

Team Description Paper

RoboCup Junior League: OnStage (Advance)

Team Name: Animal Electronics

Participants Name:

- Браулов Григорий, *braulov2004@gmail.com*
- Литвинова Мария, *masha15litvinova@yandex.ru*
- Хазанова Анастасия, *nastena.kazanova07@gmail.com*
- Чернова Маргарита, *margaritta95@icloud.com*

Mentors:

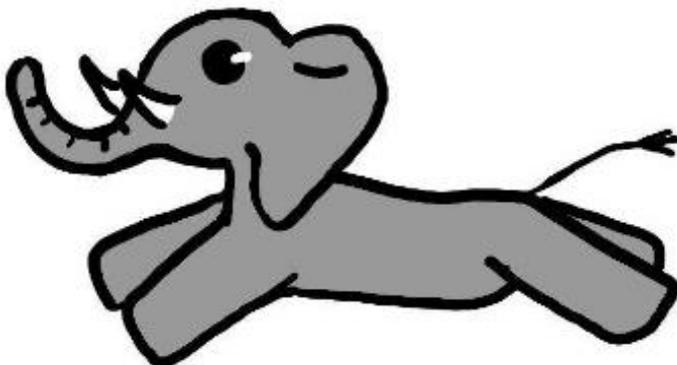
- Иванов Василий Леонидович, *kraftvergist@mail.ru*
- Казанцева Ольга Юрьевна, *kazantseva239@gmail.com*
- Коваленко Надежда Сергеевна, *nadaywolfani@mail.ru*

Institution: Президентский ФМЛ №239

Country: Россия

Date: 31.07.2020

Team logo:



Abstract

Проект представляет собой роботизированных животных, которые выступают в цирке, как настоящие.

Идея нашего представления – показать, что можно использовать роботов вместо животных.

В нашем проекте три робота: робот-слон, робот-панда и робот-обезьяна. Робот-слон – конструкция, спроектированная в САПР, напоминающая Boston Dynamic Spot, закрепленная на omni-платформе (рис. 1). Робот-панда представляет собой балансирующего робота на базе LEGO EV3 (рис. 2). Робот-обезьяна представляет собой POV-дисплей, закрепленный на платформе (рис. 3).

Для реализации проекта мы использовали такие технологии как: нейронные сети для распознавания голоса, компьютерное зрение для навигации и локализации на сцене и другие.

| <u>Team Background</u> | | | <u>RoboCup experience</u> |
|--|--|--|---|
| Name | Role | Experience | |
| Браулов Григорий, 10 класс, 16 лет | Программист нейронных сетей | Занятия робототехникой: 2 года Программирование: C++, Python. Навыки: программирование, анализ данных, пайка | - |
| Литвинова Мария, 9 класс, 14 лет | Конструктор робота-слона, программист компьютерного зрения | Занятия робототехникой: 4 года Роботы: на платформе Lego, для творческого проекта OnStage Программирование: C- подобные, Python. Навыки: конструирование, пайка, 3д-моделирование | RoboCup Russia Open 2019 (OnStage Novice): 1 место RoboCup 2019 Sydney (OnStage Novice): 3 место RoboCup Asia Pacific (OnStage Novice): 1 место Команда: Geese Swans |
| Хазанова Анастасия, 7 класс, 13 лет | Конструктор и программист балансирующего робота-панды, дизайнер | Занятия робототехникой: 3 года Роботы: на платформе Lego Программирование: C-подобный, Python. Навыки: конструирование | - |
| Чернова Маргарита, 9 класс, 15 лет | Капитан, конструктор и программист робота-обезьяны, составитель TDP, видеомонтажер | Занятия робототехникой: 4 года Роботы: на платформе Lego, для творческого проекта OnStage Программирование: C-подобные. Навыки: конструирование, пайка, 3д-моделирование | RoboCup Russia Open 2019 (OnStage Novice): 1 место RoboCup 2019 Sydney (OnStage Novice): 3 место RoboCup Asia Pacific (OnStage Novice): 1 место Команда: Geese Swans |

Introduction

- Team website: Inst: @animal_electronics,
http://robot239.ru/index/komanda_animal_electronics/0-112

○ Team/robot photo:



Собрание команды



Дистанционное собрание команды



Робот-слон (рис.1)



Робот-панда (рис.2)



Робот-обезьяна (рис.3)



Моделирование поворотного узла робота-слона



Моделирование крепления для колеса случайности



Моделирование лапы робота-слона



Моделирование туловища робота-слона



Печать лапы робота-слона



Пайка адресной LED ленты робота-обезьяны



Пайка проводов робота-слона



Пайка проводов робота-слона



Сборка туловища робота-слона



Сборка Omni-платформы



Сборка робота-слона



Сборка Omni-платформы



Сборка робота-слона



Сборка робота-обезьяны



Сборка робота-панды



Сборка колеса случайности



Программирование робота-панды



Программирование нейронных сетей



Программирование компьютерного зрения



Программирование нейронных сетей



Программирование робота-обезьяны

Team research

○ Strategy

Цель нашей лиги - показать выступление, где технологии станут искусством. Наша команда считает, что за эти 2 минуты нужно показать выступление, которое будет нести определенную мысль красочно и ярко, используя современные технологии.

Идея нашего представления на сцене – показать, что в цирке можно использовать роботов, вместо животных. Ведь, условия, в которых держат животных, не всегда благоприятны. Они живут за решетками, лишены свободы перемещения. Обезьяны не могут лазать по деревьям, львы и любые кошачьи не могут бегать. Это отнимает у животных их естественную красоту. Животные в цирке – это неизвестные несчастные актеры. Во многих странах выступления с животными в цирке запрещены. Поэтому наша команда решила представить инновационный цирк, где настоящие животные будут свободны, а люди смогут наслаждаться представлениями.

Сюжет закручивается вокруг дрессировщика, который устраивается на работу, но вместо настоящих животных у него роботы. Директор цирка сначала не верит, что роботы могут заменить животных, но дрессировщик вместе с роботом-слоном, роботом-пандой и роботом-обезьяной успешно доказывают это, выполняя трюки. В итоге, директор принимает на работу дрессировщика с его роботами.

Для реализации нашего выступления мы применяем разные технологии:

- Дрессировщик озвучивает команды голосом, робот-слон их «слышит» и выполняет необходимые трюки. Для распознавания голосовых команд мы используем обученную нейронную сеть.
- Робот-слон во время передвижения по сцене ориентируется на ARuco маркеры. Для навигации и позиционирования мы используем компьютерное зрение.
- Вместе с тем балансирующий робот-панда составляет компанию роботу-слону, кружась вокруг и искусно поддерживая равновесие за счет ПД и ПИ регуляторов в программе.
- А робот-обезьяна «обезьянничает» и «фокусничает»: повторяет все за роботом-слоном, изображая трюки с помощью быстро вращающейся адресной LED-ленты, которая благодаря эффекту персистенции создает у зрителей оптическую иллюзию.

Предлагаемая нами стратегия поведения роботов на сцене изображена в виде взаимодействия (схема 1). Взаимодействие *робот-человек* - ключевое в представлении. Робот-слон реагирует на голосовые команды человека, выполняя разные трюки. Животное

реагирует на сигналы датчика жестов. Теперь о взаимодействии *робот-робот* и *связи между роботами*. Слон двигается по сцене, выравниваясь относительно ARuco маркеров на обезьяне, а также они связаны по Bluetooth. Робот-панда двигается вокруг слона за счет показаний дальномера.

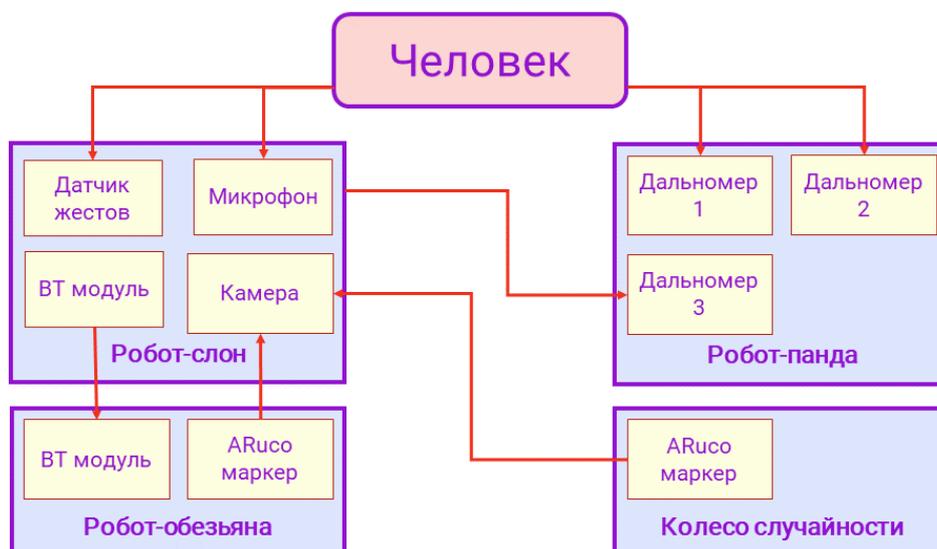


Схема взаимодействий роботов (схема 1)

○ Колесо случайности

Чтобы показать зрителям, что выбор голосовой команды, которую должен выполнить робот-слон, не был заранее predetermined, мы добавили колесо случайности (рис. 4). Это механическое устройство. Оно представляет собой конструкцию, состоящую из штатива, на котором горизонтально закреплен круглый фото-отражатель (рис. 5), на котором висит баннер с нарисованными пиктограммами команд (табл. 1).

| Пиктограмма | Голосовая команда | Краткое описание трюка |
|-------------|--------------------------|---|
| | «Пой» | Робот-слон исполняет известную мелодию, используя животные звуки |
| | «Сидеть, лежать, встать» | Робот-слон поочередно садится, ложится и встает |
| | «Задние лапы» | Робот-слон передвигается по сцене, стоя на двух задних лапах и ориентируясь на ARuco маркер |
| | «Прыжок» | Робот-слон разгоняется и поднимает поочередно лапы, создавая иллюзию прыжка через кольцо |
| | «Дай пять» | Робот-слон подъезжает к дрессировщику и выставляет лапы (чтобы дать «пять») |

Трюки робота-слона (табл. 1)

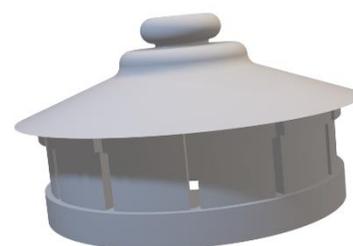
Директор цирка раскручивает колесо случайности несколько раз во время представления для того, чтобы выбрать очередной трюк, который должен выполнить робот-слон. Мы никогда не знаем, какая картинка «выпадет» после раскручивания. Таким образом невозможно предсказать, какой будет следующий трюк. Случайность выбора очередной команды позволяет продемонстрировать работу нашей нейронной сети во время представления. Вместе с тем при каждом перезапуске представления набор случайных голосовых команд получается другим, не дублирующим предыдущий, а значит нейросеть должна будет распознать каждый раз различные не predetermined команды.



Колесо случайности (рис.4)



Механизм крепления фото-отражателя к штативу (рис.5)



3д модель крепления

○ Робот-слон

● Hardware

Робот-слон – ключевой персонаж нашего циркового шоу (рис.1). Он «слышит» и «понимает» дрессировщика, выполняя всевозможные трюки: танцевать, исполнять мелодию, встать на задние лапы, «полет» и др.

Основой для внешнего облика робота послужила мягкая игрушка слон, купленная в магазине ИКЕА (рис. 6). Мы убрали из нее наполнитель и вставили «скелет», сделанный из напечатанных деталей и электронных компонентов (рис. 7).

Мягкая игрушка слон
(рис. 6)Процесс наполнения
игрушки

Процесс наполнения игрушки

Скелет робота-слона в
«костюме» (рис. 7)

Конструкция робота состоит из туловища, отпi-платформы, подъемного и поворотного механизмов.

- *Туловище робота-слона*

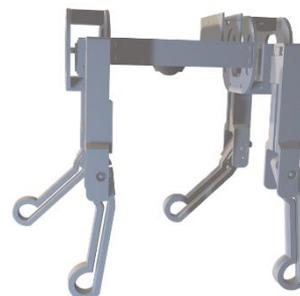
Спроектировано в САПР Autodesk Inventor (рис. 10), изготовлено в основном из пластика (рис. 9), присутствуют детали из фанеры и оргстекла (рис. 12). Нас вдохновила конструкция робота Boston Dynamics Spot (рис. 8), что побудило создать нечто похожее (рис. 13). Каждая лапа слона имеет три степени свободы, реализованные с помощью трех сервомоторов (рис.11).



Boston Dynamics Spot (рис. 8)



Печать 3D-деталей (рис. 9)



3д модель туловища робота-слона (рис. 10)



3D-модель лапы робота-слона (рис. 11)



Резка фанеры на лазерном станке (рис. 12)



Туловище робота слона (рис.13)

Запрограммировать шагающего на четырех опорах робота – это очень сложная и интересная задача для робототехников, над решением которой работают такие серьезные компании, как Boston Dynamics. Мы тоже хотели бы попробовать свои силы в этом, и это наша цель на будущее. А сейчас для реализации задумки нашего шоу нам достаточно имитировать ходьбу робота-слона. Поэтому для перемещения робота по сцене мы используем omni-платформу. Туловище робота-слона закреплено на ней с помощью направляющей, поворотного узла и подъемного механизма.

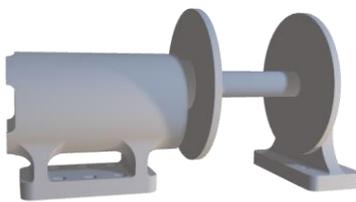
- *Подъемный механизм и поворотный узел*

Подъемный механизм нужен для того, чтобы робот мог выполнять такие трюки, как: встать на задние лапы, прыжки, «полет». Он состоит из мотора, катушки, каната и блока (рис. 14; 16). На катушку наматывается канат при помощи мотора, канат проходит внутри направляющей, проходит через блок и заканчивается внутри животного, в поворотном узле (схема 2). Для того, чтобы садиться/вставать на лапы, «летать», изменять положение туловища, внутри робота есть поворотный узел (рис. 15). Это шаровое шарнирное соединение, через которое проходит направляющая, закрепленная внутри детали так, что он свободно вращается и способен отклонять туловище от исходного положения почти на 75 градусов в любом направлении (схема

2). Шар и основная деталь, в которую он вставлен, напечатаны на 3D-принтере. Поворотный узел закреплен на внутренней пластине, внутри тела робота. Внутри шара закреплен трос, отвечающий за подъем робота относительно пола.



3D-модель поворотного узла



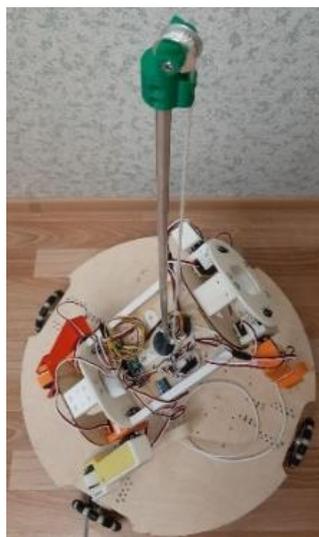
3D- модель катушки и держателя мотора
(рис. 14)



3D-модель блока



Поворотный узел в сборе (рис. 15)



Блок (рис. 16)

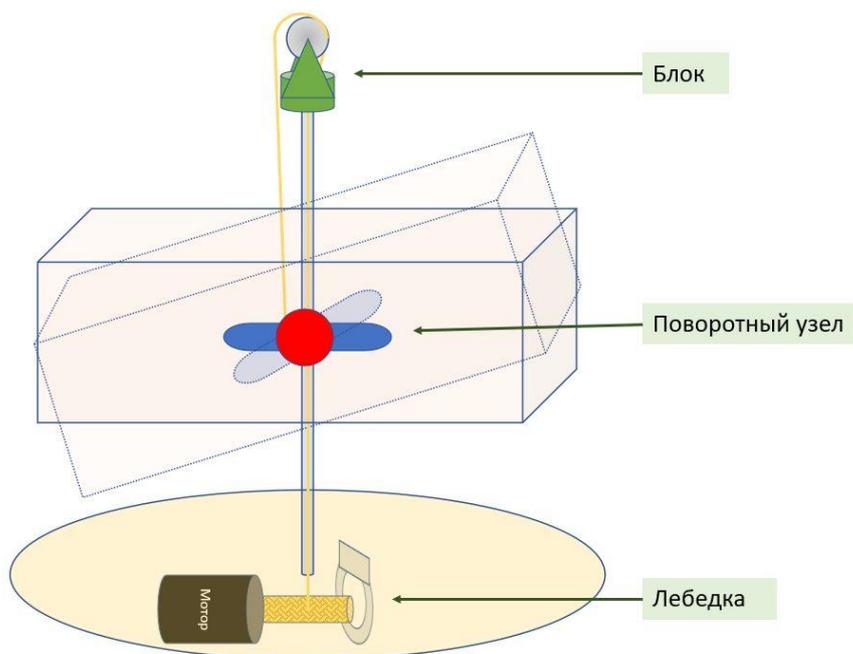
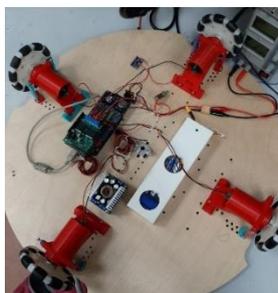


Схема поворотного узла и подъемного механизма (схема 2)

▪ *Оmnі-платформа*

Основа робота представляет собой omni-платформу, на которой закреплена основная электроника робота и 4 omni-колеса. Выполнена из фанеры округлой формы (рис.17).



Оmnі-платформа



Зд модели платформы и ее частей



Оmnі-платформа (рис. 17)



Процесс сборки omni-платформы



▪ *Оборудование:*

- ✓ Arduino Mega 2560
- ✓ 2 гироскопа MPU9250
- ✓ 4 мотора без энкодера, 1 с энкодером 120:1 12V
- ✓ 2 стабилизатора 5В
- ✓ 3 драйвера Dual VNH5019 DC Motor Driver Shield

- ✓ Mega Sensor Shield
- ✓ Аккумулятор Li-Po 3000
- ✓ 3 тумблера
- ✓ 4 omni-колеса 4" wheel TETRIX
- ✓ Камера Open MV H7
- ✓ NVIDIA Jetson Nano
- ✓ Сервоконтроллер PCA9685
- ✓ 12 сервомоторов DS3115 mg
- ✓ Микрофон ReSpeaker 2 Mics Pi HAT
- ✓ Динамик
- ✓ Датчик жестов APDS-9960

- Software

Программное обеспечение слона состоит из 3-х модулей. Основная управляющая программа находится на Arduino Mega. Она управляет omni-платформой и лапами робота. Программа написана на C-подобном языке в Arduino IDE. Программа распознавания голосовых команд выполняется на Jetson Nano. Написана на Python 3.7.3 в Visual Studio. Программа распознавания меток запускается на OpenMV. Написана на Python в OpenMV IDE.

- *Управление конечностями робота-слона*

Для позиционирования серво-ног робота мы также используем декартову систему. Для начала задаются точки или окружность по которым должны пройти все конечности, и промежуток времени, за который они должны это сделать. Затем программа каждую итерацию рассчитывает необходимую текущую позицию для каждого сервомотора в текущий момент времени и передаёт эти данные с помощью сервоконтроллера PCA 9685.

- *Позиционирование и навигация*

Мы используем камеру OpenMV H7 для распознавания маркеров, расположенных на сцене. Камера запрограммирована в OpenMV IDE. В бесконечном цикле камера ищет массив, элементами которого являются маркеры, которые она распознает с помощью встроенной функции `find_apriltags()`. Она возвращает несколько значений: вид маркера (семейство), координаты центра, координаты левого верхнего и правого нижнего угла, id маркера. Значения id маркера, его отклонение по 3 осям посылаются на Arduino Mega в формате строки `":id/x_отклонение/y_отклонение/z_отклонение;"`. Символ «:» свидетельствует о начале передачи данных, символ «;» – о конце. Программа для Arduino Mega написана в среде Arduino IDE. Здесь представлен цикл поиска маркеров: распознавание происходит по 1 символу, начинается с : и заканчивается ;, после каждого символа / переменная val обнуляется, а число, которое в ней

содержалось записывается в массив arr. В конце 4 переданных числа выводятся на консоль для проверки корректности.

■ *Нейронная сеть: распознавание голосовых команд*

✓ Постановка задачи

Дрессировщик произносит команду, а робот ее выполняет. Рассмотрим задачу детальнее. В качестве входных данных будем рассматривать аудиозапись с одной командой. Затем эта запись обрабатывается, и мы получаем индекс команды от 0 до 5 в качестве выходных параметров. Далее робот по полученному индексу выполняет команду. Мы получили задачу классификации.

✓ Данные

Для решения задачи был создан собственный набор данных. Были записаны звуковые дорожки длительностью 2 секунды каждая. На них дрессировщики произносят все команды, которые должен выполнить робот.

В итоге получилось 240 аудиозаписей в формате wav.

✓ Архитектура сети

Для решения задачи классификации при обработке сигналов используют нейронные сети, так как входные данные каждый раз разные и не существует точного алгоритма. Мы рассмотрели несколько архитектур и выбрали лучшую для решения задачи модель, а именно выполнили построение моделей и проверили их точность при определении команды. Были созданы следующие модели: собственная, resnet, alexnet. Программирование моделей выполнено с использованием библиотеки tensorflow.

Архитектура собственной модели:

1 слой - Conv1d (свертка - 9 на 9, кол-во нейронов - 64)

2 слой - Conv1d (свертка - 3 на 3, кол-во нейронов - 64)

3 слой – Dropout (0.2)

4 слой - MaxPool1d (размер пула - 3)

5 слой – Flatten ()

6 слой - FC (100 нейронов)

7 слой - FC (6 нейронов)

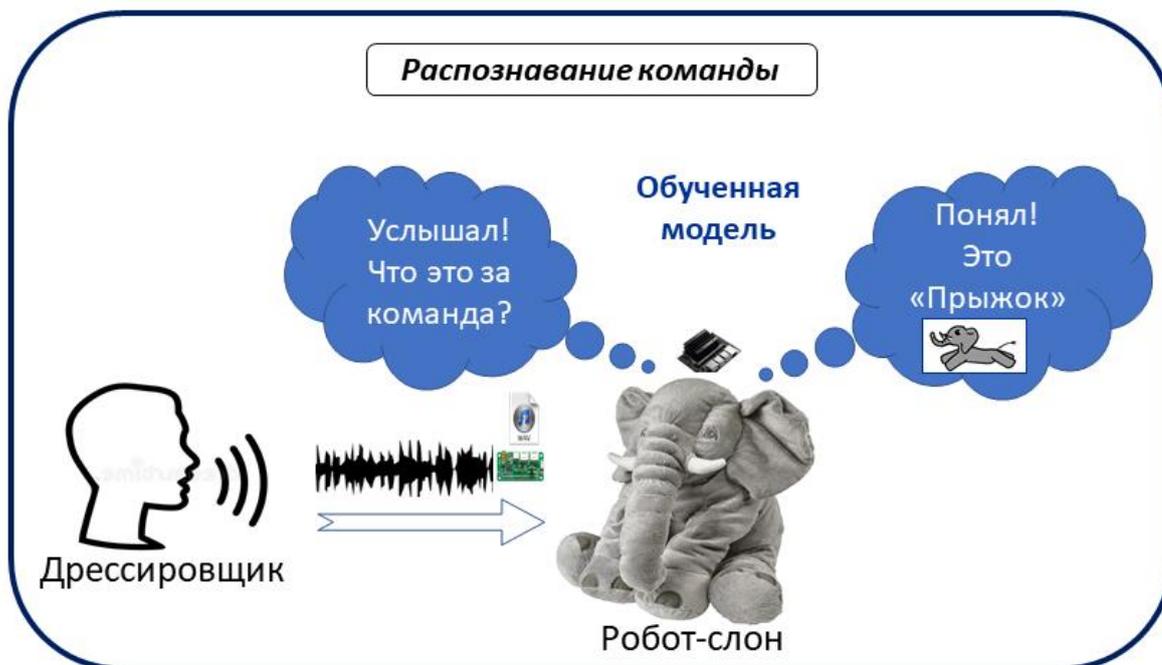
При проверке точности все модели получили меньше 40% точности. То есть из 10 команд меньше 4 определились бы корректно. Наша собственная архитектура дала результат 90%. Поэтому мы ее и выбрали как окончательную реализацию. Именно эта модель используется для роботов (табл. 2).

| Архитектура | Наша собственная | AlexNet | RezNet-50 |
|--------------------|-------------------------|----------------|------------------|
| Точность | ~90% | <40% | <40% |

Сравнение архитектур (табл. 2)

✓ Применение модели на практике

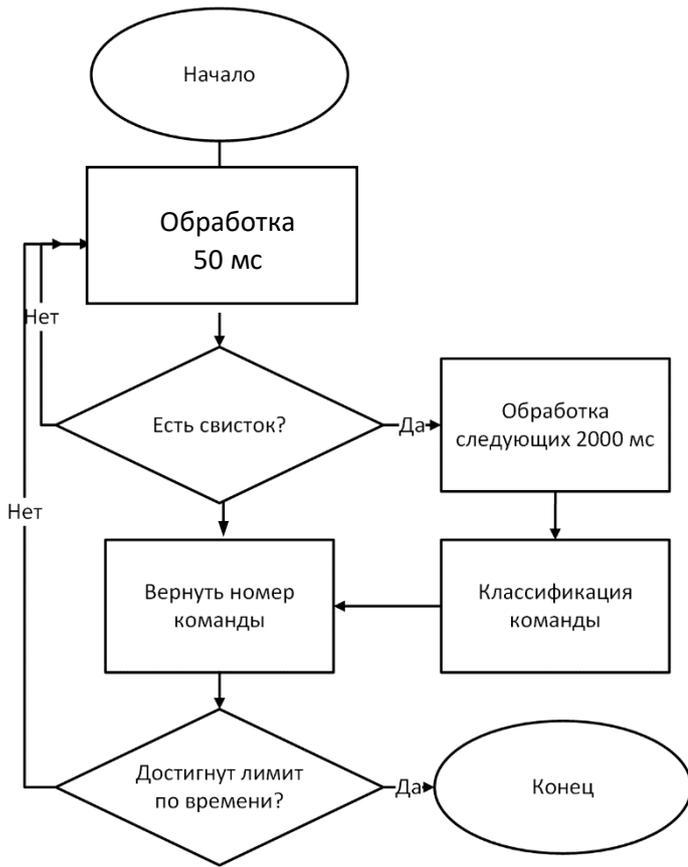
Обучение модели происходит до выступления, потому что оно может занимать очень много времени, а именно больше 4 часов. После обучения сохраняем веса модели и ее архитектуру, чтобы потом во время выступления один раз загрузить модель и использовать ее для распознавания команд (рис. 20).



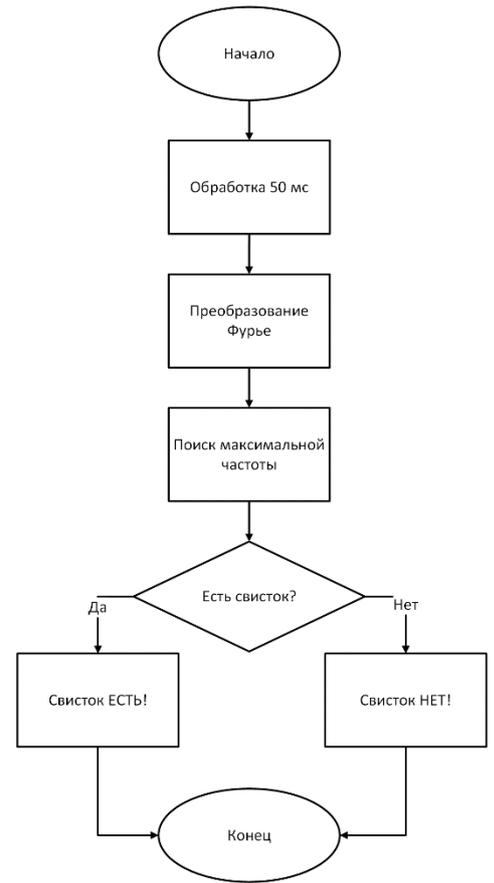
Распознавание команд (рис. 20)

Алгоритм распознавания представленный (схема 3;4): для начала берем первые 50мили секунд и определяем есть на этой части записи свисток или нет, если нет, то обрабатываем следующие 50 мили секунд. Если есть, то после свистка должна следовать команда. Поэтому берем следующие 2000мили секунд и применяем обученную модель для классификации, после этого получаем номер команды и отправляем его роботу, потом проверяем достигнут ли лимит по времени. В зрительном зале присутствует отвлекающий робота шум, из-за которого он может распознавать не только команды дрессировщика, но и шум. Чтобы избежать такой ситуации мы используем свисток на определенной частоте, робот распознает команды только после того, как распознал сигнал, производимый свистком.

Поиск свистка был сделан с помощью дерева решений. Применили преобразование Фурье, затем находим максимальное значение и оцениваем его. Если максимальное значение лежит в диапазоне частоты свистка, значит на этой части записи есть свисток и нам нужно выполнить дальнейшие действия алгоритма.



Блок-схема алгоритма распознавания команды (схема 3)



Блок-схема распознавания свистка (схема 4)

○ Робот-панда

● Hardware

Робот-панда – компаньон робота-слона. Она мастерски балансирует, кружась вокруг различных объектов, следуя за дрессировщиком и реагируя на его жесты. Представляет собой балансирующего робота на базе конструктора Lego Mindstorms Ev3. Плюшевая панда мягкая игрушка из ИКЕА (рис. 22) с прочным каркасом внутри (рис.21).



Робот-панда (рис.21)



Мягкая игрушка панда (рис. 22)

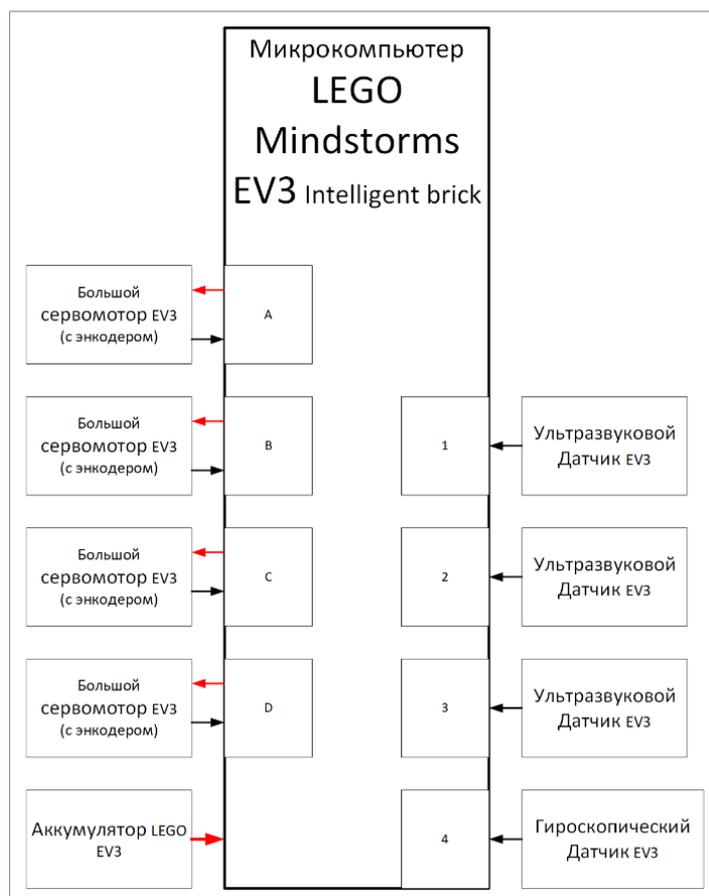


Механизм крепления колес



Процесс работы с роботом-пандой

Функциональная схема работа-панды

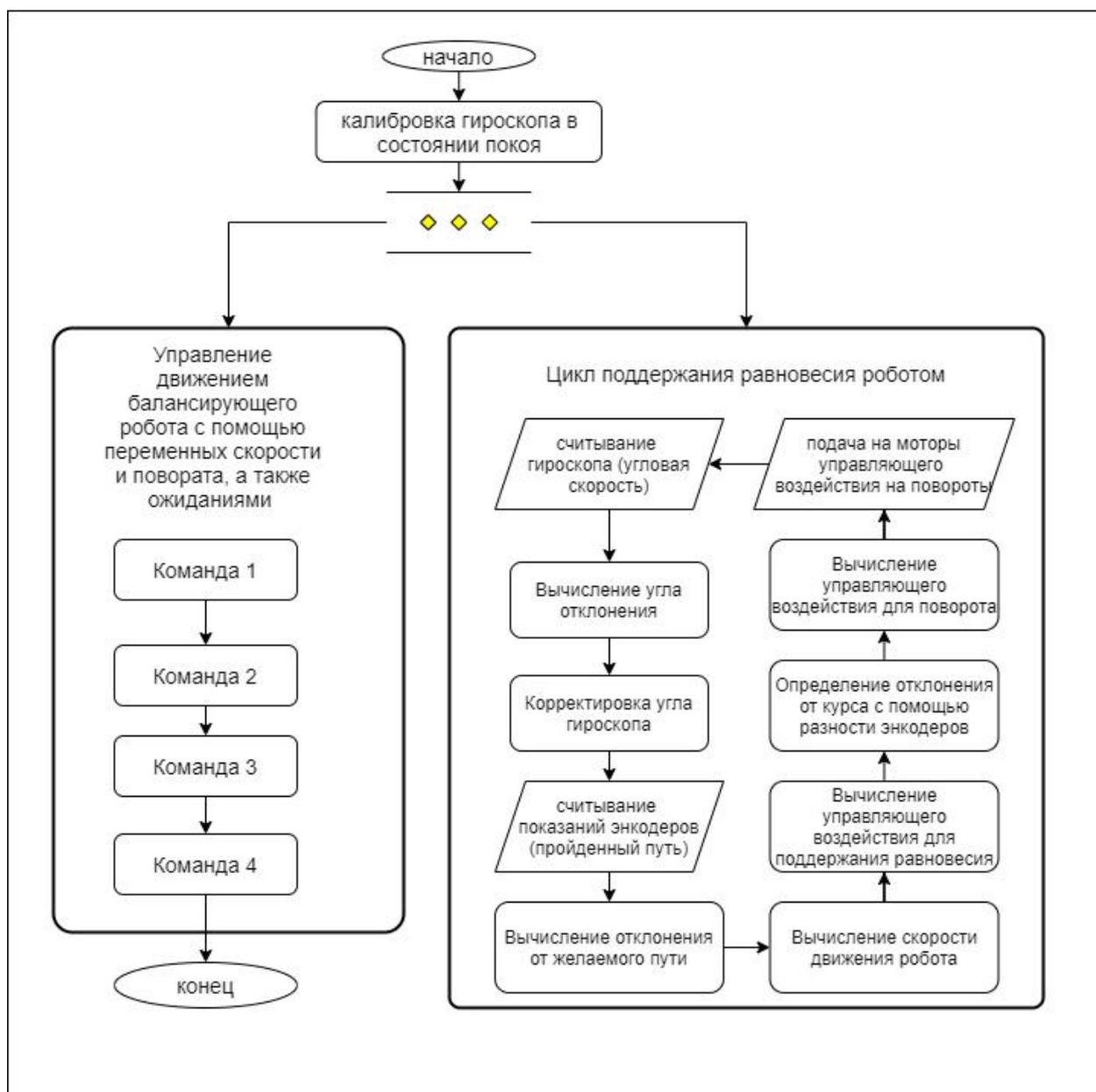


■ Оборудование:

- ✓ Микрокомпьютер LEGO Mindstorms Ev3 Intelligent Brick
- ✓ Аккумулятор LEGO EV3
- ✓ 4 больших сервомотора Ev3 (с энкодорами)
- ✓ Гироскопический датчик EV3
- ✓ 3 ультразвуковых датчика Ev3
- ✓ 2 колеса 4" Wheel TETRIX

● Software

Программа для работа-панды написана на языке RobotC. В программе используется пропорционально-интегральный регулятор для гироскопа для скорости изменения угла относительно калибровочного значения и пропорционально-дифференциальный регулятор для энкодеров относительно ориентира, изменяемого значения места нахождения робота. Блок-схема алгоритма балансирования представлена на схеме 5.



Блок-схема программы робота-панды (схема 5)

○ Робот-обезьяна

● Hardware

Робот-обезьяна представляет собой платформу, на которой закреплена мягкая игрушка, купленная в IKEA (рис. 23), на которой закреплён POV дисплей. POV дисплей – это быстро вращающаяся светодиодная лента. POV (Persistence Of Vision)-эффект персистенции. Эффект основан на возможности нашего мозга и глаз соединять в одно изображение быстро меняющиеся (движущиеся или мерцающие) картинки (рис. 24). Пикселизация изображения и вывод его реализован с помощью разных цветов светодиодов на ленте. Наш дисплей собран

на адресной ленте WS2812 с разрешением в 0,5 градусов. На этом дисплее можно воспроизводить различные картинки и анимации в формате gif (рис. 25).



Мягкая игрушка обезьяна (рис. 23)

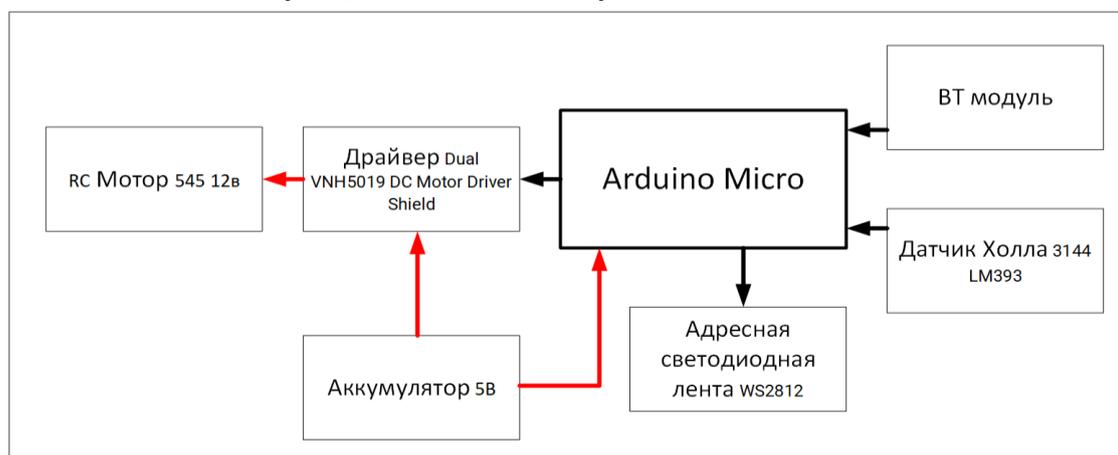


POV эффект



Робот-обезьяна (рис. 25)

Функциональная схема робота-обезьяны



▪ Оборудование:

- ✓ Arduino Micro
- ✓ адресная лента WS2812
- ✓ Модуль датчика Холла 3144 LM393
- ✓ Драйвер Dual VNH5019 DC Motor Driver Shield
- ✓ RC Мотор 545 12в
- ✓ блок питания
- ✓ контактные кольца со сквозным отверстием
- ✓ BT модуль

• Software

Программа написана на языке C в среде Arduino IDE. Она преобразовывает картинку в кодовые значения и подает их на адресную ленту. Программа находится в разработке.

Discussion and Conclusion

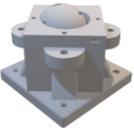
○ Share your team's learning experience :

- Проблемы, возникшие при подготовке и их решения.

В процессе работы над проектом у нас возникало много текущих задач, от решения которых зависел общий результат. Мы придерживались поэтапной разработки каждой проблемы от первоначальной идеи, ее реализации, тестирования и проверки на практике, внесения необходимых модификаций, на основе проведенных испытаний (см. схему 6)



Цикл разработки (схема 6)

| Первоначальное решение | Результаты тестирования | Недостатки → Требования для модификации | Эффективное решение |
|--|---|---|---|
| Задача 1: Реализовать балансирование робота-панды с большими габаритами (высота 47 см) | | | |
| Легкие материалы для каркаса панды, леги-колёса 92 мм в диаметре, на каждое колесо по 2 леги-мотора. Настроить ПИ и ПД регуляторы для энкодеров. | 10 попыток запуска робота показали, что панда может балансировать в течение 1 мин.  | Недостаточная устойчивость при балансировании → Улучшить настройку коэффициентов для регуляторов. Более широкие устойчивые колёса и крепление панды. | Замена колёс на 4" TETRIX wheels. Самодельное крепление. Под коэффициенты регуляторов для энкодеров → Панда стабильно балансирует 5+ мин, выполняет фигуры цветов, объезд по кругу. |
| Задача 2: Выбрать архитектуру нейронной сети для распознавания голосовых команд | | | |
| Архитектуры: resnet, alexnet | Точность моделей: 40% 40% | Недостаточная точность → Подобрать более точную модель нейронной сети | Архитектура собственной модели (см. с.11) Точность модели: 90% |
| Задача 3: Закрепить тело робота-слона на направляющей, чтобы робот мог выполнять различные трюки | | | |
| Модель шарового соединения  | Угол отклонения примерно 30 градусов  | Слишком маленький угол наклона туловища робота, нет возможности выполнять трюки. → уменьшить объем шара и крепления, изменить форму крепления | Уменьшен объем шара и крепления, изменена форма крепления → Угол отклонения 70 градусов,  тело свободно поворачивается |
| Задача 4: Организация дистанционной работы команды во время карантина | | | |
| Созвоны в discord. Ведение гугл-таблицы с фиксацией планов и результатов. Общий чат в Телеграм с фото и видео работы частей проекта. | Картина через месяц: редкие созвоны, не каждый член команды знает, чем занимаются другие. Гугл-таблица заполнена на 50%. Telegram наиболее активно используется для обмена фото/видео/сообщениями | Нет представления о продвижении проекта в целом и не хватает быстрого реагирования на возникающие проблемы у других участников. / Необходимо перестроить способ дистанционного общения команды. | В дополнение к почти ежедневному чату в Telegram мы стали созваниваться в zoom 2 раза в неделю и демонстрировать роботов прямо в конференции. → Все участники команды активно включены в общую работу команды. |

Процесс решения проблем (таблица 3)

- Чему мы научились в ходе проекта?

| Участник | Научились в ходе проекта | Улучшили навыки в ходе проекта | Самое значительное достижение в проекте |
|----------|--|--|---|
| Гриша | Программирование нейронных сетей | Математика (дискретная математика, математический анализ), алгоритмы | Относительно хорошая точность созданной модели |
| Марго | Работа с микроконтроллерами Arduino и программирование в Arduino IDE | 3D-моделирование, работа в программах office, электротехника | TDP, организация технической демонстрации |
| Маша | Программирование на языке Python в Open MV IDE, Работа с микроконтроллерами Arduino и программирование в Arduino IDE | 3D-моделирование | Создание большинства 3д моделей робота-слона и его сборка |
| Настя | Работа в Photoshop | Конструирование и программирование балансирующих роботов | Создание робота-панды |

- **Highlight collaboration with other teams if any:** В этом году мы, к сожалению, не работали с другими командами. Но мы всегда рады сотрудничеству, а также открыто делимся своими знаниями. Если вы хотите узнать что-то подробнее:
 - https://drive.google.com/drive/folders/1-8INGwnzY4QnoX0PwCaRG9Nyo_xapKai - папка на гугл диске, в которой хранятся все файлы, связанные с проектом
 - https://instagram.com/animal_electronics?igshid=13bi2hqwbwv3i – ссылка на наш инстаграм аккаунт
 - http://robot239.ru/index/komanda_animal_electronics/0-112 – ссылка на нашу страницу на сайте

○ Description of future work

| Участник | Задачи по улучшению проекта | Планы по изучению нового |
|----------|---|--|
| Гриша | Проверить систему распознавания команд в "боевых условиях" | Изучать математику, алгоритмы и машинное обучение |
| Марго | Доделать робота-обезьяну, подготовить выступление | Улучшить навыки пайки, программирования и 3д моделирования, освоить новый язык программирования (Python) |
| Маша | Переделать крепление электроники на платформе робота-слона, добавить моторы в голову с ушами и хвост, собрать 2 робота, схожего со слоном | Улучшить навыки программирования компьютерного зрения, электротехники |
| Настя | Улучшить плакат | Улучшить навыки в конструировании и программировании, научиться моделировать в САПР |

Acknowledgements

Огромное спасибо нашим тренерам: Иванову Василию Леонидовичу, Казанцевой Ольге Юрьевне, Коваленко Надежде Сергеевне за помощь в организации работы над проектом и квалифицированную и своевременную помощь в создании и реализации проекта. А также нашим спонсорам: Starline, Кировский завод за финансовую поддержку команды и возможность использовать необходимые материальные ресурсы для создания роботов и Лицею 239 и Центру робототехники за предоставленную возможность воплотить в жизнь наши идеи в сфере робототехники, развить наши способности и навыки, обрести командный дух и опыт участия в соревнованиях.

References

- <https://www.pololu.com/> - сайт Pololu
- <https://www.youtube.com/watch?v=RBnKO6c-Rck&feature=youtu.be> – выступление слонов в цирке (для того, чтобы почерпнуть трюки)
- <https://www.youtube.com/channel/UCEJ8vef1LJNkumsW11Aacng> - ютуб-канал выступлений лиги OnStage 2019 на всемирном этапе в Австралии (для того, чтобы почерпнуть какие-нибудь интересные идеи для выступления)
- <https://youtu.be/TN1qxbxKAaw> - видео, благодаря которому решилась проблема с невозможностью устойчивости робота-слона

- <https://www.bostondynamics.com/spot> - официальный сайт Boston Dynamics Spot
- https://alexgyver.ru/pov_display/ - сайт с информацией о создании POV дисплея
- <https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951> - нейронная сеть AlexNet
- <https://neurohive.io/ru/vidy-nejrosetej/resnet-34-50-101/> - нейронная сеть ResNet-50
- <https://www.machinelearningmastery.ru/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96/> - сравнение нейронных сетей
- <https://habr.com/ru/post/348000/> - общая информация по нейронным сетям
- <http://kb-au.ru/wp-content/uploads/AaSI-2015-2-21.pdf> - теория о балансирующих роботах
- <https://docs.openmv.io/openmvcam/tutorial/index.html> - документация Open MV
- <https://www.nvidia.com/ru-ru/autonomous-machines/embedded-systems/jetson-nano/> - документация Jetson Nano