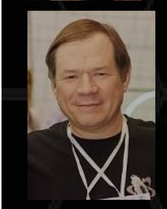
Информация о команде

Пильщиков Григорий (9 класс)

Цыганкова Мария (9 класс)



Косаченко Сергей Викторович - наставник команды

Роли участников команды: Пильщиков Григорий и Цыганкова Мария конструируют и программируют роботов вместе.

Аннотация

Цыганкова Мария и Пильщиков Григорий разработали и модифицировали робота длякатегории RCJ Rescue Maze, а также алгоритм движения и распознавания для выполнения поставленных целей.

Описание Робота

Стратегия выполнения задания роботом:

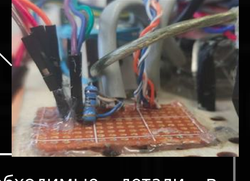
Наш робот ищет в лабиринте "жертв" по правилу "чередующийся руки". В момент запуска случайно выбирается ведущая рука для прохождения лабиринта, по которой в дальнейшем двигается робот. Во время движения робот анализирует стену: если он находит черный цвет, проверяет какую букву нашел, если нашел цвет, считает, что нашел цветную жертву. Кроме того, во время движения робот анализирует цвет тайла на котором находится для определения ямы и болота.

Использование датчиков:

В нашем роботе используется 5 инфракрасных дальномера sharp 2y0a21 f 36, HiTechnic NXT Color Sensor V2 и камера, считываемая Raspberry Pi Zero, которая передает данные на meAURIGA, кроме того у нас есть встроенные датчики считывания угловой скорости (энкодоры) с помощью которых управляем моторами из набора MakeBlock. Данные с Raspberry на meAURIGA мы передаем по UART, для этого мы самостоятельно спаяли переходную плату - так называемый UART hub с Raspberry на порт MakeBlock RJ12, для подключения моторчика выгрузки комплектов мы также спаяли переходную плату с портов MakeBlock RJ12 на разъемы с шагом 2.45 мм. Кроме того, на роботе стоит стабилизатор питания с 5 В на 3.3 В. Для более удобного старта на роботе вынесена кнопка запускающая самого робота. Распиновку meAURIGA и Raspberry Pi Zero, мы брали с официальной тех. документации.

Переходная плата с портов MakeBlock RJ12 на разъемы с шагом 2.45 мм, стабилизатор питания с 5 В до 3.3 В

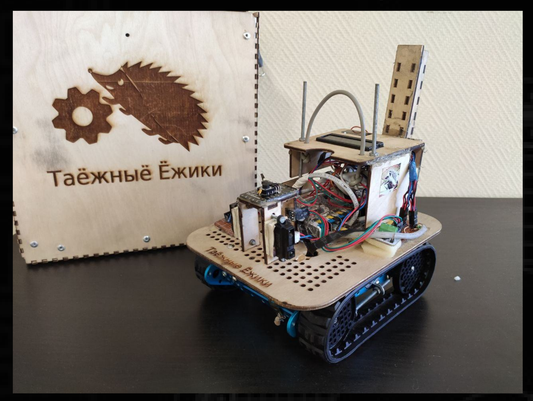
Самодельный UARThub

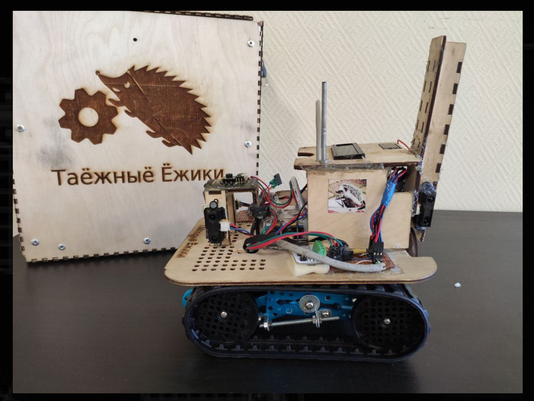


Изготовления робота:

Сначала мы начертили все необходимые детали в векторном формате (в программе inkscape), после чего вырезали на фрезерном станке, изначально вырезали из картона, но позже для прочности перешли на фанеру, однако из-за не до конца спланированных отверстий, еще раз перечертили и перевырезали основную платформу. Используя вырезанные детали, детали из набора MakeBlock и спаянные нами вышеперечисленные платы собрали самого робота

Команда "Таёжные Ёжики" из Томска представляет:



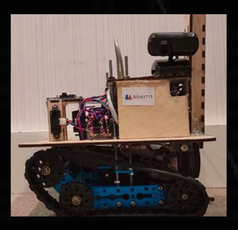


Конструкция робота

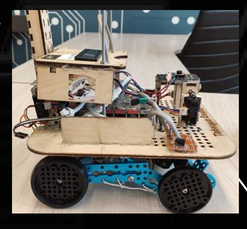
Конструкция робота предполагает несколько этажей с разным функционалом: на нулевом располагаются моторы и Color Sensor, на первом – аккамулятор для meAURIGA и пауэрбанк для Raspbberry, на втором – сама meAURIGA и Raspbberry, USBhub, самодельный UART hub, переходная плата с портов MakeBlock RJ12 на разъемы с шагом 2.45 мм, стабилизатор питания с 5 В до 3.3 В и инфракрасные дальномеры, на третьем – два экрана для вывода информации и кнопка для быстрого запуска и сама камера. В пространстве между 2-ым и 3-ем этаже робота располагается коробочка для проводов и meAURIGA.
Сначала мы начертили все необходимые детали в векторном формате (в программе inkscape), после чего вырезали на фрезерном станке, изначально вырезали из картона, но позже для прочности перешли на фанеру, однако из-за не до конца спланированных отверстий, еще раз перечертили и перевырезали основную платформу. Используя вырезанные детали, детали из набора MakeBlock и спаянные нами вышеперечисленные платы собрали самого робота

1

2



3



4

5

Модернизации конструкции:

Первая версия шасси робота была собрана полностью из конструктора MakeBlock и имел всего 1 этаж. Во 2-й версии робота (можно увидеть на 1- х чертежах конструкции) робот состоял из 2-х этажей, где на 2-ом этаже располагались ультразвуковые датчики расстояния и meAURIGA, при размещении камеры команда решила сделать 3-й этаж, на котором и будет располагаться камера. Еще позже на этом этаже появились экранчики для вывода информации, а на 2-ом этаже ультразвуковые датчики заменились лазерными. В 4-й версии полностью поменяли шасси для робота: они стали полностью плоскими (т.е. не имеют наклона гусениц, который был в предыдущий версиях) и стал иметь форму треугольника, для возможности регулировки натяжения гусениц. Позже мы разработали более удобный механизм регулировки при помощи осей с резьбой и гаек. 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 увеличило его проходимость.

Этажи

Эскиз/чертеж 2-й версии робота на бумаги

Определение жертв:

Жертвы мы обнаруживаем с помощью камеры и библиотеки OpenCV 4. Для начала настраивается бинаризация для 1 из 3 цветов: красного, жёлтого и чёрного. После бинаризации определяем контуры и проверяем самые большие контуры, подходят ли они. 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 - чёрные, то это s, если нижний и верхний - не чёрные - то это H, а если нижний – чёрный - то это U.

Опыт участия и успехи команды в различных робототехнических соревнованиях и фестивалях:

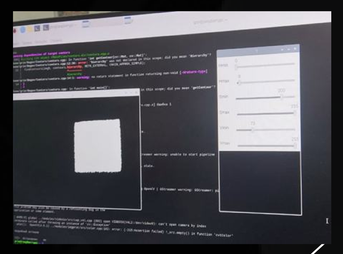
Международный фестиваль Робофинист 2022 – 3 место в категории Robocup Junior Rescue Maze, г.Санкт-Питербург
Кубок Губернатора Томской области 2023 – 1 место в состязании роботов с техническим зрением памяти Виктора Ширшина
Кубок Губернатора Томской области 2022 – 2 место в состязании роботов с техническим зрением памяти Виктора Ширшина
XIII Региональная олимпиада по образовательной робототехнике школьников Томской области 2024 – 2 место в состязании Robocup Junior Rescue Maze
XII Региональная олимпиада по образовательной робототехнике школьников Томской области 2023 – 1 место в состязании Robocup Junior Rescue Simulation Webots Erebus Открытый Российский чемпионат по робототехнике 2023 – 2 место в состязании Robocup Junior Rescue Simulation Webots Erebus
2022 год – Участие Цыганковой Марии и Пильщикова Григория в фестивале подводной робототехнике «АкваРобоФест» в г.Асино

Программное обеспечение

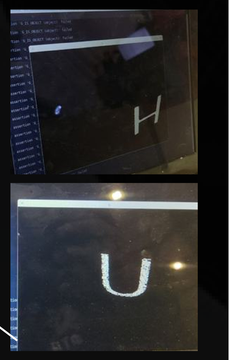
В качестве программного обеспечения мы использовали написанную самостоятельно программу на C/C++ с использованием библиотеки OpenCV 4 на операционной системе Ubuntu 21.04, для программирования MakeBlock используем программу arduino IDE.
Мы взяли веб-камеру с частотой 30 кадров в секунду. Благодаря этой веб-камере, мы захватываем с нее кадры и бинаризируем изображение, ища цвета цветных жертв и черного для букв.

Определение черной линии

Освещение на поле не предсказуемо, в прошлом году нам требовалась проводить бинаризацию для черного цвета перед каждым заездом, что задействовало очень много времени. Поэтому мы решили разработать алгоритм автобинаризации. Алгоритм, который мы придумали ищет на изображении самый светлый, самый темный и среднестатистическое значение пикселя и формулой рассчитывает данные для бинаризации на черный (чем темнее освящение на поле, тем больше делителей). Таким образом, робот может находить черный цвет даже в сумраке.
Примерная формула,которую мы используем на псевдокоде:
насыщенность_чёрного < (min+ср_знач)/2+ min



Ручная бинаризация на цветные метки



Автобинаризация на черный

Обсуждение и заключение

Во время разработки и тестирования нашего робота команда столкнулась со множеством проблем, вот несколько из них.

- Освещение на тренировочных полях было очень плохим и к тому же менялось с течением суток, настройка бинаризации очень долгой, из-за чего было принято решение придумать самодельный алгоритм автобинаризации.
- Изначально мы передавали данные с Raspbberry на meAURIGA через Serial и по какой-то причине Raspbberry пыталась питаться через meAURIGA, а не через пауэрбанк, из-за чего сгорала. Поэтому было принято решение перейти с serial на UART.
- Зачастую у Raspbberry сбивается время и программа не компилируется, выводя ошибку «Время изменения файла находится в будущем», всякий такой раз нам приходилось копировать каталог под новым временем и сносить старый с неправильным.
- Библиотеки MeAuriga.h. Мы не могли найти подходящие команды для программирования робота. Решением стало то, что мы залезли в саму библиотеку и смогли уже проанализировав её, определить необходимую команду.

Планы на будущее:

Команда будет продолжать обучаться техническому зрению, в дальнейшем планируем усовершенствовать и уменьшить конструкцию робота, поставить вторую камеру и научиться считывать информацию одной Raspberry с нескольких камер. Разработать упрощенный, но более эффективный алгоритм технического зрения.

Благодарности и ссылки

Благодарим нашего тренера Косаченко Сергея Викторовича, НПП Томскую Электронную компанию коммерческую помощь. А также Томский Физико-Технический Лицей

Подробнее вы можете посмотреть наши коды, чертежи, тех. документацию и д. информацию про наших роботов: