

Областное государственное бюджетное общеобразовательное учреждение  
"Томский физико - технический лицей"

**АВТОНОМНАЯ ГУСЕНИЧНАЯ РОБОТИЗИРОВАННАЯ ПЛАТФОРМА ДЛЯ  
ПРОВЕДЕНИЯ ПОИСКОВО-СПАСАТЕЛЬНЫХ ОПЕРАЦИЙ**

Работу выполнили:

ученица МБОУ лицей при ТПУ г. Томска  
ученик ОГБОУ «ТФТЛ»

М.С. Цыганкова  
Г.А. Пильщиков

Наставник проекта:

Заместитель директора по  
информационным технологиям ОГБОУ  
«ТФТЛ»

С.В. Косаченко

Томск  
2025

## **СОДЕРЖАНИЕ**

<b>Перечень сокращений и обозначений</b>	<b>3</b>
<b>Введение</b>	<b>5</b>
<b>Основная часть работы</b>	<b>7</b>
Анализ и сравнение аналогов	7
Планирование работ	9
Формулировка технического задания	11
Разработка механической части робота	13
Разработка электрической части робота	17
Разработка программного обеспечения	19
Тестирование	24
<b>Заключение</b>	<b>28</b>
<b>Список использованных источников</b>	<b>30</b>
<b>Приложения</b>	<b>32</b>
<b>Ссылки и контакты</b>	<b>39</b>

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей работе применяют следующие сокращения и обозначения:

ЧС	- чрезвычайная ситуация, обстановка на определенной территории, сложившаяся в результате форс-мажорных обстоятельств.
МЧС России (МЧС)	- министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий.
UGV аппараты	- Unmanned Ground Vehicle (беспилотные наземные аппараты), наземное транспортное средство, функционирующее без экипажа на борту.
UAV аппараты	- Unmanned Aerial Vehicle (беспилотные летательные аппараты. БПЛА), летательный аппарат, который управляется без экипажа на борту.
USV аппараты	- Unmanned Surface Vehicle (беспилотные надводные аппараты), аппарат, способный двигаться по поверхности воды без наличия экипажа на своём борту.
UUV аппараты	- Unmanned Undersea Vehicle (беспилотные подводные аппараты), аппарат, способный двигаться под водой без наличия экипажа на своём борту.
ТНПА	- Телеуправляемый необитаемый подводный аппарат, подводный аппарат, часто называемый роботом, который управляется оператором или группой операторов.
ПСР	- Поисково-спасательные роботы, категория сервисных роботов, предназначенных для оказания помощи при поиске и спасении людей в условиях чрезвычайных ситуаций.

ЧПУ станок	- Станки числовым программным управлением, автоматизированные станки-роботы, которые могут производить операции по заданной программе без непосредственного участия.
ПО	- Программное обеспечение.
САПР	- Система автоматизированного проектирования, комплекс программных и технических средств, эксплуатируемых для простой, недорогой и быстрой разработки проектов, моделей и чертежей.
RPI	- Raspberry Pi, маленький, доступный и мощный одноплатный компьютер.
ЛДСП	- Ламинированная древесно-стружечная плита

## **ВВЕДЕНИЕ**

### **Актуальность**

На данный момент количество чрезвычайных ситуаций (ЧС), происходящих по всему миру стремительно растёт. Так, по данным министерства чрезвычайных ситуаций (МЧС) России за 2023 год количество произошедших ЧС увеличилась на 26,03% [1]. При возникновении ЧС различного характера сложно избежать пострадавших, так по данным этой же статистике за 2023 год количество пострадавших при ЧС увеличилась на 22,24%, а количество погибших при ЧС увеличилась на 55,78% [1]. Для помощи пострадавшим при ЧС работают целые бригады врачей и спасателей, занимающихся поисково-спасательными операциями, такая работа является очень опасной для человека. В настоящее время службы спасения уже стараются привлекать БПЛА для таких операций, однако наземные роботы-спасатели, в частности – автономные или полуавтономные, в данной сфере все еще мало распространены. В связи с чем становится актуальной разработка и внедрение наземных аппаратов для помощи проведения спасательных операций при ликвидации ЧС. Миссии этих роботов могут включать такие работы как обследование объектов в очагах поражения, очищение путей для дальнейших работ спасателей, ликвидация возгораний, поиск и спасение пострадавших [2].

### **Проблема**

В текущее время в мире все чаще для проведения поисково-спасательных операций используют UAV аппараты (БПЛА), что бывает проблематично в тесных и закрытых пространствах, пострадавших при ЧС. Существующие наземные поисково-спасательные роботы (ПСР) и комплексы для помощи проведения поисково-спасательных операций, в частности автономные, мало распространены, практически недоступны к покупке или используются исключительно в военной промышленности, что замедляет рост интереса к разработке и внедрению ПСР в практику служб спасения. Таким образом данный проект решает проблему доступности таких аппаратов.

### **Круг потенциальных заказчиков / потребителей / пользователей**

Службы спасения, исследовательские центры для мониторинга объектов природы, волонтерские организации.

### **Показатели назначения**

Данная разработка предназначена для осмотра закрытых объектов (помещений, зданий), пострадавших при ЧС локального характера, анализа обстановки (в том числе

составлении карт помещений), а также поиска пострадавших при ЧС и предоставления им комплектов первой помощи, до прихода спасателей.

### **Предполагаемый результат проекта**

При разработке проекта планировалось создать уменьшенный прототип аппарата, для полностью автономного проведения поисково-спасательной операции, анализируя и приспосабливаясь к окружающей среде, следовательно, в соответствии ГОСТ Р60, робота-спасателя для выполнения поставленных задач.

### **Команда для реализации проекта**

- 1) Пильщиков Григорий - капитан команды. Разработал алгоритмы компьютерного зрения, занимался пайкой электрических компонентов, занимался сборкой робота.
- 2) Цыганкова Мария - член команды. Разработала алгоритмы движения, навигации и картографирования, составляла чертежи деталей конструкции робота, чертежи плат занималась подключением электрических компонентов, сборкой робота.
- 3) Косаченко Сергей Викторович - наставник команды.

### **Цель работы**

Разработать уменьшенный прототип автономного робота для помощи проведения поисково-спасательных операций.

### **Задачи**

1. Изучить уже существующие решения, аналоги с похожим функционалом
2. Разработать работоспособный макет робота
3. Протестировать робота, проанализировать полученные результаты
4. Доработать минусы, появившиеся в результате тестирования

## ОСНОВНАЯ ЧАСТЬ РАБОТЫ

### Анализ существующих решений

Поисково-спасательные роботы предназначены для помощи ликвидации стихийных бедствий. Конструкции таких роботов напрямую зависят от условий среды, в которой им предстоит выполнять миссии. Рассмотрим различные виды таких аппаратов.

Первым видом таких роботов относятся беспилотные наземные аппараты (Unmanned Ground Vehicle / UGV аппараты) [3]. Такие аппараты могут обследовать опасные или труднопроходимые области поражения, расчищать обвалы для дальнейшей работы спасателей. Кроме того, UGV аппараты могут быстрее и точнее выполнять миссии, требующие точных движений.

Множество UGV аппаратов было разработано, а в последствии и использовано при ликвидации аварии на Чернобыльской атомной электрической станции (ЧАЭС). Например, были разработаны радиоуправляемые роботы разведчики РР-1, РР-2, РР-3 (колесные модели) а также РР-Г1 и РР-Г2 (гусеничные аппараты, рисунок 1). Эти роботы были оснащены подвижными камерами, гамма-локаторами и дозиметрической аппаратурой [4]. Задачи таких роботов заключались в визуальном осмотре пораженной территории и в определении радиационной обстановки как внутри зданий, так и снаружи [5]. Если же говорить о крупногабаритных моделях, то нужно сказать о мобильном роботе Мобот-Ч-ХВ (рисунок 2). Робот представлял собой гусеничную платформу для очистки пораженной территории (в частности крыши ЧАЭС)

Современные модели UGV роботов спасателей более многофункциональны и автономны, а, следовательно, не нуждаются в присутствии человека. Например, робот SCRATCHbot (рисунок 3), от команды разработчиков из университетов Шеффилда и Бристоля, способен автономно проводить поисково-спасательные операции. В конструкции робота имеются особые усы-сенсоры, которые гораздо дешевле, надежнее и эффективнее, чем дорогие и сложные системы с камерами [5].



Рисунок 1 — робот РР-Г1

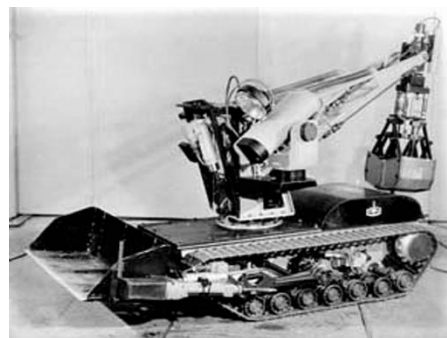


Рисунок 2 - мобильный робот Мобот-Ч-ХВ

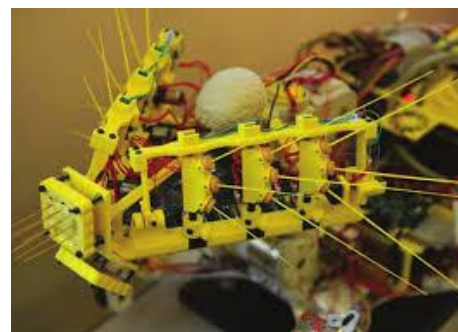


Рисунок 3 - робот SCRATCHbot

Второй вид роботов спасателей — беспилотные летательные аппараты (Unmanned Aerial Vehicle / UAV аппараты) [3]. Такие аппараты значительно выигрывают по скорости обследования больших открытых пространств. Кроме того, скорость доставки спасательных комплектов пострадавшим у UAV аппаратов превосходит большую часть других видов роботов спасателей.

Так, в штате роботов МЧС России уже имеется аэростатический комплекс видео наблюдения, управления и связи ОКО-4 (рисунок 4). Этот аппарат способен осматривать и контролировать территорию площадью до 10 км<sup>2</sup> [6]. Основной задачей этого аппарата является обследование и контроль пораженные территории и координировать команду наземных спасательных роботов. Таким образом, при совместной работе UGV и UAV аппаратов можно сильно повысить эффективность поисково-спасательных операций.



Рисунок 4 - аэростатический комплекс видео наблюдения ОКО-4

Еще одним видом роботов спасателей являются беспилотные водные аппараты. Такие аппараты делятся на надводные (Unmanned Surface Vehicle / USV аппараты) и подводные (Unmanned Undersea Vehicle / UUV аппараты) [3]. USV и UUV аппараты расширяют возможности спасателей, так как работают абсолютно в иной среде. Благодаря таким аппаратам становится возможным проведение поисково-спасательных операций не только на суше, но и в воде.



Рисунок 5 - ТНПА Гном

Примером UUV аппарата может быть робот Гном (рисунок 5), используемый МЧС России для обследования стратегических подводных объектов. Такой и подобные ему роботы относятся к классу телеуправляемых необитаемых подводных аппаратов (ТНПА). Они способны погружаться и обследовать объекты, недоступные водолазам, поднимать грузы и людей со дна, чинить трубы и другую подводную аппаратуру [7].

Из используемых на практике USV аппаратов, хочется выделить многоцелевую безэкипажную гидравлическую платформу (БЭК) Сивуч (рисунок 6), для проведения поисково-спасательных операций в море непосредственно на воде. Этот аппарат позволяет



Рисунок 6 - БЭК Сивуч



без человека обследовать водные пространства на расстоянии 15 км от берега, что позволяет спасателям не выходить в воду и не рисковать своей жизнью [8].

### **Сравнение существующих аналогов**

В данной работе планировалось разработать наземного автономного робота спасателя. Поэтому для анализа аналогов были взяты похожие по функционалу и конструкции аппараты. Для сравнения брались некоторые наземные роботы, использовавшиеся при ликвидации ЧС в Чернобыле (описание см. выше), а также некоторые другие коммерческие современные UGV аппараты.

Так одним из аналогов, разбираемых нами стала горноспасательная машина Gemini scout, разработанная компанией Sandia Laboratories при финансовой поддержке Национального института безопасности и гигиены труда США [10]. Робот представляет собой гусеничную платформу для поисково-спасательных операций в угольных шахтах, с дистанционным управлением [9].

Робот Numbat был создан при поддержке Организации Содружества по научным и промышленным исследованиям Австралии. Этот ПСР представляет из себя колесного робота для проведения поисковых операций по сбору информации о состоянии подземных шахт, необходимых спасателям, в опасных для человека условиях.

Таким образом, проанализировав некоторые аналоги и составив сравнительную таблицу (см. Приложение 1) можно сделать вывод о том, что наша разработка должна быть достаточно хорошо проходимой, но в тоже время компактной для доступа в труднодоступные места, а также позволять спасателям дистанционно получать информацию о состоянии исследуемого объекта.

### **Планирование работ**

1) Формулировка технического задания, расчет конструкции.

Необходимые ресурсы/компоненты: ноутбук, программы для черчения.

Промежуточный результат: разработанная и рассчитанная конструкция робота, список необходимых для сборки компонентов.

2) Сборка и тестирование шасси.

Необходимые ресурсы/компоненты: металлические несущие балки, двигатели с энкодерами, шестерни/колеса, гусеничная лента, набор инструментов.

Промежуточный результат: готовое рабочее шасси с возможности регулировать натяжение гусениц без разбора механизма.

3) Сборка каркаса робота, установка контроллера.

Необходимые ресурсы/компоненты: контроллер, ЧПУ станок, фанера, гайки / шайбы / болтики, резьбовые шпильки, набор инструментов.

Промежуточный результат: готовый корпус робота.

4) Пайка и подключение электрических компонентов.

Необходимые ресурсы/компоненты: паяльник, канифоль, олово, флюс, третья рука, пинцет, соединительные провода, подобранные и приобретенные электрические компоненты.

Промежуточный результат: готовая ходовая часть робота.

5) Разработка ПО, тестирование начального прототипа.

Необходимые ресурсы/компоненты: ноутбук, кабель для программирования, место для испытаний.

Промежуточный результат: начальный прототип робота с возможностью передвижения в замкнутых пространствах.

6) Установка камер, разработка системы компьютерного зрения.

Необходимые ресурсы/компоненты: ноутбук, микрокомпьютер, камеры, система подсветки камеры.

Промежуточный результат: готовый прототип робота с системой обнаружения жертв.

7) Разработка систем навигации и картографирования.

Необходимые ресурсы/компоненты: ноутбук, кабель для программирования, симулятор для быстрого тестирования алгоритмов.

Промежуточный результат: экспериментальный прототип ПСР, готовый к тестированию.

8) Тестирование и анализ результатов.

Необходимые ресурсы/компоненты: ноутбук, кабель для программирования, полигон для испытаний, монитор для снятия телеметрии, монитор.

Промежуточный результат: выявленные плюсы/минусы текущей версии ПСР

9) Доработка и усовершенствование.

Необходимые ресурсы/компоненты: в зависимости от того, что будем улучшать.

ИТОГ: Экспериментальный прототип автономного робота для проведения поисково-спасательных операций.

## Формулировка технического задания проекта, разработка концепции робота

Для простоты тестирования, за основу требований к роботу были взяты требования и регламенты различной сложности состязаний роботов-спасателей для студентов и школьников. Мы смоделировали спасательную миссию: робот должен по максимуму обследовать помещение (рисунок 7), построить его карту, отметить и сообщить об опасных или потенциально опасных местах, при обнаружении пострадавших, незамедлительно сообщить спасателям и скинуть комплект для оказания первой помощи, продолжить миссию, после окончания операции вернуться обратно к месту старта. Таким образом, нетрудно сформулировать и сами требования к роботу.

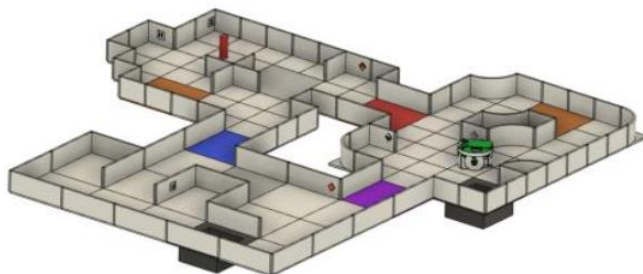


Рисунок 7 – модель поля проведения миссии

- Шасси робота должно быть достаточно прочным, надежным и проходимым, для преодоления препятствий различной трудности.
- Шины или гусеницы должны быть выполнены из нескользящего материала, для увеличения сцепления с поверхностью.
- При использовании гусеничного шасси, необходимо разработать систему удобной регулировки их натяжения.
- Следует предусмотреть энкодеры на каждый из моторов.
- Робот должен обладать датчиками и сенсорами для перемещения по объекту и построения его карты.
- Робот должен контролировать обстановку не только вокруг себя, но и под самим роботом.
- Робот должен обладать камерами и системой компьютерного зрения для обнаружения подозрительных вещей.
- Для системы компьютерного зрения, необходимо выделить микрокомпьютер, задачей которого будет являться анализ изображения и отправка сигнала в случае обнаружения уже на сам контроллер.
- Питание контроллера и микрокомпьютера следует разделить, то есть необходимо предусмотреть раздельное питание с удобством быстрой смены аккумуляторов.
- Для устойчивости конструкции, необходимо стремиться сместить центр тяжести робота как можно ниже.
- Для питания контроллера следует использовать высокотокковые аккумуляторы типа

18650.

- Необходимо предусмотреть стабилизатор питания для некоторых датчиков.
- Дополнительно, необходимо также предусмотреть систему выгрузки комплектов первой помощи.
- Для удобства тестирования, необходимо обеспечить наличие дисплейного модуля для вывода информации.

Таким образом, была разработана следующая концепция робота. В качестве контроллера была выбрана плата meAURIGA, так как она достаточно мощная и представляет из себя по сути Arduino Mega 2560 с уже припаянными необходимыми нам компонентами. В качестве микрокомпьютера была выбрана Raspberry Pi 4, так как это полноценная замена компьютеру на Linux, которая позволяет выполнять все поставленные нами задачи. Конструкция робота предполагает несколько этажей с разным функционалом: на нулевом располагаются моторы, подсветка для камер и датчик цвета, который мы выбрали для контроля состояния поля под роботом, так как при помощи параметра освещенности можно реагировать на ямы до того как их заметил гироскоп; на первом — аккумуляторы для meAURIGA, пауэрбанк Raspberry Pi (RPI); на втором — сама meAURIGA, Raspberry, жесткий диск, преобразователь логических уровней, двунаправленный, 4-х канальный, переходная плата с портов meAURIGA RJ12 на разъемы с шагом 2.45 мм, стабилизатор питания с 5В до 3.3В и инфракрасные дальномеры; на третьем — два экрана для вывода информации, кнопка для быстрого запуска и сами камеры. В пространстве между 2-ым и 3-ем этажами робота располагается бокс для проводов, жесткий диск и meAURIGA. Выгрузка комплектов происходит за счет трубы с самими комплектами первой помощи и гладкой шестерни, которая приводится в движение за счет сервопривода. Шасси плоское, гусеничное, ход заднеприводный. Основа шасси собрана из металлических балок, однако они находятся под углом и скреплены при помощи металлической оси с гайками и шайбами. За счет такой конструкции шасси робот становится более проходимым и пригодным к изменениям натяжения гусениц без разборки всей конструкции.

## **Разработка механической части робота**

### **Разработка шасси**

Начальным этапом разработки робота стал этап разработки шасси для робота. Перед началом разработки было решено, что шасси стоит сделать гусеничным, так как это увеличивает проходимость робота. Первая версия шасси робота была собрана полностью из цельных металлических балок, так как это увеличивает прочность шасси, имела форму треугольника и состояла из 6-ти небольших шестерней (рисунок 8).

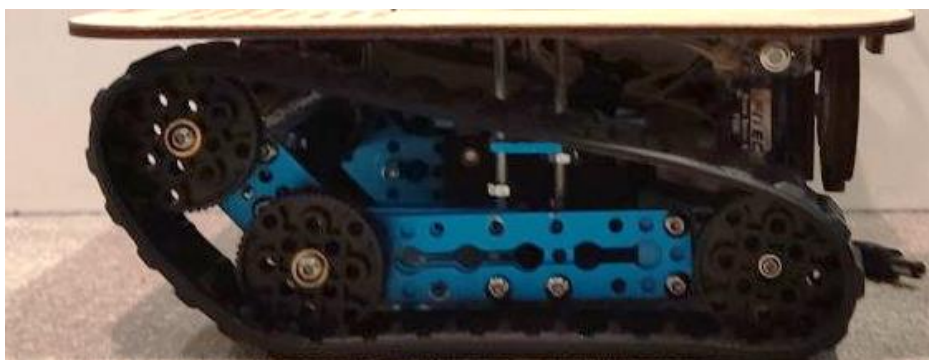


Рисунок 8 – первая версия шасси для робота

Во время тестирования шасси были выявлены две главные проблемы этой версии – отсутствие механизма регулировки натяжения гусениц и переворачивания из-за неудобств треугольной конструкции шасси. Первое, что было решено менять в этой версии это именно треугольную форму, однако металлические балки в конструкции все еще должны были остаться, так как в противном случае шасси может потерять в надежности. В результате чего была изменена конструкция шасси – оно стало полностью плоским (т.е. не имеет наклона, который был в предыдущих версиях), что позволило увеличить устойчивость робота и понизить его центр тяжести (рисунок 9).



Рисунок 9 – вторая версия шасси для робота

В последней версии этого робота в шасси добавили механизм для возможности регулировки натяжения гусениц. Система натяжения для регуляции без разбора шасси была собрана из металлической оси, гаек и шайб. В результате чего, стало возможным изменение натяжения гусениц на месте проведения миссии при помощи гаечного ключа и плоскогубцев. Благодаря этим изменениям у нас появилась возможность ездить как на гусеницах, так и без них, а также увеличилось пространство 1-го этажа, за счет чего нам удалось опустить 2-й и сместить центр тяжести ниже, что также увеличило его проходимость (рисунки 10, 11, 12).

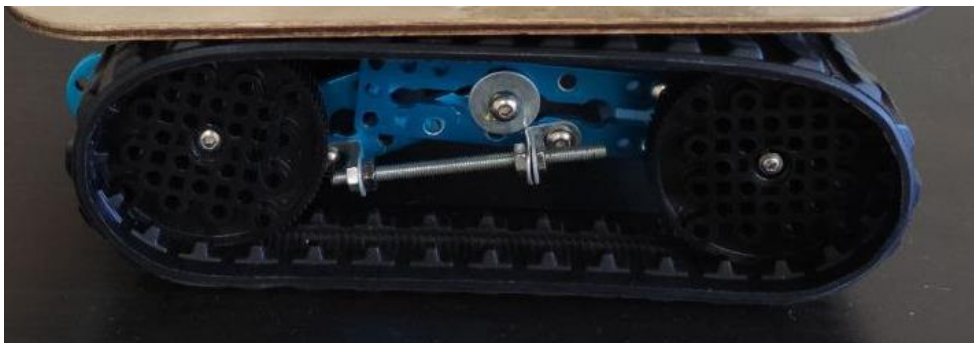


Рисунок 10 – третья версия шасси для робота



Рисунок 11, 12 – третья версия шасси для робота с разных сторон

### **Разработка корпуса**

Следующим этапом разработки стала разработка рамы робота. Форма рамы была выбрана прямоугольной с закругленными углами, которые позволят роботу избежать возможных зацеплений и залипаний угла о угол. Рама была смоделирована в системе автоматизированного проектирования (САПР), при моделировании были учтены отверстия для крепежей последующих деталей, протяжки проводов, шестерни для выгрузки и крепежа основных осей, соединяющих этажи (рисунок 13).

Далее были разработаны крепежи для датчиков расстояния. Перед проектировкой они были протестированы, после чего было рассчитано минимальное, но оптимальное расстояние для их корректной работы, именно на основе этого расстояния и были выбраны положения крепежей на раме. Кроме того, на этом этапе также был разработан и смоделирован каркас для 3-го этажа (рисунок 14).

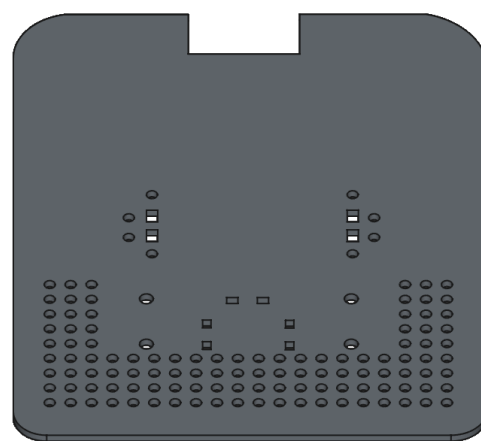


Рисунок 13 – модель готовой рамы робота

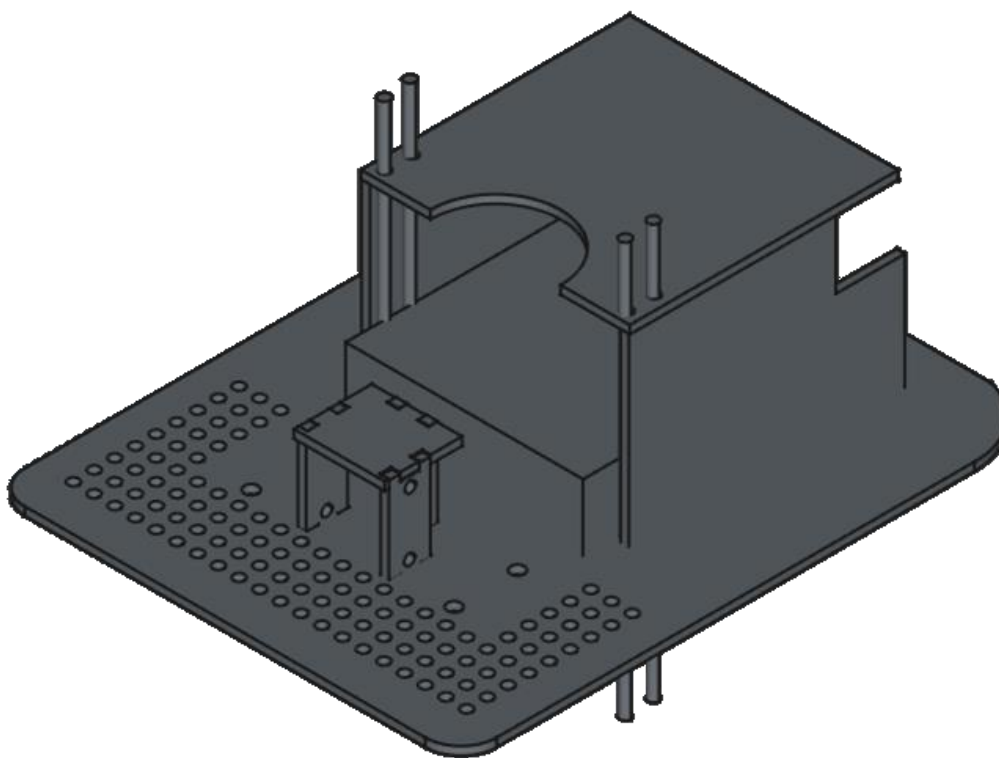


Рисунок 14 – модель основного корпуса робота

После окончания проектировки корпуса робота начался этап его изготовления. В качестве материалов для изготовления прототипа робота была выбрана фанера, так как для нас это был наиболее доступный и очень прочный материал. Все детали были вырезаны на ЧПУ станке и обработаны. В результате, чего был собран готовый корпус робота (рисунок 15).



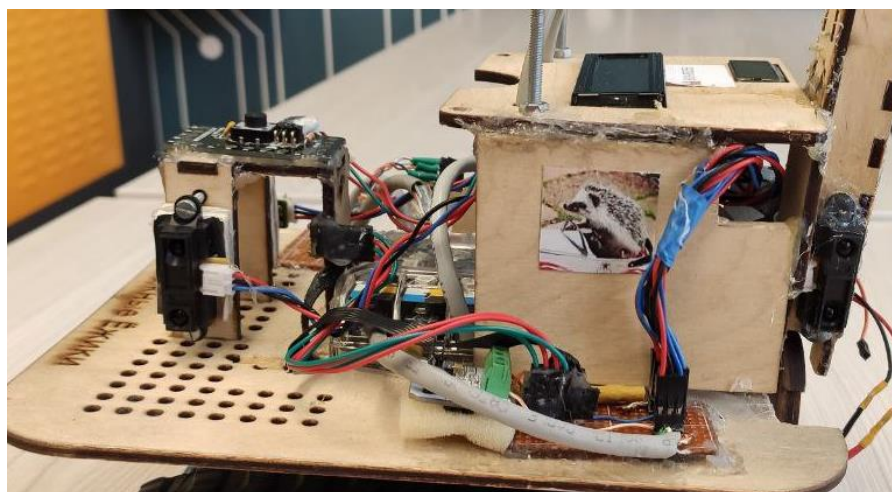


Рисунок 15 – итоговая версия корпуса робота в сборе

Для разработки механизма выгрузки, также использовали САПР. Сам же механизм состоит из трубы-контейнера, содержащего спасательные комплекты первой помощи, одного сервопривода и псевдoshестерни, на которую под действием силы тяжести падает комплект (конструкция выполнена так, что в отсек на шестерне помещается ровно один). Сама выгрузка происходит посредством вращения этого сервопривода (рисунок 16).

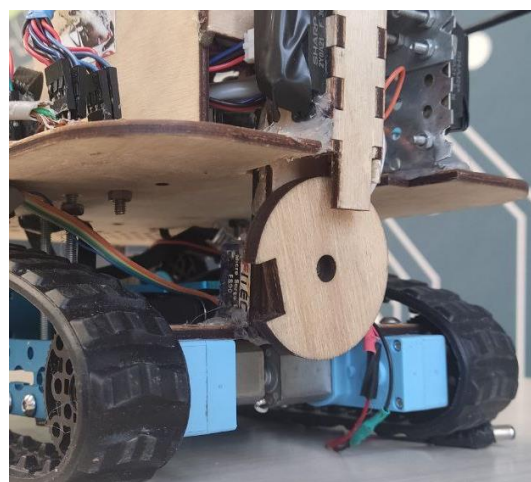


Рисунок 16 – механизм выгрузки в сборе

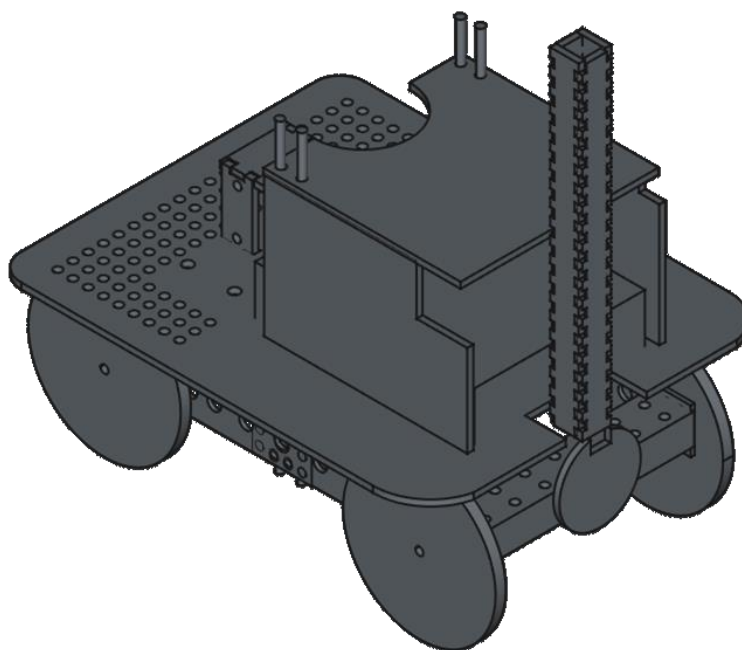


Рисунок 17 – итоговая модель корпуса робота



## Разработка электрической части робота

### Пайка и подключение электрических компонентов

В качестве контроллера робота была выбрана плата meAURIGA, так как она достаточно мощная и представляет из себя по сути Arduino Mega 2560 с уже припаянными необходимыми нам компонентами, такими как гироскоп или модуль Bluetooth. В качестве микрокомпьютера была выбрана Raspberry Pi 4. Также были выбрали инфракрасные дальномеры sharp 2y0a21 f 36, так как они достаточно точные и дешевые, датчик цвета HiTechnic NCO1038, так как он наиболее доступный и простой в подключении, веб камеры, считываемые RPI, которая анализирует данные с камеры и передает на meAURIGA, двигатели постоянного тока DC Motor-37, так как они достаточно легкие и мощные (скорость без нагрузки – 350 об/мин и редукция 39,6), высокоточные, уже со встроенными оптическими энкодорами точностью до 360°, а также поддерживает выбранную нами плату управления. Драйвера для двигателей уже встроены в контроллер. Данные с RPI на meAURIGA мы передаем по самодельному протоколу, используя для приема-передачи данных 4 GPIO с RPI и 4 цифровых пинов от meAURIGA, и из-за разницы напряжений логических выводов, пришлось использовать преобразователь логических уровней, для подключения моторчика выгрузки комплектов мы также спаяли переходную плату с портов meAURIGA RJ12 на разъемы с шагом 2.45 мм. Кроме того, на роботе стоит стабилизатор питания с 5В на 3.3В. Для более удобного старта на роботе вынесена кнопка, запускающая самого робота. Распиновку meAURIGA и RPI 4, мы брали с официальной технической документации.

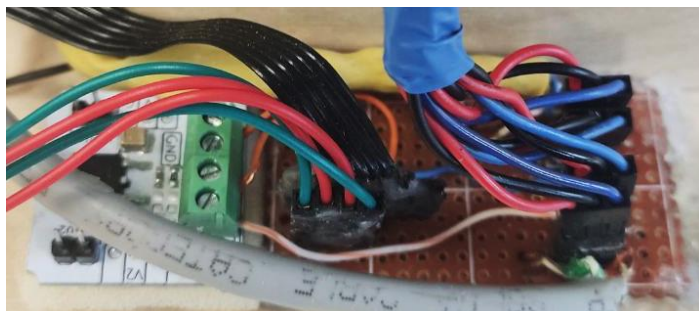


Рисунок 18 – готовая плата для подключения дальномеров

Перед изготовлением и распайкой электрических компонентов для робота необходимо было разработать и рассчитать их структурные и электрические схемы в САПР (см Приложение 2, 3). Так, для питания датчика цвета Hitechnic NCO1038, требуется 3.3В, когда контроллер выдает 5В, следовательно, возникла потребность в стабилизаторе питания с 5В на 3.3В, для подключения дальномеров, нам пришлось спроектировать и распаять переходную плату (рисунок 18) с RJ-12 на стандартные разъемы с шагом 2.45 мм, что облегчило подключение датчиков. А также для "связи" между RPI и

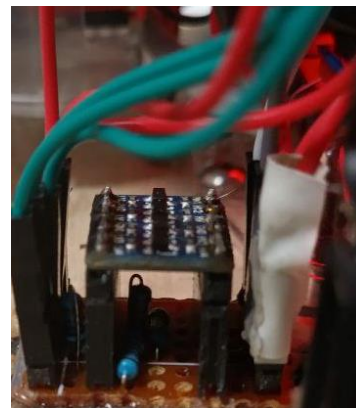


Рисунок 19 – готовая плата преобразователя логических уровней

meAURIGA было необходимо спроектировать и спаять плату-переходник (рисунок 19), с посадочным местом под плату преобразователя логических уровней, между GPIO пинами RPI и цифровыми пинами meAURIGA, со стягивающими резисторами, для нейтрализации шумов.

После расчета напряжения для каждого электрического компонента робота, все электрические узлы были распаяны, собраны в общую схему, после чего протестированы и закреплены на роботе, а провода аккуратно уложены и скрыты в боксе для проводов (рисунки 20, 21). Для питания общей электрической схемой были выбраны литий ионные аккумуляторы 18650, в количестве двух штук, так как таким образом на вход контроллера подается  $\approx 8.4V$ , что является наиболее оптимальным для спокойного управления двигателями. Для RPI же выделили отдельное питание от обычного пауэрбанка на 5V, это обусловлено тем, что при одном источнике питания RPI может начать питаться от meAURIGA и перегореть.

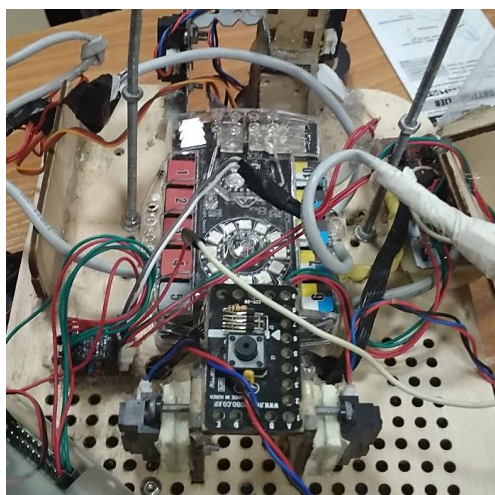


Рисунок 20 – собранная электрическая часть робота

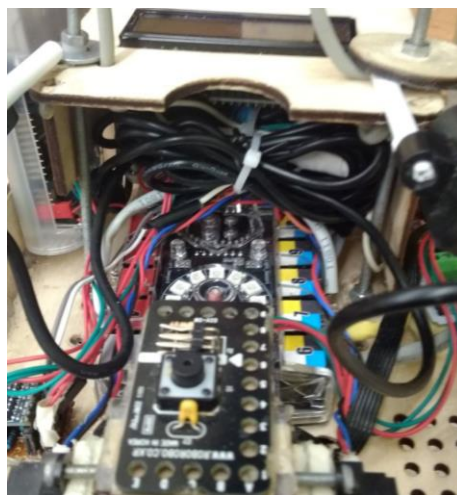


Рисунок 21 –электрическая часть робота после фиксации всех элементов и укладки проводов

## Разработка программного обеспечения

Программное обеспечение робота делится на две глобальные задачи: алгоритмы передвижения и навигации (ПО для meAURIGA), а так же алгоритмы компьютерного зрения (ПО для Raspberry).

Контроллер прошивали через текстовую среду разработки Arduino IDE, так как она наиболее доступна и поддерживает библиотеки для meAURIGA. Этот алгоритм состоит из нескольких подпрограмм: передвижение, общение с RPI, построение карты и навигация по ней. Стоит отметить, что для простоты и точности тестирования ПО для контроллера робота, использовался симулятор Webots Erebos, где были воссозданы некоторые полигоны для проведения миссий и сам робот (рисунки 22, 23). Программу же писали в обычной текстовой среде программирования на языке C++, для удобства тестирования каждая подпрограмма писалась как отдельная программа, и уже позже собиралась в одну основную. Контроль над передвижением осуществляется через оптические энкодеры, встроенные в моторы, для простоты построения карты полигон как-бы мысленно делился на клетки 30х30 см. Во время движений, поворотов, выравниваний и так далее робот постоянно собирает и анализирует показания сторонних датчиков (не дальномеры), что позволяет роботу оперативно реагировать на изменения в окружающей среде. Поворот же осуществляется при помощи гироскопа, располагающегося на самой meAURIGA, и состоит из самого поворота и доворота (выравнивания) для точного попадания в найденный проход. Для более точного перемещения в сильно замкнутых пространствах, используем пропорционально-дифференциальный регулятор, на основе показаний датчиков, что позволяет легко проходить подобные участки, не останавливаясь для выравнивания. Кроме того, при помощи гироскопа производится контроль над положением робота по оси OY, а именно, наличие ям или горок. Для более раннего обнаружения больших ям мы также используем датчик цвета, который способен определить яму еще до того, как центр тяжести перевесил в эту яму.

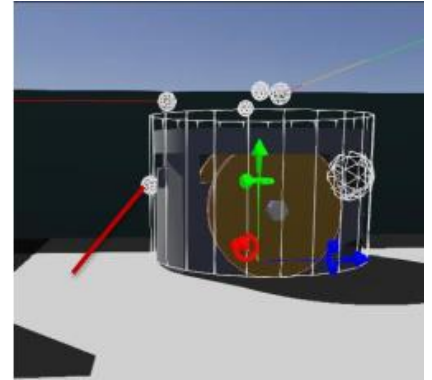


Рисунок 22 – модель робота в симуляторе

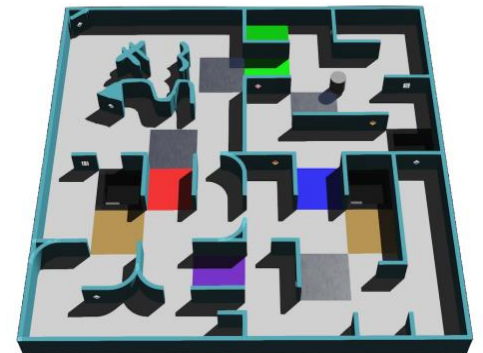


Рисунок 23 – модель полигона для миссии в симуляторе

Картографирование выглядит следующим образом: карта представляет собой довольно большой массив из структур. Каждая структура составит из нескольких однобайтных переменных, хранящих информацию о наличии препятствий на каждом из дальнометров, информацию о наличии жертв или иной информации для спасателей, а также количестве посещений этой ячейки. Для экономии памяти мы использовали по 2 байта на ячейку, так как для хранения информации о наличии препятствий, жертв, ям нам достаточно по одному биту памяти, следовательно при помощи побитового распределения памяти нам удалось сократить её использование в 40 раз, что помогло увеличить максимальный размер карты.

Фрагмент кода, для создания основных элементов хранения карты на C++:

```
struct myNode{
    uint8_t wall=0; //1 байт памяти отвечающий за объектах внутри клетки
    uint8_t count=0; // 1 байт памяти для количества посещений клетки
};
myNode mymap[40][40]; // сама карта

int cord_x=15, cord_y=25; // контроль текущей координаты
int napravlenie=2; // контроль направления робота относительно старта

uint8_t tam_l, tam_r; // для расчета смежных направлений относительно ведущего
```

Таким образом, во время посещения роботом следующей клетки, робот обновляет информацию о ячейке и пересчитывает побочные переменные, необходимые для навигации. Сама же навигация робота строится на основе алгоритмов построения маршрутов в графе: DFS и алгоритм Дейкстры (блок схему можно прочитать в Приложении 4). Робот при посещении новой клетки смотрит, может ли он продолжать свое движение прямо, если нет, смотрит находится ли он на развилке, если вариантов много, он помечает на карте, что тут не дообследованный путь и продолжает свое движение, если же он в тупике, строит кратчайший маршрут до последней (ближайшей) не дообследованной развилки и продолжает его обследование уже с нее. Окончание миссии же может быть вызвано окончанием памяти, выходом из строя какого-либо компонента, или же заикливанием карты (в данном случае будет считаться, что робот полностью обследовал помещение, рисунок 24).



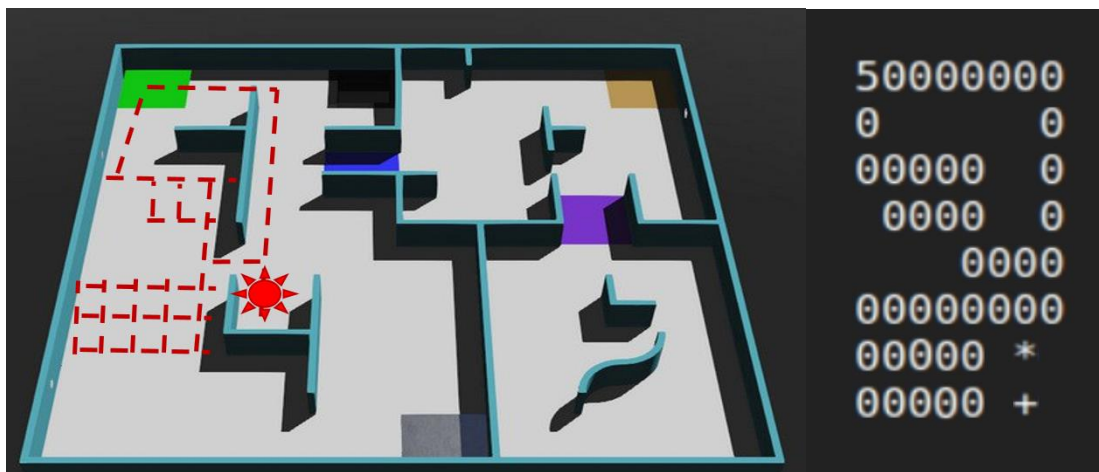


Рисунок 24- вариант вывода карты на экран (5-стартовая ячейка, \* - робот, + - направление, в которое он смотрит)

Для программирования RPI, также были составлены несколько разных программ, которые позже собирались в одну. Так, для более точного определения возможных знаков или надписей на языке C++ был разработан алгоритм чтения букв, который мы назвали «метод точек», а для быстрого реагирования на изменения в окружающей среде был обучен фреймворк искусственного интеллекта YOLO v8, написанный на языке Python.

В регламенте соревнований роботов спасателей, которого мы придерживались для формулировки задания миссии, от ПСР требуют функции определения знаков опасности и чтения некоторых букв (рисунок 25). Для этого нам понадобилось проводить бинаризацию на цвета, характерные для знаков опасности и далее – само определение знаков/чтение букв. Однако мы не можем предсказать освещение во время миссии, поэтому нам понадобилось разработать алгоритм смены параметров автобинаризации в зависимости от освещения на поле. Алгоритм, который мы придумали ищет на изображении самый светлый, самый темный, среднестатистическое значение пикселя и формулой рассчитывает данные для бинаризации на черный (чем темнее освещение на поле, тем больше делителей). Таким образом, робот может находить линию даже в сумраке (рисунок 26, 27).

Примерная формула, которую мы используем на псевдокоде:

текущий\_пиксель < (min + cr\_знач) / 2 + min



Рисунок 25 – знаки, определяемые нами

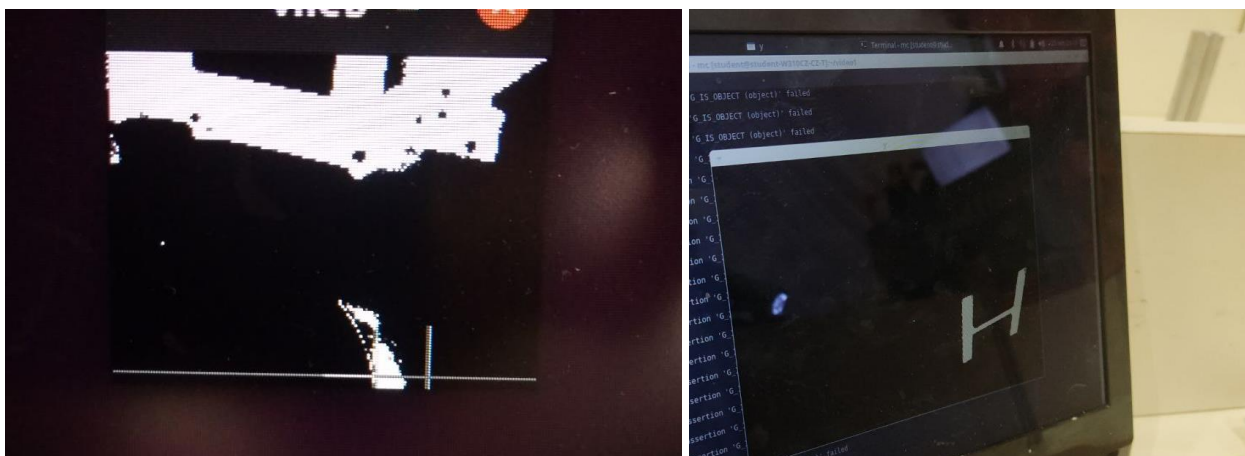


Рисунок 26,27 – результат работы алгоритма автобинаризации

Само обнаружение же происходит с помощью камеры и библиотеки OpenCV 4. Для начала настраивается бинаризация (смотрится по количеству цветных пикселей) для одного из цветов: красного, жёлтого и чёрного. После бинаризации определяем контуры и проверяем самые большие контуры, подходят ли они под параметры знаков. Если обнаружены чёрные контуры, подходящего размера, то вычисляем 3 точки прямоугольника этого контура: верхнюю срединную и нижнюю срединную, с помощью них мы и определяем какая буква и ищем в них контур на чёрной бинаризации: если верхний и нижний - черные, то это s, если нижний и верхний - не черные - то это H, а если нижний – чёрный - то это U (рисунок 28, 29). Как можно заметить, данный метод основывается на различие начертания букв. Сейчас нам достаточно всего 2-х точек, так как мы читаем всего 3 буквы, однако с увеличением количества букв, необходимых для чтения нам достаточно просто увеличить количество точек и их расположение.



Рисунок 28 – «метод точек» наглядно

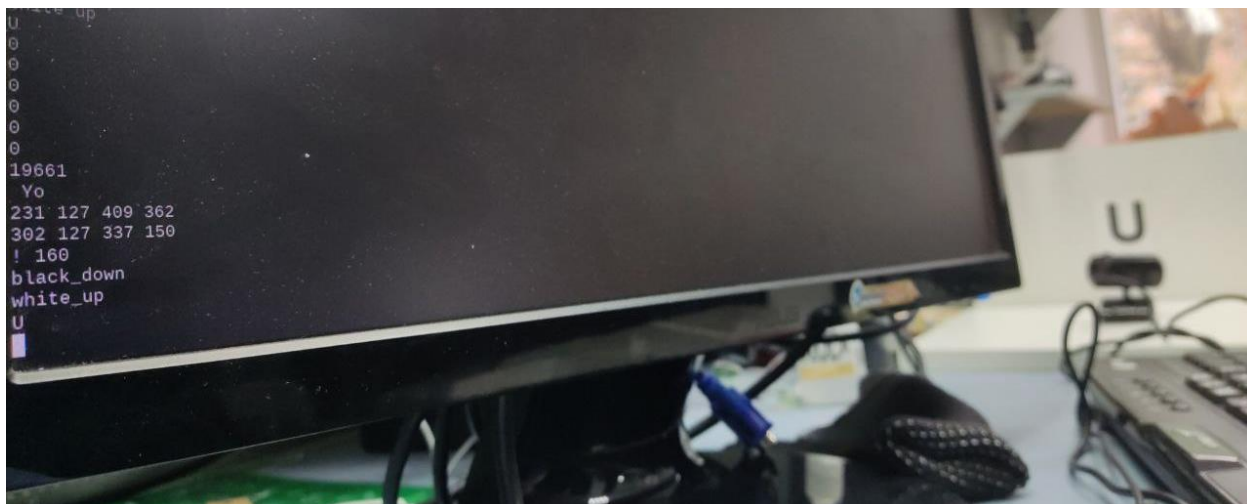


Рисунок 29 – результат работы алгоритма «метод точек»

Еще одним вариантом распознавания букв, а также вариантом распознавания жертв стало обучение фреймворка искусственного интеллекта YOLO v8 (рисунок 30). Механизм работы нашей программы предельно простой: у нас имеется заранее натренированная модель на датасете поля, преобразуем каждый кадр в тензор, сравниваем с тензора и в модели, получаем боксы детектирования, если условный "коэффициент сходства" в подобном боксе больше определённого значения, то узнаем, что за жертва/буква привязана к нему. Для реализации этого алгоритма собственно нужно было подготовить сам датасет: снять фото с камеры робота, и с помощью спец инструментов провести разметку. Так как мы использовали нейросеть YOLO v8, то и тренировали мы модель YOLO v8 (n/m/x) на нашем датасете. Далее экспортировали в расширение torchscript, передавали файл raspberries, т.к. программируем RPI на C++, а YOLO обучается на Python - использовалась библиотека libtorch.



Рисунок 30 – итог работы алгоритма распознавания на YOLO v8

## Тестирование робота

Перед тестированием самого робота на каждом этапе его разработки проводились тестирования каждого элемента ПСР. Так, конструкция в первую очередь тестируется на прочность и если ее недостаточно, то элементы дорабатываем. Таким образом в результате таких тестов мы заменили пластиковые и картонные детали на металлические и фанерные. При тестировании датчиков также приходилось проверять, что кабели не мешают их обзору, в результате чего и появился короб для проводов. Кроме того, при разработке шасси нам приходилось постоянно тестировать его проходимость - забираться на горки, проезжать через лежащие препятствия разной сложности. В результате у нас есть робот с хорошей проходимостью.

Так как для подключения большей части датчиков к meAURIGA нам приходилось паять, основным тестированием датчиков было проверка на отсутствие короткого замыкания. Кроме того, перед непосредственно пайкой и подключением, нам было необходимо отдельно проверять датчики при помощи Arduino Uno на работоспособность.

Тестирование ПО, вне симулятора производилось непосредственно на поле при выполнении миссии. Для обучения YOLO мы запустили робота и сохраняли кадры с камеры на роботе раз в 0,5/1 секунд. А позже для тестирования распознавания жертв и букв мы использовали распечатанные буквы, так как это удобнее, чем на поле.

Тестирования прохождения самой миссии проводили на полигонах для соревнований макетов роботов спасателей. Полигоны собраны из ЛДСП и симулируют выполняемую миссию (рисунки 31,32,33), также на полигонах можно встретить препятствия, мусор, подъемы-спуски, рампы и лестницы. Робот же выполнял миссии в нескольких режимах: с передачей информации людям, и обособлено, без вмешательств, так как нам не интересны показания датчиков, нам было достаточно подключить планшет к RPI и отслеживать, что она пишет в командную строку. Таким образом мы можем оперативно получать информацию о нахождении жертв прямо на планшет.

Видеозаписи большей части испытаний можно посмотреть на нашем облачном диске (см. ссылки на авторов).





Рисунок 31, 32 – примеры полей для испытаний

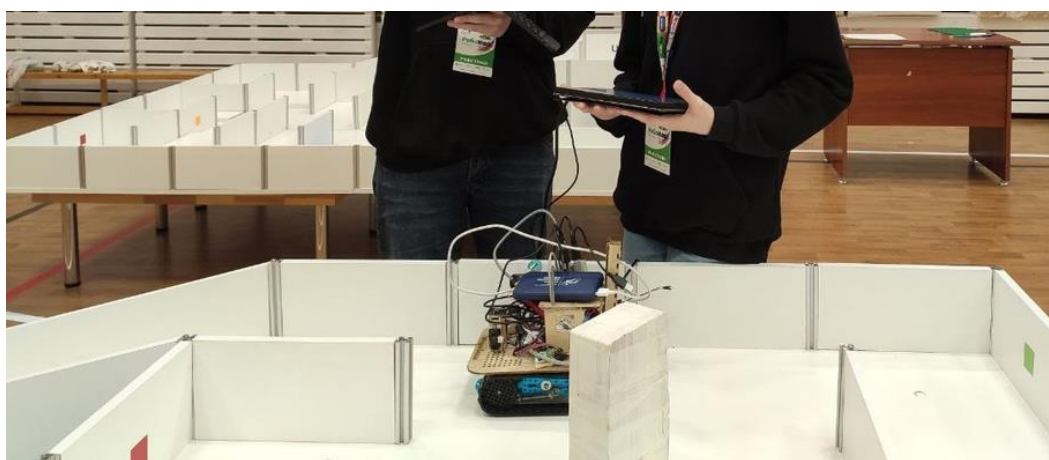


Рисунок 33 – отладка робота на поле

Таким образом, после доработки промежуточных результатов разработки на каждом из этапов, были пройдены несколько этапов разработки и итераций робота (рисунки 34-37), получила и протестировала рабочий экспериментальный прототип робота для помощи проведения поисково-спасательных операций (рисунок 38).

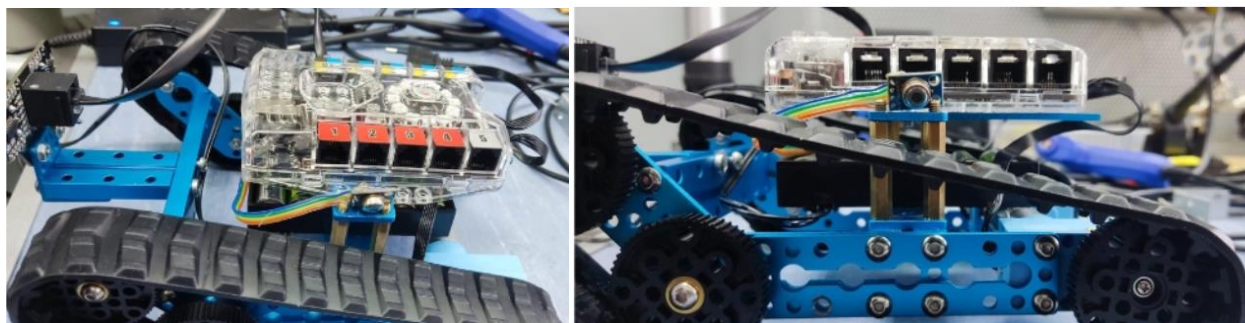


Рисунок 34,35 – первая версия робота

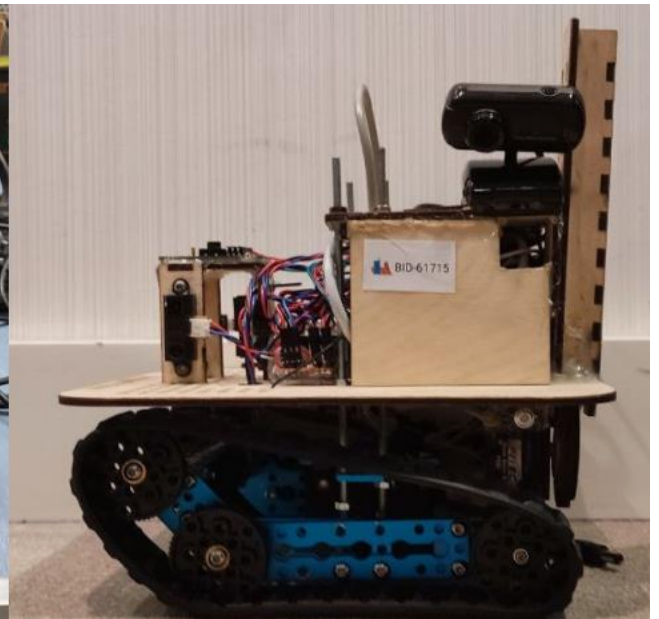
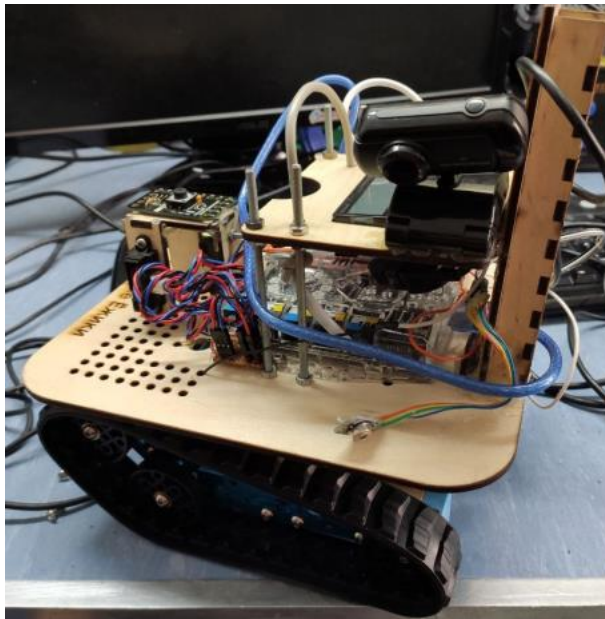


Рисунок 35,36 – вторая версия робота

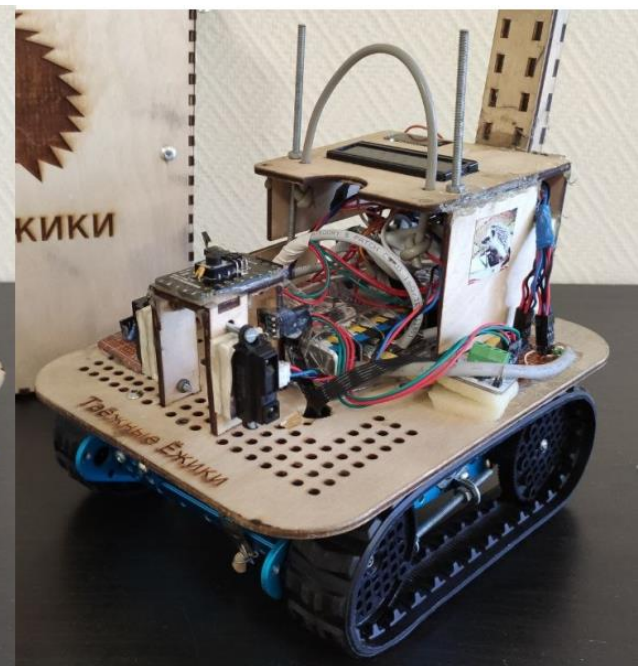
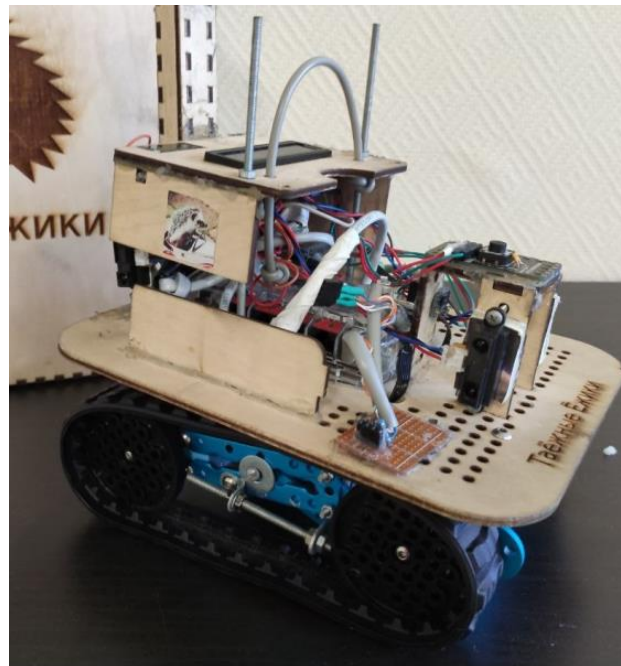


Рисунок 36,37 – третья версия робота



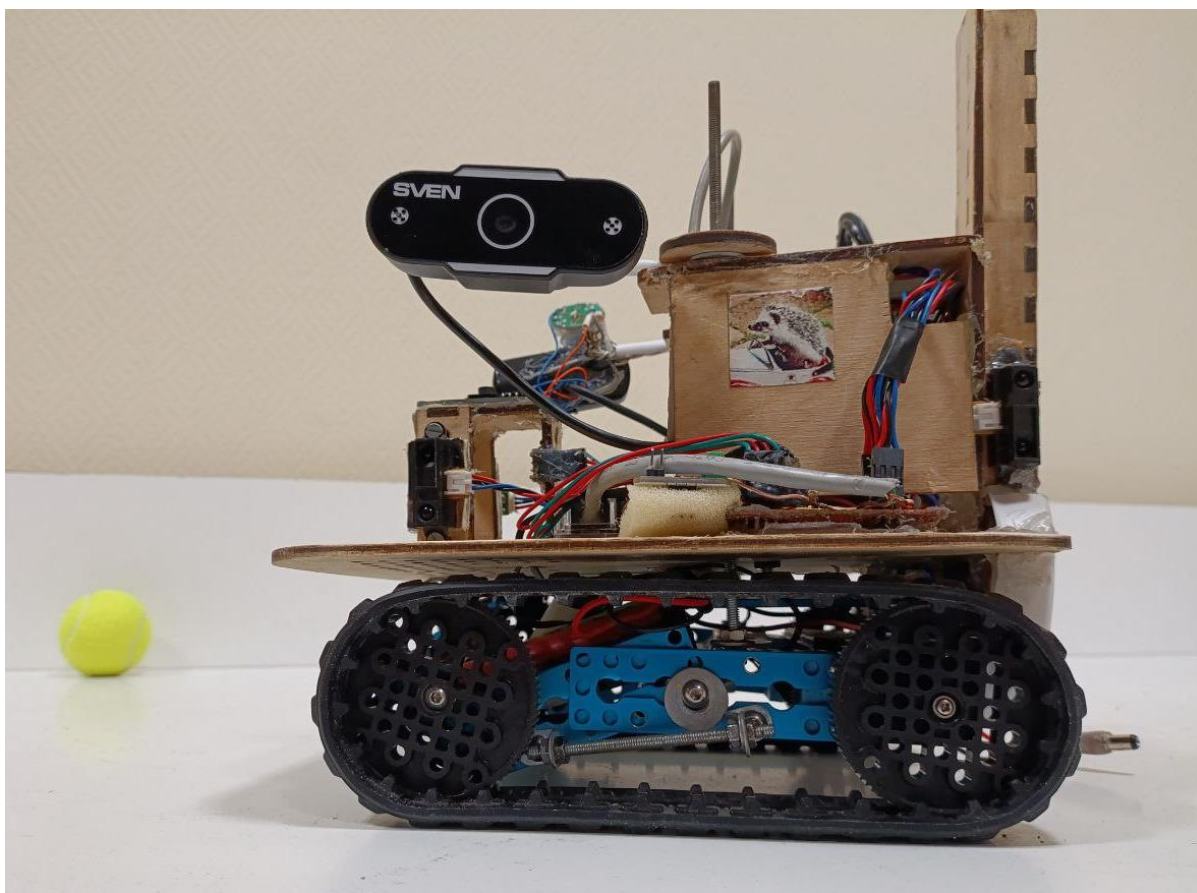


Рисунок 38 – итоговый прототип робота для проведения поисково-спасательных операций

## **ЗАКЛЮЧЕНИЕ**

### **Достигнутые результаты**

В результате проделанной работы желаемый результат был достигнут, был разработан и протестирован рабочий прототип робота для проведения поисково-спасательных операций. Данный робот способен в автономном режиме обследовать территорию, пострадавшую при ЧС, строить его карту и определять наличие знаков опасности.

### **Проблемы, с которыми мы столкнулись на этапе реализации**

Во время разработки и модифицирования нашего робота команда столкнулась со следующими проблемами.

1) При разработке чертежей робота, необходимо было сразу предусмотреть все необходимые отверстия и крепежи, из-за постоянной модификации конструкции, необходимо заново переделывать их. (основной платформы у нас 3 варианта, последнюю из которых также пришлось модифицировать).

2) Освещение на тестовых полях было очень плохим и к тому же менялось с течением суток, настройка бинаризации очень долгий процесс, из-за чего было принято решение разработать самодельный алгоритм автобинаризации.

3) Изначально мы передавали данные с Raspberry на meAURIGA через Serial и по какой-то причине Raspberry пыталась питаться через meAURIGA, а не через пауэрбанк, из-за чего сгорала Raspberry. В результате было принято решение перейти с serial на UART. А позже – и на логические сигналы. Возникшая проблема была решена.

4) Зачастую у Raspberry сбивается время, и программа не компилируется, выводя ошибку «Время изменения файла находится в будущем», для решения данной проблемы мы копируем новый каталог под новым временем и сносим старый с неправильным.

5) Библиотеки MeAuriga.h. Мы не могли найти подходящие команды для программирования робота. Мы провели анализ и разбор библиотеки и смогли определить необходимую команду.

6) Для масштабной и подробной карты необходимо много памяти, которой сильно не хватало. Решением стало использование методов побитовых операций и сжатие переменных до 1 байтных.

7) Raspberry постоянно поедает sd карты, они ломаются и порой даже не поддаются форматированию, мы не смогли до конца решить данную проблему, поэтому просто делаем бэкапы и заменили microSD на жесткий диск

8) Обучение нейросетей, оказалось довольно удача затратным занятием, так как

очень сложно предугадать, сколько ресурсов вам понадобится, чтобы нейросеть обучилась, но не переобучилась. В связи с чем было сделано множество вариаций этой нейросети и со временем научились это высчитывать.

9) Технологии машинного обучения тратят довольно много ресурсов, к тому же мы не отказывались от метода точек, в связи с чем возникла проблема с использованием Raspberry Pi Zero и произошёл переход на Raspberry Pi 4.

### **Планы на дальнейшую доработку**

Для данного проекта был закуплен датчик– лидар. В связи с чем уже ведется разработка корпуса под него, а также модернизация алгоритма навигации, так как теперь по сути мы можем анализировать не 5 лазеров, а все 360 и данная модернизация позволит улучшить качество построенных нами карт. Также планируется продолжать обучать нейросеть для определения человека и степень его повреждения при ЧС.

### Список использованных источников

1. Основные показатели и оценка состояния защиты населения и территорий от ЧС природного и техногенного характера / статистические данные за 2023 год / МЧС России [Электронный ресурс]. – URL: <https://mchs.gov.ru/uploads/document/2024-04-10/dc46867323bf3868fecf566e1e34e4d5.pdf> (дата обращения — 11.11.2024)
2. Мажуховский Э.И. Аварийно-спасательные и другие неотложные работы // Технологии гражданской безопасности. – 2006. – Т. 3, № 2. – С. 88–92.
3. Robin R. Murph, Daniele Nardi, Paolo Fiorini, Aydan M. Erkmen, Satoshi Tadokoro, Adam Jacoff, Howie Choset / Search and Rescue Robotics // Springer Handbook of Robotics — 2008 [Electronic resource]. – URL: [https://www.researchgate.net/profile/Aydan-Erkmen/publication/225916883\\_Search\\_and\\_Rescue\\_Robotics/links/02bfe50d21d1b8a6b6000000/Search-andRescue-Robotics.pdf](https://www.researchgate.net/profile/Aydan-Erkmen/publication/225916883_Search_and_Rescue_Robotics/links/02bfe50d21d1b8a6b6000000/Search-andRescue-Robotics.pdf) (дата обращения — 11.11.2024)
4. В.П. Малышев, О.В. Виноградов / Возможные направления оптимизации состава мероприятий при ликвидации долговременных последствий крупномасштабных чрезвычайных ситуаций // Технологии гражданской обороны, том 17, 2020 год, №3 (65) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/vozmozhnye-napravleniya-optimizatsii-sostava-meropriyatiy-pri-likvidatsii-dolgovremennyh-posledstviy-krupnomasshtabnyh> (дата обращения — 15.11.2024)
5. С. В. Ефимов, Н. И. Попов / Применение робототехники в локализации последствий ЧС [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/primeneniye-robototekhniki-v-lokalizatsii-posledstviy-chs> (дата обращения — 11.11.2024)
6. С.Г. Цариченко / Экстремальная робототехника в МЧС России — задачи и перспективы [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/ekstremalnaya-robototekhnika-v-mchs-rossii-zadachi-i-perspektivy> (дата обращения — 12.11.2024)
7. Подводный телеуправляемый аппарат Гном / Аварийно спасательная техника МЧС России [Электронный ресурс]. – URL: <https://mchs.gov.ru/ministerstvo/o-ministerstve/tehnika/avariyno-spasatelnye-tehnika/podvodnyy-teleupravlyayemyy-apparat-gnom> (дата обращения — 15.11.2024)
8. Многоцелевая безэкипажная гидравлическая платформа Сивуч / АКВАРОБОТЕК / Официальный сайт компании [Электронный ресурс]. – URL: <https://aquarobotics.ru/> (дата обращения — 15.11.2024)
9. National Technology and Engineering Solutions of Sandia. (n.d.) Gemini-Scout Mine Rescue Vehicle. [Electronic resource]. – URL: <https://www.sandia.gov/research/gemini-scout-mine-rescue-vehicle/> (дата обращения — 19.10.2024).

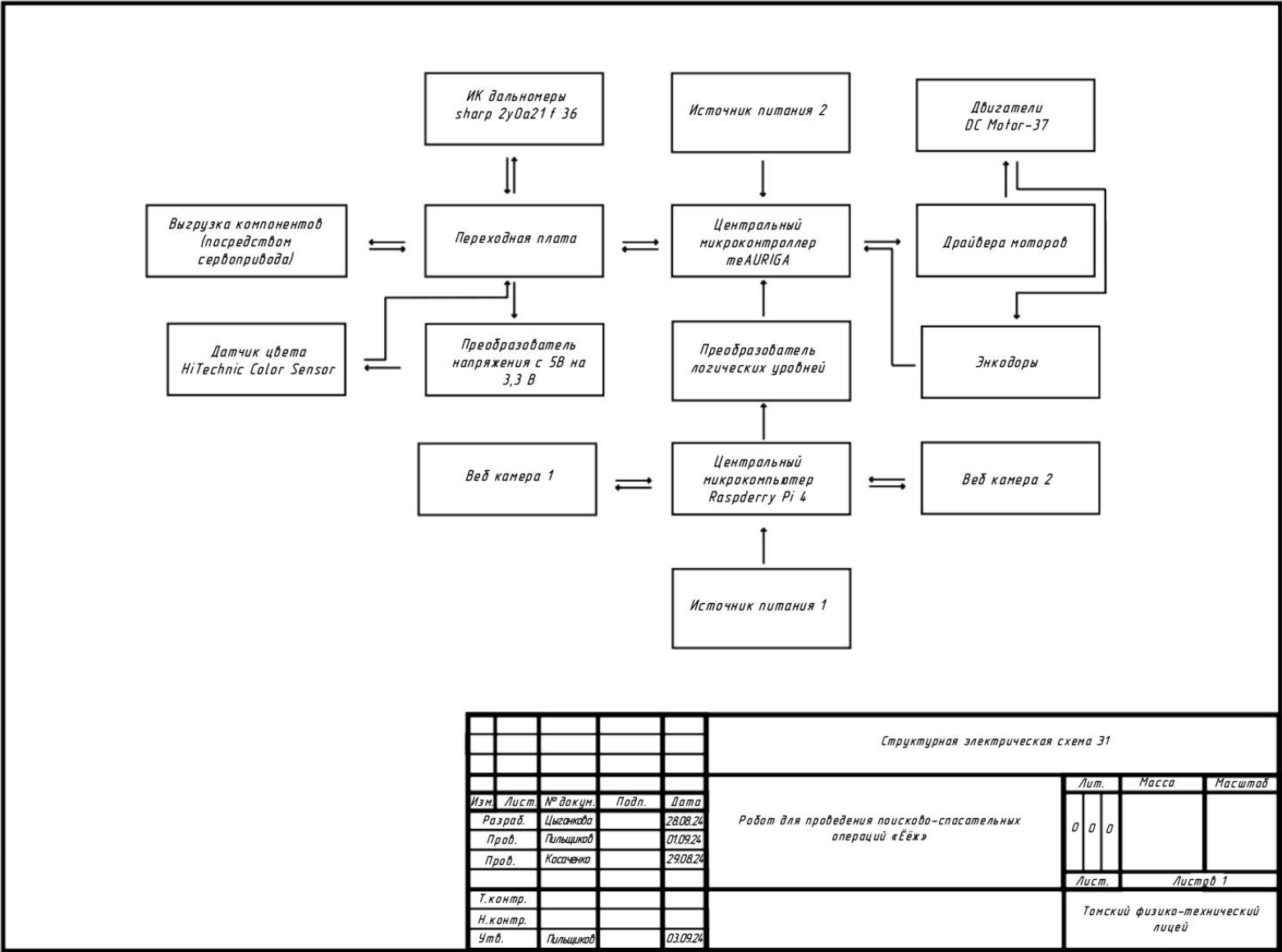
10. А. Ш. Буреев, Д. С. Жданов, Я. В. Костелей, Е. В. Голобокова / Спасательные роботы: краткий обзор технических решений // Технологии безопасности жизнедеятельности. 2023. № 3. С. 78–87 [Электронный ресурс]. – URL: <https://vital.lib.tsu.ru/vital/access/services/Download/koha:001016767/SOURCE1> (дата обращения — 10.01.2025).
11. Ralston, J.C. & Hainsworth, D.W. (1998) The Numbat: A remotely controlled mine emergency response vehicle. In: Field and Service Robotics. Springer London. pp. 53–59.

**Сравнительная таблица аналогов**

Робот	Проходимость	Автономность	Комментарии
РР-Г1	Гусиные шасси/высокая	Нет/управляется оператором	Функциональное назначение – измерение уровня радиации. Серийный выпуск не осуществлялся.
Мобот-Ч-ХВ	Гусиные шасси/высокая	Нет/управляется оператором	Функциональное назначение – измерение уровня радиации. Серийный выпуск не осуществлялся.
Gemini scout	Гусиные шасси/высокая	да	Полученная информация передается оператору для принятия решения
Numbat	Колесные шасси/снижена	Нет/управляется оператором	Полученная информация передается оператору для принятия решения
Ёёж	Гусиные шасси/высокая	да	Реализована возможность самостоятельного принятия решений (выдача аптечки первой помощи при обнаружении пострадавшего).

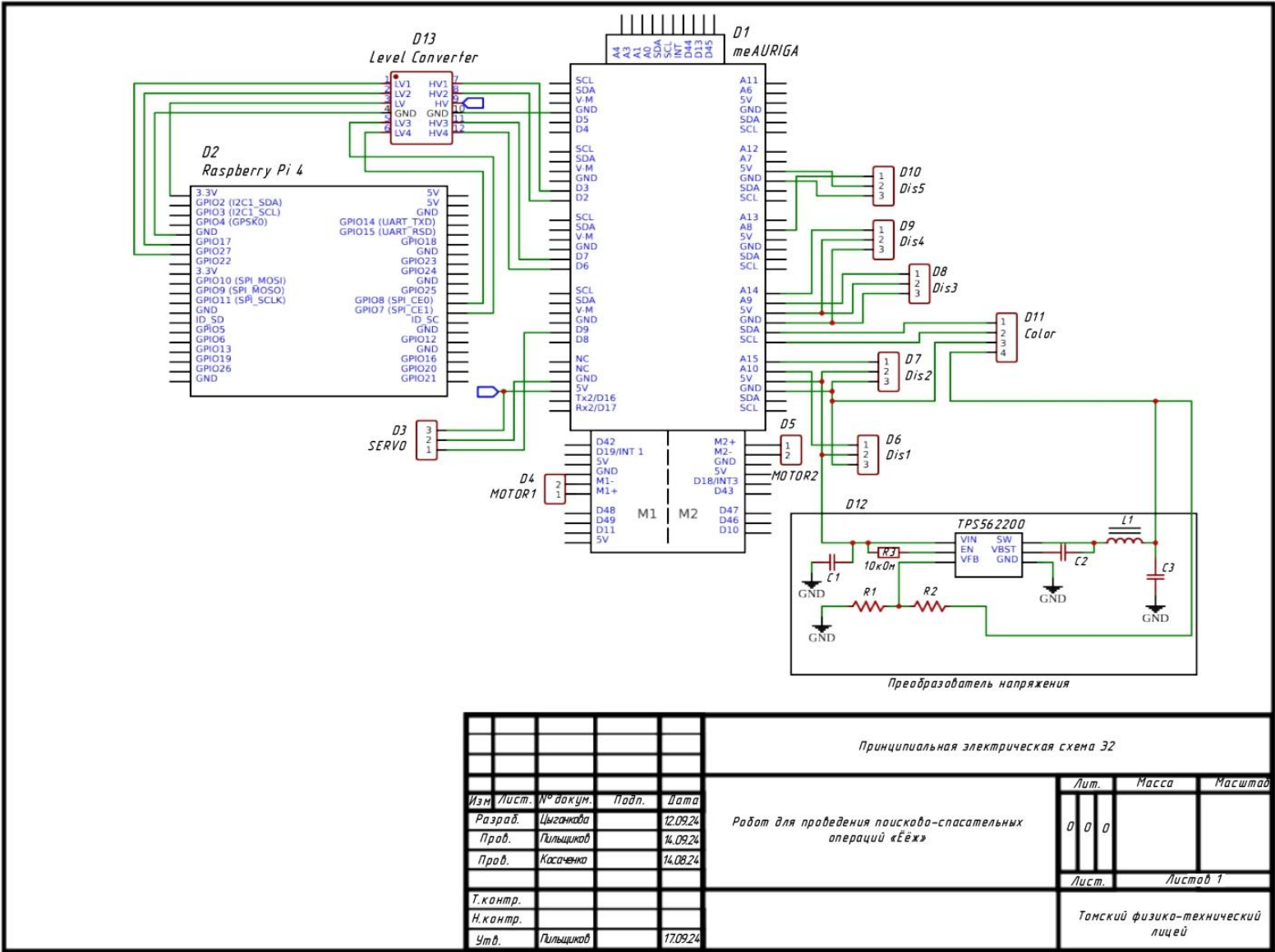


Структурная схема Э1



					Структурная электрическая схема Э1							
Изм.	Лист	№ докум.	Подп.	Дата	Работ для проведения поисково-спасательных операций «Ёёж»			Лит.	Масса	Масштаб		
Разраб.	Цыганкова			28.08.24				0	0	0		
Пров.	Пильщикова			01.09.24								
Пров.	Косаченко			29.08.24								
								Лист.	Листов 1			
Т.контр.					Томский физико-технический лицей							
Н.контр.												
Утв.	Пильщикова			03.09.24								

Электрическая схема Э2



### Блок-схема алгоритма навигации

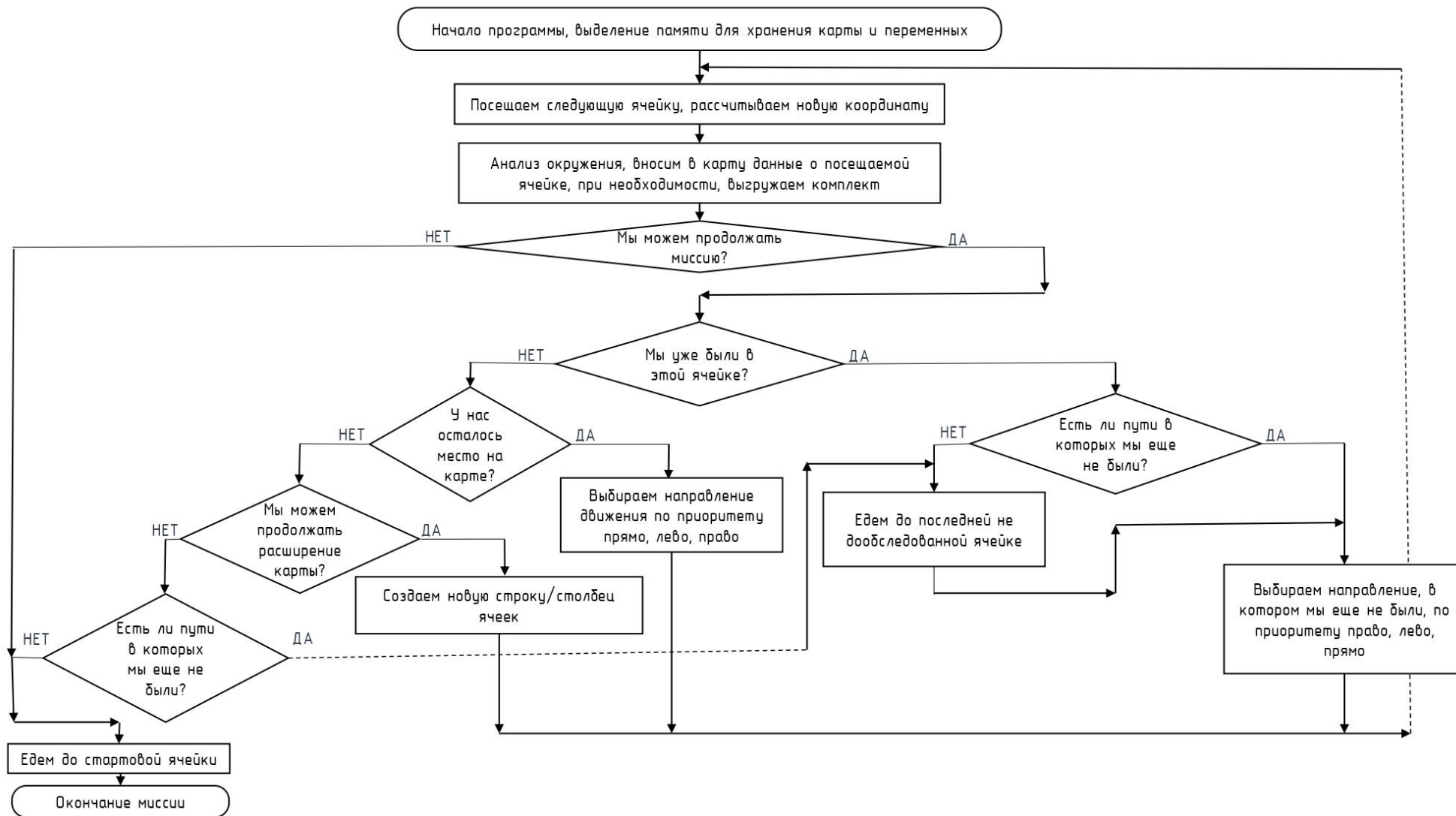
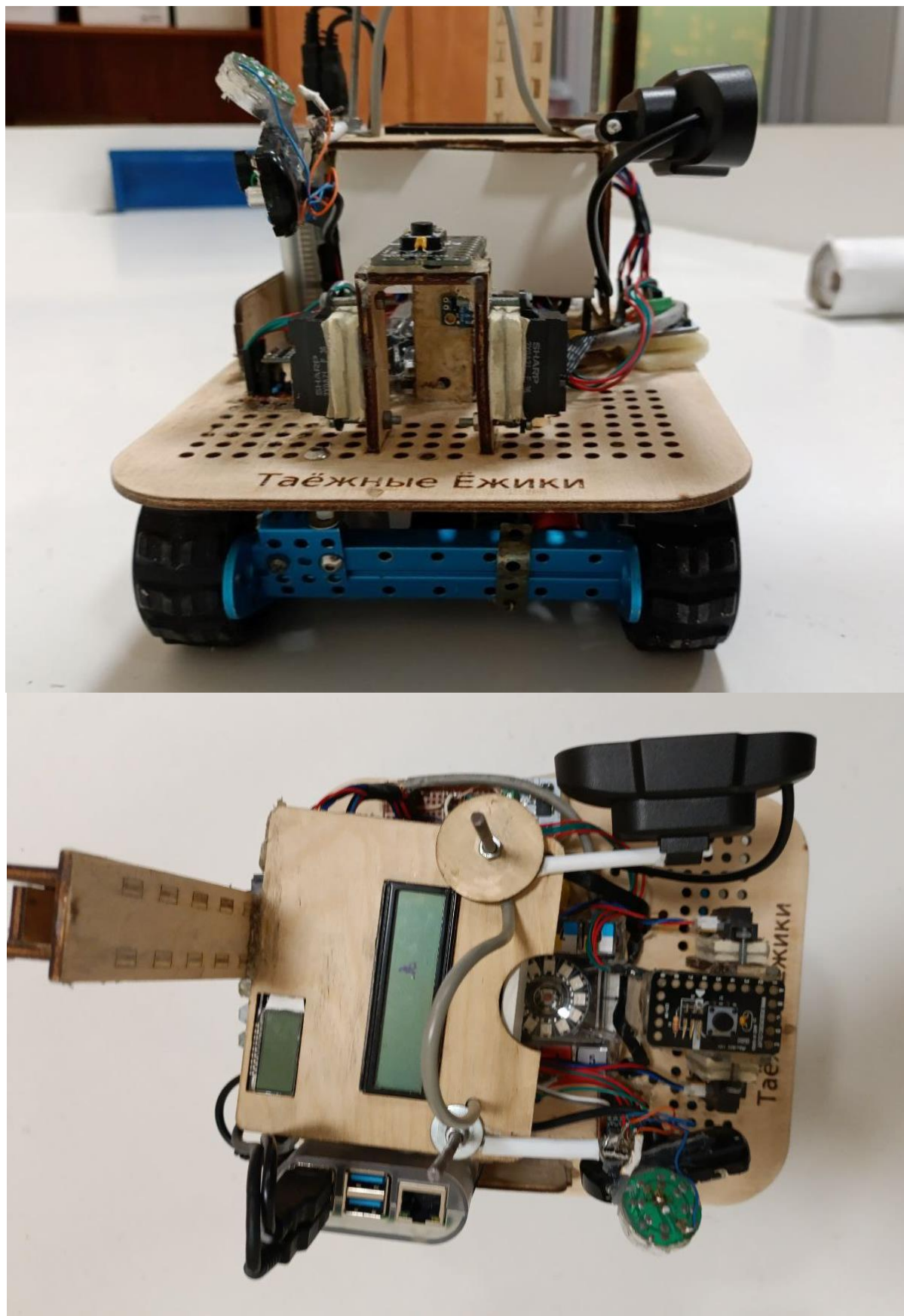
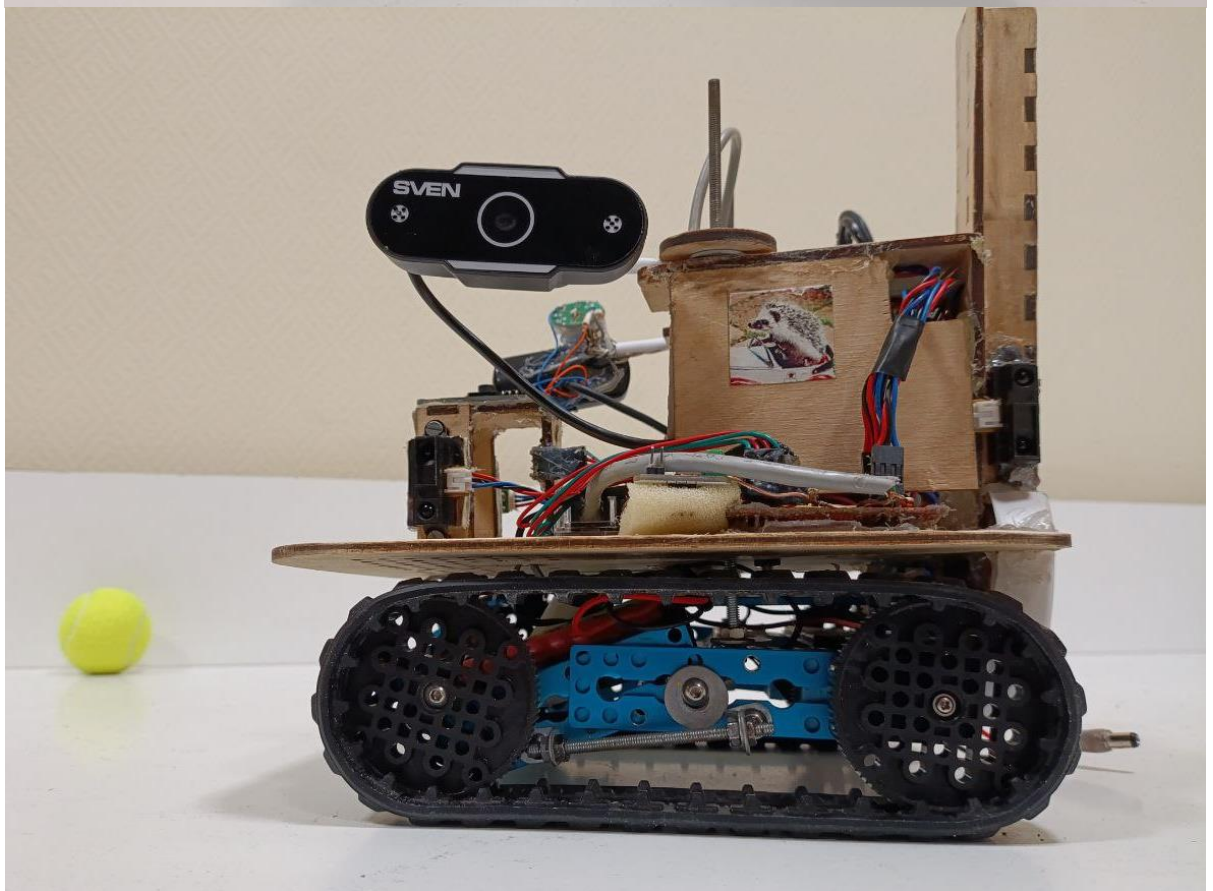
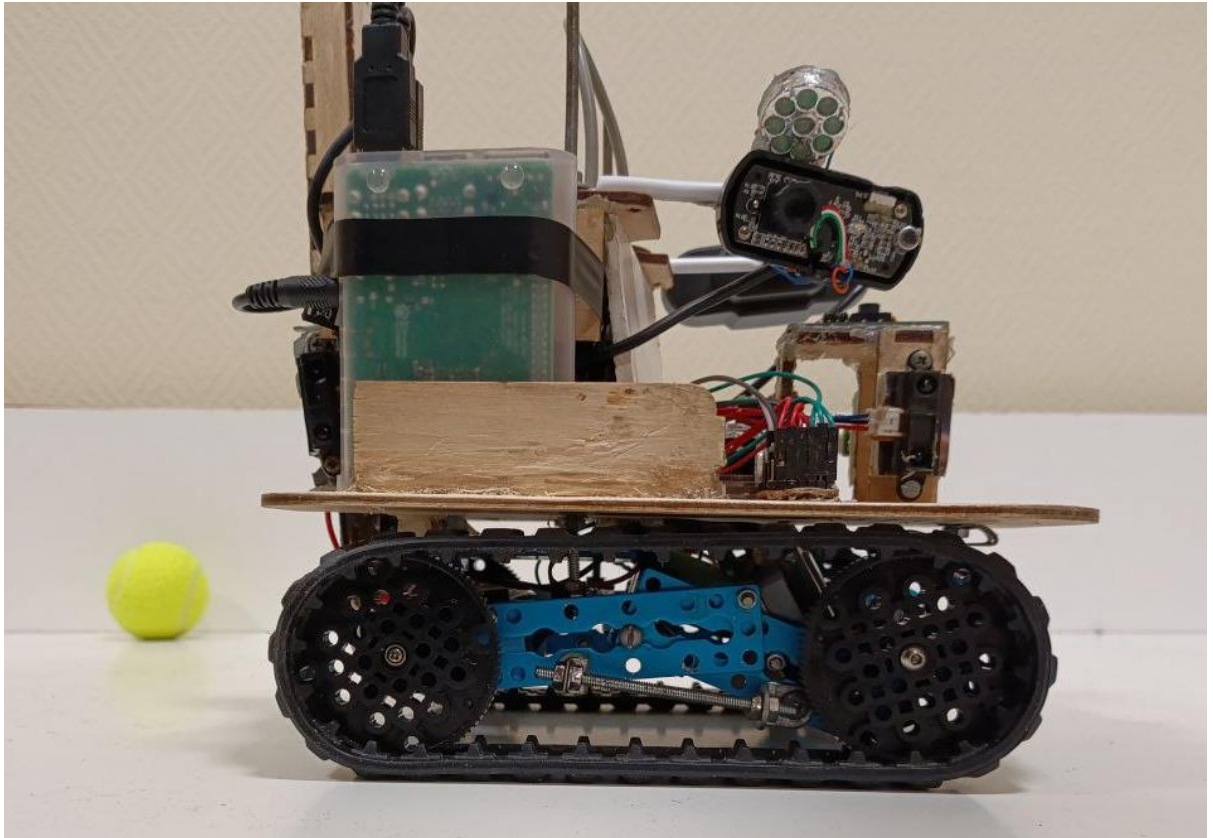
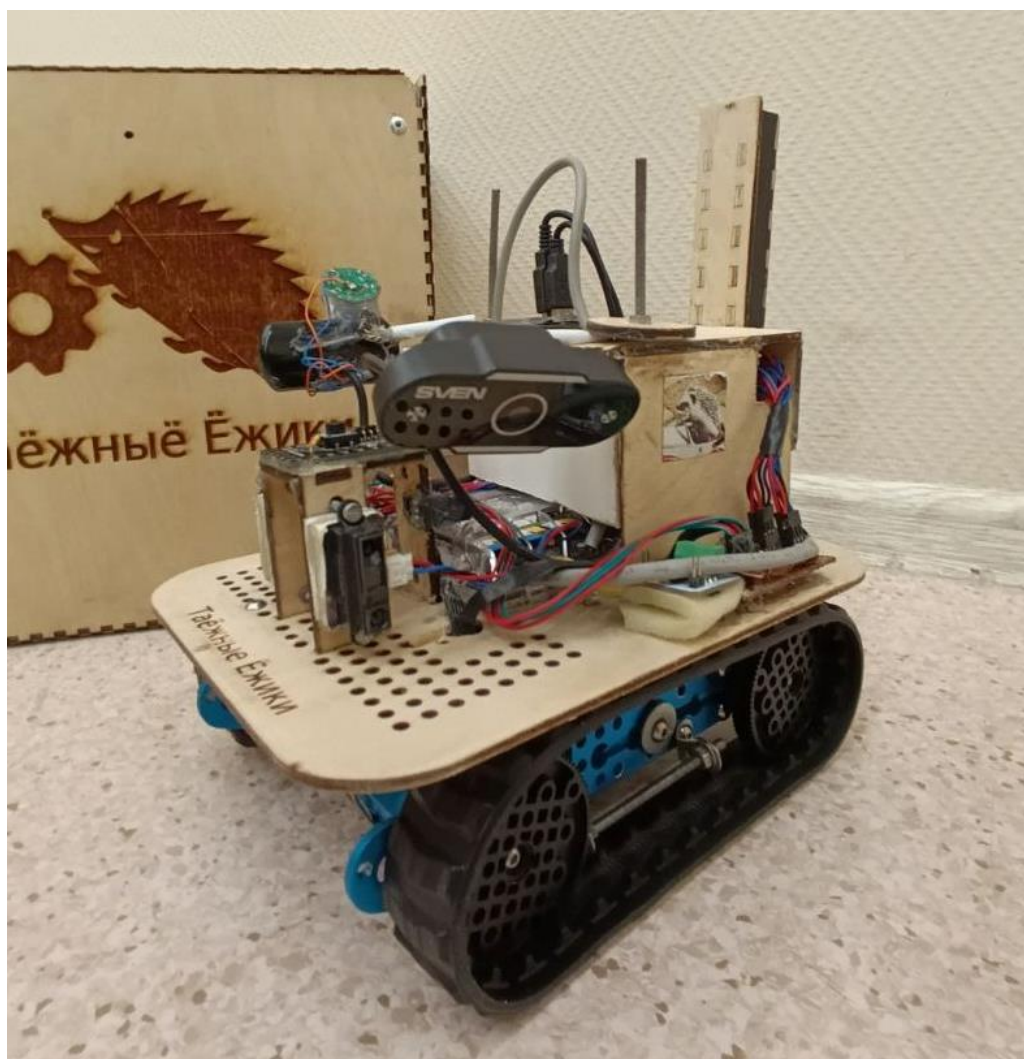


Фото итогового прототипа ПСР









## **Ссылки на нас**

Здесь вы можете подробнее прочитать про этот и другие проекты команды «Таёжные Ёжики», изучить чертежи, код, датасеты, а также посмотреть на все испытания и тесты робота.

<https://github.com/Tsygankova-Maria/TE2022/tree/main> (наш GitHub)

<https://drive.google.com/drive/folders/1fmRRXI1r2HDXeKdHgOfJ3aFNQoEvYZH1?usp=sharing> (облачный диск, где собираются видео испытаний робота)

<https://www.youtube.com/@user-cj5jf6zu6l> (наш YouTube)

<https://www.youtube.com/@SergKosachenko> (YouTube нашего тренера, здесь можно посмотреть некоторые записи испытаний)

## **Контактные данные авторов**

М.С. Цыганкова: [maria.tsygankova.tftl@gmail.com](mailto:maria.tsygankova.tftl@gmail.com)

Г.А. Пильщиков: [p.g.a.2019@mail.ru](mailto:p.g.a.2019@mail.ru)

С.В. Косаченко: [kosachenkosv@yandex.ru](mailto:kosachenkosv@yandex.ru)