

## **Содержание документа «Техническое описание робота»**

Состязание: **состязание роботов с техническим зрением памяти Виктора Ширшина.**

Название команды: **Таёжные Ёжики.**

Участники команды: **Цыганкова Мария, Пильщиков Григорий.**

Тренер команды: **Косаченко Сергей Викторович.**

Организация: **ОГБОУ Томский Физико-Технический Лицей.**

## **Основное содержание**

Аннотация: Цыганкова Мария и Пильщиков Григорий собрали шасси (из деталей лего и модифицированного колеса ball caster), на шасси сверху поместили ноутбук, который определяет цвет семафора с помощью технического зрения и посылает команды контроллеру шасси.

Роли участников команды: Цыганкова Мария и Пильщиков Григорий собирают шасси, электрическую схему и программируют робота вместе.

## **Опыт участия и успехи команды в робототехнических соревнованиях и фестивалях:**

На Всероссийских соревнованиях по подводной робототехнике 7-8 мая 2021 года в г.Владивосток участвовали Гетагазов Беслан и Пильщиков Григорий, заняли 4 место.

2020 - 2 место по робофутболу в Кубке Губернатора Томской Области 2020.

До 2022 года участником команды был Гетагазов Беслан, затем по уважительным причинам произошла замена, и в команду пригласили Цыганкову Марию.

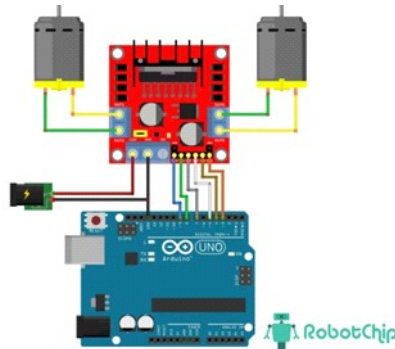
2022 год — Участие Цыганковой Марии и Пильщикова Григория в фестивале подводной робототехнике «АкваРобоФест» в г.Асино.

## **Описание робота**

Стратегия выполнения задания роботом: робот должен дождаться разрешающего сигнала семафора и после этого следовать по черной линии до финиша. Определять цвет и форму семафора мы решили с помощью технического зрения: если цвет семафора красный, то стоять на месте, и если цвет семафора зелёный начать движение по линии.

Ноутбук программируется на C/C++ с использованием библиотеки OpenCV, а контроллер шасси на C/C++ в Arduino IDE.

Электрическую схему для подключения двигателей к Arduino мы использовали с сайта RobotChip. Для подключения лего-моторов к драйверу с Arduino мы обжали провода специальными клещами и модифицировали стандартный разъем RJ-12 на 6 пинов, срезав защелку по центру (у Лего-разъемов она справа).



В ходе серии испытаний и тестирований робота на поле мы делали несколько модификаций и улучшений конструкции, пока не добились устойчивой и сбалансированной конструкции шасси, способной перевозить ноутбук.

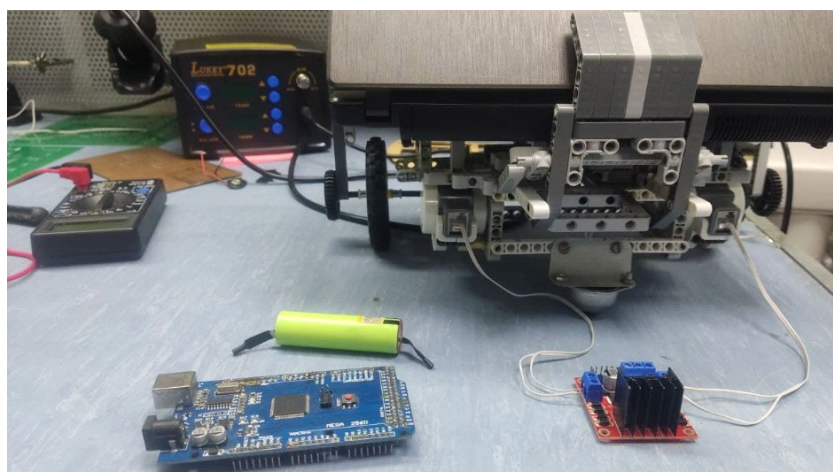
**Первая версия шасси:**



**Финальная версия шасси:**

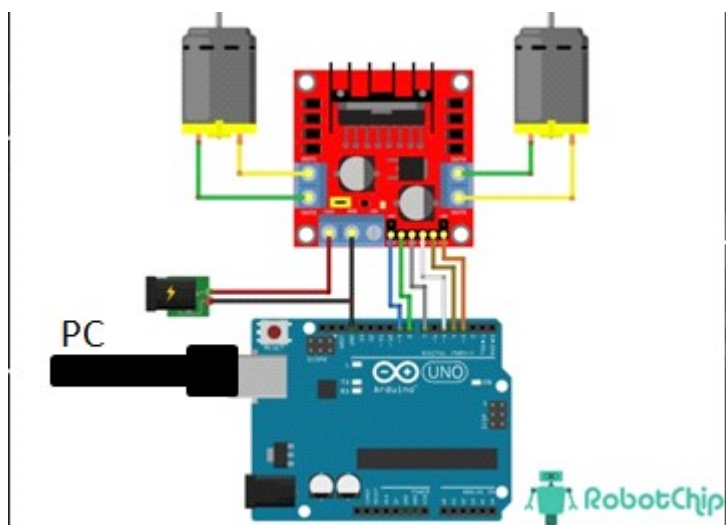


**Фото самодельных разъёмов для лего-моторов:**

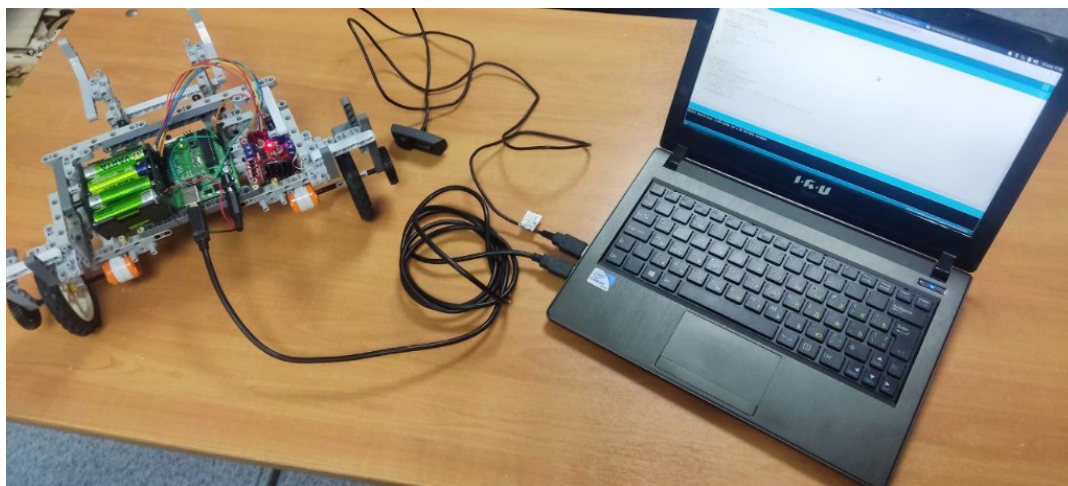




**Принципиальная электрическая схема шасси:**



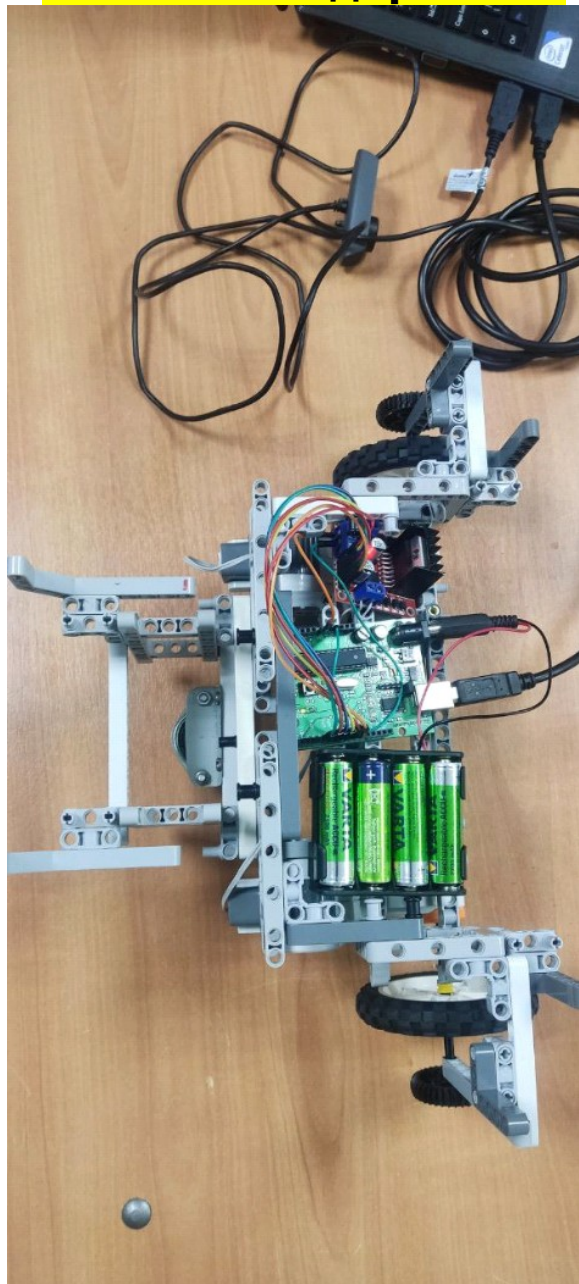
**Схема нашего робота в реальности:**

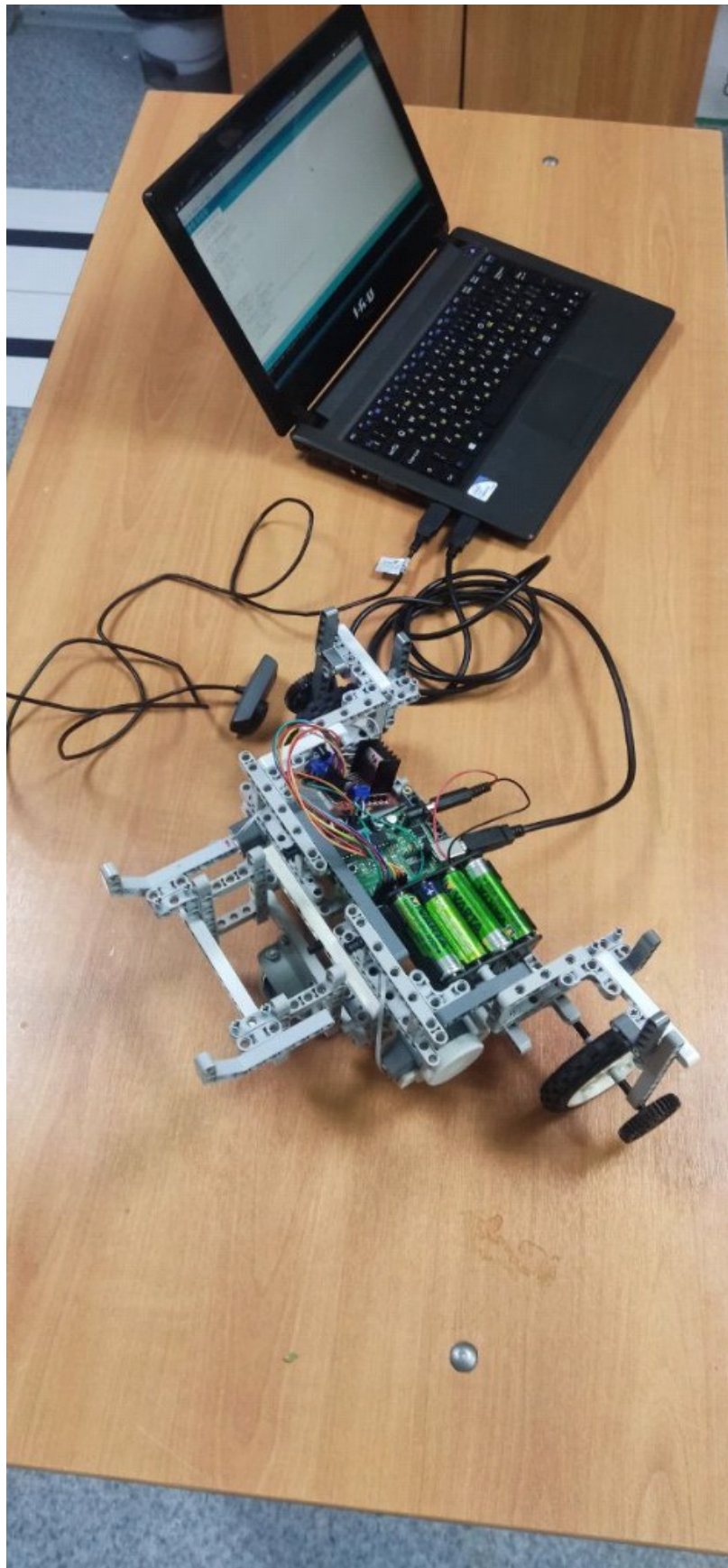




**Вид сверху**

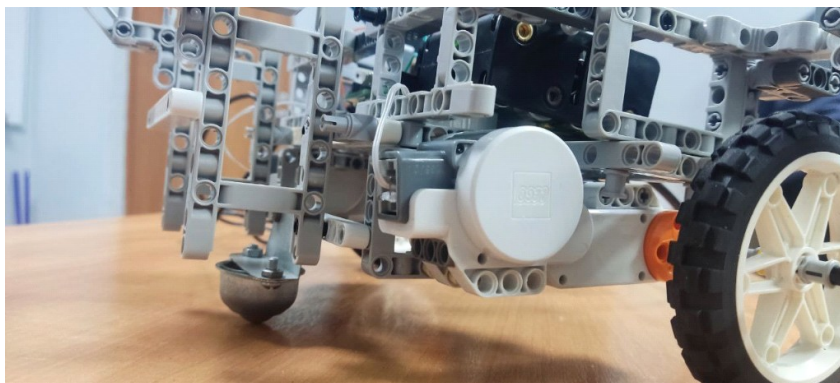
**Основное содержание**



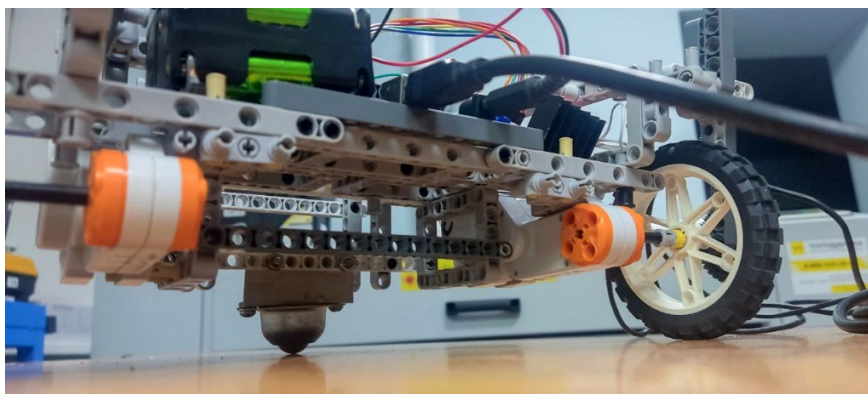
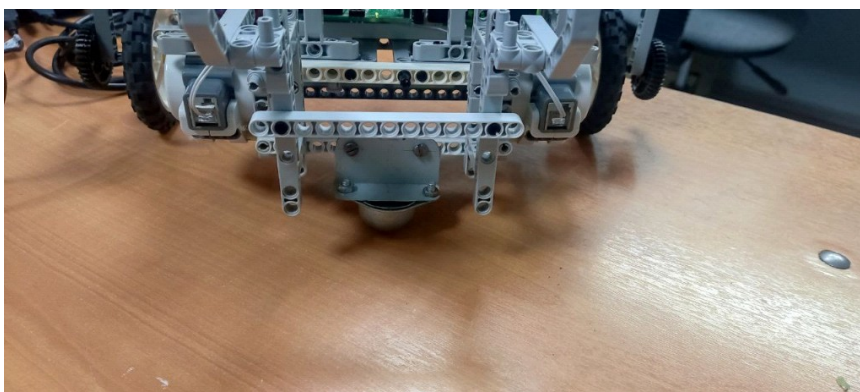


## **Кастомное колесо Ball Caster**

Так же мы применили в шасси кастомное всенаправленное колесо Ball Caster, сделали крепежный элемент из жести, совместимый по отверстиям с лего-балкой:



## **Демонстрация модифицированного колеса Ball Caster**





## Программное обеспечение

В качестве программного обеспечения мы использовали написанную самостоятельно программу на C/C++ с использованием библиотеки **OpenCV 2.4** на операционной системе **Xubuntu 20.04 LTS (Long Time Service)**



## Работа с веб-камерой и OpenCV

Мы взяли веб-камеру с частотой 30 кадров в секунду. Благодаря этой веб-камере, мы захватываем с нее кадры и бинаризируем изображение, выделяя зеленый цвет, и среди выделенных фигур мы ищем эллипс (зелёный).

Фрагмент программы, работающей с камерой на OpenCV 2.4

```
while (true) {  
    //cv::Mat image = mur.getCameraOneFrame();  
    cap >> image;  
    cv::imshow("Image", image);  
    cv::cvtColor(image, image, cv::COLOR_BGR2HSV);  
    cv::Scalar lower(hMin, sMin, vMin);  
    cv::Scalar upper(hMax, sMax, vMax);  
    cv::inRange(image, lower, upper, image);  
    cv::imshow("Bin", image);  
    char c = cv::waitKey(10);  
    if (c == 27) { // 192 - 192 - 192 - 192  
        break;  
    }  
}
```

## Бинаризация для определения зелёного семафора в кадре

Сначала мы создаём кадр и в нём ищем контур круга (эллипса). Если такой эллипс найден, то этот контур обводится зелёным

цветом, и робот начинает своё движение по линии.

Фрагмент кода на OpenCV 2.4

## Фрагмент кода, который ищет контуры:

```
for (auto contour : contours) {
    if (contour.size() < 6) continue;
    cv::Scalar color = cv::Scalar(0, 0, 255);

    auto ellipse = cv::fitEllipse(contour);
    auto width = ellipse.size.width;
    auto height = ellipse.size.height;

    auto majorAxisLength = std::max(width, height);
    auto minorAxisLength = std::min(width, height);

    auto eccentricity = std::sqrt(1.0 - std::pow(minorAxisLength / majorAxisLength,
2));
    if (eccentricity > (double)eccentricityThreshold / 100.0) continue;

    auto area = cv::contourArea(contour);
    if (area < areaThresholdInterpolated) continue;

    cv::ellipse(result, ellipse, color, 2);
}
```

## Распознавание отклонения чёрной линии от центра кадра

Распознавание отклонения чёрной линии от центра кадра происходит следующим образом. Передача команд управления для шасси по протоколу UART с использованием библиотеки MySerial

**Передача команд управления для шасси мы организовали следующим образом. Сначала контроллер пересылает значения скоростей моторов на плату Arduino, после чего она обрабатывает эти команды, находит само число из строки и управляет моторами через драйвер.**

## Программа управления для контроллера шасси

Ноутбук по камере находит отклонение центра робота от линии и стремится уменьшить отклонение от линии благодаря П-Регулятору. Следовательно, робот должен выравниваться и ехать по линии.

Программу управления для контроллера мы писали на Arduino IDE

### Фрагмент кода на Arduino IDE, который демонстрирует разбиение получаемой от ноутбука строки и отправку команд на драйвера

```
if (stringComplete) {
    char string = inputString.charAt(0);
    inputString.setCharAt(0, '0');

    if (string != 'A' && string != 'B') {
        string = inputString.charAt(1);
        inputString.setCharAt(1, '0');
    }

    Serial.println(string);
    Serial.println(inputString);
    int speed = inputString.toInt();
    Serial.println(speed);

    if(string == 'A'){
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
        analogWrite(9,speed);
    }
    if(string == 'B'){
        digitalWrite(7,HIGH);
        digitalWrite(6,LOW);
        analogWrite(10,speed);
    }
    speed = inputString.toInt();

    // clear the string:
    inputString = "";
    stringComplete = false;
    for (int i=0; i< string - '0';i++){
        digitalWrite(13, 1);
        delay(300);
        digitalWrite(13, 0);
    }
}
```

```
    delay(300);  
}  
  
}
```

## Обсуждение и заключение

Решение проблем: Наша команда столкнулась с проблемами.

**Первая проблема** - это программирование. Так как наша команда никогда не сталкивалась с техническим зрением, для нас было сложностью научиться программировать. Было много ошибок, которые нужно было исправлять. Порой это занимало очень много времени. При том это было самой долгой частью нашей подготовки к соревнованиям.

### Вторая проблема:

Участники команды на нетбуки впервые установили Xubuntu и OpenCV.

### Третья проблема:

Выбор контроллера. Нашей команде надо было выбрать контроллер. У нас были варианты, такие как Raspberry Pi, Orange Pi, NVIDIA Jetson и др. Но наша команда решила использовать нетбук, ибо регламент соревнований это позволяет. Так как по регламенту соревнований допустимый размер 40 см - по ширине и 40 см - по длине (высота не регламентирована), то нетбук отлично нам подошёл. Также вес робота с шасси менее 10 кг, что соответствует требованиям регламента соревнований. Также нетбук имеет большую вычислительную мощность, чем имеющаяся в данный момент Raspberry Pi 2 Model B. Также возникла проблема выбора либо Arduino, либо лего блок. Так как Arduino более эффективно связывается с нетбуком по UART, мы можем использовать любой блок питания, то нашей команде болуьше подходил вариант с Arduino. Также плата Arduino имеет более меньший вес, чем лего блок.

Чему научились члены команды:

Цыганкова Мария и Григорий Пильщиков познакомились и научились применять функции из OpenCV, захватывать изображение с веб-камеры, применили бинаризацию для обнаружения зеленого семафора и др.

Благодарности: Благодарим нашего тренера Косаченко Сергея Викторовича за то, что он проводил с нами время и направлял нас на правильные пути и решения. А также мы благодарим Томский Физико-Технический Лицей за предоставленные нам ноутбуки.

Список источников информации — сайт ТФТЛ.

## Ссылки



YouTube: [youtube.com/channel/UCNHBKTFS28L79RUSjuYjb3A](https://www.youtube.com/channel/UCNHBKTFS28L79RUSjuYjb3A)

GitHub: <https://github.com/Grin2020/TE2022>