БОЛЬШИЕ ВЫЗОВЫ

МЕЖДУНАРОДНЫЙ КОНКУРС НАУЧНО-ТЕХНОЛОГИЧЕСКИХ ПРОЕКТОВ

Направление: *«Транспортно-логистические системы, морские, авиационные и беспилотные технологии», «Большие данные, искусственный интеллект, автоматизированные системы и информационная безопасность»*

ПРОЕКТНАЯ РАБОТА

Тема: «Разработка мобильного робота с элементами искусственного интеллекта и удаленным веб-управлением»

Выполнила: ученица 9 «А» класса

ГБОУ СОШ №282 г. Санкт-Петербурга

Ковалева Алиса Ивановна

Санкт-Петербург

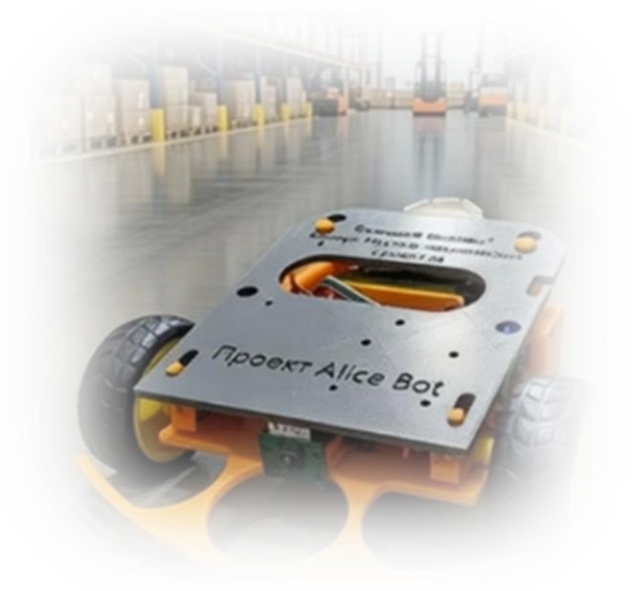
2026 г.

1. ВВЕДЕНИЕ

Актуальность проекта Развитие современной робототехники неразрывно связано с компьютерным зрением. Большинство существующих роботизированных платформ перемещаются «вслепую», опираясь на таймеры или базовые датчики расстояния (ультразвуковые или лидары). В отличие от них, системы визуальной навигации позволяют роботу не просто избегать механических столкновений, но и узнавать объекты в пространстве, анализировать их положение и самостоятельно принимать решения о корректировке маршрута.

Проблематика Одной из главных проблем современной складской логистики и автоматизации закрытых помещений (например, больниц или "чистых" зон) является недоступность спутникового сигнала GPS. Для ориентации тяжелых транспортных платформ внутри зданий нередко применяются дорогостоящие лазерные дальномеры (лидары). Использование дешевых оптических камер в связке с бумажными ArUco-маркерами решает эту проблему, позволяя создать экономичную и высокоточную систему навигации.

Цель проекта Самостоятельно создание автономного мобильного робота, из имеющихся в продаже недорогих компонентов, способного не только управляться дистанционно через веб-интерфейс, но и самостоятельно следовать за целью по заданным визуальным меткам.



Задачи проекта:

1. Спроектировать и собрать работающий аппаратный прототип устройства (разработка шасси, подбор компонентов, реализация подсистемы питания).

2. Реализовать алгоритмы компьютерного зрения для распознавания ArUco-маркеров и настроить захват видеопотока в операционной системе Linux.

3. Разработать и развернуть кроссплатформенный веб-интерфейс для обеспечения удобной системы удаленного контроля и FPV-управления.

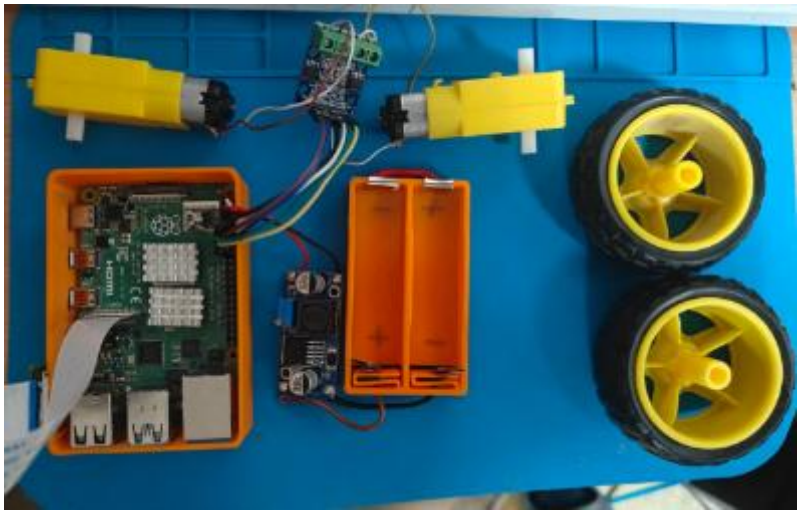
Объект исследования: Автономные мобильные роботизированные платформы и система компьютерного зрения.

Предмет исследования: Алгоритмы визуальной навигации (компьютерного зрения) и системы дистанционного управления на базе веб-технологий.

Практическая значимость: Реализованный в проекте программный код и аппаратная архитектура являются готовой масштабируемой базой (прототипом) для создания парка умных тележек, автономно перевозящих грузы и способных безопасно функционировать рядом с людьми.

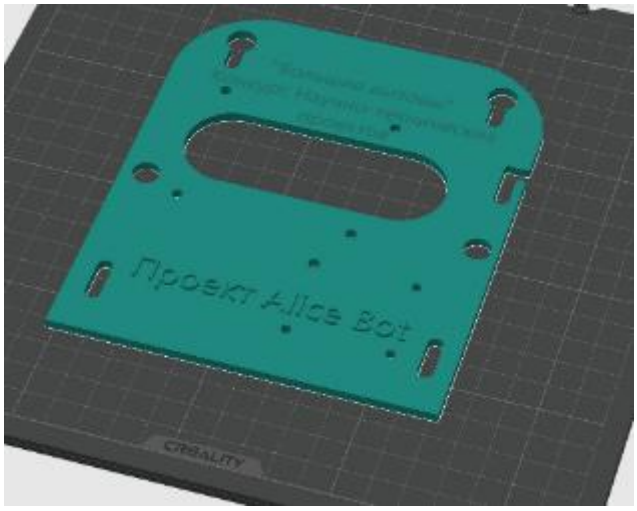
ГЛАВА 1. АППАРАТНАЯ РЕАЛИЗАЦИЯ (КОНСТРУКЦИЯ И ЭЛЕКТРОНИКА)

1.1. Выбор вычислительной платформы и базовых компонентов Для обеспечения достаточной вычислительной мощности, необходимой для обработки видеопотока в реальном времени, в качестве «мозга» робота мною был выбран



имеющийся у меня в наличии одноплатный микрокомпьютер Raspberry Pi 4. В качестве органа «зрения» я дополнительно приобрела недорогой компактный модуль камеры OV5647, подключенный напрямую к интерфейсу CSI микрокомпьютера. Приводом послужили два мотор-редуктора постоянного тока (TT-моторы) с пластиковыми колесами, обеспечивающие схему управления за счет разности скоростей вращения левого и правого колес.

1.2. Проектирование шасси и 3D-печать (Конструктивная часть)



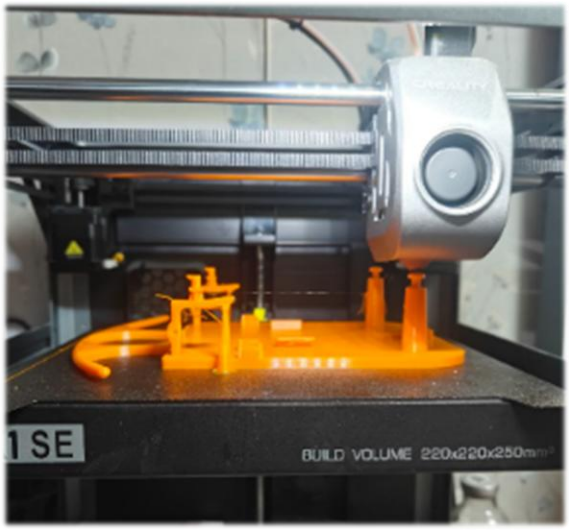
На начальном этапе прототип робота был смонтирован на временном шасси при помощи термоклея для проверки принципиальной работоспособности электронных схем. После успешных тестов я приступила к проектированию полноценного корпуса. Основные элементы корпуса и крепления я изготовила с применением аддитивных технологий (3D-печать). Это позволило мне спроектировать точные посадочные места под все компоненты: Raspberry Pi, держатель для аккумуляторов и моторы.

Особенности компоновки:

- **Снижение центра тяжести:** Тяжелые элементы (аккумуляторы и мотор-редукторы) я расположила максимально низко, что исключает опрокидывание платформы при резких стартах и торможениях.

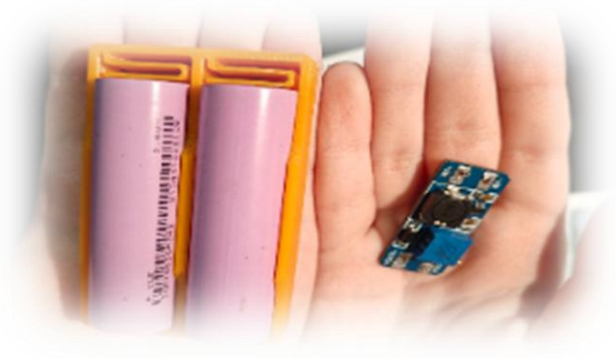
- **Экологичный подход к деталям:** В дифференциальном приводе роботу требуется третья, пассивная точка опоры. Вместо покупки готового поворотного колеса, создающего люфты, мною был применен творческий подход: в корпус был интегрирован гладкий шар от использованного роликового дезодоранта. Это обеспечило роботу скольжение по любым поверхностям.

- Специально для конкурса «Большие вызовы» мною была спроектирована и распечатана кастомная верхняя крышка с названием проекта.



1.3. Разработка подсистемы питания Главная сложность аппаратной части заключалась в высоких требованиях Raspberry Pi к стабильности тока. При пиковых нагрузках на процессор (во время обработки видео, старте двигателей) обычные пауэрбанки давали просадку напряжения, что приводило к перезагрузке системы. Для решения проблемы была спроектирована раздельная система питания на базе двух высокотоковых Li-ion аккумуляторов формата 18650 со встроенной защитой (конфигурация 2S, номинальное напряжение 7.4В). Было принято решение разделить питание на два контура:

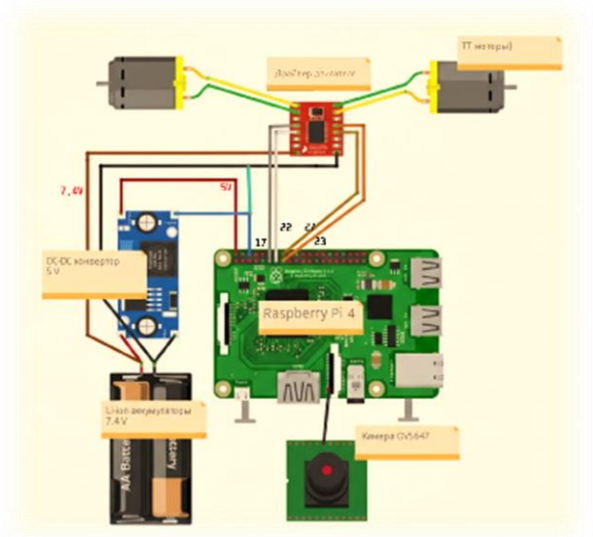
1. **Логический контур:** Напряжение 7.4В подается на импульсный DC-DC понижающий преобразователь, который выдает стабильные 5В напрямую на пины GPIO (4 и 6) Raspberry Pi.



2. **Силовой контур:** Питание на драйвер двигателей подается напрямую от аккумуляторов (7.4В), минуя микрокомпьютер, что защищает его от помех работающих моторов.

1.4. Инженерный вызов: модернизация драйвера двигателей.

В первой ревизии прототипа управление двигателями осуществлялось через популярный компактный драйвер L9110S. Однако в ходе полевых испытаний возникла критическая аппаратная проблема. TT-моторы рассчитаны на напряжение 3–6В. При подаче на них напряжения напрямую от аккумуляторов 2S, пусковые токи в момент старта



или при встрече с препятствием резко возрастали до. Лимит чипа L9110S составляет всего 800 мА, из-за чего во время одного из тестов базовый драйвер сгорел от перегрева.

Проблема была успешно локализована и решена путем замены силового ключа на имеющийся у меня в наличии более мощный но прожорливый промышленный модуль L298N. Этот драйвер оснащен массивным радиатором охлаждения и способен выдерживать пиковые токи до 3А на канал, что сделало работу шасси абсолютно безотказной даже при повышенном напряжении, но ценою этого стало большее потребление тока. Управление модулем L298N осуществляется через ШИМ-сигналы с пинов GPIO 17, 27, 22 и 23.

1.5. Экономическая эффективность выбранной архитектуры

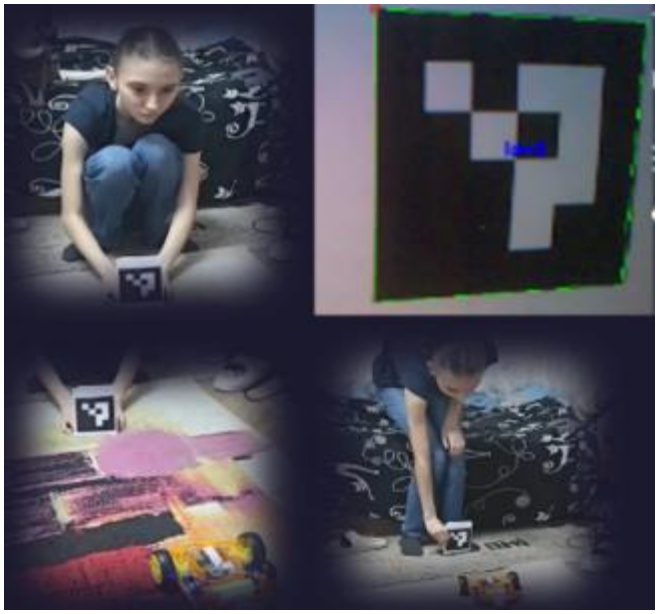
Традиционные промышленные AGV-платформы, используемые на складах, опираются на системы лазерного сканирования пространства (лидары). Стоимость простейшего 2D-лидара начинается составляет десятки тысяч рублей, что многократно увеличивает итоговую стоимость каждого робота в парке. Предложенная в моем проекте архитектура визуальной навигации опирается на связку дешевой оптической камеры (купленной нами буквально за 500 рублей) и распечатанных на принтере бумажных ArUco-маркеров, себестоимость которых в сравнении с лидаром стремится к нулю. Использование алгоритмов компьютерного зрения позволяет достичь точности позиционирования, сопоставимой с лазерными радарами, при этом снижая стоимость аппаратной базы робота в несколько раз. Это делает технологию экономически привлекательной для масштабирования на предприятиях малого и среднего бизнеса.

ГЛАВА 2. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И КОМПЬЮТЕРНОЕ ЗРЕНИЕ

2.1. Выбор программного стека и архитектуры

Проект написан на высокоуровневом языке программирования Python 3 и функционирует под управлением операционной системы Raspberry Pi OS (на базе ядра Linux). Для прямого взаимодействия с аппаратной частью (управления мотор-редукторами) мною применена библиотека GPIOZero, а для захвата видеокадров с аппаратного интерфейса камеры — современная библиотека Picamera2 (старая библиотека в новой ОС отказалась работать). Обработка визуальных данных реализована с помощью открытой библиотеки компьютерного зрения OpenCV, а математические расчеты векторов движения и расстояний выполняются средствами библиотеки NumPy.

2.2. Система визуальной навигации (ArUco-маркеры)



В отличие от простых роботизированных систем, которые перемещаются «вслепую» по запрограммированным таймерам, мой бот использует полноценную систему технического зрения (Computer Vision). Основой навигации стала технология распознавания пространственных ArUco-маркеров. В проекте мною задействован стандартный словарь маркеров «DICT_4X4_50». Я выбрала данную технологию из-за ее достаточно высокой устойчивости к визуальному шуму и изменениям

освещенности. Отладка программы, корректировка коэффициентов и визуальный контроль по камере осуществлялись через VNC-сервер, установленный в ОС робота посредством ПК.

2.3. Конвейер обработки визуальных данных Обработка видеопотока на мобильном микрокомпьютере требует жесткой оптимизации для сохранения высокой частоты кадров (FPS). Исходный видеокадр с камеры захватывается в высоком разрешении для качественной трансляции оператору по сети. Однако для модуля машинного зрения этот кадр программно ужимается в разрешении и переводится в градации серого (Grayscale). Алгоритм OpenCV производит бинаризацию изображения (превращая пиксели в строго черные и белые), после чего ищет замкнутые контуры. Если найденный контур представляет собой четырехугольник, программа извлекает его внутренний паттерн и сверяет с матрицей словаря DICT_4X4_50. Если паттерн совпадает с искомым ID, функция возвращает координаты четырех углов маркера, на основе которых высчитывается его центральная точка. Весь этот процесс выполняется десятки раз в секунду.

2.4. Алгоритм следования и П-регулятор Логика автономного следования построена на непрерывном математическом вычислении координат центра ArUco-маркера в кадре. Для корректировки траектории движения платформы внедрен Пропорциональный регулятор. Принцип его работы заключается в следующем: чем сильнее целевой маркер смещается (отклоняется) от центра поля зрения камеры, тем большая разность скоростей подается на левое и правое колесо. Это заставляет привод робота плавно подруливать к цели без резких рывков.

2.5. Система безопасности «Автостоп»

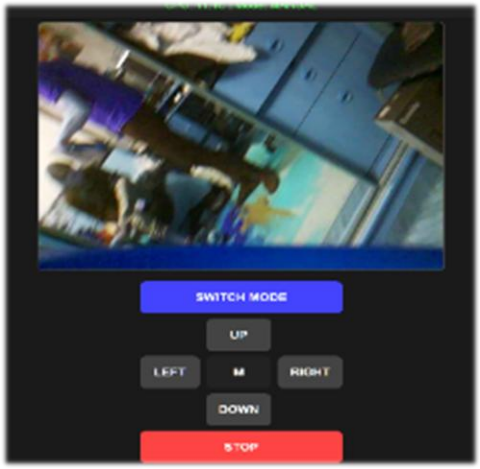
Для предотвращения физических столкновений робота с объектами или оператором я реализовала программную функцию "Автостоп". В процессе движения алгоритм непрерывно анализирует ширину захваченного маркера в пикселях. Экспериментальным путем было установлено пороговое значение: если ширина маркера превышает 180 единиц, это означает, что объект находится слишком близко. Примерное расстояние автоматической остановки при этом составило около 30 см. В этот момент колеса мгновенно останавливаются, и выполнение программы переходит в безопасный режим ожидания. Если снова отдалить метку от камеры, робот автоматически возобновляет следование за ней.

ГЛАВА 3. ВЕБ-ИНТЕРФЕЙС И СИСТЕМА УДАЛЕННОГО УПРАВЛЕНИЯ

3.1. Разработка кроссплатформенного сервера.

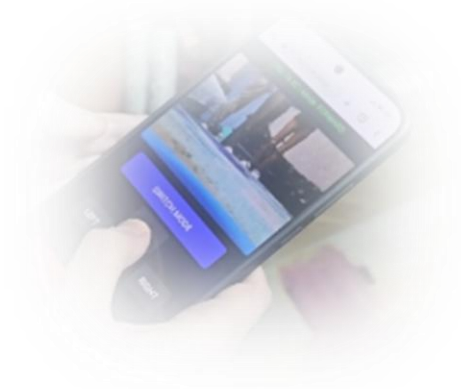
Для обеспечения удобства взаимодействия с пользователем, помимо автономного режима, было разработано легковесное веб-приложение для перевода бота в режим прямого FPV-контроля. Серверная часть реализована на базе фреймворка Flask. При запуске скрипта Raspberry Pi поднимает локальный веб-сервер, который транслирует видеопоток в формате MJPEG и принимает входящие HTTP-запросы от смартфона или компьютера пользователя. Данное решение обеспечивает полную всеядность и автономность системы без необходимости устанавливать дополнительное ПО. В режиме прямого контроля управление роботом осуществляется прямо через браузер.

3.2. Пользовательский интерфейс и телеметрия



На панель управления в реальном времени выводится видеопоток с камеры, что позволяет оператору управлять роботом вне зоны прямой видимости. Дополнительно на экран выводятся важные показатели телеметрии, такие как текущий режим работы (Ручной/Авто) и температура CPU микрокомпьютера. Взаимодействие клиента и сервера настроено асинхронно: отправка команд на двигатели и переключение режимов происходят в фоновом режиме через Fetch API, что исключает необходимость перезагрузки веб-страницы.

3.3. Сенсорная оптимизация (Устранение задержек)



При работе с ПК проблем не возникло, но в браузере смартфона обработка нажатий приводился к сбоям. Поэтому критически важной задачей при создании пульта управления на базе браузера было устранение задержек (пинга) при нажатии кнопок направления. При создании интерфейса стандартные программные клики мыши я заменила на прямую обработку касаний экрана (Touch Events). Эта оптимизация позволила добиться мгновенного отклика мобильного интерфейса, сопоставимого с аппаратными джойстиками и нормализации работы с мобильными дисплеями.

ЗАКЛЮЧЕНИЕ (ВЫВОДЫ И РЕЗУЛЬТАТЫ)

Проект по созданию мобильного робота с элементами искусственного интеллекта был сопряжен с рядом сложностей, но успешно прошел все этапы полевых испытаний. В ходе работы был создан полностью функционирующий прототип, который доказал эффективность выбранных инженерных и программных решений. Главная цель работы достигнута: робот способен не только управляться дистанционно, но и самостоятельно следовать за оператором по визуальным меткам (была даже реализована программа следования за Felis catus, путем прикрепления к нему ArUco-маркера). Разработанный кроссплатформенный веб-интерфейс обеспечил надежную систему удаленного управления с минимальными задержками как на ПК, так и на экране мобильного телефона.

Стратегии применения и масштабируемость

Созданный робот — это не просто игрушечная радиоуправляемая модель, это готовый программно-аппаратный прототип автономной транспортной платформы для закрытых помещений. Использование компьютерного зрения и системы ArUco-маркеров успешно решает фундаментальную проблему отсутствия или нестабильности спутникового сигнала GPS на крупных складах маркетплейсов и в промышленных цехах.

Данная технология визуальной навигации позволяет роботу ориентироваться в пространстве с высокой точностью и безопасно взаимодействовать с объектами без применения дорогостоящих лазерных дальномеров. Это значительно снижает

себестоимость каждого отдельного устройства по сравнению с промышленными аналогами, делая возможным массовое развертывание парков умных тележек даже на предприятиях малого и среднего бизнеса.

Помимо складской логистики, гибкость программного кода и кроссплатформенность веб-интерфейса по моему мнению, открывают следующие сферы применения технологии:

- **Медицина и социальная сфера:** Создание роботов-ассистентов для работы в стерильных зонах больниц и инфекционных отделениях. Робот может автоматически следовать за врачом, перевозя тяжелое диагностическое оборудование или медикаменты, а также использоваться как "умная" тележка для бесконтактной доставки еды пациентам, минимизируя риск заражения персонала.
- **Умный дом и автоматизация зданий:** Прототип может стать основой для создания персональных домашних помощников. Благодаря возможности работы в локальной сети, такие роботы способны интегрироваться с системами умного дома, осуществлять видеопатрулирование помещений, переносить предметы или работать в связке с другими IoT-устройствами внутри единой экосистемы здания.
- **Беспилотная авиация (БПЛА):** Использование визуальных маркеров, расположенных на корпусе наземного бота, для высокоточной автоматической посадки дронов на заданные, в том числе движущиеся, платформы. Это критически важно для создания мобильных зарядных станций и полностью автономных систем курьерской доставки.
- **Промышленность и интралогистика:** Синхронизация работы роботизированных платформ на заводских конвейерах со сборкой деталей. Автоматизированный подвоз тяжелых комплектующих от склада к рабочим станциям строго по расписанию или по вызову сотрудника (через захват персональной метки-идентификатора).
- **Сельское хозяйство (тепличные комплексы):** Навигация в крупных крытых агропромышленных комплексах, где сигнал GPS глушится металлическими конструкциями. Роботы могут следовать за сборщиками урожая, принимая на себя вес собранных овощей, или автономно вывозить ящики с готовой продукцией к зоне отгрузки.

- **Мониторинг и диспетчеризация:** Встроенная система удаленного управления посредством веб-приложения позволяет не только осуществлять прямой FPV-контроль, но и в перспективе организовать единый диспетчерский центр. Это даст возможность отслеживать перемещения всего парка роботов в реальном времени, анализировать их телеметрию и оптимизировать маршруты на предприятиях

Перспективы развития (Модернизация)

В будущем планируется внедрить ряд аппаратных и программных улучшений для повышения автономности и надежности системы:

1. Оптимизация питания: Замена текущего драйвера моторов на более современный и экономичный модуль DRV8833. Данная замена позволяет уменьшить вес и энергопотребление робота.

2. Контроль батарей: Интеграция полноценной платы BMS (Battery Management System) и датчиков уровня заряда для безопасной эксплуатации Li-ion аккумуляторов и реализации функции автоматического возврата на зарядную станцию.

3. Резервная безопасность: Добавление ультразвуковых датчиков (сонаров) в качестве дублирующей системы для предотвращения физических столкновений с объектами, на которых нет визуальных маркеров.

4. Внедрение нейросетей: В будущем я планирую заменить Raspberry Pi 4 на более мощную пятую версию или аналог с ИИ-ускорителем. Это позволит роботу отказаться от бумажных меток и использовать настоящие нейросети (например, YOLO). Он сможет распознавать силуэты людей, уверенно объезжать любые препятствия и даже понимать жестовые команды в реальном времени.

5. Интеграция в умный дом (IoT): Я хочу связать робота с системами автоматизации зданий по протоколу MQTT. Поскольку у меня уже есть опыт настройки серверов Home Assistant (моя Raspberry Pi ранее являлась сервером подобного умного дома), я планирую сделать робота полноценным узлом такой сети. В перспективе это позволит объединить несколько подобных тележек в единый «рой», где они будут сами обмениваться данными, координировать маршруты и распределять задачи без участия человека

ГЛОССАРИЙ (ТЕРМИНЫ И СОКРАЩЕНИЯ)

AGV (Automated Guided Vehicle) — автоматизированная транспортная тележка, перемещающаяся по заданному маршруту без участия человека (применяется в складской логистике).

ArUco-маркер — синтетическая квадратная метка с уникальным контрастным паттерном, используемая в компьютерном зрении для определения координат и ориентации объектов в пространстве.

BMS (Battery Management System) — электронная плата управления батареей, защищающая литий-ионные аккумуляторы от переразряда, перезаряда и короткого замыкания.

Computer Vision (Компьютерное зрение) — область искусственного интеллекта, связанная с анализом изображений и видео.

CSI (Camera Serial Interface) — высокоскоростной аппаратный интерфейс для прямого подключения модулей камер к микрокомпьютерам.

DC-DC преобразователь — электронное устройство, преобразующее постоянное напряжение одной величины в постоянное напряжение другой (в данном проекте — понижающее напряжение с 7.4В до стабильных 5В).

Flask — легковесный веб-фреймворк для языка Python, предназначенный для создания веб-приложений и локальных серверов.

FPV (First Person View) — вид от первого лица; трансляция видео с камеры робота на экран оператора в реальном времени.

GPIO (General-Purpose Input/Output) — интерфейс ввода-вывода общего назначения, позволяющий микрокомпьютеру управлять внешними электронными компонентами (моторами, датчиками, реле).

IoT (Internet of Things / Интернет вещей) — концепция вычислительной сети физических предметов («вещей»), оснащенных встроенными датчиками и ПО для взаимодействия друг с другом или с внешней средой без участия человека

MQTT (Message Queuing Telemetry Transport) — легковесный сетевой протокол обмена сообщениями, специально созданный для телеметрии и межмашинного взаимодействия (M2M) в системах интернета вещей в условиях нестабильных каналов связи.

OpenCV (Open Source Computer Vision Library) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов с открытым исходным кодом.

Raspberry Pi — одноплатный микрокомпьютер, обладающий процессором и оперативной памятью, работающий под управлением полноценной операционной системы семейства Linux.

YOLO (You Only Look Once) — архитектура высокоскоростных сверточных нейронных сетей, предназначенная для детекции, распознавания и классификации множества объектов на видео в режиме реального времени.

Лидар (LiDAR) — технология измерения расстояний путем излучения лазерных импульсов; используется для сканирования окружающего пространства.

П-регулятор (Пропорциональный регулятор) — алгоритм управления, формирующий управляющий сигнал, прямо пропорциональный отклонению текущей величины от целевой (в проекте используется для плавного подруливания робота).

ШИМ (Широтно-импульсная модуляция) — процесс управления мощностью, подводимой к электромотору, путем изменения скважности импульсов, что позволяет плавно регулировать скорость вращения колес.

Ссылки на видео демонстрацию:

https://disk.yandex.ru/i/8JQ_mCUT42CxUg

https://disk.yandex.ru/i/dgs2dP-XnfjGSw

https://disk.yandex.ru/i/KrElrjvzCOmklQ

Весь исходный код проекта, чертежи деталей и видео демонстрация работы алгоритмов опубликованы в открытом доступе в репозитории.

https://github.com/ezicvtumane/pi-smart-follower.git

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ

1. Юревич, Е. И. Основы робототехники: учебное пособие / Е. И. Юревич. — 4-е изд., перераб. и доп. — Санкт-Петербург: БХВ-Петербург, 2020. — 304 с. [Электронный ресурс]. – Режим доступа: https://elib.spbstu.ru/dl/325.pdf/download/325.pdf

2. Siegwart R., Nourbakhsh I. R., Scaramuzza D. Introduction to Autonomous Mobile Robots. — 2nd ed. — Cambridge: MIT Press, 2011. — 472 p.

3. Основы навигации в пространстве: методическое пособие / ГК Геоскан. [Электронный ресурс]. – Режим доступа: https://disk.yandex.ru/i/HxiQNz8YN4EqgQ

4. Официальная документация Raspberry Pi (Raspberry Pi OS, GPIO) [Электронный ресурс]. – Режим доступа: https://www.raspberrypi.com/documentation/

5. Документация библиотеки компьютерного зрения OpenCV (Open Source Computer Vision Library) [Электронный ресурс]. – Режим доступа: https://docs.opencv.org/

6. Документация микрофреймворка Flask (Python) [Электронный ресурс]. – Режим доступа: https://flask.palletsprojects.com/

7. Основы работы с модулем камеры Picamera2 [Электронный ресурс]. – Режим доступа: https://picamera.readthedocs.io/

8. Ковалева А. И. Исходный код, видео работы и документация проекта «Alice's Bot» (Репозиторий GitHub) [Электронный ресурс]. – Режим доступа: https://github.com/ezicvtumane/pi-smart-follower.git