

Робот удаленного присутствия



robotup.github.io/robot

Авторы

Федор Родин
Иван Коновалов
Игорь Малыш

Руководитель

Андрей Мухачёв

ГБОУ «Школа №2033»

Москва, 2020

Создать

универсальную гусеничную платформу удаленного присутствия

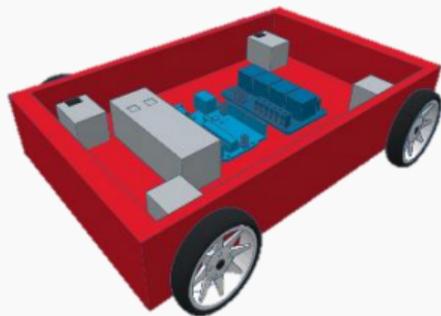
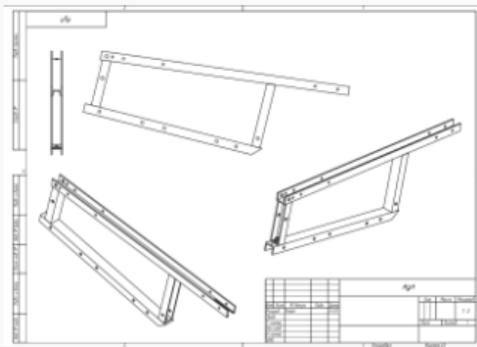
Для

- помощи человеку

И решения

- различных бытовых задач
- специальных задач

Задачи проекта



1. Проанализировать устройство существующих гусеничных роботов
2. Создать модель надежного и проходимого робота
3. Разработать подвеску
4. Собрать прототип
5. Разработать и установить электронную начинку
6. Разработать полный комплект ПО для управления роботом
7. Провести испытания
8. По результатам испытаний доработать робота

Гусеничные роботы



Робот-спасатель, МЧС



WALL-E, Buy 'n Large



GROVER, NASA, Гренландия

Первый прототип

Корпус – конструктор **Lego**

Двигатели – **Lego 1**

Питание – сборки **батарей AA**

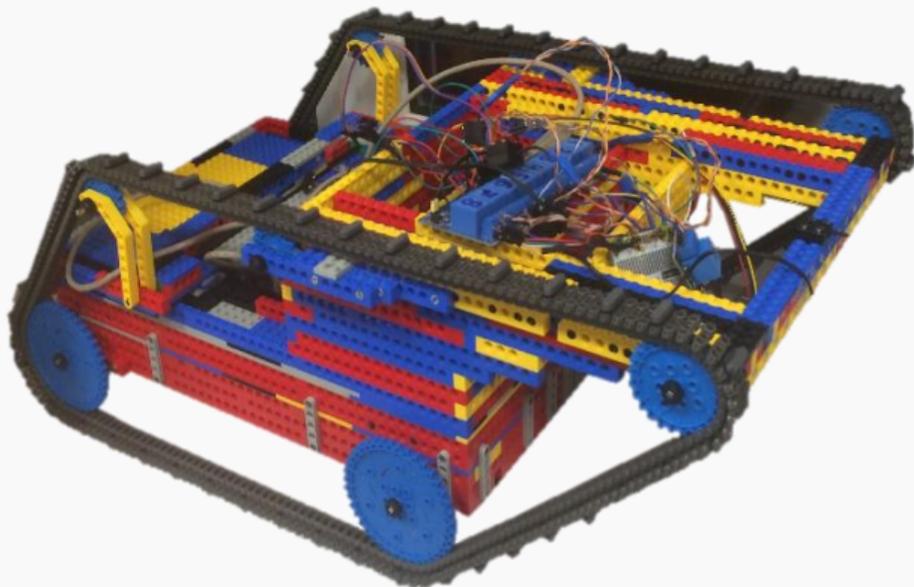
Колесная схема

Крайне плохо маневрировал,
не мог разворачиваться на месте

*На нем тестировались схемы
редукторов и первые версии
ПО управления*



Второй прототип



Версия с двигателями Lego 1

Корпус – конструктор **Lego**
Двигатели – **Lego 1** / Мотор-редуктор
Питание – сборки батарей **AA / 12 В**
Аккумулятор

Гусеничная схема

Конструкция обладала крайней
ненадежностью

Испытания второго прототипа

Конструкция обладала крайней ненадежностью

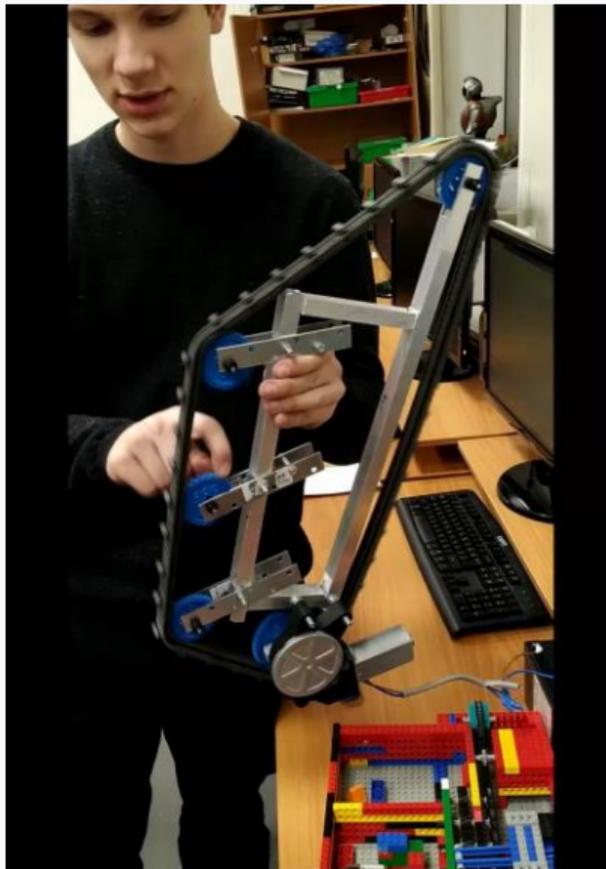
В процессе испытаний второй прототип разрушился

Но по итогам испытаний второго прототипа **гусеничная схема** признана удачной

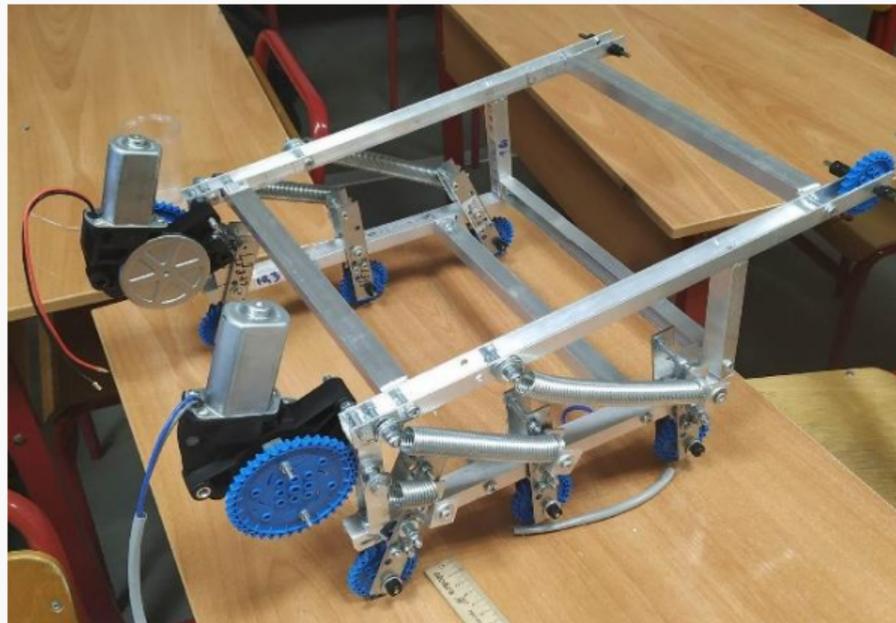
*На видео версия с **Мотор-редукторами**
в качестве двигателей*



Опытный образец



Было решено с учетом опыта прототипов изготовить опытный образец с усиленной конструкцией из **металла**



Опытный образец



Тележка робота

Размеры тележки	- 56x40x27 см
Масса	- 7.5 кг
Полезная нагрузка	- до 10 кг
Скорость	- 4 км/ч

Каркас – **алюминиевые профили**, скрепляются болтовыми соединениями М4/М5/М6

Обшит алюминиевым листом и прозрачным пластиком

Гусеницы – пластиковые траки VEX, с **грунтозацепами** либо без них

Внутри расположен **герметичный отсек** для электроники

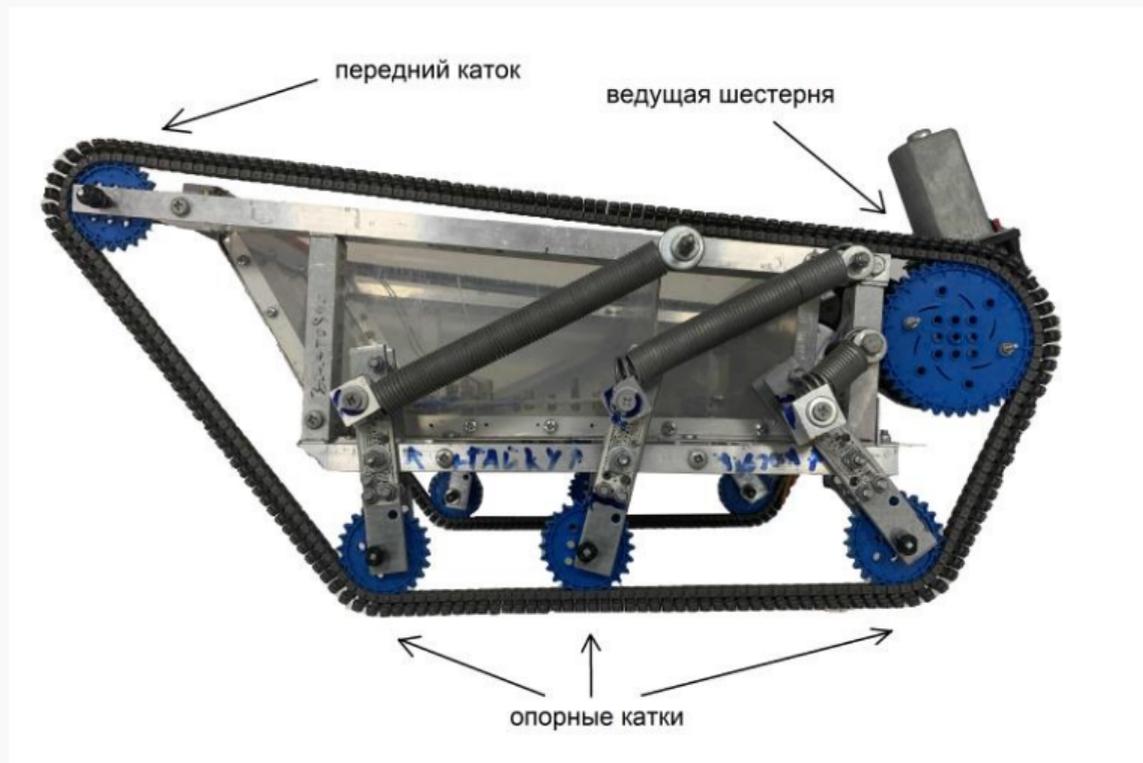
Подвеска Кристи

Преимущества

- высокая прочность
- простота сборки и ремонта
- амортизация
- проходимость
- грузоподъемность

Недостатки

- занимает много места

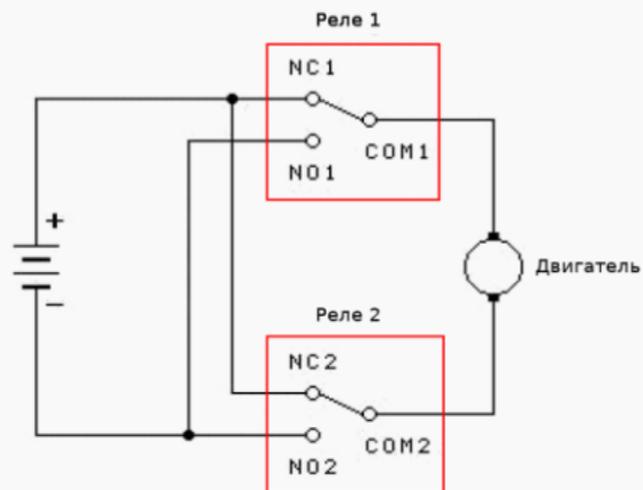




Тяговые двигатели:
2 мотор-редуктора от
стеклоподъемника автомобиля

- напряжение питания **12В**
- мощность **70Вт**
- большой крутящий момент
- малые габариты
- дешевизна

Каждый двигатель управляется 2-мя реле по схеме **H-моста**:



Свинцово-кислотный аккумулятор

напряжением **12 В**

емкостью **7 А*ч**

Одного заряда хватает на **1.5 часа работы**





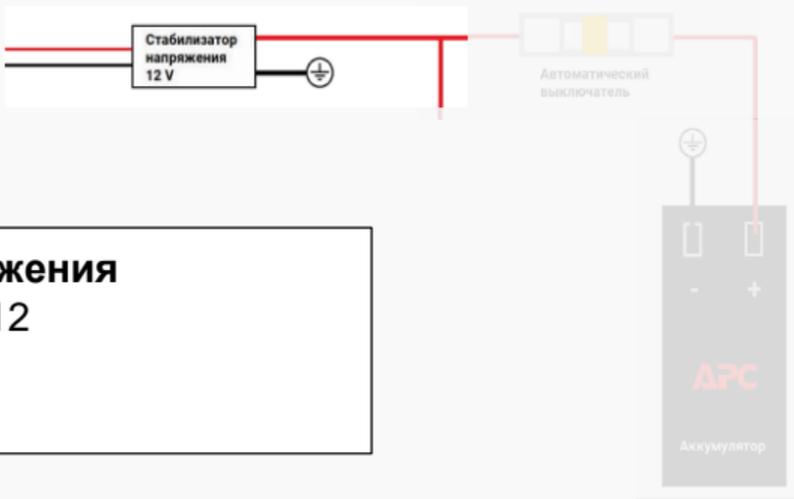
Автоматизированный комплекс по замене аккумуляторов

позволит не тратить время на
зарядку

*Меняет севший аккумулятор на
заряженный*

Автоматический выключатель

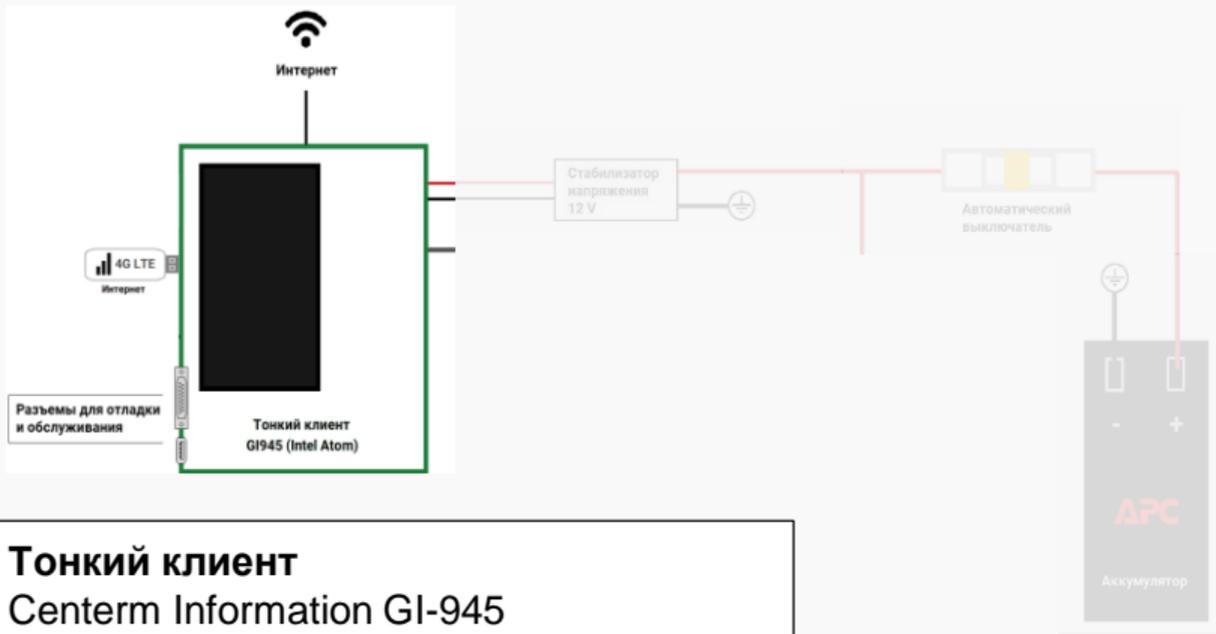




Стабилизатор напряжения

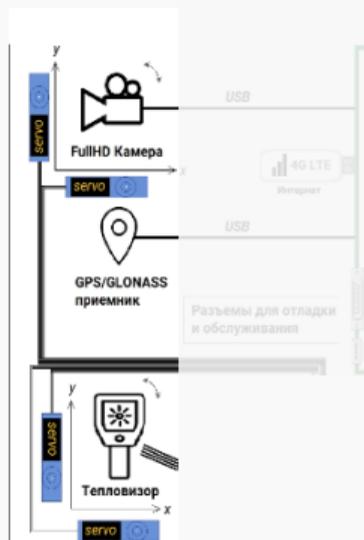
MEAN WELL SD-50A-12

МОЩНОСТЬЮ **50.4 Вт**



Тонкий клиент
Centerm Information GI-945

Процессор: **Intel Atom N270**
Оперативная память: **1 Гбайт**
Сеть: **Wi-Fi, 4G**



На подвижном подвесе

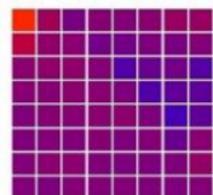
Видеокамеры

1. **Передняя**
1600x1200 px
(1.2 МПикс),
30 fps, H.264
2. **Задняя**
1280x1024 px,
30 fps

Тепловизор AMG8833

Разрешение 8x8 px

Позволяет различать **теплые и холодные** объекты, например **людей** или **возгорания**



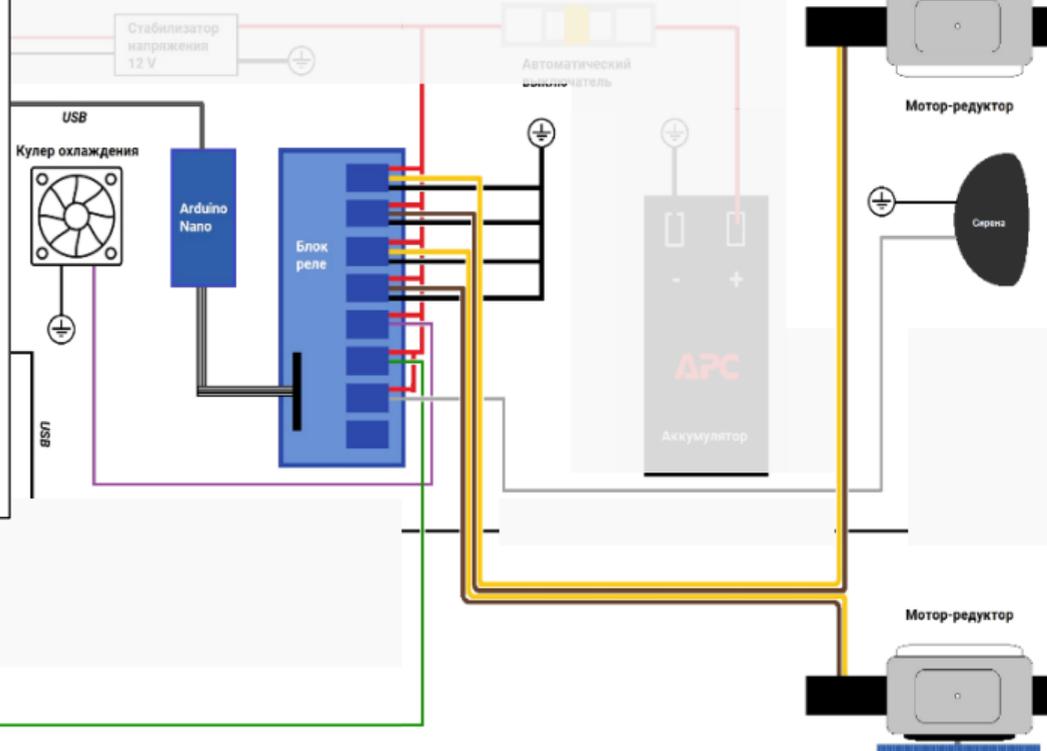
Управление оборудованием и тяговыми двигателями



Микроконтроллер
Arduino Nano (ATmega328P)

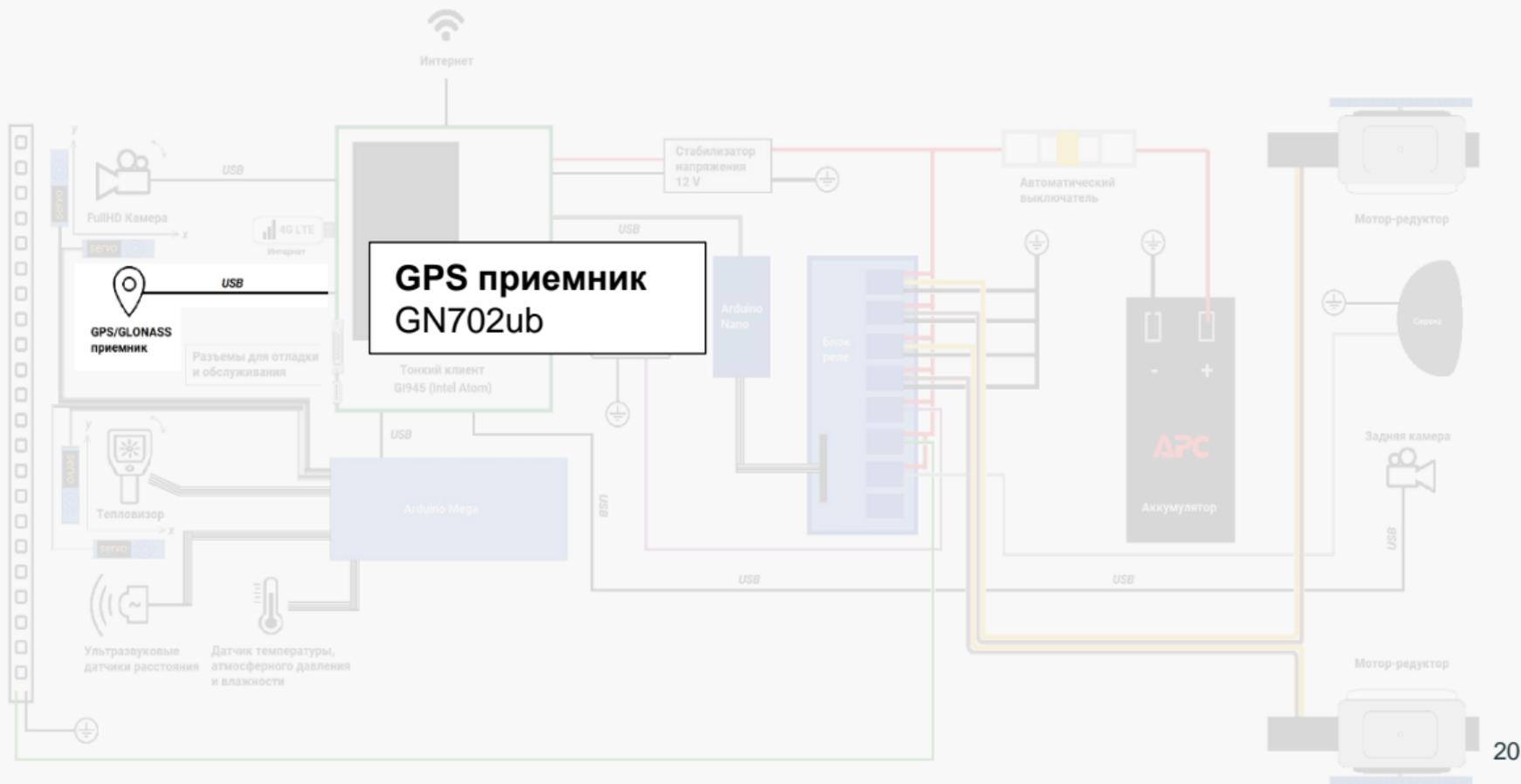
Через 8-канальный **блок реле** управляет мощным оборудованием:

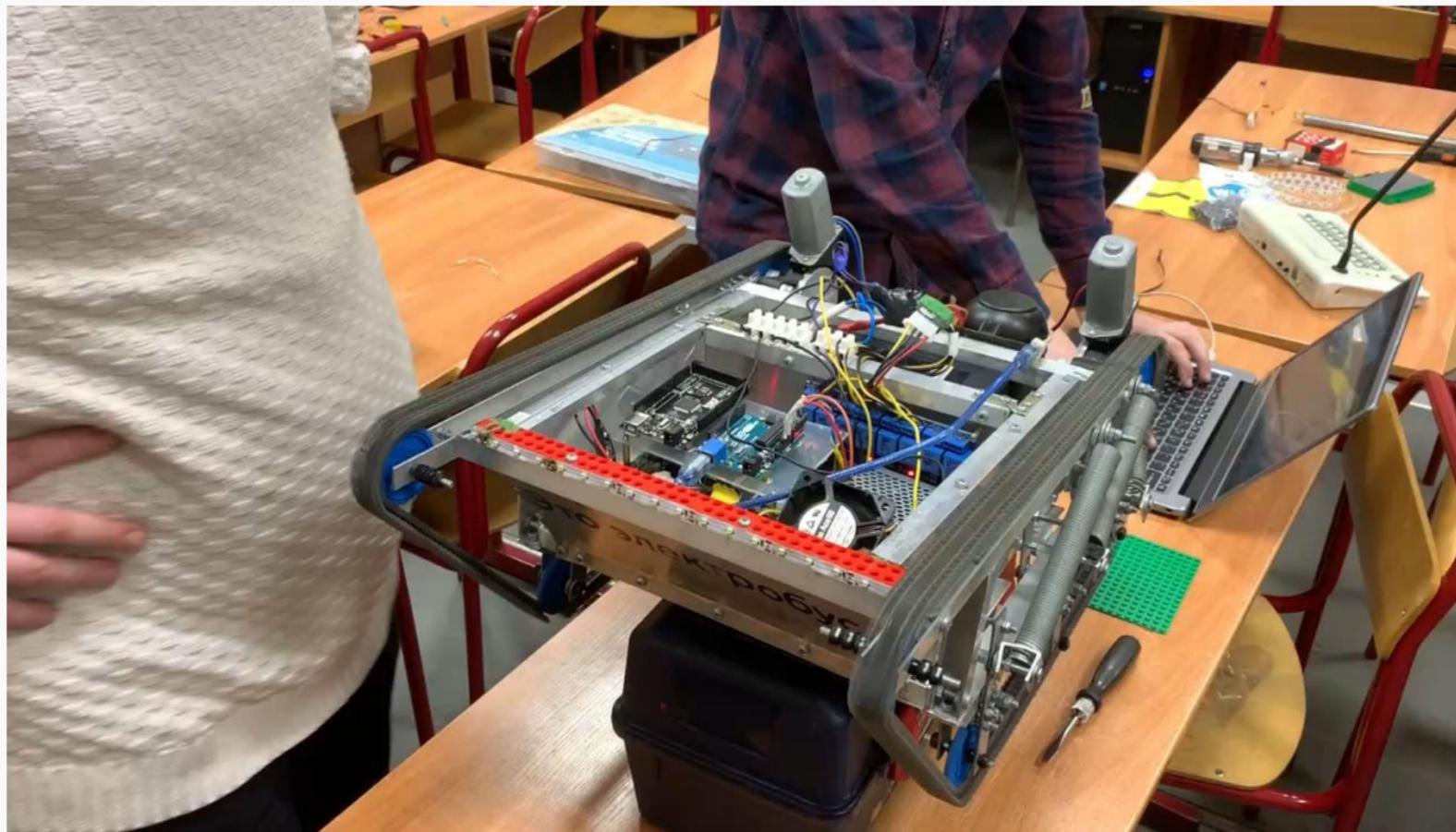
- фарой
- кулером охлаждения
- сиреной
- тяговыми двигателями



Светодиодная фара

Положение в пространстве



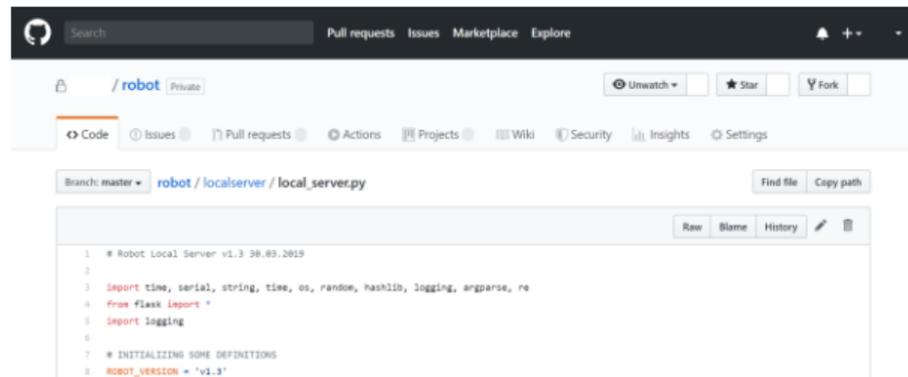


Программное обеспечение

Python-приложение на работе контролирует системы и оборудование, взаимодействует с оператором

Видео передается при помощи **Motion** либо **FFMpeg + Nginx**

```
ROBOT WSGIY SERVER v1.1 Alpha --
PASSWORD: bb7x69K5 --
Error: Could not connect to /dev/ttyOCH0 ? --
 * Serving Flask app "robot-console" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:9000/ (Press CTRL-C to quit)
132.168.1.195 - - [05/Dec/2018 20:10:56.1] "GET / HTTP/1.1" 401 -
132.168.1.195 - - [05/Dec/2018 20:11:16.1] "GET / HTTP/1.1" 200 -
132.168.1.195 - - [05/Dec/2018 20:11:16.1] "GET /bootstrap/static/css/bootstrap.min.css?footst
132.168.1.195 - - [05/Dec/2018 20:11:16.1] "GET /bootstrap/static/css/font-awesome-all.min.css?
132.168.1.195 - - [05/Dec/2018 20:11:17.1] "GET /bootstrap/static/js/jquery.min.js?bootstrap=4.0
132.168.1.195 - - [05/Dec/2018 20:11:17.1] "GET /bootstrap/static/js/popper.min.js?bootstrap
132.168.1.195 - - [05/Dec/2018 20:11:17.1] "GET /bootstrap/static/js/bootstrap.min.js?bootstrap
-- Cannot toggle lights! Please check connection.
132.168.1.195 - - [05/Dec/2018 20:11:21.1] "GET /control?token=1ngitw0ixk1tdrcc636ef13663
-- Cannot toggle lights! Please check connection.
132.168.1.195 - - [05/Dec/2018 20:11:22.1] "GET /control?token=1ngitw0ixk1tdrcc636ef13663
-- Cannot toggle lights! Please check connection.
132.168.1.195 - - [05/Dec/2018 20:11:23.1] "GET /control?token=1ngitw0ixk1tdrcc636ef13663
```



The screenshot shows a GitHub repository for a project named 'robot'. The file 'local_server.py' is open, showing the following code:

```
1 # Robot Local Server v1.3 30.03.2018
2
3 import time, serial, string, time, os, random, hashlib, logging, argparse, re
4 from flask import *
5 import logging
6
7 # INITIALIZING SOME DEFINITIONS
8 ROBOT_VERSION = 'v1.3'
```

Тонкий клиент работает на **Linux** (Ubuntu Server 16.04 LTS)

Веб-интерфейс работает на **Flask (Python)** через **uWSGI**, локальная база данных - **SQLite 3**

Робот может управляться через:

- **веб-интерфейс**
- API
- SSH-консоль

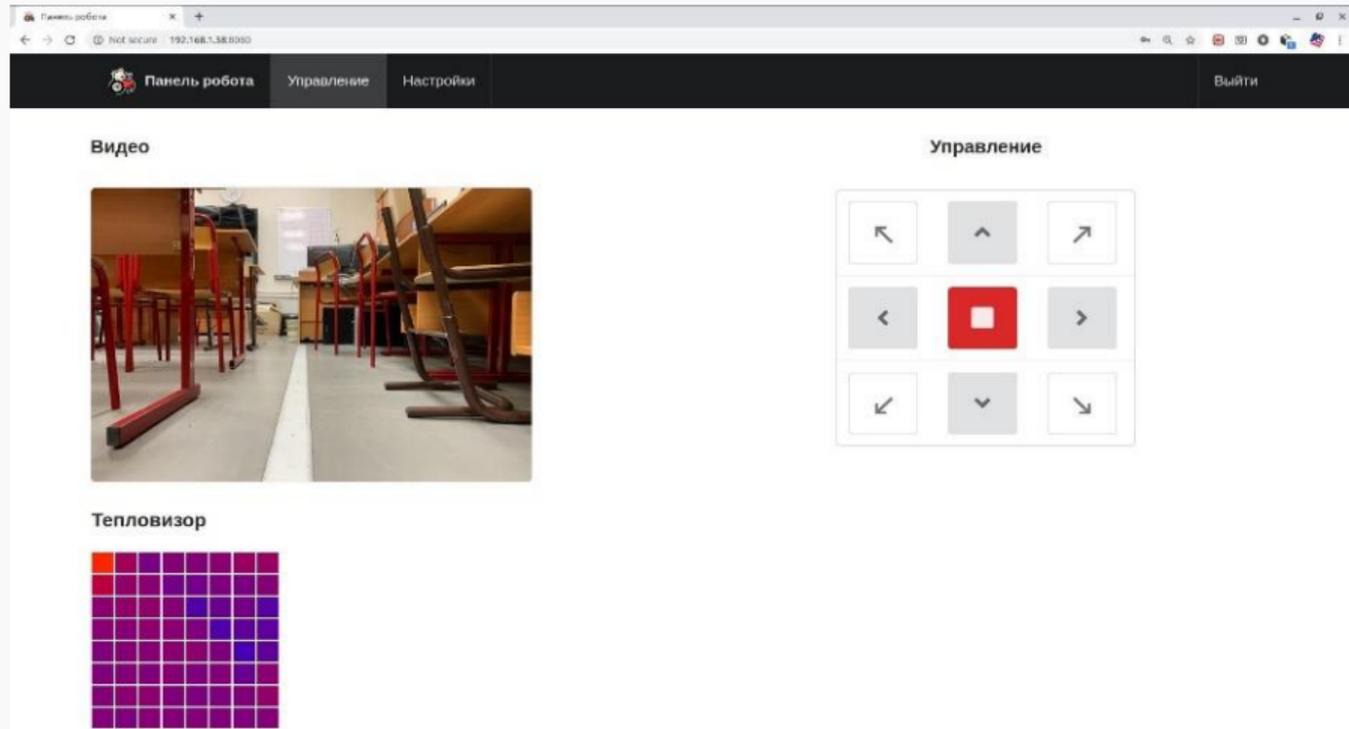
Связь:

- **Wi-Fi**
- **моб. интернет**
- радиоканал (*перспективно*)

Веб-интерфейс оператора



на смартфоне



на компьютере

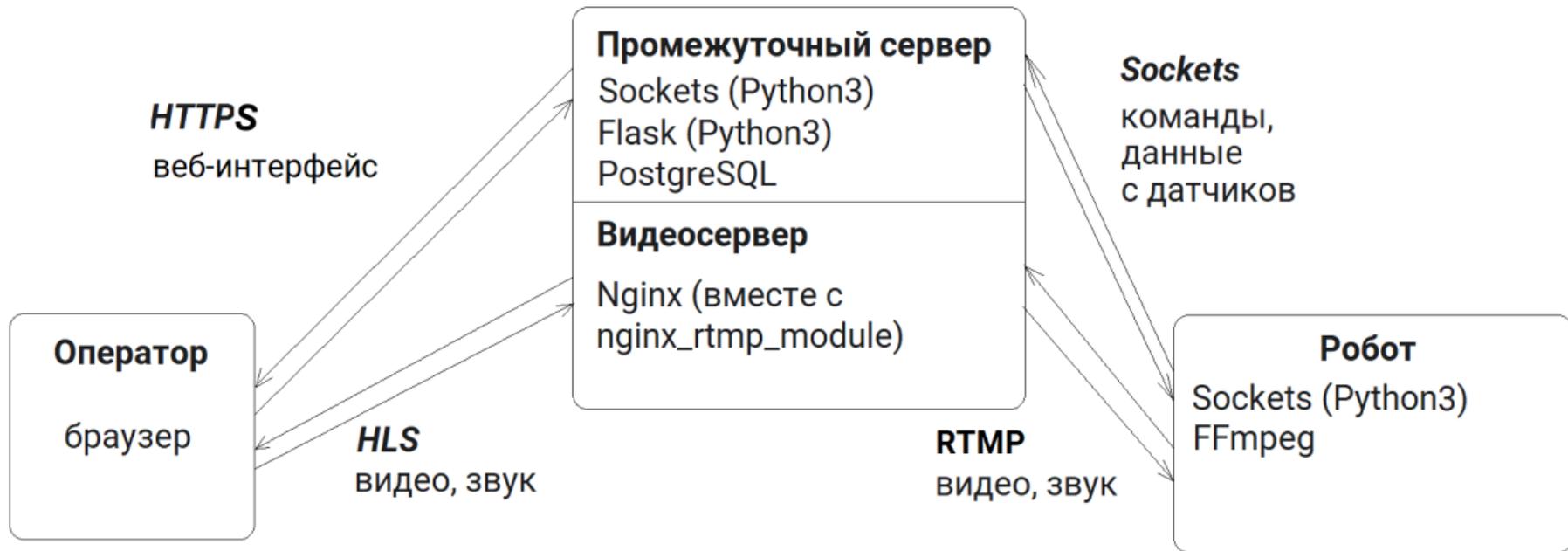
Схема работы программного обеспечения

в пределах локальной сети



Схема работы программного обеспечения

через интернет





На пересечённой местности

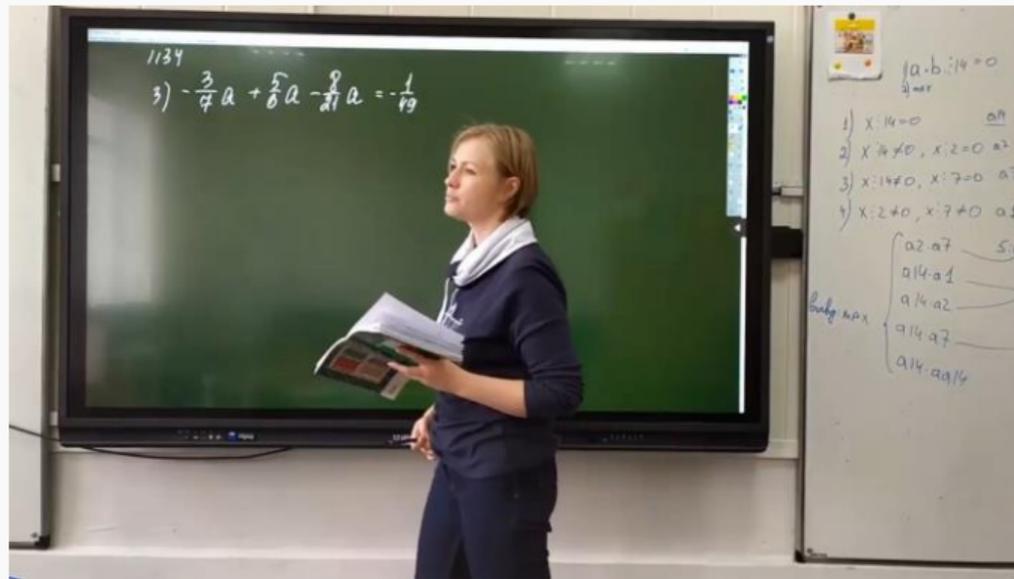
Результат

Мы создали свою универсальную платформу удаленного присутствия с большой **проходимостью** и **грузоподъемностью**



Дистанционное присутствие

создание эффекта
присутствия на уроке в школе
для заболевших учеников



Перспективы использования

Работа в **опасных зонах** без риска для человека



Исследование **труднодоступных мест**



Планы на будущее

- **Автоматизировать** работу робота
- Научить робота **распознавать объекты вокруг**
- Добавить бортовой **манипулятор**
- Добавить плавное **регулирование оборотов** тяговых двигателей
- Расширять функционал



Спасибо за внимание!



robotup.github.io/robot