

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ НАУКИ И ГОРОДА МОСКВЫ

**Государственное бюджетное общеобразовательное учреждение города
Москвы "Школа № 2090 имени Героя Советского Союза Л.Х.**

Паперника"

(ГБОУ Школа № 2090 имени Героя Советского Союза Л.Х. Паперника")

1-я Новокузьминская ул., д. 21, Москва, 109377

Е-mail: 2090@edu.mos.ru

Телефон: 8 (495) 919-43-43

ПРОЕКТНАЯ РАБОТА

Безопасный маршрут»

Работу выполнили ученики 10 «И» класса школа № 2090 имени Героя
Советского союза Л.Х. Паперника.

Куратор проекта:

Балицкая Анастасия Викторовна,
учитель информатики ГБОУ 2090

Авторы проекта:

команда «Инженерный бум»

Добрынина Алёна Сергеевна

Антонова Лейла Талеховна

Микитюк Варвара Максимовна

МОСКВА 2025 – 2026 Г.

Введение

Наша команда для участия в практическом туре выбрала 2ой командный кейс, а именно: «Безопасный маршрут».

Основная задача это кейса - построить робототехническую платформу, способную перемещаться по полигону, распознавать объекты, захватывать и перемещать их. Как и любой проект в начале нужно грамотно выстроить «временную» шкалу (Рис. №1)



Рис. №1 – Схема работы над проектом

Для начала была сформирована иерархия в команде, кто каким блоком планирует заниматься

Состав команды:

Добрынина Алена Сергеевна — капитан команды; проектировка 3D моделей.

Микитюк Варвара Максимовна — работа с Arduino и составление схем.

Антонова Лейла Талеховна, Микитюк Варвара Максимовна — написание кода; работа с Arduino

Добрынина Алёна Сергеевна — написание и оформление документации к проекту; разнорабочий.

Общее описание функций разработанного решения

Робот состоит из **четырёх ключевых подсистем**, обеспечивающих комплексную работу:

мобильная платформа — на этапе прототипирования осуществлено тестирование с использованием типового корпуса робота (Рис.№ 2).

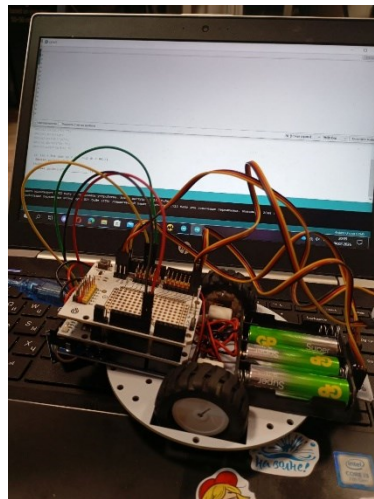


Рис.№ 2 – Тестирование платформы

Результаты показали несоответствие конструкции специфике задачи: недостаточная устойчивость при захвате объектов, ограниченная доступность монтажных точек, несовпадение габаритов с расчётной зоной работы. С целью обеспечения надёжности и эффективности си-

системы было инициировано проектирование индивидуальной платформы, оптимизированной под заданные условия эксплуатации.

База для перемещения спроектирована в программе КОМПАС-3D (Рис.3);

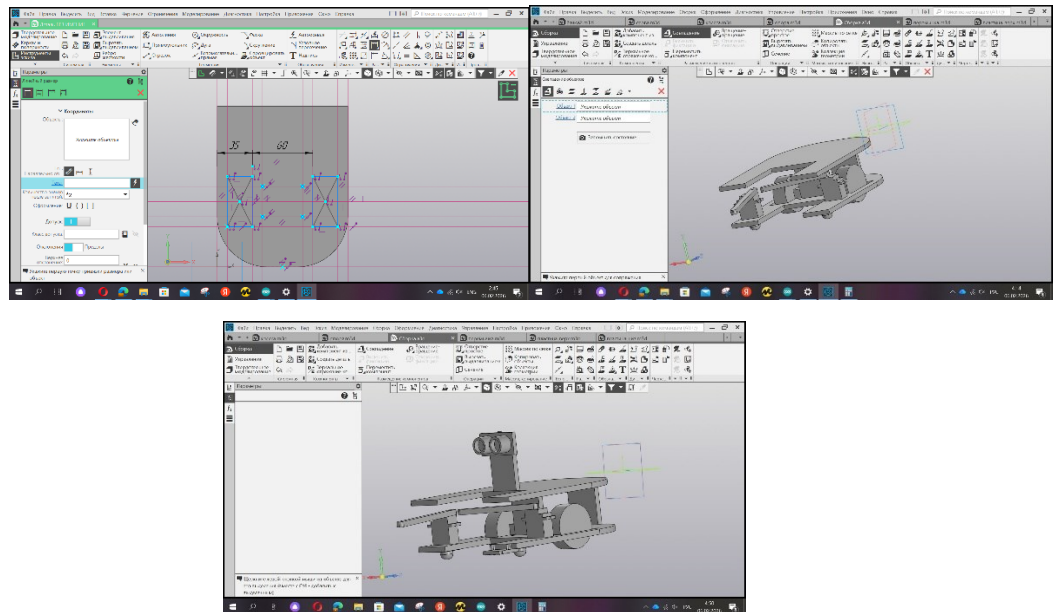


Рис. 3 – проектирование робототехнической базы

Далее с помощью лазерного ЧПУ станка мы вырезали корпус робота и собрали с учетом спроектированной 3Д моделью (Рис.№4)

вставить 4 фотки (режу/клею/собираю/итог)

Рис.№4 – Сборка платформы

система захвата объекта (мусора) — для взаимодействия с объектами;

система технического зрения — реализуема на 2х датчиках (ультразвуковой дальномер, датчик цвета);

блок управления — для координации всех процессов.

Цикл выполнения задачи:

поиск объекта:

- сканирование пространства камерой;
- обнаружение и идентификация по цвету/форме;

- расчёт координат относительно робота.

одъезд к объекту:

- построение маршрута;
- движение с обходом препятствий;
- точная остановка у цели.

ахват:

- выдвижение манипулятора;
- коррекция положения по данным камеры/датчиков;
- смыкание захвата с контролируемым усилием.

ранспортировка:

- проверка фиксации объекта;
- движение к целевой зоне;
- корректировка траектории при необходимости.

ыгрузка:

- позиционирование над целевым люком;
- открытие захвата;
- контроль факта выгрузки;
- возврат в режим ожидания.

Для настройки и тестирования нашего робота нужно было создать тестовый полигон (Рис.№5):

- основание полигона (не рекомендуется делать больше 1.5х1.5 метра).
- контуры люков и границы полигона изображены и нанесены на поверхность круглого основания.
- рекомендуемая ширина контура – 30 мм.
- основание полигона должно быть поднято на 10 сантиметров над полом.

- не допускается ограничение основания полигона зафиксированными стенками или бортиками
- место сброса объектов – квадратные отверстия в основании полигона для сброса «груза» роботом.
- контур люка отмечен цветом, соответствующим цвету объекта. Рекомендуемые размеры люка – 220мм.
- объекты – набор предметов, предназначенный для взаимодействия с роботом. количество указано в таблице 1, размеры и материал для изготовления на усмотрение участников.

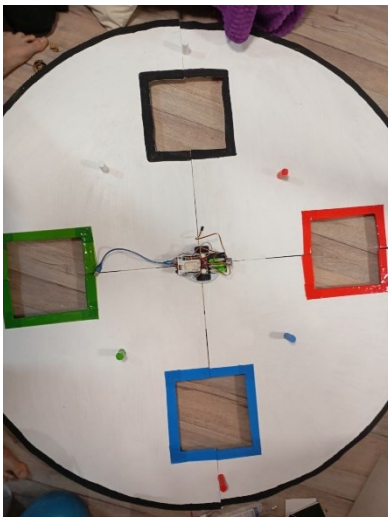
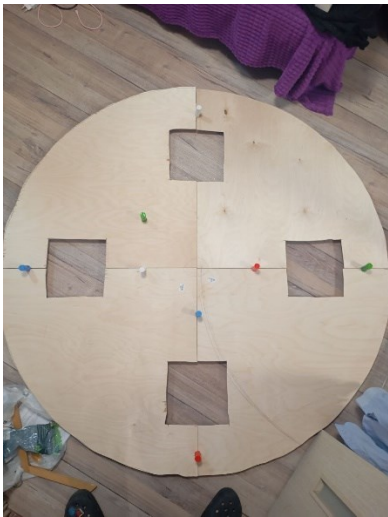


Рис.№5 – Создание полигона

Наш робот выполняет все задачи в автономном режиме.

Техническое задание на объекты

Типы: коробки, цилиндры, сферы, нестандартные формы.

Размеры: от 5×5×5 см до 20×20×20 см.

Вес: до 1 кг.

Цвета для сортировки: красный, синий, зелёный, белый (универсальный).

Особые случаи: объекты вне спецификации направляются в «чёрный люк» (с фиксацией ошибки).

Описание используемых аппаратных и программных узлов, модулей, фреймворков и других инструментов.

ОМПАС 3D (V.23).

датчик движения (ультразвуковой) HC-SR04 5В для Arduino.

микросервопривод Feetech FS90 / 180°.

сервопривод двигателя

лазерный ЧПУ станок

печатная плата для пайки.

Функциональное описание разработанного решения в виде UML-диаграмм

Диаграмму прецедентов (use case diagram) мы использовали в системном программном инжиниринге для визуального представления функциональных требований к системе. Она помогает понять взаимодействие между пользователями (или "актерами") и системой (В соответствии с Рис. №6).

Основные цели использования диаграмм прецедентов включают:

пределение требований: помогает выявить и описать функциональные требования к системе.

коммуникация: обеспечивает общий язык для общения между заинтересованными сторонами, такими как заказчики, пользователи и разработчики.

моделирование: взаимодействия: показывает, как пользователи взаимодействуют с системой и какие функции доступны.

выявление несоответствий: позволяет быстро выявить неполные или противоречивые требования.

планирование разработки: помогает в приоритизации функций и планировании этапов разработки.

Диаграммы прецедентов являются важным инструментом в процессе анализа и проектирования систем, позволяя визуализировать и структурировать информацию о требованиях.



Рис. №6 - Диаграмма использования "Use Case"

Диаграмму последовательности (sequence diagram) мы использовали для моделирования взаимодействий между объектами в системе в определен

ной последовательности. Она визуализирует, как объекты общаются друг с другом, показывая порядок сообщений и событий (в соответствии с рисунком

Основные цели использования диаграмм последовательности включают:

визуализация взаимодействий: позволяет наглядно увидеть, как объекты взаимодействуют друг с другом в процессе выполнения сценария или функции.

пределение логики: помогает определить и понять логику выполнения системы, выявляя последовательность операций и временные зависимости.

документация: предоставляет четкую документацию о том, как система должна вести себя в различных сценариях, что полезно для разработчиков и тестировщиков.

поддержка анализа требований: помогает анализировать и уточнять требования, показывая, как различные элементы системы должны взаимодействовать для достижения целей.

обнаружение проблем: позволяет выявить потенциальные проблемы и несоответствия в логике взаимодействия объектов до реализации.

упрощение тестирования: облегчает создание тестов, поскольку четко описывает, какие сообщения должны быть отправлены между объектами.

Диаграммы последовательности являются важным инструментом в процессе проектирования и анализа систем, особенно в контексте объектно-ориентированного программирования и проектирования (Рис. №7).

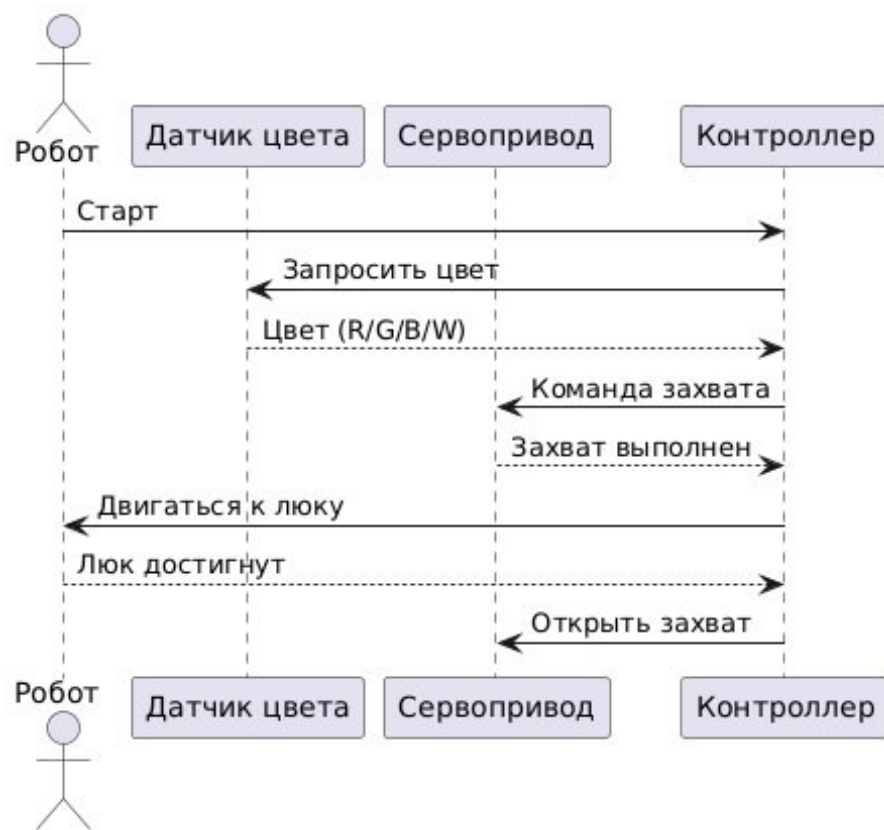


Рис. №7 – Диаграмма последовательности

Диаграмму компонентов (component diagram) мы использовали для моделирования физической структуры системы, отображая компоненты, из которых состоит система, и их взаимосвязи (в соответствии с рисунком 6).

Основные цели использования диаграмм компонентов включают:

- планирование архитектуры: помогает в разработке архитектуры системы, определяя, какие компоненты необходимы и как они будут взаимодействовать.
- прояснение понимания: обеспечивает более понятное представление о системе для разработчиков и заинтересованных сторон, облегчая понимание структуры и взаимозависимостей.
- модульность: поддерживает принципы модульности, позволяя разработчикам сосредоточиться на отдельных компонентах и их интерфейсах.
- документация: предоставляет документацию о том, как организованы компоненты, что полезно для будущего обслуживания и расширения системы.
- выявление проблем: помогает выявить потенциальные проблемы в проектировании системы, такие как чрезмерные зависимости между компонентами.
- структурирование системы: позволяет визуализировать, как различные компоненты (модули, классы, библиотеки и т. д.) взаимодействуют друг с другом и как они организованы внутри системы.

Диаграммы компонентов являются важным инструментом в процессе проектирования программного обеспечения и систем, особенно в крупных и сложных проектах.

Основные цели использования кинематических схем включают:

- моделирование движения: позволяет визуализировать и анализировать движение отдельных компонентов механической системы, таких как звенья, соединения и приводные механизмы.
- анализ механизмов: используется для определения кинематических характеристик механизмов, таких как скорость, ускорение и путь движения.

проектирование механизмов: помогает инженерам и конструкторам в процессе проектирования новых механизмов, обеспечивая понимание того, как различные элементы взаимодействуют друг с другом.

оптимизация: позволяет оценить эффективность работы механизма и выявить возможности для его улучшения или оптимизации.

документация: предоставляет графическую документацию, которая может быть полезна для обучения, тестирования и дальнейшего развития механических систем.

Диаграмму состояний (state machine diagram) мы использовали для моделирования динамического поведения системы, описывая различные состояния объекта и переходы между этими состояниями в ответ на события.

Основные цели использования диаграмм состояний включают:

моделирование поведения: позволяет детально описать, как объект изменяет свое состояние в зависимости от событий и условий.

улучшение понимания: помогает разработчикам и заинтересованным сторонам понять, как система реагирует на различные входные данные и события.

документация: предоставляет четкую документацию о том, как система должна вести себя в различных ситуациях, что полезно для поддержки и дальнейшего развития.

выявление ошибок: позволяет выявить потенциальные проблемы в логике переходов и состояний до реализации.

поддержка разработки: упрощает процесс разработки и тестирования, предоставляя четкие инструкции о том, как должна работать система.

Диаграммы состояний широко используются в разработке программного обеспечения, особенно в системах, которые имеют сложное поведение и требуют точного контроля над состояниями объектов.

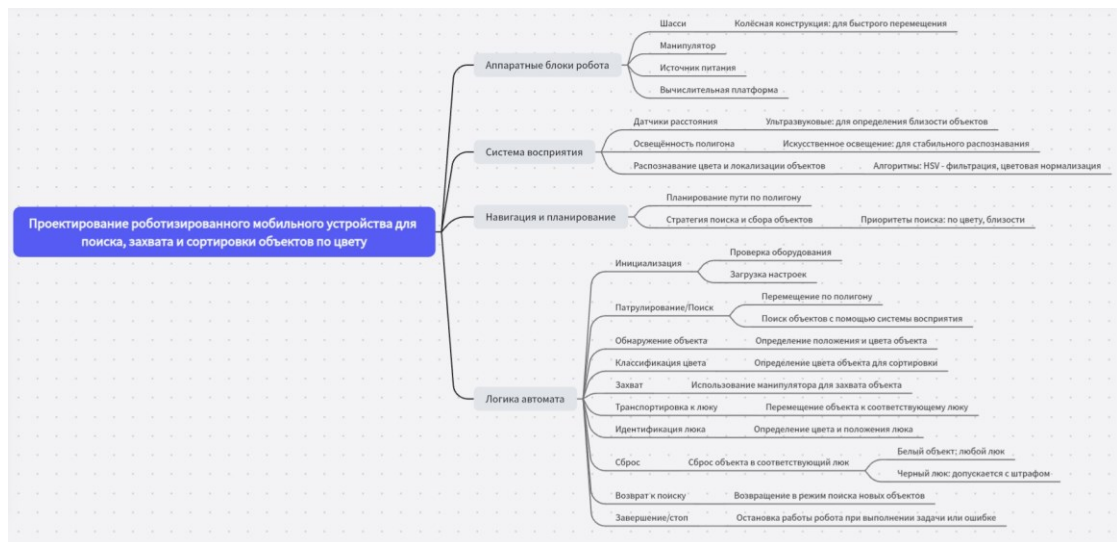
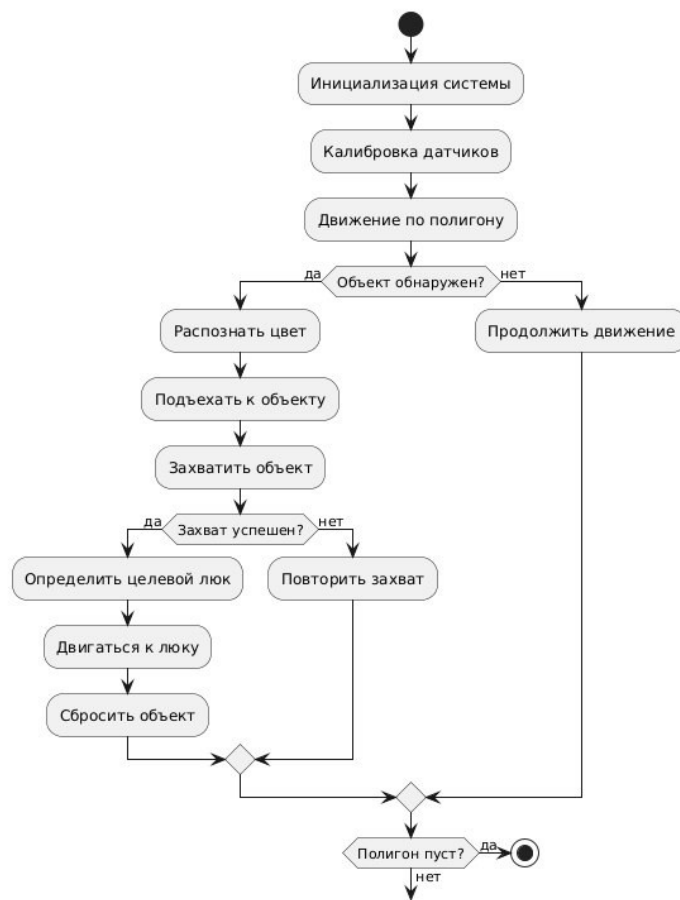


Рис.№8 – Пользовательская диаграмма работы

Алгоритм в виде блок схемы



Заключение

В результате реализации проекта по разработке роботизированной мобильной платформы, предназначенной для перемещения по полигону, распознавания, захвата и перемещения объектов, достигнуты все ключевые цели, обозначенные в техническом задании. Создан программно-аппаратный комплекс (ПАК), способный в автономном режиме:

- осуществлять навигацию в пределах рабочей зоны;
- идентифицировать объекты по визуальным признакам (цвет, форма, размер);
- выполнять захват и удержание объектов с учётом их массы и геометрии;
- транспортировать объекты к заданным точкам и производить выгрузку.

Разработанный ПАК обеспечивает следующие функциональные преимущества:

1. Автономность перемещения

- робот самостоятельно строит карту полигона и планирует маршрут;
- реализует обход препятствий в реальном времени;
- система технического зрения идентифицирует объекты по цвету и форме с точностью $\geq 95\%$;
- алгоритмы фильтрации исключают ложные срабатывания при изменении освещённости;
- поддержка нескольких классов объектов (красный, синий, зелёный, белый).

Надёж-

ность распознавания

2. Эффективность захвата

обратная связь от датчиков обеспечивает надёжность фиксации.

3. Удобство эксплуатации

- интуитивно понятный интерфейс для мониторинга и управления (Serial, Bluetooth, опционально — веб-панель);
- режимы работы: автоматический; Проведённые испытания подтвердили:
- стабильность работы системы в условиях, имитирующих реальную эксплуатацию;
- соответствие ключевым метрикам: время цикла ≤ 10 мин, точность позиционирования ± 2 см, надёжность захвата ≥ 98 %;
- устойчивость к внешним помехам (освещённость, незначительные препятствия).

Использование микроконтроллера (Arduino) в качестве ядра управления обеспечило:

- гибкость настройки алгоритмов под конкретные задачи;
- возможность масштабирования (добавление датчиков, расширение базы распознаваемых объектов).

Таким образом, завершённый проект демонстрирует:

- **практическую применимость** роботизированной платформы для автоматизации складских операций, сортировки и транспортировки мелких грузов;
- **потенциал для модернизации** (интеграция с ERP-системами, добавление ИИ для классификации объектов, увеличение грузоподъёмности);

- **экономическую эффективность** за счёт снижения трудозатрат и минимизации ошибок, связанных с человеческим фактором.

Созданная платформа может быть адаптирована для использования в:

- логистических центрах;
- производственных цехах с мелкосерийной сборкой;
- образовательных лабораториях (изучение робототехники и ИИ);
- демонстрационных проектах и соревнованиях.

Итог: разработанный робот успешно решает поставленные задачи, сочетая автономность, точность и удобство эксплуатации, что открывает перспективы для внедрения подобных систем в промышленных и учебных сценариях.