

Российская научно-социальная программа
для молодежи и школьников «Шаг в будущее»

Международный форум научной молодёжи «Шаг в будущее»
(Россия, Москва, 25 марта- 30 апреля 2024 г.)

ПРОМЫШЛЕННЫЙ РОБОТ-ПОМОЩНИК ROBOTRACK. РОБОТИЗИРОВАННАЯ УБОРКА ГОРОДСКОЙ СРЕДЫ

Авторы:

Герасименко Георгий Андреевич
Россия, Омская область, г. Омск
БОУ г. Омска "Лицей БИТ", 8 класс

Научный руководитель:

Шайхетдинов Михаил Альфредович,
Педагог дополнительного образования
Технопарк "Кванториум" (квантум Хайтек)
Аллагулов Станислав Сайфуллович,
Педагог дополнительного образования
Технопарк "Кванториум" (IT-квантум)

Я, Шайхетдинов М.А., подтверждаю, что данный проект содержит не более 22 страниц, из них текст статьи и список литературы содержат не более 11 страниц, приложения – не более 10 страниц

подпись

ПРОМЫШЛЕННЫЙ РОБОТ-ПОМОЩНИК ROBOTRACK. РОБОТИЗИРОВАННАЯ УБОРКА ГОРОДСКОЙ СРЕДЫ

Герасименко Георгий Андреевич

Омская область, г.Омск, БОУ г. Омска "Лицей БИТ", 8 класс

Аннотация

В городах много мусора, а в моем городе зимой ещё и много снега. Если не поддерживать чистоту городского пространства, то город не только потеряет визуальную привлекательность, но и станет опасным для проживания. Поэтому у меня возникла идея разработать автономного робота-помощника, который бы помогал коммунальным работникам поддерживать чистоту городского пространства.

Для реализации своей задумки я разработал в ПО Компас 3D: гусеничную тележку, гусеничное полотно, рассчитать силовую установку, спроектировал механизм подъема отвала на винтовой передаче, спроектировал роботизированную руку, установил камеры и датчики.

Рабочий прототип изготовил с помощью ЧПУ-фрезерного станка и 3D принтера. Управление роботом осуществляется как с помощью радиопульта, так и в автономном режиме с помощью машинного зрения. FPV камера позволяет передавать изображение на шлем оператора.

Робот Robotrack имеет огромный потенциал использования для автономной круглосуточной поддержки городской среды в чистом виде. Когда я внедрю искусственный интеллект, робот станет самостоятельным и один человек сможет контролировать целую группу роботов.

Я уверен, что мой ROBOTRACK будет приносить пользу людям, т.к. чистый город – это красиво и безопасно!



Ключевые слова: Промышленный робот-уборщик, 3D моделирование, проектирование, гусеничная платформа, FDM печать, ЧПУ-фрезеровка, программирование на Arduino/Python, машинное зрение, FPV камера.

Введение

В городах много мусора, а в моем городе зимой ещё и много снега. Если не поддерживать чистоту городского пространства, то город не только потеряет привлекательность, но и станет опасным для проживания горожан. Поэтому у меня возникла идея разработать автономного робота-помощника, который бы помогал коммунальным работникам поддерживать чистоту городского пространства.

Зимой и летом в общественных зонах города очень остро стоит вопрос своевременной уборки тротуаров и территории от снега, пыли и мусора. Сейчас за чистотой следят дворники. Промышленный робот-помощник Robotrack позволит заменить тяжелый человеческий труд работой машины. Дворники-операторы будут выдавать задания роботам-уборщикам и следить за работоспособностью. В случае поломки или другой нештатной ситуации, робот-уборщик пришлет уведомление и фотографию, чтобы человек принял меры.

Анализ ситуации

Новая техника для ЖКХ позволит сократить расходы на содержание коммунальных работников, группа промышленных роботов-помощников Robotrack будет способна сама убирать улицы в автономном режиме. Будет требовать небольшая команда дворников-операторов для того, чтобы держать в чистоте весь город!

На текущий момент вся коммунальная техника или дизельная или бензиновая, мой робот-помощник оснащен электромоторами и является экологичной техникой идеально вписываясь в городскую среду. Чистые тротуары — это не только безопасно, но и красиво!

Решение проблемы

Я проанализировал существующие конструкции и решения и выработал свою стратегию по разработке робота-помощника Robotrack:

1. Проектирование гусеничной платформы, роборуки и навесного оборудования

Я спроектировал робота-помощника в САПР Компас 3D. Получилось больше 50 деталей, которые я собрал в сборку и получил 3D модель робота-помощника.

2. Разработка электрической схемы и сборка электроники

Мною была разработана схема на базе arduino и raspberry pi.

3. Изготовление деталей и сборка функционального прототипа

Большие детали робота-помощника я изготовил на ЧПУ фрезере, остальные распечатал на 3D принтере. Запрограммировал arduino+raspberry pi, реализовав движение с помощью машинного зрения по цветной линии или за целеуказателем.

Техническое описание механизмов робота-помощника

Разработка гусеничного трака

Промышленного робота спроектировал на гусеничном ходу для повышения проходимости и манёвренности [2, с.7]. Обрезиненные гусеничные траки гораздо эффективнее резинового колеса.

Гусеничный трак спроектировал с грунтозацепом специальной формы, который обеспечивает максимальную эффективность. После серии неудачных экспериментов по обрезиниванию трака в домашних условиях траки были установлены в первоначальном виде. Шип на траках, который входит в зацеп с ведущим колесом я сделал съемным, это позволит в случае поломки заменить только шип, без замены всего трака. Трак представлен на рис.1.

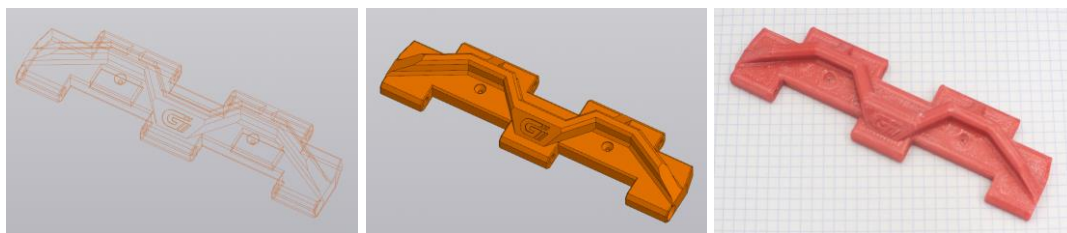


Рис.1

Всё проектирую в САПР Компасе 3D [1, с.3], детали собираю в сборку, если всё ок, то печатаю на 3D принтере и собираю.

Также я спроектировал узел автоматической натяжки полотна с помощью самонатяжных роликов с применением пружин. Конструкция вышла простой и надежной. Кроме того, в гусеничной тележке имеются два опорных катка.

Силовая установка. Двигатели и планетарные редуктора

За основу ведущего колеса гусеничной тележки были взяты BLDC-моторы от гироскутера. Моторы модифицировал, просверлил крепежные отверстия и нарезал резьбу М4 для крепления специализированных зубчатых накладок, так у меня получились зубчатое ведущие колеса, см.рис2. Выбор моторов был не случаен, для уверенного и плавного движения требуются моторы с высоким крутящим моментом на низких оборотах. BLDC-моторы от гироскутера подходят для этих целей идеально, низкий kV^1 , высокая мощность и высокий крутящий момент за счет датчиков Холла.

Робот Robotrack получился очень маневренным, для разворота требуется не больше максимальной характеристики габаритов, в моем случае длины робота-помощника. Благодаря высокому крутящему на низкой скорости вращения я смог отказаться от применения дополнительных редукторов в гусеничных движителях.

¹ kV - число оборотов в минуту, с которым двигатель будет вращаться при подаче одного вольта



Рис.2

Разработка шасси

Шасси - основной силовой элемент, на который приходится все нагрузки, также он является пространственной рамой. Изначально я планировал шасси изготовить из алюминия Д16Т, но финансово меня никто не поддержал и поэтому выбор пал в сторону 10мм многослойной фанеры. Детали я изготовил на ЧПУ-фрезере, для этого я прошел большой путь и научился не только проектировать детали под ЧПУ изготовление, но и создавать управляющие программы в ArtCAM, подбирать фрезы и режимы работы для них.

Шасси спроектировал с шиповым соединением, это делает его более жестким и прочным. После непосредственной фрезеровки рамы потребовалась механическая обработка на ленточном шлифовальном станке и напильником [3, с.14], а также сборка на специальные метизы, см.рис.3.

Сквозь раму проходят две алюминиевые трубы, выполняющие роль держателей гусеничных движителей. Специально спроектированные муфты надежно скрепляют всю конструкцию, делая её монолитной.



Рис.3

Разработка роботизированной руки

Роборука – это многоцелевая роботизированная манипуляторная установка с захватом. Изучил учебное пособие по роботизированным системам [4, с.8] и спроектировал роборуку с тремя степенями свободы №1, №2, №3 (см.рис.5) и редукторной клешней захвата №4 (см.рис.6), это позволяет использовать маломощный сервопривод для удерживания больших предметов. Роборука представлена на рис.6. До изготовления мною было произведено математическое моделирование в ПО ARTAS SAM механики роборуки, для того чтобы оценить рабочие характеристики, результаты представлены на рис.4.

Максимальный вертикальный вылет получился 576мм, максимальный горизонтальный вылет 366мм. Весь рабочий диапазон изображен на рис.4 оранжевым цветом. Этих характеристик достаточно для выполнения задач по сбору, перекладыванию и транспортированию крупных предметов.

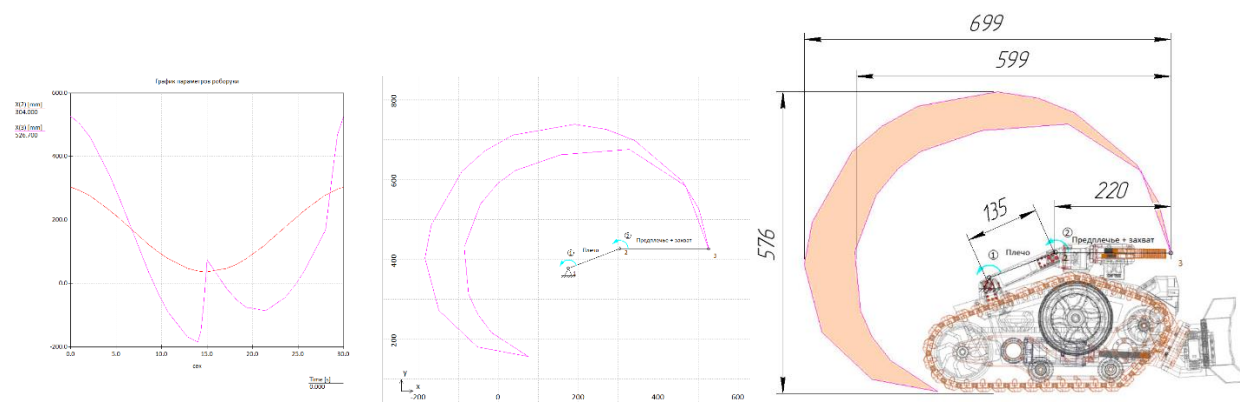


Рис.4

Также я предусмотрел поворотную ось №3 (см.рис.5) для захвата, что помогает захватывать предметы любой формы. Максимальный поворот составляет $\pm 100^\circ$

Редукторный захват №4 (см.рис.5) на роботизированной руке является съемным, его можно заменить на другой захват в зависимости от целей. На текущий момент я спроектировал редукторный захват в виде клешни.

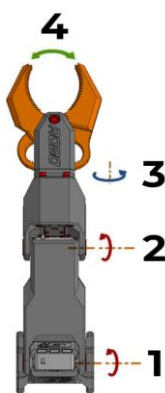


Рис.5

В качестве сервоприводов были использованы сервоприводы с двигателем постоянного тока и максимальным крутящим моментом 25кг*см. Такой сервопривод способен удерживать груз до 25кг на рычаге длиной 1см от вала сервопривода. Зная длины и все всех сегментов роботизированной руки, я провел расчет полезной нагрузки, которую робот может перемещать в пространстве.

Расчет максимальной полезной нагрузки роборуки

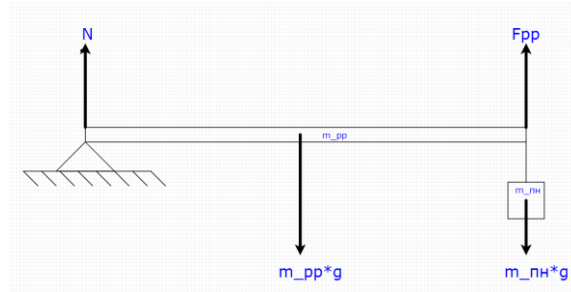
Так как я использую одинаковые сервоприводы, то максимальную нагрузку определяет сервопривод, который стоит на максимальном рычаге, это сервопривод плеча №1 (см.рис.5)

$$M_{\text{сервопривода}} = 2,45 \text{ Н} * \text{м} \text{ (момент на валу сервопривода)}$$

$$l = 0,36 \text{ м} \text{ (длина роборуки)}$$

F_{pp} – сила требуемая роборуке для поднятия груза массой $m_{\text{пн}}$

$$m_{\text{pp}} = 0,5 \text{ кг} \text{ (масса роборуки без полезной нагрузки)}$$



Составим уравнение суммы моментов относительно точки крепления роборуки

$$m_{\text{pp}} * g * \frac{l}{2} + m_{\text{пн}} * g * l = N * 0 + F_{\text{pp}} * l$$

$$M_{\text{сервопривода}} = F_{\text{pp}} * l$$

$$m_{\text{пн}} = \frac{M_{\text{сервопривода}}}{g * l} - \frac{m_{\text{pp}}}{2}$$

$$m_{\text{пн}} = \frac{2,45}{9,8 * 0,36} - \frac{0,5}{2} = 0,44 \text{ кг}$$

Таким образом мы получили, что максимальная полезная нагрузка составляет 440 гр. Этот показатель легко можно увеличить в 2,9 раза заменив сервопривод плеча на более мощный, с максимальным крутящим моментом 5,36 Н*м (55кг*см)

В ходе проведения испытаний выяснилось, что угловые ускорения и скорости работы сервоприводов слишком высокие. При захвате полезной нагрузки близкой к максимальной возникали сложности в управлении из-за инерции и перегреву сервоприводов. Самым высоконагруженным сервоприводом оказался самый нижний, он у меня сгорал дважды на испытаниях. Выходы из строя сервоприводов меня навели на мысль, что нужно как-то ограничивать ток или ограничивать ускорения и скорости. Идеальным решением проблемы стала бы установка дополнительных редукторов, но для этого нужно было бы менять сервоприводы на шаговые двигатели, т.к. максимальный ход у сервоприводов 180 градусов и переделываться концептуально всю роборуку. Но для первой версии я счел достаточным использование специальной библиотеки для сервоприводов, в которой можно ограничить ускорение и скорость сервопривода. Результат меня устроил, роборука стала медленнее и плавнее перемещаться с грузом.

Еще одной особенностью роборуки является скрытая проводка, я очень много времени уделил кабель-менеджменту, в текущем исполнении роборука и выглядит красиво и защищена от механических воздействий при перевозке крупных грузов, см рис. 6.

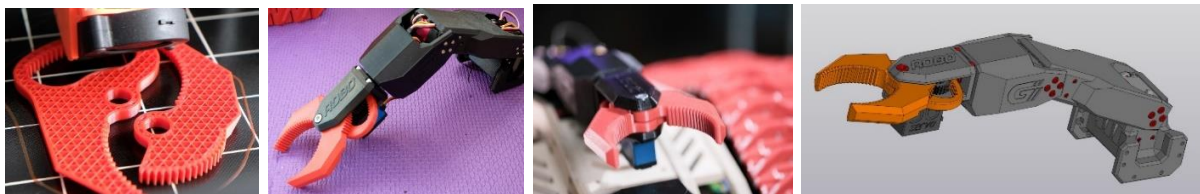


Рис.6

Разработка отвала с изменяемой геометрией и механизма подъема

Дополнительным механизмом робота-помощника является отвал и механизм подъёма. При проектировании автономной системы чистки снега я уделил внимание универсальности, и сделал отвал с изменяемой геометрией, изображенного на рис.7.

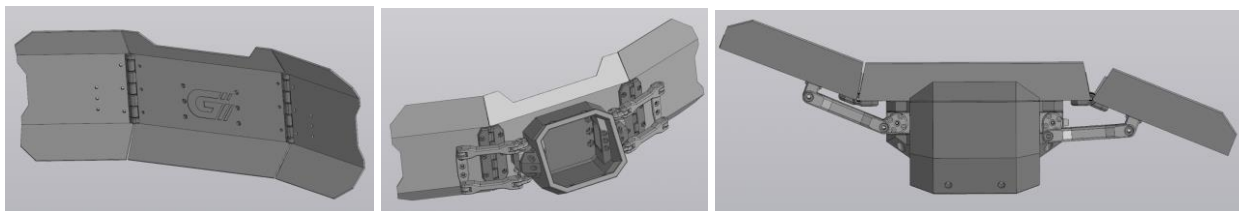


Рис.7.

На отвале боковые сегменты имеют шарнирное соединение, к которым подключены тяги сервоприводов. Изменяемый угол боковых сегментов позволяет реализовать отвал такой формы, чтобы сгрести снег влево или вправо по ходу движения, а также сгрести в кучу или пробивать тропинку как ледокол, отбрасывая снег по обе стороны отвала.

Сам отвал является съемным навесным оборудованием, поэтому я спроектировал быстросъемный механизм. Навесное оборудование устанавливается на нижний подъемный механизм, который позволяет регулировать высоту подъёма отвала относительно поверхности, по которой движется робот-помощник. Механизм подъема реализован на винтовой передаче, см. рис.8. В качестве винтовой передачи я использовал обычный винт М14 длиной 120мм и гайку. Винт вращает коллекторный мотор из шуруповерта. Для понижения оборотов и увеличения тяги из шуруповерта также взял шестеренки планетарного редуктора и втулку, спроектировал для мотора и шестерёнок редуктора новый единый корпус, так у меня получился мотор-редуктор.

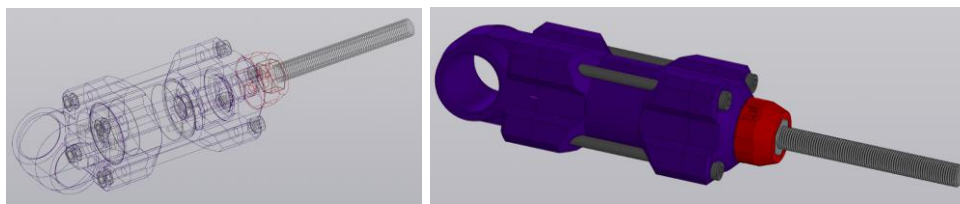


Рис.8

Я предусмотрительно сделал корпус влагозащищенным, т.к. располагается он под днищем робота. Корпус мотор-редуктора собрал на герметик. Для гайки был спроектирована

ответная деталь, которая и перемещала быстросъемный механизм от самого низкого, до самого высокого положения.

В будущем я изготовлю отвал из металла и спроектирую дополнительное навесное оборудование: турбощетку, щетку для мойки бордюров или даже газонокосилку

Дизайнерское решение

Промдизайн при проектировании занимает далеко не последнее значение, поэтому я переделываю эскиз по несколько раз, провожу опрос среди своих друзей и одноклассников, и только потом как начинаю изготавливать. Техника должна гармонично вписываться в городскую среду и радовать взгляд прохожих.

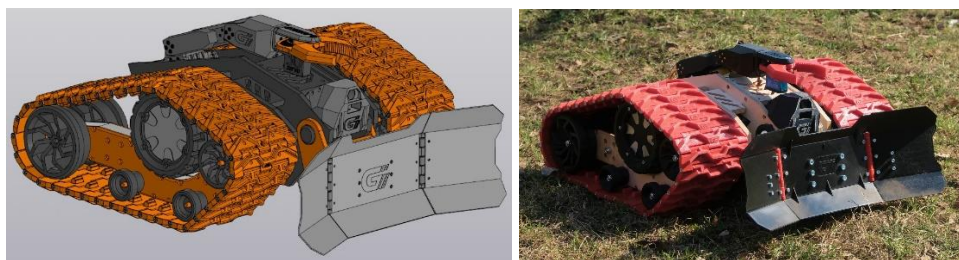


Рис.9

Шасси покрасил покрыл водоотталкивающим раствором. Роботизированную руку и гусеничные траки напечатал пластиком из экологического ПЭТГ² (Рис.9).

Проектирование схемы электроники и программирование

Схему собирал на Arduino, Raspberry pi, web-камере, регуляторе BLDC-моторов, инфракрасном дальномере, также взял передатчик и камеру FPV³. Электрическая схема приведена в прил.2. Установлена кнопка «СТОП» для аварийного отключения робота.

Скетч для Arduino писал в Arduino IDE, писал очень долго, читал книжку [6, с.250], смотрел ролики в youtube, боролся с багами. Скрипт для Raspberry PI 4 писал ещё дольше. С использованием библиотеки OpenCV [5] удалось реализовать распознавание цвета, выделение области контуром, поиск центра контура и передача этих координат на Arduino для дальнейшего управления моторами.

Сборка, тестирование. Экспериментальная часть.

Большую раму я отфрезеровал на ЧПУ-фрезере из 10мм фанеры, все пластиковые элементы распечатал на своём 3D принтере пластиком ПЭТГ. Процесс печати деталей очень

² Полиэтилентерефталатгликоль

³ First Person View — вид от первого лица

сильно растянулся во времени из-за медленной технологии FDM⁴. Не все детали подходили с первого раза, даже если в сборки САПР Компас3D всё выглядело отлично. Т.к. пластик ПЭТГ имеет усадку после печати, то ее тоже нужно было учитывать, иначе либо зазоров не было, либо были большие. С опытом начинаешь понимать, как правильно проектировать детали для печати на 3D-принтере и становиться меньше брака и потерянного времени.

После того как был собран робот-помощник Robotrack я отправился на испытательный полигон, там были выявлены слабые узлы. Самым большим моим просчетом является малое количество зубьев одновременного зацепа с ведущим колесом, приходится держать гусеницу в хорошо натянутом состоянии, что существенно увеличивает энергопотребление и снижает автономность работы. Также была переработана поворотная ось роботизированной руки, которая оказалось слабой и не выдерживала расчетную нагрузку.

Мне нравится то, чем я занимаюсь, и я делюсь этим со зрителями через свой youtube канал «ТехноМейкер». На канале у меня выложены видеоролики со сборкой и тест-драйвом робота-помощника Robotrack [7].

Результаты проекта

Промышленный робот-помощник Robotrack получился очень маневренным, функциональным и красивым. На одном заряде АКБ робот способен работать до 1 часа. Robotrack уже сейчас имеет автономные функции движения по линии или за целеуказателем с помощью машинного зрения, также может управляться с помощью радиоаппаратуры как в прямой видимости, так и через FPV-шлем. Полные технические характеристики представлены в Прил.1.

Перспективы проекта

Уже многое сделано. Я с промышленным роботом-помощником Robotrack стал победителем на Всероссийском конкурсе изобретателей ШУСТРИК 2023 (см. Прил. 7), я заработал диплом I степени и золотую медаль на Международный конкурс научно-исследовательских и инженерно-технических проектных работ «Ученый будущего 2023» (см. Прил. 5), меня пригласили в Суперфинал и я уже подготовил тезисы на английском языке (см. Прил. 4). Получил диплом за I место на Международном фестивале по робототехнике «Робофинист 2023» (см. Прил. 6). Также я принял участие с роботом III Конгрессе молодых ученых в Сириусе, где представил свой проект первым лицам страны и получил благодарственное письмо от Росконгресса (см. Прил. 8).

⁴ Fusion Deposition Modeling и переводится на русский как «моделирование методом послойного наплавления»

Мне еще очень много предстоит сделать. Я ищу программиста в команду, чтобы ускорить разработку проекта, т.к. сам я больше люблю моделировать и создавать новые механизмы и машины. Хочу сделать робота автономным и безопасным, для этого нужно реализовать целый комплекс мер безопасного движения, начиная от установки вращающегося лазерного лидара для построения цифрового пространства до внедрения искусственного интеллекта для анализа информации с датчиков и принятия решений.

Коммерциализация

В рамках акселератора «Технолидеры будущего» я провел экономический анализ, сравнение с конкурентами и подсчитал емкость рынка. Если промышленного робота-помощника Robotrack вывести на рынок, то можно заработать 5 млрд.руб, см.Табл.1.

Таблица 1. Оценка объема рынка

Сверху вниз	Снизу вверх
TAM - 102,0 млрд руб. Рынок уличных клининговых услуг России SAM – 51,0 млрд руб. 30% услуг оказывается для BTG (государства) 10% услуг оказывается для BTC (бизнес) 10% услуг оказывается для BTV (частных лиц) SOM – 5.1 млрд руб. В конкурентной среде продукт займет 5%	SOM – 1.08 млрд руб. В конкурентной среде продукт займет 5% при цене в 220 000руб. SAM – 10.8млрд руб. 30% услуг оказывается для BTG (государства) 10% услуг оказывается для BTC (бизнес) 10% услуг оказывается для BTV (частных лиц) TAM – 21.6 млрд руб. Рынок уличных клининговых услуг России. 180000 ед.
https://businessstat.ru/images/demo/cleaning_russia_demo_businessstat.pdf https://rb.ru/news/smart-vacuum-cleaner/	

Заключение

Подводя итоги создания промышленного робота-помощника Robotrack, виден огромный потенциал использования его для поддержания чистоты как городского пространства, так и частных территорий. Робот Robotrack будет делать рутинную работу по уборке территорий качественно и в любую погоду, в итоге человек будет получать красивую и безопасную окружающую среду.

Я уверен, что мой Robotrack будет приносить пользу людям!

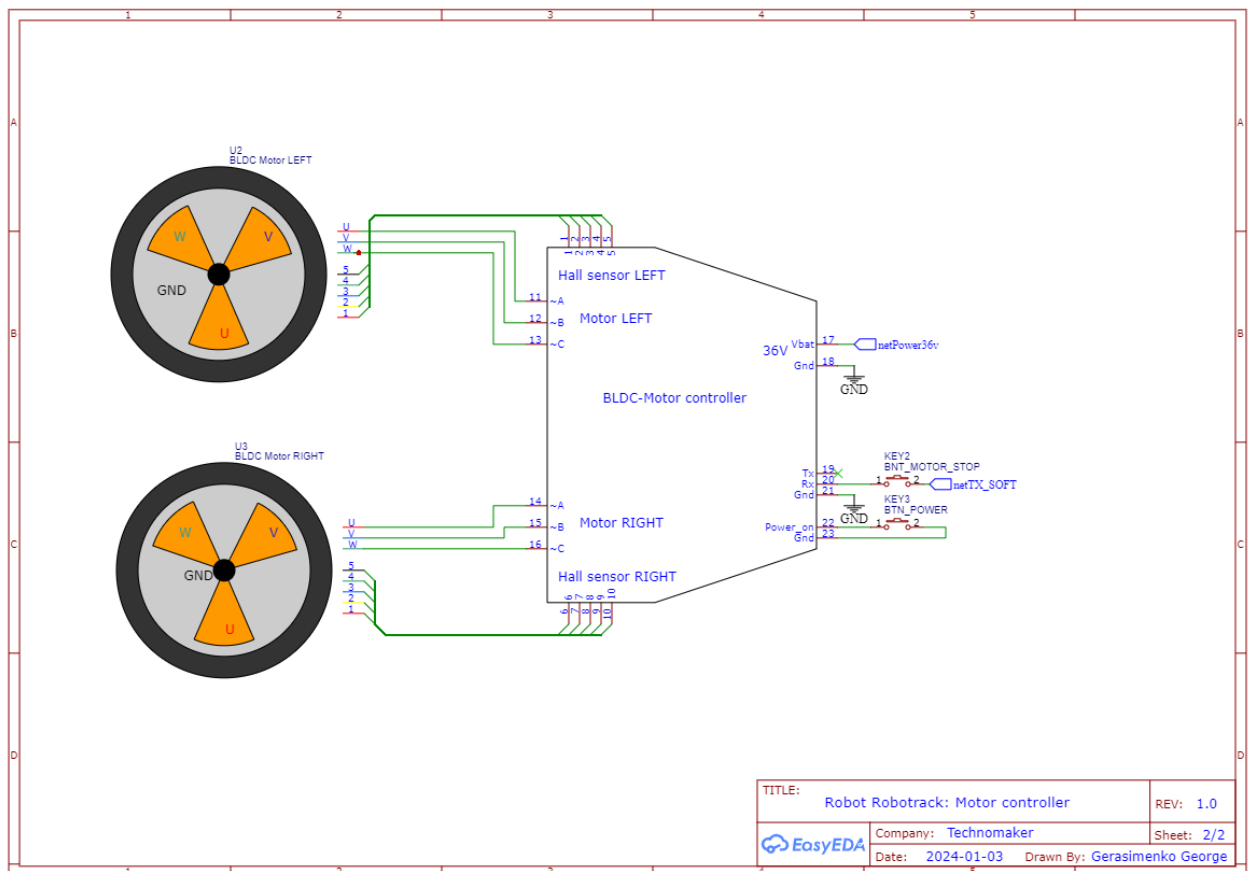
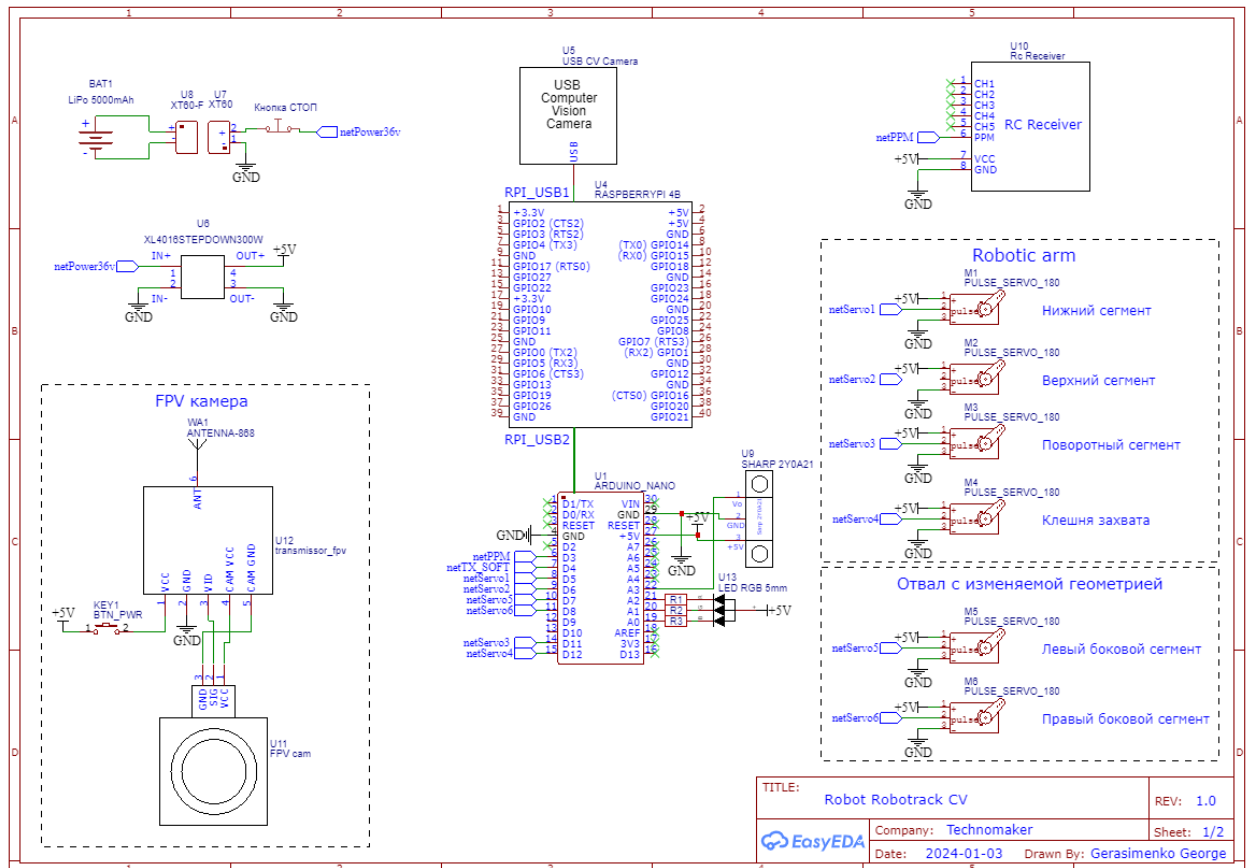
Список литературы:

1. Никонов Вячеслав. КОМПАС-3D: создание моделей и 3D-печать./В.В.Никонов — СПб.: Питер, 2020. — 208 с.: ил. — (Серия «Учебное пособие»).
2. Стрелков А. Г. Конструкция быстроходных гусеничных машин./А.Г. Стрелков. - М.: МГТУ «МАМИ», 2005. - 616 с.
3. Богданов В.К. Опиливание металла: методические указания к лабораторной работе/ В.К. Богданов. – Оренбург: – ИПК ГОУ ОГУ, 2007. – 30 с.
4. Робототехнические системы: Учебное пособие / В. Г. Хомченко — Омск: ОмГТУ – 195 стр., 2016.
5. Цикл статей OpenCV в Python /Sthitaprajna Mishra, MaxRokatansky / [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/otus/articles/558426/>
6. Петин В. А. Проекты с использованием контроллера Arduino./ В.А.Петин — СПб.: БХВ-Петербург, 2014. — 400 с.
7. Герасименко Георгий Андреевич. ТехноМейкер. Робот-помощник Robotrack / Г.А. Герасименко / [Электронный ресурс]. – Режим доступа: <https://youtu.be/sAg2JjmrN94?si=w6CvWURCGeV-Iu4Q>

Приложение 1. Технические характеристики

Модификация	Robotrack CV
Эксплуатационная масса, кг	20
Двигатель	BLDC-250hall-gyro
Мощность двигателя, Вт	250
Номинальная частота вращения, об/мин	2000
Привод подъема отвала	Винтовой
Высота подъёма отвала, мм	50
Гусеничная тележка	
Привод хода	Мотор-колесо
Звездочка, механизм натяжения, опорные катки	PETG пластик
Ширина гусеничного полотна, мм	150
Удельное давление на грунт, кг/см ²	0,02
Роборука	
Степеней свободы	3
Крутящий момент сервопривода, кг*см	25
Полезная нагрузка, г	500
Длина роботизированной руки, мм	355
Захват	Клешневой, редукторный
Рабочие характеристики	
Максимальная скорость, км/ч	20
Емкость АКБ, мАч	5000
Время работы на одном заряде, мин	60
Габаритные размеры (ДхШхВ) без отвала, мм	510x510x310
Материал изделия	Пластик PETG, металл, фанера

Приложение 2. Схема электрическая принципиальная



Приложение 3. Программный код для Arduino

```

1.  // *****
2.  //  Arduino Nano код для Robotrack CV v.1.0
3.  //
4.  //  За основу был взят пример работы с контроллером моторов
5.  //  https://github.com/EmanuelFeru/hoverboard-firmware-hack-FOC
6.  //  *****
7.  //  INFO:
8.  //  Скетч для промышленного робота-помощника Robotrack
9.  //  Реализовано:
10. //  1. Работа с радиопультом по PPM
11. //  2. Автономная работа с помощью машинного зрения (Малинка подключается по USB)
12. //  3. Реализован алгоритм плавного управления сервами и фильтрация помех
13. //  4. Реализован механизм выбора режимов на лету с помощью кнопок на пульте
14. //  5. Реализована программная блокировка робота
15. //
16. //  Copyright (C) 2023 Gerasimenko George <ggeorge3d@yandex.ru>
17. //  *****
18.
19.
20. #include "ArduinoJson.h"          // подключаем либу для json-пакетов
21. StaticJsonDocument<200> jsondoc;  // json буфер для пакетов от raspberry pi
22. int16_t throttleRPI = 0;          // газулька с малинки
23. int16_t steeringRPI = 0;          // поворот с малинки
24. #define AMOUNT 4                  // кол-во сервоприводов
25. #include <ServoSmooth.h>           // подключаем либу для серв
26. ServoSmooth servos[AMOUNT];       // генерим массив серв
27. uint32_t myTimer;                 // таймер
28. #include <PPMReader.h>             // подключаем либу для RC приемника
29. byte interruptPin = 3;            // RC PPM на D3
30. byte channelAmount = 8;           // кол-во каналов в пакете PPM
31. PPMReader ppm(interruptPin, channelAmount); // Инициализируем
32.
33. // ***** DEFINES *****
34. #define HOVER_SERIAL_BAUD 115200  // Скорость программного uart для связи с контроллером моторов
35. #define SERIAL_BAUD 9600          // Скорость встроенного uart
36. #define START_FRAME 0xABCD       // Стартовый фрейм
37. #define TIME_SEND 100             // [мсек] Временной интервал отправки
38. #include <SoftwareSerial.h>        // подключаем либу софтового uart
39. SoftwareSerial HoverSerial(5,4);  // RX на D5, TX на D4
40. // Global variables
41. typedef struct                   // создание структуры для отправки пакета в контроллер моторов
42. {
43.     uint16_t start;              // сначала стартовый фрейм
44.     int16_t steer;               // потом поворот
45.     int16_t speed;               // затем скорость
46.     uint16_t checksum;           // заканчивается пакет контрольной суммой
47. } SerialCommand;
48. SerialCommand Command;
49.
50. // ***** SETUP *****
51. void setup()
52. {
53.     Serial.begin(SERIAL_BAUD);
54.     Serial.println("Robotrack CV v1.0");
55.     HoverSerial.begin(HOVER_SERIAL_BAUD);
56.     //настройка роборуки
57.     servos[0].attach(11,90); // поворот
58.     servos[1].attach(12,90); // захват
59.     servos[2].attach(6,150); // предплечье
60.     servos[3].attach(5,150); // плечо
61.     // настраиваем макс. скорости и ускорения
62.     servos[0].setSpeed(1000);
63.     servos[0].setAccel(0);
64.     servos[1].setSpeed(1000);
65.     servos[1].setAccel(0);
66.     servos[1].setAutoDetach(false); // не отключаем серву, держит захват
67.     servos[2].setSpeed(1500);
68.     servos[2].setAccel(1000);
69.     servos[3].setSpeed(1500);
70.     servos[3].setAccel(1000);
71.     void smoothStart(); // инициализируем сервы
72. }
73.
74. // ***** ОТПРАВКА ПАКЕТА НА КОНТРОЛЛЕР МОТОРОВ *****
75. void Send(int16_t uSteer, int16_t uSpeed)
76. {
77.     // Подготавливаем пакет на отправку
78.     Command.start = (uint16_t)START_FRAME;
79.     Command.steer = (int16_t)uSteer;
80.     Command.speed = (int16_t)uSpeed;
81.     Command.checksum = (uint16_t)(Command.start ^ Command.steer ^ Command.speed);
82.     // Отправляем пакет на контроллер моторов
83.     HoverSerial.write((uint8_t *) &Command, sizeof(Command));
84. }

```

```

84.
85. // ##### LOOP #####
86. unsigned long iTimeSend = 0; // таймер
87. short rot_prev=101; // старое значение поворотной оси
88. short zah_prev=101; // старое значение захвата
89. short arm2_prev=90; // старое значение предплечья
90. short arm1_prev=90; // старое значение плеча
91. short rot_cur; // текущее значение поворотной оси
92. short zah_cur; // текущее значение захвата
93. short arm2_cur; // текущее значение предплечья
94. short arm1_cur; // текущее значение плеча
95. boolean state = servos[0].tick();
96.
97. void loop(void)
98. {
99.     int steeringAngle = map(ppm.latestValidChannelValue(2, 0),1000,2000,-500, 500); // принимаем и мапим поворот
100.    int throttlePercent = map(ppm.latestValidChannelValue(1, 0),1000,2000,-200, 200); // принимаем и мапим скорость
101.    int selectMode = map(ppm.latestValidChannelValue(7, 0),1000,2000,1,6); // принимаем и мапим режим работы
102.
103.    if (selectMode==3) //Режим 3 - автономный режим с использованием компьютерного зрения
104.    {
105.        DeserializationError err = deserializeJson(jsondoc, Serial); // получаем сообщение от малинки через uart
106.        if (err == DeserializationError::Ok) // если сообщение принято
107.        {
108.            throttleRPI = (float)jsondoc["speedA"]; // сохраняем скорость (приходит диапазон [-100; 100])
109.            steeringRPI = (float)jsondoc["speedB"]; // сохраняем поворот с малинки
110.            Serial.print("You sent me json: throttle:"); // отладочный вывод принятых данных с малинки в порт
111.            Serial.print(throttleRPI); // выводим требуемую скорость по машинному зрению
112.            Serial.print(", steering:");
113.            Serial.print(steeringRPI); // выводим требуемый поворот по машинному зрению
114.            Serial.print(", parameter:");
115.            Serial.print(selectMode); // выводим режим работы
116.            Serial.println();
117.            jsondoc["Mode"] = selectMode; // засылаем режим работы в малинку
118.        }
119.        else
120.        {
121.            while (Serial.available() > 0) Serial.read(); // если сообщение не принято, то чистим буфер
122.        }
123.    }
124.
125.    // Формируем отправку данных на контроллер моторов в зависимости от режима работы
126.    unsigned long timeNow = millis();
127.    if (iTimeSend > timeNow) return;
128.    iTimeSend = timeNow + TIME_SEND;
129.    switch (selectMode) { // Выбор режима управления роботом
130.    case 1: // Режим 1 - работы с пульта управления
131.        servos[0].start(); // Активируем серву
132.        servos[1].start(); // Активируем серву
133.        servos[2].start(); // Активируем серву
134.        servos[3].start(); // Активируем серву
135.        Send(steeringAngle, throttlePercent); // Отправляем данные с пульта в контроллер моторов
136.        break;
137.    case 2: // Режим 2 - машинное зрение - линия/целеуказание
138.    case 3: // Режим 3 - машинное зрение - линия/целеуказание
139.        servos[0].start(); // Активируем серву
140.        servos[1].start(); // Активируем серву
141.        servos[2].start(); // Активируем серву
142.        servos[3].start(); // Активируем серву
143.        Send(steeringRPI, throttleRPI); // Отправляем данные с малинки в контроллер моторов
144.        break;
145.    case 4: // Режим 4 - программная блокировка робота
146.    case 5: // Режим 5 - программная блокировка робота
147.    case 6: // Режим 6 - программная блокировка робота
148.        servos[0].stop(); // Деактивируем серву
149.        servos[1].stop(); // Деактивируем серву
150.        servos[2].stop(); // Деактивируем серву
151.        servos[3].stop(); // Деактивируем серву
152.        Send(0, 0); // Блочим двигатели
153.        break;
154.    default: // выполнить, если значение не совпадает ни с одним из case
155.        servos[0].stop(); // Деактивируем серву
156.        servos[1].stop(); // Деактивируем серву
157.        servos[2].stop(); // Деактивируем серву
158.        servos[3].stop(); // Деактивируем серву
159.        Send(0, 0); // Блочим двигатели
160.        break;
161.    }
162.    // считываем данные положения роборуки с пульта и мапим в диапазон работы сервоприводов
163.    if (millis() - myTimer >= 40) {
164.        myTimer = millis();
165.        rot_cur = map(ppm.latestValidChannelValue(6, 0),980,2030,170,10); // поворотная ось
166.        zah_cur = map(ppm.latestValidChannelValue(5, 0),980,2030,10,170); // захват
167.        arm2_cur = map(ppm.latestValidChannelValue(4, 0),980,2030,160,10); // предплечье
168.        arm1_cur = map(ppm.latestValidChannelValue(3, 0),980,2030,160,10); // плечо
169.

```

```

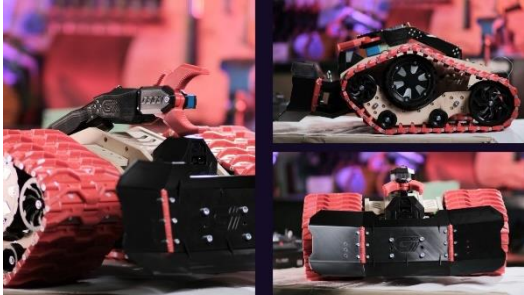
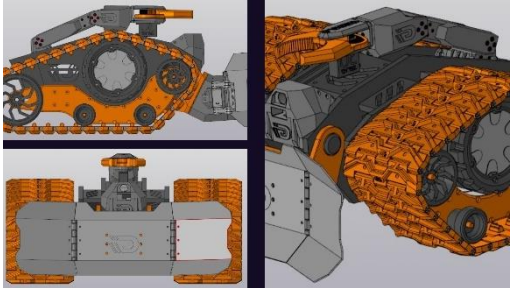
170. // управление поворотной осью роборуки
171.   if (abs(rot_prev-rot_cur)>2) { // зря не дергаем сервой
172.       servos[0].setTargetDeg(rot_prev+0.1*(rot_cur-rot_prev)); // если отклонение значительное, то плавно идем к значению
173.       rot_prev = rot_prev+0.1*(rot_cur-rot_prev); // сохраняем текущее положение
174.   }
175.   else {
176.       servos[0].setTargetDeg(rot_prev); // если меньше 2 градусов, возможно это помеха, ничего не делаем
177.   }
178. // управление захватом роборуки
179.   if (abs(zah_prev-zah_cur)>2) { // зря не дергаем сервой
180.       servos[1].setTargetDeg(zah_prev+0.1*(zah_cur-zah_prev)); // если отклонение значительное, то плавно идем к значению
181.       zah_prev = zah_prev+0.1*(zah_cur-zah_prev); // сохраняем текущее положение
182.   }
183.   else {
184.       servos[1].setTargetDeg(zah_prev); // если меньше 2 градусов, возможно это помеха, ничего не делаем
185.   }
186.
187. // управление предплечьем роборуки
188.   if (abs(arm2_prev-arm2_cur)>2) { // зря не дергаем сервой
189.       servos[2].setTargetDeg(arm2_prev+0.1*(arm2_cur-arm2_prev)); // если отклонение значительное, то плавно идем к значению
190.       arm2_prev = arm2_prev+0.1*(arm2_cur-arm2_prev); // сохраняем текущее положение
191.   }
192.   else {
193.       servos[2].setTargetDeg(arm2_prev); // если меньше 2 градусов, возможно это помеха, ничего не делаем
194.   }
195.
196. // управление плечом роборуки
197.   if (abs(arm1_prev-arm1_cur)>2) { //зря не дергаем сервой
198.       servos[3].setTargetDeg(arm1_prev+0.1*(arm1_cur-arm1_prev)); // если отклонение значительное, то плавно идем к значению
199.       arm1_prev = arm1_prev+0.1*(arm1_cur-arm1_prev); // сохраняем текущее положение
200.   }
201.   else {
202.       servos[3].setTargetDeg(arm1_prev); // если меньше 2 градусов, возможно это помеха, ничего не делаем
203.   }
204.
205. servos[0].tick(); // здесь происходит движение серво по встроенному таймеру
206. servos[1].tick(); // здесь происходит движение серво по встроенному таймеру
207. servos[2].tick(); // здесь происходит движение серво по встроенному таймеру
208. servos[3].tick(); // здесь происходит движение серво по встроенному таймеру
209. }
210. }
211.
212. // ##### END #####

```

Приложение 4. Quad-Chart

Industrial cleaning robot ROBOTRACK Robotic cleaning of the urban environment

George Gerasimenko, Lyceum "Business and Information Technology", Омск city, Russia

<p><i>Q1: Engineering Goal</i></p> <p>Dream: To make an unmanned industrial cleaner on tracks for cleaning the city in winter and summer in automatic mode so that the city is always clean.</p> <p>Relevance of the work: In winter and summer, the issue of timely cleaning of sidewalks and territories from snow, dust and garbage is very acute in public areas of the city. The Robotrack will allow replacing heavy human labor with machine work.</p>	<p><i>Q3: Data Analysis & Results</i></p> <p>Now the prototype is printed on a 3D printer and is a fully working model, controlled by a remote control and autonomously using machine vision</p> <p>The circuit was assembled on Arduino and Raspberry PI, motor controllers, servos, lidar, transmitter and WEB camera and FPV camera.</p> 
<p><i>Q2: Project Design</i></p> <p>The industrial cleaning robot Robotrack required the development of tracked platform, tracks, and calculate the power plant. I have designed a mechanism for quick change of brush heads, dumps, augers.</p> <p>I have also developed a 3-axis robotic arm for moving garbage.</p> <p>I designed a robot in KOMPAS3D software.</p> 	<p><i>Q4: Interpretation & Conclusion</i></p> <p>Summing up the results of the creation of the Robotrack robot, I see a huge potential for its use to solve the household needs of any person.</p> <p>My robot will be able to help humanity solve everyday issues of cleaning the territory automatically, without human intervention, without polluting the environment with harmful emissions.</p> <p>The clean Smart City is beautiful and safe!</p>

Приложение 5. Диплом за работа-помощника Robotrack



Приложение 6. Диплом за работа-помощника Robotrack



Награждается участник команды

ФИО Герасименко Георгий Андреевич

Команда Робот-помощник ROBOTRACK

БУ ДО Омская областная станция юных техников
(название организации)

За I место в категории Свободная творческая категория: средняя

Международного фестиваля робототехники РобоФинист 2023
г. Санкт-Петербург

Заместитель председателя
комитета по образованию
Борщевский А.А.

Президент
Благотворительного Фонда «Финист»
Аминджанов Т.А.

Директор ГБОУ
«Президентский ФМЛ №239»
Пратусевич М.Я.

Руководитель
организационного комитета
Боточева О.В.

Главный судья соревнований
Филиппов С.А.

29 октября — 1 ноября 2023

000-104-011

РОБОФИНИСТ

StarLine



Приложение 7. Диплом за робота-помощника Robotrack



Приложение 8. Благодарственное письмо за участие с Robotrack



**III КОНГРЕСС
МОЛОДЫХ
УЧЕНЫХ**

РОСКОНГРЕСС
Время действовать



МИНОБРНАУКИ
РОССИИ



Совет при Президенте Российской Федерации
по науке и образованию
Координационный совет по делам молодежи
в научной и образовательной сферах

Оператор проведения
Десятилетия науки и технологий

НАЦИОНАЛЬНЫЕ
ПРИОРИТЕТЫ

Г.А.Герасименко
Ученику

БОУ города Омска «Лицей «Бизнес и информационные технологии»

Уважаемый Георгий Андреевич!

Разрешите поблагодарить Вас за поддержку и деятельное участие в работе **III Конгресса молодых ученых** (федеральная территория «Сириус», 28–30 ноября 2023 г.).

Ежегодный Конгресс молодых ученых проходит как итоговое мероприятие года, проводимое в рамках объявленного в 2022 году Президентом Российской Федерации Десятилетия науки и технологий в России. Организаторами Конгресса выступают Министерство науки и высшего образования Российской Федерации, Координационный совет по делам молодежи в научной и образовательной сферах Совета при Президенте Российской Федерации по науке и образованию и Фонд Росконгресс.

Центральным событием Конгресса традиционно стала беседа Президента Российской Федерации с молодыми учеными, в ходе которой обсуждались вопросы реализации прикладных проектов по направлениям стратегических инициатив, закрепленных в Стратегии научно-технологического развития Российской Федерации.

В этом году Конгресс собрал более 5000 участников из 85 регионов Российской Федерации и 36 зарубежных стран, представителей ведущих научных школ и образовательных организаций, органов власти и бизнес-сообщества, мэтров отечественной науки, а главное – молодых ученых, победителей конкурсов грантов, студентов и школьников, добившихся значимых результатов в своих исследованиях.

Деловая программа, в рамках которой состоялось свыше 150 различных мероприятий, была посвящена как вопросам создания и внедрения в производство перспективных технологий, формирования благоприятных условий для проведения исследований и разработок, соответствующих лучшим российским и мировым практикам в интересах укрепления технологического суверенитета России, так и вопросам международной повестки – продвижения конкурентоспособных высокотехнологичных продуктов на внешние рынки, расширения и укрепления научно-технологического и научно-образовательного сотрудничества с иностранными государствами.

За эти годы Конгресс молодых ученых превратился в авторитетную международную площадку для открытого прикладного диалога и обмена передовыми научными идеями. Он позволяет участникам приобщиться к решению важнейших задач развития общества и страны, содействует распространению информации о достижениях и перспективах развития российской и мировой науки.

Позвольте пригласить Вас на следующий Конгресс молодых ученых, который состоится в ноябре 2024 года. Рассчитываем на Ваше деятельное участие в содержательной и организационной работе по его подготовке.

С уважением,

Советник
Президента Российской Федерации,
руководитель межведомственной
рабочей группы по подготовке
и проведению Конгресса молодых
ученых и мероприятий-спутников

А.Кобяков