



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 301 Группа М30-402Бки-20

Направление подготовки 27.03.04 «Управление в технических системах»

Профиль 27.03.04.Б1 «Управление и информатика в технических системах»

Квалификация Бакалавр

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

На тему: Разработка алгоритмов и программного обеспечения для формирования
циклограмм управления испытаниями жидкостного ракетного
двигателя

Автор ВКРБ Саид Ахмад Заки Алхабши Бин Саид Худ (_____)

(фамилия, имя, отчество полностью)

Руководитель Егоров Александр Александрович (_____)

(фамилия, имя, отчество полностью)

Консультант _____ (_____)

(фамилия, имя, отчество полностью)

Консультант _____ (_____)

(фамилия, имя, отчество полностью)

Рецензент _____ (_____)

(фамилия, имя, отчество полностью)

К защите допустить

Заведующий кафедрой 301 Желтов Сергей Юревич (_____)

(№ каф)

(фамилия, имя, отчество полностью)

“ ____ ”

20 ____ г.

Москва 2024г.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт № 3 «Системы управления, информатика и электроэнергетика»
Кафедра 301 Группа М30-402Бки-20
Направление подготовки 27.03.04 «Управление в технических системах»
Профиль 27.03.04.Б1 «Управление и информатика в технических системах»
Квалификация Бакалавр

УТВЕРЖДАЮ

Заведующий кафедрой №301 С.Ю.Жельтов
(№. Каф.) (подпись) (инициалы, фамилия)
« » 20 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра

Обучающийся Саид Ахмад Заки Алхабши Бин Саид Худ
(фамилия, имя, отчество полностью)
Руководитель Егоров Александр Александрович
(фамилия, имя, отчество полностью)
к.т.н., доцент каф. 301 МАИ
(ученая степень, ученое звание, должность и место работы)

1. Наименование темы Разработка алгоритмов и программного обеспечения для формирования циклограмм управления испытаниями жидкостного ракетного двигателя

2. Срок сдачи обучающимся законченной работы 20.06.2024

3. Задание и исходные данные к работе

Разработать систему управления аварийной посадкой летательного аппарата. Система состоит из специализированной (интеллектуальной) платформы, отслеживающей движение летательного аппарата и обеспечивающей его безопасную посадку с учетом его положения, скорости и возможности поддержки при посадке. Предусматривается использование аппаратно-программных средств, например, на платформе Arduino. Для получения данных с летательного аппарата используются датчики и линии связи. В ходе натурного моделирования системы управления будут моделированы аппаратная и программная части. В качестве аппаратной части моделируются летательный аппарат (самолет) и интеллектуальная платформа. Программная часть включает разработку соответствующего программного обеспечения с использованием протоколов обмена данными (например, I2C, SPI, UART).

Перечень иллюстративно-графических материалов: *при наличии

№ п/п	Наименование	Количество листов
1	2	3
1.	Структурная схема системы аварийной посадки летательного аппарата	1

2.	Функциональная схема системы аварийной посадки	1
3.	Алгоритмы функционирования системы управления аварийной посадки	2
4.	Моделирование системы управления аварийной посадки	1
5.	Экспериментальная установка системы управления аварийной посадки	1
6.	Схема электрических соединений	1
7.	Спецификация к схеме электрических соединений	1
8.	Результаты экспериментов	1
9.	Программный код	2

4. Перечень подлежащих разработке разделов и этапы выполнения работы

№ п/п	Наименование раздела или этапа	Трудоёмкость в % от полной трудоёмкости дипломной работы	Срок выполнения	Примечание
1	2	3	4	5
1.	Обзор систем аварийной посадки летательных аппаратов.	10	20.02.2024	
2.	Анализ актуальности системы управления аварийной посадки. Постановка задачи	10	29.02.2024	
3.	Выбор аппаратных и программных средств реализации системы аварийной посадки	15	04.03.2024	
4.	Разработка структурной схемы и алгоритма функционирования системы аварийной посадки	25	24.03.2024	
5.	Программно-аппаратная реализация системы аварийной посадки	20	17.04.2024	
6.	Анализ результатов	20	20.06.2024	

5. Исходные материалы и пособия

- 1 Симонов В.Л., Ерпелев А.В., Давыдова Е.К., Хохлов Е.Г. Моделирование системы управления вертикальным взлетом и посадкой / В сборнике: Современные информационные технологии в образовании, науке и промышленности. Сборник трудов XV Международной конференции, XIII Международного конкурса научных и научно-методических работ. Москва, 2020. - с. 159-161.
- 2 Смулов М.Ю. Обеспечение безопасности полётов воздушных судов на этапах взлёта и посадки в условиях неопределённости информации о внешних возмущениях. Автореферат на соискание ученой степени доктора технических наук (специальность 05.22.14 "Эксплуатация воздушного транспорта"). - МГТУ ГА. - 2004 г. - 40 с.

6. Дата выдачи задания 09.02.2024

Руководитель _____
(подпись)

Задание принял к исполнению _____
(подпись)

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	6
ВВЕДЕНИЕ	7
1 АНАЛИЗ АКТУАЛЬНОСТИ СИСТЕМЫ УПРАВЛЕНИЯ АВАРИЙНОЙ ПОСАДКИ ЛЕТАТЕЛЬНОГО АППАРАТА	10
1.1 Основная потребность самолета в безопасной посадке	10
1.2 Основные части самолета, задействованные при посадке	11
1.3 Процесс захода на посадку и приземления самолета	12
1.4 Распространенные типы аварий при заходе на посадку летательного аппарата	14
1.5 Случаи из реальной жизни, которые показывают опасность аварийной посадки летательного аппарата	17
1.6 Расчет системы, позволяющий успешно и безопасно посадить самолет	18
1.7 Комплексная система интеллектуальной посадочной платформы	23
1.8 Выводы к первому разделу и постановка задачи	29
2 ВЫБОР АППАРАТНЫХ И ПРОГРАММНЫХ СРЕДСТВ РЕАЛИЗАЦИИ СИСТЕМЫ АВАРИЙНОЙ ПОСАДКИ	32
2.1 Выбор средств аппаратной реализации	32
2.2 Выбор средств программной реализации	53
2.3 Разработка алгоритма функционирования системы управления аварийной посадки летательного аппарата	59
2.4 Моделирование системы управления аварийной посадки летательного аппарата	65
2.5 Выводы ко второму разделу	67
3 РЕАЛИЗАЦИИ И АНАЛИЗ РЕЗУЛЬТАТОВ СИСТЕМЫ УПРАВЛЕНИЯ АВАРИЙНОЙ ПОСАДКИ ЛЕТАТЕЛЬНОГО АППАРАТА	69
3.1 Структурная схема системы управления аварийной посадки летательного аппарата	69
3.2 Экспериментальная установка и проведение эксперимента	69
3.3 Испытание механизма фиксации	71
3.4 Испытание системы слежения за положением самолета	72
3.5 Испытание системы слежения за наклоном самолета	74
3.6 Схема электрических соединений	76
ЗАКЛЮЧЕНИЕ	79
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	81

ПРИЛОЖЕНИЕ	84
Приложение А (Обязательное) Листинг программ для передатчика	
Приложение Б (Обязательное) Листинг программ для приемника	
Приложение В (Обязательное) Иллюстративно-графические материалы	

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем отчете о ВКР применяют следующие термины с соответствующими определениями

ВКР	– выпускная квалификационная работа;
ВС	– воздушное судно;
ВСУ	– вспомогательная силовая установка;
ЛА	– летательный аппарат;
ОЗУ	– оперативное запоминающее устройство;
ПЗУ	– постоянное запоминающее устройство;
ПО	– программное обеспечение;
ТЗ	– техническое задание;
ТО	– техническое описание;
ШИМ	– широтно-импульсная модуляция;
ЭД	– эксплуатационная документация;
ЦП	– Центральный процессор
FAR	– федеральные авиационные правила (англ. federal aviation rules);
IDE	– интегрированной средой разработки (англ. integrated development environment
I/O	– ввода-вывода (англ. input/output);
SPI	– последовательного периферийного интерфейса (англ. serial peripheral interface);
VREF	– пороговая скорость (англ. velocity reference);

ВВЕДЕНИЕ

Можно с уверенностью сказать, что авиаперелеты – один из самых безопасных видов транспорта. Однако, когда авиационные происшествия все же происходят, результаты часто бывают катастрофическими. Не существует таких вещей, как незначительные столкновения самолетов или аварийная посадка, эти происшествия являются разрушительными, изменяющими жизнь событиями. В 2007 году Национальный совет по безопасности на транспорте оценил общий налет почти в 24 миллиона часов. Из этих 24 миллионов часов 6,84 из каждых 100 000 летных часов привели к авиакатастрофе, а 1,19 из каждых 100 000 – к катастрофе со смертельным исходом. Согласно последнему опубликованному отчету Международной ассоциации воздушного транспорта (IATA), в 2022 году произошло в общей сложности 39 авиационных происшествий.

Заход на посадку и приземление, как доказано, являются наиболее рискованным этапом полета. Согласно исследованию, проведенному Университетом Крэнфилда, более 50% авиационных происшествий происходят во время посадки и захода на посадку. К числу распространенных типов аварий при заходе на посадку относятся: CFIT (управляемый полет на местности), LOC (потеря управления) и отклонения от взлетно-посадочной полосы. Как инженеры, нельзя полагаться исключительно на пилотов, поскольку они управляют самолетом в течение длительного периода времени.

В этой работе, Система Управления Аварийной Посадки Летательного Аппарата, была разработана интеллектуальную посадочную платформу, которая используется в случае аварийной посадки чтобы свести к минимуму человеческую ошибку, допущенную пилотом. Интеллектуальная посадочная платформа также может компенсировать механические ошибки, такие как неисправность шасси летательного аппарата. В настоящее время этот подход широко используется и для беспилотных летательных аппаратов. Интеллектуальная платформа будет оснащена датчиками, которые вычисляют положение самолета и расстояние до него. Для отслеживания угла наклона

самолета платформа будет оснащена несколькими двигателями, которые получают информацию через модуль приемника. В случае приземления с неконтролируемой скоростью интеллектуальная платформа является подвижной, и после соприкосновения с самолетом она замедлит его движение по аэродрому.

Можно рассмотреть самолет, которому необходимо совершить посадку в очень плохую погоду, например, посадка в очень ветреный день или в снежную бурю. Пилоту будет трудно разглядеть платформу, а человек на аэродроме мало что сможет сделать, если его не будет видно. Именно поэтому будет предусмотрен индикатор, который сигнализирует пилоту о переходе на интеллектуальную платформу, чтобы можно оказать ему максимальную помощь. Наша работа заключается в создании системы управления, которая обеспечивает безопасность пассажиров и не влияет на состояние воздушного судна каждый раз, когда возникают трудности для его безопасной посадки.

Принцип работы системы аварийной посадки состоит из нескольких шагов. Сначала самолет связывается с диспетчерской, чтобы сообщить о своем состоянии. Это делается для того, чтобы убедиться, что пилот готов и знает о своем собственном состоянии и состоянии самолета. Затем диспетчерская служба должна убедиться, что посадочная площадка самолета хорошо подготовлена и безопасна для посадки. Если во время захода на посадку будут обнаружены какие-либо проблемы, диспетчерская служба направит самолет на посадку в указанное место, на котором расположена интеллектуальная платформа. Пилот будет знать местоположение, указанное нашим индикатором, и сможет безопасно приземлиться, несмотря на любые проблемы, возникающие с самолетом. Чтобы обеспечить безопасные условия посадки, необходимо изучить способ контроля и регулирования случайной ситуации, с которой мы не ожидаем столкнуться при посадке самолета.

Для моделирование систем управления аварийной посадки летательного аппарата в настоящей ВКР использовались современный подход, который прост в реализации и может быть применен к реальной системе. Также

использовались платформу Arduino, которая не является редкостью среди инженеров-мехатроников, которые создают и тестируют проекты в небольших масштабах, прежде чем внедрять их в реальном мире.

Разработка проводилась в соответствии с требованиями стандартов и нормативов:

- ГОСТ 2.114 – 95 «Единая система конструкторской документации. Технические условия»;
- ГОСТ 19.701-90 (ИСО 5807-85). Межгосударственный Стандарт. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Группа Т55э;
- и ряда других.

Структура и объем работы: ВКР состоит из введения, трех глав, заключения, списка использованных источников и приложений.

1 АНАЛИЗ АКТУАЛЬНОСТИ СИСТЕМЫ УПРАВЛЕНИЯ АВАРИЙНОЙ ПОСАДКИ ЛЕТАТЕЛЬНОГО АППАРАТА

1.1 Основная потребность самолета в безопасной посадке

Прежде всего, чтобы самолет мог приземлиться, он должен получить разрешение от управления воздушного движения на посадку в указанном аэропорту. Это предполагает согласование с управлением воздушного движения наличия взлетно-посадочной полосы и последовательности выполнения других прибывающих и вылетающих рейсов. Следующим шагом является то, что самолет должен быть выровнен по безопасной траектории захода на посадку к взлетно-посадочной полосе. Это предполагает выполнение навигации в соответствии с установленными правилами полета и снижение до необходимой высоты для посадки.

Кроме того, погода в аэропорту должна быть подходящей для посадки, включая видимость, ветровые условия и отсутствие неблагоприятных погодных явлений, таких как грозы или сильный туман, которые могли бы повлиять на безопасную посадку. Кроме того, важнейшие системы самолета, такие как шасси, тормоза, двигатели и органы управления полетом, должны быть полностью исправны и готовы к посадке.

Посадка самолета – непростая задача, поэтому важно, чтобы пилот был должным образом обучен, находился в курсе событий и был готов выполнить безопасную посадку. Это включает в себя управление скоростью, высотой и темпом снижения самолета, чтобы приземление было плавным и безопасным. И, наконец, хотя это и не является прямой частью обычных посадок, на борту самолета должны быть предусмотрены аварийные протоколы на случай возникновения непредвиденных проблем в процессе посадки. Каждый из этих факторов способствует обеспечению безопасной и успешной посадки самолета.

1.2 Основные части самолета, задействованные при посадке

На этапе приземления самолета несколько ключевых частей и систем работают сообща, обеспечивая безопасное и контролируемое приземление. Первой важной частью приземления, безусловно, является шасси самолета. Шасси включает в себя колеса, шины, стойки и амортизаторы, которые поддерживают самолет во время посадки и руления. Шасси раскрывается перед приземлением и обеспечивает устойчивость и амортизацию при посадке. Следующими частями являются закрылки и предкрылок. Это подвижные поверхности на крыльях, которые выдвигаются при посадке. Закрылки увеличивают площадь и кривизну крыла, что увеличивает подъемную силу на более низких скоростях, позволяя самолету снижаться более плавно. Предкрылки – это передовые устройства, которые улучшают обтекание крыла воздушным потоком на больших углах атаки, помогая поддерживать подъемную силу и управляемость.

Для снижения скорости после приземления самолет оснащен спойлерами и тормозами. Спойлеры – это панели на крыльях, которые можно поднимать после приземления, чтобы уменьшить подъемную силу и снизить скорость самолета. Они также помогают поддерживать контакт с землей и устойчивость во время взлета. Тормоза самолета используются для замедления и остановки самолета после приземления. Обычно они расположены на основной стойке шасси и могут управляться пилотом с помощью тормозной системы самолета.

Для поддержания правильной высоты и центровки при посадке самолет оснащен системой управления полетом. Система управления полетом самолета, включая элероны, руль высоты и направление поворота. Эти элементы управления имеют решающее значение для управления скоростью и направлением снижения самолета. Также предусмотрена система противоскольжения, которая предотвращает блокировку колес во время торможения, что может привести к заносу. Система противоскольжения регулирует тормозное давление для оптимизации эффективности торможения

и поддержания сцепления шин с дорогой. На рисунке 1 показан узел тормозной системы на основной стойке шасси 737NG.



Рисунок 1 – Тормозная система в сборе на основной стойке шасси 737NG

Противоскользящие системы предназначены для минимизации аквапланирования и потенциального повреждения шин, которые могут возникнуть, когда колесо заблокировано или вращается со скоростью, не соответствующей скорости самолета.

1.3 Процесс захода на посадку и приземления самолета

Заход на посадку и приземление самолета представляют собой критический этап полета, требующий точных маневров и тщательного планирования для обеспечения безопасного и контролируемого прибытия в аэропорт назначения. Этот процесс начинается со снижения с крейсерской высоты по мере приближения самолета к месту назначения. Пилоты начинают снижение за несколько минут до прибытия в аэропорт, постепенно снижая высоту под руководством службы управления воздушным движением (УВД). Служба УВД предоставляет инструкции относительно высоты, скорости и направления полета, которым следует следовать на этапе снижения, что способствует плавному переходу к аэропорту.

Когда самолет входит в начальную фазу захода на посадку, пилот начинает подготовку самолета к посадке. Это включает в себя выпуск шасси и регулировку закрылков и предкрылков для увеличения подъемной силы на более низких скоростях. Эти подготовительные мероприятия необходимы для обеспечения стабильной глиссады по направлению к взлетно-посадочной полосе. Тем временем пилот связывается с центром УВД, чтобы получить разрешение на заход на посадку, обеспечивая координацию и безопасное отделение от других воздушных судов, находящихся поблизости.

После выравнивания с расширенной центральной линией взлетно-посадочной полосы самолет переходит к завершающей фазе захода на посадку. Пилот аккуратно направляет самолет по стабилизированной глиссаде к порогу взлетно-посадочной полосы. На протяжении всего этого этапа пилот контролирует ключевые параметры, такие как воздушная скорость, высота и скорость снижения, внося необходимые коррективы для обеспечения плавного и контролируемого захода на посадку. Такое тщательное внимание к деталям имеет решающее значение для обеспечения безопасной посадки.

Перед дальнейшим снижением пилот выполняет полный контрольный список перед посадкой, чтобы убедиться, что все системы работают правильно для посадки. Это включает в себя проверку того, что шасси выпущены и зафиксированы, настройку закрылков и предкрылков для посадки и соответствующую настройку мощности двигателя. После завершения этих приготовлений самолет направляется к порогу взлетно-посадочной полосы. Когда самолет приближается к взлетно-посадочной полосе, пилот выполняет маневр разворота, слегка задирая нос самолета вверх, чтобы снизить скорость снижения. Этот метод обеспечивает плавное приземление, когда основные колеса соприкасаются с поверхностью взлетно-посадочной полосы. Пилот продолжает управлять скоростью и направлением полета самолета, используя педали управления рулем направления для поддержания выравнивания по центральной линии взлетно-посадочной полосы.

После приземления пилот задействует интерцепторы (если они установлены), чтобы снизить подъемную силу и повысить эффективность торможения. Для замедления самолета используются тормоза, а реверсоры тяги (если они имеются на реактивных самолетах) – для ускорения торможения. Пилот сохраняет контроль над самолетом на протяжении всего этапа выведения, обеспечивая безопасный и своевременный выезд с взлетно-посадочной полосы на обозначенную рулежную дорожку.

В заключение следует отметить, что заход на посадку и посадка самолета требуют тщательной координации, умелого выполнения и эффективной коммуникации между пилотом и органами управления воздушным движением. Каждый этап процесса – от планирования снижения и захода на посадку до приземления и разворачивания – играет жизненно важную роль в обеспечении безопасного и эффективного прибытия в аэропорт назначения. Пилоты проходят интенсивную подготовку, чтобы освоить эти процедуры и справиться с различными ситуациями, которые могут возникнуть на этапах захода на посадку и приземления, что подчеркивает приверженность авиационной отрасли безопасности и профессионализму.

1.4 Распространенные типы аварий при заходе на посадку летательного аппарата

На этапе посадки во время полета может возникнуть несколько критических проблем, которые могут привести к необходимости аварийной посадки. Среди потенциальных проблем, которые могут возникнуть во время посадки, требующих аварийной посадки, является неисправность шасси. Это одна из наиболее распространенных проблем при посадке. Это может включать в себя отказ шасси при раскрытии или фиксации в нужном положении, что может потребовать от пилота выполнения аварийной посадки без использования шасси, известной как посадка с поднятым шасси или посадка на брюхо, или перелета в аэропорт со специализированным оборудованием для аварийных посадок. Следующий тип проблем, с которыми

может столкнуться пилот при посадке, – это отказ двигателя или потеря мощности. Если один или несколько двигателей отказывают или теряют мощность во время посадки, пилот может быть вынужден выполнить аварийную посадку, чтобы обеспечить безопасное приземление с уменьшенной тягой двигателя.

Кроме того, отказ гидравлической системы также является одной из причин аварийной посадки. Проблемы с гидравлической системой самолета, которая контролирует такие важные функции, как выпуск шасси, закрылков и тормозов, могут привести к трудностям в управлении самолетом во время посадки. На рисунке 2 показан пример гидравлической системы самолета Сухой Суперджет–100».

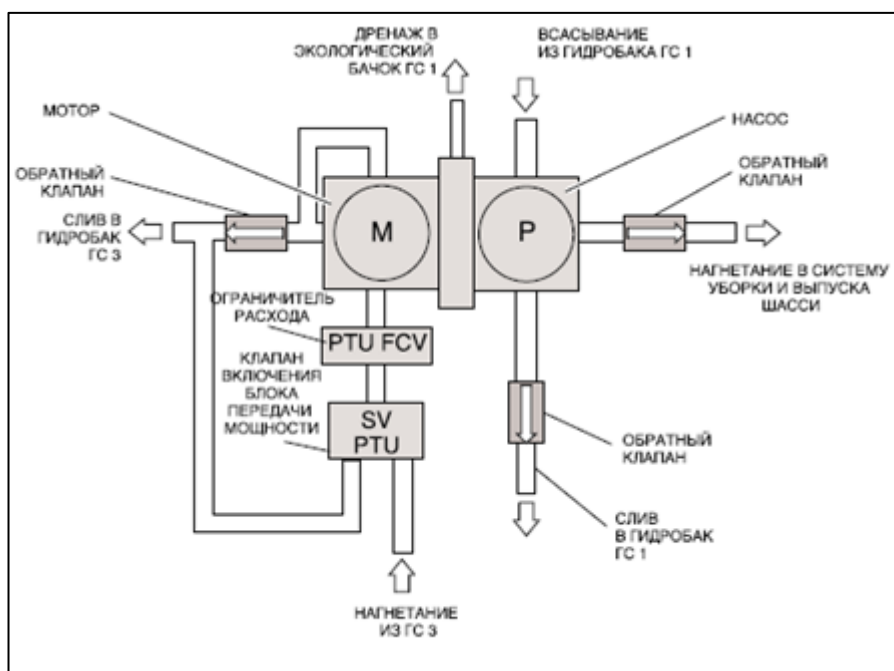


Рисунок 2 – Гидравлическая система самолета Сухой Суперджет–100»

Следующей проблемой является отказ тормозов, что затрудняет снижение скорости или остановку воздушного судна после приземления. Пилотам может потребоваться применение экстренных мер, таких как экстренное торможение или другие методы, для безопасной остановки воздушного судна.

Еще одна проблема, вызывающая трудности при посадке, связана с взлетно-посадочной полосой, которая представляет собой отклонение от

взлетно-посадочной полосы. Отклонение от взлетно-посадочной полосы – это инцидент, при котором воздушное судно во время посадки или взлета отклоняется от взлетно-посадочной полосы, что приводит к непреднамеренному отклонению от подготовленной поверхности. Это может произойти из-за различных факторов, таких как неблагоприятные погодные условия, ошибка пилота, механические неполадки или состояние взлетно-посадочной полосы. Отклонения от взлетно-посадочной полосы могут иметь значительные последствия для безопасности посадки воздушного судна, поскольку они могут привести к потере управления, превышению пределов взлетно-посадочной полосы или столкновению с препятствиями за пределами взлетно-посадочной полосы. Когда самолет во время посадки отклоняется от взлетно-посадочной полосы, это может привести к потере управляемости по курсу, потенциальному повреждению воздушного судна и повышенному риску получения травм пассажирами и экипажем. Кроме того, отклонения от взлетно-посадочной полосы могут нарушить работу аэропорта и требуют тщательного расследования для определения причин и предотвращения подобных происшествий в будущем, что подчеркивает важность соблюдения правил безопасности на взлетно-посадочной полосе и эффективной подготовки пилотов для управления сложными сценариями посадки. Воздействие этого эффекта мы можем наблюдать на рисунке 3.



Рисунок 3 – Отклонение от взлетно-посадочной полосы летательного аппарата

При любой аварийной посадке пилоты обучены следовать определенным протоколам, взаимодействовать с органами управления воздушным движением и уделять приоритетное внимание безопасности пассажиров и экипажа. Аварийная посадка требует тщательного принятия решений и может потребовать координации действий с наземными аварийно-спасательными службами для обеспечения безопасного исхода. Полагаться только на пилотов рискованно, и должно быть решение, которое можно бы принять, чтобы избежать несчастных случаев при посадке самолетов.

1.5 Случаи из реальной жизни, которые показывают опасность аварийной посадки летательного аппарата

За прошедшие годы история показала нам случаи аварийных посадок самолетов, которые приводили к большим жертвам. Вот почему эта ВКР актуально, поскольку, возможно, она может снизить процент ошибок и в один прекрасный день полностью спасти самолет от ужасающих аварий, подобных инциденту, о котором будет рассказано.

2 августа 2005 года Airbus A340–300, выполнявший рейс 358 авиакомпании Air France, вылетел за пределы взлетно-посадочной полосы в международном аэропорту Торонто Пирсон при попытке приземлиться во время грозы. Самолет, на борту которого находились 297 пассажиров и 12 членов экипажа, выкатился за пределы взлетно-посадочной полосы и упал в овраг. Чудом все пассажиры выжили в авиакатастрофе, хотя некоторые получили травмы. Причиной аварии стали несколько факторов, включая неблагоприятные погодные условия, ошибку пилота и недостаточную безопасность на взлетно-посадочной полосе.

Следующий случай показан 6 июля 2013 года Boeing 777–200ER, выполнявший рейс 214 авиакомпании Asiana Airlines, потерпел крушение при попытке приземлиться в международном аэропорту Сан–Франциско (SFO). Воздушное судно ударилось о дамбу и затем рухнуло на взлетно-посадочную полосу, что привело к пожару и значительному повреждению воздушного

судна. Три пассажира погибли, и многие другие получили ранения. Авиакатастрофа была вызвана, в первую очередь, неумелым управлением пилотов при заходе на посадку и недостаточным контролем скорости полета во время попытки посадки. Инцидент подчеркнул важность подготовки экипажа и бдительности на критических этапах полета.

Последний случай который доказывает актуальности этой работы это про самолета Boeing 737–800, выполнявший рейс 812 авиакомпании Air India Express, в 22 мая 2010, вылетел за пределы взлетно-посадочной полосы и разбился в аэропорту Мангалора (ныне аэропорт Мангалуру) в Индии. Самолет летел из Дубая, ОАЭ, со 166 пассажирами и членами экипажа на борту. При посадке самолет не смог остановиться на взлетно-посадочной полосе, упал с обрыва и загорелся. К сожалению, погибли 158 человек, что сделало это авиакатастрофу одной из самых смертоносных в Индии. Причиной крушения стала ошибка пилота, в том числе неадекватная техника захода на посадку.

Эти примеры подчеркивают сложность и риски, связанные с посадками воздушных судов, особенно в сложных погодных условиях или ситуациях, требующих точных навыков пилотирования. Отчеты о расследованиях и уроки, извлеченные из таких происшествий, способствовали продолжающимся усилиям по повышению безопасности полетов и снижению рисков при выполнении посадочных операций, таких как эту ВКР.

1.6 Расчет системы, позволяющий успешно и безопасно посадить самолет

1.6.1 Стандартный метод расчета расстояния, необходимого для приземления

Чтобы посадить самолет, должно учитывать несколько факторов, включая скорость, высоту, атмосферное давление, вес самолета и другие факторы, которые потенциально могут повлиять на характеристики самолета при посадке. Формула, используемая для расчета посадочной дистанции,

может быть сложной и обычно включает эмпирические данные, таблицы или вычислительные методы, полученные на основе данных о летно-технических характеристиках воздушного судна.

Основная концепция заключается в расчете общего расстояния, необходимого летательному аппарату для снижения скорости при заходе на посадку с определенной скоростью до полной остановки на взлетно-посадочной полосе. Следующее, что нам нужно сделать, это собрать всю информацию, которая нам понадобится, чтобы использовать графики, специально разработанные для пилотов, для оценки расстояния, необходимого для посадки самолета. Это включает в себя прогнозируемые приземные ветры, высоту над уровнем моря, направление взлетно-посадочной полосы и прогнозируемую температуру поверхности. На рисунке 4 показано расстояние, необходимое самолету от захода на посадку до полной остановки.

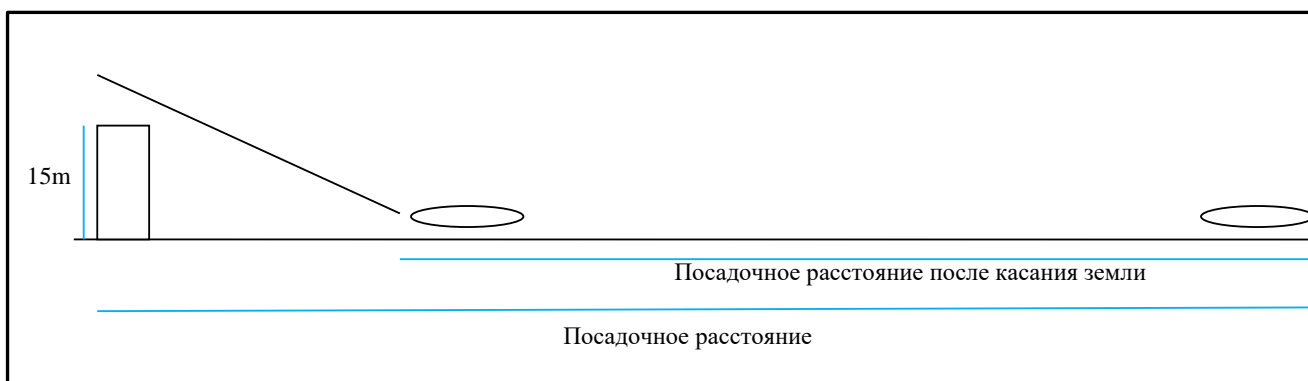


Рисунок 4 – Расстояние, необходимое самолету от захода на посадку до полной остановки

Прежде чем самолет сможет начать заход на посадку, ему необходимо рассмотреть ближайшее препятствие высотой 15 м, поскольку самолет должен пролететь над ним, прежде чем он сможет начать снижать высоту. Расстояние от препятствия до точки полной остановки самолета называется “посадочное расстояние”. Как только самолет касается земли, расстояние от точки, где самолет впервые касается земли, до точки, где самолет полностью останавливается, называется “посадочное расстояние после касания земли”.

Первым фактором, который следует учитывать, является скорость захода на посадку воздушного судна. Это скорость, с которой воздушное судно летит непосредственно перед приземлением. Обычно она указывается как пороговая скорость (VREF) или посадочная скорость. Во-вторых, вес самолета влияет на скорость сваливания, аэродинамические характеристики и возможности торможения. Это связано с первым законом Ньютона, который гласит, что под инерцией, или инертностью, понимают стремление тела сохранить неизменным свое состояние по отношению к инерциальной системе отсчета [1].

Затем высота взлетно-посадочной полосы и прогнозируемая высота полета используются для определения расстояния, необходимого для посадки воздушного судна. Большая высота и более низкое атмосферное давление могут повлиять на характеристики воздушного судна, включая мощность двигателя, подъемную силу и аэродинамическое сопротивление. Чтобы определить высоту давления, используется высоту взлетно-посадочной полосы, атмосферное давление, высоту полета и показания альтиметра, установленного на самолете. Формула расчета высоты давления выглядит следующим образом:

$$h_d = h_n + p_a \cdot (h - a), \quad (1)$$

где h_d – высота давления,

h_n – подъем по высоте взлетно-посадочной полосы,

p_a – давление атмосферы,

a – показание альтиметра.

После определения высотного давления также следует рассчитать направление ветра. Сначала следует определить, является ли это встречный ветер или попутный. Используя следующую формулу:

$$V_{\text{бв}} = v_{\text{в}} \cdot \cos(\theta), \quad (2)$$

где $V_{\text{бв}}$ – скорость бокового ветра,

$v_{\text{в}}$ – скорость ветра,

θ – угол направления ветра, параллельный самолету.

Пример метод расчета скорости бокового ветра приведен на рисунке 5.

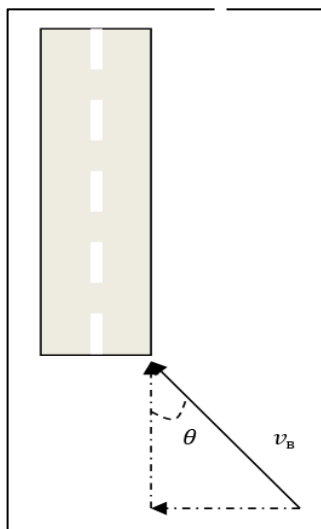


Рисунок 5 – Расчет скорость бокового ветра

После определения всех важных факторов пилот обращается к таблице посадочного расстояния, которая представлена на рисунке 6.

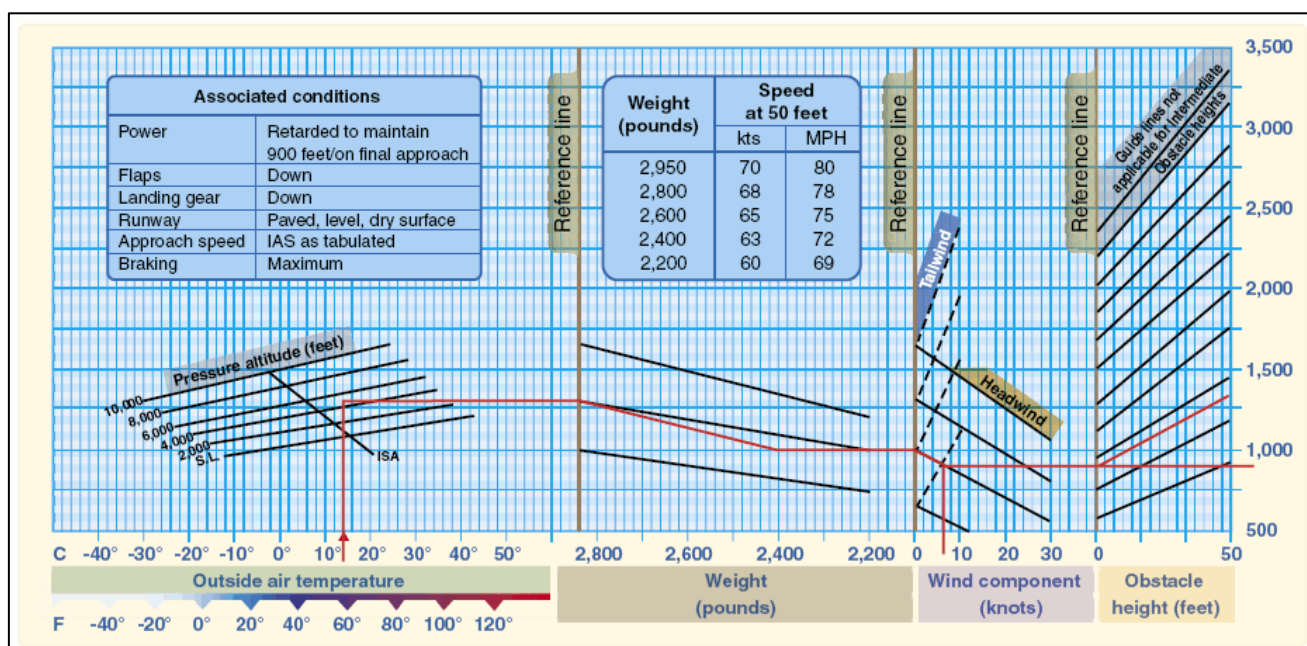


Рисунок 6 – Таблица посадочного расстояния

Эта таблица используется большинством пилотов для приблизительного определения расстояния, необходимого для посадки самолета и взлета. Это важно, потому что максимальное расстояние до взлетно-посадочной полосы покажет, может ли самолет взлететь или приземлиться в определенном месте.

1.6.2 Отличие в расчете посадочного расстояния с помощью интеллектуальной платформы.

Основное различие во времени остановки или расстоянии между обычной посадкой на стандартную взлетно-посадочную полосу и посадкой с помощью интеллектуальной платформы заключается в том, что платформа играет очень большую роль. Это связано с тем, что интеллектуальная посадочная платформа оснащена устройством фиксации, таким как то, которое используется на авианосцах и которое облегчает посадку в морской авиации. Стопорное устройство быстро замедляет самолет, прилагая значительное тормозное усилие за счет зацепления хвостового крюка за трос.

Отличие аэрофинишеров заключается в том, что пилоты, совершающие посадку на морских перевозчиках, должны быть хорошо обучены, чтобы иметь возможность совершать посадку с использованием аэрофинишера, в то время как интеллектуальная платформа оснащена системой слежения, которая определит расстояние до самолета и подключит их, как только самолет достигнет платформы. Это приводит к гораздо более высокой скорости торможения по сравнению с обычными колесными тормозами, используемыми при посадке на обычные взлетно-посадочные полосы.

Приблизительное расстояние, необходимое для посадки самолета с помощью интеллектуальной платформы, зависит от типа воздушного судна, скорости, размера, веса и условий эксплуатации. Если воздушное судно получило незначительные повреждения и для посадки потребовалась помощь интеллектуальной платформы, эффективность стопорного устройства на интеллектуальной платформе будет высокой, а если воздушное судно вообще не сможет контролировать свою скорость, интеллектуальной платформе потребуется большее расстояние или больше времени для полной остановки самолета. На рисунке 7 показан военно-морской самолет, совершающий посадку на авианосец, оснащенный системой удержания.

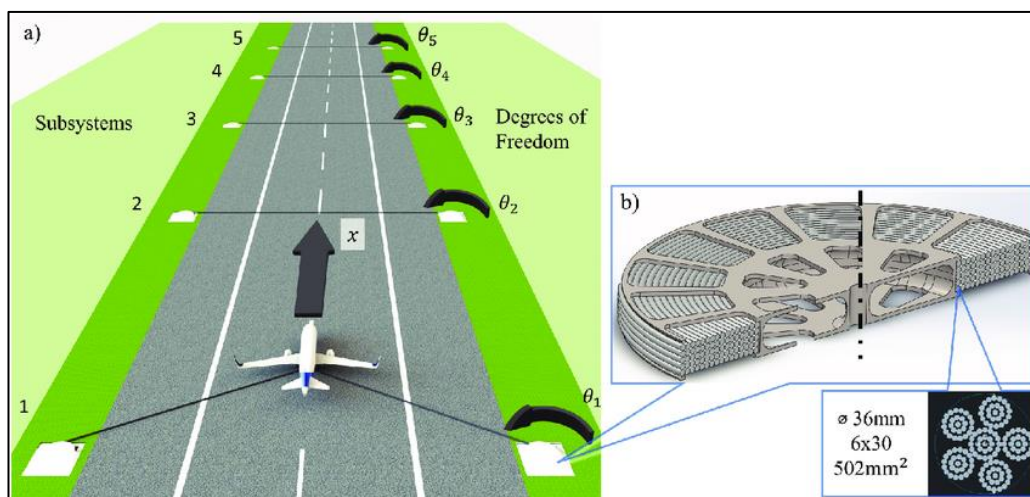


Рисунок 7 – Военно-морской самолет приземляется на авианосец, оснащенный системой удержания.

На авианосце, оснащенный системой стоп-сигналов, тормозной путь при посадке самолета может быть сокращен примерно на 50–75% по сравнению с обычной посадкой на взлетно-посадочную полосу без стоп-сигналов.

1.7 Комплексная система интеллектуальной посадочной платформы

1.7.1 Конструкция и принцип работы системы

Конструкция интеллектуальной платформы, предназначенной для поддержки аварийной посадки самолета, сложна. Это связано с тем, что аварийная посадка может произойти в любой ситуации, и интеллектуальная платформа должна быть способна оказать помощь в большинстве из них. Самолет, который готовится совершить аварийную посадку, может быть не в состоянии должным образом контролировать свою скорость, положение и оборудование. Чтобы компенсировать эту ситуацию и помочь самолету восстановиться и безопасно приземлиться, конструкция интеллектуальной платформы должна быть точной и хорошо оснащенной соответствующими датчиками и исполнительными устройствами.

Принцип работы интеллектуальной платформы заключается в ее способности отслеживать движение самолета. В этой работе мы отслеживаем движение самолета, отправляя сигнал с гироскопа, подключенного к самолету, на платформу посредством радиосвязи. Это предполагает движение по трем

осям. Оси OX, OY и OZ. Между платформами будет размещен индикатор, позволяющий пилоту лучше ориентироваться в положении платформы. Кроме того, предусмотрена система фиксации, позволяющая самолету не соприкасаться с платформой после того, как самолет достигнет платформы и установит контакт. И последнее, но не менее важное – это тормозная система, которая постепенно помогает самолету замедляться быстрее по сравнению с торможением при обычной посадке на обычную взлетно-посадочную полосу.

1.7.2 Система слежения за самолетом на платформе

В области авиационной безопасности и готовности к чрезвычайным ситуациям разработка интеллектуальной посадочной платформы, оснащенной передовыми технологиями слежения и управления движением, представляет собой революционную инновацию. Эта передовая платформа предназначена для точной и эффективной аварийной посадки, обеспечивая безопасность как воздушного судна, так и его пассажиров. В основе этой системы лежит сложный гироскоп, способный определять положение самолета по трем различным осям – тангажу, крену и рысканию. Этот важнейший датчик передает данные в режиме реального времени в систему слежения платформы, позволяя ей плавно отражать движение самолета при заходе на посадку.

Инновационный механизм слежения, встроенный в интеллектуальную посадочную платформу, является ключевым компонентом ее функциональности. Точно имитируя движения самолета, платформа может динамически изменять свое положение, обеспечивая максимально возможную посадочную поверхность. Эта адаптивная система слежения дополнена тремя независимыми двигателями, каждый из которых управляет своей осью. Эти двигатели работают автономно, чтобы быстро адаптироваться к внезапным изменениям ориентации самолета, обеспечивая точное выравнивание между самолетом и посадочной платформой в критические моменты аварийной посадки.

Когда самолет приближается к интеллектуальной посадочной площадке, система слежения прекращает свое движение и фиксирует положение

самолета. Это решительное действие гарантирует, что посадочная площадка остается стабильной и безопасной, готовой принять приближающийся самолет. Как только самолет соприкасается с платформой, включаются двигатели, чтобы постепенно выровнять самолет в параллельном положении для контролируемого снижения. Такая организованная последовательность событий подчеркивает эффективность платформы в обеспечении безопасной и стабильной обстановки при аварийных посадках, снижении рисков и повышении общей безопасности авиационных операций.

1.7.3 Яркий индикатор, помогающий пилотам легко видеть платформу

Четкое визуальное изображение при посадке действительно важно, а световые индикаторы очень помогают пилотам, особенно во время критических ситуаций при посадке. Как часть традиционных систем освещения взлетно-посадочной полосы, различные цветные огни стратегически расположены вдоль взлетно-посадочной полосы, чтобы обеспечить пилотов необходимыми визуальными сигналами во время посадки. Однако во время полетов в ночное время или при неблагоприятных погодных условиях эти огни может быть трудно различить, что усложняет процесс посадки для пилотов. Одних только фар самолета часто бывает недостаточно для четкого освещения взлетно-посадочной полосы и безопасного снижения.

Для решения этих проблем установка ярких световых индикаторов непосредственно на платформе аварийной посадки улучшает видимость и ориентирует пилотов, совершающих аварийную посадку. Эти видимые огни, отличающиеся разными цветами, служат важными маркерами, обозначающими границы платформы и четко разграничивающими начальную и конечную точки. Предоставляя это визуальное руководство непосредственно на посадочной платформе, пилоты получают более полную информацию о ситуации и ориентирах, что способствует более плавной и точной посадке даже в неблагоприятных условиях.

Кроме того, интеграция ярких световых индикаторов на платформе аварийной посадки дополняет существующие навигационные системы самолета и помогает повысить общую точность посадки. Сочетание этих фонарей с навигационными приборами самолета обеспечивает бесперебойную координацию между бортовыми системами и наземными визуальными сигналами, повышая уверенность пилота и эффективность принятия решений во время аварийных посадок. Такая симбиотическая взаимосвязь между технологическими показателями и авиационными системами способствует повышению эксплуатационной безопасности и эффективности в критических ситуациях посадки. На рисунке 8 показано, как работает подсветка на взлетно-посадочной полосе, которая помогает улучшить видимость.

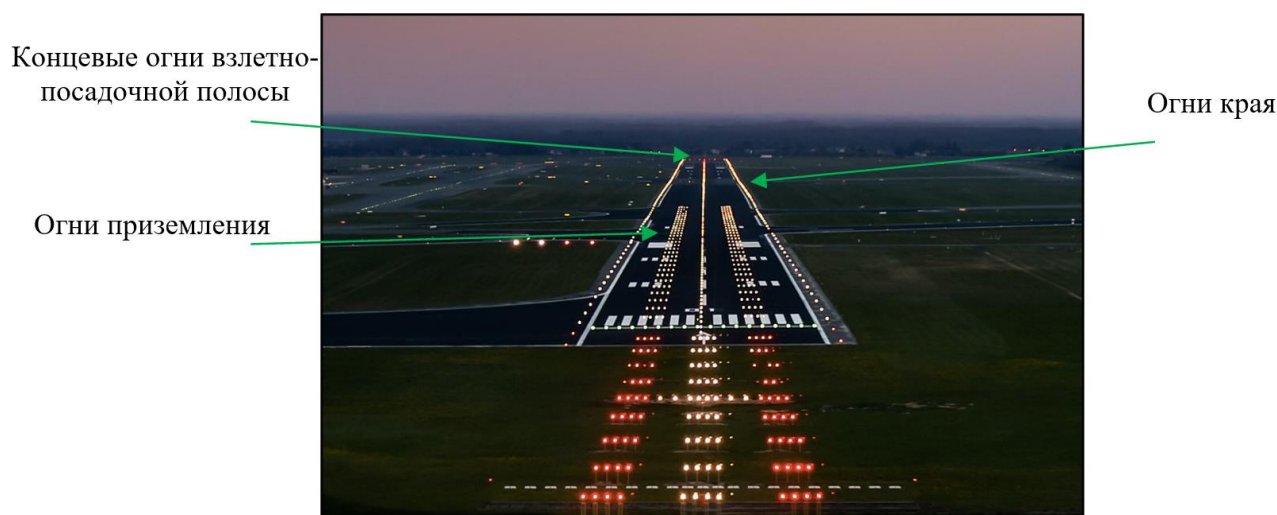


Рисунок 8 – Подсветки на взлетно-посадочной полосе

В таблице 1 приведено описание цвета огней на взлетно-посадочной полосе.

Таблица 1 – Описание цвета огней на взлетно-посадочной полосе

Цвета огней на взлетно-посадочной полосе	Описание
Зелёный и красный	Конец взлетно-посадочной полосы
Белый и жёлтый	Края взлетно-посадочной полосы
Белый	Зона приземления
Белый и красный	Осевая линия взлетно-посадочной полосы

В заключение следует отметить, что внедрение интеллектуальной платформы аварийной посадки, оснащенной видимыми яркими световыми индикаторами, представляет собой значительный прогресс в области авиационной безопасности. Благодаря использованию возможностей систем наведения на основе света пилоты получают улучшенную видимость и четкость при аварийных посадках, особенно в сложных ночных или неблагоприятных погодных условиях. Интеграция этих ярких световых индикаторов не только способствует более безопасной посадке, но и усиливает общую синергию между авиационными навигационными технологиями и наземными визуальными средствами, что в конечном итоге оптимизирует возможности реагирования на чрезвычайные ситуации и обеспечивает максимальную безопасность полетов.

1.7.4 Система фиксации для сохранения целостности самолета и платформы после установления контакта

Система интеллектуальной посадочной платформы оснащена сложной системой фиксации. Система фиксации играет ключевую роль в надежном прикреплении самолета к контакту крепления платформы, повышая общую стабильность и безопасность процесса посадки. Используя усовершенствованные датчики расстояния, автоматически включается механизм фиксации самолета на месте, применяя фундаментальные физические принципы передачи импульса для поглощения кинетической энергии самолета при контакте.[2] Этот инновационный подход не только предотвращает возможные столкновения самолета с платформой, но и обеспечивает контролируемый и плавный переход, когда платформа начинает двигаться в тандеме с закрепленным самолетом.

Кроме того, синхронизированное движение платформы с подрезанным самолетом дополняется сложной системой торможения, которая обеспечивает дополнительный уровень контроля и безопасности в процессе аварийной посадки. Поскольку самолет надежно прикреплен к платформе с помощью системы фиксации, встроенный тормозной механизм обеспечивает плавное и

контролируемое перемещение, способствуя стабильной и скоординированной посадке. Это гармоничное сочетание технологии фиксации и тормозных систем демонстрирует целостный подход к авиационной безопасности, использующий инновационные инженерные решения для повышения возможностей реагирования на чрезвычайные ситуации и снижения рисков, связанных с критическими ситуациями посадки. На рисунке 9 представлена система отсечения до и после обнаружения присутствия самолета.

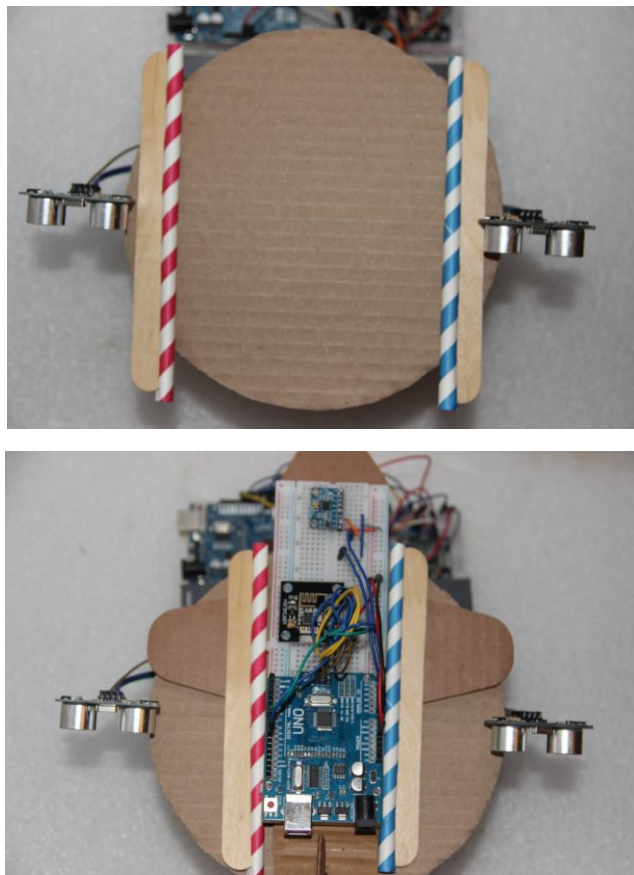


Рисунок 9 – Система фиксации до и после обнаружения присутствия самолета

Система фиксации обеспечивает достаточное натяжение, чтобы удерживать самолет и в то же время не повредить его.

1.7.5 Тормозная система, в которой используется аэрофинишёр

Обычно в морской авиации аэрофинишёр используется для быстрого торможения самолета из-за недостаточной длины взлетно-посадочной полосы. В системе интеллектуальной посадочной платформы, аэрофинишёр служит той же цели, но зависит от типа аварийной посадки, выполняемой самолетом.

В морской авиации пилот проходит специальную подготовку, чтобы убедиться, что крюк точно прикреплен к тросам аэрофинишёра.[3] Отличие интеллектуальной платформы в том, что в этой системе есть система фиксации, которая может подсоединять приземляющийся самолет к тормозной системе аэрофинишёра.

Эффективность средств фиксации при безопасном торможении самолета во время посадки трудно переоценить, особенно при заходе на посадку на высокой скорости на авианосцах или в чрезвычайных ситуациях. Когда самолет достигает платформы, система фиксации захватывает самолет, инициируя контролируемый и мощный процесс торможения. Этот метод позволяет значительно сократить тормозной путь по сравнению с использованием исключительно колесных тормозов и аэродинамического сопротивления, что крайне важно при работе в ограниченном пространстве и сложных условиях.

Ключевое преимущество тормозная система интеллектуальной посадочной платформы заключается в его способности обеспечивать надежное и предсказуемое замедление независимо от скорости захода на посадку или веса воздушного судна. Используя энергию инерции самолета, система остановки быстро поглощает кинетическую энергию за счет сил трения, эффективно и контролируемо останавливая самолет. Надежность и эффективность системы вносят значительный вклад в общую безопасность, снижая риск выезда за пределы взлетно-посадочной полосы или возникновения аварийных ситуаций, связанных с отклонением взлетно-посадочной полосы во время критических посадочных маневров.

1.8 Выводы к первому разделу и постановка задачи

В этой части работы были рассмотрены основные потребности самолета в безопасной посадке, что показывает нам, что у самолета есть много важных потребностей и подготовки к посадке, которые необходимо учитывать еще до взлета. Также были рассмотрены основные части самолета, участвующие в

посадке. Это включает в себя шасси, закрылки и амортизатор, которые используются для управления самолетом во время посадки. Спойлеры и тормоза используются для снижения скорости после обычной посадки.

Далее были рассмотрены процессы захода на посадку и приземления летательного аппарата. Этот процесс сложен и может быть выполнен только квалифицированным пилотом. Процесс начинается с момента, когда самолет связывается с диспетчерской, чтобы получить разрешение на посадку в определенном аэропорту, и заканчивается, когда самолет полностью останавливается. Эти процессы должны выполняться тщательно, чтобы обеспечить безопасность всех экипажей и пассажиров, включая самого пилота.

В следующей части мы рассмотрим типы аварийных ситуаций, которые могут возникнуть во время посадки летательного аппарата. Существует множество факторов, которые могут привести к аварийной посадке, в том числе неисправность шасси самолета. Кроме того, если возникнет какая-либо проблема с какими-либо частями летательного аппарата, самолету также потребуется выполнить аварийную посадку. Однако проблема связана не только с оборудованием самолета, но и может быть вызвана ошибками самих пилотов. Пилот, как человек, также может случайно совершить ошибку, которая может привести к аварийной посадке. Человеческие ошибки и ошибки оборудования – не единственные факторы, приводящие к аварийной посадке. Состояние взлетно-посадочной полосы и его окружения также может привести к аварийной посадке самолета.

В следующей части также был рассмотрен исторический опыт, который доказал, что аварийная посадка – это непростая проблема. Она затрагивает жизни сотен людей одновременно и может быть очень травмирующей для тех, кому даже посчастливилось пережить подобную трагедию. И чтобы убедить в актуальности этого ВКР, были обсуждены расчеты, которые обычно выполняются пилотом перед оценкой расстояния, необходимого ему для безопасной посадки в определенном месте. Затем он сравнивается с процентом

эффективности системного управления аварийной посадки, чтобы снизить процент рисков и повысить вероятность успешного выполнения плавной и безопасной посадки при возникновении аварийных ситуаций.

Наконец, рассмотрена сложная система интеллектуальной посадочной платформы. Конструкция платформы и принцип ее работы описаны, чтобы дать общее представление о том, как платформа может помочь пилоту успешно посадить самолет. Существует несколько систем, которые работают сообща и объединяются для обеспечения точности и функциональности интеллектуальной посадочной платформы. Это система слежения, яркие индикаторы для улучшения видимости платформы и комбинированная система торможения, которая играет большую роль в замедлении полета самолета.

Учитывая вышеуказанное, в настоящей ВКР поставлена задача моделирования (прототипирования) системы управления аварийной посадки летательного аппарата, который обеспечивает плавную и безопасную посадку, несмотря на проблемы, с которыми сталкиваются самолеты, нуждающиеся в аварийной посадке.

Таким образом, в настоящей ВКР моделируется управления:

- слежением положения самолета;
- тормозной системой посадочной платформы;
- индикацией положения интеллектуальной посадочной платформы.

2 ВЫБОР АППАРАТНЫХ И ПРОГРАММНЫХ СРЕДСТВ РЕАЛИЗАЦИИ СИСТЕМЫ АВАРИЙНОЙ ПОСАДКИ.

2.1 Выбор средств аппаратной реализации

2.1.1 Общие знания о микроконтроллере и ознакомление с Arduino

Платформы для микроконтроллеров в настоящее время довольно дешевы, и они очень хорошо функционируют. Существует множество известных микроконтроллеров, таких как Arduino, IskraJS, Piranha и другие. [4] Они очень ценны для студентов, которые учатся и хотят создавать новые проекты, поскольку микроконтроллер может эффективно выполнять задачу в одном времени.

Самой известной платформой, которая широко используется, является Raspberry pi, работающая на операционной системе, которая может быть запущена на полностью оборудованном компьютере. Обычно люди легко запускают Linux на Raspberry pi, поскольку микроконтроллер очень мощный и современный.

Arduino Uno – это популярная плата для микроконтроллеров, которая обладает рядом преимуществ для проектов в области электроники и программирования, что делает ее предпочтительным выбором как для любителей, студентов, так и для профессионалов. Arduino Uno разработан с учетом подхода, удобного для начинающих, что делает его доступным для пользователей с различным уровнем опыта в электронике и программировании. Он оснащен простой и интуитивно понятной интегрированной средой разработки (IDE), которая поддерживает широкий спектр языков программирования, включая C и C++, что позволяет пользователям быстро писать и загружать код на плату. Arduino Uno поддерживает широкий спектр электронных компонентов и модулей с помощью стандартных контактов ввода-вывода (I/O), что открывает бесконечные возможности для реализации проектов. Он может взаимодействовать с датчиками, исполнительными механизмами, дисплеями,

двигателями и коммуникационными модулями, такими как Wi-Fi и Bluetooth, позволяя пользователям создавать разнообразные проекты от робототехники и автоматизации до приложений Интернета вещей.

Arduino пользуется преимуществами большого и динамичного сообщества пользователей, создателей и разработчиков, которые активно участвуют в форумах, блогах и онлайн-репозиториях. Эта система поддержки, управляемая сообществом, гарантирует, что пользователи смогут легко находить ресурсы, учебные пособия и рекомендации по устранению неполадок для своих проектов Arduino. Независимо от того, ищет ли новичок руководства или опытный разработчик передовых технологий, богатство знаний, которыми делится сообщество Arduino, ускоряет обучение и решение проблем.

Одной из отличительных особенностей Arduino Uno является его доступность по цене и широкому распространению. Сама плата относительно недорогая по сравнению с другими микроконтроллерными платформами, что делает ее доступной для широкой аудитории, включая студентов, любителей и преподавателей. Кроме того, широко производятся компоненты, совместимые с Arduino, такие как датчики, приводы и экраны, по конкурентоспособным ценам, что еще больше снижает барьер для участия в разработке проектов. Такая доступность в сочетании с широкой доступностью гарантирует, что любой желающий может экспериментировать и внедрять инновации с технологией Arduino без значительных финансовых вложений. На рисунке 10 показан микроконтроллер Arduino, который можно найти в интернет-магазине всего за 400 рублей [20].



Рисунок 10 – Микроконтроллер Arduino Uno

Arduino Uno построен на принципах открытого исходного кода, и его аппаратные средства и библиотеки программного обеспечения доступны для модификации и распространения. Принцип открытого исходного кода способствует инновациям и сотрудничеству в сообществе разработчиков, позволяя пользователям настраивать платформу Arduino в соответствии с конкретными требованиями проекта. Используя открытый исходный код Arduino, разработчики могут расширить возможности платы, поделиться улучшениями с сообществом и принять участие в непрерывном развитии платформы. Такая гибкость и свобода позволяют пользователям исследовать новые возможности и расширять границы того, чего можно достичь с помощью технологии Arduino.

2.1.2 Компоненты микроконтроллера Arduino UNO

Платформа Arduino лучше всего подходит для работы над этими проектами благодаря своей простоте в использовании. Она состоит из нескольких ключевых компонентов, которые работают вместе для выполнения различных задач. В основе платы Arduino лежит микроконтроллер – универсальная и мощная интегральная схема, которая выполняет множество задач. Понимание ключевых компонентов этого микроконтроллера, включая центральный процессор, память, порты ввода-

вывода и периферийные устройства, обеспечивает основу для полного использования его потенциала в различных проектах.

Центральный процессор (ЦП) является мозгом микроконтроллера Arduino. В Arduino Uno это – микроконтроллер ATmega328P. Центральный процессор выполняет программные инструкции, выполняет арифметические и логические операции и управляет другими компонентами. Он управляет потоком данных и обрабатывает входные данные от датчиков и других устройств, подключенных к Arduino, принимая решения на основе программного кода. ATmega328P построен на архитектуре AVR, которая представляет собой 8-разрядную вычислительную систему с сокращенным набором команд (RISC). Это означает, что он обрабатывает данные в виде 8-разрядных блоков и использует упрощенный набор инструкций для выполнения операций. Архитектура RISC обеспечивает более быструю обработку за счет минимизации количества циклов на команду, что делает центральный процессор эффективным и способным быстро справляться с различными задачами. Центральный процессор Arduino Uno работает с тактовой частотой 16 МГц. Центральный процессор ATmega328P оснащен усовершенствованной системой обработки прерываний. Прерывания – это сигналы, которые временно останавливают текущие операции центрального процессора для выполнения определенной задачи, такой как считывание данных с датчика или реакция на нажатие кнопки. Центральный процессор Arduino Uno, микроконтроллер ATmega328P, представляет собой компактный, но мощный процессор, который позволяет использовать широкий спектр приложений.

Память – еще один важный компонент микроконтроллера Arduino Uno. Она делится на три типа: ОЗУ (оперативная память), ПЗУ (встроенная память) и флэш-память. ОЗУ используется для временного хранения данных во время выполнения программы, обеспечивая пространство для переменных и промежуточных результатов. Объем оперативной памяти Arduino Uno составляет 2 КБ, что относительно невелико, но достаточно для многих

простых приложений. ПЗУ содержит загрузчик – небольшую программу, которая позволяет программировать Arduino с помощью компьютера. Эта память обычно не используется для хранения обычных программ. Флэш-память, с другой стороны, хранит фактический программный код (скетч) и сохраняет его даже при выключенном питании. Arduino Uno имеет 32 КБ флэш-памяти, небольшая часть которой зарезервирована для загрузчика.

Порты ввода-вывода (I/O) имеют решающее значение для взаимодействия Arduino с внешними устройствами. Arduino Uno оснащен 14 цифровыми контактами ввода-вывода и 6 аналоговыми входными контактами. Цифровые контакты могут быть сконфигурированы как входы или выходы, что позволяет микроконтроллеру считывать цифровые сигналы с датчиков или управляющих устройств, таких как светодиоды, двигатели и реле. Аналоговые контакты, подключенные к аналого-цифровому преобразователю (АЦП), используются для считывания аналоговых сигналов с различных датчиков. АЦП преобразует эти аналоговые сигналы в цифровые значения, которые может обрабатывать центральный процессор, что позволяет Arduino работать с широким спектром типов входных данных. На рисунке 11 показан Arduino UNO с обозначениями всех портов ввода-вывода на нем.

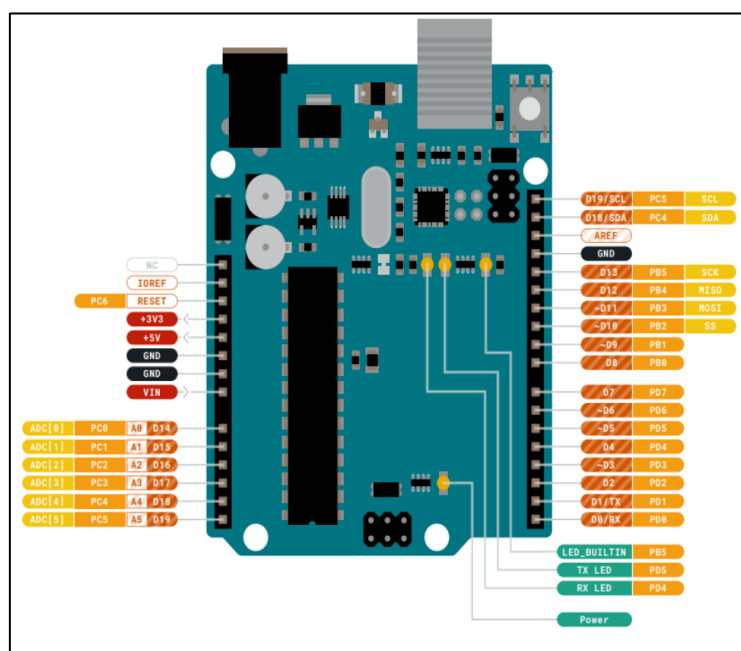


Рисунок 11 – Обозначения портов Arduino UNO

В таблице 2 написаны названия портов ввода-вывода Arduino UNO и их описание.

Таблица 2 – Порты ввода-вывода Arduino UNO

ИМЯ	ТИП	ОПИСАНИЕ
NC	Отсутствие связи	Отсутствие связи
IOREF	Опорное напряжение	Опорный сигнал для цифровой логики V – подключен к 5 В
Reset	Сброс	Сброс
+3V3	Питание	+3V3 силовая шина
+5V	Питание	+5V силовая шина
GND	Питание	земля
GND	Питание	земля
VIN	Питание	входное напряжение
A0	Аналоговый/Ввод-вывод общего назначения (GPIO)	Аналоговый вход 0 /GPIO
A1	Аналоговый/(GPIO)	Аналоговый вход 0 /GPIO
A2	Аналоговый/(GPIO)	Аналоговый вход 0 /GPIO
A3	Аналоговый/(GPIO)	Аналоговый вход 0 /GPIO
A4/SDA	Аналоговый вход/ I2C	Аналоговый вход 0 /GPIO/ Линия передачи данных
A5/SCL	Аналоговый вход/ I2C	Аналоговый вход 0 /GPIO/ часовая линия
D0	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D1	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D2	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D3	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D4	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D5	Цифровой/(GPIO)	Цифровой вывод 0/GPIO

Продолжение таблицы 2

D6	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D7	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D8	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
D9	Цифровой/(GPIO)	Цифровой вывод 0/GPIO
SS	Цифровой	Выбор микросхемы последовательного периферийного интерфейса(SPI)
MOSI	Цифровой	SPI Основной Выход Вторичный вход
SCK	Цифровой	Основной вход Вторичный выход
SD4	Цифровой	Аналоговый вход 4/ I2C Линия передачи данных (дублирование)
SD5	Цифровой	Аналоговый вход 5/ I2C часовая линия (дублирование)

В Arduino также встроены периферийные устройства, которые расширяют его возможности и универсальность. Например, таймеры используются для выполнения задач с определенными интервалами или для измерения продолжительности времени, позволяя точно управлять такими процессами, как ШИМ (широтно-импульсная модуляция) для затемнения светодиодов или регулирования частоты вращения двигателя. АЦП – это периферийное устройство, которое преобразует аналоговые сигналы в цифровые значения, позволяя микроконтроллеру обрабатывать входные сигналы от аналоговых датчиков. Arduino UNO не имеет встроенного цифроаналогового преобразователя (ЦАП), но предлагает дополнительный

модуль, который может обеспечить эту функциональность, расширяя спектр возможных применений.

В заключение отметим, что компоненты микроконтроллера Arduino – процессор, память, порты ввода-вывода и периферийные устройства – в совокупности обеспечивают его мощную и гибкую работу, что действительно помогает в этом проекте. Доступность и универсальность Arduino делают его бесценным инструментом в области электроники и программирования, способствующим инновациям и творчеству широкого круга пользователей. Arduino хорошо взаимодействует со всеми датчиками, используемыми в этом проекте. И выходные данные также плавно передаются исполнительными устройствами, которые также подключены к Arduino UNO.

2.1.3 Датчики связи между двумя микроконтроллерами

Коммуникация для Arduino – это очень обширная тема, и сейчас на рынке доступно множество вариантов. Они могут взаимодействовать с помощью последовательной связи (UART), I2C-связи, SPI-связи, Bluetooth-связи, Wi-Fi связи и радиосвязи. Лучшим выбором, выбранным для этого проекта, является радиосвязь.

Радиосвязь используется с 1895 года [7], и с тех пор она широко используется по всей стране для связи и других целей. В Arduino есть множество радиомодулей на выбор. Среди них NRF24L01, LoRa, CC1101, HC12, 433 МГц и HC 05. При работе с некоторыми проектами, требующими беспроводной связи, некоторые радиомодули выделяются своей производительностью, дальностью действия, энергопотреблением и простотой использования. В таблице 3 представлено сравнение пяти лучших радиомодулей, доступных для Arduino.

Таблица 3 – Сравнение радиомодулей, доступных для Arduino

Радиомодул и	NRF24L01	HC-12	LoRa SX1278	XBee Series 2	ESP8266
Особенност и					

Частота	2,4 ГГц	443МГц	443МГц	2,4 ГГц	2,4 ГГц
Скорость передачи данных	До 2 МБ/с	До 115,2 кБ/с	До 300кБ/с	До 250 кБ/с	До 54 МБ/с
Диапазон	До 100м	До 1.8 км	До 15км	До 1.6км	До 100м
Потребляемая мощность: • передавать • принимать	<ul style="list-style-type: none"> • 11.3 мА • 13.5 мА 	<ul style="list-style-type: none"> • 100мА • 16мА 	<ul style="list-style-type: none"> • 120 мА • 10.8 мА 	<ul style="list-style-type: none"> • 40 мА • 40 мА 	<ul style="list-style-type: none"> • 170мА • 50мА
Интерфейс	SPI	UART	SPI	UART	UART
Преимущество	Низкое энергопотребление	Отличная дальность стрельбы и способность преодолевать препятствия.	Высокая устойчивость к помехам и многолучевым затуханиям	Подходит для сложных сетевых топологий	Универсален и может функционировать как в качестве модуля Wi-Fi, так и в качестве микроконтроллера
Недостатки	подвержен помехам в переполненном диапазоне частот 2,4 ГГц	Более высокое энергопотребление в режиме передачи	Более низкая скорость передачи данных по сравнению с другими модулями	Более высокая стоимость по сравнению с более простыми модулями	Более сложная настройка для выполнения основных задач радиосвязи

Из всех доступных вариантов, которые можно использовать с Arduino UNO, NRF24L01 считается лучшим с точки зрения цены, расстояния, скорости и энергопотребления. NRF24L01+ – это однокристалльный приемопередатчик с частотой 2,4 ГГц и встроенным протоколом базовой полосы частот, который подходит для беспроводных приложений со сверхнизким энергопотреблением. Он может осуществлять связь на расстоянии до 100 метров в открытом пространстве.

NRF24L01+ – это универсальный приемопередающий модуль, работающий в ISM-диапазоне 2,4 ГГц, что делает его подходящим для различных промышленных, научных и медицинских применений. Он поддерживает скорость передачи данных 250 Кбит/с, 1 Мбит/с и 2 Мбит/с, обеспечивая гибкость в скорости и диапазоне передачи данных. Отличительной особенностью является низкое энергопотребление: потребляемая мощность при передаче составляет от 11,3 мА при 0 дБм до всего лишь 7,0 мА при –18 дБм, а в режиме приема потребляется 13,5 мА. Для дальнейшей экономии энергии модуль включает режимы отключения питания и ожидания. nRF24L01+ работает в диапазоне напряжений от 1,9 до 3,6 В, обычно питаясь от напряжения 3,3 В. Он поставляется со встроенной антенной на печатной плате, а также доступны варианты с внешними антеннами для обеспечения большего радиуса действия. Взаимодействие с микроконтроллерами осуществляется через последовательный периферийный интерфейс (SPI). Усовершенствованный протокол ShockBurst™ обеспечивает высокоскоростную передачу данных с автоматической обработкой пакетов, повышая надежность. Модуль поддерживает 125 радиочастотных каналов, что позволяет нескольким устройствам взаимодействовать без помех. Кроме того, он оснащен функцией многоканальная коммуникационная труба, позволяющей управлять до 6 каналами передачи данных для одновременной связи с несколькими устройствами, что делает его идеальным выбором для сложных сетей с несколькими устройствами.

Модуль nRF24L01+ имеет компактную конструкцию с 8 выводами, которые облегчают его подключение и работу с микроконтроллерами [7]. В распиновку входит VCC, который обеспечивает питание модуля типичным рабочим напряжением 3,3 В. Вывод GND предназначен для заземления. Вывод CE (Chip Enable) активирует режим RX или TX, если он установлен на высоком уровне. CSN (Chip Select Not) используется для включения связи SPI; он активен на низком уровне. SCK (Serial Clock) – это линия синхронизации SPI, в то время как MOSI (Master Out Slave In) и MISO (Master In Slave Out)

являются линиями передачи данных SPI для связи. Кроме того, вывод IRQ (запрос на прерывание) используется для оповещения микроконтроллера о появлении новых данных для считывания или о завершении передачи. Такая конфигурация вывода обеспечивает эффективное и универсальное подключение к различным платформам микроконтроллеров, обеспечивая надежную беспроводную связь. На рисунке 12 четко показан модуль NRF24L01+ с его выводами.

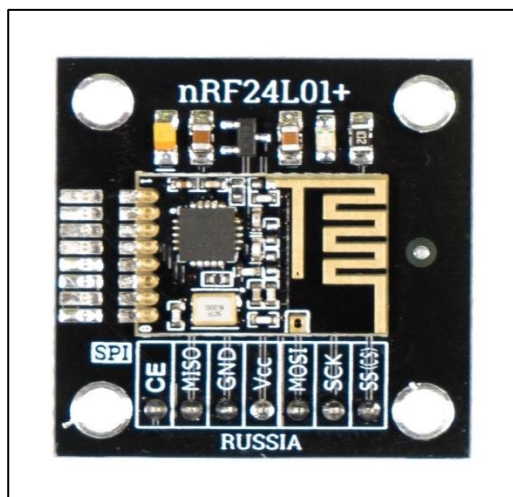


Рисунок 12 – Модуль NRF24L01+ и его выводы

Модуль nRF24L01+ работает в нескольких режимах для оптимизации энергопотребления и производительности в зависимости от требований приложения. Его основные режимы работы включают отключение питания, режим ожидания, RX (прием) и TX (передача). В режиме отключения модуль потребляет минимальную мощность, что делает его идеальным для экономии энергии, когда устройство не используется активно. Режим ожидания позволяет быстрее переключаться в активные режимы при сохранении низкого энергопотребления. В режиме RX модуль активно прослушивает поступающие данные, потребляя больше энергии, чем в режиме ожидания, но обеспечивая надежный прием сигналов. Для передачи данных используется режим TX, в котором потребляемая модулем мощность варьируется в зависимости от выбранного уровня мощности передачи. nRF24L01+ также оснащен усовершенствованным режимом ShockBurst™, который управляет

высокоскоростной передачей данных с автоматической обработкой пакетов, обеспечивая эффективную связь при минимальном вмешательстве программного обеспечения. Эти универсальные режимы работы позволяют модулю сбалансировать энергоэффективность и производительность, что делает его подходящим для широкого спектра приложений беспроводной связи.

Модуль обрабатывает пакетирование данных, включая адресацию и циклическую проверку избыточности. При получении пакета модуль может автоматически подтвердить его получение, что обеспечивает надежную связь между модулями. Этот модуль поддерживает полезную нагрузку данных переменной длины от единицы до 32 байт. Этот модуль также поддерживается библиотекой RF24, которая очень проста в использовании. Библиотека широко используется и пользуется поддержкой сообщества, которое постоянно обновляет качество библиотеки, чтобы улучшить программирование этого модуля. В этом проекте модули используются в качестве источника связи для интеллектуальной посадочной платформы для получения данных с самолета.

2.1.4 Датчики измерения расстояние между платформой и самолетом

С помощью Arduino можно измерить расстояние многими способами. Одним из распространенных методов измерения расстояния является использование ультразвукового волнового датчика. Этот метод основан Сергеем Яковлевичем Соколовым, который является советским физиком (1897-1957) [9]. Расстояние возможно определить с помощью этого метода по формуле 3, приведенной.

$$d = \frac{t * v}{2}, \quad (3)$$

где t – время получения даваемый сигнал,

v – скорость звука.

HC-SR04 – это широко используемый ультразвуковой датчик, предназначенный для точного и надежного измерения расстояний в различных

областях применения. Этот датчик работает за счет излучения ультразвуковых звуковых волн с частотой 40 кГц через свой передатчик, которые затем отражаются от объектов и возвращаются к приемнику датчика. Время, необходимое для возвращения эхо-сигнала, измеряется и используется для расчета расстояния до объекта на основе скорости звука. На рисунке 13, показанном, можно увидеть датчик HC-SR04.



Рисунок 13 – Датчик ультразвуковой HC-SR04

Датчик работает при напряжении 5 В постоянного тока, что делает его совместимым с большинством микроконтроллеров, включая Arduino, который используется в этом проекте. Регулятор напряжения не понадобится, так как Arduino UNO может легко обеспечить питание датчика напряжением 5 В. Ультразвуковой датчик может измерять расстояния от двух до 400 см с разрешением 0,3 см. Он потребляет относительно мало энергии, около 15 мА во время работы. Для запуска измерения датчику также требуется короткий импульс длительностью 10 микросекунд на спусковом штифте. Для текущего проекта очень важна точность измерений, и, согласно техническому описанию ультразвукового датчика HC-SR04, его погрешность минимальна и составляет около ± 3 мм [10].

Для повышения точности измерений и минимизации ошибок при измерении учитываются шумы. Для минимизации этих ошибок используется фильтрация сигналов. Шум можно разделить на два типа которые являются постоянным и случайным [11]. Различные шумы можно отфильтровать с помощью цифровые фильтры. Используется фильтр бегущее среднее

арифметическое, который работает по принципу буфера. Несколько последних измерений хранятся в буфере для усреднения. После появления нового значения вызывается фильтр, который перемещает буфер. Затем удаляется самое старое значение и в буфер добавляется новое значение. Затем буфер усредняется с использованием среднего арифметического значения, соответствующего формуле 4, приведенной.

$$A = \frac{1}{n} \sum_{i=1}^n a_i, \quad (4)$$

где A – среднее арифметическое значение,

n – число значения,

a_i – значения набора данных.

Фильтр среднего арифметического – один из самых простых и эффективных алгоритмов фильтрации данных, обеспечивающий простое, но в то же время высокоэффективное решение. Его основное преимущество заключается в простоте и быстродействии, что упрощает его реализацию и вычисления. Эффективность фильтра можно регулировать с помощью коэффициента в диапазоне от нуля до 1,0, при этом меньший коэффициент обеспечивает более плавный эффект фильтрации. Этот фильтр выполняет только одно измерение за раз, гарантируя, что он не блокирует код на длительный период, что особенно полезно для приложений реального времени. Кроме того, при более частых измерениях производительность фильтра повышается, что позволяет получать более точные и стабильные результаты. При использовании небольших значений коэффициента фильтр среднего арифметического работает очень плавно, что делает его подходящим для приложений, требующих плавных и постепенных изменений данных.

Ультразвуковой датчик HC-SR04 оснащен четырьмя выводами, которые необходимы для его работы и интеграции с микроконтроллерами. Вывод VCC обеспечивает питание датчика напряжением 5 В постоянного тока, что делает его совместимым с большинством плат микроконтроллеров. Вывод GND

обеспечивает необходимое заземление. Вывод триггера (Trig) используется для инициирования процесса измерения; короткий 10-микросекундный импульс, посылаемый на этот вывод, запускает ультразвуковой импульс датчика. Вывод Echo принимает отраженный сигнал; оно остается высоким в течение всего времени, в течение которого излучаемые ультразвуковые волны отражаются от объекта и возвращаются к датчику. Измерив время, в течение которого эхо-сигнал остается высоким, можно рассчитать расстояние до объекта. Такая простая конфигурация контактов позволяет датчику HC-SR04 легко взаимодействовать с микроконтроллерами для широкого спектра задач измерения расстояния.

Разбив процесс на несколько этапов, можно было бы лучше понять, как работает ультразвуковой датчик. Ультразвуковой датчик HC-SR04 измеряет расстояние с помощью процесса излучения и приема ультразвуковых звуковых волн. Сначала на контакт триггера (Trig) подается импульс длительностью 10 микросекунд, в результате чего датчик испускает 8-циклический ультразвуковой импульс частотой 40 кГц из своего передатчика. Эти звуковые волны распространяются по воздуху и отражаются от любого объекта на своем пути. Отраженные волны, или эхо-сигналы, затем улавливаются приемником датчика. Уровень эхо-сигнала увеличивается на время, необходимое звуковым волнам для прохождения до объекта и обратно. Эта высокая длительность сигнала прямо пропорциональна расстоянию от объекта до датчика. Путем определения времени и скорости звука, которая при комнатной температуре составляет приблизительно 343 метра в секунду, рассчитывается расстояние до объекта. Этот эффективный метод позволяет проводить точные и надежные измерения расстояния в различных областях применения.

2.1.5 Датчик гироскопа, измеряющий положение объекта

Гироскоп – это сложное устройство, используемое для измерения или поддержания ориентации на основе принципа углового момента. Оно состоит из вращающегося колеса или ротора, который установлен таким образом, что

его ось может принимать любую ориентацию. Ось вращения гироскопа при вращении сохраняет свое направление независимо от движения основания, к которому он прикреплен, благодаря сохранению углового момента. Это уникальное свойство делает гироскопы незаменимыми в различных областях применения от навигационных систем в самолетах и кораблях до стабилизирующих механизмов в смартфонах и камерах. Устойчивый к изменениям своей ориентации, гироскоп обеспечивает точные измерения вращательного движения, обеспечивая повышенную стабильность и управляемость во многих технологических и научных областях.

Гироскоп, который мы видим сейчас, отличается от того, каким он был в первые дни. Он претерпел значительные изменения, продемонстрировав принципы определения углового момента и стабильности. Концепция простых волчков восходит к глубокой древности, но современные гироскопы начали формироваться только в 19 веке. В 1817 году Иоганн Боненбергер построил раннюю версию, а в 1852 году Леон Фуко назвал ее гироскопом и использовал для демонстрации вращения Земли [12]. Этот период ознаменовался началом его практического применения, что привело к разработке Германом Аншютцем-Кемпфе в 1904 году гироскопического компаса, который произвел революцию в навигации. Мировые войны и последующие десятилетия способствовали значительному развитию гироскопических технологий для военных и аэрокосмических целей. Гиростабилизатор Элмера Сперри, разработанный в 1940-х годах, улучшил системы наведения торпед, самолетов и ракет. В последнее время появление микроэлектромеханических систем (MEMS) сделало гироскопы меньше и чувствительнее, что позволило интегрировать их в смартфоны, игровые консоли и носимые устройства. Сегодня гироскопы незаменимы в различных областях, от виртуальной реальности до автономных транспортных средств, демонстрируя постоянное совершенствование этой технологии и ее растущее значение в современной жизни.

Для платформы Arduino существует три основных типа гироскопов, которые очень популярны и широко используются. Это MPU6050, L3G4200D и MPU-9250. Все они являются гироскопами, но имеют несколько отличительных особенностей, которые можно учитывать при подготовке к определенным проектам. В таблице 4 приведены сравнения этих трех популярных гироскопических датчиков, которые можно использовать с Arduino.

Таблица 4 – Сравнение трех популярных гироскопических датчиков Arduino

Датчики Гироскопа	MPU6050	L3G4200D	MPU-9250
Особенности			
Компоненты	3-осевой гироскоп и 3-осевой акселерометр	3-осевой гироскоп	3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer
Дополнительные Функции	Встроенный цифровой процессор движения (DMP) для объединения датчиков и обработки движения	Встроенные фильтры высоких частот для устранения смещений и низкочастотных ошибок	Усовершенствованные алгоритмы обработки движения и слияние датчиков
Преимущества	<ul style="list-style-type: none"> – как правило, низкая стоимость и широкая доступность. – Сочетает в себе гироскоп и акселерометр, упрощая проекты по обнаружению движения. 	<ul style="list-style-type: none"> – Хорошая чувствительность и точность при измерениях вращения. – Два интерфейса (I2C и SPI) для обмена данными. 	<ul style="list-style-type: none"> – Включает в себя магнитометр для полного отслеживания движения по 9 осям. – Высокая точность и расширенные возможности для комплексного обнаружения движения и ориентации.

Недостатки	Изменение температуры может повлиять на производительность.	В комплект поставки не входит акселерометр, поэтому для полного отслеживания движения могут потребоваться дополнительные датчики.	Более сложный в использовании и программировании по сравнению с более простыми датчиками.
------------	---	---	---

В этом проекте используется датчик MPU-6050, поскольку он имеет 6-осевое отслеживание движения, которое измеряет угловую скорость в градусах в секунду вдоль оси крена, тангажа и рыскания. Он также может измерять линейное ускорение в g-force вдоль осей OX, OY и OZ. В MPU6050 встроен DMP, который обрабатывает сложные алгоритмы движения, снижая вычислительную нагрузку на главный процессор. Этот датчик имеет программируемые диапазоны как для гироскопа, так и для акселерометра. Он может работать при температуре от -40°C до +85°C.

MPU6050 оснащен несколькими выводами, каждый из которых выполняет определенные функции, необходимые для его работы и взаимодействия с другими устройствами [14]. К основным выводам относятся VCC (источник питания, обычно 3,3 В), GND (заземление) и SCL (последовательная тактовая линия для связи I2C). Вывод SDA (последовательной линии передачи данных) также используется для связи I2C, позволяя передавать данные между MPU6050 и микроконтроллером. Вывод AD0 используется для установки адреса I2C, позволяя подключать несколько устройств MPU6050 к одной шине I2C, изменяя ее состояние (высокое или низкое). Вывод INT – это выходной сигнал прерывания, который сигнализирует главному процессору о готовности данных или обнаружении событий движения, обеспечивая эффективную обработку данных. Кроме того, имеются дополнительные выводы I2C (AUX_DA и AUX_CL) для

подключения внешних датчиков, что позволяет MPU6050 расширить свои возможности и собирать больше данных. Эти контакты в совокупности обеспечивают универсальную функциональность MPU6050 в различных приложениях, от простого ведения журнала данных до сложных систем отслеживания движения. На рисунке 14 показан MPU6050 и его контакты.

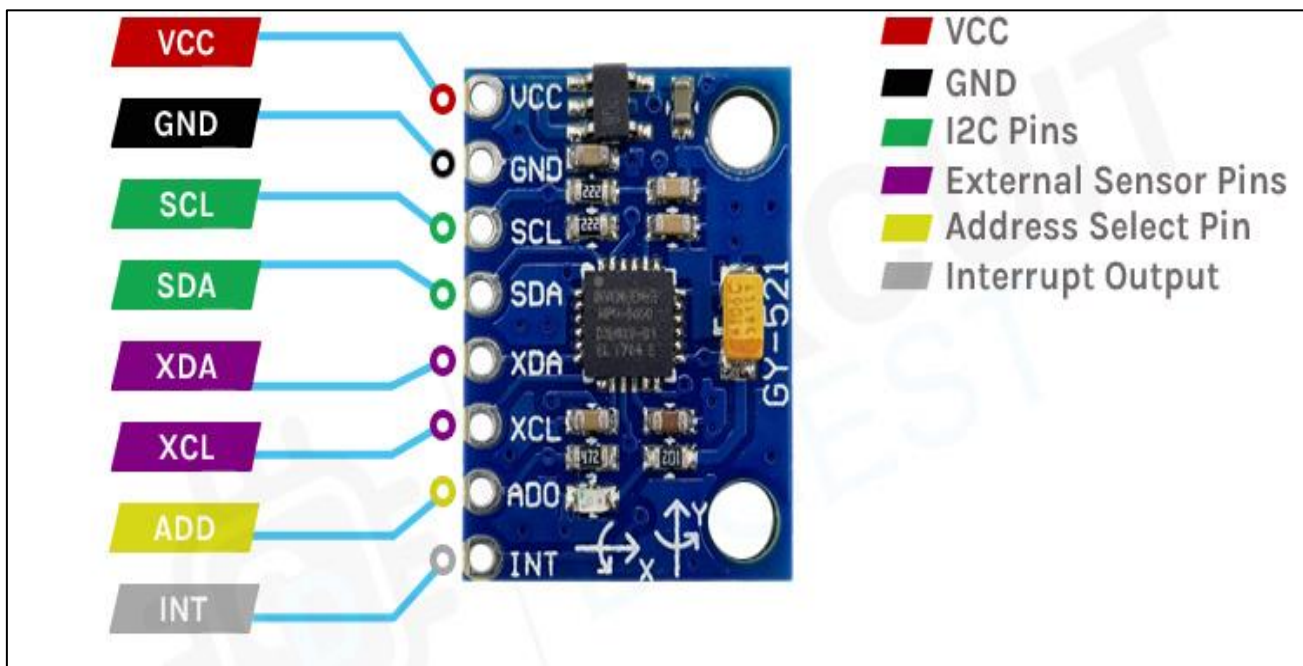


Рисунок 14 – MPU6050 и его контакты

При использовании MPU6050 необходимо учитывать несколько важных факторов для обеспечения оптимальной производительности и точности. Регулярная калибровка необходима, особенно для высокоточных применений, поскольку она компенсирует смещение датчика и обеспечивает надежность измерений. Правильный монтаж также имеет решающее значение; датчик должен быть надежно и правильно установлен, чтобы избежать любых физических помех или вибраций, которые могут исказить показания. Кроме того, использование доступных библиотек программного обеспечения для различных микроконтроллеров и платформ разработки, таких как Arduino, может значительно упростить интеграцию и использование MPU6050. Эти библиотеки предоставляют предварительно написанный код и функции, которые помогают в настройке датчика, считывании данных и обработке алгоритмов движения, тем самым сокращая время и усилия на разработку.

2.1.6 Серводвигатель

Серводвигатели в основном используются в большинстве проектов в качестве исполнительного устройства. Серводвигатели являются неотъемлемыми компонентами в проектах Arduino, которые широко ценятся за их точность управления и универсальность. В отличие от обычных двигателей постоянного тока, серводвигатели предназначены для точного контроля углового положения, что делает их идеальными для приложений, требующих точных перемещений. На рисунке 15 показаны серводвигатели, которые обычно используются в разработке, связанной с Arduino.



Рисунок 15 – Серводвигатель

Серводвигатели обеспечивают исключительную точность управления, обеспечивая точное позиционирование в диапазоне от 0 до 180 градусов. Этот высокий уровень точности достигается благодаря внутреннему механизму обратной связи, который непрерывно регулирует положение двигателя в соответствии с требуемым углом. Внутренний потенциометр и схема управления работают в тандеме, обеспечивая точное управление, что делает серводвигатели незаменимыми в проектах, требующих точного позиционирования. Это особенно важно в таких приложениях, как роботизированные манипуляторы, где для манипулирования объектами необходимы точные движения, антенны, требующие точной ориентации для

оптимального приема сигнала, и системы камер, где точное позиционирование обеспечивает согласованность и стабильность снимков. Способность поддерживать точное управление делает серводвигатели эффективным выбором для широкого спектра применений, требующих высокой точности.

Серводвигатели обеспечивают оптимальный баланс скорости и крутящего момента, что делает их универсальными для широкого спектра применений от легких до среднетяжелых задач. Этот баланс позволяет эффективно использовать их в автоматических системах открывания дверей, механизмах поворота и наклона и небольших роботизированных транспортных средствах, где важно контролируемое движение. Одной из отличительных особенностей серводвигателей является их контур обратной связи, который обеспечивает достижение двигателем желаемого положения и его поддержание даже при изменении условий нагрузки. Эта позиционная обратная связь особенно ценна в динамичных условиях, когда нагрузка на двигатель может колебаться, поскольку позволяет постоянно регулировать его положение для поддержания точности. Эта функция обеспечивает надежную и стабильную работу в различных областях применения, где требуется точное регулирование скорости и крутящего момента.

Серводвигатели известны своей простотой как в распиновке, так и в использовании, что делает их очень доступными для различных проектов. Как правило, серводвигатель имеет три контакта: питание (обычно 5 В), заземление и вывод управляющего сигнала. Такая простая конфигурация обеспечивает легкую интеграцию с микроконтроллерами, такими как Arduino. Использование библиотеки Arduino Servo еще больше упрощает процесс. Библиотека предоставляет интуитивно понятные функции, такие как `attach()` (присоединение), `write()` (запись) и `detach()` (отсоединение), которые позволяют пользователям быстро подсоединить сервопривод к определенному контакту, установить нужный угол наклона и отсоединить его при необходимости. Такая простота использования в сочетании с простой распиновкой позволяет даже новичкам эффективно внедрять точное

управление серводвигателями в свои проекты без сложного кодирования или подключения.

2.2 Выбор средств программной реализации

2.2.1 Разработка функциональной схемы

Функциональная схема системы аварийной посадки летательного аппарата представлена на рисунке 15.



Рисунок 15 – Функциональная схема системы управления аварийной посадки ЛА

Сначала необходимо разобраться в том, что такое функциональная схема. Функциональная схема – это набор прямоугольников и стрелок, которые указывают на компонент и способ передачи данных по всей системе. Она дает четкое представление о том, как различные части системы взаимодействуют друг с другом для создания функционирующей системы. Схемы такого типа часто используются в инженерии, разработке программного обеспечения и системном проектировании для иллюстрации и передачи потока информации, управления и процессов. Функциональная схема имеет свое назначение и область применения, что будет очень удобно и полезно. Она действительно помогает при проектировании и разработке работы, поскольку эффективна при планировании и структурировании системы, разбивая ее на управляемые части и определяя их взаимодействие.

Для обычного пользователя функциональная схема обеспечивает четкое визуальное представление, которое можно легко понять как средство коммуникации. Если в системе есть какие-либо недостатки, функциональную схему можно легко проанализировать и устранить. И последнее, но не менее важное: функциональная схема является ключевым моментом, используемым в документации. Это гарантирует, что функциональные аспекты четко записаны для дальнейшего использования и сопровождения.

Таким образом, функциональная схема является ценным инструментом для понимания, проектирования и передачи информации о функциональных аспектах системы, гарантируя, что все части работают слаженно для достижения целей системы. Такое всестороннее понимание необходимо для успешной разработки системы, эффективного устранения неполадок и эффективной коммуникации между пользователями.

На рисунке 15 выше функциональная схема может быть разделена на 4 части. Первая часть – это те, что расположены слева. Эта часть в основном содержит датчики, которые собирают данные для обработки. В этой работе используются ровно четыре датчика: датчик гироскопа, ультразвуковой датчик, передатчик и приемник. Передатчик и приемник одной модели, но запрограммированы на выполнение разных задач. Все данные собираются в режиме реального времени и непрерывно передаются на микроконтроллер.

В средней части функциональной схемы находятся микроконтроллер и источник питания, которые будут рассмотрены отдельно. Arduino UNO может питаться от нескольких источников, что обеспечивает гибкость для различных проектных нужд. Он может питаться через USB-соединение с компьютера или внешний USB-адаптер питания, что удобно для программирования и простых приложений. Кроме того, Arduino UNO оснащен гнездом для подключения внешнего источника питания, обеспечивающим входное напряжение в диапазоне от 7 до 12 вольт. Это особенно полезно для автономных проектов, требующих более надежного питания. Кроме того, имеется вывод VIN, который можно использовать для подачи нерегулируемого напряжения

непосредственно на плату, что является еще одной альтернативой для подачи питания. Плата оснащена встроенным регулятором напряжения, обеспечивающим стабильную работу независимо от источника питания. Такая универсальность в выборе источников питания позволяет Arduino UNO адаптироваться к широкому спектру применений, от настольных экспериментов до мобильных и встраиваемых систем.

С другой стороны, микроконтроллер используется для обработки данных, и его можно назвать "мозгом" этой работы. Использование микроконтроллерных плат Arduino неуклонно развивалось на протяжении многих лет[16]. Стрелка в левой части направлена в сторону микроконтроллера, чтобы показать, что данные, собранные датчиками, отправляются на микроконтроллер. Затем Arduino UNO отправляет данные, которые обрабатываются для выполнения определенной задачи, в правую часть нашей функциональной схемы.

В правой части функциональной схемы расположены исполнительные устройства. Два серводвигателя используются для отслеживания движения самолета, приближающегося к платформе. Они использовали данные, которые обрабатываются микроконтроллером и программируются определенным образом, подходящим для отслеживания приближающегося самолета. Другим устройством, получающим информацию от микроконтроллера, является последовательный монитор, который отображает результаты, полученные от микроконтроллера, которые измеряются датчиками. Процессы связаны, и они работают вместе, создавая сложную функционирующую систему.

2.2.2 Выбор языка программирования

Выбор языка программирования для платформ Arduino зависит от сложности проекта, знания языка программистом и конкретных требований, таких как производительность, простота использования и доступные библиотеки. Существует несколько языков программирования и сред, которые могут использоваться в зависимости от требований проекта и предпочтений

пользователя. В таблице 5 для наглядности показано сравнение трех наиболее часто используемых языков программирования для платформ Arduino.

Таблица 5 – Сравнение языков программирования для платформ Arduino.

Языки	Arduino C/C++	C/C++	Python
Обзор языки	Языком программирования Arduino по умолчанию является упрощенная версия C++ с набором функций и библиотек, предоставляемых ядром Arduino.	C и C++ – это более продвинутые варианты, которые обеспечивают больший контроль и эффективность при выполнении сложных проектов.	MicroPython – это экономичная реализация Python 3, работающая на микроконтроллерах.
Особенности	<p>— Простота использования: Разработан таким образом, чтобы быть удобным для начинающих, с простым синтаксисом и обширной документацией.</p> <p>— Библиотеки: Обширная коллекция библиотек для различных датчиков,</p>	<p>— Производительность: Обеспечивает высокую производительность и низкоуровневый доступ к оборудованию.</p> <p>— Гибкость: Позволяет использовать передовые методы программирования и напрямую управлять регистрами микроконтроллера.</p> <p>— Цепочки инструментов: Могут использоваться с</p>	<p>— Простота изучения: простой синтаксис Python делает его доступным для начинающих.</p> <p>— Интерактивность: Поддерживает интерактивную разработку с помощью интерфейса REPL (цикл чтения-оценки-печати).</p> <p>— Библиотеки: Растет число библиотек, адаптированных для</p>

	модулей и аппаратных компонентов. — Поддержка IDE: Полностью поддерживается интегрированной средой разработки Arduino (IDE), которая включает в себя редактор кода, компилятор и загрузчик.	различными средами разработки, такими как Atmel Studio, Eclipse с подключаемым модулем Arduino или инструментами командной строки.	работы с микроконтроллерами.
--	--	--	------------------------------

Для разрабатываемой работы используется языка Arduino C/C++, благодаря его сочетанию простоты и мощности. Доступность языка программирования Arduino C/C++ является непревзойденной. Язык и прилагаемая интегрированная среда разработки (IDE) разработаны таким образом, чтобы быть удобными в использовании, особенно для начинающих [17]. Синтаксис Arduino C/C++ прост, с четкой и лаконичной структурой, которая позволяет новым пользователям начинать программировать без сложного обучения. Arduino IDE упрощает процесс написания, компиляции и загрузки кода в микроконтроллер, позволяя пользователям сразу же увидеть результаты своих усилий по программированию [4].

Одним из главных преимуществ использования Arduino C/C++ является быстрый набор доступных библиотек. Эти библиотеки охватывают широкий спектр функциональных возможностей, от управления двигателями и датчиками до работы с протоколами связи и пользовательскими интерфейсами. Использование этих библиотек может значительно сэкономить время и усилия, так как отпадет необходимость писать сложный код с нуля.

Последовательное использование Arduino C/C++ в этих библиотеках обеспечивает совместимость и простоту интеграции, что еще больше укрепляет его позиции в качестве языка программирования.

Arduino C/C++ является родным языком для платформы Arduino, обеспечивая полную совместимость со всем официальным оборудованием и библиотеками Arduino. Эта бесшовная интеграция гарантирует, что код, написанный на Arduino C/C++, будет надежно работать на различных платах и экранах Arduino без необходимости в значительных модификациях. Кроме того, многие сторонние библиотеки и инструменты разработаны специально для работы с Arduino C/C++, что улучшает экосистему и расширяет возможности для инновационных работ. На рисунке 16 показан менеджер библиотек, который можно найти в Arduino IDE, который постоянно обновляется при поддержке большого сообщества, которое всегда активно.

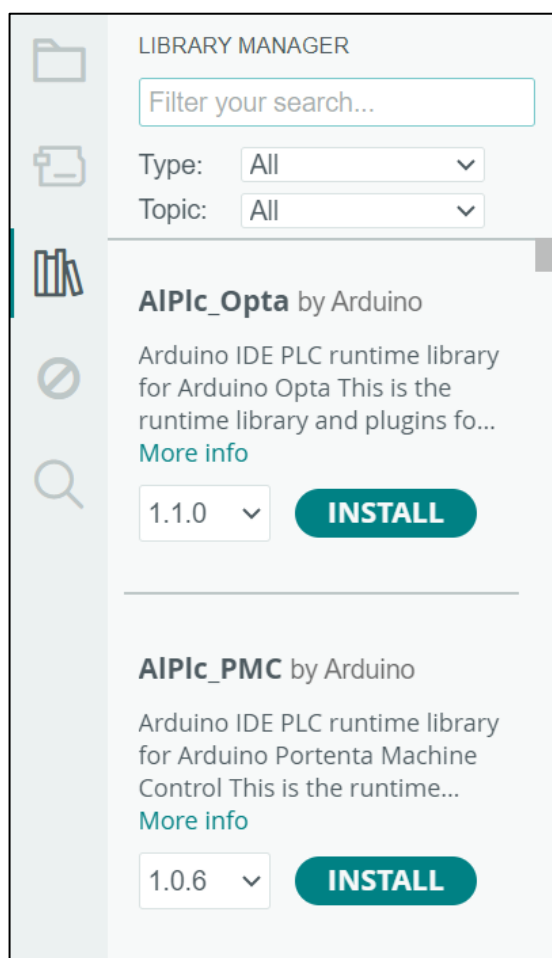


Рисунок 16 – Менеджер библиотек Arduino IDE

В заключение, выбор Arduino C/C++ для программирования проектов Arduino дает множество преимуществ, включая простоту, обширную библиотечную поддержку, универсальность, сильную поддержку сообщества и полную совместимость с оборудованием Arduino. Эти преимущества делают Arduino C/C++ идеальным выбором как для начинающих, так и для опытных разработчиков, стремящихся создавать эффективные, надежные и высокопроизводительные проекты Arduino. Интерфейс редактор кода Arduino IDE представлена на рисунке 17.

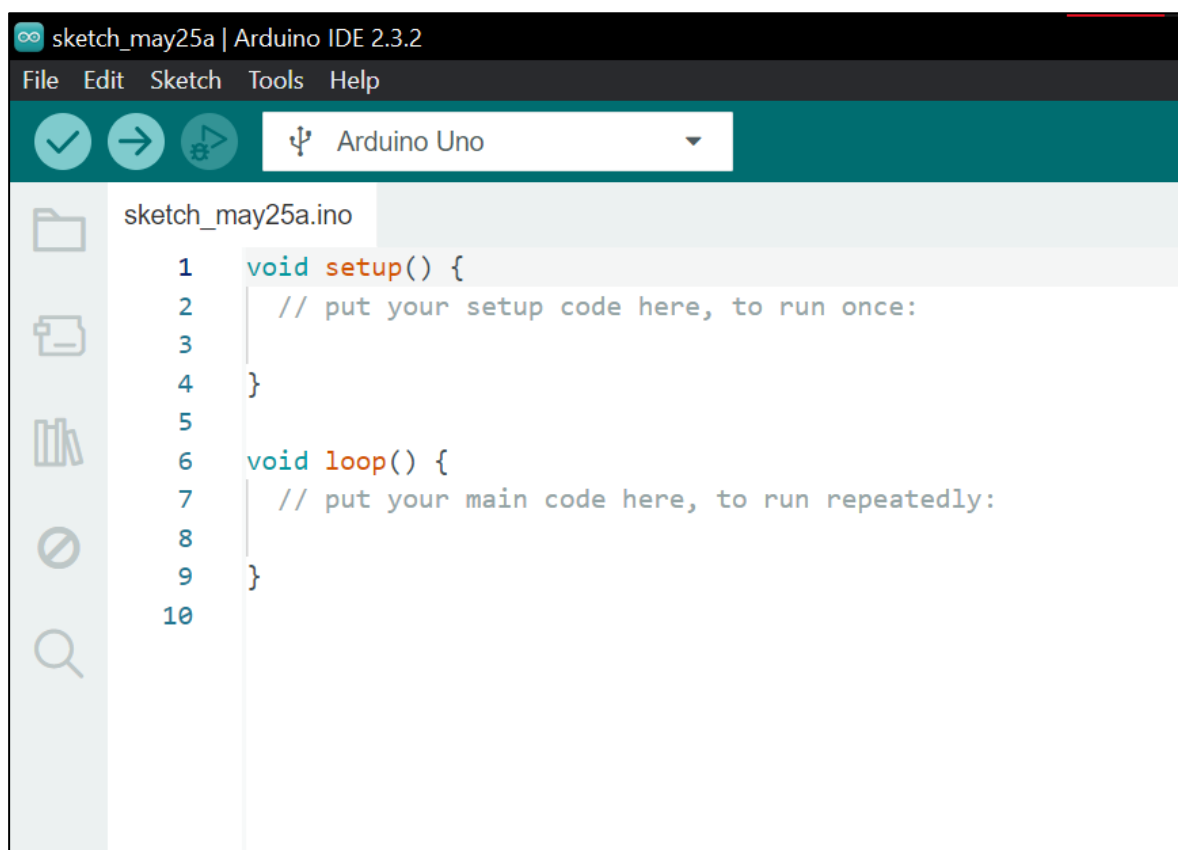


Рисунок 17 – Интерфейс редактор кода Arduino IDE

2.3 Разработка алгоритма функционирования системы управления аварийной посадки летательного аппарата

Алгоритм – это четко определенная пошаговая процедура или набор правил, предназначенных для выполнения конкретной задачи или решения конкретной проблемы [18]. Алгоритмы являются фундаментальными для информатики и математики, но они также применимы в широком спектре областей, от повседневной деятельности до сложных научных вычислений.

Алгоритм обладает несколькими ключевыми характеристиками, которые делают его действительно полезным для понимания того, как работает определенная система. Одна из них – это четко определенные инструкции. Алгоритм состоит из последовательности четких и недвусмысленных шагов. Каждый шаг должен быть точно определен, чтобы не было неопределенности в отношении того, что необходимо сделать. Алгоритм также должен иметь конечное число шагов. Это означает, что в конечном итоге он должен завершиться после выполнения серии операций. Следующей ключевой характеристикой алгоритма является то, что он всегда будет выдавать один или несколько выходных данных, которые являются результатами обработки, выполненной на входных данных. Последняя ключевая характеристика алгоритма заключается в том, что он всегда должен быть способен выдавать один и тот же результат для одних и тех же входных данных при каждом его выполнении. Все эти характеристики делают алгоритм своего рода командной строкой, но в базовой форме, которая может быть выполнена простым человеком с помощью карандаша и бумаги.

На рисунке 18 представлена алгоритма функционирования системы управления аварийной посадки летательного аппарата.

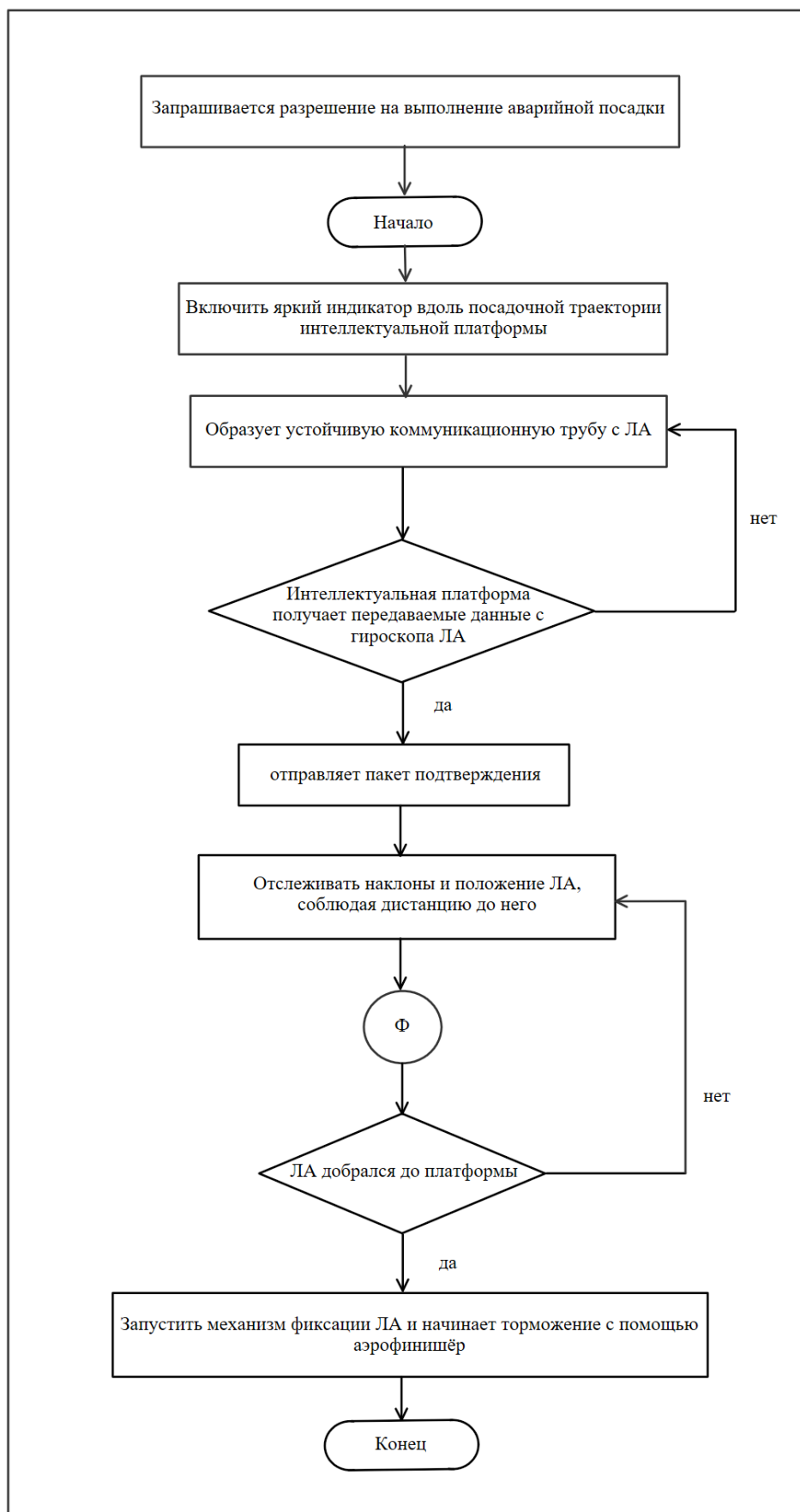


Рисунок 18 – Алгоритма функционирования системы управления аварийной посадки летательного аппарата

Блок-схема алгоритма обеспечивает наглядное представление о том, как работает система с самого начала и до самого конца процесса. Она служит важным инструментом для понимания последовательных шагов и взаимодействий внутри системы. Разбивка процесса на управляемые разделы позволяет получить четкое представление о функциональности и структуре системы. Все датчики и исполнительные механизмы, которые играют важную роль в системе, включены в процесс, как показано на рисунке 18. Такое включение обеспечивает учет функций и интеграции каждого компонента, подчеркивая их роли и взаимодействие. Блок-схема алгоритма не только помогает определить поток данных и управляющих сигналов, но и помогает в устранении неполадок и оптимизации системы. Визуализируя весь процесс, можно лучше понять вклад каждого отдельного компонента в общую работу. Такое всестороннее представление важно как для проектирования, так и для анализа сложных систем.

Процесс выполнения аварийной посадки с помощью интеллектуальной посадочной платформы включает в себя ряд скоординированных действий, направленных на обеспечение безопасности как летательного аппарата, так и его пассажиров. Эта усовершенствованная система начинает свою работу только после получения официального запроса и одобрения диспетчерами воздушного движения или аэродрома, гарантируя, что аварийные процедуры будут запущены только в случае крайней необходимости.

После активации интеллектуальная посадочная платформа подсвечивается яркими индикаторами, которые служат важными визуальными ориентирами для пилотов, ориентирующихся в аварийной ситуации. Эти индикаторы помогают пилотам определить местоположение и ориентацию платформы, особенно в неблагоприятных условиях, таких как плохая видимость или ночные операции. Одновременно платформа устанавливает устойчивый канал связи с приближающимся летательным аппаратом, что имеет решающее значение для бесперебойного обмена данными и координации на протяжении всего процесса посадки.

Канал связи тщательно проверяется для обеспечения точности данных. Если первоначальное соединение было плохим, система пытается установить новый, более надежный канал. Как только стабильное соединение подтверждено, платформа отправляет пакеты подтверждения на борт воздушного судна, обеспечивая синхронизацию обеих систем. Используя сигналы, поступающие от гироскопа летательного аппарата, интеллектуальная посадочная платформа непрерывно отслеживает наклон и положение летательного аппарата. Это отслеживание жизненно важно для обеспечения максимальной поверхности контакта при посадке, что сводит к минимуму риск столкновения и обеспечивает плавное приземление.

Измеренные данные соответствующим образом фильтруются, что обозначается как "Ф" в алгоритме функционирования системы управления аварийной посадки. Алгоритма фильтрации данных представлена на рисунке 19.

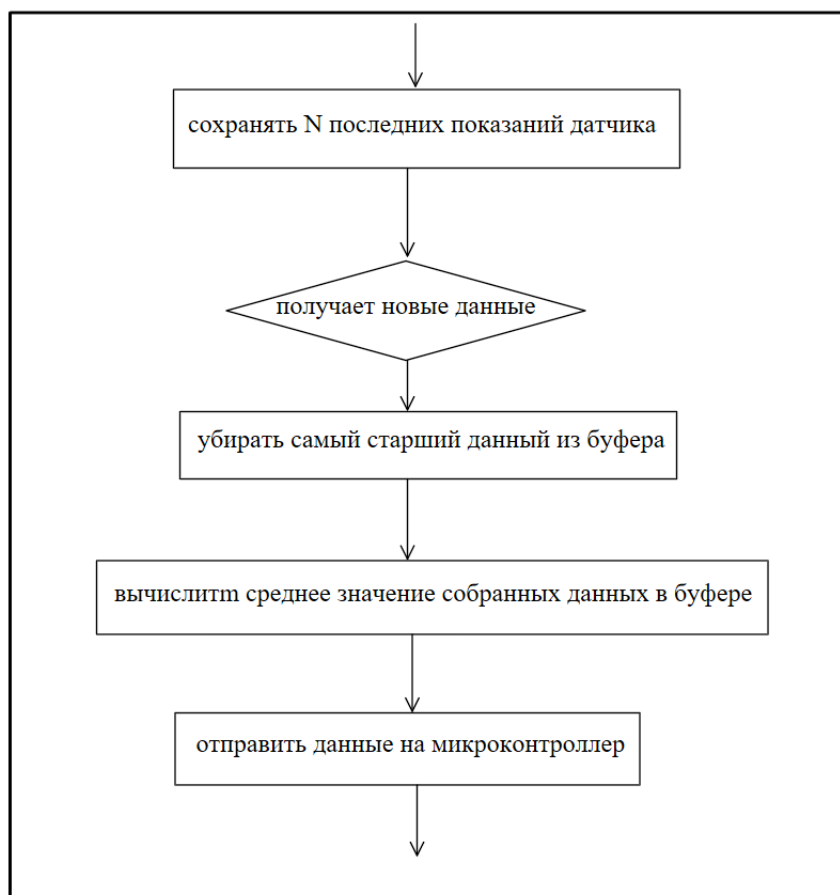


Рисунок 19 – Алгоритма фильтрации данных

В алгоритме фильтрации данных используется метод бегущего среднего арифметического. Этот метод вычисляет среднее значение из последних N измерений, из-за чего значение задерживается [11]. В данной работе значение N равно 5, что считается наилучшим значением, поскольку скорость вычисления также является одной из важных переменных. При каждом добавлении нового измерения добавляются пять последних значений и усредняются для получения точного результата измерения. Это также делается для того, чтобы отфильтровать шумы, которые могут возникнуть из-за несовершенства датчика в реальных условиях.

Одновременно ведется тщательное наблюдение за расстоянием между летательным аппаратом и платформой. Если летательный аппарат еще не достиг платформы, система продолжает отслеживать его местоположение и контролировать расстояние. По прибытии на платформу активируется сложный механизм фиксации. Этот механизм закрепляет самолет на платформе, гарантируя, что он останется целым и выровненным по взлетно-посадочной полосе. Система фиксации предназначена для поглощения инерции самолета, предотвращая его перелет или отклонение от курса.

Как только самолет надежно закреплен на платформе, начинается процесс торможения. Этот этап тщательно контролируется для обеспечения того, чтобы скорость торможения оставалась в пределах, безопасных как для пассажиров, так и для целостности конструкции воздушного судна. Торможению также способствуют стопорные устройства, установленные на платформе вдоль взлетно-посадочной полосы, что обеспечивает дополнительную тормозную способность и снижает нагрузку на собственные тормозные системы самолета.

Наконец, как только самолет полностью останавливается, интеллектуальная посадочная платформа завершает свою работу. Тщательная координация этих действий гарантирует, что аварийная посадка может быть выполнена с максимальной безопасностью и эффективностью, используя

передовые технологии для снижения рисков, присущих таким критическим ситуациям.

2.4 Моделирование системы управления аварийной посадки летательного аппарата

Разработанная схема подключения периферийных устройств системы управления аварийной посадки летательного аппарата представлена на рисунке 20 (выполнено в среде Fritzing).

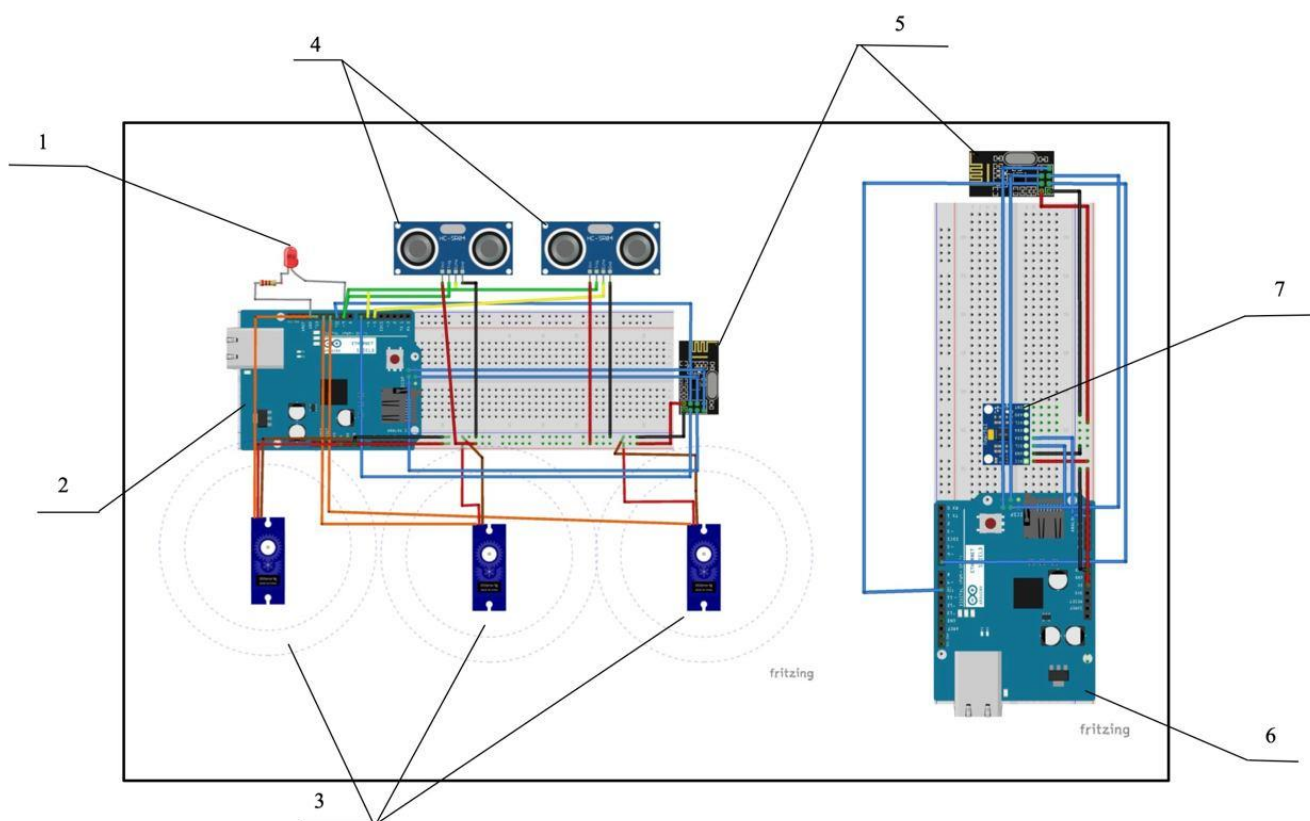


Рисунок 20 – Схема подключения периферийных устройств системы управления аварийной посадки летательного аппарата:

1 – светодиод; 2 – плата Arduino UNO; 3 – сервомоторы; 4 – датчики ультразвуковые; 5 – радиомодули; 6 – плата Arduino UNO; 7 – датчик гироскопа

На приведенной выше схеме система управления аварийной посадки летательного аппарата состоит из двух микроконтроллеров Arduino UNO. Все датчики и исполнительные устройства подключены так, как показано на

схеме. Arduino слева служит в качестве микроконтроллера для интеллектуальной посадочной платформы, в то время как Arduino справа отвечает за систему самолета. Связь между двумя Arduino осуществляется посредством радиосвязи с использованием модулей NRF24L01+. Радиомодуль, подключенный к гироскопической системе самолета, действует как передатчик, а модуль, подключенный к интеллектуальной посадочной платформе, функционирует как приемник. Этот приемник обрабатывает данные, поступающие от датчика гироскопа самолета, для обеспечения точного позиционирования и центровки. Коммуникационный адрес для радиомодулей указывается в начале процесса кодирования для установления надежного соединения между двумя системами.

Если посмотреть на кодировку, которую можно найти в разделе “Приложение А”, то основное кодирование начинается с включения всех библиотек, которые будут использоваться со всеми выбранными датчиками. Библиотека <SPI.h> используется для работы с шинами SPI, в то время как библиотеки <nRF24L01.h> и <RF24.h> используются для работы с радиомодулем, который соединяет систему самолета и систему платформы. Разница между этими двумя системами заключается в том, что в самолетной системе будут библиотеки <MPU6050_light.h> и <Wire.h>, которые помогают работать с датчиком гироскопа, в то время как в системе smart landing platform библиотека <Servo.h> используется для работы с исполнительными устройствами, которые являются основой системы серводвигателя.

Затем, в ходе процедуры Setup() системы самолета, запускается монитор последовательного порта для наблюдения за показаниями датчиков. Канал радиосвязи, скорость передачи данных и уровень амплитуды настраиваются таким образом, чтобы обеспечить наилучшую бесперебойность связи между обеими системами. С другой стороны, Setup() системы интеллектуальной посадочной платформы также началась с запуска монитора последовательного порта и точной настройки радиомодуля. Скорость передачи данных и амплитуда радиочастот установлены на максимальный уровень, чтобы

обеспечить надежную и быструю связь. Далее серводвигатели и ультразвуковые датчики помечаются в соответствии с их контактами, которые они подключают к плате Arduino UNO. Начальный угол для всех серводвигателей устанавливается посередине, в данном случае 90°. Это объясняется тем, что выбранные серводвигатели могут вращаться точно в диапазоне от 0° до 180°.

В разделе loop() в виде кода описан точно такой же алгоритм, как и тот, что показан на рисунке 20 выше. В конечном счете, эта процедура стала проще благодаря использованию библиотек, упомянутых в начале абзаца. Системы самолета и система интеллектуальной посадочной платформы работают одновременно в режиме реального времени, что можно четко наблюдать на последовательном мониторе.

2.5 Выводы ко второму разделу

В текущей главе, "Выбор аппаратных и программных средств для системной рассылки", представлен подробный обзор аппаратных и программных компонентов, необходимых для этой ВКР. В этой главе подробно сравниваются различные варианты, чтобы определить наиболее подходящие компоненты для системы. После оценки нескольких микроконтроллеров был выбран Arduino UNO за оптимальный баланс функций, простоту использования и соответствие требованиям проекта.

В главе также приводится подробное сравнение различных датчиков, позволяющее убедиться в том, что выбранные компоненты соответствуют необходимым стандартам производительности и надежности. Приводится подробная функциональная схема, иллюстрирующая, как выбранные датчики и исполнительные устройства интегрируются с микроконтроллером Arduino UNO. Эта схема не только демонстрирует физические соединения, но и объясняет поток данных по всей системе, подчеркивая взаимодействие между аппаратными и программными компонентами. Кроме того, в главе подробно описывается алгоритм работы системы, дающий четкое представление о

процессе от начала до завершения. Этот алгоритм описывает каждый шаг, который система предпринимает во время аварийной посадки, обеспечивая ясность и точность его выполнения. Кроме того, в комплект входит алгоритм фильтрации сигналов, демонстрирующий, как программные процессы фильтрации реализуются в режиме реального времени для поддержания точности и надежности системы в процессе эксплуатации.

В целом, в этой главе представлен всесторонний обзор аппаратной и программной реализации, гарантирующий тщательное планирование и выполнение каждого аспекта системы управления аварийной посадкой. Это служит важной основой для понимания того, как работает система, и подчеркивает тщательное рассмотрение каждого компонента для достижения целей проекта.

3 РЕАЛИЗАЦИИ И АНАЛИЗ РЕЗУЛЬТАТОВ СИСТЕМЫ УПРАВЛЕНИЯ АВАРИЙНОЙ ПОСАДКИ ЛЕТАТЕЛЬНОГО АППАРАТА

3.1 Структурная схема системы управления аварийной посадки летательного аппарата

Тестирование разработанной системы управления аварийной посадки летательного аппарата можно разбить на несколько этапов. Первым шагом, который необходимо предпринять, является оценка ситуации, в которой находится самолет. Для контроля параметров, измеряемых датчиками, интеллектуальная посадочная платформа будет регулировать полученные или измеряемые значения. Весь процесс очень важен для обеспечения стабильности системы. Структурная схема системы управления аварийной посадки самолета представлена на рисунке 21.

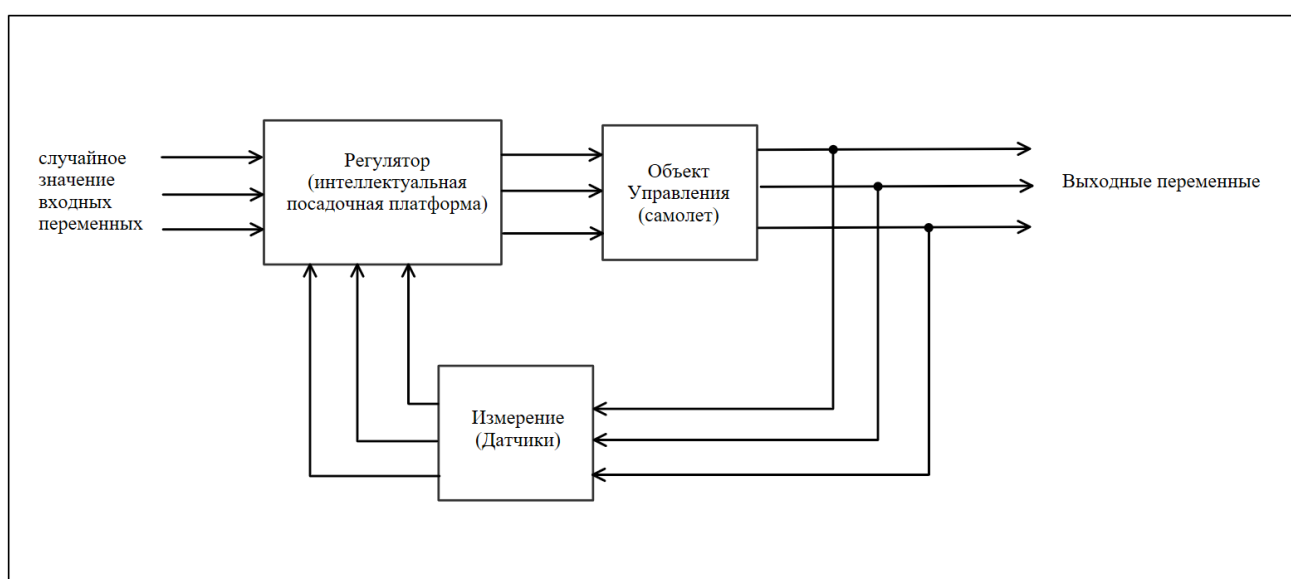


Рисунок 21 – Структурная схема системы управления аварийной посадки
летательного аппарата

3.2 Экспериментальная установка и проведение эксперимента

В соответствии с моделированием, которое выполнено в программе Fritzing, показанной на рисунке 20 выше, была собрана экспериментальная установка для тестирования в реальных условиях как показано на рисунке 22.

Все результаты тестирования были записаны соответствующим образом для каждой из систем.

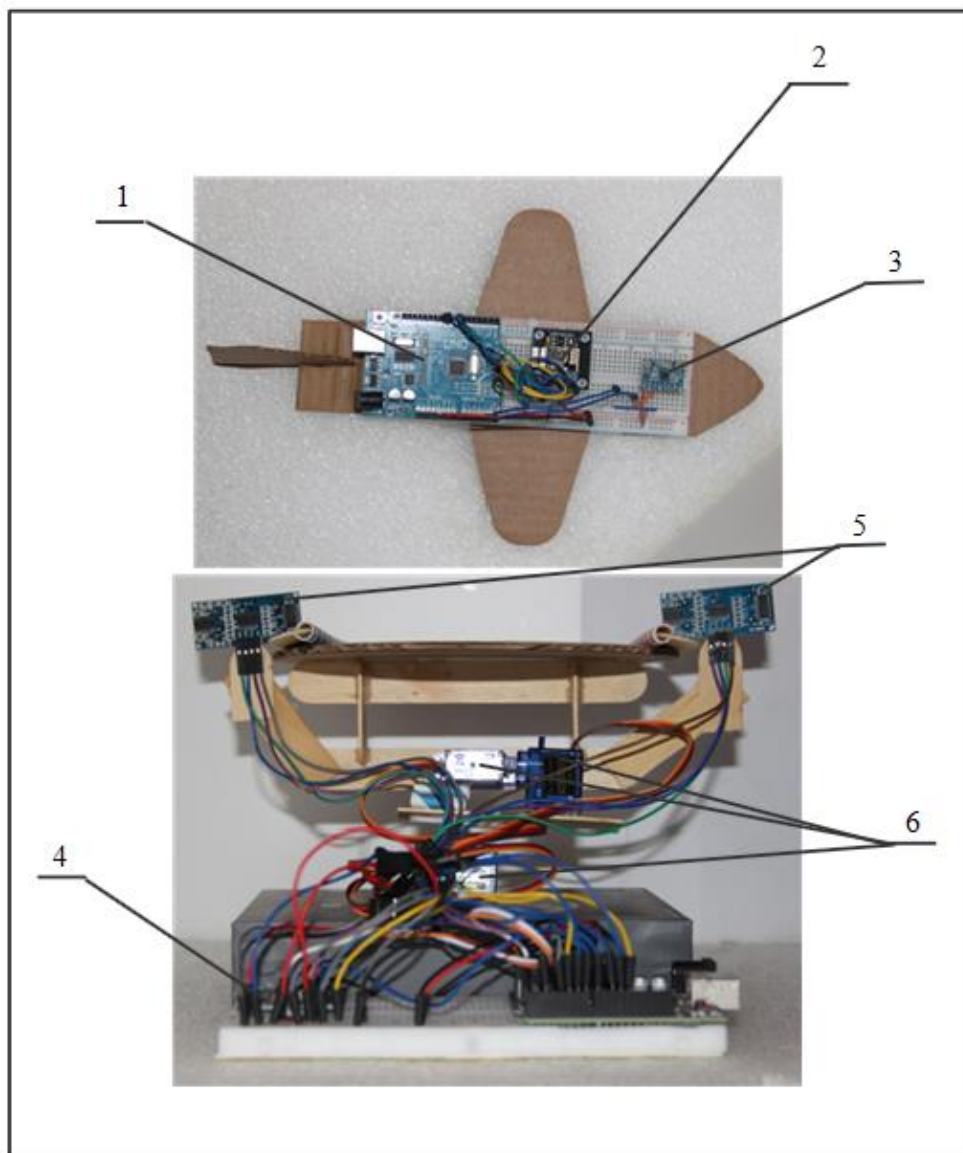


Рисунок 22 – Экспериментальная установка системы управления аварийной посадки летательного аппарата:

1 – плата Arduino UNO; 2 – радиомодуль; 3 – датчик гироскопа; 4 – радиомодуль; 5 – датчики ультразвуковые; 6 – сервомоторы

Чтобы проверить работоспособность системы, все компоненты выстраиваются в соответствии с ее положением, в котором она должна работать. Из-за мало бюджета и поскольку эта работа будет сосредоточена на системе, а не на качестве сборки, материал, из которого изготовлена эта система, не самый лучший, и его можно было бы усовершенствовать, если бы

для проектирования каждой части модели использовался 3D-принтер. Несмотря на все это, система функционирует бесперебойно, и есть несколько основных систем, которые можно наблюдать самостоятельно. Несколько систем, объединенных вместе, могут образовать одну большую сложную систему [19].

3.3 Испытание механизма фиксации

Механизм фиксации является одной из основных систем, которая гарантирует, что при посадке самолет и интеллектуальная посадочная платформа не столкнутся друг с другом. Эта система работает с помощью ультразвукового датчика, который определяет расстояние до самолета. Двигатель, прикрепленный к механизму фиксации, будет мгновенно приведен в действие, как только самолет приблизится на достаточное расстояние к платформе.

Фиксирующий механизм оснащен амортизатором, который гарантирует, что самолет не будет поврежден во время процесса, который происходит в мгновение ока. Давление на захват самолета оптимально для того, чтобы самолет не соскользнул с платформы. Тестирование механизма отсечения происходит так, как показано на рисунке 23.

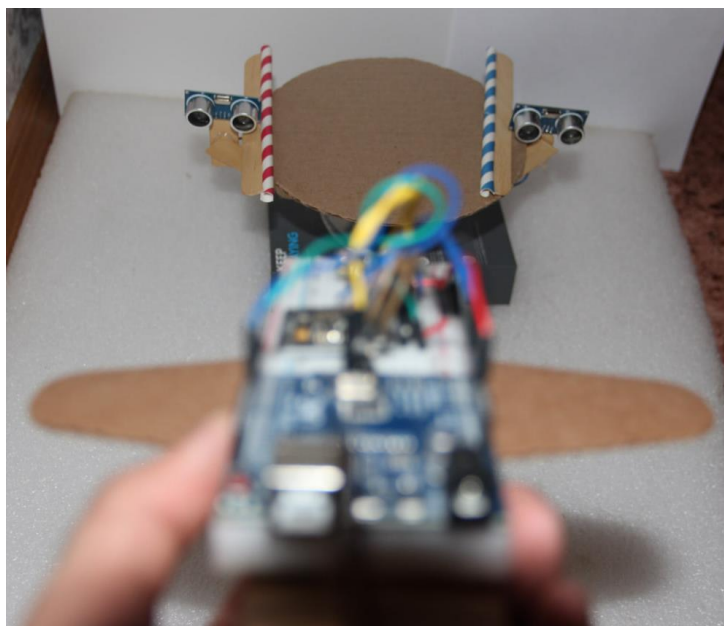


Рисунок 23 – Самолет находится в расстоянии далеко от интеллектуальной платформы

Пока расстояние между самолетом и интеллектуальной посадочной платформой не достигнет 8 сантиметров, что в данном случае соответствует радиусу модели платформы, механизм фиксации будет оставаться в открытом положении. В это время платформа активно отслеживает движение самолета и его наклон. На рисунке 24 представлена механизма фиксации после того, как самолет достигнет платформы.

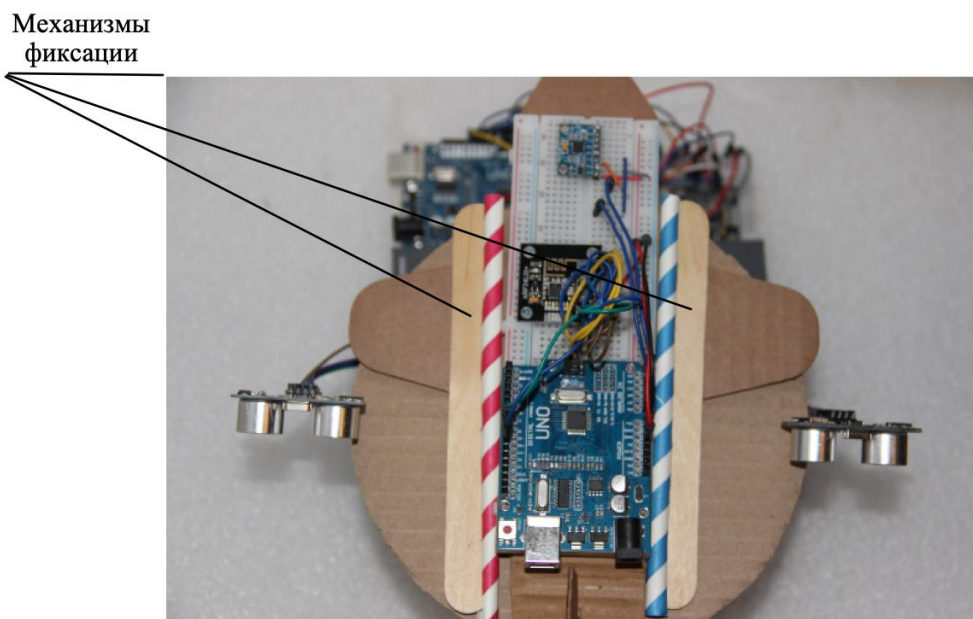


Рисунок 24 – Самолет достигает платформы и запускает механизм фиксации

Как только модель самолета оказывается на близком расстоянии от платформы, срабатывает фиксирующий механизм, удерживающий ее на месте, и платформа медленно меняет свое положение и угол наклона, чтобы быть параллельной земле, одновременно замедляясь с помощью тормозного механизма, который представляет собой аэрофинишёр, прикрепленное к интеллектуальной посадочной платформе.

3.4 Испытание системы слежения за положением самолета

Поскольку платформа ожидает, что самолет приблизится к ней с любого направления перед ней под углом 60 градусов справа или слева, платформа должна будет отслеживать положение и курс летательного аппарата, чтобы

гарантировать, что, когда самолет достигнет платформы, он подлетит под правильным углом, чтобы платформа могла обеспечить наилучшие условия для посадки. В этой работе ультразвуковой датчик используется для отслеживания положения самолета. Есть два ультразвуковых датчика, которые установлены в правой и левой частях платформы. Ультразвуковой датчик начинает активно отслеживать самолет, когда тот оказывается на расстоянии 100 сантиметров от платформы. Вращение платформы при отслеживании положения самолета происходит вдоль оси Y . Данная ситуация представлена на рисунке 25.

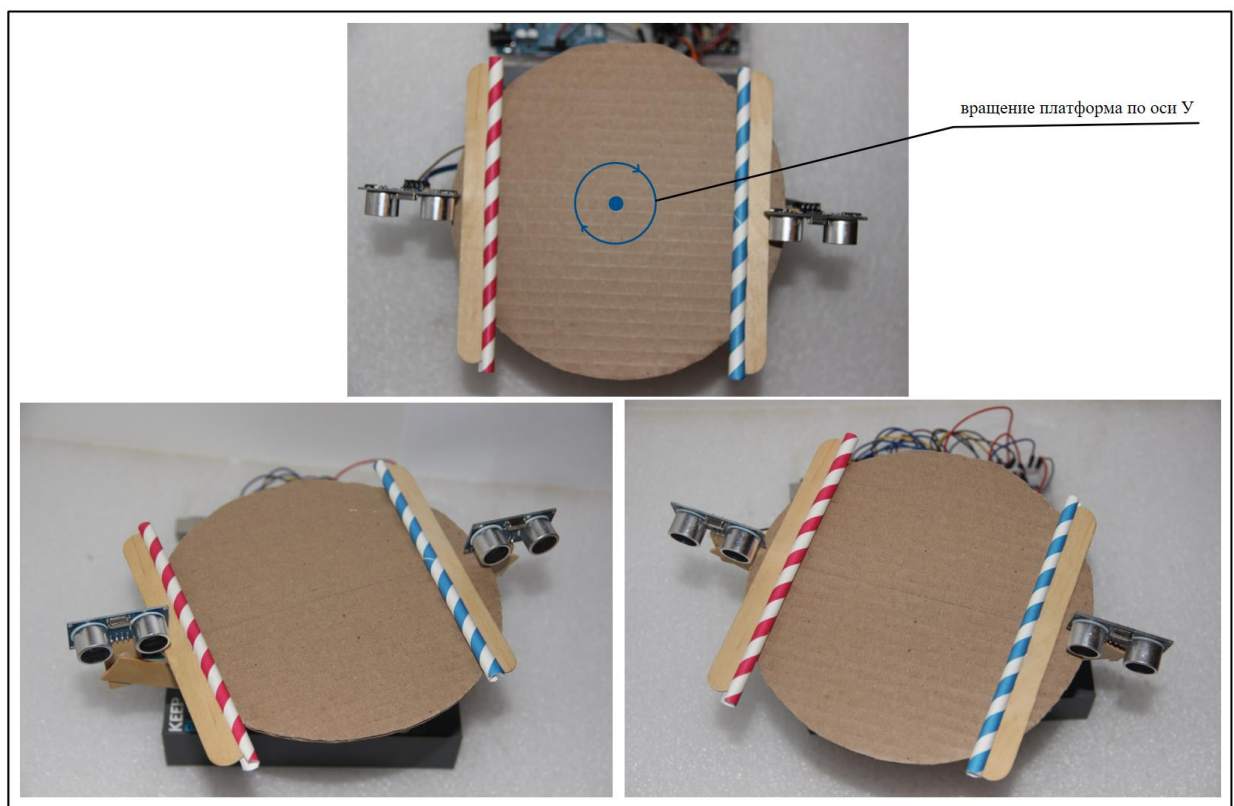


Рисунок 25 – Системы слежения за положением самолета

Если контролируемый объект, которым в данном случае является самолет, обнаружен ближе к правой части ультразвукового датчика, платформа будет отслеживать его местоположение до тех пор, пока расстояние от обоих датчиков не станет одинаковым. То же самое происходит, когда самолет находится близко к левому ультразвуковому датчику. Расстояние самолета от левого и правого ультразвуковых датчиков показано на рисунке 26.

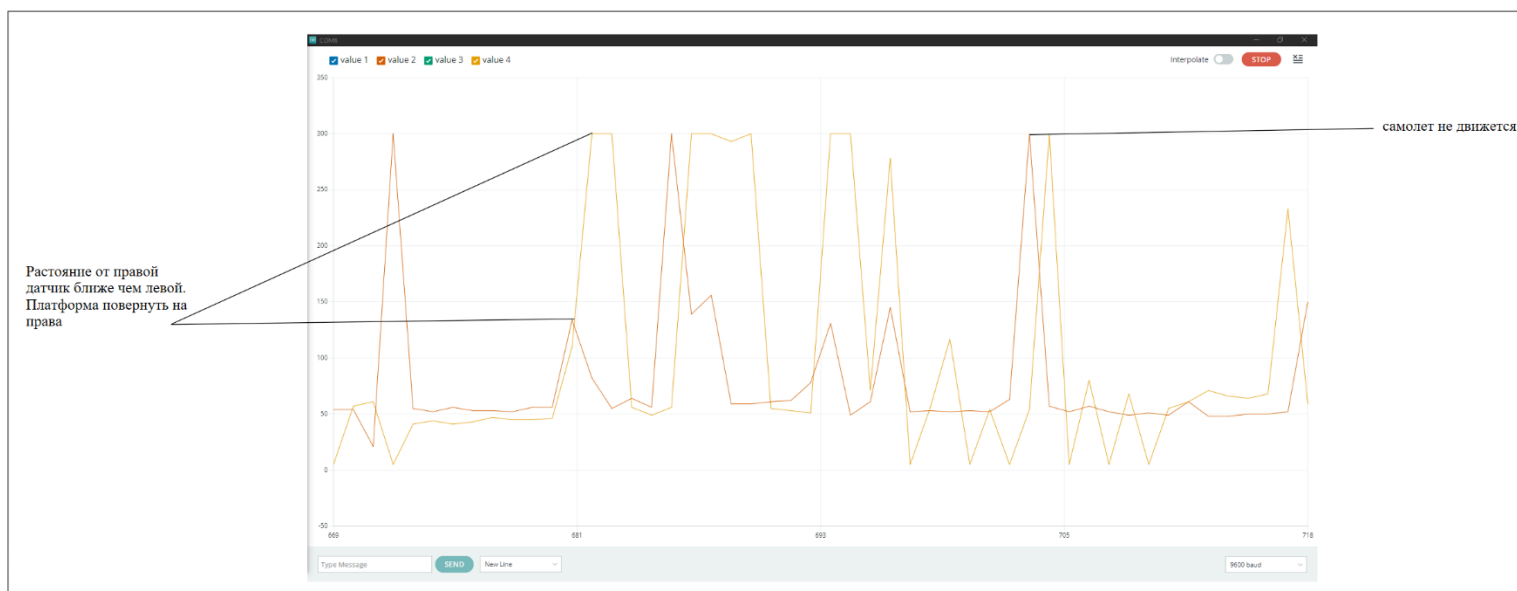


Рисунок 26 – Показание ультразвукового датчика

Красная линия обозначает правый ультразвуковой датчик, а желтая – левый. Когда оба показания находятся на одной высоте, это означает, что платформа остается неподвижной, но если одна из сторон находится ближе к самолету, исполнительный механизм (серводвигатель) поворачивает платформу в сторону самолета.

3.5 Испытание системы слежения за наклоном самолета

Случайный наклон самолета – одна из самых значительных проблем при аварийной посадке. Самолет в этом тесте наклоняется по-разному, чтобы обеспечить реакцию интеллектуальной посадочной платформы. Модель самолета оснащена гироскопом, который обеспечивает правильное измерение его наклона. Чтобы отправить данные быстро и без задержек, управляемый объект (модель самолета) взаимодействует с интеллектуальной посадочной платформой посредством радиосвязи. Это позволяет платформе поворачиваться под соответствующим углом, под которым находится самолет, обеспечивая широкую посадочную площадку и гарантируя, что механизм фиксации не прерывается из-за наклона самолета.

Первый учитываемый угол, показанный на рисунке 27, представляет собой наклон самолета вдоль оси z . Он рассматривается как угол крена самолета. Платформа начинается под углом, параллельным земле. Как только

калибровка гироскопа управляемого объекта (модели самолета) будет завершена, платформа наклонится в соответствии с текущим углом наклона самолета.

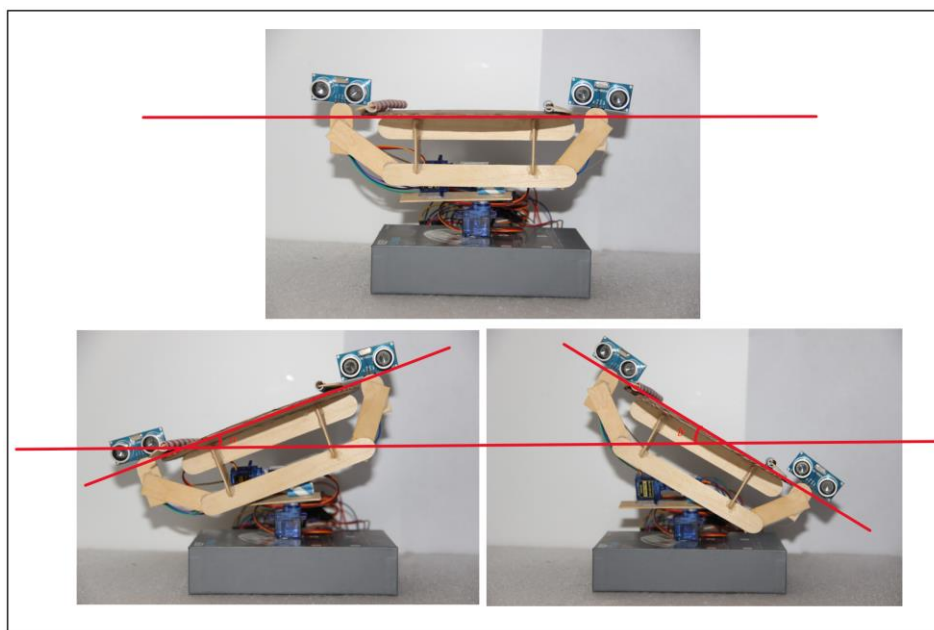


Рисунок 27 – Системы слежения за наклон крена самолета

То же самое относится и к платформе для отслеживания наклона модели самолета вдоль оси x . Это рассматривается как отслеживание угла тангажа самолета. Если посмотреть на красную линию, то ясно видно, что угол меняется вдоль оси x . Наклон платформы вдоль оси x показан на рисунке 28.

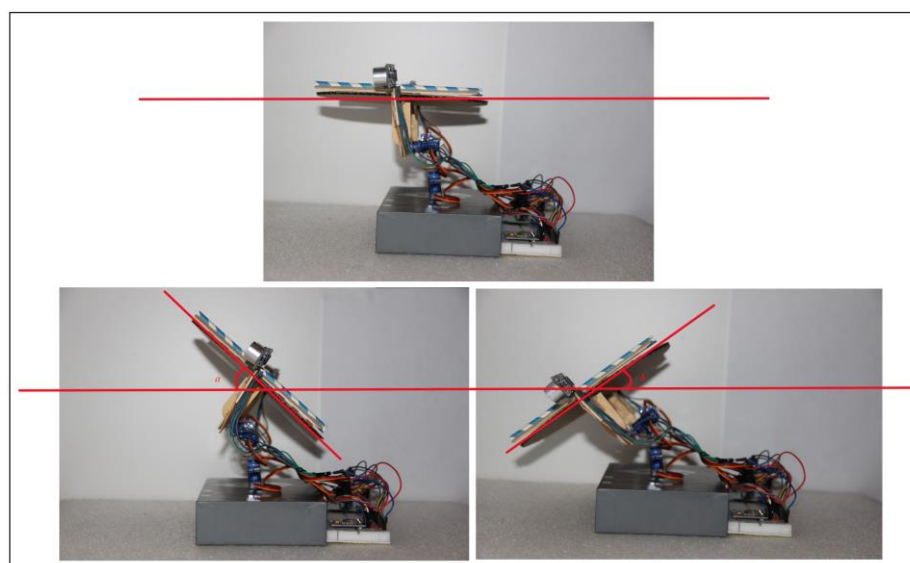


Рисунок 28 – Системы слежения за наклон тангажа самолета

Значение наклона и угла наклона модели самолета отслеживается с помощью монитора последовательного порта, который подключен к Arduino UNO платформы. Модель самолета просто подключается к источнику питания от аккумулятора, но данные передаются в режиме реального времени на интеллектуальную посадочную платформу. Показания наклонов самолета показаны на рисунке 29.

Output	Serial Monitor	×
Message (Enter to send message to 'Arduino Uno' on 'COM6')		
23:09:57.390 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.490 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.623 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.690 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.820 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.889 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.989 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.089 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.189 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.289 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.388 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.519 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.588 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.688 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.788 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.888 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.021 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.087 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.187 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.287 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.386 -> Наклон тангажа=	0	Наклон крена= 0

Output	Serial Monitor	×
Message (Enter to send message to 'Arduino Uno' on 'COM6')		
23:10:36.526 -> Наклон тангажа=	-34	Наклон крена= 4
23:10:36.626 -> Наклон тангажа=	-34	Наклон крена= 4
23:10:36.726 -> Наклон тангажа=	-30	Наклон крена= 4
23:10:36.859 -> Наклон тангажа=	-24	Наклон крена= 4
23:10:36.989 -> Наклон тангажа=	-18	Наклон крена= 4
23:10:37.059 -> Наклон тангажа=	-11	Наклон крена= 4
23:10:37.158 -> Наклон тангажа=	-4	Наклон крена= 3
23:10:37.258 -> Наклон тангажа=	1	Наклон крена= 3
23:10:37.359 -> Наклон тангажа=	5	Наклон крена= 4
23:10:37.458 -> Наклон тангажа=	10	Наклон крена= 4
23:10:37.558 -> Наклон тангажа=	14	Наклон крена= 4
23:10:37.691 -> Наклон тангажа=	18	Наклон крена= 5
23:10:37.791 -> Наклон тангажа=	22	Наклон крена= 5
23:10:37.891 -> Наклон тангажа=	26	Наклон крена= 5
23:10:37.991 -> Наклон тангажа=	29	Наклон крена= 6
23:10:38.090 -> Наклон тангажа=	31	Наклон крена= 6
23:10:38.223 -> Наклон тангажа=	33	Наклон крена= 6
23:10:38.290 -> Наклон тангажа=	34	Наклон крена= 6
23:10:38.390 -> Наклон тангажа=	35	Наклон крена= 7

Рисунок 29 – Показания датчика гироскопа на монитор последовательного порта

На верхнем рисунке показаны значения угла наклона за несколько мгновений до завершения калибровки датчика гироскопа. Цифра в нижней части рисунка 29 – Это значение, которое быстро меняется в зависимости от модификации модели самолета.

3.6 Схема электрических соединений

Схема электрических соединений системы управления аварийной посадки летательного аппарата представлена на рисунке 30.

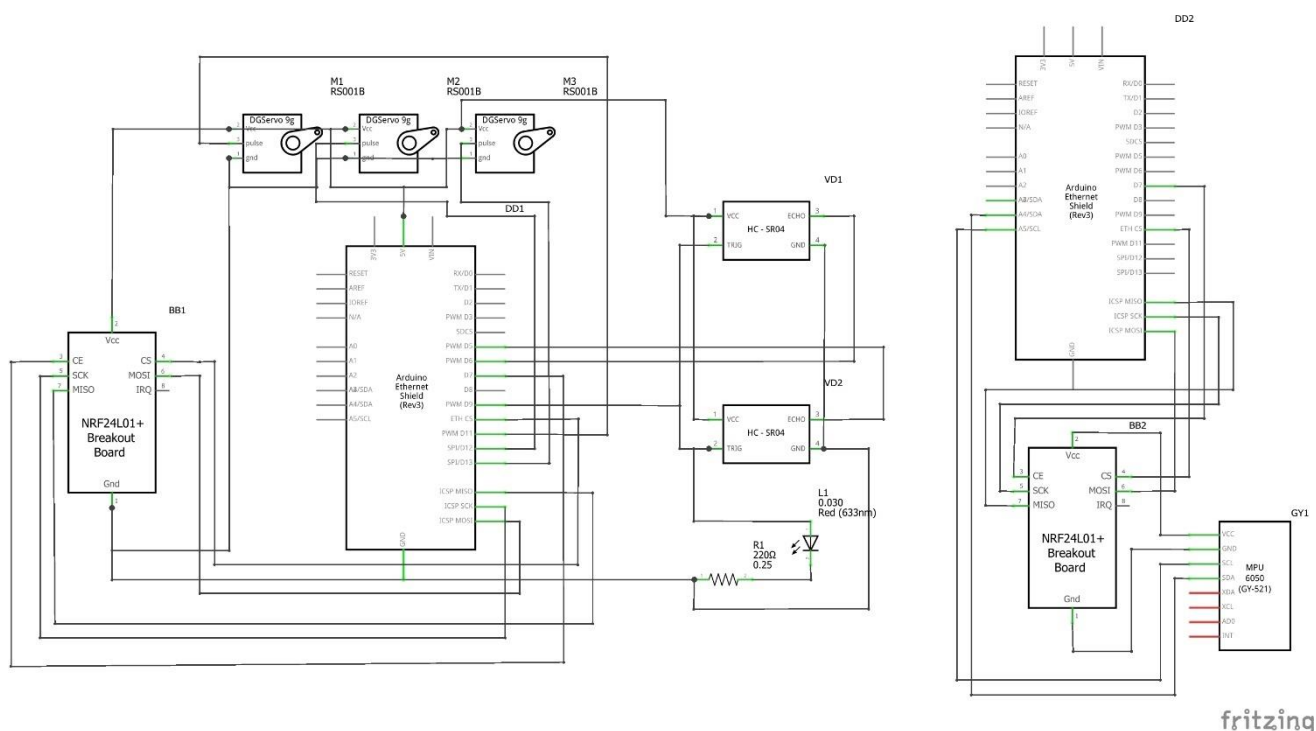


Рисунок 30 – Схема электрических соединений системы управления аварийной посадки летательного аппарата

Схемы электрических соединений имеют важное значение для успеха и полноты проектных отчетов. Они повышают ясность и понимание, а также помогают при устранении неполадок и обслуживании, обеспечивают документирование и соответствие требованиям, облегчают сотрудничество, поддерживают будущие обновления и служат для образовательных целей. Для такого сложного и ответственного с точки зрения безопасности проекта, как система управления аварийной посадкой самолета, эти схемы незаменимы, поскольку дают четкое и детальное визуальное представление об электрической архитектуре. Их важность трудно переоценить, поскольку они внесли значительный вклад в общую надежность, эффективность и безопасность проекта.

Спецификация (лист элементов) к схеме электрических соединений.

[illegible]

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе система управления аварийной посадки летательного аппарата тщательно обсуждена, исследована, смоделирована и протестирована. Модельная система продемонстрировала эффективную работу, но ее внедрение в реальной жизни потребует дальнейших исследований и разработок, особенно при создании интеллектуальной посадочной платформы в гораздо большем масштабе – потенциально более чем в тысячу раз превышающем масштаб модели. Будущее развитие системы является многообещающим, тем более что платформы Arduino продолжают совершенствоваться в плане эффективности, цены и возможностей. Выбор Arduino для этого проекта был основан на тщательном сравнении его преимуществ, что делает его экономически выгодным вариантом для создания системы. Хотя внедрение такой же системы с более качественными компонентами, необходимыми для реальных ситуаций, было бы дорогостоящим, ценность обеспечения безопасности намного перевешивает финансовые затраты.

В ходе этого проекта все датчики и исполнительные устройства были тщательно подобраны. Проект включал подробные функциональные и структурные схемы, иллюстрирующие, как простые системы соединяются в сложное функционирующее целое. Кроме того, был объяснен алгоритм работы, демонстрирующий работу системы управления при аварийной посадке. Проведенные эксперименты показали, что система работает без сбоев, и все результаты были успешно собраны. Однако иногда возникали проблемы, связанные с качеством сборки и энергопотреблением, из-за использования датчиков, которые не являются лучшими на рынке. Высококачественные датчики, хотя и значительно более дорогие, повысили бы надежность системы.

В последние годы развитие систем управления аварийной посадкой было минимальным из-за высокой стоимости и редкости авиакатастроф. Тем не менее, система, разработанная в рамках этого проекта, может значительно

повысить уровень безопасности в авиации. Эта работа представляет ценность для студентов, изучающих системы управления, энтузиастов и авиационных компаний, стремящихся расширить свои знания и генерировать идеи для обеспечения более безопасной посадки. Проект подчеркивает важность постоянного совершенствования и инноваций в системах авиационной безопасности, гарантируя, что стоимость обеспечения безопасности полетов никогда не будет недооценена.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Симонов В.Л., Ерпелев А.В., Давыдова Е.К., Хохлов Е.Г. Моделирование системы управления вертикальным взлетом и посадкой / В сборнике: Современные информационные технологии в образовании, науке и промышленности. Сборник трудов XV Международной конференции, XIII Международного конкурса научных и научно-методических работ. Москва, 2020. – с. 159–161.
- 2 Смулов М.Ю. Обеспечение безопасности полётов воздушных судов на этапах взлёта и посадки в условиях неопределённости информации о внешних возмущениях. Автореферат на соискание ученой степени доктора технических наук (специальность 05.22.14 "Эксплуатация воздушного транспорта"). – МГТУ ГА. – 2004 г. – 40 с.
- 3 Иванов А.А. "Основы динамики машин". Санкт-Петербург: БХВ-Петербург, 2018. 416 с.
- 4 Антонова Н.В., Дубровин Л.Д., Егоров Е.Е. Проектирование авиационных систем кондиционирования воздуха. – М.: Машиностроение, 2006. – 384 с.
- 5 Фильтрация сигналов // alexgyver.ru 2022. URL: <https://alexgyver.ru/lessons/filters/> (дата обращения 04.03.2024)
- 6 Макрушин, С. Н. "Беспроводные технологии. nRF24L01+". Журнал "Мастерок", №6, 2019, с. 45–48.
- 7 Иваницкий, А.В. История радио: От первых экспериментов до современных технологий. М.: Радио и связь, 2010. – 368 с.
- 8 iArduino. Радиомодуль nRF24L01+ 2.4G (Trema-модуль) [Электронный ресурс]. М.: iArduino, 2024. URL: <https://iarduino.ru/shop/Expansion-payments/radio-modul-nrf24l01-2-4g-trema-modul-v2-0.html> (дата обращения: 17.03.2024).
- 9 Zhang N., Liu H., Li J., et al. A Comprehensive Survey on the Research and Development of Renewable Energy Integration in Power Grids [Электронный ресурс]. ScienceDirect: Elsevier, 2022. URL:

<https://www.sciencedirect.com/science/article/pii/S1000936122000012> (дата обращения: 25.03.2024).

10 Техническое описание HC-SR04 Ультразвуковой датчик измерения расстояния. [Электронный ресурс]. URL: <https://iarduino.ru/lib/8c65ac1b85b79ef3fed8c9a9fa699147.pdf> (iarduino.ru) (дата обращения: 02.04.2024).

11 Анисимов А.А. Программирование микроконтроллеров arduino. Москва: ДМК Пресс, 2018.

12 Белоусов Д.И. Arduino для начинающих. Санкт-Петербург: БХВ-Петербург, 2017.

13 Гусев С.Н. Робототехника на основе Arduino. Москва: Технопресс, 2019.

14 Дмитриев А.А. Системы автоматического управления на базе Arduino. Санкт-Петербург: Питер, 2020.

15 Ефимов В.В. Arduino: компьютер для всех. Москва: ДМК Пресс, 2019.

16 Захаров И.П. Программирование на Arduino для начинающих. Москва: ДМК Пресс, 2017.

17 Иванов А.С. Arduino: практический курс. Москва: ДМК Пресс, 2016.

18 Киселев С.Е. Arduino и Raspberry Pi для начинающих. Москва: ДМК Пресс, 2018.

19 Лебедев К.П. Arduino для робототехники. Санкт-Петербург: БХВ-Петербург, 2020.

20 Макетная плата USB Uno R3 Atmega328P (CH340G), для Arduino - купить с доставкой по выгодным ценам в интернет-магазине OZON. [Электронный ресурс]. URL: <https://www.ozon.ru/product/maketnaya-plata-usb-uno-r3-atmega328p-ch340g-dlya-arduino-> (дата обращения 17.04.2024)

21 Никитин Е.В. Arduino: от установки до настройки. Москва: Техносфера, 2017.

- 22 Орлов Д.М. Arduino: сборка, настройка, применение. Москва: ДМК Пресс, 2018.
- 23 Петров И.Н. Arduino: практическое руководство. Москва: Техника, 2019.
- 24 Романов А.В. Arduino для начинающих: от обучения до применения. Москва: Эксмо, 2018.
- 25 Смирнов В.А. Arduino для школьников: учимся делать проекты. Москва: ДМК Пресс, 2019.
- 26 Колесников И.А. Оценка рисков и обеспечение безопасности полетов. - М.: Издательство Питер, 2019.
- 27 Никитин В.Ф. Основы безопасности воздушных перевозок. - М.: КНОРУС, 2018.
- 28 Родионов Г.А. Авиационная безопасность: Учебное пособие. - М.: Федерация, 2019.
- 29 Зайцев П.Н. Методология оценки безопасности полетов. - М.: Издательство Флинта, 2018.
- 30 Теро Карвинен, Киммо Карвинен, Вилле Валтокари. Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi / М.: ООО "И.Д. Вильямс", 2016. – 448 с.

ПРИЛОЖЕНИЕ

Приложение А

(Обязательное)

Листинг программ для передатчика

```
#include <SPI.h> // Подключаем библиотеку для работы с шиной SPI.
#include <nRF24L01.h> // Подключаем файл настроек из библиотеки RF24.
#include <RF24.h> // Подключаем библиотеку для работы с nRF24L01+.
#include <Wire.h>
#include <MPU6050_light.h>
#include "SPI.h"

RF24 radio(7, 10); // Создаём объект radio для работы с библиотекой RF24,
указывая номера выводов модуля (CE, SS)
int myData[3]; // Объявляем массив для хранения и передачи данных.
int ackData[10]; // Объявляем массив для получения данных из пакета
подтверждения приёма (до 32 байт включительно).
MPU6050 mpu(Wire);
unsigned long timer=0;
void setup(){ //
  Serial.begin(9600);
  radio.begin (); // Иницилируем работу модуля nRF24L01+.
  radio.setChannel (70); // Указываем канал передачи данных (от 0 до 125), 27 -
значит передача данных осуществляется на частоте 2,427 ГГц.
  radio.setDataRate (RF24_2MBPS); // Указываем скорость передачи данных
(RF24_250KBPS, RF24_1MBPS, RF24_2MBPS), RF24_1MBPS - 1Мбит/сек.
  radio.setPALevel (RF24_PA_MAX); // Указываем мощность передатчика
(RF24_PA_MIN=-18dBm, RF24_PA_LOW=-12dBm, RF24_PA_HIGH=-6dBm,
RF24_PA_MAX=0dBm).
  radio.enableAckPayload(); // Указываем что в пакетах подтверждения приёма
есть блок с пользовательскими данными.
```

```

// radio.enableDynamicPayloads(); // Разрешить динамически изменяемый
размер блока данных на всех трубах.

radio.openWritingPipe (0xAABBBCCDD11LL); // Открываем трубу с адресом
0xAABBBCCDD11 для передачи данных (передатчик может одновременно
вещать только по одной трубе).

Wire.begin();

// setup mpu

byte situasi = mpu.begin();

Serial.print(F("MPU6050 status: "));

Serial.println(situasi);

while (situasi != 0) { }

Serial.println(F("Calculating offsets, do not move MPU6050"));

delay(1000);

// mpu.upsideDownMounting = true; // uncomment this line if the MPU6050 is
mounted upside-down

mpu.calcOffsets(); // gyro and accelero

Serial.println("Done!\n");

}

void loop(){ //
// Заполняем данными массив для отправки:

mpu.update();

myData[0] = mpu.getAngleX(); // Данные для сервомотора на приемной
стороне myData[1] = random(200, 250);

myData[1] = mpu.getAngleY();

myData[2] = mpu.getAngleZ();

radio.write(&myData, sizeof(myData)); // Отправляем данные из массива
myData указывая сколько байт массива мы хотим отправить.

delay(10); // Устанавливаем задержку

}

```

Приложение Б

(Обязательное)

Листинг программ для приемника

```
#include <SPI.h> // Подключаем библиотеку для работы с шиной SPI.
#include <nRF24L01.h> // Подключаем файл настроек из библиотеки RF24.
#include <RF24.h> // Подключаем библиотеку для работы с nRF24L01+.
#include <Servo.h> // подключаем библиотеку для работы с серводвигатели.

//radio prep
RF24 radio(7, 10); // Создаём объект radio (CE, SS).
int myData[3]; // Объявл.массив для приёма и хран.данных (до 32 байт)
//#include <iarduino_MultiServo.h> // Подключаем библиотеку
iarduino_MultiServo
// iarduino_MultiServo MSS; // Объявляем объект MSS, для работы с
библиотекой
//motor prep
Servo myservoyaw;
int pos = 90;
//ultrasonice prep
const int trigPin1 = 12;
const int trigPin2 = 8;
const int echoPin1 = 4;
const int echoPin2 = 2;
//distance to start tracking for ultrasonic
int t_distance = 100; //in cm
int distance1;int distance2;
const int NUM_READ = 3;
void setup(){ //
Serial.begin(9600);
while(!Serial){} // Ждём готовность Serial к передаче данных в Serial()
```

```

// MSS.servoSet(0, SERVO_SG90); // Сервопривод SG90 подключён к выводу
№ 0 MultiServoShield.

// MSS.begin();

//radio setup
radio.begin(); // Иницилируем работу nRF24L01+
radio.setChannel (70); // Указываем канал передачи данных (от 0 до 125), 27 -
значит передача данных осуществляется на частоте 2,427 ГГц.
radio.setDataRate (RF24_2MBPS); // Указываем скорость передачи данных
(RF24_250KBPS, RF24_1MBPS, RF24_2MBPS), RF24_1MBPS - 1Мбит/сек.
radio.setPALevel (RF24_PA_MAX); // Указываем мощность передатчика
(RF24_PA_MIN=-18dBm, RF24_PA_LOW=-12dBm, RF24_PA_HIGH=-6dBm,
RF24_PA_MAX=0dBm).

radio.enableAckPayload(); // Указываем что в пакетах подтверждения приёма
есть блок с пользовательскими данными.

radio.openReadingPipe (1, 0xAABBCCDD11LL); // Открываем 1 трубу с
адресом 0xAABBCCDD11, для приема данных.

radio.startListening (); // Включаем приемник, начинаем прослушивать
открытые трубы.

//radio.writeAckPayload (1, &ackData, sizeof(ackData) ); // Помещаем данные
всего массива ackData в буфер FIFO. Как только будут получены любые
данные от передатчика на 1 трубе, то данные из буфера FIFO будут
отправлены этому передатчику вместе с пакетом подтверждения приёма его
данных.

//motor setup
myservoyaw.attach(11);

//myservoyaw.write(pos);

//ultrasonic setup
pinMode(trigPin1, OUTPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin1, INPUT);

```

```

pinMode(echoPin2, INPUT);
} // В модуле имеется 3 буфера FIFO, значит в них одновременно может
находиться до трёх разных или одинаковых данных для ответа по одной или
разным трубам.

// После отправки данных из буфера FIFO к передатчику, соответствующий
буфер очищается и способен принять новые данные для отправки.

void loop(){
  //Left sensor
  digitalWrite(trigPin2, LOW);
  //delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  //delayMicroseconds(5);
  digitalWrite(trigPin2, LOW);
  int duration = pulseIn(echoPin2, HIGH);
  runMiddleArifmOptim(duration);
  //Right sensor
  digitalWrite(trigPin1, LOW);
  //delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  //delayMicroseconds(5);
  digitalWrite(trigPin1, LOW);
  int duration2 = pulseIn(echoPin1, HIGH);
  runMiddleArifmOptim(duration2);
  //calculate distance based on duration of each sensor
  distance1 = (duration2*.0343)/2;
  distance1 = constrain(distance1, 5, 300);
  distance2 = (duration*.0343)/2;
  distance2 = constrain(distance2, 5, 300);
  //int diff = distance1 - distance2;
  //int teleng = pos+diff;

```



```

follow();

if(radio.available()){ // Если в буфере приёма есть принятые данные от
передатчика, то ...

radio.read ( &myData, sizeof(myData) );} // Читаем данные из буфера приёма в
массив myData указывая сколько всего байт может поместиться в массив.

//myservo.write(myData[0]); bagi servo motor gerak ikot bacaan gyro

// Выводим в Serial то, что пришло от передатчика:
// ДЛЯ УСКОРЕНИЯ РАБОТЫ ВСЕЙ СИСТЕМЫ В ОКОНЧАТ.ВЕРСИИ
ЗАКОММЕНТИРОВАТЬ ВСЕ Serial.print(..)

Serial.print("Тангажа=");          Serial.print('\t');          Serial.print(myData[0]);
Serial.print('\t');

Serial.print("Крена="); Serial.print('\t'); Serial.print(myData[1]); Serial.print('\t');
Serial.print("Рыскание=");          Serial.print('\t');          Serial.print(myData[2]);
Serial.print('\t');

Serial.println();
}

void follow()
{
if (distance1 <= t_distance || distance2 <= t_distance)
{
if (distance1 + 5 < distance2){ pos = pos -3; }
if (distance2 + 5 < distance1) { pos = pos +3; }
}
if (pos > 180)
{
pos = 180;
}
if (pos < 0)
{
pos = 0;
}
}

```

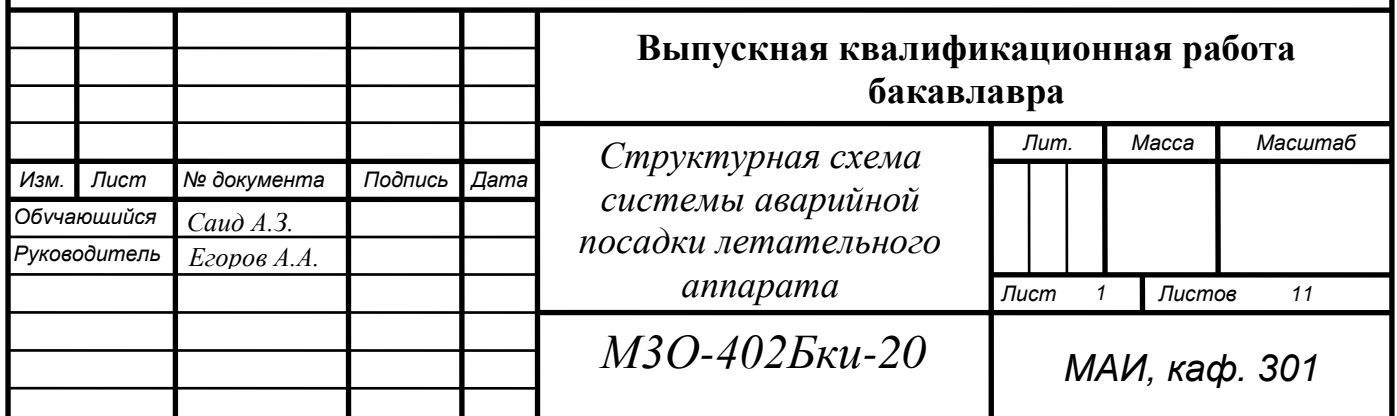
```

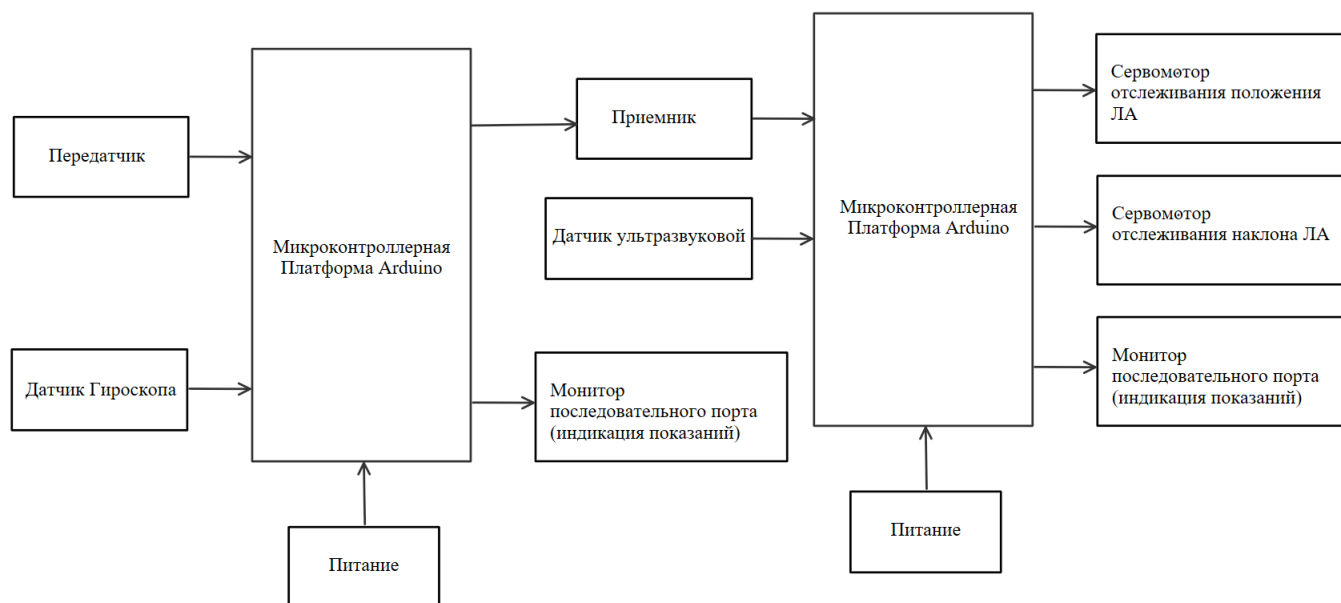
}
myservoyaw.write(pos);
}
//оптимальное бегущее среднее арифметическое
float runMiddleArifmOptim(float newVal) {
    static int t = 0;
    static float vals[NUM_READ];
    static float average = 0;
    if (++t >= NUM_READ) t = 0; // перемотка t
    average -= vals[t];        // вычитаем старое
    average += newVal;         // прибавляем новое
    vals[t] = newVal;          // запоминаем в массив
    return ((float)average / NUM_READ);
}

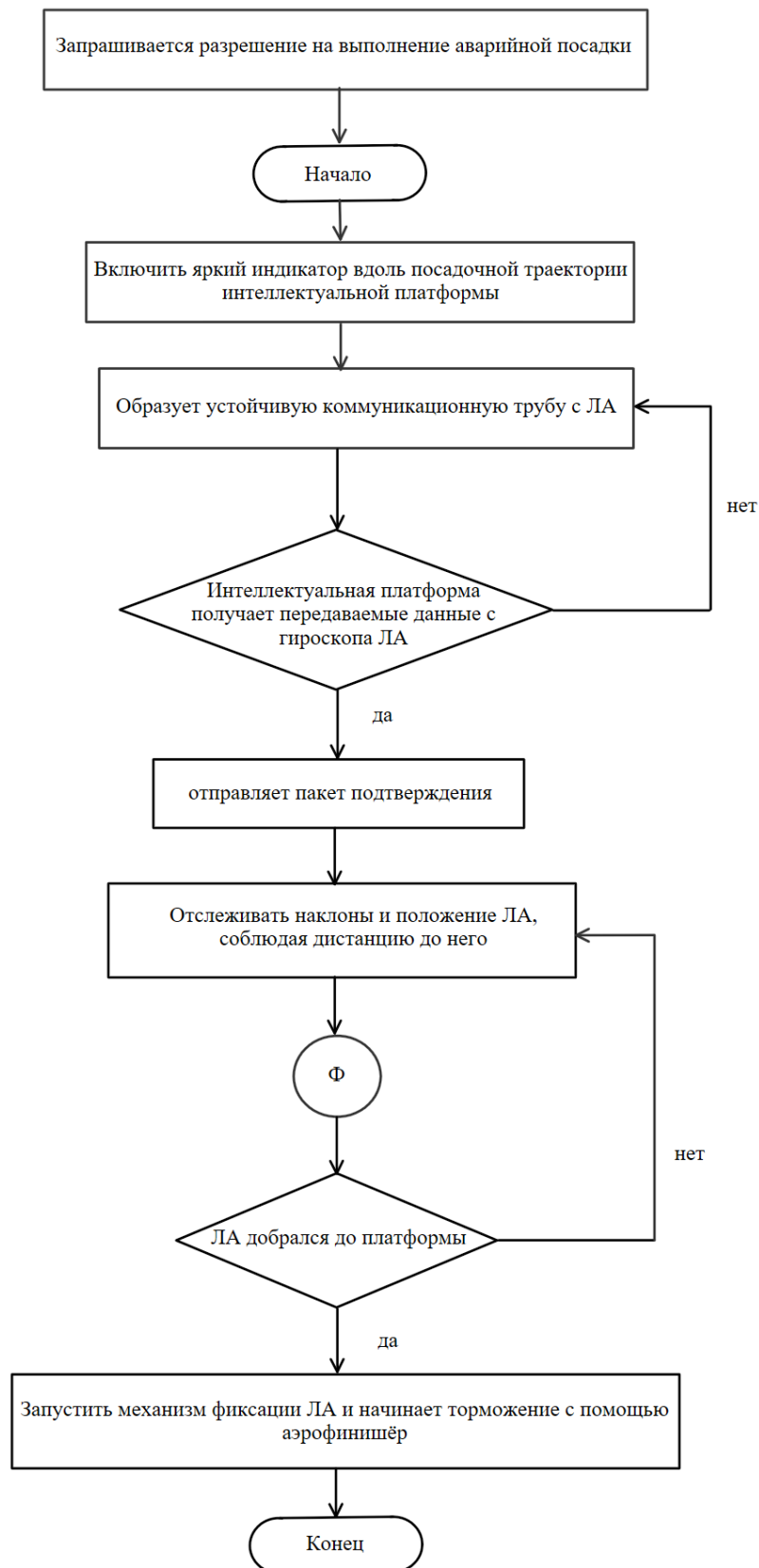
```

Приложение В
(Обязательное)

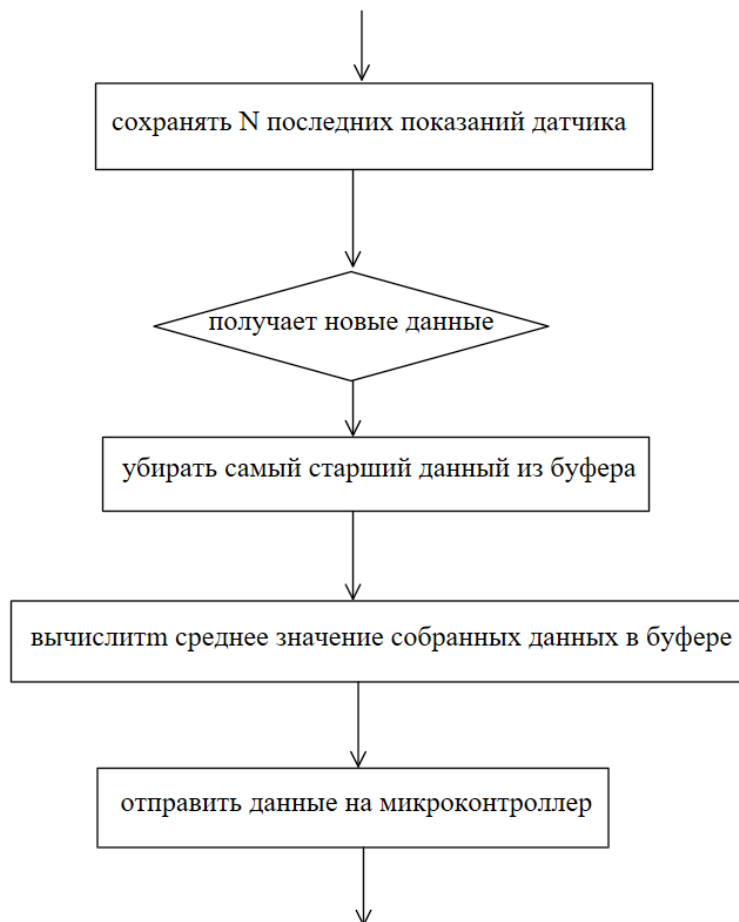
Иллюстративно-графические материалы



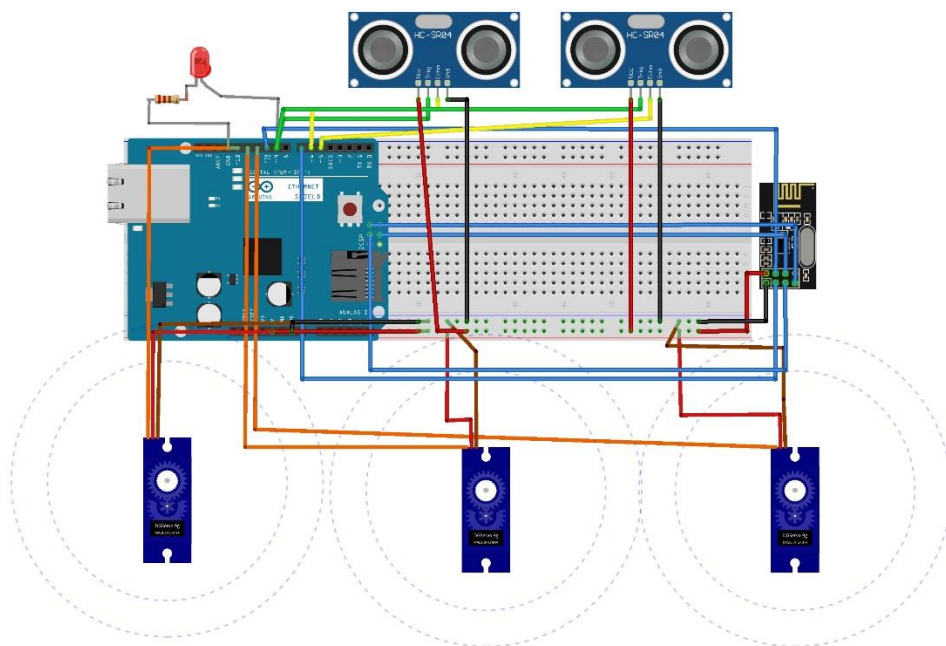
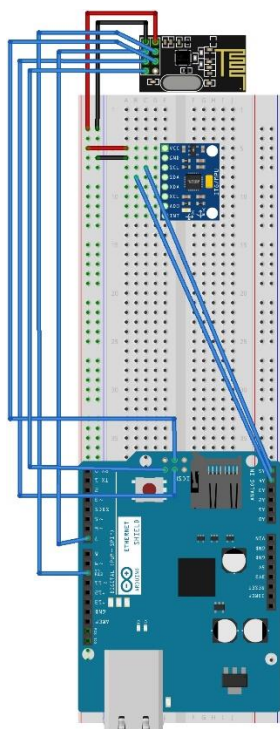




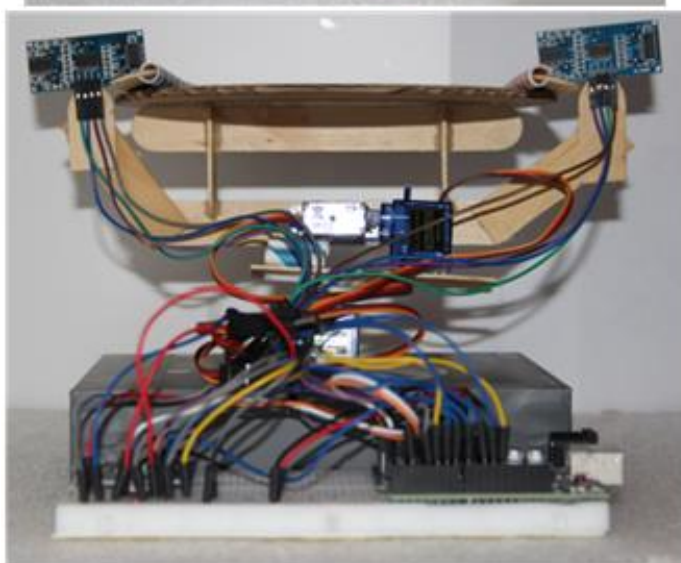
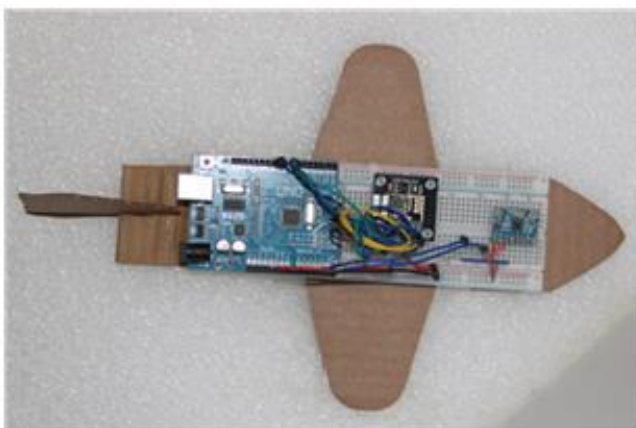
					Выпускная квалификационная работа бакалавра								
					<i>Алгоритмы функционирования системы управления аварийной посадки</i>				Лит.	Масса	Масштаб		
Изм.	Лист	№ документа	Подпись	Дата	<i>МЗО-402Бки-20</i>				Лист	3	Листов	11	
Обучающийся	Саид А.З.												
Руководитель	Егоров А.А.												
									<i>МАИ, каф. 301</i>				



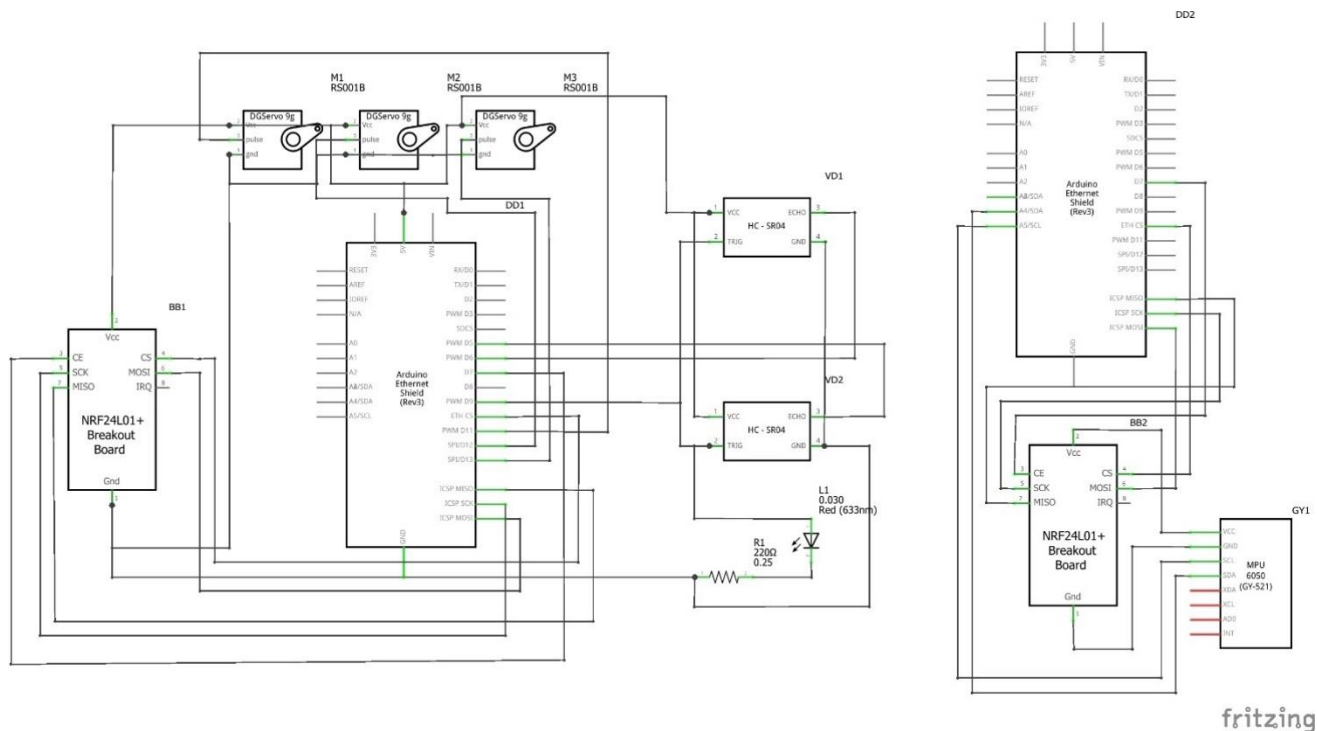
					Выпускная квалификационная работа бакалавра					
					Алгоритмы функционирования системы управления аварийной посадки	Лит.	Масса	Масштаб		
Изм.	Лист	№ документа	Подпись	Дата						
Обучающийся	Саид А.З.									
Руководитель	Егоров А.А.									
						Лист	4	Листов	11	
					МЗО-402Бки-20		МАИ, каф. 301			



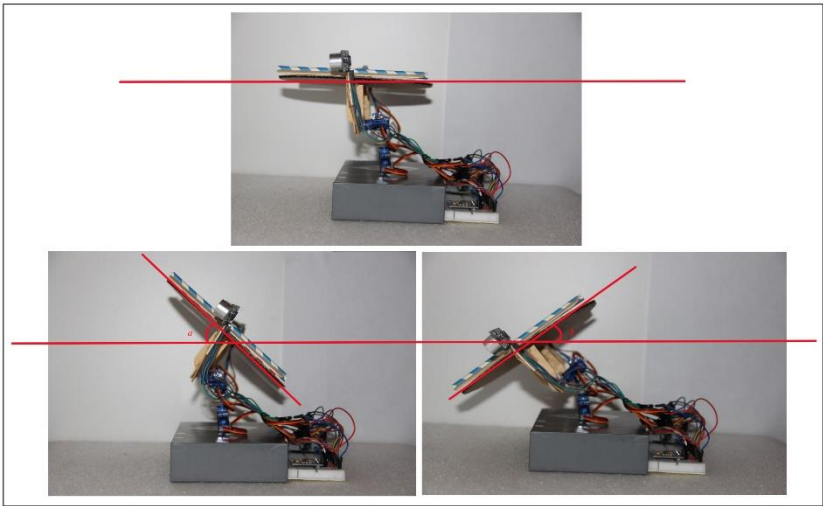
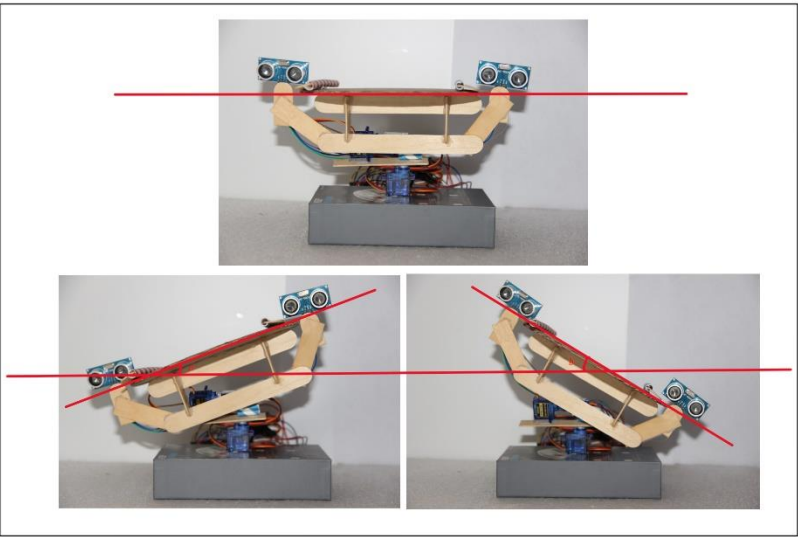
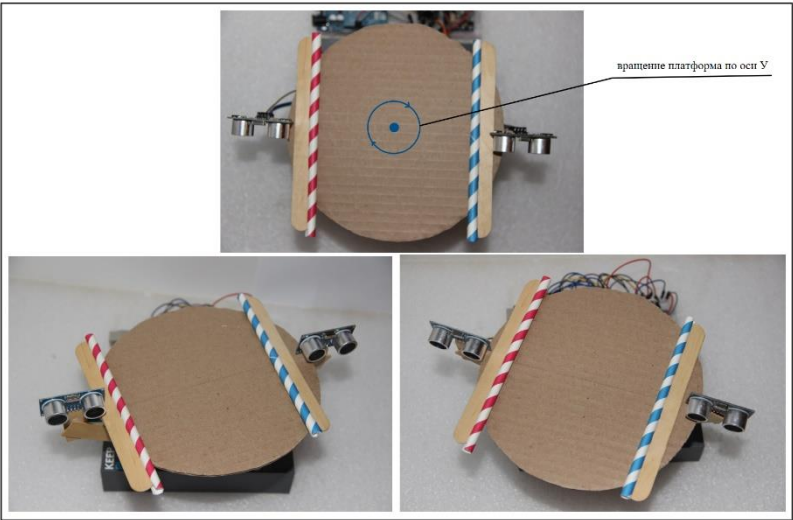
					Выпускная квалификационная работа бакалавра			
					<i>Моделирование системы управления аварийной посадки</i>	Лит.	Масса	Масштаб
Изм.	Лист	№ документа	Подпись	Дата				
Обучающийся		Саид А.З.						
Руководитель		Егоров А.А.						
						Лист	5	Листов 11
					<i>МЗО-402Бки-20</i>		<i>МАИ, каф. 301</i>	



					Выпускная квалификационная работа бакалавра			
					Экспериментальная установка системы управления аварийной посадки	Лит.	Масса	Масштаб
Изм.	Лист	№ документа	Подпись	Дата				
Обучающийся	Саид А.З.							
Руководитель	Егоров А.А.							
						Лист 6	Листов 11	
					МЗО-402Бки-20	МАИ, каф. 301		



					Выпускная квалификационная работа бакалавра				
Изм.	Лист	№ документа	Подпись	Дата	Схема электрических соединений	Лит.	Масса	Масштаб	
Обучающийся	Саид А.З.								
Руководитель	Егоров А.А.								
					МЗО-402Бки-20	Лист	7	Листов	11
						МАИ, каф. 301			



					Выпускная квалификационная работа бакалавра			
					Результаты экспериментов	Лит.	Масса	Масштаб
Изм.	Лист	№ документа	Подпись	Дата				
Обучающийся		Саид А.З.						
Руководитель		Егоров А.А.						
						Лист 9	Листов 11	
					МЗО-402Бки-20		МАИ, каф. 301	

```
// Выводим в Serial то, что пришло от передатчика:
// ДЛЯ УСКОРЕНИЯ РАБОТЫ ВСЕЙ СИСТЕМЫ В ОКОНЧАТ.ВЕРСИИ ЗАКОММЕНТИРОВАТЬ ВСЕ Serial.print(..)
Serial.print("Тангажа="); Serial.print('\t'); Serial.print(myData[0]); Serial.print('\t');
Serial.print("Крена="); Serial.print('\t'); Serial.print(myData[1]); Serial.print('\t');
Serial.print("Рыскание="); Serial.print('\t'); Serial.print(myData[2]); Serial.print('\t');
Serial.println();
}
```

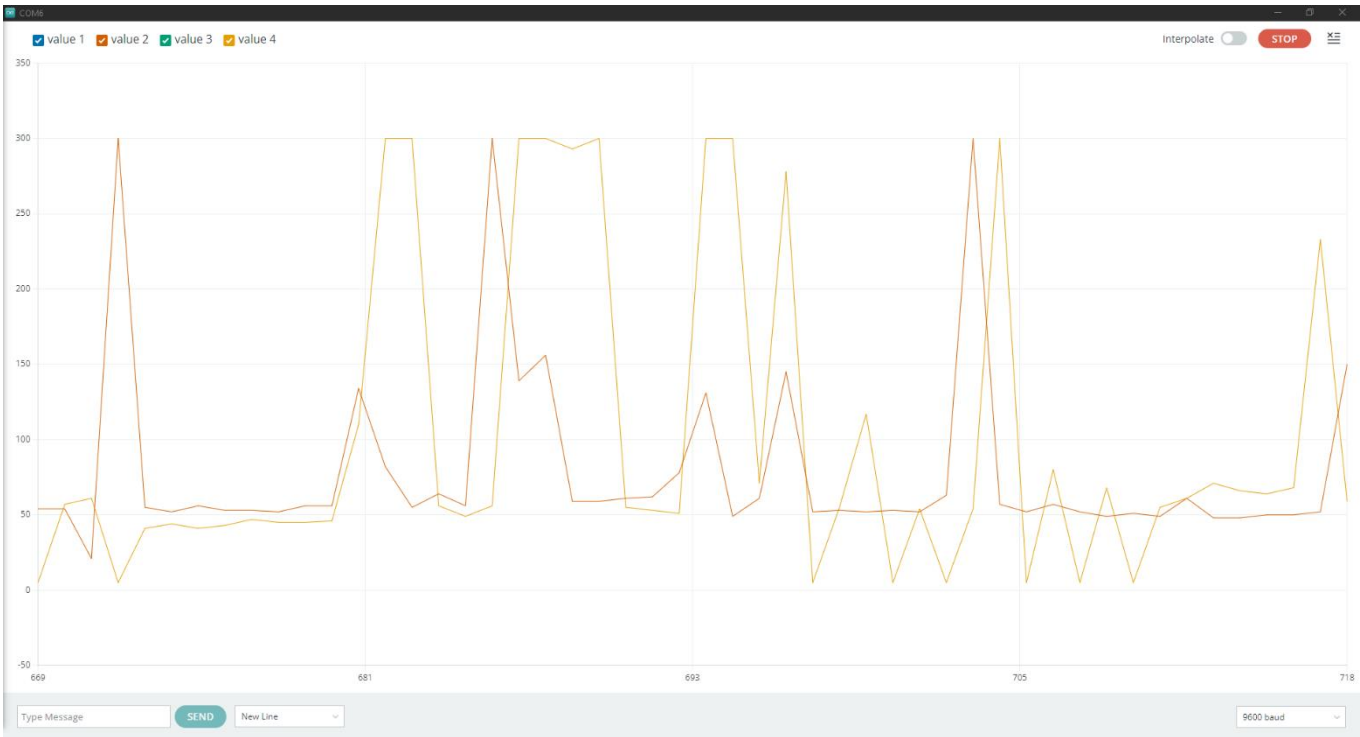
Output	Serial Monitor	×
Message (Enter to send message to 'Arduino Uno' on 'COM6')		
23:09:57.390 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.490 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.623 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.690 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.820 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.889 -> Наклон тангажа=	0	Наклон крена= 0
23:09:57.989 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.089 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.189 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.289 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.388 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.519 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.588 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.688 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.788 -> Наклон тангажа=	0	Наклон крена= 0
23:09:58.888 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.021 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.087 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.187 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.287 -> Наклон тангажа=	0	Наклон крена= 0
23:09:59.386 -> Наклон тангажа=	0	Наклон крена= 0

Output	Serial Monitor	×
Message (Enter to send message to 'Arduino Uno' on 'COM6')		
23:10:36.526 -> Наклон тангажа=	-34	Наклон крена= 4
23:10:36.626 -> Наклон тангажа=	-34	Наклон крена= 4
23:10:36.726 -> Наклон тангажа=	-30	Наклон крена= 4
23:10:36.859 -> Наклон тангажа=	-24	Наклон крена= 4
23:10:36.989 -> Наклон тангажа=	-18	Наклон крена= 4
23:10:37.059 -> Наклон тангажа=	-11	Наклон крена= 4
23:10:37.158 -> Наклон тангажа=	-4	Наклон крена= 3
23:10:37.258 -> Наклон тангажа=	1	Наклон крена= 3
23:10:37.359 -> Наклон тангажа=	5	Наклон крена= 4
23:10:37.458 -> Наклон тангажа=	10	Наклон крена= 4
23:10:37.558 -> Наклон тангажа=	14	Наклон крена= 4
23:10:37.691 -> Наклон тангажа=	18	Наклон крена= 5
23:10:37.791 -> Наклон тангажа=	22	Наклон крена= 5
23:10:37.891 -> Наклон тангажа=	26	Наклон крена= 5
23:10:37.991 -> Наклон тангажа=	29	Наклон крена= 6
23:10:38.090 -> Наклон тангажа=	31	Наклон крена= 6
23:10:38.223 -> Наклон тангажа=	33	Наклон крена= 6
23:10:38.290 -> Наклон тангажа=	34	Наклон крена= 6
23:10:38.390 -> Наклон тангажа=	35	Наклон крена= 7

					<h2 style="text-align: center;">Выпускная квалификационная работа бакалавра</h2>				
Изм.	Лист	№ документа	Подпись	Дата	<h3>Программный код</h3>	Лит.		Масса	Масштаб
Обучающийся	Саид А.З.								
Руководитель	Егоров А.А.								
					Лист 10		Листов 11		
					МЗО-402Бки-20		МАИ, каф. 301		

```
distance1 = (duration2*.0343)/2;
distance1 = constrain(distance1, 5, 300);
distance2 = (duration*.0343)/2;
distance2 = constrain(distance2, 5, 300);
Serial.print(distance1); Serial.print("\t");
Serial.println(distance2);

follow();
```



					Выпускная квалификационная работа бакалавра						
					Программный код	Лит.		Масса	Масштаб		
Изм.	Лист	№ документа	Подпись	Дата							
Обучающийся	Саид А.З.										
Руководитель	Егоров А.А.										
					МЗО-402Бки-20	Лист 11		Листов 11			
						МАИ, каф. 301					