**Тренировочное задание по компетенциям**

**«Электроника» и «Ардуино»**

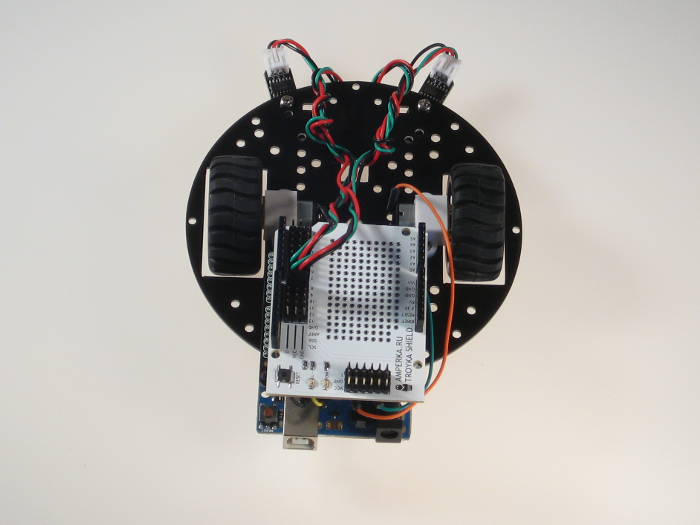
Робот, ездящий по линии под управлением Arduino

### В данной статье будет описан процесс создания робота, ездящего по линии. Эта задача является классической, идейно простая, она может решаться много раз, и каждый раз вы будете открывать для себя что-то новое. Решение этой задачи и реализация полученного решения позволяют приобрести необходимые начальные навыки для дальнейшего совершенствования в робототехнике.

### Существует множество подходов для решения задачи следования по линии. Выбор одного из них зависит от конкретной конструкции робота, от количества сенсоров, их расположения относительно колёс и друг друга.

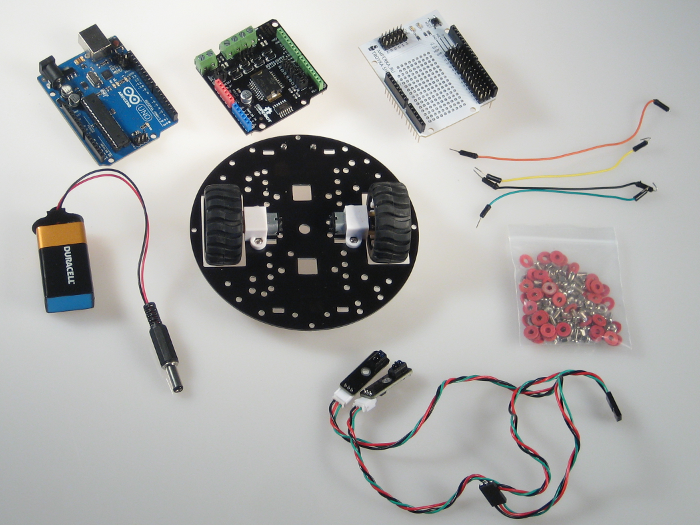
### В нашем примере будет собран робот на лёгкой платформе с двумя колёсами и двумя датчиками линии, расположенными на днище робота перед колёсами.

### В результате выглядеть он будет так:



Что понадобится

### Для нашего примера понадобятся следующие детали:



### [Arduino Uno](https://amperka.ru/product/arduino-uno)

### [Двухколёсная платформа miniQ](https://amperka.ru/product/miniq-chassis)

### [Motor Shield](https://amperka.ru/product/arduino-motor-shield)

### [Troyka Shield](https://amperka.ru/product/arduino-troyka-shield)

### Пара [датчиков линий](https://amperka.ru/product/digital-line-sensor)

### Несколько [соединительных проводов](https://amperka.ru/product/wire-mm) и болтов и гаек для крепления датчиков и Arduino Uno

### [Кабель питания от батарейки Крона](https://amperka.ru/product/krona-21mm-cable) и сама батарейка

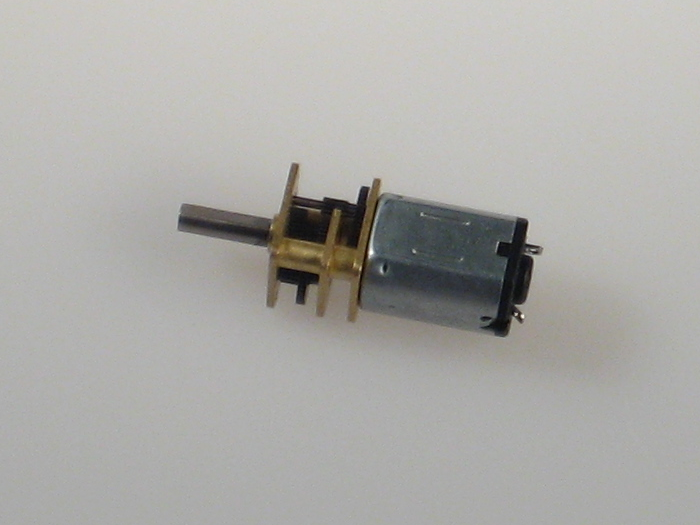
### Вообще говоря, лучше было бы использовать NiMH-аккумуляторы: они лучше отдают ток и значительно дольше держат напряжение, но для целей этого проекта одной батарейки на 9 В вполне хватило.

Собираем робота

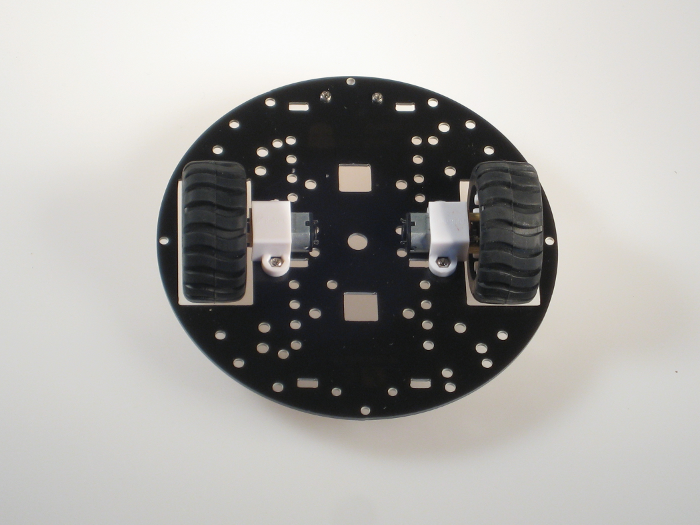
### Сначала соберём робота, установим всю механику и электронику.

Собираем платформу

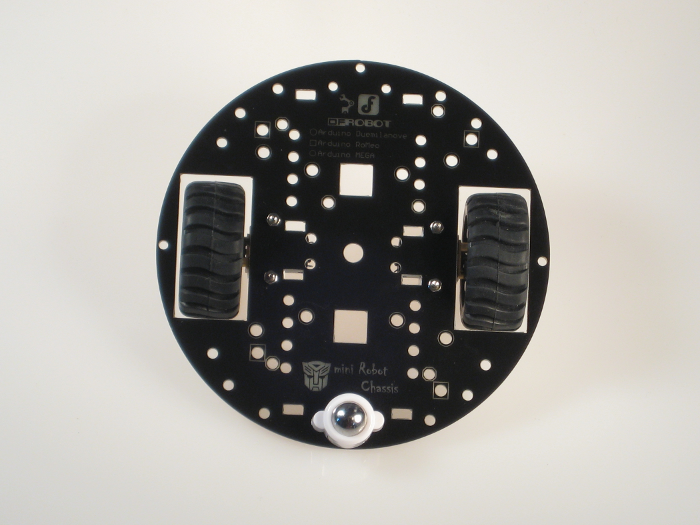
### Для начала прикрепим колёса к моторам.



### Затем с помощью пластиковых П-образных креплений прикручиваем моторчики к платформе. Обратите внимание на взаимное расположение крепления и моторчики: в креплении есть небольшие углубления, так что если всё соединить правильно, то моторчики будут крепко держаться и никуда не выскочат.



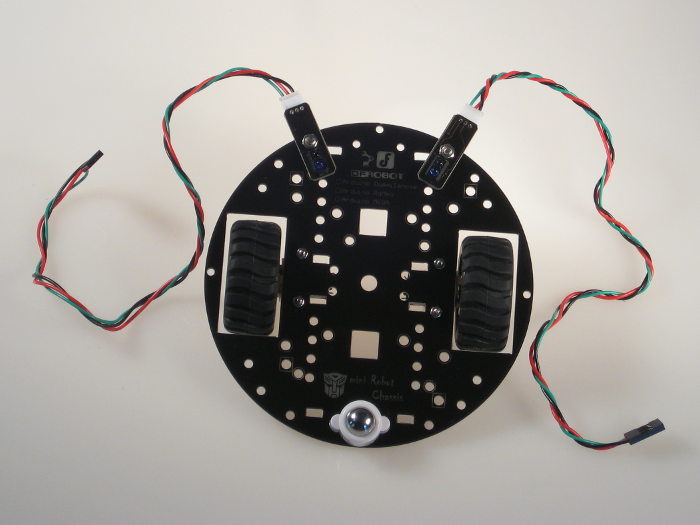
### Теперь крепим балансировочный шар.



### Отлично! Платформа собрана. Если вам кажется, что колёсам отведено слишком мало места и они трутся о платформу, то скорее всего вам нужно посильнее надавить на колёса, чтобы они плотнее сели на вал мотора.

Крепим сенсоры

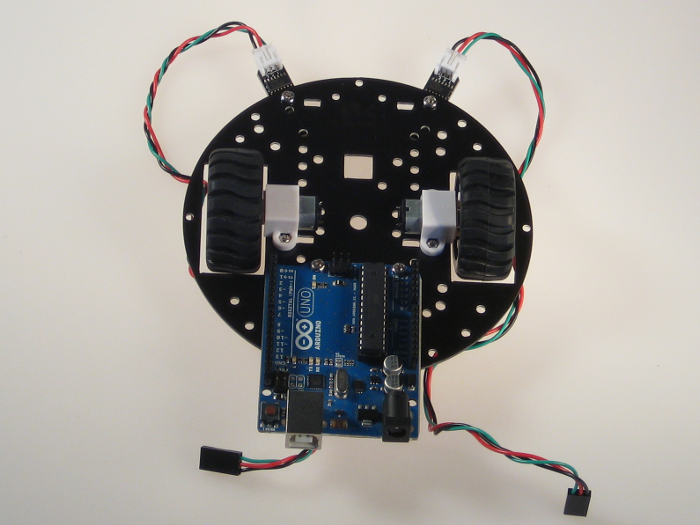
### Закрепим их, как показано на фото:



### Можно было бы выбрать и другое место. Это могло бы сделать контроль проще или сложнее, а самого робота более или менее эффективным. Оптимальное расположение — вопрос серии экспериментов. Для этого проекта просто был выбран такой способ крепления.

Крепим Arduino

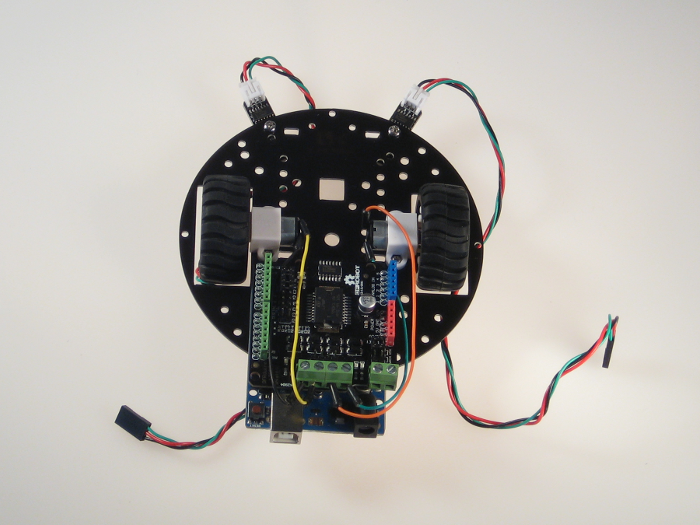
### Arduino закрепим с противоположной стороны двумя винтиками и гайками.



### Опять же, можно выбрать и другое место. Например над колёсами, если приподнять Arduino на латунных стойках. Это изменило бы положение центра масс и повлияло бы на эффективность робота в лучшую или худшую сторону.

Крепим Motor Shield и соединительные провода

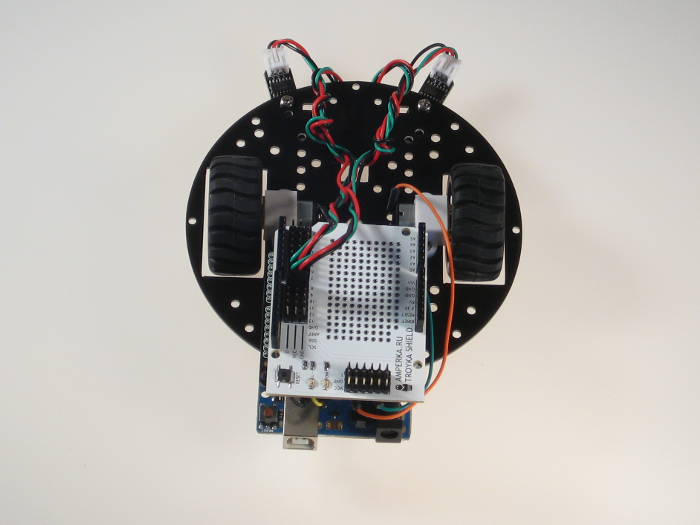
### Установим Motor Shield на Arduino и подсоединим соединительные провода. Обратите внимание, чтобы соотвествовать программному коду из примера ниже, моторчики соединены с Motor Shield так: правый — к клеммам M1 с прямой полярностью (плюс к плюсу), а левый — к M2 с обратной (плюс к минусу).



### В этом проекте, для экономии времени концы соединительных проводов просто скручены с контактами моторов. При работе «начисто» стоит жёстко припаять провода к моторам.

Крепим Troyka Shield

### Присоединяем сверху Troyka Shield и подключаем датчики к 8 и 9 цифровым контактам. В итоге получаем следующую конструкцию:



Программирование

### Далее нам нужно разобраться со средой в которой мы будем писать программу. В этом нам поможет небольшой видео курс по программированию в Mblock5 <https://www.youtube.com/watch?v=0x4kwg7aVwY&t=13s>