторов, но существует универсальный способ настройки, подходящий практически всем регуляторам, в том числе фирмы ReadyToSky. Этот способ называется «All at once»[16].

5. Настройка функций FAILSAFE

Автопилот оснащен рядом механизмов предотвращения аварий и блуждания в случае потери управления или выхода из строя компонентов дрона[17], имеющий название FAILSAFE. Для настройки FAILSAFE нужно

выбрать алгоритм действий для каждой из ситуаций: низкий заряд батареи, потеря управляющего сигнала(с пульта, наземной станции или бортового компьютера), неисправность одного из датчиков телеметрии. Для всех событий было выбрано действие посадки в месте нахождения(переход в режим полета RTL).

6. Проверка вибраций

МЭМС и в том числе акселерометры крайне чувствительны к вибрациям и высокочастотным колебаниям. Чрезмерные вибрации приводят к значительным ошибками вычисления местоположения по данным блока IMU. Это крайне негативно сказывается на управлении квадрокоптером. Дрон начинает вести себя нестабильно и непредсказуемо, что может привести к аварии. Все это касается как ручных, так и автоматических\полуавтоматических режимов полета. Для определения необходимо проанализировать показания блока инерциальной измерительной системы в логах одного из полетов в режиме STABILIZE. Лог это файл с данными телеметрии квадрокоптера. График ускорений по осям X, Y, Z ССК показан на Рис.4-2. Как видно из графика значения ускорений по осям X и Y не превосходят по модулю 3, а значения ускорений по Z лежат в пределах -5 - -15 м/с². Эти значения являются допустимым пределом для показаний акселерометра согласно официальной документации ArduPilot[18]. Следовательно виброгаситель справляется со своей задачей.

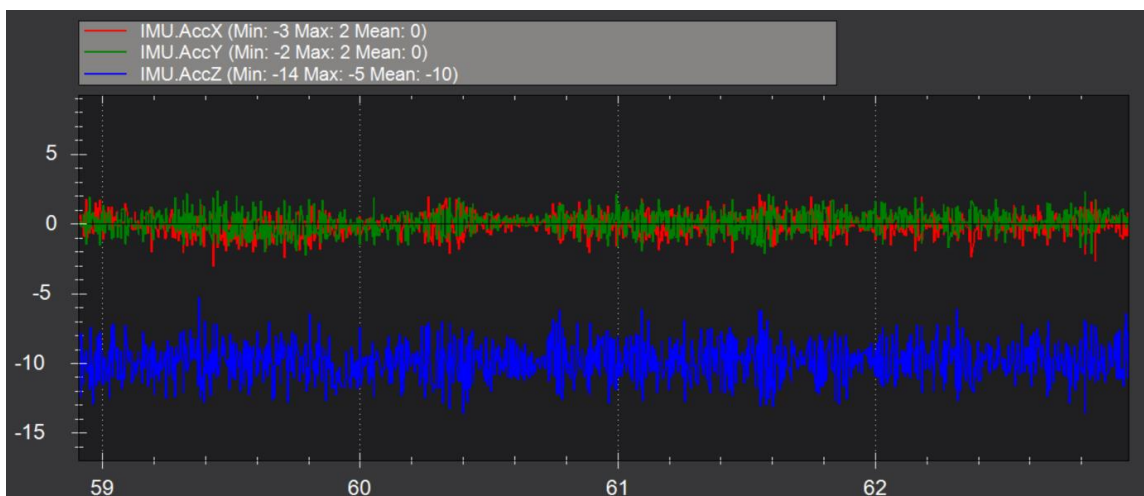


Рис. 4-2. Значения ускорений осям X, Y, Z.

4.2. ПИД-регулятор

Пропорционально-интегрально-дифференциальный (ПИД) регулятор - устройство в управляющем контуре с обратной связью. За счет своей универсальности этот регулятор широко используется в технических системах, где необходимо автоматическое управление. Общая блок-схема ПИД-регулятора приведена на рис.4-3.

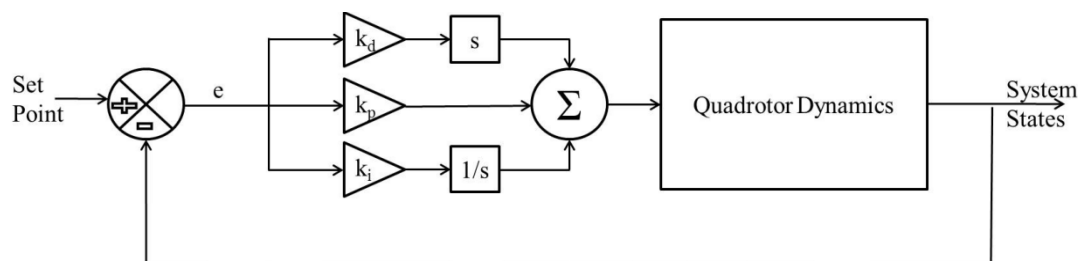


Рис. 4-3. Структурная схема ПИД-регулятора.

Уравнение для формирования управляющего сигнала выглядит следующим образом:

$$U(t) = k_p \cdot e(t) + k_i \cdot \int_0^t e(\tau) d\tau + k_d \cdot \frac{d(e(t))}{dt}$$

, где $U(t)$ - формируемое управление, $e(t)$ - ошибка регулирования, $k_{P,I,D}$ - коэффициенты пропорциональной, интегральной и дифференциальной связи соответственно.

В дроне ПИД-регулятор выполняет задачу стабилизации положения во время полета. Регулятор сравнивает значения, полученные с датчиков, с требуемыми значениями и строит управляющий сигнал. Прошивка Arducopter имеет структуру ПИД-регулятора[19], приведенную на Рис.4-4.

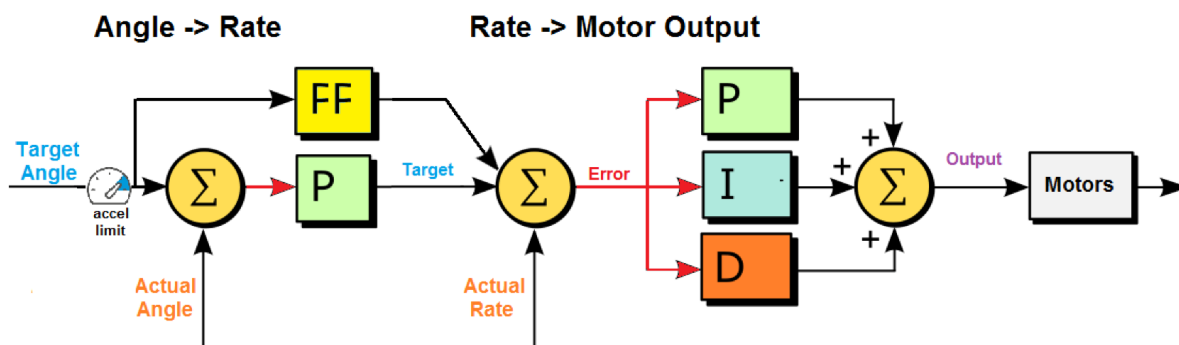


Рис. 4-4. Структура ПИД-регулятора автопилота.

Данный алгоритм используется для регулирования крена, тангажа и рысканья, а также линейных и угловых скоростей квадрокоптера. Пропорциональная обратная связь по углу преобразует разницу между желаемым и фактическим углом в желаемую скорость вращения, за которой следует PID-регулятор для преобразования ошибки скорости вращения в команду двигателя высокого уровня.

Помимо настройки параметров регулятора по углу и угловой скорости в Mission Planner можно настроить параметры для некоторых полетных режимов, как это приведено на Рис.4-5.



Рис. 4-5. Параметры ПИД-регуляторов, которые можно настроить.

На этом же рисунке приведены стандартные параметры ПИД-регуляторов, используемые для рам 450 мм. После выполнения пробного полета с ручным управлением, было выявлено, что квадрокоптер ведет себя устойчиво и плавно. Полет был произведен 29 апреля. Все полеты, включая дальнейшие были произведены в Московской Области. Также не было выявлено проблем с управлением, перерегулированием и колебаниями во время полета даже в ветренную погоду. Все это подтверждают графики желаемых и реальных углов тангажа и крена, приведенные на Рис.4-6 и Рис.4-7.

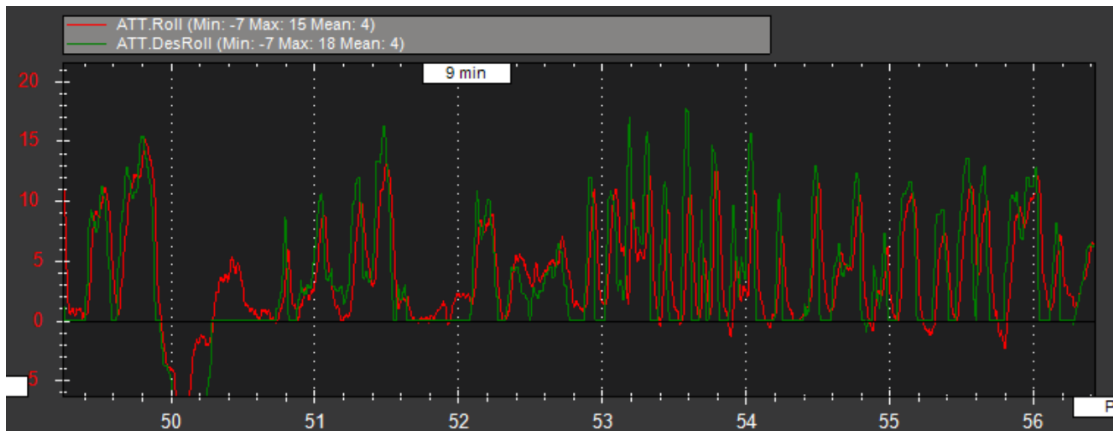


Рис. 4-6. Графики желаемых и реальных углов крена квадрокоптера. Зеленым - желаемый угол, красным - фактический.

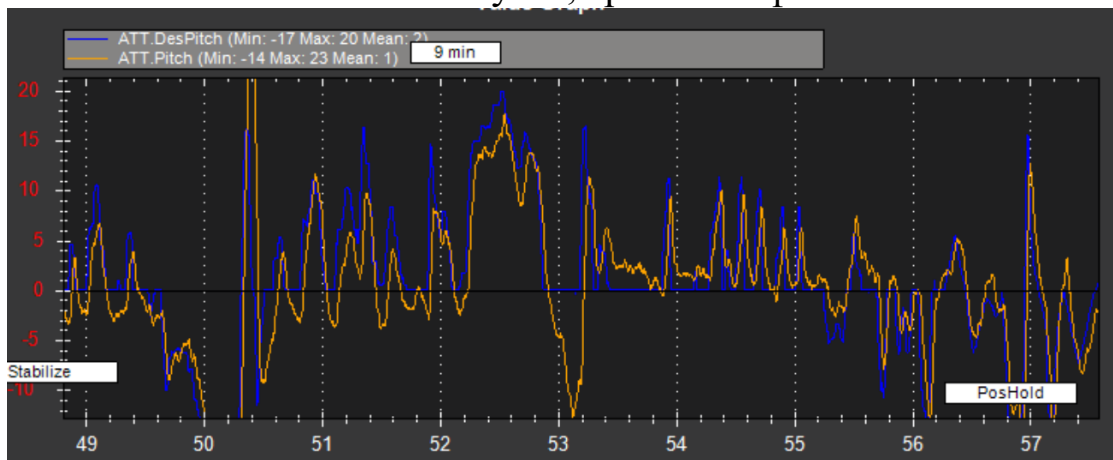


Рис. 4-7. Графики желаемых и реальных углов тангажа квадрокоптера. Синим- желаемый угол, желтым- фактический.

Таким образом, было принято решение оставить стандартные параметры ПИД-регуляторов, так как они обеспечивают желаемый управляемый полет.

4.3. Настройка полуавтономных режимов полета

Под полуавтономными подразумеваются режимы PosHold, Loiter и другие, которые обеспечивают висение в позиции по измерениям инерциальной навигационной системы и показаниям GPS. А также режим RTL (return to launch), который производит посадку на место взлета. При переключении на эти режимы оператор косвенно участвует в процессе управления, задавая только точку висения или корректируя подлет к месту взлета в критической ситуации.

Проблема их настройки заключается в том, что ошибки в предыдущих шагах критически сказываются на процессе управления. Например, из-за малой несимметричности конструкции или неправильной настройки GPS дрон начинает движение по расширяющейся спирали с нарастанием горизонтальной скорости. Это может привести к полной потере управления и падению.

Во время тестовых полетов, произведенных 6-7 мая, были выявлены проблемы с этими тремя режимами полета. На Рис.4-8. изображен фрагмент траектории полета, выделенной синим цветом, длительностью полторы минуты, на котором коптер должен удерживать позицию в режиме PosHold. При этом можно наблюдать колебательные движения около точки желаемого висения, что также показано на Рис.4-9. и Рис.4-10.

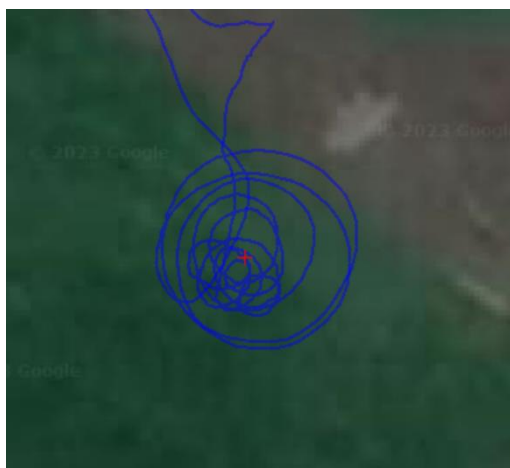


Рис. 4-8. Фрагмент полета в режиме PosHold. Красный крест ничего не обозначает



Рис. 4-9. Желаемая и реальная позиции по X в режиме PosHold. Красным - желаемая, зеленым - реальная.

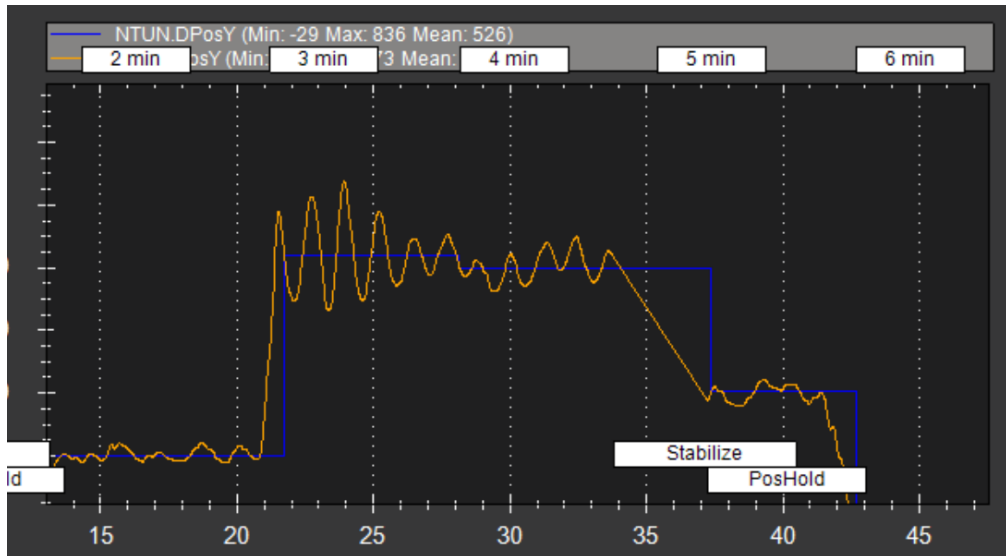


Рис.4-10. Желаемая и реальная позиции по Y в режиме PosHold. Синим - желаемая, желтым - реальная.

В режиме RTL наблюдалась аналогичная ситуация при попытках обеспечить висение над точкой взлета. Как результат, коптер садился с большой горизонтальной скоростью, и заваливался на бок.

Так как при тестировании параметров ПИД-регуляторов, не было выявлено проблем, то было выдвинуто предположение, что причина неправильного управления заключается в ложных показаниях GPS. Проанализировав показания GPS в логах полета, было обнаружено, что HDop (Horizontal Dilution Of Precision) имеет очень высокие значения[20], что показано на Рис.4-11. HDop характеризует ошибку определения местоположения, чем это значение выше - тем меньше точность GPS. Для решения данной проблемы необходимо провести продвинутую настройку.

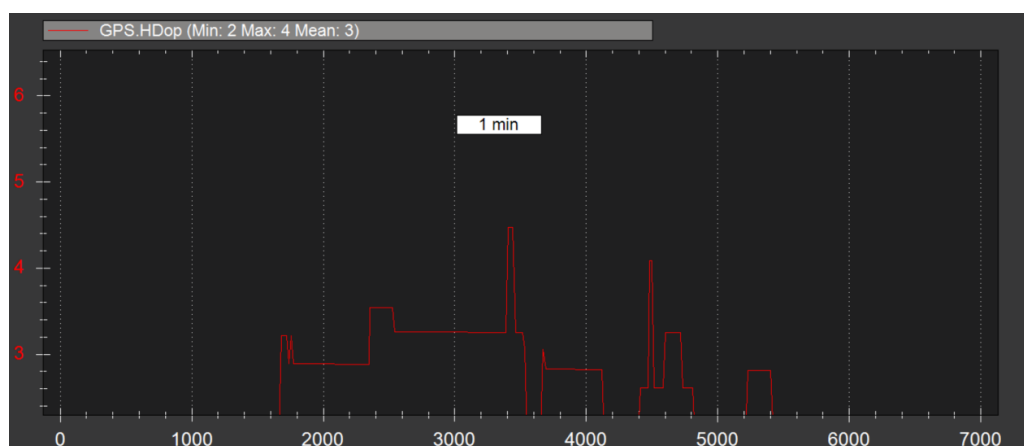


Рис. 4-11. Значения коэффициента потери точности определения горизонтального местоположения.

У каждой точки Земли есть свое магнитное склонение - отклонение физического северного полюса от истинного. Неправильные значения магнитного склонения, внесенные в полетный контроллер, приводят к ложным показаниям по координатам и курсу. Это в свою очередь приводит к расхождению реальной и желаемой точек, и пытаясь попасть в желаемую точку, квадрокоптер изменяет курс, но не может попасть в нужную точку из-за ложных показаний компаса. Это продолжается до тех пор, пока пилот не возьмет управление на себя, либо дрон не упадет. Магнитное склонение местности, в которой проводились эксперименты равняется $+12^{\circ} 4'$ [21]. Это значение было введено в параметр Magnetic Declination компаса. Кроме того, ошибка может быть вызвана магнитным полем, создаваемым двигателями во время работы. Исправить это можно учитывая создаваемое ими магнитное поле. Это делается с помощью калибровки Compass/Motor[22]. После вышеописанных операций 8 мая был проведен пробный вылет для определения точности полета и правильной работоспособности БПЛА. График положения БПЛА, выделенный синим цветом, в режиме висения в течении одной минуты приведен на Рис.4-12. Графики желаемых и реальных координат в сантиметрах от точки взлета в зависимости от времени в минутах показаны на Рис.4-13 и Рис.4-14. Показания HDop приведены на Рис.4-15 в виде зависимости ошибки, измеряемой в метрах, от времени в минутах .



Рис. 4-12. Полет в режиме PosHold после произведенной настройки.

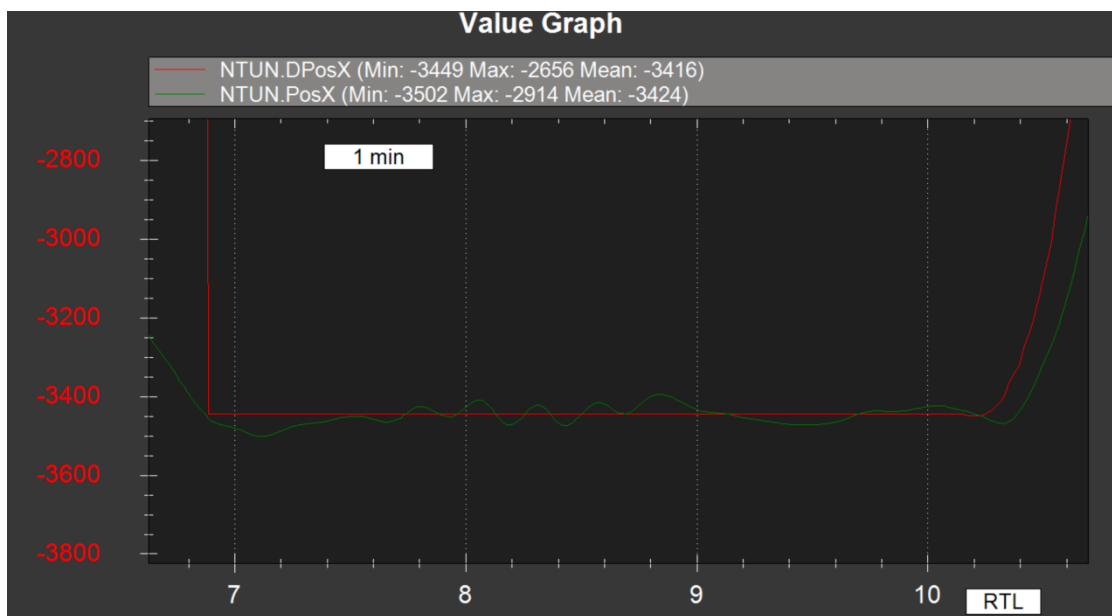


Рис. 4-13. График желаемой и реальной позиции по X после настройки. Красным - желаемая позиция, зеленым - реальная.

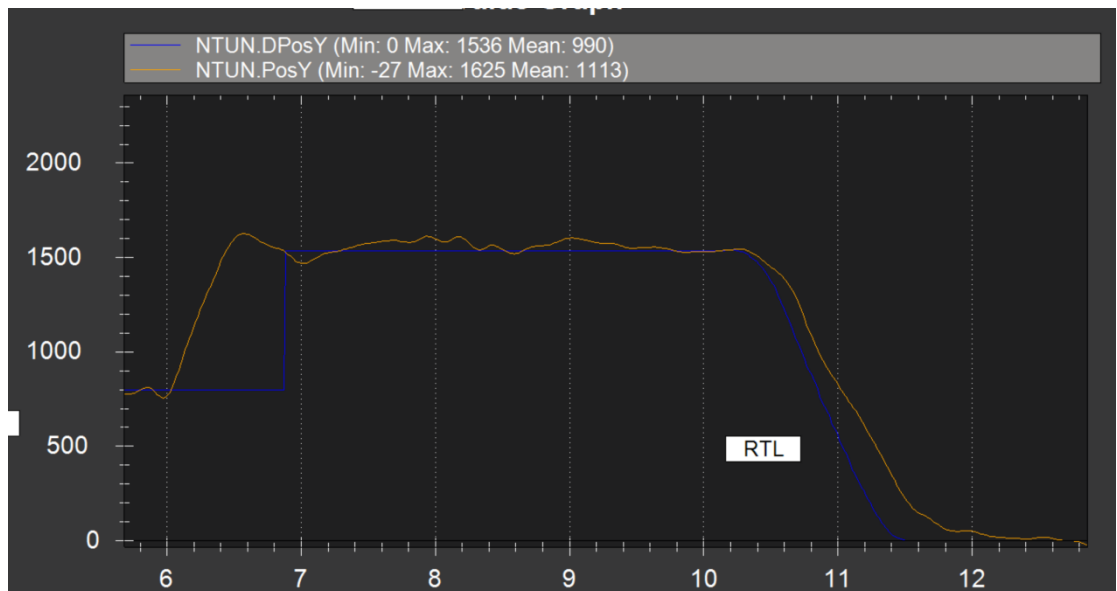


Рис. 4-14. График желаемой и реальной позиции по Y после настройки. Синим - желаемая позиция, Желтым - реальная.

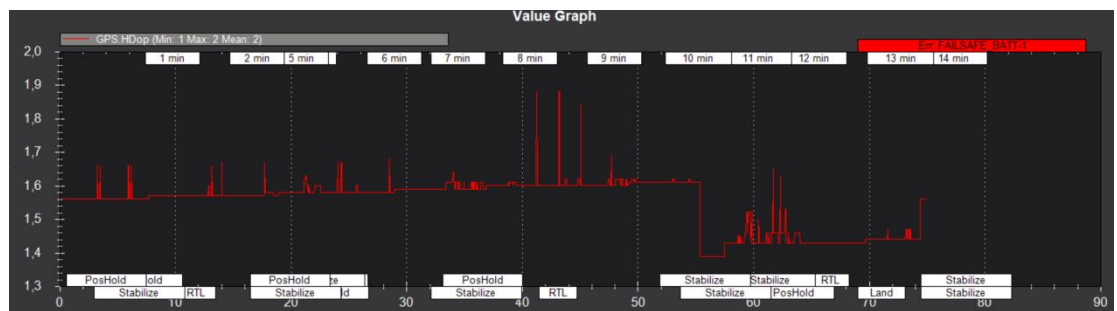


Рис. 4-15. Значения ошибки отслеживания положения (HDop) после настройки.

Как видно из графиков, HDop имеет значения, не превышающие 1.9, что является хорошим показателем. Также видно, что коптер держит позицию не улетая в стороны и не совершая круговые движения. Погрешность при висении составляет не более 1 метра, что также является хорошим показателем с учетом используемых компонентов.

Глава 5: Разработка ПО

5.1. Структура управления

Высокоуровневая логика запрограммирована на микрокомпьютере Raspberry Pi 3b. Тот в свою очередь управляет полетным контроллером APM2.8. Упрощенная схема реализованной системы приведена на Рис.5-1.

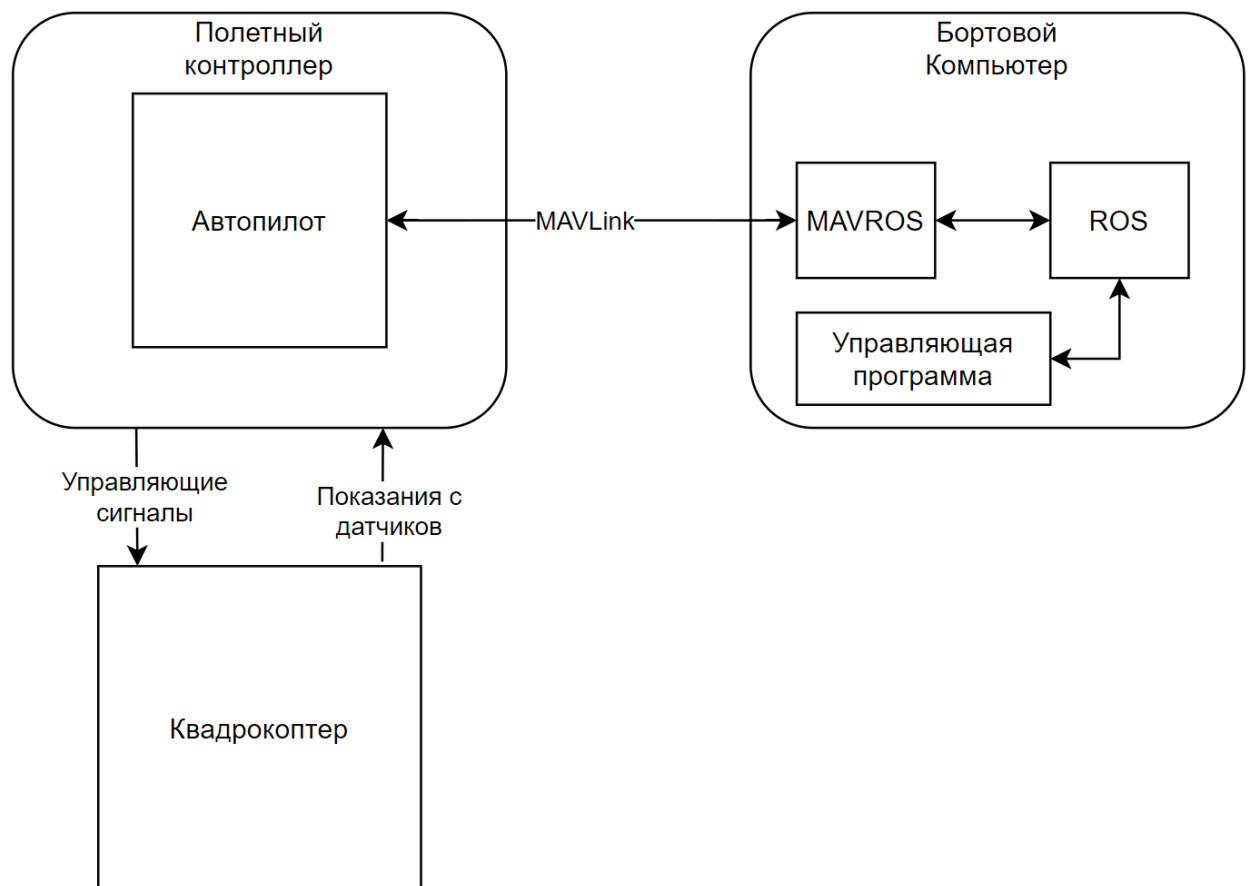


Рис. 5-1. Структура системы управления прототипа

Связь между полетным контроллером и бортовым компьютером осуществляется при помощи протокола MAVLink[23]. Micro Air Vehicle Link (MAVLink) - это протокол связи для беспилотных летательных систем, определяющий набор сообщений, которыми обмениваются беспилотные системы и наземные станции\бортовые компьютеры.

Для наиболее удобного программирования используется фреймворк ROS [24], который поддерживает Publisher/Subscriber архитектуру программного обеспечения, и пакет MAVROS[25], позволяющий управлять беспилотниками по протоколу MAVLink. Обмен сообщениями будет

организован через интерфейс Serial. Это самый надежный и удобный вариант среди всех доступных, необходимо будет просто подключить APM к Rpi проводом USB.

5.1.1. Автопилот ArduCopter

ArduCopter поддерживает 25 различных режимов полетов. Они определяют то, как дрон реагирует на сигналы управления и предоставляют различные уровни поддержки автопилота для пользователя, начиная от автоматизации общих задач, таких как посадка, заканчивая удержанием позиции, или полетом по точкам, полученным с наземной станции. Режим полета можно изменять с помощью пульта, наземной станции или бортового компьютера. Кроме того автопилот следит за показаниями батареи и в случае разряда будет выполнен алгоритм, предусмотренный FAILSAFE

В данной работе интерес представляет режим AUTO. В этом режиме управление, поступающее с пульта, игнорируется и квадрокоптер перемещается согласно программе или полетной миссии. Для правильной работы AUTO требует устойчивого полета в режимах Poshold и RTL, допустимых уровней ошибок компаса и вибраций. Без этого дрон может попасть в аварийную ситуацию и разбиться. Для корректной работы также необходима связь с наземной станцией или бортовым компьютером, так как автопилот может сохранить в памяти только 128 точек без необходимой информации о полете, полезной нагрузке и прочем.

Приложение Mission Planner позволяет создавать планы автономного полета, также именуемые полетными миссиями. При этом существуют продвинутые функции планирования, например, облет области по созданной сетке полета, зависание, следование за другим объектом по GPS трекеру и так далее. Смысл составления полетной миссии заключается в расстановке целевых точек, определяющих маршрут дрона, выставлении радиуса входа в целевую точку, определяющий точность прохождения точки. Также для

корректного полета необходимо выставить желаемую высоту и скорость полета. Интерфейс приведен на Рис. 5-2.

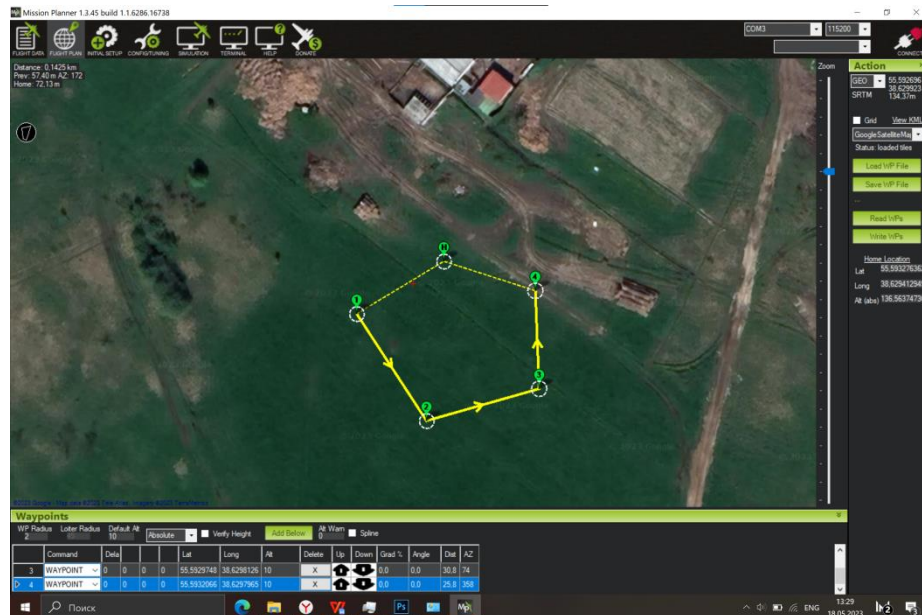


Рис. 5-2. Интерфейс для создания полетных миссий

Полетную миссию можно загрузить сразу на полетный контроллер, при этом размер ее ограничен. Альтернативным вариантом является сохранение полетной миссии на бортовой компьютер и последующее ее выполнение через управляющую программу. Такое решение позволяет не ограничивать размер миссии, а также расширяет полезный функционал дрона.

5.1.2. Фреймворк Robot Operating System

ROS – это широко используемый фреймворк для создания сложных, распределенных робототехнических систем. ROS представляет собой мета-операционную систему с открытым исходным кодом, структура которой представлена в виде узлов(ROS-node), которые обмениваются сообщениями между собой. Узлы - это программы, написанные на языках Python или C++, созданные пользователями или загруженные с официального сайта. Каждый узел отвечает за одну конкретную задачу, например, управление двигателями или построение карты местности. Узел, как и любая другая программа, поддерживает различные библиотеки. Общение узлов происходит путем передачи сообщений в различные шины данных, именуемых топиками(ROS-

topic). Сообщение представляет собой структуру данных, содержащую типизированные элементы. ROS содержит множество стандартных сообщений различных типов, но при необходимости можно создать свой тип сообщений. Каждый узел может получать и отправлять сообщения в различные топики. Эту концепцию можно увидеть на Рис.5-3.

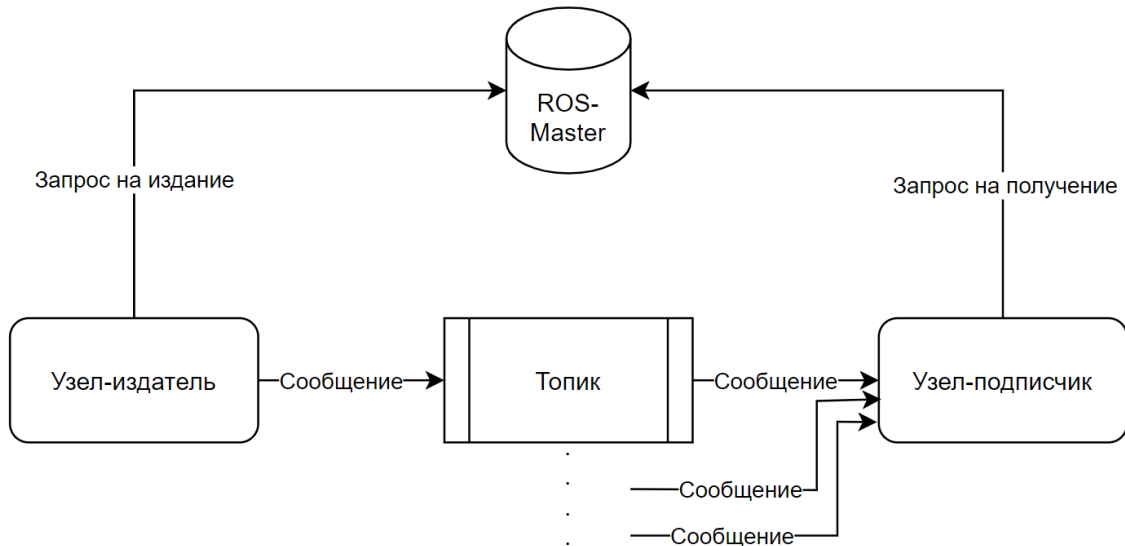


Рис. 5-3. Принцип работы программы, использующей ROS.

Мастер(ROS-Master) занимается поддержанием функционирования сети из узлов-издателей и узлов подписчиков и контролирует связь между ними. Основная задача Мастера - предоставить возможность обмена информацией между узлами. Кроме того, Мастер регулирует пространство имен, так как все узлы и топики должны иметь разные названия.

Совокупность узлов может быть объединена в специальный пакет(ROS-package). Пакет является основной рабочей единицей ROS, так как все программное обеспечение ROS организовано в пакетах. Цель пакетов - предоставить полезный функционал так, чтобы его было легко повторно использовать в различных проектах.

ROS в основном используется совместно с операционной системой Ubuntu. В данном проекте используется версия Ubuntu Server 20.04, предназначенная специально для Raspberry Pi и версия ROS-Noetic.

5.1.3. Протокол MAVLink

Как говорилось ранее, MAVLink или Micro Air Vehicle Link — это протокол информационного взаимодействия с дронами или малыми беспилотными аппаратами. Протокол определяет набор сообщений[26], включая сообщения для передачи телеметрии или полетных команд. Структура сообщения приведена на Рис.5-4.

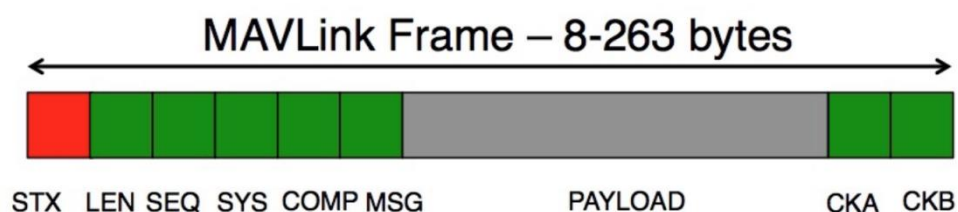


Рис. 5-4. Структура сообщения MAVLink.

Начальным индикатором сообщения является флаг пакета (STX), который всегда имеет значение 0xFE в шестнадцатичной системе счисления. Длина полезной нагрузки (LEN) указывает на размер тела пакета. Последовательность пакетов (SEQ) используется для обнаружения потери пакетов. Системный идентификатор (SYS) представляет собой уникальный номер системы, отправившего сообщение. COMP - идентификатор отправляющего компонента. Идентификатор сообщения (MSG) определяет структуру полезной нагрузки пакета и то, какие данные там будут лежать. Полезная нагрузка пакета (PAYLOAD) представляет собой тело сообщения, размером от 0 до 255 байт. Например, одно из основных сообщений, которым происходит постоянный обмен частотой 1 Гц это HEARTBEAT, имеет идентификатор сообщения «0». Основная задача этого сообщения - информировать о функционировании системы, это делают как наземная станция, так и БПЛА. В качестве полезной нагрузки это сообщение несет информацию о типе компонента, типе автопилота(наземные станции также учтены), а также о режиме полета. Функция этого сообщения CKA и CKB это нижний и верхний байты, которые, соответственно, содержат контрольную сумму пакета. MAVLink позволяет кодировать и декодировать пакеты согласно протоколу, но не регламентирует, какими аппаратными и

программными средствами данные будут отправлены. Физически обмен сообщениями может происходить чем угодно, что обеспечивает двусторонний обмен информацией. Полетный контроллер APM использует MAVLink для связи с наземной станцией или бортовым компьютером.

5.1.4. Пакет MAVROS

MAVROS (MAVLink + ROS) — это пакет для ROS, предоставляющий возможность управлять беспилотниками по протоколу MAVLink. MAVROS поддерживает автопилоты PX4 и APM. Этот пакет организует обмен сообщениями между дроном и бортовым компьютером\наземной станцией посредством топиков. MAVROS принимает и отправляет на дрон сообщения закодированные по протоколу MAVLink, но переводит их в другую форму и после публикует в соответствующие топики.

Некоторые топики предназначены только для чтения и несут в себе некоторую информацию состояния квадрокоптера. Другие же предназначены для публикации и при этом они меняют состояние квадрокоптера. Например, топик */mavros/battery* несет в себе информацию о напряжении аккумулятора и других параметрах батареи, а */mavros/setpoint_position/local* предназначен только для публикации и устанавливает целевую точку, в которую полетит квадрокоптер, при обращении к нему. Помимо топиков MAVROS предоставляет сообщения специального типа, используемые узлами этого пакета для корректной передачи данных автопилоту.

Таким образом, если необходимо выполнение какого-либо действия дроном, или получение информации с его датчиков, нужно обратиться к интересующему топiku. Грубо говоря, MAVROS - это удобный и понятный человеку «мост» между MAVLink и ROS. Кроме того, система, построенная таким образом, легко расширяема и встраиваема в любые другие беспилотные системы, так как MAVLink един для беспилотных летательных аппаратов различной конфигурации.

5.2. Алгоритм управления автономным полетом квадрокоптера

Для наиболее удобного использования и возможности дальнейшего расширения программа была адаптирована под использование ROS совместно с пакетом MAVROS. Для выполнения автономных полетных миссий был разработан алгоритм, который является легко улучшаемым и универсальным, так как может отработать фактически любую желаемую траекторию. Этот алгоритм прост в реализации, а также имеет возможность дальнейшего расширения функционала и может использоваться совместно с БПЛА различной конфигурации. Блок схема алгоритма программы приведена на Рис.5-5. Для реализации алгоритма был использован язык C++.

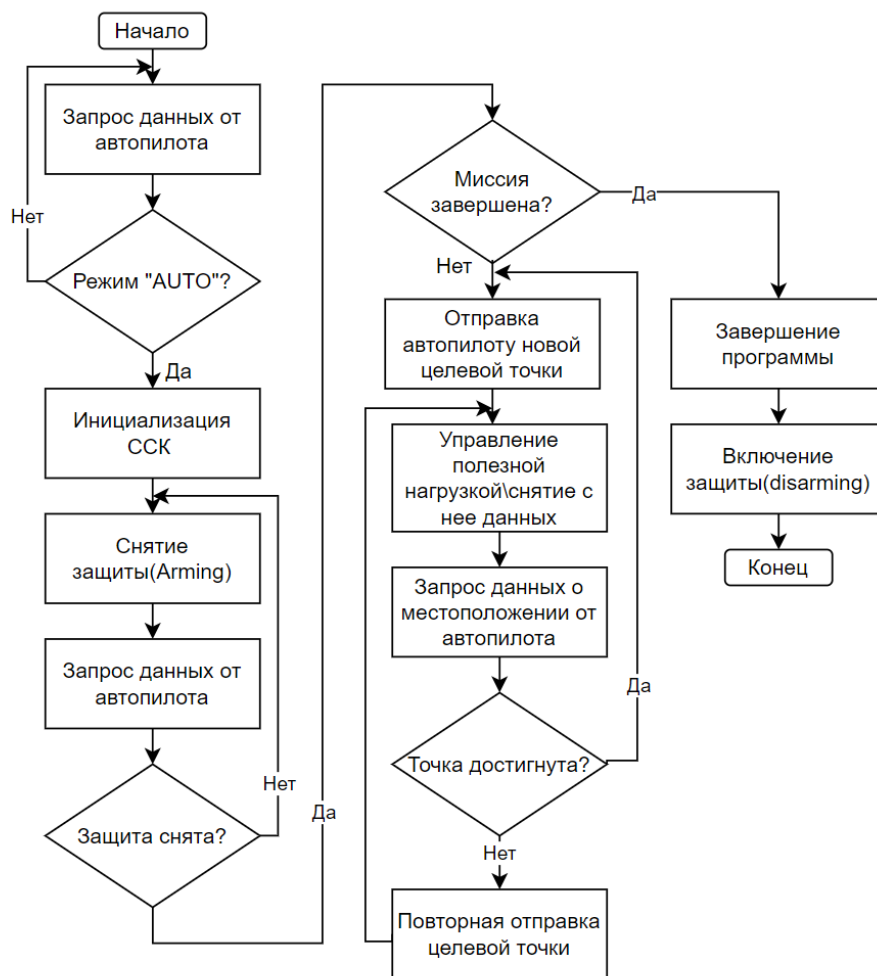


Рис. 5-5. Блок-схема алгоритма программы.

Для обеспечения универсальности, расширяемости и простоты программирования была реализована библиотека на языке C++, содержащая

наиболее часто используемые функции. Например, для того, чтобы инициализировать начальную точку полета, достаточно вызвать одну функцию, вся работа с топиками производится, как говорится «за кадром». Кроме того, функционал написанной библиотеки позволяет загружать файлы миссий, созданные в приложении Mission Planner, что значительно облегчает работу, при этом число целевых точек не ограничено. Код библиотеки приведен в источнике[31].

Блок «Запрос данных от автопилота» подразумевает получение данных о текущем состоянии защиты, полетном режиме дрона и о состоянии подключения к автопилоту. Эти данные содержатся в топике */mavros/state*.

Режим AUTO имеет особенность. Для поддержания его работы необходимо публиковать данные о целевой точке с частотой 2 Гц. Данную задачу решает функция инициализации ССК.

Для начала выполнения миссии дрон снимается защиты или другими словами производится Arming. Arming - это механизм для защиты человека и квадрокоптера от повреждений. Когда защиты включена, полетный контроллер игнорирует все поступающее на него управление. В MAVROS снятие с защиты реализовано публикацией в топик */mavros/cmd/arming* сообщения типа Bool с значением True.

Далее идет блок проверки завершения миссии. Завершение миссии наступает, когда все желаемые точки были достигнуты. При наличии целевых точек или команд управления нагрузкой миссия продолжается. Также миссия прерывается, если оператор перевел дрон в режим ручного управления.

После отработки всех точек происходит завершение работы алгоритма, посадка в точку взлета и включение защиты.

5.3. Тестирование работы алгоритма в симуляторе Gazebo

Тестирование алгоритма необходимо для выявления проблем и несостыковок на раннем этапе разработки до того, как программа будет

загружена на бортовой компьютер. Это необходимо для предотвращения поломок и сохранения имущества целостности.

В качестве симулятора для испытаний работоспособности был выбран Gazebo[27], так как это один из наиболее часто используемых и популярных симуляторов для тестирования программного обеспечения роботов. Помимо этого, в Gazebo существует огромная библиотека роботов, что экономит время на разработку, поскольку не придется с нуля писать код управляющих скриптов и плагинов, которые необходимы для симулирования каждого отдельно-взятого управляемого сочленения робота. Компания Ardupilot разработала собственную 3D-модель квадрокоптера IRIS+ и управляющие скрипты к нему, реализующие достоверную физику полета и, в частности, работы винтов.

Для тестирования была использована модель квадрокоптера IRIS+, так как это значительно упростит и ускорит процесс отладки. К тому же, команды, подаваемые на полетный контроллер, используют протокол MAVLink, поэтому независимо от аппаратной конфигурации высокоуровневые команды будут отработаны одинаково. Это значит, что для тестирования можно использовать вообще любую модель квадрокоптера, способного принимать сообщения MAVLink.

После установки[28] и запуска пакета для симуляции, открывается сцена мира Gazebo с квадрокоптером, изображенная на Рис. 5-6. Для лучшей видимости модель поверхности, на которой находится БПЛА, была сделана полупрозрачной.

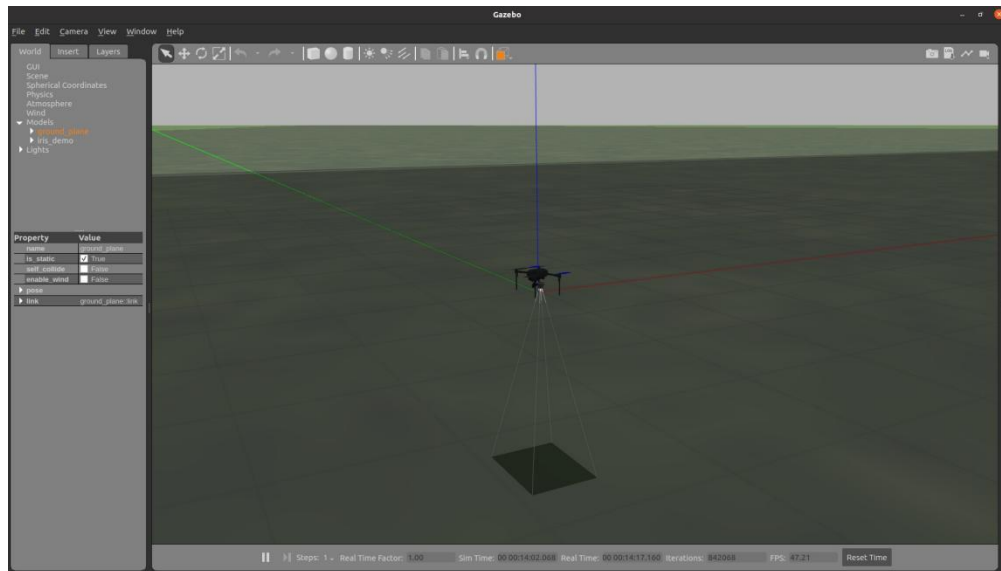


Рис. 5-6. Сцена мира с квадрокоптером в симуляторе Gazebo.

Помимо окна со сценой запускается SITL(Software In The Loop)[29]. Это программа-симулятор полетного контроллера, автопилот, запускаемый на стационарном компьютере и позволяющий тестировать поведение квадрокоптера без полетного контроллера и любых электромеханических и электронных компонентов. Данные для обработки берутся из динамики БПЛА основного симулятора, в данном случае из сцены Gazebo. Обмен информацией между двумя симуляциями происходит при помощи протокола обмена информацией транспортного уровня UDP. Также SITL позволяет принимать сообщения типа MAVLink из командной строки Linux.

Для использования MAVROS необходимо подключиться к симуляции полетного контроллера по UDP[30], с указанием адреса SITL. Предварительно необходимо запустить ядро ROS, также называемое ROS-Master. После проделанных действий, появляется множество различных топиков, как это обговаривалось в пункте 5.1.4. Теперь можно написать программу для тестирования автономного полета, с использованием разработанной ранее библиотеки[31].

В качестве теста было принято выполнить полет в симуляторе Gazebo по траектории квадрат, размером 5х5 метров на высоте 3 метра с автоматическим взлетом и посадкой.

Результаты показали[32], что полет был выполнен без ошибок и с заданной точностью. Это говорит о корректности алгоритма для автономного полета. Код теста приведен в Приложении А.

5.4. Перенос ПО на бортовой компьютер и тестирование прототипа

Для начала работы с Raspberry Pi, необходимо смонтировать образ дистрибутива Ubuntu Server версии 20.04 на SD карту. Монтаж образа можно производить различными способами, но самый простой и надежный - использование официального приложения Raspberry Pi Imager. Оно позволяет загружать различные дистрибутивы, разработанные для микрокомпьютеров архитектуры ARM, в том числе Ubuntu. Внешний вид приложения приведен на Рис.5-7.



Рис. 5-7. Приложение Raspberry Pi Imager для монтажа образа дистрибутивов.

После загрузки образа, необходимо загрузить ROS[33] и MAVROS[34]. Графическая оболочка у дистрибутива Ubuntu Server отсутствует, это означает, что все действия необходимо выполнять через терминал - программу, которая используется для взаимодействия с командной оболочкой. Терминал также называют командной строкой. Raspberry обладает ограниченными вычислительными возможностями, поэтому установка ROS была произведена без графических пакетов, таких как, rqt, MoveIt, rviz и тд.

Также был загружен пакет MAVROS. После установки необходимых фреймворков и пакетов необходимо перенести код, отлаженный и протестированный в симуляторе на персональном компьютере. Выполнить это можно загрузив исходный код в систему Github, предназначенную для хранения и контроля версий кода, а после загрузить выгруженный репозиторий непосредственно на Raspberry.

Выполнив все эти действия, можно приступать к подключению Rpi и полетного контроллера. Физически соединение реализовано через интерфейс Serial. Для этого используется обыкновенный кабель USB. После этого необходимо запустить launch-файл, который содержит информацию о порте FCU(Flight Control Unit), в нашем случае - о порте полетного контроллера, и другие, не интересующие сейчас данные. Запустив в терминале команду `roslaunch mavros apm.launch fcu_url:=udp://:14550@`, подключаемся к полетному контроллеру. Другими словами, сообщения в топиках MAVROS будут публиковаться напрямую с полетного контроллера.

Проверить корректность подключения можно путем обращения к топик `/mavros/state`. Сообщения этого топика содержат информацию о подключении, режиме полета, состоянии защиты и других параметрах автопилота. При правильном подключении в командную строку начнут выводиться данные, поступающие в топик. Обратившись к указанному топик с помощью команды `rostopic echo /mavros/state`, введенной в командную строку, убеждаемся в корректности подключения.

Для тестирования программы для автономного полета в реальной жизни была составлена полетная миссия из пяти точек, приведенная на Рис.5-8. Эта миссия была загружена на микрокомпьютер. Одна из функций написанной библиотеки позволяет совершать полет, передав файл полетной миссии в качестве входного параметра. Фактически, управляющая программа будет состоять из вызова одной этой функции.

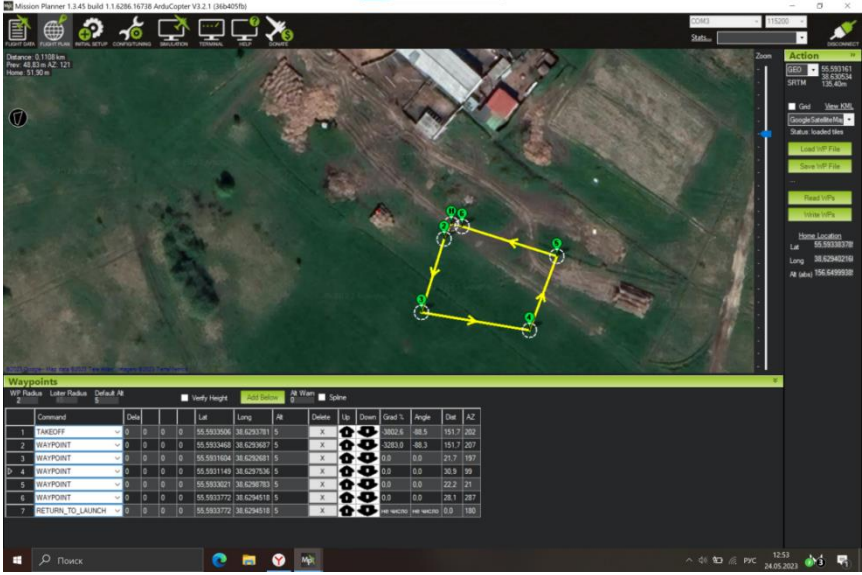


Рис. 5-8. Полетная миссия из пяти точек для тестирования корректности работы в реальной жизни.

При выполнении тестового полета[35], осуществленного 24 мая, не было выявлено проблем с полетом. При анализе логов было обнаружено, что точность траектории имеет видимые различия от желаемой. Траектория полета приведена на Рис. 5-9. Отклонение от желаемой траектории может быть связано с тем, что ошибки определения местоположения достигают 1.8-х метров. Этого может быть достаточно для того, чтобы иметь такие различия от желаемой траектории. Исправить это можно использованием более совершенного модуля GPS.

Рис. 5-9. Реальная и идеальная траектории полета. Синим - реальная, желтым - идеальная.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы был создан работоспособный прототип БПЛА с возможностью автономного полета. Были выполнены все поставленные задачи.

В рамках работы была разработана аппаратная конфигурация квадрокоптера и подобраны рама, двигатели, полетный контроллер, бортовой компьютер и другие комплектующие, обеспечивающие выполнение поставленной задачи и работоспособность дрона. Спроектирована 3D-модель квадрокоптера, разработаны индивидуальные детали для закрепления батареи и бортового компьютера. Собран прототип дрона. Произведена первоначальная и продвинутая настройки полетного контроллера, что позволило добиться необходимой точности полета и корректности выполнения управляющих команд. Разработано управляющее программное обеспечение: на языке C++ написана библиотека для взаимодействия с пакетом MAVROS и непосредственно управляющая программа. Автономный полет был протестирован в симуляторе Gazebo. Выполнена настройка Raspberry Pi 3 и загрузка необходимого программного обеспечения и управляющих программ. Произведены тестовые автономные полеты, проверена корректность функционала разработанной библиотеки и управляющей программы в реальности.

В работе описан процесс разработки прототипа автономного квадрокоптера, возможность по улучшению и модернизации которого огромна.

Библиографический список

- [1] Официальный сайт ГЕОСКАН <https://www.geoscan.aero/ru/products/gemini>; дата обращения к ресурсу - ноябрь 2022 г.
- [2] Официальный сайт DJI <https://www.dji.com/ru/matrice-300>; дата обращения к ресурсу - ноябрь 2022 г.
- [3] Высокоточное определение относительных координат по сигналам ГНСС методом RTK - ResearchGate 2021 г.
- [4] Новостной сайт Хабр <https://habr.com/ru/articles/503290/>; дата обращения к ресурсу - ноябрь 2022 г.
- [5] Официальный сайт COEX <https://ru.coex.tech/pelican>; дата обращения к ресурсу - март 2022 г.
- [6] Использование дрона для дезинфекции улиц <https://www.youtube.com/shorts/opH1vnS-zNY>; дата обращения к ресурсу - март 2022 г.
- [7] Прогноз рынка промышленных и военных дронов <https://www.pvsm.ru/issledovanie/57654>; дата обращения к ресурсу - ноябрь 2022 г.
- [8] BYU ScholarsArchive Citation Beard, Randal, "Quadrotor Dynamics and Control Rev 0.1" (2008). Faculty Publications. 1325. <https://scholarsarchive.byu.edu/facpub/1325> ; дата обращения к ресурсу - март 2022 г.
- [9] Комплекующие для квадрокоптера: <https://drongeeek.ru/profi/kvadrokopter-f450>; дата обращения к ресурсу - январь 2022 г.
- [10] Подбор бесколлекторных синхронных двигателей: <https://kopterinfo.ru/rcdetails-blog-kak-vybrat-motory-dlya-kvadrokoptera-ili-gonochnogo-drona/>; дата обращения к ресурсу - январь 2022 г.
- [11] Подбор и параметры LiPo аккумуляторов: <https://profpv.ru/kakoy-nuzhen-akkumulyator-dlya-kvadroko/>; дата обращения к ресурсу - январь 2022 г.

[12] Т. Килби. Make: дроны с нуля: Пер. с англ./ Т. Килби, Б. Килби. - СПб.: БХВ-Петербург, 2016. -192 с.: ил. с.36; дата обращения к ресурсу - апрель 2022 г.

[13] Филамент PLA: обзор, пошаговые настройки, возникшие проблемы и их решение <https://rec3d.ru/rec-wiki/pla-plastik-kharakteristiki-nastroyki-pechati-sovety/>; дата обращения к ресурсу - апрель 2022 г.

[14] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/configuring-hardware.html>; дата обращения к ресурсу - март 2022 г.

[15] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/common-accelerometer-calibration.html>; дата обращения к ресурсу - март 2022 г.

[16] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/esc-calibration.html>; дата обращения к ресурсу - март 2022 г.

[17] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/failsafe-landing-page.html>; дата обращения к ресурсу - март 2022 г.

[18] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/common-measuring-vibration.html?highlight=square+root>; дата обращения к ресурсу - апрель 2022 г.

[19] Официальная документация прошивки ArduPilot: <https://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html>; дата обращения к ресурсу - апрель 2022 г.

[20] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/common-diagnosing-problems-using-logs.html?highlight=hdop>; дата обращения к ресурсу - апрель 2022 г.

[21] Сайт для определения магнитуды местности: <https://www.magnetic-declination.com/#>; дата обращения к ресурсу - май 2022 г.

- [22] Официальная документация прошивки ArduPilot: <https://ardupilot.org/copter/docs/common-compass-setup-advanced.html>; дата обращения к ресурсу - май 2022 г.
- [23] A. Koubaa, A. Allouch, M. Alajlan, M. Khalgu. Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey https://www.researchgate.net/publication/333937231_Micro_Air_Vehicle_Link_MAVlink_in_a_Nutshell_A_Survey; дата обращения к ресурсу - май 2022 г.
- [24] D. Herath. The Robot Operating System (ROS1 &2): Programming Paradigms and Deployment https://www.researchgate.net/publication/363842243_The_Robot_Operating_System_ROS1_2_Programming_Paradigms_and_Deployment с.105; дата обращения к ресурсу - май 2022 г.
- [25] Официальный сайт ROS: <https://wiki.ros.org/mavros>; дата обращения к ресурсу - май 2022 г.
- [26] Официальный сайт MAVLink: <https://mavlink.io/en/messages/common.html>; дата обращения к ресурсу - май 2022 г.
- [27] Официальный сайт Gazebo: <https://gazebo.org/>; дата обращения к ресурсу - май 2022 г.
- [28] Официальная документация прошивки ArduPilot: <https://ardupilot.org/dev/docs/sitl-with-gazebo.html#sitl-with-gazebo>; дата обращения к ресурсу - май 2022 г.
- [29] Официальная документация прошивки ArduPilot: <https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>; дата обращения к ресурсу - май 2022 г.
- [30] Официальная документация прошивки ArduPilot: <https://ardupilot.org/dev/docs/ros-connecting.html>; дата обращения к ресурсу - май 2022 г.

[31] Котов Д.Д. Библиотека для управления АРМ с помощью MAVROS: https://github.com/NOTMOVETON/apis_api; дата обращения к ресурсу - май 2022 г.

[32] Котов Д.Д. Видео с тестовым полетом в симуляторе Gazebo: <https://youtu.be/OqBeNJev5wg>; - май 2022 г.

[33] Официальный сайт ROS: <https://wiki.ros.org/Installation>; дата обращения к ресурсу - май 2022 г.

[34] Github MAVROS:
<https://github.com/mavlink/mavros/tree/master/mavros#installation>; дата
 обращения к ресурсу - май 2022 г.

[35] Видео автономного полета по точкам: <https://youtu.be/7tuFJWDCvXY>

Приложение А

```
#include <apis_api.hpp>

int main(int argc, char** argv)
{
    ros::init(argc, argv, "gnc_node");
    ros::NodeHandle apis_node("~");
    init_publisher_subscriber(apis_node);
    wait4connect();
    wait4start();
    initialize_local_frame();
    takeoff(3);

    std::vector<apis_api_waypoint> waypointList;
    gnc_api_waypoint nextWayPoint;
    nextWayPoint.x = 0;
    nextWayPoint.y = 0;
    nextWayPoint.z = 3;
    nextWayPoint.psi = 0;
    waypointList.push_back(nextWayPoint);
    nextWayPoint.x = 5;
    nextWayPoint.y = 0;
    nextWayPoint.z = 3;
    nextWayPoint.psi = -90;
    waypointList.push_back(nextWayPoint);
    nextWayPoint.x = 5;
    nextWayPoint.y = 5;
    nextWayPoint.z = 3;
    nextWayPoint.psi = 0;
    waypointList.push_back(nextWayPoint);
    nextWayPoint.x = 0;
    nextWayPoint.y = 5;
    nextWayPoint.z = 3;
    nextWayPoint.psi = 90;
    waypointList.push_back(nextWayPoint);
    nextWayPoint.x = 0;
    nextWayPoint.y = 0;
    nextWayPoint.z = 3;
    nextWayPoint.psi = 180;
    waypointList.push_back(nextWayPoint);
    nextWayPoint.x = 0;
```

```

nextWayPoint.y = 0;
nextWayPoint.z = 3;
nextWayPoint.psi = 0;
waypointList.push_back(nextWayPoint);

ros::Rate rate(2.0);
int counter = 0;
while(ros::ok())
{
    ros::spinOnce();
    rate.sleep();
    if(check_waypoint_reached(.3) == 1)
    {
        if (counter < waypointList.size())
        {
            set_destination(waypointList[counter].x, waypointList[counter].y, waypointList[counter].z, waypointList[counter].psi);
            counter++;
        }else{
            //land after all waypoints are reached
            land();
        }
    }
}
return 0;
}

```