Решение кейса #1 хакатона BRD2022 от команды öööö

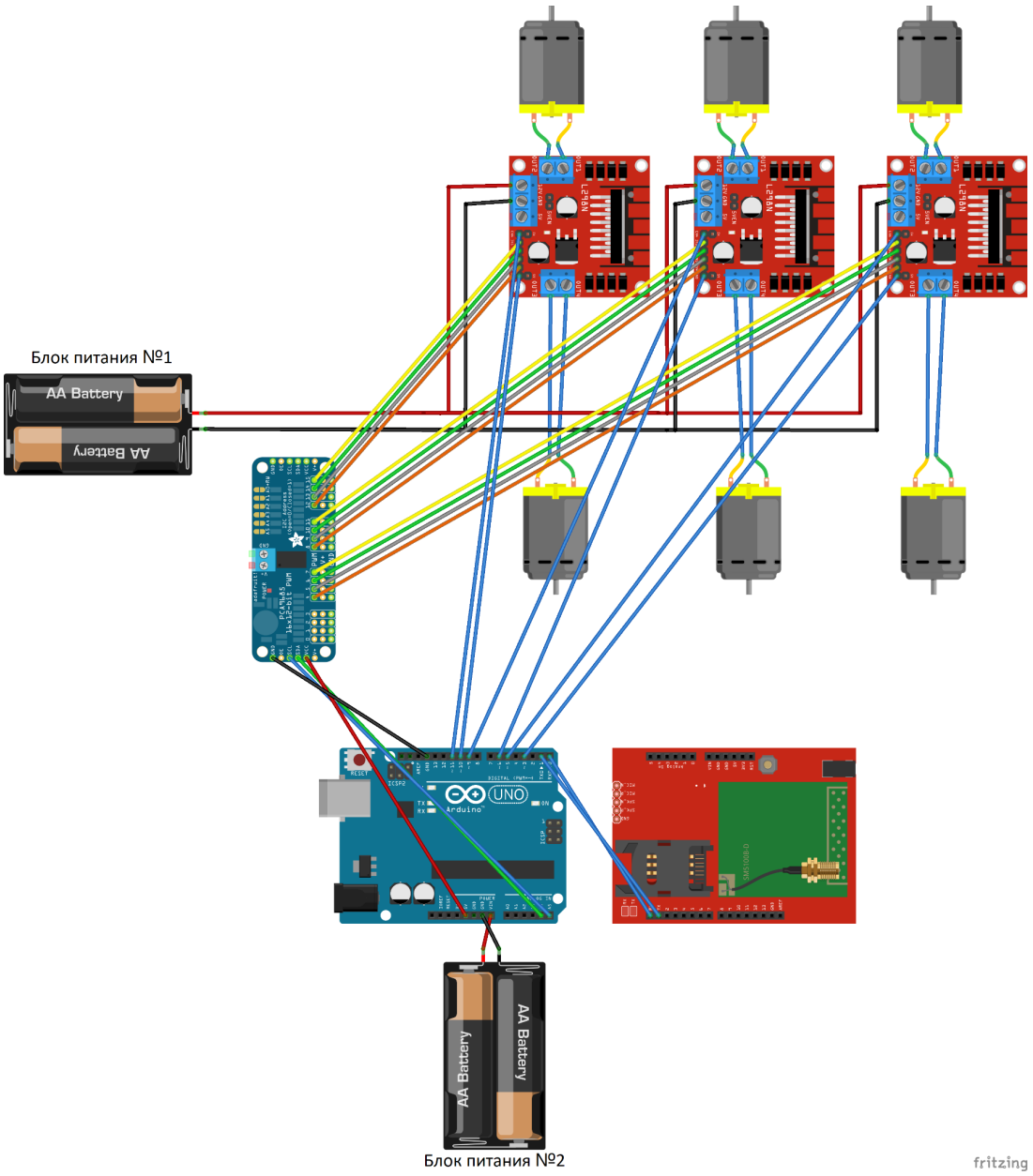
Выполнили Савицкий Илья и Тетерин Дмитрий

Концепт

Создание робота на базе Arduino Uno. Управление будет осуществляться на основе данных с IP камеры, установленной на корпус робота. В корпус установлены 6 колёс, по 3 с каждой стороны. Каждое на своём моторе постоянного тока. Чтобы контролировать скорость вращения, решено установить драйверы L298N через сервомодуль с I2C PCA9685. Для дополнительной безопасности на переднюю часть дна решено установить 4 ИК датчика.

Проблема в том, что Arduino недостаточно мощная, чтобы обрабатывать поток с камеры (или даже чтобы самостоятельно передавать его на сервер), поэтому хочется отказаться от этого решения в пользу базы Raspberry PI.

Для питания решили использовать два блока по 4 аккумулятора для двигателей (на схеме Блок питания №1) и блок из 4 аккумуляторов типа C для ардуины и GPRS модуля (Блок питания №2).



Полная схема находится [тут](#).

Комплектация “Arduino”

В этой комплектации плата отвечает только за движение робота. Камера, установленная на корпусе, отсылает поток на сервер, на котором исполняется код, анализирующий поток с помощью OpenCV. Эта библиотека определяет препятствия, стоящие на пути, и выдаёт вердикт: ехать вперёд, остановиться или совершать разворот в ту или иную сторону.

Ардуино периодически обращается к этому серверу. Для этого нужен любой подходящий чипсет(в нашем случае - SIM800 в составе GPRS Shield, однако код умеет обращаться с более широким спектром чипсетов). Однако два устройства, обращающиеся через интернет, в одном устройстве, – достаточно грустная затея.

Комплектация “Raspberry PI”

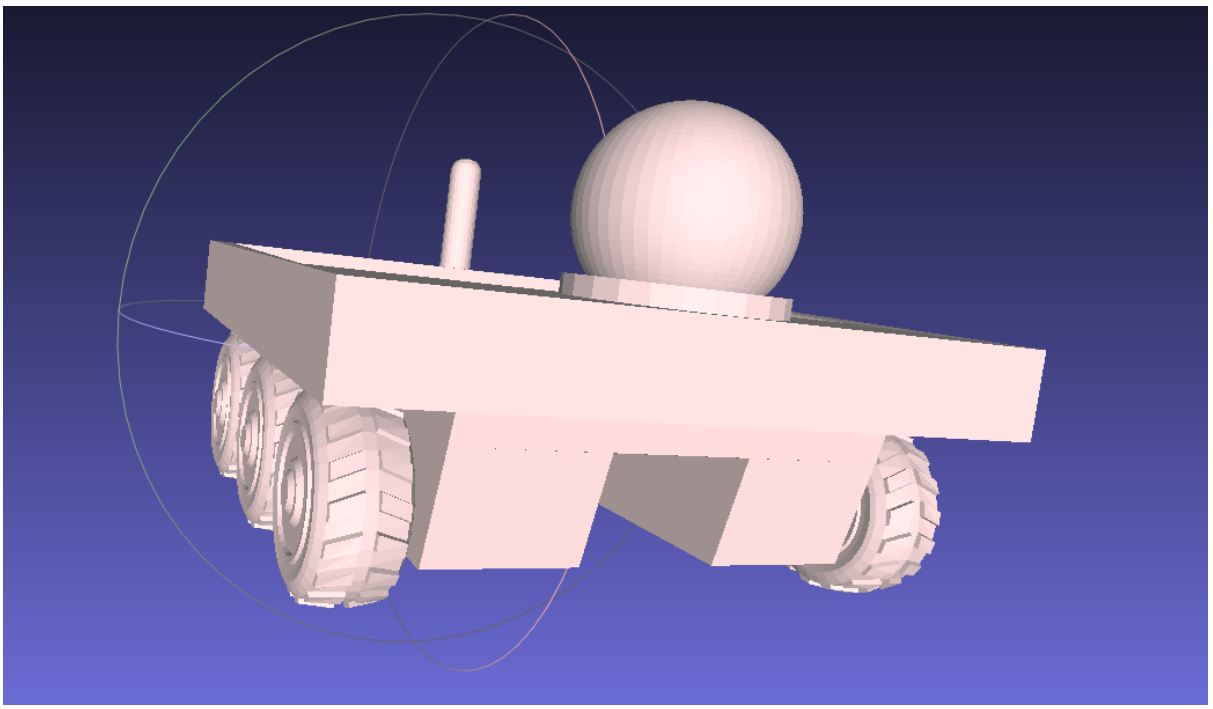
Этот микропроцессор достаточно мощный, чтобы самостоятельно справляться с обработкой изображения. Соответственно, отпадает необходимость в GPRS модуле, а также дополнительный риск с тем, что интернет-соединение может быть нестабильно. Более того, вместо IP-камеры можно использовать обычную вебкамеру.

Описание серверной части (в случае с Raspberry это часть внутренней логики робота)

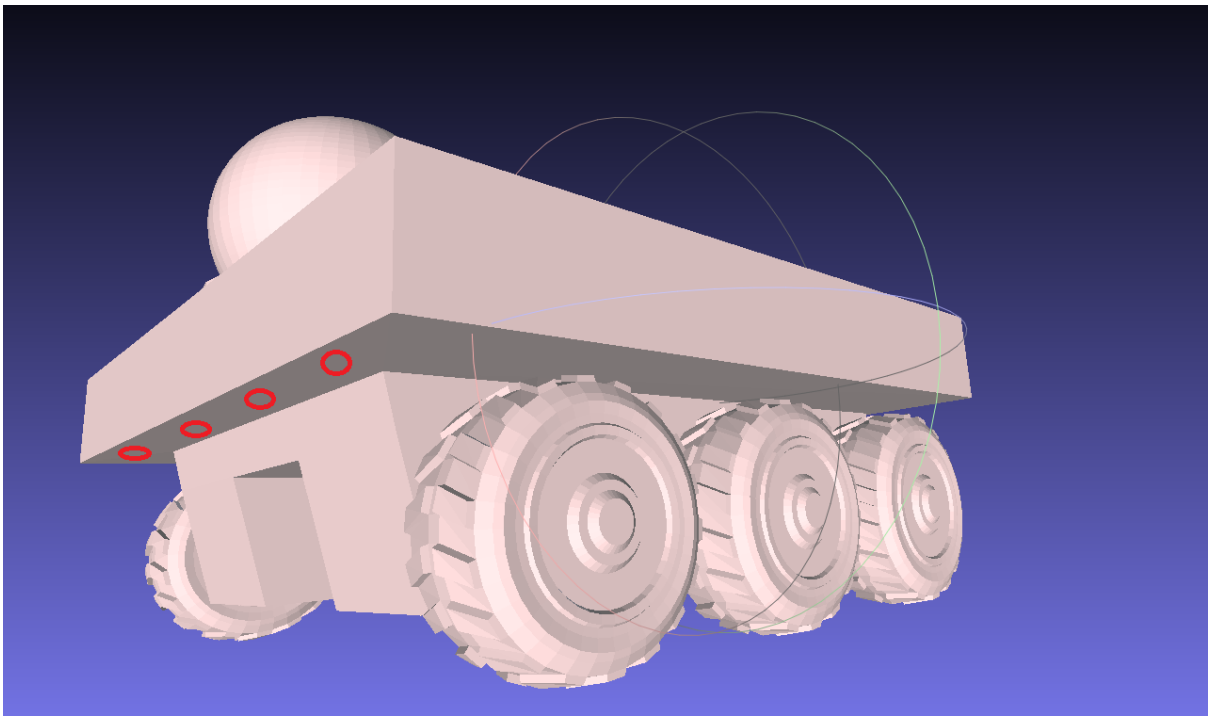
Для обработки запросов с робота был разработан сервис, анализирующий трассу перед роботом. При получении картинки с IP камеры первым делом строятся контуры при помощи алгоритма Кэнни. Далее, получив информацию о контурах, они аппроксимируются полиномами для того, чтобы впоследствии по ним было легко построить так называемые “ограничивающие коробки” - специальные объекты, которые помогают определить примерное положение объекта и процент экрана, который он занимает. После этого все “коробки” классифицируются по принадлежности к левой или правой половине экрана и по занимаемой ими площади, из чего система делает вывод о том, поворачивать роботу влево или вправо. Все это реализовано без методов машинного обучения и на быстрых библиотеках OpenCV и numpy, а значит сервис достаточно быстрый, чтобы обрабатываться в реальном времени (в том числе, и на Raspberry Pi).

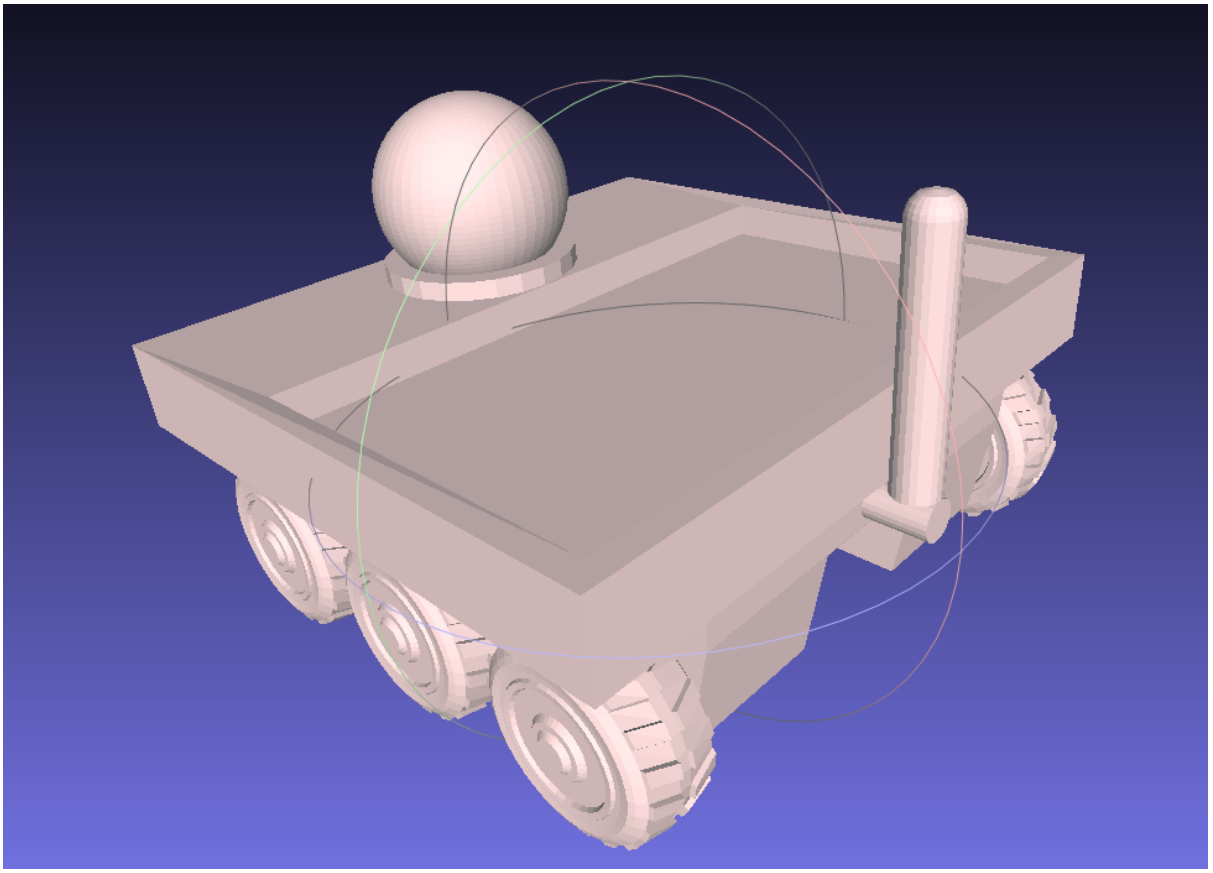
Особенности 3D Модели

Габариты робота максимальные возможные – 25см x 25см. Высота зависит от тестового груза.



Шар наверху – IP-камера, направленная вперёд. В нижней панели 4 ИК-датчика.





Сзади - антенна GPRS модуля. Углубление – багажник. Под углублением место для электроники и блоков питания.

В силу особенностей трассы, которую предстоит пройти, решено сделать 6 колёс и путём расположения блоков питания в отсеке рядом с колёсами стянуть центр тяжести вниз, чтобы не иметь проблем с начальным крутым спуском и последующим подъёмом. В планах добавить амортизацию колёсам и, возможно, “спарить” ведущие пары колёс наподобие колёс в конструкции “Лунохода-1”.

Чтобы обойти цилиндрическое препятствие во второй половине трассы, сделали продольный клиренс. Соответственно, в этом месте OpenCV будет настроен таким образом, чтобы, когда робот проезжает это препятствие, он оставлял его строго по центру.

Почему наше решение лучшее

- Высокая проходимость за счёт шести колёс со своими двигателями
- Использование современных технологий в области компьютерного зрения
- Бесконечная расширяемость возможностей путём улучшения кодовой базы обработки изображения (без изменения кодовой базы самих роботов)
- Высокая надёжность за счёт страхующих ИК-датчиков
- Лёгкая и простая масштабируемость за счёт серверной архитектуры

Что готово

- Проанализирована трасса. В соответствии с препятствиями на ней были выявлены характеристики, которыми должен обладать робот
- Проанализирован рынок комплектующих
- Разработан прототип 3D-модели робота
- Разработан код сервиса компьютерного зрения
- Разработан код “ходовой” части робота на Arduino Uno
- Создана схема контактов робота. Зафиксированы большинство комплектующих

План развития

- По чему есть наработки:
    - ‘полировка” и “отладка” модели робота. Концепция зафиксирована до первых испытаний. Имеется ввиду само качество модели
    - отладки и доработка сервиса компьютерного зрения
- По чему нет наработок:
    - отладка кода “ходовой части” на Arduino
    - доработка сервиса компьютерного зрения - добавление функций трекинга препятствий во время движения робота

Строение репозитория

Все, что было выполнено нами на протяжении этого хакатона(за исключением этого документа) выложено здесь. Вот как этот репозиторий устроен:

- robot-code
    - robot.ino - код “ходовой” части робота для Arduino. Зависит от библиотек Adafruit_PWMServoDriver и TinyGsmClient. Пареметрами этого кода являются адрес и порт сервера, на котром работает сервис компьютерного зрения, а так же некоторые другие параметры, связанные с работой SIM-карты.
- robot-object
    - robot.obj - объектный файл робота. Без текстурных координат.
- robot-schema
    - robot.fzz - схема соединений в роботе в формате Fritzing.
    - robot_bb.png - экспортируемая картинка соединений. Может отставать на пару коммитов от схемы во Fritzing
- service
    - Pipfile & Pipfile.lock - файлы системы контроля зависимостей pipenv
    - viewer.py - скрипт, позволяющий интерактивно просмотреть работу всех алгоритмов компьютерного зрения в реальном времени
    - verdict.py - скрипт оценки окружения при помощи компьютерного зрения с убранной визуализацией с целью ускорения.

Развертка системы

1. Склонировать репозиторий

```
git clone https://github.com/ooo-team/BRD2022
```

2. Зависимости кода ходовой части

Для компиляции кода ходовой части из менеджера библиотек Arduino установите следующие библиотеки с исходным кодом

- TinyGSM
- Adafruit PWM Driver

После этого возможно верифицировать, скомпилировать и загрузить код на Arduino Uno

3. Синхронизация зависимостей для сервиса компьютерного зрения

Контроль зависимостей осуществляется при помощи Pipenv. Убедитесь что он установлен у Вас перед использованием

```
cd service
pipenv shell
pipenv sync
```

После чего можно запускать сервис

Комплектующие

- Напечатанный на 3D принтере корпус (первый прототип модели приложен)
- ИК датчики (4x, ссылка, 190*4=760₽)
- Драйвер L298N (3x, ссылка, 3*450 = 1350₽)
- Сервомодуль PCA9685 (ссылка, 250₽)
- Колесо с двигателем и редуктором (6x,ссылка, 6*110=660₽); Возможно, понадобится этот комплект двигатель+редуктор: ссылка

Комплектация “Arduino”

- IP камера (например, такая, 4900₽)
- Arduino UNO (ссылка, 3540₽)
- GPRS плата (ссылка, 3340₽)
- Сим-карта с выходом в интернет для Arduino и Камеры

Комплектация “Raspberry PI”

- Raspberry PI (хочется верить, что около 4000₽, но в силу некоторых нюансов получается 15000₽)
- Камера (ссылка, 1000₽)