НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И.САТПАЕВА»

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

Шанаева Маншук Темиргалиевна

Движение робота с использованием распознавания цветных объектов

Дипломная работа

Образовательная программа: 6B07111 Робототехника и мехатроника

Алматы 2025

НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И.САТПАЕВА»

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой РТиТСА
канд.тех.наук,профессор
_____К.А. Ожикенов
«___»_____2025г

Дипломная работа

На тему: «Движение робота с использованием распознавания цветных объектов»

Образовательная программа: 6В07111 Робототехника и мехатроника

Выполнил Шанаева Маншук Темиргалиевна
Рецензент Научный руководитель
PhD,ассоциированный профессор Магистр технических наук, преподаватель
_____Алимбаева Ж. Н. _____Е.А. Игембай
"___"_____2025 г. "___"_____2025 г.

Алматы 2025

НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И.САТПАЕВА»

Институт автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»

УТВЕРЖДАЮ
Заведующий кафедрой РТиТСА
канд.тех.наук,профессор
К.А. Ожикенов
«01» 06 2025г

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающиеся: Шаншаева Маншук Темиргалиевна
Тема: Разработка интеллектуального протеза с использованием сенсорных систем
Утверждена приказом проректора по академической работе:

№ 521-П/Б	13.12.2024
Срок сдачи законченного проекта	«23» мая 2025 г.

Исходные данные к дипломному проекту:
А) Проведен анализ необходимых роботов в плане захвата
Б) Разработана и протестирована прототип робота
В) Изучен робот для улучшения

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): представлены 12 слайда презентации. Рекомендуемая основная литература: из 19 наименований.

ГРАФИК
подготовка дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Теоретическая часть	04.03.2025	выполнено
Практическая часть	12.04.2025	выполнено
Сбор данных	30.04.2025	выполнено
Финальная версия	14.05.2025	выполнено

Подписи
консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты И.О.Ф. (уч.степень.звание)	Дата подписания	Подпись
Нормоконтролер	Кальменов Е. Т. М.т.н., старший преподаватель	05.06.2025	

Научный руководитель _____ Игембай Е. А.

Задание принял к исполнению обучающийся _____ Шанаева М. Т.

Дата «05» 06 2025 г

АҢДАТПА

Дипломдық жұмыс түсті объектілерді тану және бақылау функциялары бар компьютерлік көру жүйесі негізінде мобильді роботты әзірлеуге арналған. Кадрларды талдау арқылы мобильді роботтың бастапқы маршрутын құру мүмкіндігі қарастырылған.

Әзірленген робот камерамен және түсті объектілерді бөліп, олардың кеңістіктегі орнын анықтай алатын бағдарламалық қамтамасыз етумен жабдықталған. Жүйе Raspberry Pi бірплаталы компьютері мен CSI-камера негізінде іске асырылған.

Ұсынылған робот қоршаған ортаның өзгерістеріне жедел әрекет ету талап етілетін динамикалық ортада, денсаулық сақтау саласында көмекші ретінде, сондай-ақ білім беру саласында, робототехника және бағдарламалау негіздерін үйрету үшін бейімделетін жүйе мысалы ретінде қолданылуы мүмкін. Осылайша, бұл жұмыс ғылыми және қолданбалы маңыздылыққа ие.

АННОТАЦИЯ

Дипломная работа посвящена разработке мобильного робота с функциями компьютерного зрения, предназначенного для распознавания и отслеживания цветных объектов, и с помощью анализа кадров строить базовый маршрут мобильного робота.

Разработанный робот оснащён камерой и программным обеспечением, способным выделять объекты по цвету и определять их положение в пространстве. Система реализована на базе одноплатного компьютера Raspberry Pi с использованием CSI-камеры.

Представленный робот может быть использован в динамических средах, где требуется реакция на изменения окружающей обстановки, здравоохранение, в качестве помощника, а также в образовании, для обучение основам робототехники и программирование в качестве примера адаптивных систем. Таким образом, работа несёт в себе как научную, так и прикладную ценность.

ABSTRACT

This thesis is dedicated to the development of a mobile robot equipped with computer vision capabilities for recognizing and tracking colored objects. By analyzing video frames, the robot is able to construct a basic movement route.

The developed robot is equipped with a camera and software capable of detecting objects by their color and determining their position in space. The system is implemented using a Raspberry Pi single-board computer and a CSI camera.

The proposed robot can be used in dynamic environments where rapid response to environmental changes is required, in healthcare as an assistant, and in education as a tool for teaching the fundamentals of robotics and programming, serving as an example of an adaptive system. Thus, this work has both scientific and practical significance.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире робототехника и автоматизация играет одной из самых ключевых ролей в различных сферах человеческой деятельности. Начиная от промышленного производства до бытового применения. Одним из важных аспектов в робототехнике является обеспечения автономного движения робота а также устройств в пространстве. Для решение таких задач активно используются технологии компьютерного зрения, которые позволяют роботам воспринимать окружающую среду и принимать решения на основе визуальной информации. Использование цветных объектов для движение робота является одним из способов для навигации, что делает эту тему более актуальной.

В данное время есть разные способы визуализации движения роботов и устройств. Однако распознавание цветовых ориентиров остаётся одним из наиболее понятных и легко реализуемых способов. Это открывает возможности для его применения в таких сферах, как логистика, здравоохранение, сельское хозяйство и образование, где требуется простая, но надёжная автономная навигация.

Целью данной дипломной работы является разработка и реализация системы движения мобильного робота с использованием распознавания цветных объектов как основного метода ориентации в пространстве. Работа охватывает как аппаратную часть сборки робота, так и программную реализацию алгоритмов компьютерного зрения на базе одноплатного компьютера Raspberry Pi [15]. Разработка направлена на создание недорогой, универсальной платформы, пригодной для дальнейшего расширения и адаптации под различные прикладные задачи. Для достижения цели были сформулированы следующие задачи:

– Изучить основы мобильной робототехники и компьютерного зрения;

– Спроектировать и собрать аппаратную часть робота на базе Raspberry Pi;

– Разработать алгоритм управления движением робота с использованием цветовой сегментации изображения;

– Провести тестирование системы в реальных условиях;

– Выполнить 3D-моделирование конструкции для основного робота а также улучшение программной части;

– Сбор конструкции основного робота.

Таким образом, данная работа направлена на создание простого и эффективного решения для автономного передвижения мобильного робота. Полученные результаты могут быть применены в образовательной сфере, а также служить основой для дальнейших исследований и усовершенствований в области мобильной робототехники.

1 Мобильные роботы

Мобильные роботы — это автономные или полуавтономные устройства, способные перемещаться в пространстве без фиксированной базы. В отличие от стационарных роботов, они могут передвигаться при помощи различных механизмов: колёс, гусениц, шагающих опор и т.д. Они оснащены сенсорами, системами управления и алгоритмами планирования маршрута, что позволяет им адаптироваться к меняющейся среде и выполнять задачи в реальном времени.[1]

Основу функционирования мобильного робота составляют три ключевых компонента:

- Аппаратная платформа – шасси с приводом (колёсным, гусеничным, шаговым и т.п.), обеспечивающее физическое перемещение.
- Сенсорная система – различные датчики (ультразвуковые, инфракрасные, лазерные, инерциальные и визуальные), используемые для восприятия окружающей среды.
- Управляющая система – вычислительный модуль, который анализирует данные сенсоров, принимает решения и управляет приводами.[2]

Мобильные роботы могут быть классифицированы по типу среды, в которой они функционируют (наземные, воздушные, подводные), а также по степени автономности – от полностью управляемых оператором до полностью автономных систем с элементами искусственного интеллекта.

Ключевой особенностью мобильных роботов является способность локализовываться и планировать маршрут. Для этого используются методы одновременной локализации и построения карты (SLAM), а также алгоритмы планирования движения (например, A*, D*, Rapidly-Exploring Random Trees). Робот который предоставлен в моей работе не использует выше упомянутые методы локализации, такие как SLAM или алгоритмы планирования маршрута, но все же относиться к классу мобильных роботов, поскольку способен самостоятельно передвигаться в окружающей среде и реагировать на визуальные ориентиры, такие как цвет или линии.

Первым мобильным роботом который мог воспринимать и рассуждать об окружающей среде был робот Shakey, робот был предметом исследований Центра искусственного интеллекта SRI с 1966 по 1972 год. Он был способен выполнять задачи, связанные с планированием, поиском маршрутов и перемещением простых объектов. Shakey стал важным шагом в развитии ИИ и робототехнике, Сегодня его можно увидеть в Музее истории компьютеров. Ниже предоставлено изображение первого мобильного робота Shakey. [4]

Рисунок 1.1 – Первый мобильный робот Shakey

Сегодня мобильные роботы активно применяются в логистике, сельском хозяйстве, медицине, военной сфере, поисково-спасательных операциях и в исследовательских миссиях (например, марсоходы NASA).

Таким образом, мобильные роботы представляют собой комплексные киберфизические системы, сочетающие механику, электронику и программное обеспечение, позволяющие эффективно выполнять задачи в условиях динамичной и частично неизвестной среды.

1.1 Движение в робототехнике

Движение является одним из основных аспектов в робототехнике, которое дает роботу способность перемещаться в пространстве, взаимодействовать окружающей средой и выполнять поставленные задачи. Управление движением робота включает в себя как аппаратные, так и программные компоненты, обеспечивающие точное и стабильное выполнение команд. Способность робота к перемещению – это основа его взаимодействия с окружающей средой. Благодаря мобильности робот может выполнять широкий спектр задач: от простой транспортировки предметов до навигации в сложных или недоступных для человека условиях.

В зависимости от конструктивных особенностей и условий эксплуатации, роботы могут использовать различные способы перемещения, которые принято называть локомоцией.

Эти способы определяются типом привода, формой взаимодействия с поверхностью, устойчивостью и проходимостью робота. Ниже рассмотрим основные классификации способов передвижения роботов, их характеристики, преимущества и ограничения.

Классификация способов передвижения роботов:

1) Колёсный способ передвижения:

Колёсная локомоция является одной из самых распространённых форм передвижения в робототехнике. В зависимости от числа и расположения колёс различают несколько подтипов: двухколёсные, трёхколёсные, четырёх- и шестиколёсные платформы, а также роботы с омни или механум-колёсами. Преимущества способа колесного передвижения является простата конструкции и реализация в собираемых проектах, также при небольшом количестве колес возможно достичь высокий уровень маневренности, самым оптимальным количеством колес является четыре, так как геометрически обеспечивает минимально необходимое число точек опоры, необходимое для достижения состояние равновесия. При этом стоит учитывать ограниченную проходимость по неровной местности, нелинейная скорость на гладких и наклонных поверхностях.

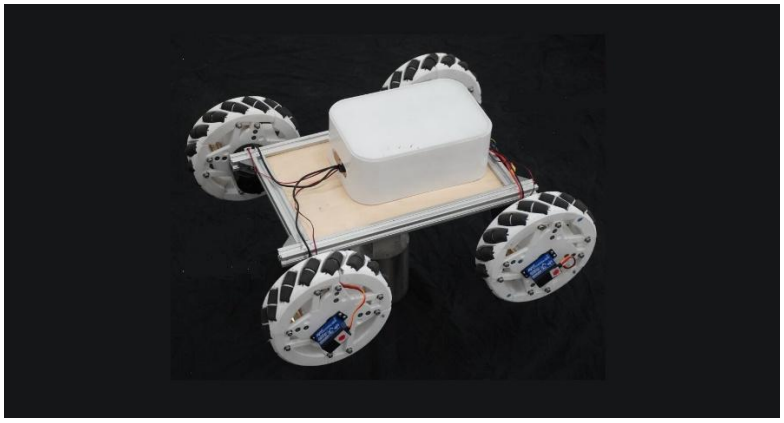


Рисунок 1.2 – Модель робота с колесной системой движения

2) Гусеничный способ передвижения:

Гусеничная локомоция используется в тех случаях, когда требуется высокая проходимость и устойчивость на пересечённой местности. Роботы с гусеницами обладают увеличенной площадью контакта с поверхностью, что обеспечивает лучшее сцепление и устойчивость. Среди преимуществ гусеничного привода можно выделить отличную проходимость на рыхлой, сыпучей или каменистой поверхности, повышенную устойчивость за счёт широкой зоны контакта, а также надёжность в условиях бездорожья. Однако у такой системы есть и недостатки: более сложная и тяжёлая конструкция по сравнению с колёсными платформами, пониженная манёвренность, более

высокая потребляемая мощность и быстрый износ гусениц при интенсивной эксплуатации.

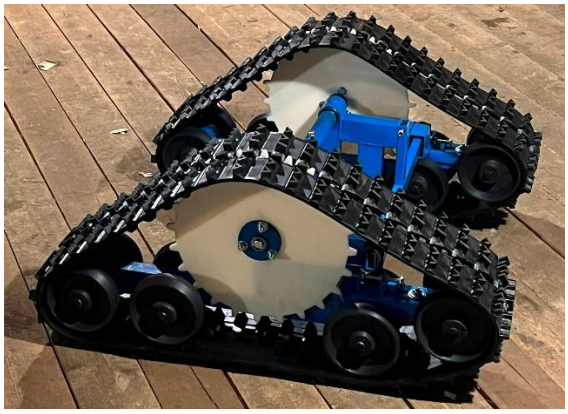


Рисунок 1.3 – Тип передвижения: гусеничный механизм

3) Шагающий способ передвижения:

Шагающий способ передвижения, или шагающая локомоция, предполагает использование роботом «ног» - подвижных конечностей, аналогичных конечностям живых существ. Количество ног может варьироваться от двух (например, у гуманоидных роботов) до шести и более (у инспекционных или биомиметических роботов). Основными преимуществами такого способа передвижения являются способность перемещаться по сложным, неровным и узким поверхностям, возможность обхода препятствий без необходимости их преодоления, а также более "живое" и адаптивное поведение. Однако шагающие роботы отличаются высокой сложностью конструкции и алгоритмов управления, требуют точного поддержания баланса и устойчивости, обладают повышенной потребностью в вычислительных ресурсах и, как правило, имеют более низкую скорость передвижения по сравнению с колёсными аналогами.

Рисунок 1.4 – Шагающий способ передвижения на примере робота Laikago

4) Воздушный способ передвижения:

Воздушный способ передвижения применяется в летающих роботах, таких как беспилотные летательные аппараты (БПЛА), дроны, ракеты, самолёты и вертолёты с автопилотами. Основной принцип их движения заключается в использовании подъёмной силы, создаваемой винтами или аэродинамическими поверхностями. К основным преимуществам воздушной локомоции относятся возможность быстрого перемещения по воздуху, доступ к труднодоступным объектам, таким как крыши, высотные здания и труднопроходимые зоны, а также широкое применение в картографии, наблюдении и мониторинге. Однако этот способ передвижения имеет и ряд недостатков: ограниченное время полёта из-за высокой энергоёмкости, зависимость от погодных условий (например, ветер или дождь), сложность стабилизации в полёте и риск аварий или падений.

Рисунок 1.5 - БПЛА

5) Плавающий способ передвижения:

Плавающий способ передвижения применяется для роботов, функционирующих в водной среде — как на поверхности, так и под водой. Для движения используются гребные винты, водомёты, силы водоизмещения, а также биоинспирированные механизмы, имитирующие движение плавниками. Среди основных преимуществ такого типа локомоции можно выделить возможность работы в водной среде, адаптацию к как надводным, так и подводным условиям, а также незаменимость в задачах мониторинга водоёмов, проведения экологических исследований и участия в спасательных операциях. Вместе с тем плавание предъявляет и определённые требования: необходима герметичная защита электроники, управление осложняется воздействием течений и плотности воды, а также наблюдается ограниченная грузоподъёмность подобных устройств[3].

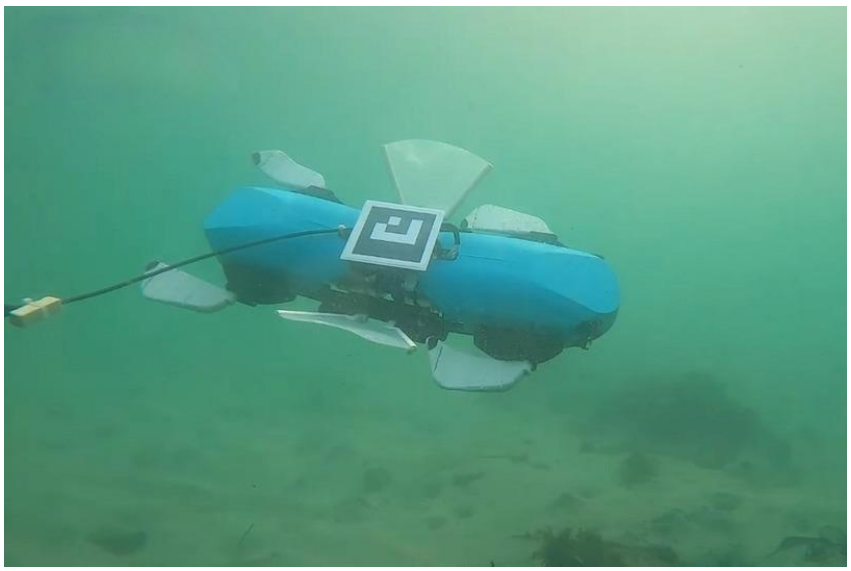


Рисунок 1.6 – Передвижение подводного робота с помощью искусственных ласт

Таким образом, выбор способа передвижения робота напрямую зависит от условий эксплуатации, требование к проходимости, маневренности и устойчивости. Каждый тип обладает уникальными преимуществами и ограничениями, но именно правильное сочетание конструктивных решений для конкретной сферы применения позволяет создать эффективную и надежную роботизированную систему.

1.2 Цель исследования и полезность в решении проблем

Актуальность данной темы обусловлена тем что в мире идет большой спрос на мобильные роботы, которые могут работать на простых функциях. К этому и относиться робот который может двигаться с помощью распознавание цвет объекта. К примеру в современных логистических и транспортных системах а также в производстве и складах возрастает потребность в мобильных роботах, способных эффективно ориентироваться и выполнять задачи в условиях динамически изменяющейся среды. И для достижение этой цели в программной обеспечении основной библиотекой является OpenCV.

Использование библиотеки OpenCV в сочетании с другими программными средствами позволяет адаптировать системы для различных сфер деятельности. Также можно использовать робота в научных и образовательных целях. К примеру:

- LEGO Mindstorms, SPIKE Prime позволяют программировать робота, чтобы он реагировал на цвета с помощью встроенных цветовых сенсоров или камер. Можно настроить его, чтобы он двигался в сторону определённого цвета.
- Также можно создать робота для автоматизированного сбора спелых овощей и фруктов, определяя степень зрелости по яркости и насыщенности цвета плодов. Для этого необходимо реализовать систему компьютерного зрения с настройкой параметров цветового анализа, а также снабдить робота манипулятором для захвата и аккуратного отделения урожая.
- Еще одним интересным использованием библиотеки Open.cv является моделирование биоинспирированного поведения, робот повторяет поведение животного или другого объекта ориентируясь на цветовые метки.

Перечисленные образовательные и научные направления находят своё подтверждение в реальных проектах и экспериментах. Далее рассмотрим конкретные примеры и научные исследования, в которых применялись мобильные роботы и технологии компьютерного зрения. Реальные примеры и исследования:

- В компании Boston dynamics есть робот Spot, это хорошо известный мобильный робот, похожий на собаку. Он умеет ходить, обходить препятствия, держать равновесие и выполнять задания. По умолчанию Spot не настроен ходить за цветными объектами, но у него есть камера и SDK что значит, разработчики и исследователи могут написать собственную программу, чтобы робот реагировал на определенные цвета.[6]

Рисунок 1.7 Spot, мобильный робот с возможностью адаптации через SDK.

• Одним из ярких еще одних примеров является Hexapod-роботы с цветовым зрением. Это роботы с шестью ногами, которые могут двигаться по различным поверхностям, обходить препятствия и сохранять устойчивость в сложных условиях. В университетских лабораториях такие роботы используются для изучения: Биомеханики движения – моделируются принципы ходьбы насекомых или паукообразных; Компьютерного зрения – в том числе распознавания объектов по цвету; когнитивной робототехники – где робот должен самостоятельно принимать решения на основе сенсорных данных.

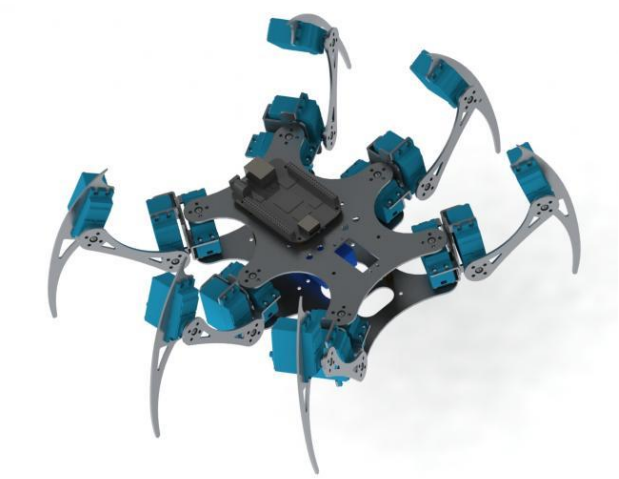


Рисунок 1.8 – Hexapod

Компьютерное зрение стремительно развивается благодаря усовершенствованиям алгоритмов ИИ, повышению вычислительной мощности и распространению данных. Улучшенные модели машинного обучения позволяют более точно распознавать и обрабатывать изображения . Растущий спрос на автоматизацию и интеллектуальные технологии в различных секторах, включая здравоохранение, автомобилестроение и розничную торговлю, стимулирует дальнейшее развитие.[5]

Одним из основание выбора темы, связано с быстрорастущим рынком Искусственного интеллекта. В данное время Искусственный интеллект используется в разных сферах деятельности. Взяв это как основным из причин я хотела провести работу в сфере изучения компьютерного зрения.

Компьютерное зрение также имеет хороший прогноз в большом рынке. По анализам компании Grand View Research объем мирового рынка компьютерного зрения оценивается в 19,82 млрд долларов США в 2024 году и, по прогнозам, будет расти в среднем на 19,8% в период с 2025 по 2030 год. Основными движущими факторами роста рынка являются различные факторы, такие как

возросший спрос на автоматизацию в различных отраслях, рост технологий искусственного интеллекта и машинного обучения (МО), достижения в области аппаратного обеспечения и датчиков изображений, а также растущий спрос на автономные транспортные средства. Использование компьютерного зрения в системах наблюдения и безопасности стремительно растет.[7]

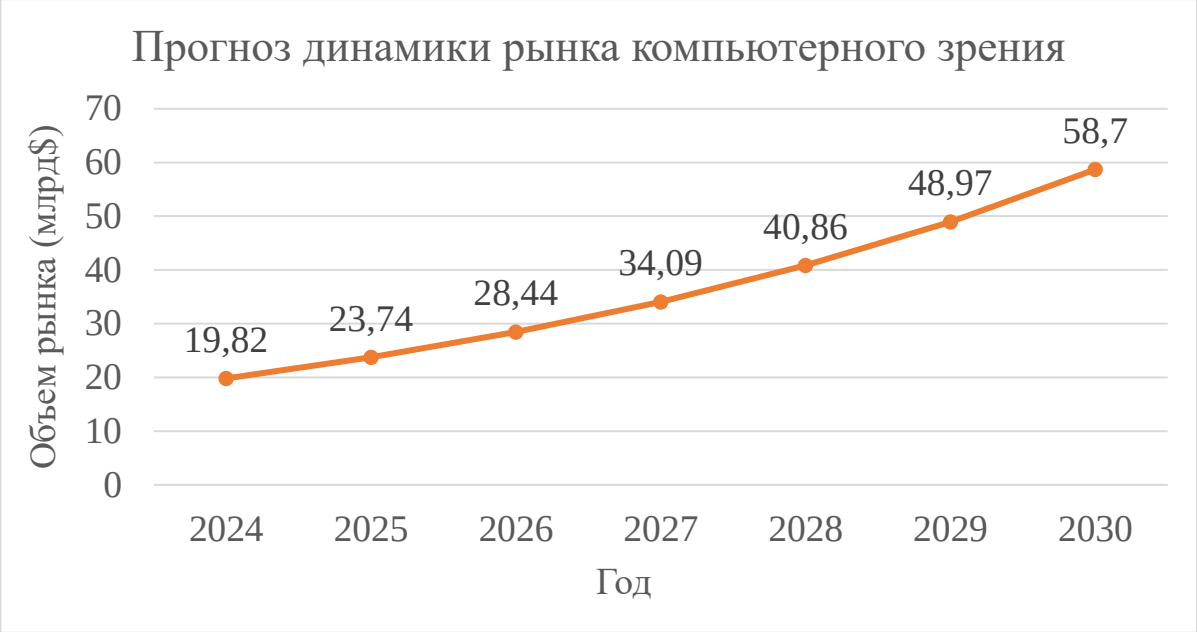


Рисунок 1.9 – График динамики компьютерного рынка

Значение на таблиц были взяты с расчета через формулу сложного процента[17]. Формула сложного роста широко используются в экономики, инвестициях а также в нашем случае для расчета прогнозировании динамики рынка компьютерного зрения.

Формула вычисления сложных процентов:

$$B = A * \left(1 + \frac{P}{100\%}\right)^n \quad (1)[17]$$

где:

B – будущая стоимость(в нашем случае будущий прогноз роста на 2030 год);

A – текущая стоимость;

P – процент нарастания рынка;

n – количество расчетных периодов.

Прогнозируемый рост мирового рынка компьютерного зрения на 19,8% ежегодно свидетельствует о стремительном развитии этой технологии и ее

возрастающей роли в различных сферах. Уже к 2030 году объем рынка может достичь $58,7 млрд, что почти в 3 раза превысит показатели 2024 года.

Такой значительный рост обусловлен:

– Развитием аппаратных технологий (CMOS-датчики, камеры, процессоры), обеспечивающих более точную и быструю обработку изображений.

– Расширением областей применения – от автономного транспорта и распознавания лиц до AR/VR и промышленной автоматизации.

– Повышением спроса на интеллектуальные системы в ритейле, медицине, безопасности и других отраслях.

Ожидается, что компьютерное зрение продолжит трансформировать бизнес-процессы и повседневную жизнь, открывая новые возможности для автоматизации и анализа данных. Этот рынок остается одним из самых перспективных в сфере искусственного интеллекта и цифровых технологий.

1.3 Определение компьютерного зрения

С покорением новых вершин человеком перед ним открывается множество путей, а вместе с ними и проблемы, которые нужно решать. Одной из таких проблем является автоматизация и роботизация технологии. Ранее считалось что техника может работать только по заданной логике или исключительно под руководствам человека. Однако современные технологии достигли уровня, на которым системы способны самостоятельно принимать решение и обучаться. Одним из главных средств в робототехнике стало анализ окружающей среды и принятие решение на его основе. Восприятие информации роботом может основываться на датчиках и сенсорах, которые обеспечивают высокую точность измерений и скорость обработки данных. Если же требуется навигация и распознавание объектов, то применяются камеры вместе с искусственным интеллектом (AI). Но также есть наилучший вариант это комбинированный подход, обеспечивающий более полную информацию об окружающей среды.

Мы хорошо знакомы с принципами работы датчиков и сенсоров: они позволяют точно задавать параметры и прокладывать ходы для достижение цели. Однако с камерами ситуация обстоит иначе. Чтобы робот мог распознавать перед собой объекты, камеру нужно обучить, настроить, обработать данные, оптимизировать алгоритмы и запустить систему. Таким образом, остановимся на компьютерном зрении как одним из важнейших направлений в робототехнике.

Компьютерное зрение – это область искусственного интеллекта, которая использует машинное обучение и нейронный сети для обучения компьютеров и систем извлекать значимую информацию из цифровых изображений, видео и

других визуальных данных, а также давать рекомендации или предпринимать действия при обнаружении дефектов или проблем.[9]

Компьютерное зрение выполняет функции, схожие с человеческим зрением, но у людей есть ключевое преимущество – жизненный опыт и способность воспринимать контекст. Это помогает им легко распознавать объекты**,** определять расстояния, отслеживать движение и замечать отклонения на изображении. В отличие от человека, компьютерное зрение использует алгоритмы, камеры и данные, а не биологические механизмы, такие как сетчатка или зрительные нервы. Машины, запрограммированные на анализ продукции или контроль промышленных процессов, способны обрабатывать тысячи объектов в минуту, выявляя мельчайшие дефекты и несоответствия, которые сложно заметить невооруженным глазом. Если коротко сказать, ИИ позволяет компьютерам думать, то компьютерное зрение позволяет им видеть, наблюдать и понимать.

Сегодня компьютерное зрение применяется во многих сферах от энергетики и промышленности до автомобильной отрасли. С каждым годом его использование расширяется, а рынок стремительно развивается.

Краткая история компьютерного зрение:

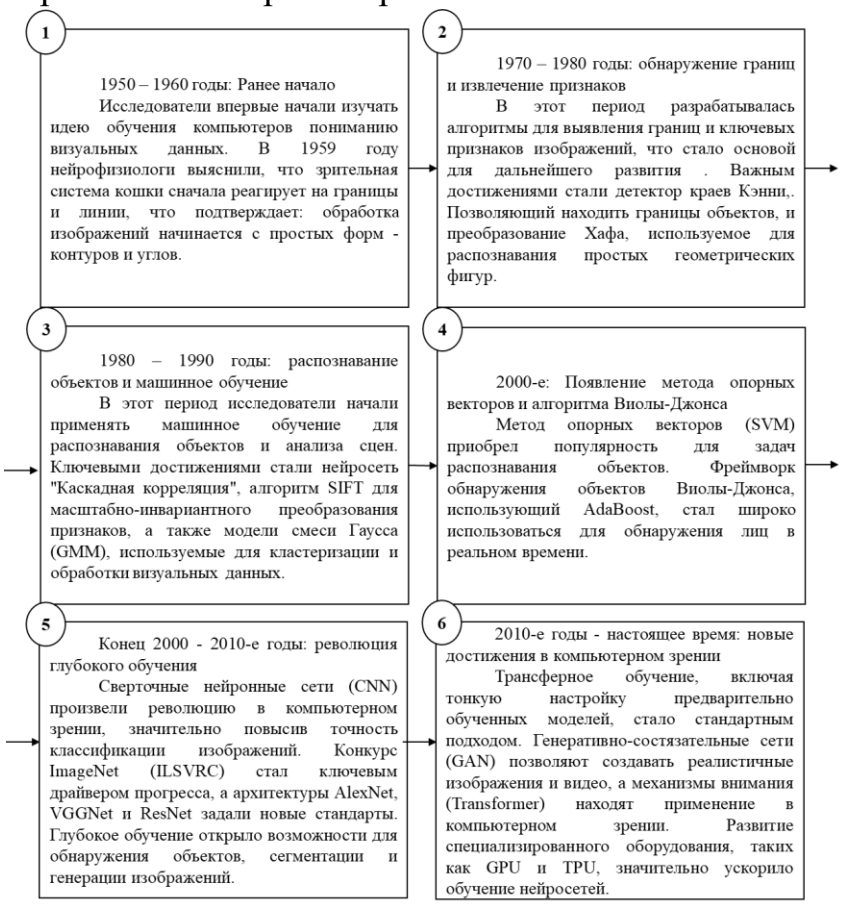


Рисунок 1.10 – Развитие компьютерного зрения

1.4 Основные алгоритмы обработки изображения

Обработка изображений - это процесс улучшения и извлечения полезной информации из изображений. Изображения рассматриваются как двумерные сигналы, а входные данные для этого процесса - это фотография или видео. Входные данные – это изображение, а выходные данные могут быть улучшенным изображением или характеристиками/особенностями, связанными с ним.[10]

Для начало давайте поймем различия между компьютерным зрением и обработкой изображений. Обработка изображений это визуальное совершенство. Основная целью является улучшение качество изображения. Основными терминами можно назвать усиление контрастности, настройка цветов или сглаживание краев, фокус делается на том, чтобы сделать изображение более привлекательным или пригодным для дальнейшего использования. Обработка изображений фокусируется на улучшении и преобразовании изображений. Это жизненно важно в таких областях, как цифровая фотография для цветокоррекции, медицинская визуализация для более четких сканов и графический дизайн для создания потрясающих визуальных эффектов. Эти преобразования не только улучшают эстетику, но и делают изображения более подходящими для анализа, закладывая основу для более глубокой интерпретации, в том числе с помощью систем компьютерного зрения.

С другой стороны, компьютерное зрение стремится извлекать смысл из изображений. Цель состоит не в том, чтобы изменить внешний вид изображения, а в том, чтобы понять, что оно представляет. Это включает в себя идентификацию объектов, интерпретацию сцен и даже распознавание шаблонов и поведения в пределах изображения. Это больше касается понимания, а не изменения.

Однако обработка изображений и компьютерное зрение тесно взаимосвязаны: первая обеспечивает необходимые инструменты для предварительного анализа, в то время как вторая использует эти данные для более сложных интерпретаций и принятия решений.

Например, многие алгоритмы компьютерного зрения требуют предварительно обработанных изображений. Такие методы, как шумоподавление и повышение контрастности, применяемые на этапе обработки изображений, существенно повышают точность распознавания объектов и других задач компьютерного зрения. Кроме того, упрощённые или улучшенные изображения, полученные после предварительной обработки, значительно облегчают извлечение признаков, что делает их более понятными для алгоритмов анализа и интерпретации.[11]

В дипломной работе чтобы робот мог распознавать цветные объекты были проделаны такие алгоритмы как сегментация, выделения объектов по цвету, RGB в open.cv как BGR перевод кадра в HSV, контура , цветовая модель HSV, моменты изображении.

Цветовая модель RGB представляет собой систему, в которой цвета описываются с помощью трёх основных компонентов: красного (R), зелёного (G) и синего (B). Эта модель основана на аддитивном принципе смешения цветов, при котором разные цвета получаются за счёт комбинирования различной интенсивности света трёх указанных цветов. При равномерном смешивании красного, зелёного и синего света с максимальной интенсивностью возникает белый цвет. Если интенсивность всех трёх каналов равна нулю, получается чёрный цвет, то есть отсутствие света. Каждый из каналов – красный, зелёный и синий – обычно задаётся 8-битным числом, что позволяет задавать значения от 0 до 255. Значение 0 означает полное отсутствие данного цвета, а 255 – его максимальную интенсивность. Таким образом, цвет в RGB представляется в виде кортежа или триплета: (R, G, B). Различные цвета создаются путем смешивания различных интенсивностей трех основных цветов. Красный (255, 0, 0) – чистый красный. Зеленый (0, 255, 0) – чистый зеленый. Синий (0, 0, 255) – чистый синий. При смешивании двух или более цветов образуются новые цвета. Белый (255, 255, 255) – это комбинация красного, зеленого и синего при полной интенсивности.

Цветовая модель RGB часто визуализируется в виде трехмерного цветового куба , где каждая ось представляет один из трех основных цветов: красный (R) , зеленый (G) и синий (B). Куб демонстрирует, как сочетания красного, зелёного и синего цветов с разной интенсивностью позволяют получить широкий диапазон оттенков. Разбор концепции куба RGB и то, как оси и координаты представляют цвета:

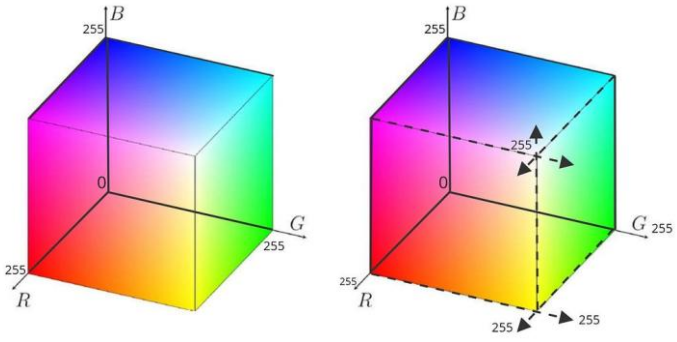


Рисунок 1.11 – Наглядный пример куба RGB

В цветовом пространстве RGB каждая ось представляет один из основных цветов: красный, зеленый и синий. Значения по этим осям находятся в диапазоне от 0 до 255 , где 0 представляет отсутствие интенсивности этого цвета (отсутствие света). 255 представляет полную интенсивность этого цвета (максимальная яркость).

Каждый угол куба соответствует определенной цветовой комбинации.

- (0, 0, 0) — черный угол, означающий отсутствие света (отсутствие красного, зеленого или синего).
- (255, 255, 255) — белый угол, то есть полная интенсивность всех трех цветов объединяется, в результате чего получается белый свет.
- Диагональная ось, проходящая от (0, 0, 0) до (255, 255, 255), представляет собой смесь возрастающей интенсивности по всем трем каналам, что приводит к диапазону от черного до белого .

Грани куба отображают сочетания двух основных цветов при их полной интенсивности, тогда как третий цвет при этом равен нулю:

- Красный (255, 0, 0) находится в углу, где красный канал имеет максимальную интенсивность, а зеленый и синий отсутствуют.
- Зеленый (0, 255, 0) находится в углу, где зеленый канал имеет максимальную интенсивность, а красный и синий отсутствуют.
- Синий (0, 0, 255) находится в углу, где синий канал имеет максимальную интенсивность, а красный и зеленый отсутствуют.

Грани куба представляют собой комбинации двух основных цветов полной интенсивности, создающие вторичные цвета:

- Желтый (255, 255, 0) : Смесь красного и зеленого при полной интенсивности, дающая желтый цвет. Синий компонент отсутствует (0).
- Голубой (0, 255, 255) : Смесь зеленого и синего при полной интенсивности, дающая голубой. Красный компонент отсутствует (0).
- Маджента (255, 0, 255) : Смесь красного и синего при полной интенсивности, дающая в результате маджента. Зеленый компонент отсутствует (0).

Точки внутри куба представляют третичные цвета , которые являются комбинациями различных уровней интенсивности красного , зеленого и синего . По мере удаления от осей и по направлению к внутренней части куба можно увидеть полный спектр цветов, создаваемых различными комбинациями этих трех каналов.

Цветовая модель HSV (Hue, Saturation, Value) является альтернативой цветовой модели RGB, разработанной для большего соответствия тому, как люди воспринимают и классифицируют цвета. В то время как RGB использует

интенсивность красного, зеленого и синего для определения цвета, HSV разделяет определение цвета на три компонента:

1. Оттенок (H) : представляет тип цвета и является углом на цветовом круге. Он определяет семейство цветов (например, красный, синий, зеленый, желтый и т. д.). В модели HSV оттенок представлен в виде градуса на цветовом круге в диапазоне от 0° до 360°. 0° представляет красный, 120° представляет зеленый, 240° представляет синий, промежуточные значения создают другие цвета (желтый, голубой, пурпурный и т. д.).

2. Насыщенность (S) : представляет интенсивность или чистоту цвета. Значение насыщенности 0% означает, что цвет полностью ненасыщенный (серый), а 100% означает, что цвет полностью насыщен (чистый цвет без примеси серого). По мере увеличения насыщенности цвет становится более ярким.

3. Значение (V) : Это представляет яркость цвета. Диапазон значений от 0% до 100% :

- 0% означает, что цвет полностью черный , независимо от оттенка и насыщенности.
- 100% означает, что цвет имеет полную яркость , без добавления затемнения.

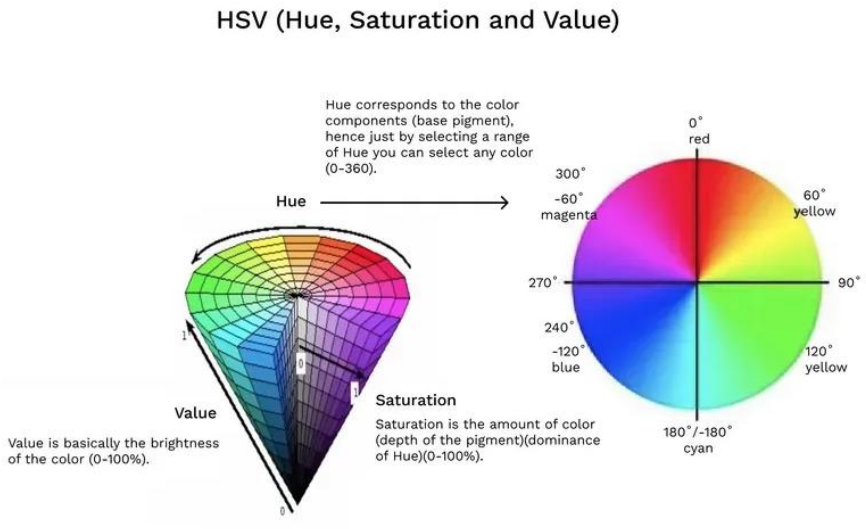


Рисунок 1.12 – Цветовая модель HSV

На изображении выше обратите внимание на цилиндрическую форму, где:

- Оттенок (H) представлен в виде угла вокруг центральной вертикальной оси (подобно спицам колеса).

- Насыщенность (S) представлена расстоянием от центра цилиндра (центр соответствует отсутствию насыщенности или серому цвету, а край соответствует полной насыщенности).
- Значение (V) представлено высотой цилиндра от низа (темного) до верха (светлого).

HSV проще для понимания, так как описывает цвет как в жизни: оттенок (цвет), насыщенность (интенсивность) и яркость. Позволяет легко менять отдельные параметры, поэтому популярен в графике.

В OpenCV цветовая модель HSV представлена в несколько ином диапазоне, чем типичные 0°–360° для Hue и 0–100% для Saturation и Value . Вот как OpenCV обрабатывает диапазон HSV:

1) Оттенок (H): Диапазон : от 0 до 179 (вместо 0°–360°)

OpenCV использует 8-битные значения для представления канала оттенка, поэтому его максимальное значение составляет 255. Чтобы представить полный диапазон оттенков 360°, диапазон в OpenCV уменьшается до 0–179 , что соответствует цветовому кругу 180°.

– 0 : Красный
– 30 : Желтый
– 60 : Зеленый
– 120 : Голубой
– 150 : Синий
– 180 : Красный (снова, но оттенок меняется на 180)

2) Насыщенность (S): Диапазон : от 0 до 255

Насыщенность в OpenCV представлена как 8-битное значение, поэтому она может находиться в диапазоне от 0 (нет цвета, оттенки серого) до 255 (максимальная интенсивность цвета).

– 0 : Нет насыщенности (оттенки серого)
– 255 : Полная насыщенность (чистый цвет, без серого)

3) Значение (B): Диапазон : от 0 до 255

Как и насыщенность, значение (яркость) также представлено в виде 8-битного значения, варьирующегося от 0 (черный) до 255 (максимальная яркость).

– 0 : Черный (без яркости)
– 255 : Полная яркость (чистый цвет)[12].

Также одним из важнейших вещей в компьютерном зрении является момент изображения. В компьютерном зрении и обработке изображений моменты изображения часто используются для характеристики формы объекта на изображении. Эти моменты фиксируют основную информацию, такую как

площадь объекта, координаты его центра масс другими словами центроид, ориентация и другие геометрические свойства [13]. Общая формула момента:

$$M_{ij} = \sum_x \sum_y x^i * y^j * I(x, y) \quad (2)$$

где:

M_{ij} – момент порядка $i+j$,

x, y – координаты пикселей,

$I(x,y)$ – это значения изображения в точке (x, y), в бинарном изображении которые мы используем значение 0 или 1.

i, j – порядки по осям x и y.

Формула центра масс:

$$\bar{x} = M_{10}/M_{00} \quad \text{и} \quad \bar{y} = M_{01}/M_{00} \quad (3)$$

В работе моменты изображения были использованы чтобы найти центр изображения, это способ узнать где именно в кадре находиться объект, и с помощью этой информаций робот может точно понимать в какую сторону ему нужно следовать.

2 Проектирование мобильного колесного робота с системой слежения

В качестве основы для реализации робота, способного двигаться за цветным объектом, был использованы видеоматериалы с YouTube-канала, в котором продемонстрирована работа с библиотекой Open.CV [14]. На их основе было адаптирована и доработана программа управления.

2.1 Аппаратная часть

- Микрокомпьютер Raspberry pi 4
- Камера Raspberry Pi Board
- Мотор редуктор
- Драйвер L298N
- Литий ионный аккумулятор
- Карты памяти MicroSD
- Провода для подключения
- Полимерный материал

Таблица 2.1 – Компоненты мобильного колесного робота

№	Наименование	Кол-во	Ед. изм.	Изображение
1	Микрокомпьютер Raspberry Pi 4 Model B с 4 ГБ RAM – мощный микрокомпьютер для замены десктопа на Linux. Оснащён процессором Broadcom BCM2711 (4 ядра Cortex-A72, 1,5 ГГц), 4 ГБ RAM, портами USB, Wi-Fi, Bluetooth и гигабитным Ethernet. Поддерживает 4K видео, два монитора. Повышенная производительность на 50% по сравнению с предыдущими версиями, идеально подходит для веб-сёрфинга и работы с видео. 15]	1	шт	

Продолжение таблицы 2.1

2	Raspberry Pi Camera Board - компактная камера для Raspberry Pi, подключающаяся через интерфейс CSI, что снижает нагрузку на процессор. Она идеально подходит для роботов, беспилотников и охранных систем. Камера имеет разрешение 5 Мп, снимает фото до 2592×1944 пикселей и видео в форматах 1080p30, 720p60 и 640×480p с частотой до 90 кадров в секунду. Размеры камеры — 25×20×9 мм, вес 5 г. Использует MMAL и V4L API для интеграции в проекты.[16]	1	шт	
3	Мотор-редуктор RKP-GMP048R1-90D с прямым корпусом и передаточным отношением 1:48 предназначен для легких роботов, моделей, игрушек и механических проектов. Редуктор включает пластиковые шестеренки и приспособление для установки энкодера, что позволяет отслеживать обороты колес в реальном времени и программировать систему для выполнения задач. Для крепления на шасси или платформе имеются два отверстия диаметром 3 мм и одно 1,4 мм в пластиковом корпусе.[17]	4	шт	

Продолжение таблицы 2.1

4	Драйвер L298N – это модуль для управления двигателями постоянного тока и шаговыми двигателями. L298N содержит два H-моста, каждый из которых управляет одним двигателем или обмоткой шагового двигателя. Изменение направления происходит через разъемов логических уровней на IN1/IN2 (для 1-го мотора) или IN3/IN4 (для 2-го) меняет полярность напряжения на моторе . Регулировка скорости ШИМ-сигнал на ENA/ENB уменьшает среднее напряжение, снижая обороты. Также есть разъем для питание логики +5V. [18]	1	шт	
5	Для питание моторов в работе были использованы две литий ионный аккумуляторов по 3.7В. Они обладают высокой энергоёмкостью, малым саморазрядом и отсутствием эффекта памяти, что позволяет заряжать их в любое время без потери ёмкости.	2	шт	
6	Карты памяти MicroSD используем в работе как основной накопитель данных, выполняя роль жесткого диска компьютера	1	шт	

Продолжение таблицы 2.1

7	Филамент – это пластиковая проволока, используются в 3д принтерах	1	кг	
8	Провода	80	см	

При выборе материалов и деталей для будущего робота я руководствовалась соотношением доступности и взаимозаменяемости.

2.2 Схема подключения компонентов

Учитывая вышеперечисленные требования к технологической оснастке и удобству реализации данного мобильного колесного робота, следует всегда обращаться к его главной полезной функций. Ниже приведена функциональная схема мобильного колесного робота способного распознавать цвета в видимом спектре.

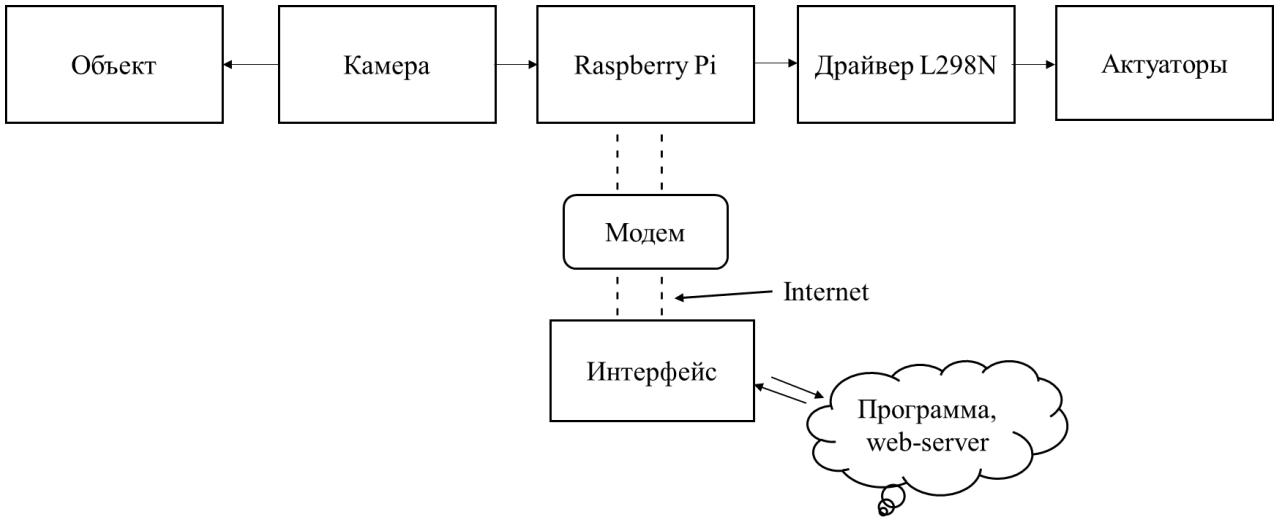


Рисунок 2.1 – Функциональная схема системы распознавания цветных объектов на борту мобильного колесного робота

Кроме функциональной схемы мобильного робота было создана принципиальная схема подключение для работы движение робота с

использованием цветных объектов было выполнено в рабочей среде “Fritzing”. Принципиальная схема собрано в следующем порядке:

Для начало работы чтобы схема работала нужно правильно подобрать зарядное устройство. В работе были использованы литий ионные аккумуляторы подключенные последовательно, для питание моторов нужно 3V-6V, но так используются четыре моторов разом и они работают через драйвер L298N то и напряжение должно быть соответствующим, которое приводят к движению моторов. В последовательном соединении элементов электрической цепи напряжения на каждом из них складываются. Это явление описывается вторым законом Кирхгофа, согласно которому алгебраическая сумма напряжений в замкнутом контуре равна нулю.

$$\sum V = 0 \tag{4}$$

$$V_{\text{общ}} = V_1 + V_2 + \cdots + V_n \tag{5}$$

где: V – напряжение электрической цепи.

Двигатели подключены по две к одному из выходных каналов драйвера l298N: OUT1 и OUT2 – для первого и второго мотора(к примеру с левой стороны робота передний и задний), OUT3 и OUT4 – для третьего и четвертого(также, но с правой стороны).

Сам драйвер позволяет управлять направлением вращения и скоростью моторов с помощью сигналов, поступающих от Raspberry pi через входы IN1, IN2, IN3 и IN4.

Основным управляющим элементом в работе является Raspberry pi. Питание берется от power bank, которая при выходе дает 5V-3A что нужно для питание Raspberry pi. У Raspberry Pi есть множество GPIO-пинов, через которые можно подключать датчики, моторы, светодиоды, сервоприводы и другие устройства. Подключение элементов к Raspberry pi:

– К GPIO-пинам 17,18, 22, 23 были подключены входы IN1, IN2, IN3 и IN4 которые предназначены для управления двигателями.

– К входу 5V был подключен VCC питание сервопривода.

– Через CSI-разъем было подключена камера Raspberry pi с помощью плоского шлейфа (FFC/FPC).

Компоненты были подключены к общей земле что берется от аккумулятора, чтобы в электрической цепи имели одну точку отсчета напряжение. Это необходимо, чтобы устройства корректно передавали сигналы

и понимали друг друга. Если не соединить землю у разных источников питания или устройств, сигналы могут искажаться как во многих случаях или вообще не работать, а также возрастет риск повреждения компонентов.

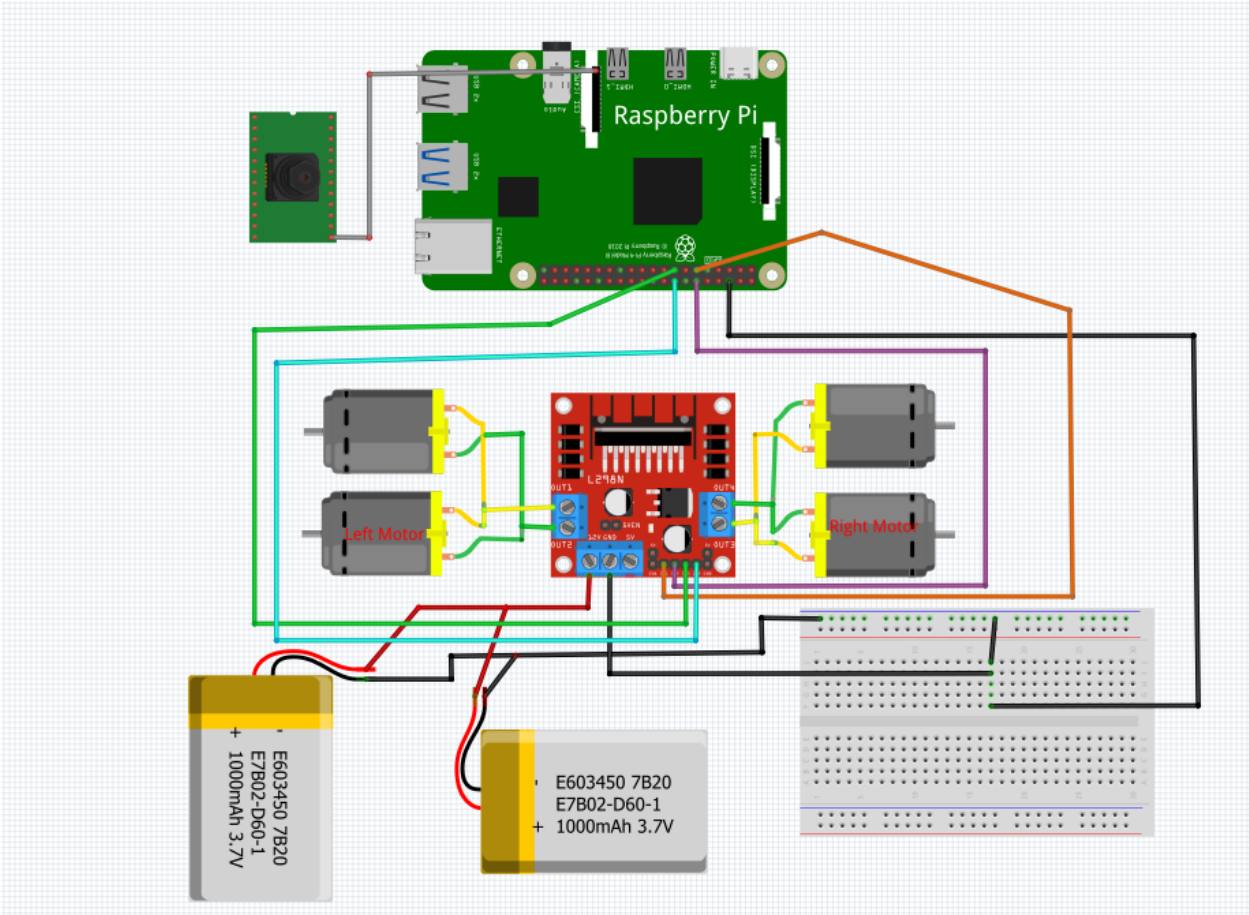


Рисунок 2.2 – Принципиальная схема подключения компонентов робота

Вся система питается от аккумуляторных батареи типа Li-ION 18750 емкостью 5В. Передача сигналов между роботом и веб сервером производиться по протоколу Wi-Fi 802.11n. Камера снимает в видимом спектре – это диапазон от 400 до 700 нанометров. Диапазон соответствует тому, что видит человеческий глаз, от фиолетового до красного света.

2.3 Инструкция по сборке и наладке системы

Raspberry pi – это одноплатный компьютер, который имеет свою операционную систему Raspberry Pi OS также можно запускать и другие операционные системы (например, Linux), и он будет работать как обычный ПК. Используется в проектах робототехники, автоматизации, обучению

программированию и др. [12] У Raspberry pi есть разные порты и пины что расширяет его возможности. К ним относиться:

Подготовка к работе Raspberry Pi:

1) Сперва нужно установить Raspberry Pi OS: для этого мы заходим в официальный сайт raspberrypi.com где вкладках программное обеспечение есть установка Raspberry Pi OS с помощью Raspberry Pi Imager, выбираем установку смотря на свой ПК.

2) Далее нам нужен микро SD карта для загрузки. Формат микро SD карты должна быть FAT32, если формат другой то форматируем карту для установки. Память микро SD карты должна быть от 8ГБ до 32ГБ желательно, но также можно взять карту с большой памятью. Вставляем микро SD карту ПК для установки OS.

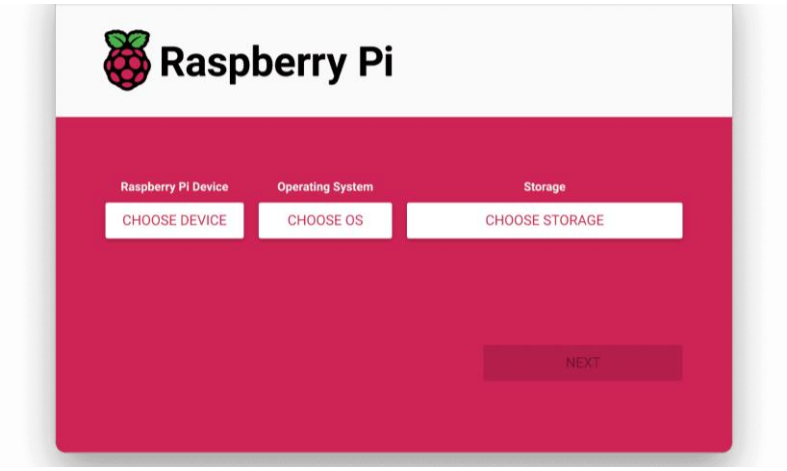


Рисунок 2.3 Установочное окно Raspberry Pi

3) Выбираем вид Raspberry Pi, OS также установочное место. В нашем случае микро SD карту.

4) После установки OS, вставляем карту в Raspberry Pi. Подключаем Raspberry Pi к питанию а также через HDMI порт соединяем с монитором.

5) Проходим регистрацию и первичную настройку устройства. Это выбор языка, часовой пояс, подключение к Wi-Fi, установка обновления и в самом конце перезагрузка.

6) Настройка для удаленного управление. Для этого через подключенный монитор в терминале запускаем “sudo raspi-config” в открывшейся вкладке проходим →Interface Options → SSH→Enable. В этой же вкладке подключаем и другие интерфейсы которые пригодиться в работе.

Затем через терминал узнаем IP-адрес Raspberry Pi:

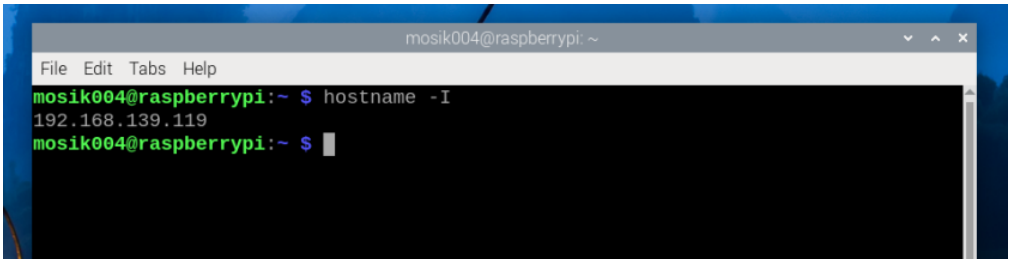


Рисунок 2.4 Терминал Raspberry Pi

Если IP-адрес выходит правильно, то нам нужно скачать приложение RealVNC Viewer для удаленного управления. В приложении нас встречает окно поиск доступных устройств. Туда и вставляем IP-адрес:

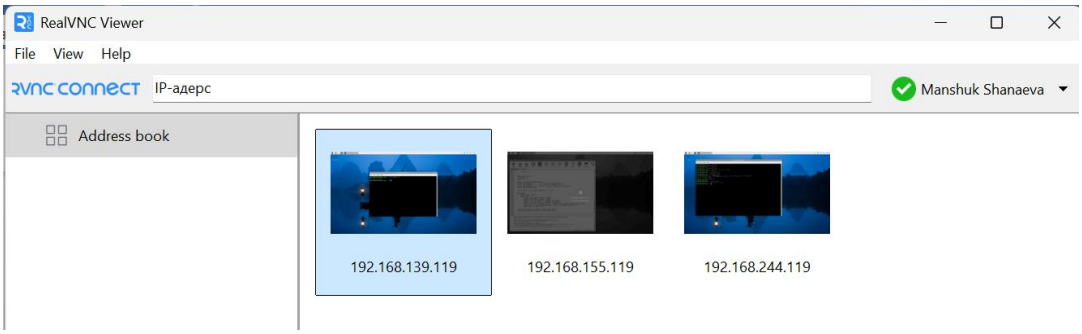


Рисунок 2.5 Вход в RealVNC Viewer через поиск нужного IP-адреса

После того как приложение нашло IP-адрес Raspberry Pi мы должны ввести свой пароль для подтверждение:

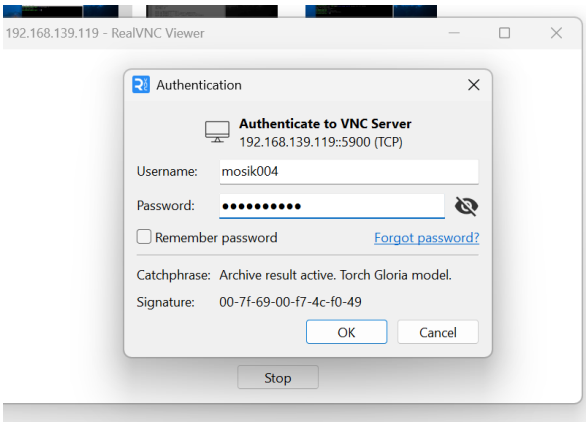


Рисунок 2.6 Подтверждение IP-адреса

На этом Raspberry Pi готов к работе:

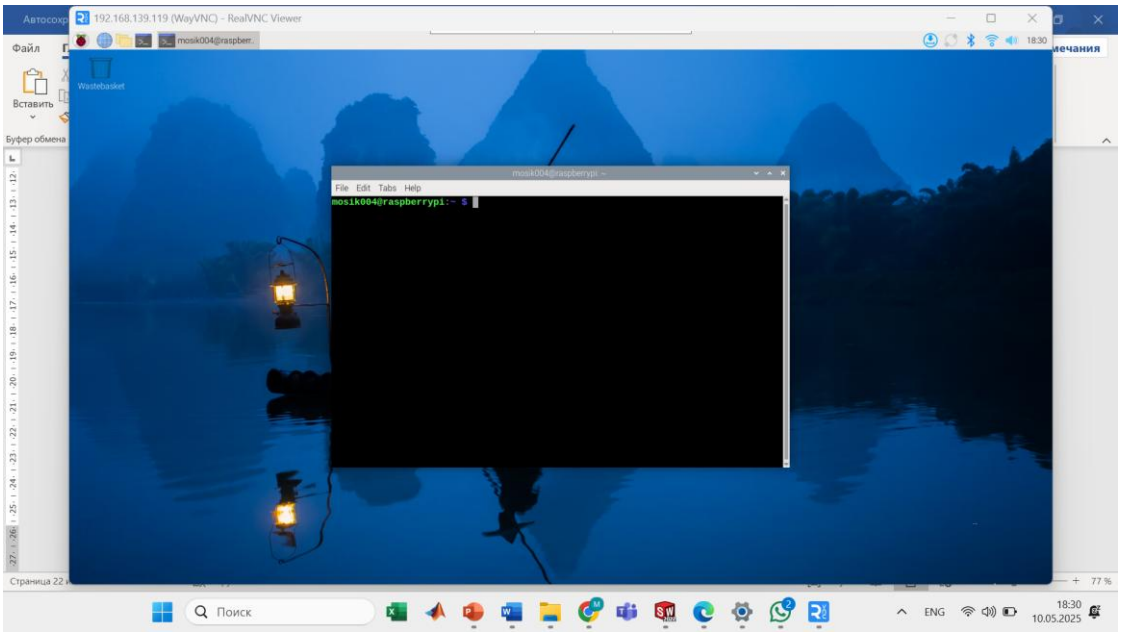


Рисунок 2.7 Главное окно RealVNC Viewer во время сеанса подключения

2.4 Алгоритм кода

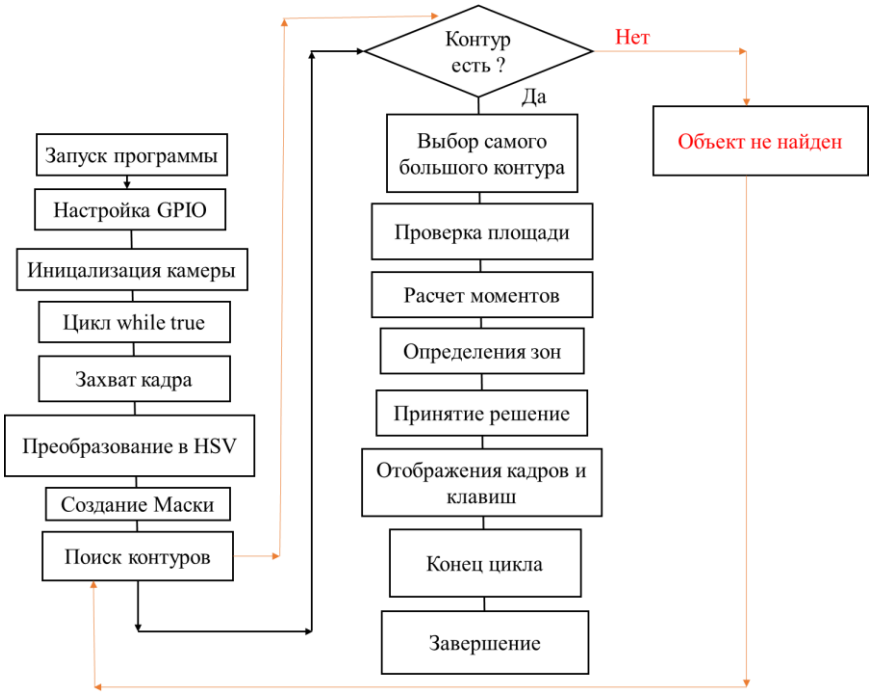


Рисунок 2.8 Алгоритм работы кода

Разработка программного кода является ключевым этапом автоматизации работы робота в рамках дипломного проекта. С помощью написания кода задаётся логика выполнения определённых действий роботом. Код был написан в приложении RealVNC Viewer через запуск Thonny, язык кода которым является Python.

Перед началом работы нужно установить нужные библиотеки:

Таблица 2.2 Библиотеки, применённые в программной части

№	Библиотека	Назначение	Установка
1	cv2 (OpenCV)	Обработки изображения с камеры, преобразования изображения в HSV, нахождения контуров объекта, отображения окна с видео (imshow), отрисовки прямоугольника и текста на кадре.	“sudo apt install python3-opencv” или “pip install opencv-python”
2	RPi.GPIO	Позволяет управлять GPIO-выводами Raspberry Pi. В этом коде используется для управления мотором через драйвер (например, L298N). Команды GPIO.output(...) включают и выключают выводы.	sudo apt install python3-rpi.gpio
3	numpy	Используется для создания и работы с массивами – в частности, для задания нижней и верхней границы цвета в HSV (np.array([H, S, V])).	“sudo apt install python3-numpy” или “pip install numpy”
4	time (встроенная библиотека)	Используется для задержек: • при запуске камеры (sleep(2) – прогрев); • при управлении скоростью обновления (sleep(0.1) в цикле).	Не требуется

Продолжение таблицы 2.2

5	picamera2	Это новая библиотека для управления CSI-камерой (шлейфной) Raspberry Pi. Она заменяет старую picamera и работает с системой libcamera. В коде используется для: инициализации камеры (Picamera2()), настройки разрешения и формата, захвата кадров (capture_array()).	“sudo apt install python3-picamera2” или “sudo apt install libcamera-apps”

Для подключение внешних библиотек в Python используется import. Эти библотеки содержат готовые функции, классы и инструменты, которые можно использовать в коде. И каждой библиотеке в строке нужно записывать отдельный import.

```
import cv2
import RPi.GPIO as GPIO
import numpy as np
import time
from picamera2 import Picamera2
```

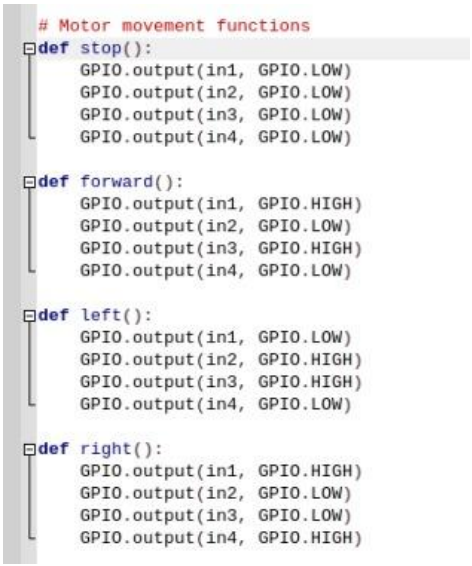
Рисунок 2.9 Подключение библиотек

На рисунке 2.9 инициализируем переменные in1, in2, in3, in4 в которых указаны номера GPIO-пинов Raspberry Pi. Они используются для подключения моторов через драйвер L298N. С помощью которых мы направляем робота.

```
# Motor pins
in1 = 17
in2 = 18
in3 = 22
in4 = 23
```

Рисунок 2.10 – Инициализация GPIO-пинов

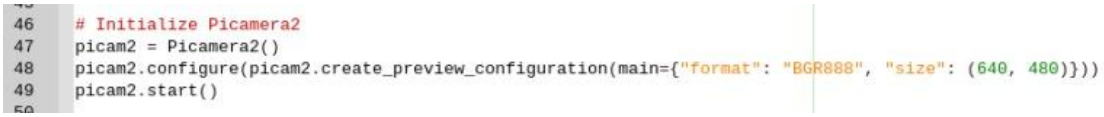
После этого у нас идет функция управления движением робота с помощью функции def. Функция – это блок кода который выполняет действие и может вызываться многократно: def имя_функции(): (действия, которые выполняет функция).



```
# Motor movement functions
def stop():
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in4, GPIO.LOW)

def forward():
    GPIO.output(in1, GPIO.HIGH)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in3, GPIO.HIGH)
    GPIO.output(in4, GPIO.LOW)

def left():
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in2, GPIO.HIGH)
    GPIO.output(in3, GPIO.HIGH)
    GPIO.output(in4, GPIO.LOW)

def right():
    GPIO.output(in1, GPIO.HIGH)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in4, GPIO.HIGH)
```

Рисунок 2.11 – Код функции для управления роботом

Чтобы робот начал ходить мы используем камеру как основной элемент в работе. Сперва для этого нужно сделать Запуск камеры Raspberry Pi с настройкой разрешения и формата кадра.



```
46  # Initialize Picamera2
47  picam2 = Picamera2()
48  picam2.configure(picam2.create_preview_configuration(main={"format": "BGR888", "size": (640, 480)}))
49  picam2.start()
```

Рисунок 2.12 – Запуск камеры

Так как робот движется с помощью распознавание цветных объектов то для этого нужно настроить цветовой диапазон HSV. В коде написано три диапазона три диапазона HSV для разных оттенков выбранного цвета, которые можно переключать во время работы. Что делает этот фрагмент с рисунка :

1. Создаёт список blue_ranges — несколько диапазонов HSV, в которых может находиться цвет синего объекта.

2. Выбирает один из диапазонов по индексу range_index = 0, то есть первый из списка.

3. Присваивает выбранный диапазон переменным lower_blue и upper_blue.

В коде time.sleep(2) используем как время на запуск это является важным, для корректной работы робота.

```
# Initialize Picamera2
picam2 = Picamera2()
picam2.configure(picam2.create_preview_configuration(main={"format": "BGR888", "size": (640, 480)}))
picam2.start()

# Three different blue HSV ranges
blue_ranges = [
    (np.array([5, 150, 150]), np.array([15, 255, 255])),    # Normal Blue
    (np.array([0, 100, 100]), np.array([25, 255, 255])),    # Wide range
    (np.array([10, 100, 100]), np.array([25, 255, 255]))    # Darker Blue
]

# Choose which range to use
range_index = 0   # 0, 1, or 2
lower_blue, upper_blue = blue_ranges[range_index]

time.sleep(2)  # Camera warm-up
```

Рисунок 2.13 – Диапазон нужного цвета для движение робота

Главный цикл кода начинается с try, try: начинает блок кода, где программа может выполнять операции, которые могут вызвать исключения, например, если камера перестанет работать. После этого идет while True, while True: создаёт бесконечный цикл, который продолжает выполняться до тех пор, пока не произойдёт какое-либо завершение. Захват кадра с камеры происходит с помощью frame = picam2.capture_array(). Здесь вызывается метод capture_array() библиотеки picamera2, который захватывает текущий кадр с камеры и сохраняет его в переменную frame. hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) эта строка преобразует изображение frame из цветной модели BGR в модель HSV. HSV используем так как он более удобен для обработки в задачах компьютерного зрения и лучше соответствует восприятию цвета человеком. После того как изображение с камеры получено и преобразовано в цветовое HSV, следующим шагом является выделение области изображения, где присутствует синий цвет. Это делается с помощью функции cv2.inRange(), которая создаёт маску — чёрно-белое изображение, где белым цветом отмечены пиксели, попадающие в заданный диапазон синего цвета, а все остальные пиксели становятся чёрными. Здесь hsv — изображение в HSV-формате, а lower_blue и upper_blue — границы диапазона синего цвета. Все пиксели, значения которых попадают в этот диапазон, становятся белыми (255) на маске, а остальные — чёрными (0). На основе положения синего объекта в кадре, программа проверяет есть ли контуры на маске синего цвета, если находиться хоть один синий объект или даже больше камера выбирает самый большой контур, но сначала проверяется достаточно ли крупный объект перед камерой. С функцией определения, где находиться объект на кадре, у найденного объекта берется граница прямоугольника, с помощью которого легче понять за каким объектом следует робот. После этого происходит выбор действия. Программа

принимает решение в зависимости от того, где находиться центр синего объекта и движется. Если объект слишком маленький или не виден в кадре робот останавливается.

```
try:
    while True:
        frame = picam2.capture_array()

        # Convert to HSV
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

        # Mask for blue
        mask = cv2.inRange(hsv, lower_blue, upper_blue)

        # Find contours
        contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

        movement_text = "Stopped"

        if len(contours) > 0:
            largest_contour = max(contours, key=cv2.contourArea)

            if cv2.contourArea(largest_contour) > 500:
                x, y, w, h = cv2.boundingRect(largest_contour)
                center_x = x + w // 2
                frame_width = frame.shape[1]

                # Определение зон: 30% слева, 40% центр, 30% справа
                left_bound = frame_width * 0.3
                right_bound = frame_width * 0.7

                # Draw rectangle around the detected object
                cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

                # Нарисуем границы зон
                cv2.line(frame, (int(left_bound), 0), (int(left_bound), frame.shape[0]), (255, 0, 0), 2)
                cv2.line(frame, (int(right_bound), 0), (int(right_bound), frame.shape[0]), (255, 0, 0), 2)

                if center_x < left_bound:
                    left()
```

Рисунок 2.14 – Строчки кода мобильного робота

Для того чтобы облегчить процесс захват изображения – необходимо разделить кадр на 3 зоны, от 0% до 30% - левая зона, от 30% до 70% - центр , от 70% до 100% - правая зона, для наглядности ниже представлен рисунок объясняющий этот процесс.

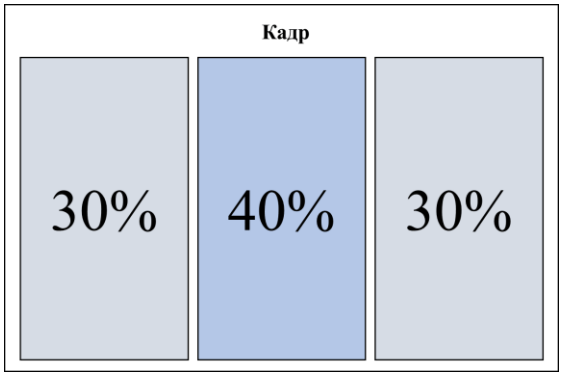


Рисунок 2.15 – Объяснение разделение кадра на три части

Данный рисунок иллюстрирует приблизительное разбиение кадра на сегменты.

2.5 Сбор прототипа

Данный этап моей работы является сборка рабочего прототипа робота, обеспечивающего взаимодействие всех элементов систем. Он позволяет оценить, соответствует ли поведение робота заданным функциям.

Как основным каркасам было принято решение использовать картон для архитектурного макетирования, простыми словами пенокартон толщиной 3мм. Для прочности пенокартон был из двух слоев с размером 25х15см. С нижней стороны пенокартона, были закреплены четыре мотора, также драйвер L298N для управления мотором был прикреплен с нижней стороны. К драйверу L298N был подключён аккумулятор на вход 12 В, расположенный в верхней части каркаса. Для проводов от контактов in1, in2, in3, in4, а также проводов питания и земли было выполнено отдельное отверстие в корпусе, через которое они аккуратно выведены верхнею часть каркаса. Это помогает сделать робот более компактным.

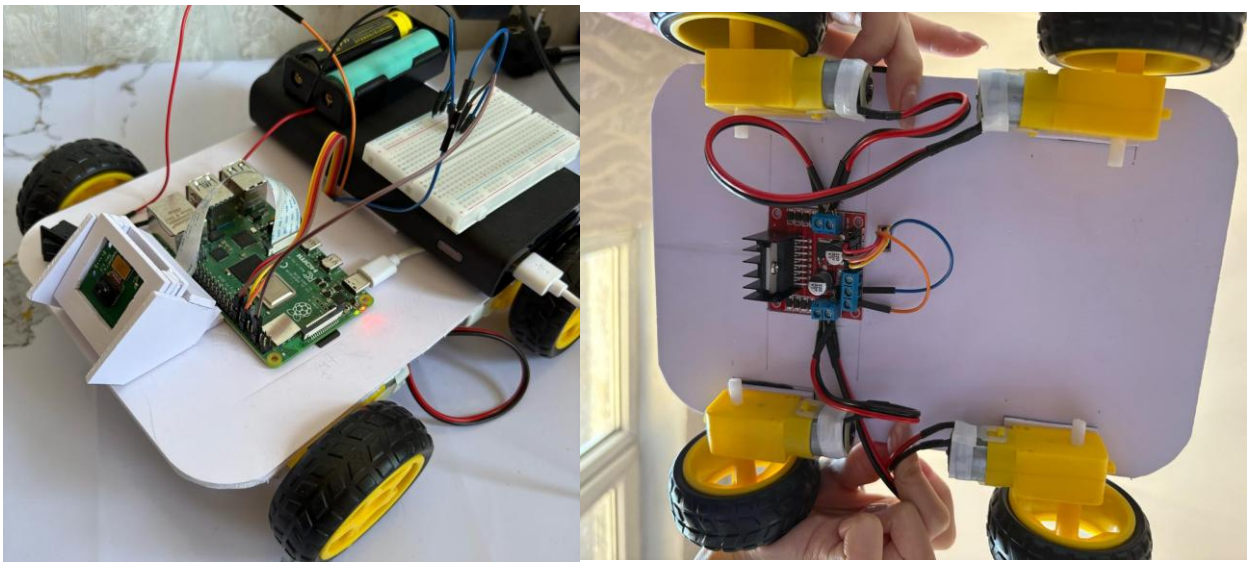


Рисунок 2.16 – Прототип для тестирование робота

Для запуска программы в терминале RealVNC Viewer нужно запустить код. Запуск кода осуществляется с помощью python3, после этого пишем

название сохраненного файла где находиться код и нажимаем на кнопку Enter. Через терминал мы можем смотреть как запускается программа.

Если посмотреть на вышедшее окно после запуска то мы можем увидеть два кадра показанные на рисунке 2.16 . Первый это называется маской или бинарным изображением. Это результат обработки изображения, когда выполняется фильтр по цвету. Белый цвет – там, где пиксели соответствуют нужному цвету. Черный цвет – все остальное, что не соответствует нужному цвету.

Второе изображение – это исходное изображение с камеры. В этом изображение мы можем более ясно понимать на какой цвет реагирует камера. Цвет который нам нужен выведен контуром зеленого квадрата. А красные линий это разделение изображение на три части, по левой и правой стороне тридцать процентов, а основная часть это движение вперед.

Также мы можем изменять яркость и насыщенность нужного цвета. В программе предусмотрено управление с клавиатуры: после запуска кода можно нажимать клавиши 1, 2, 3 – каждая из них отвечает за определённые параметры яркости и насыщенности цвета, что позволяет точнее выделить нужный объект на изображении.

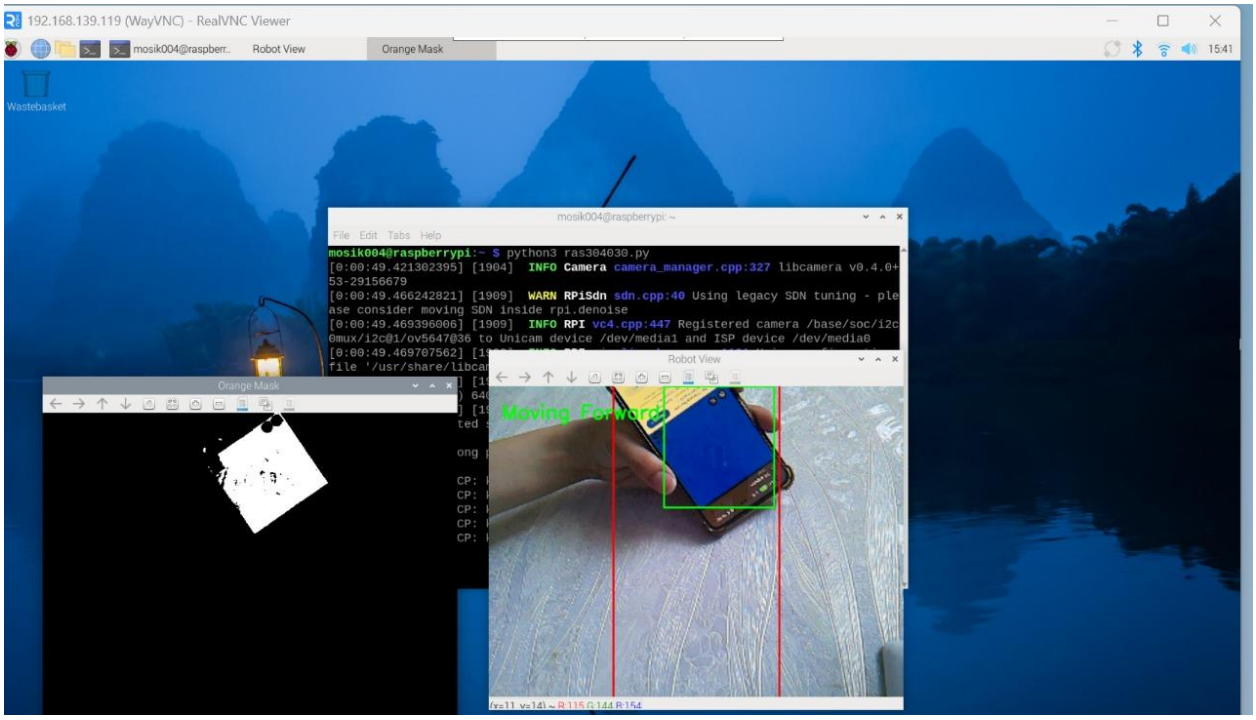


Рисунок 2.17 – Исходное и бинарное изображение полученные с камеры

Как видно на кадрах (рисунок 2.16, 2.17) если цвет находиться в середине то он едет вперед, а если с левой стороны то на лево. Окончить работу можно с помощью клавиши Q, и будет выход и программы.

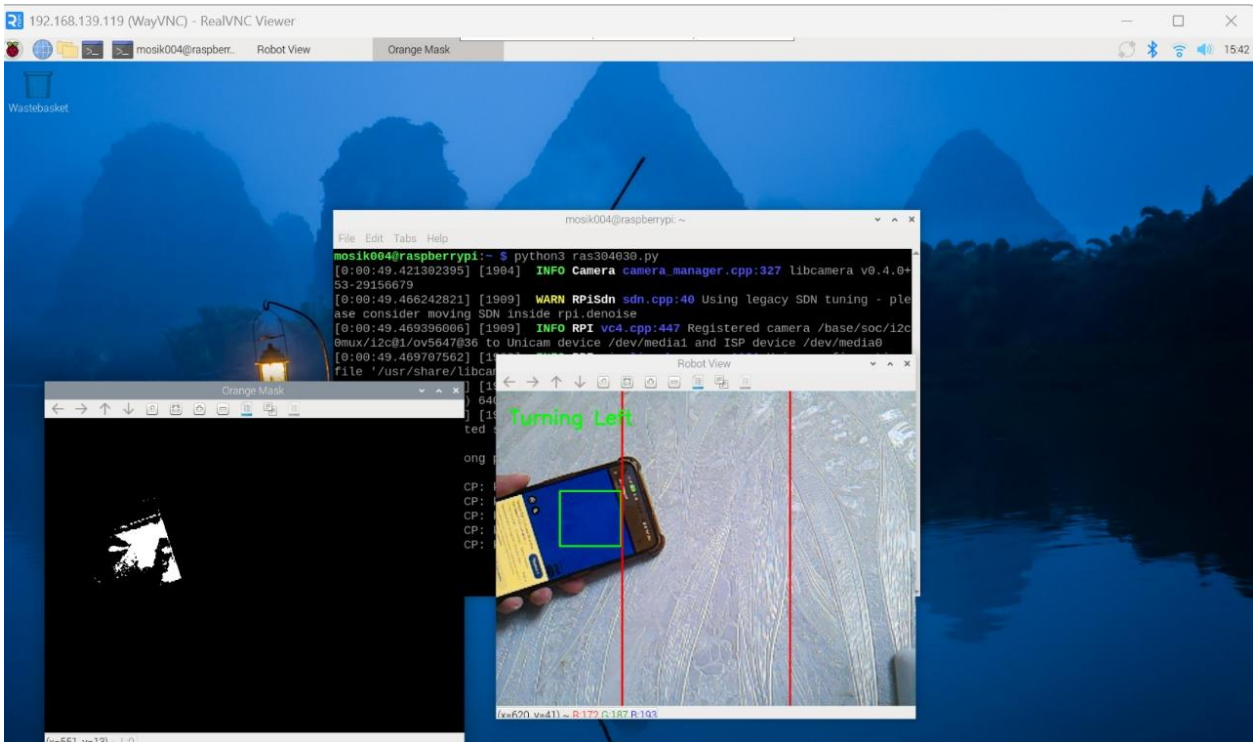


Рисунок 2.18 – Изменение положение преследуемого объекта робота

После тестирование прототипа на движение с помощью распознавание цветных объектов, можно переходить к сбору основного модели робота.

3 Сборка и реализация основного робота

В ходе апробации работы движения мобильного колесного робота с интегрированной системой распознавания цветных объектов, мною было замечено ограниченность угла обзора объектива камеры, вследствие чего я решила добавить дополнительный электромеханический элемент, на который можно будет жестко закрепить камеру.

3.1 Добавленные компоненты

– Сервопривод SG90
– Тумблер 10X15 MM

Сервопривод – это компактное электромеханическое устройство, преобразующее управляющий сигнал в точное угловое перемещение выходного вала. Конструктивно он состоит из электродвигателя, управляющей платы и понижающего редуктора, заключённых в прямоугольный корпус. Особенностью сервопривода является способность точно позиционировать вал в заданное положение в пределах ограниченного угла поворота, обычно составляющего от 60° до 180°. В работе был использован компактный сервопривод SG90.

Рисунок 3.1 – Сервопривод для изменение положение камеры

Тумблер 10X15 MM – это маленькая кнопка переключатель с двумя пинами. Рассчитан на максимальный ток 3А при напряжении 250В. Такие кнопки больше всего используются в электронных устройствах с низким уровнем напряжение. Данный тумблер является двухпозиционным фиксирующим выключателем который выдерживает стандарты пыле и влагозащиты IP56.

Рисунок 3.2 – Тумблер для подачи питания

3.2 Настройка и доработка программного обеспечения и схемы подключения компонентов

Основа кода также остается движением робота с использованием цветных объектов. Но для того чтобы робот был удобен в управлении были добавлены команды в память программы.

Первое что было добавлено это функция изменения позиций расположение угла обзора камеры с помощью угла наклона сервопривода. Сервопривод меняет расположение камеры верх и вниз. Для этого он был подключен к цифровому выходу 12 на плате Raspberry Pi как показано на схеме ниже.

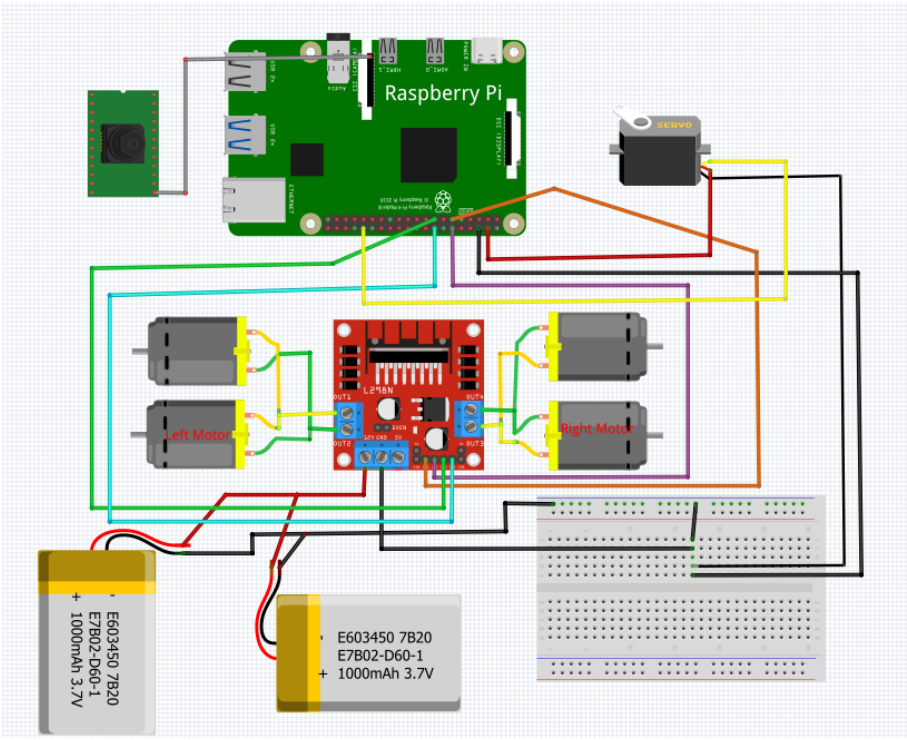
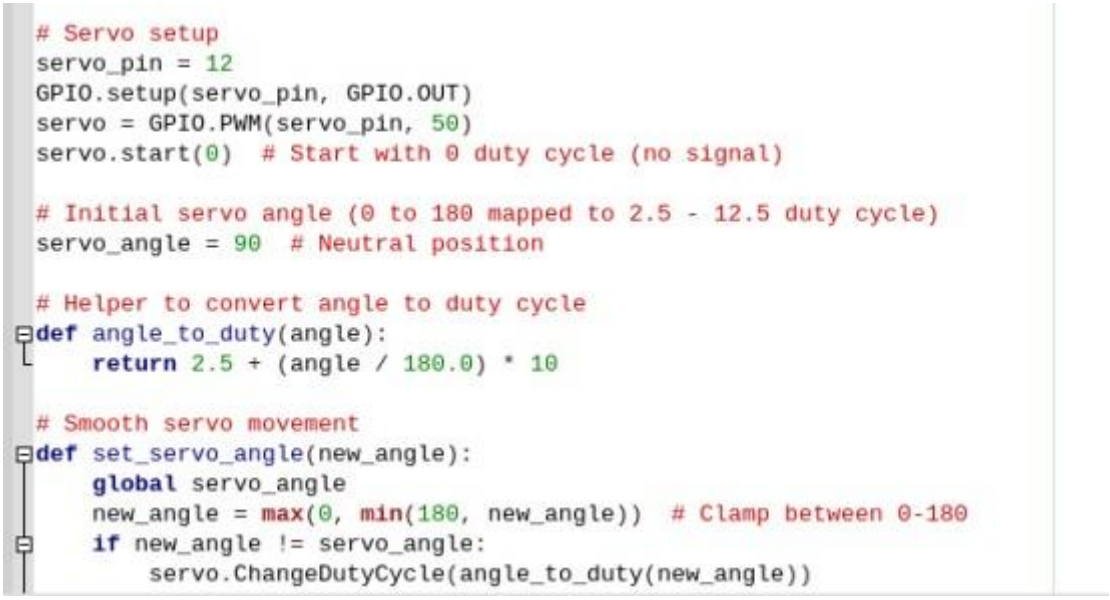


Рисунок 3.3 – Доработка принципиальной схемы робота

После добавление компонентов, можно преступить к изменению основного кода робота. Было выполнено настройка управлявшего пина для сервопривода, при изменении угла обеспечено плавное перемещение сервопривода к новой позиции. Вот основные функции добавленные в код:

- servo_pin = 12 — выбирается пин №12, к которому подключён сервопривод.
- GPIO.setup(..., GPIO.OUT) — пин настраивается как выход.
- GPIO.PWM(..., 50) — создаётся PWM-сигнал (широтно-импульсная модуляция) с частотой 50 Гц, что подходит для большинства сервоприводов.
- servo.start(0) — запускается ШИМ с нулевым сигналом, чтобы не подавать напряжение на сервопривод до первого поворота.
- servo_angle хранит текущий угол сервопривода.



```
# Servo setup
servo_pin = 12
GPIO.setup(servo_pin, GPIO.OUT)
servo = GPIO.PWM(servo_pin, 50)
servo.start(0)  # Start with 0 duty cycle (no signal)

# Initial servo angle (0 to 180 mapped to 2.5 - 12.5 duty cycle)
servo_angle = 90  # Neutral position

# Helper to convert angle to duty cycle
def angle_to_duty(angle):
    return 2.5 + (angle / 180.0) * 10

# Smooth servo movement
def set_servo_angle(new_angle):
    global servo_angle
    new_angle = max(0, min(180, new_angle))  # Clamp between 0-180
    if new_angle != servo_angle:
        servo.ChangeDutyCycle(angle_to_duty(new_angle))
```

Рисунок 3.4 – Добавление сервопривода в основной код

Основы работы сервопривода в том что работает команда вверх и вниз. В команде использованы такие функции как: def tilt_down() и def tilt_down() объявляет новую функцию а также наклон вверх и вниз, в зависимости что нам нужно. global servo_angle - говорит Python, что мы используем глобальную переменную servo_angle, а не создаём новую локальную. servo_angle 20 – вычитает 20 градусов из текущего угла, то есть поднимает камеру вверх. set_servo_angle(...) – вызывает основную функцию, которая поворачивает сервопривод на новый угол (и следит, чтобы он был от 0 до 180 градусов). Также для наклона вверх.

```
# Tilt control (small steps)
def tilt_up():
    global servo_angle
    set_servo_angle(servo_angle - 20)

def tilt_down():
    global servo_angle
    set_servo_angle(servo_angle + 20)

# Tracking flag
tracking = False
```

Рисунок 3.5 – Изменения позиции сервопривода

После того как дополнили основной код, где храниться основная часть кода, нужно создать новый файл templates в которым будет находиться код HTML для управления робота через веб-сайт. Но для того чтобы с сайта поступала информация основной код и код для управления должны находиться в одном файле. Как выглядит связь между файлами:

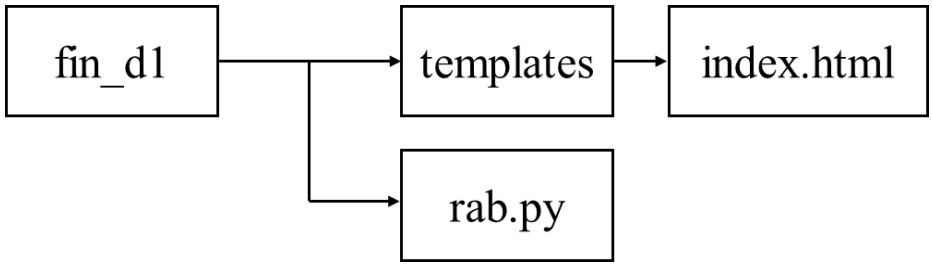


Рисунок 3.6 – Связь между файлами

fin_d1 это папка где храниться вся работа и управления робота. Создание папки осуществляется с помощью команды mkdir. После этого через команду cd открываем саму папку fin_d1. В папке fin_d1 можно запускать основной код с помощью команды python3, который был выше указано мной.

3.3 Трехмерное моделирование конструкции

Модель конструкции мобильного робота был собран в приложений SolidWorks. SolidWorks – это не просто программа для создание двухмерной или трехмерной графики, но также приложение может быть использована для симуляции и анализа в проведении расчетах на прочность, деформация, вибрация и т.д. , также в нем есть возможности для анимации и визуализации. Тем не менее, SolidWorks, как и другие подобные приложения, может

использоваться для подготовки моделей к производству, включая обработку на станках с ЧПУ и 3D-печать. Из-за этого приложения SolidWorks стало широко использоваться в машиностроении, промышленном дизайне, робототехнике, строительстве и в других инженерных отраслях. [19]

Для того чтобы робот был разбирающимся корпус робота был разделен на разные части, и спроектирован соотношением всех частей между собой. Первым делом было спроектировано основа робота. Это у нас базовая платформа. На базовый платформе размещаются основные модули, такие как: драйвер двигателей, аккумулятор, моторы приводящие робота к движению и т.д. Так как вся основная часть модулей находиться на базовой платформе, толщина платформы является толще чем другие части робота. Чтобы моторы робота двигались без препятствии на базовой платформе были сделаны посадочные места для моторов вместе с креплениями для надежности конструкции.

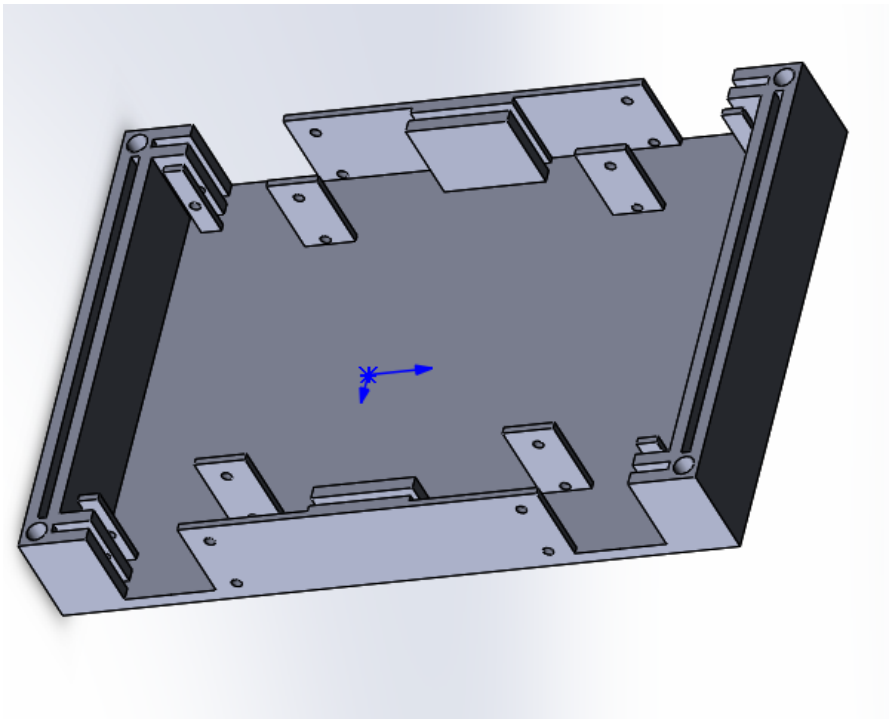


Рисунок 3.7 – Базовая платформа робота

В местах соединение базовый платформы и верхней части корпусов были созданы углы так скажем крепление для объединение всей части робота. Крепления имели зазоры, предназначенные для установки передней и боковых сторон корпуса робота. Также были отверстия и вытянутые цилиндры для соединение крепления между собой.

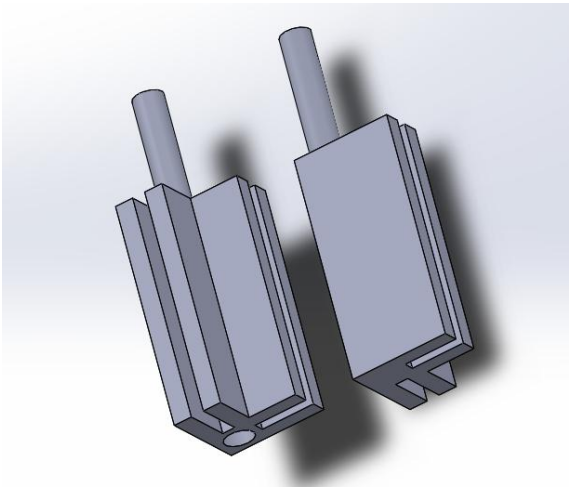


Рисунок 3.8 – Крепления для платформ между собой

Одним из самых простых частей робота является боковые стороны робота. Это две пластины с толщиной меньше чем у базовый платформы. А передняя часть робота была спроектирована так чтобы провода подключения сервопривода и шлейф камеры не мешали во время движения робота.

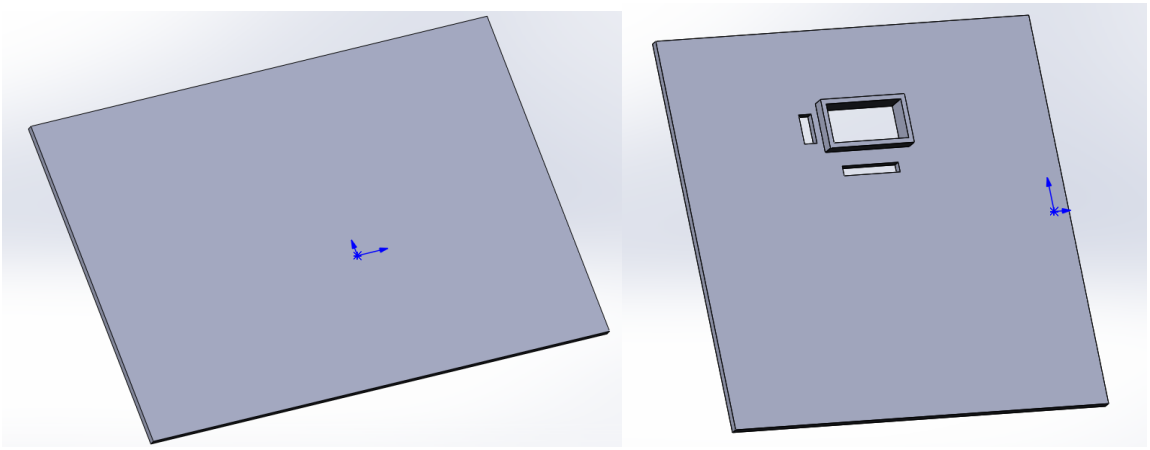


Рисунок 3.9 – Боковые стороны робота

После того как все компоненты робота были изготовлены, можно приступить к сбору робота. Сбор происходит очень легко, так как сам мобильный робот собирается как конструктор. Как видим на изображение 3.11 ниже мобильный робот оснащен камерой, остановленной на передней платформе с сервоприводом, что позволяет изменять угол обзора. Основная функция системы заключается в распознавании объектов по цвету с помощью алгоритмов компьютерного зрения. На основе анализа изображения с камеры

робот определяет положение цветного объекта в кадре и принимает решение о направление движения.

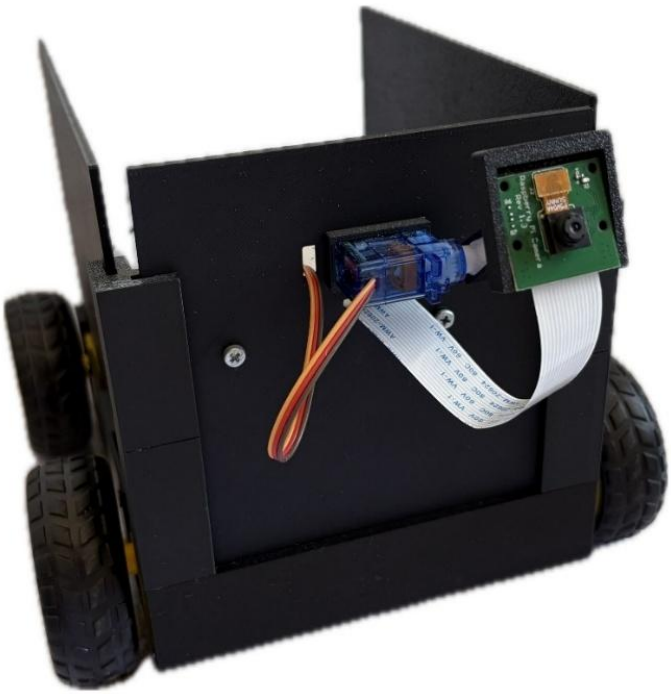


Рисунок 3.11 – Собранный мобильный робот

Таким образом робот способен самостоятельно реагировать на визуальные сигналы из окружающей среды и осуществлять базовую навигацию.

Заключение

В ходе выполнения данной работы был разработан и реализован прототип мобильного робота, способного перемещаться в пространстве, ориентируясь на цветные объекты с использованием технологии компьютерного зрения. В процессе реализации были изучены основы работы с одноплатным компьютером Raspberry Pi, камера CSI, а также библиотеки OpenCV и другие необходимые программные инструменты.

Разработанная система успешно выполняет задачи распознавания объектов по цвету и управления движением робота в зависимости от их положения в кадре. Это подтверждает возможность применения подобных решений в реальных условиях, где необходима простая и доступная система навигации.

Созданный робот может служить основой для дальнейших улучшений – таких как использование более сложных алгоритмов компьютерного зрения, интеграция других сенсоров (ультразвуковых, ИК, GPS) или разработка полноценной автономной системы.

Таким образом, поставленные цели и задачи были достигнуты. Работа показала, что простые методы компьютерного зрения могут эффективно применяться в робототехнических системах для реализации автономного поведения.

Список терминов и сокращений

HSV (Hue, Saturation, Value – тон, насыщенность, значение) - цветовая модель, описывающая цвет через оттенок (0-360°), насыщенность (0-100%) и яркость (0-100%). Удобна для графики и обработки изображений, так как ближе к человеческому восприятию, чем RGB.

Локализовываться – определять своё положение в пространстве.

SSH (Secure Shell) – это протокол для безопасного удалённого доступа к другому устройству через сеть.

GPS (Global Positioning System) – это глобальная система позиционирования, которая позволяет определить местоположение объекта на Земле с помощью спутников.

Камера CSI (Camera Serial Interface) – это камера, которая подключается к специальному разъёму на плате Raspberry Pi с помощью плоского шлейфа (FFC).

OpenCV – это библиотека с открытым исходным кодом для компьютерного зрения и обработки изображений.

AR (Augmented Reality) – это технология, при которой цифровая информация (изображения, текст, 3D-объекты) накладывается на изображение реального мира в режиме реального времени.

VR (Virtual Reality) – это технология, создающая полностью искусственную (виртуальную) среду, в которую пользователь погружается с помощью специальных устройств (например, VR-шлемов).

MMAL – это низкоуровневый мультимедийный API, разработанный специально для Raspberry Pi.

V4L2 – это стандартный Linux API для работы с видеоустройствами (веб-камеры, ТВ-тюнеры и т.д.).

Список использованной литературы

1. Kelly, A. (2013). Mobile Robotics: Mathematics, Models, and Methods — стр. 1–10.
2. Siegwart, R., Nourbakhsh, I., & Scaramuzza, D. (2011). Introduction to Autonomous Mobile Robots – стр. 15–45.
3. Юревич Е. И. Основы робототехники : учеб. пособие. – 4-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2020. – 302 с. : ил. - (Учебная литература для вузов).58стр http://library.atu.kz/flgl/48811.pdf
4. Первый мобильный робот https://www.sri.com/hoi/shakey-the-robot/
5. Развитие компьютерного зрение https://www.databridgemarketresearch.com/ru/news/global-computer-vision-market?utm_source=chatgpt.com
6. Spot, мобильный робот https://dev.bostondynamics.com/python/examples/docs/perception_world_objects_examples.html?utm_source=chatgpt.com , https://dev.bostondynamics.com/python/examples/spot_tensorflow_detector/readme?utm_source=chatgpt.com
7. Рынок КП https://www.grandviewresearch.com/industry-analysis/computer-vision-market
8. Формула сложного процента https://ru.onlinemschool.com/math/library/percent/percent10/?utm_source=chatgpt.com
9. Определения компьютерного зрения https://www.ibm.com/think/topics/computer-vision
10. Обработка изображении https://www.pre-scient.com/knowledge-center/image-processing/image-processing-algorithms/
11. Взаимосвязь между обработкой изображений и компьютерным зрением https://opencv.org/blog/computer-vision-and-image-processing/
12. RGB и HSV https://medium.com/@DIYCoding/color-models-in-image-processing-understanding-rgb-and-hsv-for-computer-vision-a629641cdc40
13. Момент изображении https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/#:~:text=In%20computer%20vision%20and%20image,orientation%2C%20and%20other%20desirable%20properties.
14. Основа для реализации

https://youtu.be/axirwURA0aU?si=mVc8mOOq9YYv0SD3
15. Микрокомпьютер Raspberry pi 4
https://unitsolutions.kz/sistemy-upravleniya/5671-mikrokompyuter-raspberry-pi-4-model
b.html?srsltid=AfmBOorhjgrmvUCm6e20fDq33Dgqi2ddFXCGlz9uFNNYeWTufTG7dein
16. Камера Raspberry Pi Board
https://radiomart.kz/raspberry-pi/819-raspberry-pi-camera-board-8471800000.html
17. Мотор редуктор
https://robot-kit.ru/3187/
18. Драйвер L298N
https://3d-diy.ru/blog/drayver-dvigatelya-l298n/?srsltid=AfmBOorTrLn1cxUA5tw_RIjDce031sh_uzI9lNiLRY1pgbtHzU1UlvpB
19. Самоучитель SolidWorks 2010 / Н. Ю. Дударева, С. А. Загайко. — СПб.: БХВ-Петербург, 2011. — 416 с.: ил. + CD-ROM. 13стр

ПРИЛОЖЕНИЕ А

```
from flask import Flask, render_template, request
import RPi.GPIO as GPIO
import threading
import cv2
import numpy as np
import time
from picamera2 import Picamera2

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

in1, in2, in3, in4 = 17, 18, 22, 23
for pin in [in1, in2, in3, in4]:
    GPIO.setup(pin, GPIO.OUT)

servo_pin = 12
GPIO.setup(servo_pin, GPIO.OUT)
servo = GPIO.PWM(servo_pin, 50)
servo.start(0)

servo_angle = 90

def angle_to_duty(angle):
    return 2.5 + (angle / 180.0) * 10

def set_servo_angle(new_angle):
    global servo_angle
    new_angle = max(0, min(180, new_angle))
    if new_angle != servo_angle:
        servo.ChangeDutyCycle(angle_to_duty(new_angle))
        time.sleep(0.15)
        servo.ChangeDutyCycle(0)
        servo_angle = new_angle

picam2 = Picamera2()
picam2.configure(picam2.create_preview_configuration(main={"format":
"BGR888", "size": (640, 480)}))
```

```
picam2.start()
time.sleep(2)

blue_ranges = [
    (np.array([5, 150, 150]), np.array([15, 255, 255])),
    (np.array([0, 100, 100]), np.array([25, 255, 255])),
    (np.array([10, 100, 100]), np.array([25, 255, 255]))
]
lower_blue, upper_blue = blue_ranges[0]

def stop():
    for pin in [in1, in2, in3, in4]:
        GPIO.output(pin, GPIO.LOW)

def forward():
    GPIO.output(in1, GPIO.HIGH)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in3, GPIO.HIGH)
    GPIO.output(in4, GPIO.LOW)

def left():
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in2, GPIO.HIGH)
    GPIO.output(in3, GPIO.HIGH)
    GPIO.output(in4, GPIO.LOW)

def right():
    GPIO.output(in1, GPIO.HIGH)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in4, GPIO.HIGH)

def tilt_up():
    global servo_angle
    set_servo_angle(servo_angle - 20)

def tilt_down():
    global servo_angle
    set_servo_angle(servo_angle + 20)
```

```
tracking = False

def track_object():
    global tracking
    while tracking:
        frame = picam2.capture_array()
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(hsv, lower_blue, upper_blue)
        contours, _ = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

        if contours:
            largest = max(contours, key=cv2.contourArea)
            if cv2.contourArea(largest) > 500:
                x, y, w, h = cv2.boundingRect(largest)
                center_x = x + w // 2
                width = frame.shape[1]
                left_bound, right_bound = width * 0.3, width * 0.7

                if center_x < left_bound:
                    left()
                elif center_x > right_bound:
                    right()
                else:
                    forward()
            else:
                stop()
        else:
            stop()
        time.sleep(0.1)
    stop()

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/tilt", methods=["POST"])
```

```
def tilt():
    direction = request.form["direction"]
    if direction == "up":
        tilt_up()
    elif direction == "down":
        tilt_down()
    return ("", 204)

@app.route("/track", methods=["POST"])
def track():
    global tracking
    action = request.form["action"]
    if action == "start" and not tracking:
        tracking = True
        threading.Thread(target=track_object).start()
    elif action == "stop":
        tracking = False
        stop()
    return ("", 204)

if __name__ == "__main__":
    try:
        app.run(host="0.0.0.0", port=5000)
    finally:
        stop()
        servo.stop()
        GPIO.cleanup()
```

ПРИЛОЖЕНИЕ Б

```
<!DOCTYPE html>
<html>
<head>
  <title>Robot Control Panel</title>
  <style>
    body {
      text-align: center;
      font-family: Arial, sans-serif;
      background-color: #f2f2f2;
      padding-top: 50px;
    }
    h1, h2 {
      font-size: 2em;
      margin-bottom: 20px;
    }
    form {
      margin: 20px auto;
    }
    button {
      font-size: 1.5em;
      padding: 15px 30px;
      margin: 10px;
      cursor: pointer;
      background-color: #007BFF;
      color: white;
      border: none;
      border-radius: 10px;
      transition: background-color 0.3s;
    }
    button:hover {
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <h1>Robot Control Panel</h1>
```

```
    <h2>Camera Tilt</h2>
    <form action="/tilt" method="post">
        <button name="direction" value="up">Up</button>
        <button name="direction" value="down">Down</button>
    </form>

    <h2>Object Tracking</h2>
    <form action="/track" method="post">
        <button name="action" value="start">Start</button>
        <button name="action" value="stop">Stop</button>
    </form>
</body>
</html>
```

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. К.И. САТПАЕВА

РЕЦЕНЗИЯ

на дипломный проект студентки образовательной программы
6B07111 – Робототехника и мехатроника
Шанаевой Маншук Темиргалиевны

Тема работы: **«Движение робота с использованием распознавания цветных объектов»**

Объём выполненной работы:
а) графическая часть — 28 листов
б) пояснительная записка — 43 страниц

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Дипломный проект Маншук посвящён актуальной и практически значимой теме — разработке мобильного робота с системой компьютерного зрения, ориентирующегося в пространстве по цветовым меткам. Работа демонстрирует уверенное владение современными технологиями программирования, обработки изображений и интеграции аппаратных компонентов.

Актуальность проекта обусловлена растущим применением интеллектуальных систем в логистике, образовании и автоматизации. Робот построен на базе Raspberry Pi с использованием библиотеки OpenCV и CSI-камеры, реализована функция распознавания цветных объектов, управление движением и поворот камеры с помощью сервопривода. Дополнительно внедрён веб-интерфейс для дистанционного управления.

Проект имеет чёткую структуру: изложены теоретические основы, обоснован выбор комплектующих, описаны этапы сборки, алгоритмы работы и результаты тестирования. Отдельного внимания заслуживает использование SolidWorks для трёхмерного моделирования конструкции и грамотное распределение компонентов.

Работа отличается высоким уровнем самостоятельности и инженерной проработки. Автор продемонстрировал понимание перспектив развития темы, предложив дальнейшее расширение системы за счёт дополнительных сенсоров и более сложных алгоритмов компьютерного зрения.

ОТМЕЧАЕМЫЕ ОСОБЕННОСТИ И РЕКОМЕНДАЦИИ

Работа демонстрирует высокий уровень практической реализации: представлен собранный и протестированный прототип. Вместе с тем, не представлены количественные данные по точности распознавания и устойчивости работы в различных условиях. Текст требует небольшой стилистической доработки для улучшения логической связности и читаемости отдельных разделов.

ЗАКЛЮЧЕНИЕ

В целом, работа выполнена на высоком уровне, демонстрирует профессиональную компетентность автора, владение инструментами проектирования и ориентацию на прикладные задачи в области робототехники. Работа соответствует установленным требованиям и заслуживает оценки **«отлично» (A)** с рекомендацией к защите.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. К.И. САТПАЕВА

Рецензент Старший преподаватель кафедры информационных технологий и библиотечного дела Казахского национального женского педагогического университета, PhD Алимбаева Ж. Н.

«5» 06 2025 г

НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И. САТПАЕВА»

ОТЗЫВ

дипломного проекта (работы)

студента специальности 6В07111 – «Робототехника и мехатроника»

Шанаева Маншук Темиргалиевна

На тему: **«Движение робота с использованием распознавания цветных объектов»**

Дипломная работа посвящена разработке и реализации мобильного робота с системой компьютерного зрения, способного ориентироваться в пространстве с помощью распознавания цветных объектов. Тема актуальна и востребована в условиях активного внедрения автономных и интеллектуальных систем в различные сферы деятельности.

В ходе работы студентка проявила высокий уровень самостоятельности, инициативности и заинтересованности в изучении современных технологий. Она успешно справилась с проектированием аппаратной части, сборкой устройства, разработкой программного обеспечения и проведением испытаний. Работа охватывает как теоретические аспекты, так и практическую реализацию, включая использование OpenCV, Raspberry Pi, CSI-камеры, сервопривода и 3D-моделирования в SolidWorks. Кроме того, реализована веб-интерфейсная система управления, что расширяет функциональные возможности робота.

Следует отметить и творческий подход к доработке конструкции: добавлена возможность изменения угла обзора камеры, что повышает адаптивность устройства. Работа выполнена с соблюдением инженерных требований и подтверждена прототипом, прошедшим тестирование.

Некоторые разделы требуют небольшой стилистической доработки, а количественные оценки эффективности системы могли бы дополнительно усилить практическую значимость проекта. Тем не менее, поставленные цели достигнуты в полном объёме.

Считаю, что дипломная работа заслуживает оценки «отлично», а её автор – присвоения квалификации бакалавра техники и технологий.

Научный руководитель
Магистр технических наук, преподаватель
_____ Игембай Е.А.
«__» ____ 2025 г.

Отчет подобия

Метаданные

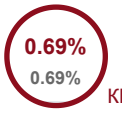
Название организации
Satbayev University
Название
Движение робота с использованием распознавания цветных объектов

Автор
Шанаева Маншук Темиргалиевна
Научный руководитель / Эксперт
Ерболат Игембай

Подразделение
ИАиИТ

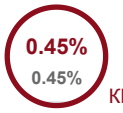
Объем найденных подобий

КП-ия определяют, какой процент текста по отношению к общему объему текста был найден в различных источниках.. Обратите внимание!Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



КП1

КП2



КЦ

25
Длина фразы для коэффициента подобия 2

8313
Количество слов

62417
Количество символов

Тревога

В этом разделе вы найдете информацию, касающуюся текстовых искажений. Эти искажения в тексте могут говорить о ВОЗМОЖНЫХ манипуляциях в тексте. Искажения в тексте могут носить преднамеренный характер, но чаще, характер технических ошибок при конвертации документа и его сохранении, поэтому мы рекомендуем вам подходить к анализу этого модуля со всей долей ответственности. В случае возникновения вопросов, просим обращаться в нашу службу поддержки.

Замена букв		10
Интервалы	A⟶	0
Микропробелы		70
Белые знаки		0
Парафразы (SmartMarks)	a	7

Подобия по списку источников

Ниже представлен список источников. В этом списке представлены источники из различных баз данных. Цвет текста означает в каком источнике он был найден. Эти источники и значения Коэффициента Подобия не отражают прямого плагиата. Необходимо открыть каждый источник и проанализировать сожержание и правильность оформления источника.

10 самых длинных фраз — Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://imacrosoft.ru/raznoe/chto-vhodit-v-sostav-kompyutera-chto-vhodit-v-sostav-personalnogo-kompyutera-parametry-i-vidy-pk.html	19 0.23 %
2	https://imacrosoft.ru/raznoe/chto-vhodit-v-sostav-kompyutera-chto-vhodit-v-sostav-personalnogo-kompyutera-parametry-i-vidy-pk.html	16 0.19 %

3	Применение Agile методологии в деятельности компании [redacted] 5/2/2023 Almaty Management University (Менеджмент, маркетинг и логистика)	14 0.17 %
4	https://imacrosoft.ru/raznoe/chto-vhodit-v-sostav-kompyutera-chto-vhodit-v-sostav-personalnogo-kompyutera-parametry-i-vidy-pk.html	8 0.10 %

из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

из домашней базы данных (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

из программы обмена базами данных (0.17 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	Применение Agile методологии в деятельности компании [redacted] 5/2/2023 Almaty Management University (Менеджмент, маркетинг и логистика)	14 (1) 0.17 %

из интернета (0.52 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://imacrosoft.ru/raznoe/chto-vhodit-v-sostav-kompyutera-chto-vhodit-v-sostav-personalnogo-kompyutera-parametry-i-vidy-pk.html	43 (3) 0.52 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)