

Робот- картограф



Группа: С-12а-19

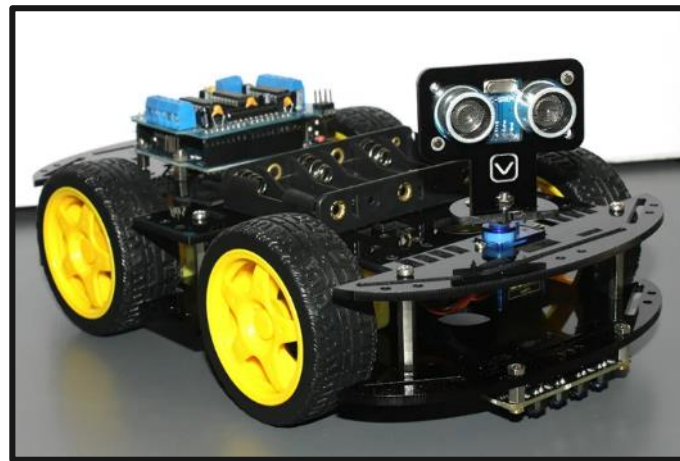
Студенты:
Сергеев К. О.
Мхитарян Д. К.

Цель исследования

Мы стремимся научиться проектировать и программировать робототехническое устройство при помощи платы Arduino и датчиков движения, а также пользоваться 3D-принтером и лазерной резкой для создания необходимых комплектующих.

Особенностью данного исследования является собственный алгоритм, основанный на волновом алгоритме (алгоритме Ли), используемый спроектированным нами мобильным роботом.

Примерный вид нашего робота



Способность ориентироваться является фундаментальной проблемой как для животных, так и для мобильных искусственных систем.

Настоящая работа посвящена проблеме построения и интерпретации карты автономным роботом.

Например, это используется в роботах-пылесосах.

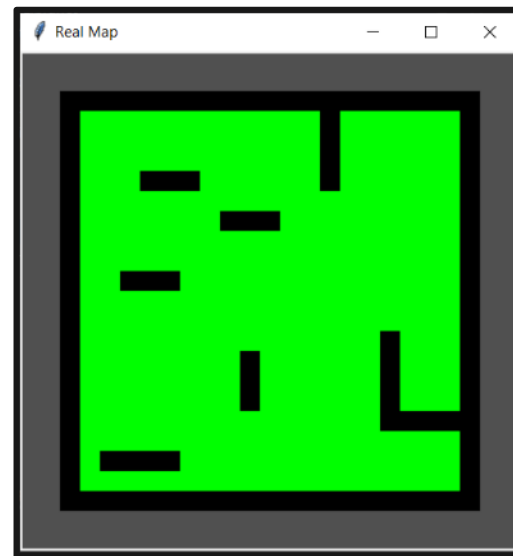
Робот-пылесос



Задача

*Пример результата работы
нашего робота*

Задачей нашего робота является получение карты помещения путём перемещения в нём и считывания данных с ультразвукового датчика.



Описание проекта

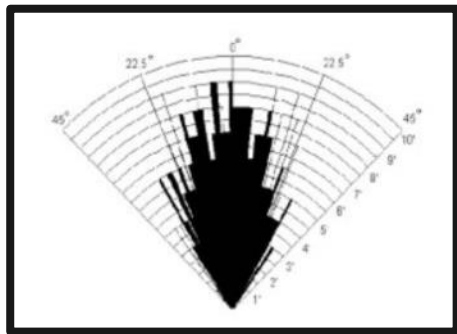


Наше робототехническое устройство предназначено для построения 2D-карты закрытого помещения, например, комнаты с находящимися в нем предметами для дальнейшей навигации и передвижения.

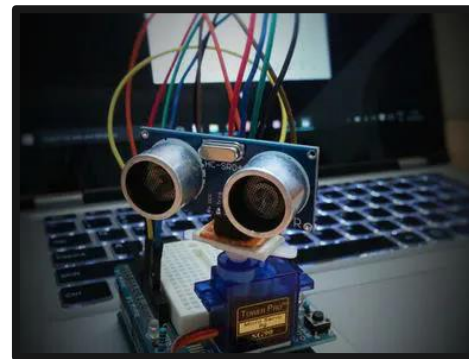
Для построения карты будем использовать ультразвуковой датчик расстояния. Принцип его работы в том, что он генерирует звуковые волны в диапазоне, недоступным для слуха, а затем измеряет расстояние. Расстояние определяется по времени, которое необходимо для того, чтобы эти волны дошли до какого-либо препятствия и вернулись обратно. Тот же принцип используется летучими мышами при полетах ночью.

Ультразвуковой датчик расстояния мы поставим на сервопривод, так как рабочий угол ультразвуковой датчика в среднем 30 градусов, а эффективный диапазон замера составляет всего 15 градусов.

Описание проекта



Область работы дальномера

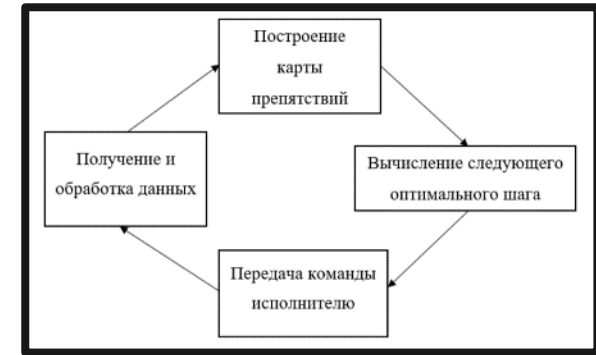


Примерный вид дальномера на сервоприводе

Так как вычисления планируются объёмные, а плате Arduino не хватает вычислительной мощности, то мы будем использовать личный ноутбук в качестве удаленного сервера, на котором будут происходить все вычисления. Связь удаленного сервера и робота будет происходить с помощью Bluetooth соединения. Для этого предварительно поставим на робота Bluetooth-модуль.

Этапы работы

1. Сборка робота на колёсах
2. Разработка алгоритма 1 “Получение и обработка данных с ультразвукового датчика”
3. Разработка алгоритма 2 “Построение карты препятствий”
4. Разработка алгоритма 3 “Вычисление следующего оптимального шага”
5. Разработка алгоритма 4 “Передача команды движения плате Arduino”
6. Разработка управляющего циклического алгоритма удаленного сервера (Алгоритмы 1 -> 2 -> 3 -> 4 -> 1 -> ...)
7. Компьютерная симуляция работы нашего робота на Python
8. Тестирование в прямоугольной комнате с препятствиями на полу

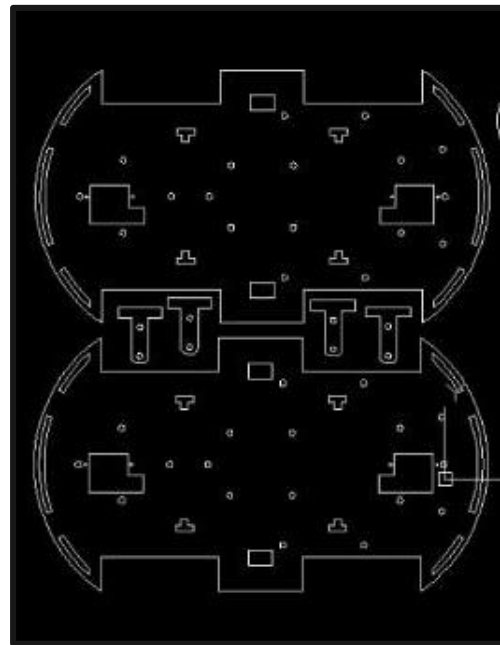


Управляющий алгоритм

Комплектующие

- ✓ 2 платформы и крепления
- ✓ 1 сервопривод
- ✓ 1 Bluetooth-модуль HC-06
- ✓ 1 Управляющая плата Arduino UNO
- ✓ 1 Ультразвуковой датчик расстояния HC-SR04
- ✓ 1 Удаленный сервер (ноутбук)
- ✓ 4 мотора-редуктора
- ✓ 4 колеса
- ✓ Батарейки
- ✓ Провода и стяжки
- ✓ Драйвер для двигателя L298N
- ✓ Винты и гайки
- ✓ Бокс для батареек

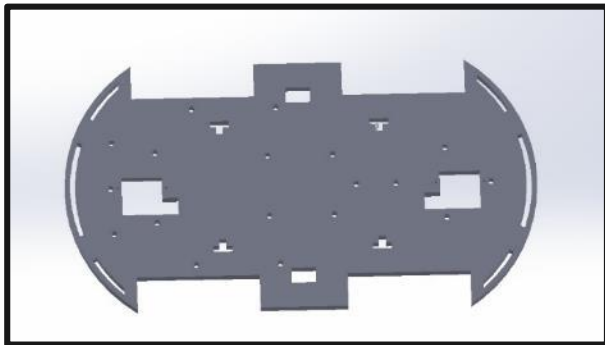
Эскиз платформы для лазерной резки



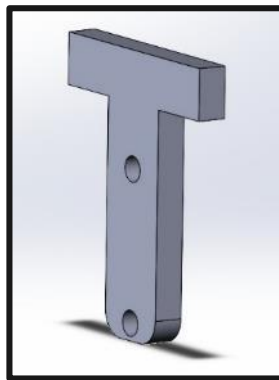
Моделирование робота на колёсах

Некоторые детали в САПР SolidWorks

Платформа



Т-образное крепление

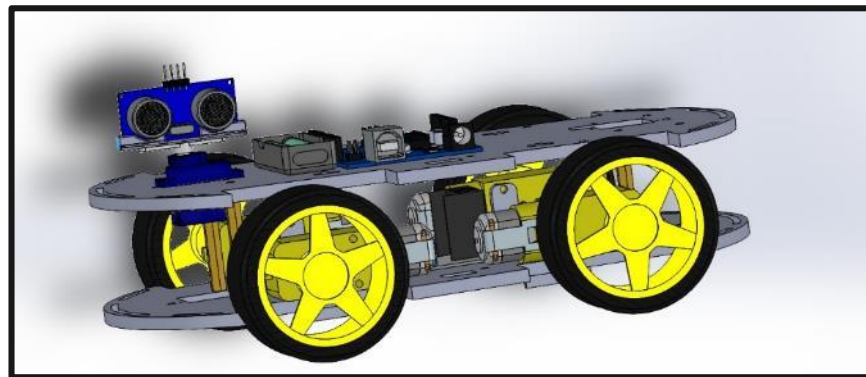
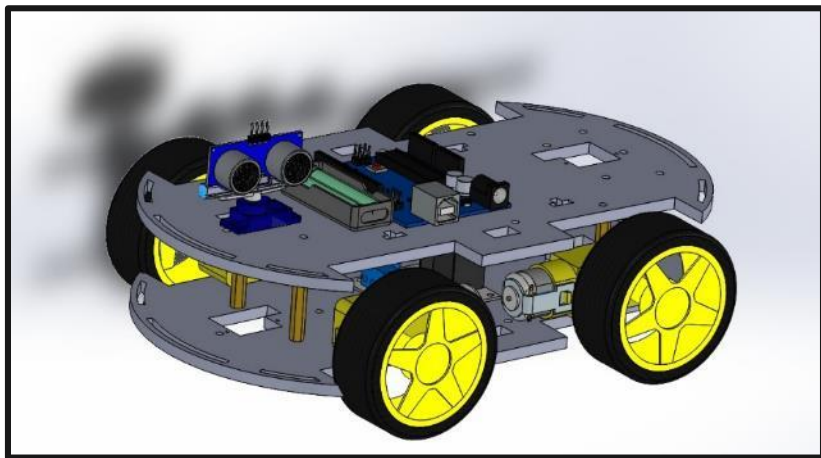


Винт



Моделирование робота на колёсах

Готовая сборка в САПР SolidWorks



Разработка алгоритма



Мы определили эффективную область работы нашего ультразвукового датчика на расстоянии 50 см вокруг него.

Если датчик сигнализирует, что расстояние перед ним больше 50 см, считаем, что перед нами нет стены на расстоянии 50 см.

Если датчик сигнализирует, что расстояние перед ним меньше 50 см, но больше 25 см, считаем, что перед нами стена на расстоянии 25 см.

Если датчик сигнализирует, что расстояние перед ним меньше 25 см, считаем, что стена находится непосредственно перед нами.

Длина нашего робота 25 см.

Пусть 1 клетка – 25x25 см.

Пусть робот может двигаться только вверх, влево, вправо, вниз.

Разработка алгоритма



Рабочий угол обзора дальномера на сервоприводе 180° .

Мы получаем данные от дальномера, учитывая его угол поворота относительно карты.

Пусть начальный угол поворота робота относительно карты 270° .

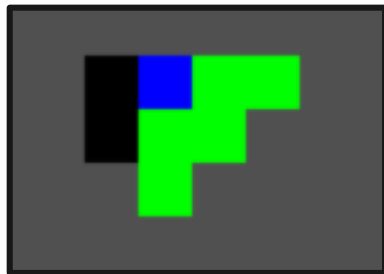
Например, если угол поворота робота относительно карты 90° , угол поворота дальномера относительно карты 0° , а данные 30 см. То мы отмечаем клетку справа от робота исследованной, а клетку справа через одну стеной.

А если угол поворота дальномера относительно карты 45° , и справа или сверху нет стены, то клетка по диагонали справа сверху будет стеной, так как $25\text{см} \cdot \sqrt{2} \approx 35.36\text{ см}$, а $30\text{см} < 35.36\text{ см}$.

Если справа и сверху стены, то мы считаем, что не можем ничего сказать о клетке по диагонали справа сверху, так как мы не можем туда добраться.

Разработка алгоритма

Пример построения карты роботом на первом шаге



Зелёные клетки – это исследованная зона роботом. Черные клетки – это стены.

Синяя клетка – это робот.

В этот момент робот развёрнут относительно карты на 270° .

Так как непосредственно справа от робота находится стена, то через эту стену он не видит, поэтому клетка справа через одну осталась неисследованной.

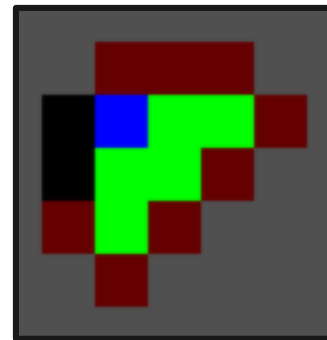
Если робот идёт вправо, то его угол относительно карты будет 0° , если вверх, то 90° и т.д.

Разработка алгоритма

Находим ближайшие неисследованные точки(потенциальные). В примере они выделены бордовым цветом.

Будем перебирать все неисследованные точки и выбирать те, до которых можно добраться, то есть если один из соседей точки является исследованной точкой.

Пример

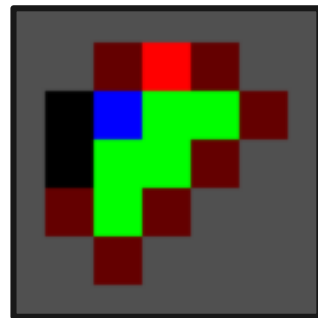


Разработка алгоритма

Найдём из этих точек точку, до которой короче всего. Для этого применим алгоритм, основанный на волновом алгоритме Ли с окрестностью фон Неймана. Но так как робот не может двигаться назад, убираем точку сзади из окрестности для алгоритма.

Будем идти волной от нашего робота. Путь до нашего робота считаем 0. Пути до остальных клеток будем хранить в матрице путей, как и ориентации робота если бы он пришёл в эти клетки. Изначально у всех остальных клеток путь будет -2, а ориентация -1. При итерации (волне) проверяем соседние клетки кроме той, которая прямо сзади робота, и если для них не найден путь, и это не стена, записываем в эти клетки путь до предыдущей клетки плюс один и записываем ориентацию робота, если бы он в эти клетки пришёл. Добавляем клетки, для которых был записан путь, и они не являются потенциальными для участия в следующей волне. Алгоритм завершается, когда найдены пути до всех потенциальных точек найденных до этого. Дальше берётся первая точка, до которой короче всего (в примере отмечена ярко красным).

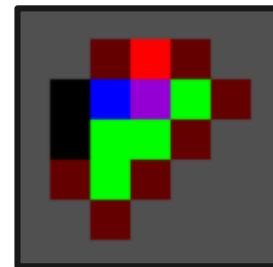
Пример



Разработка алгоритма

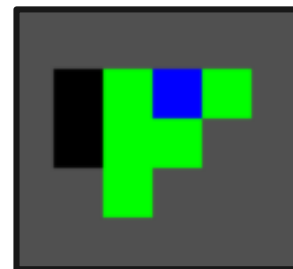
Пример восстановленного пути

Далее восстанавливаем этот кратчайший путь. Начинаем с точки, до которой короче всего. Если путь до соседней точки в матрице путей меньше на один, то это будет следующая точка. Алгоритм завершается, когда мы дошли до робота. Пример восстановленного пути отмечен фиолетовым.



Дальше мы делаем один шаг по кратчайшему пути и повторяем все предыдущие этапы по порядку, начиная с обработки данных и построения карты. Робот останавливается, когда до всех оставшихся неисследованных точек он не может добраться.

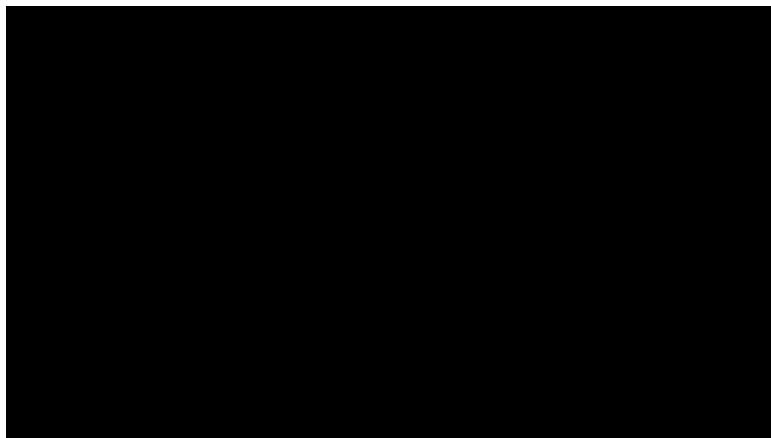
Делаем один шаг по восстановленному пути



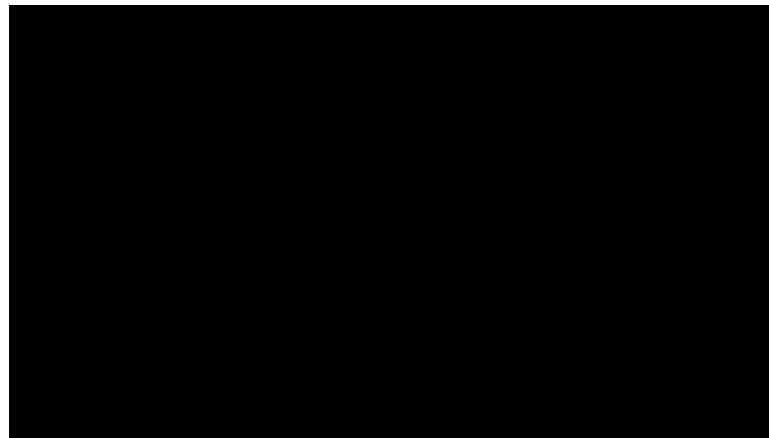
Компьютерная симуляция



Симуляция с остановкой в 0.3 секунды между этапами (видео)



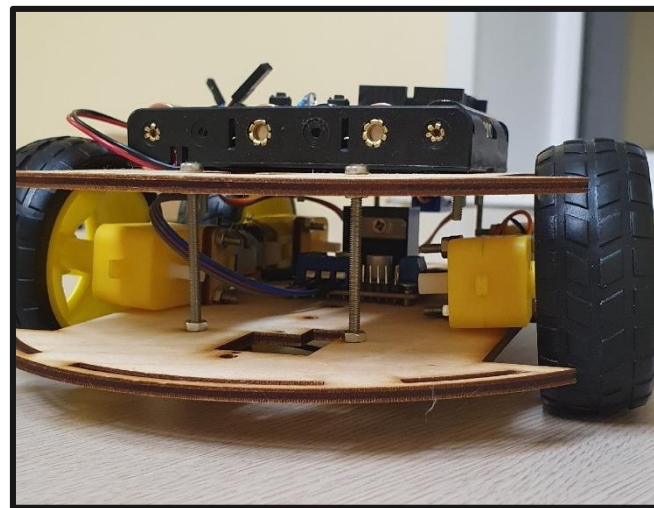
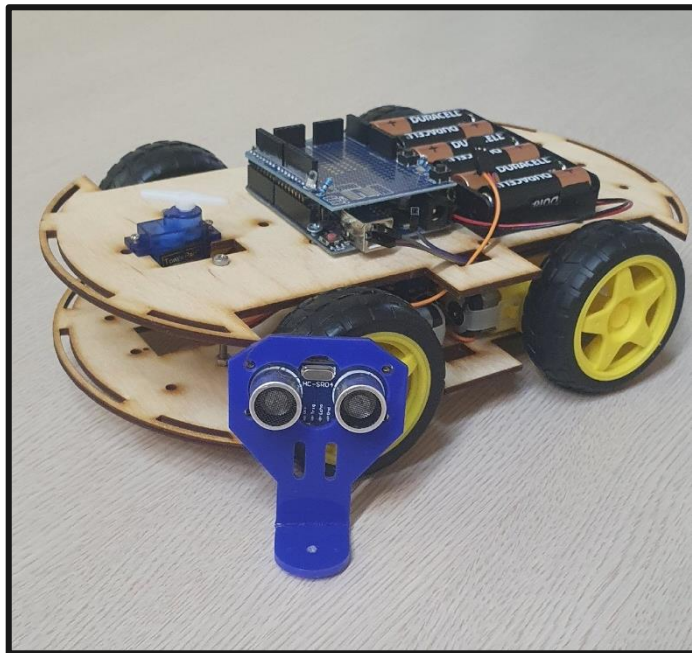
Полная симуляция без остановок между этапами (видео)



Если видео не запускается:

<https://disk.yandex.ru/i/1ykQNF9dWVEKRg> <https://disk.yandex.ru/i/lZa3JtDdoZULxg>

Собранный робот



- Обучить нейронную сеть находить оптимальную точку из потенциальных, зная матрицу расстояний, а не просто выбирать точку до которой короче всего, внедрить нейронную сеть в алгоритм.
- Перенести алгоритм в Arduino
- Протестировать работу робота в прямоугольной комнате с препятствиями

1. Ссылка на программу со всеми комментариями:

https://github.com/FKz11/Portfolio_Data_Science/blob/main/Robot_Map/Robot_Map.py

2. Ссылки на приложение, чтобы можно было запустить симуляцию на личном компьютере:

С остановкой в 0.3 секунды между этапами:

<https://disk.yandex.ru/d/HPm-2ao6ccnx0A>

Без остановок между этапами:

<https://disk.yandex.ru/d/8tt179VbTXlk9Q>

1. Книга: Мобильные роботы на базе Arduino. Автор: Михаил Момот
2. ВКР 2018: <https://nauchkor.ru/pubs/mobilnaya-sistema-dlya-optimalnogo-postroeniya-2-d-shemy-zamknutogo-prostranstva-s-prepyatstviyami-na-mk-arduino-5c1a61377966e104f6f856ba>
3. Алгоритм Ли: https://ru.wikipedia.org/wiki/Алгоритм_Ли
4. Окрестность фон Неймана: https://ru.wikipedia.org/wiki/Окрестность_фон_Неймана