

КОНСТРУКТОР-РОБОТ ARDUINO ROBOT CAR С ВИДЕО-КАМЕРОЙ И WI-FI

Инструкция по сборке и первоначальной
настройке



СОДЕРЖАНИЕ

Комплектующие	3
Сборка	6
Запуск готовых примеров кода	19
Назначение комплектующих	21
Управление по Bluetooth	29
Стриминг видео по WiFi	30
Заключение	31

КОМПЛЕКТУЮЩИЕ

Для начала сделаем мини-обзор внутренностей коробки, чтобы не паниковать от такого количества деталей.

Платы



Arduino UNO



Motor Shield



WiFi Модуль

Платы - мозги робота. Каждая из плат более подробно рассмотрена в разделе “Платы”.

Двигатели



Двигатели в ответе за всё, что поворачивается и перемещается. Больше информации о двигателях вы найдете в разделе “Моторы”

Датчики



Ультразвуковой
дальномер

ИК сенсор

Датчик
линии

Bluetooth
модуль

Вольтметр

Датчики, по-другому сенсоры, собирают информацию о мире. Каждый датчик подробно рассмотрен в разделе “Датчики”

Вращающиеся механизмы



Крепление
дальномера

Камера

Башня
камеры

Пластина
крепления
башни

Помимо колёс, робот снабжён двумя вращающимися частями: башней для камеры и креплением ультразвукового дальномера. За вращение этих частей отвечают сервомоторы. Про сервомоторы можно узнать в разделе “Моторы”. Камера же здесь расположена для полноты картины. На самом деле она относится к сенсорам.

Питание



Чтобы двигатели крутились, датчики работали, а платы считали нужно электричество. Электричество робот черпает из аккумуляторов. Подробнее про питание роботов в разделе “Моторы”

Мелочь



Кроме всего вышеперечисленного есть ещё немного проводов, стоек, винтов, гаек и прочей мелочи.

СБОРКА

Сборка базовой конфигурации

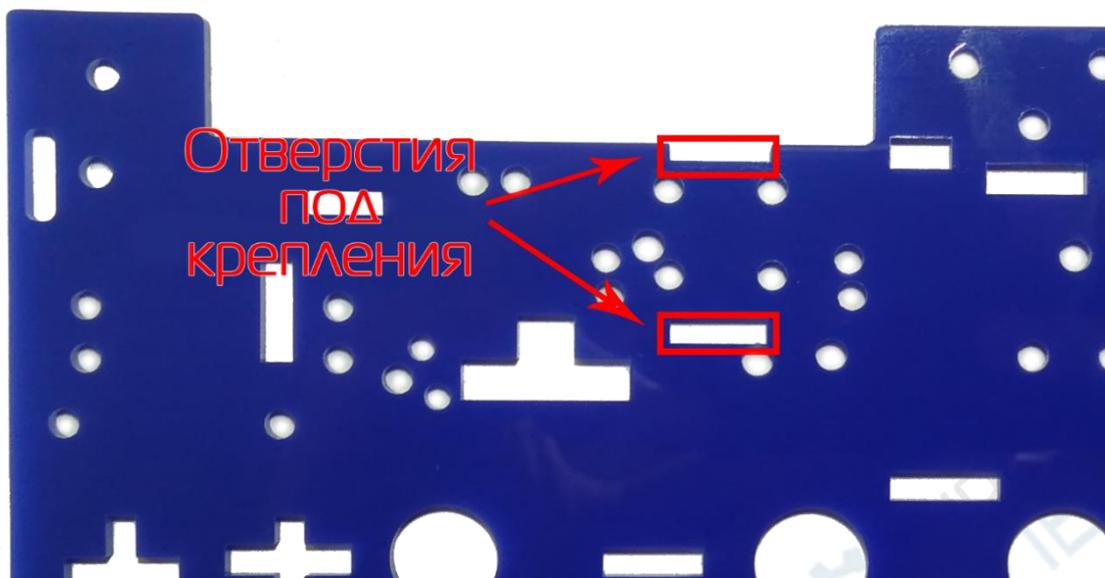
Перед сборкой немного поговорим о робототехнике. Робототехника при всей своей технической направленности является ещё и искусством. Умение продумать конструкцию робота, разместить датчики, придумать необычное применение обычным вещам требует смекалки, воображения и креативности. К чему это все? К тому, что робот-платформа имеет множество вариантов сборки. Каждый вариант уникален, все описать не представляется возможности. В связи с этим был выбран свободный вариант сборки. Вы вольны собирать робота на ваше усмотрение. С нашей стороны будут советы по сборке и базовая версия робота.

Процесс сборки любой конфигурации данного робота вращается вокруг двух платформ. К платформам крепятся все остальные детали. В базовую версию сборки робота войдут моторы, Arduino UNO с платой расширения Motor Shield, Bluetooth модуль и батареи питания.

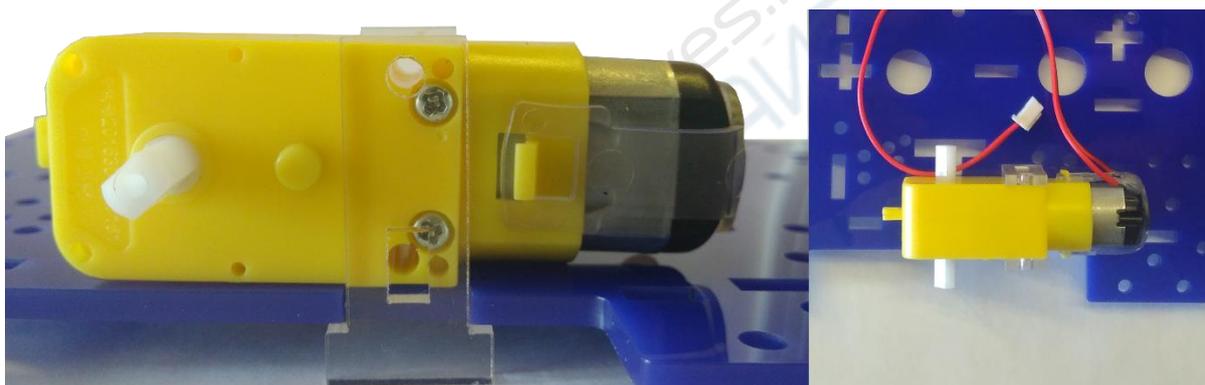
Начнём с того, что к одной из платформ прикрутим моторы. Для каждого мотора нам понадобятся:



Гайки на рисунке представлены только в этот раз. В дальнейшем подразумевается использование гаек вместе с винтами, если не оговорено иное. Крепления мотора соединяют моторы и платформу. В платформе и моторах есть специальные отверстия.

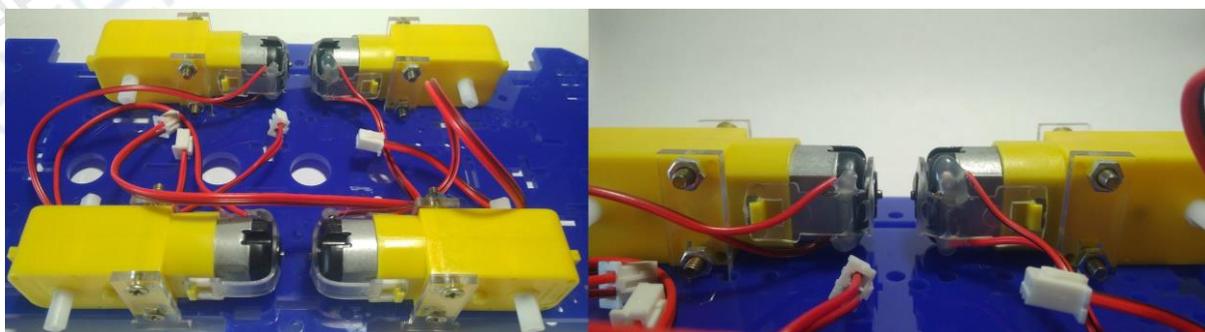


В эти отверстия вставляем крепежи мотора, между ними вставляем мотор, проводами внутрь, и стягиваем эту конструкцию винтами М3 х 25 мм.



Таким образом собираем все 4 мотора. После крепления моторов можно надеть колёса.

Однако, рекомендуем надеть колеса, когда будет собран весь робот. Так как без колес продолжить сборку удобнее.



Прикручиваем крепления для соединения двух платформ. Нам понадобятся:

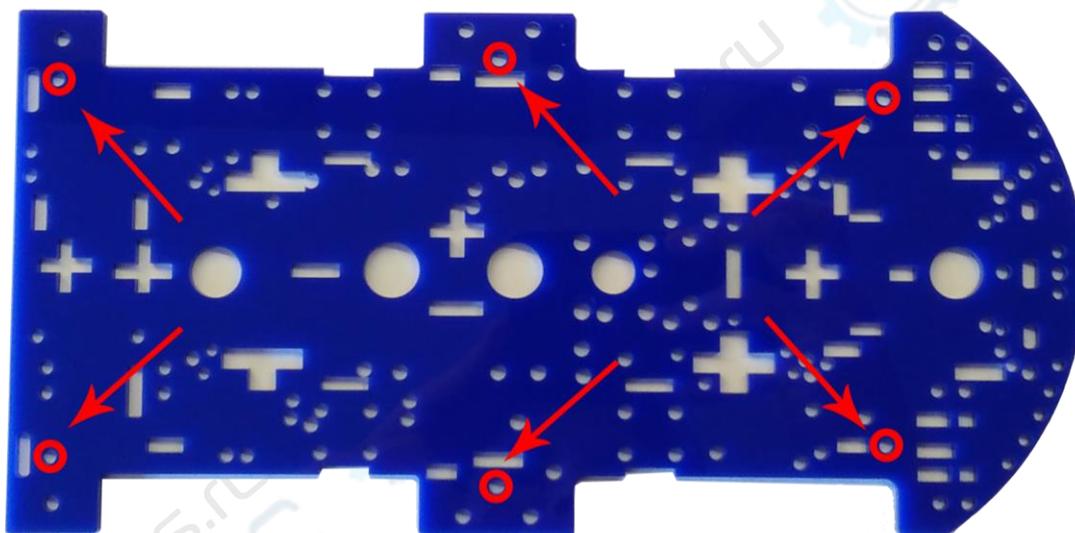


Стойки
М3 x 25 мм



Винты
М3 x 10 мм

Когда имеем дело со стойками - гайки не нужны. Винты вкручиваются в стойки. Выбираем места для крепления стоек.



Почему выбраны эти 6 точек, хотя можно взять и другие?

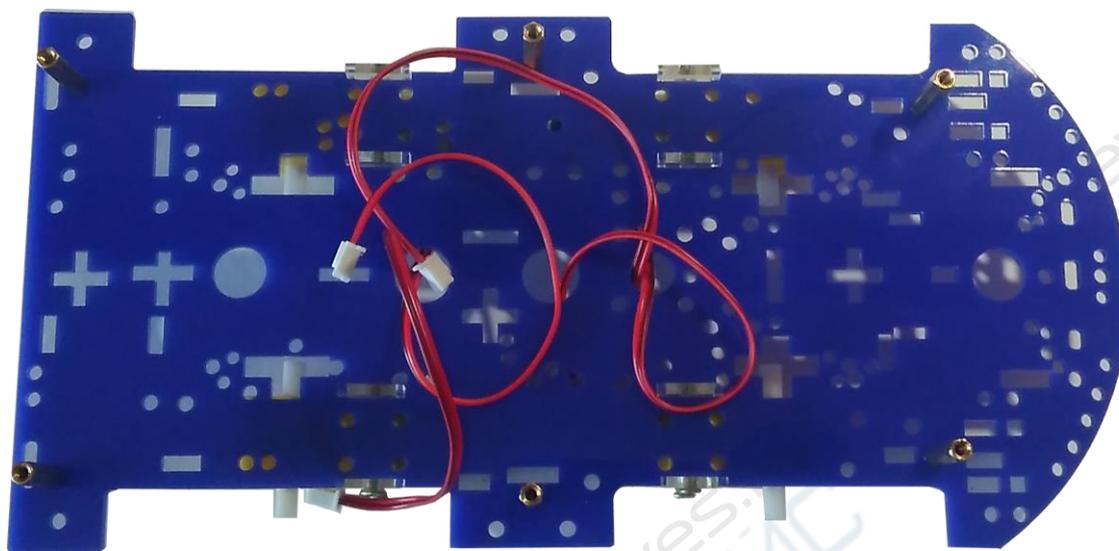
Такой вариант расположения креплений имеет ряд преимуществ перед другими.

Первое преимущество - жёсткость конструкции. Крепления разнесены достаточно далеко друг от друга, чтобы платформа сверху была прочно закреплена и не шаталась. Второе преимущество - отверстия с краю свободны для сенсоров. Большинство сенсоров будет располагаться по периметру робота, и крепиться сенсоры будут с помощью этих отверстий. Третье преимущество - середина платформы свободна. В центр платформы можно установить отсек с батареями, например.

Можно подвести итог по креплениям. Места расположения креплений выбраны исходя из соображений удобства, практичности и соответствия назначению

конструкции. Проще говоря - закрепили так, как кажется правильным. К тому же, никто не препятствует перекрутить крепления в случае необходимости.

Прикручиваем крепления в выбранные места с помощью винтов, просовываем провода моторов через круглые отверстия.



Кстати, между платформами можно что-нибудь закрепить, например, батарейный отсек. К нему будет не так просто добраться, чтобы поменять аккумуляторы, но зато он будет лучше защищён. В нашем примере батарейный отсек закреплен сверху - для удобства.

Все остальные элементы было решено прикрепить на другую платформу, поэтому ее собираем отдельно. Сначала прикрутим Arduino UNO к платформе. Нам понадобятся:

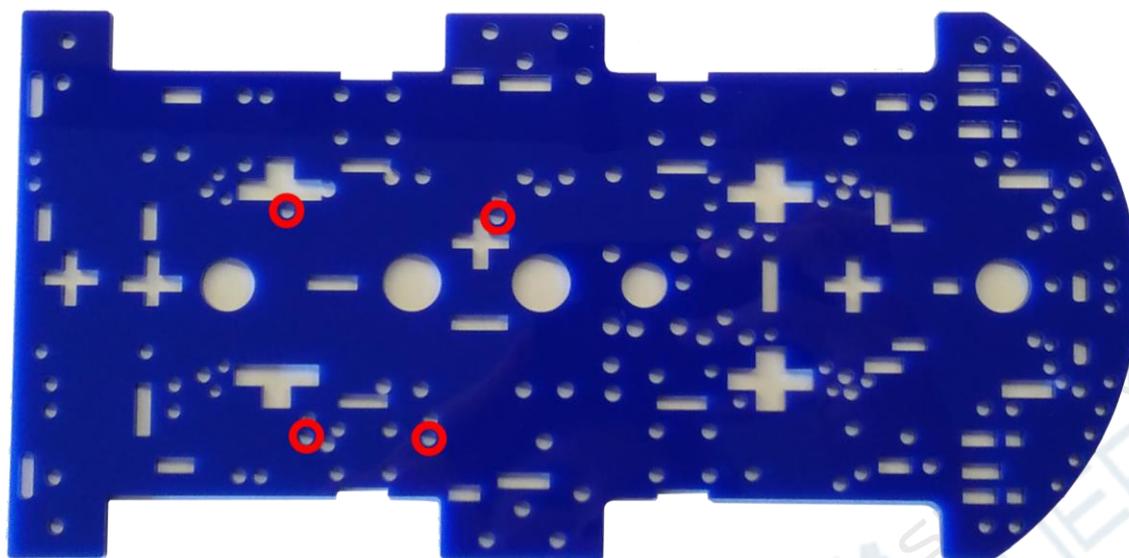


Стойки
М3 x 10 мм

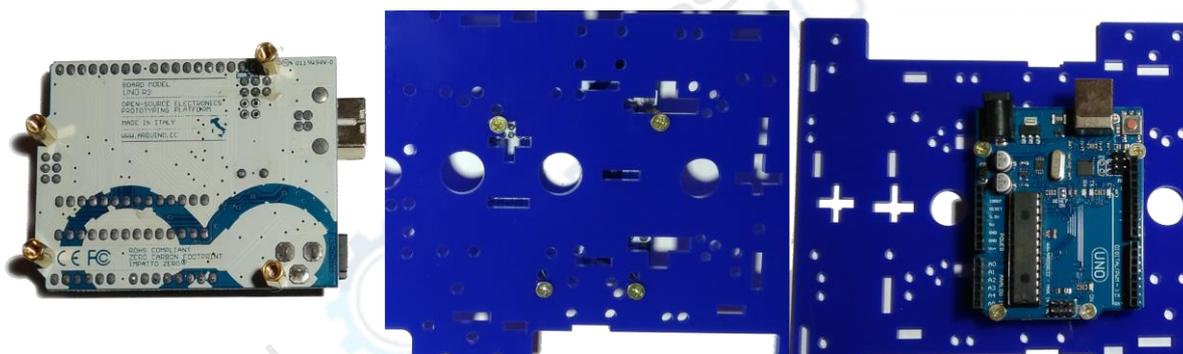


Винты
М3 x 8 мм

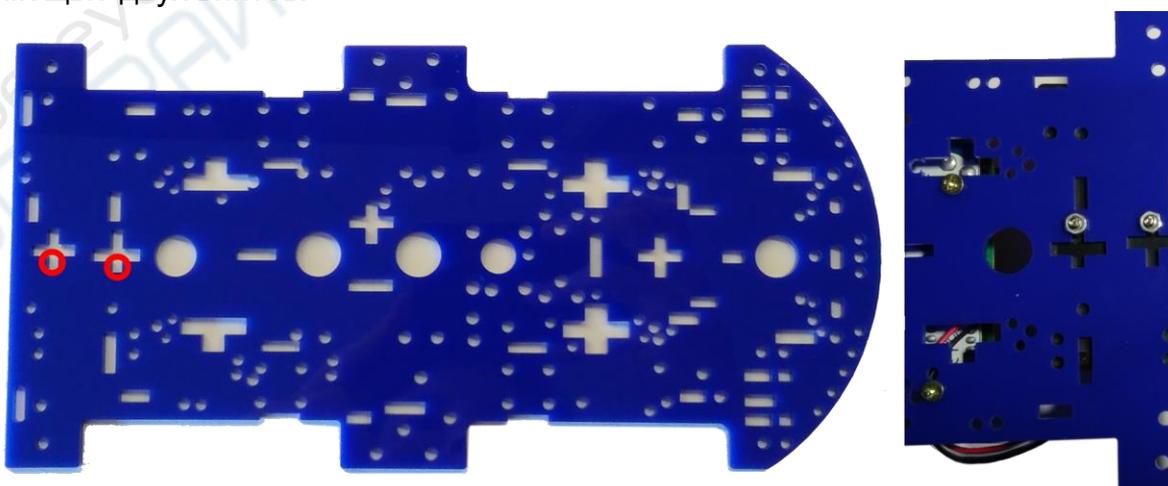
Отверстия под крепление Arduino.



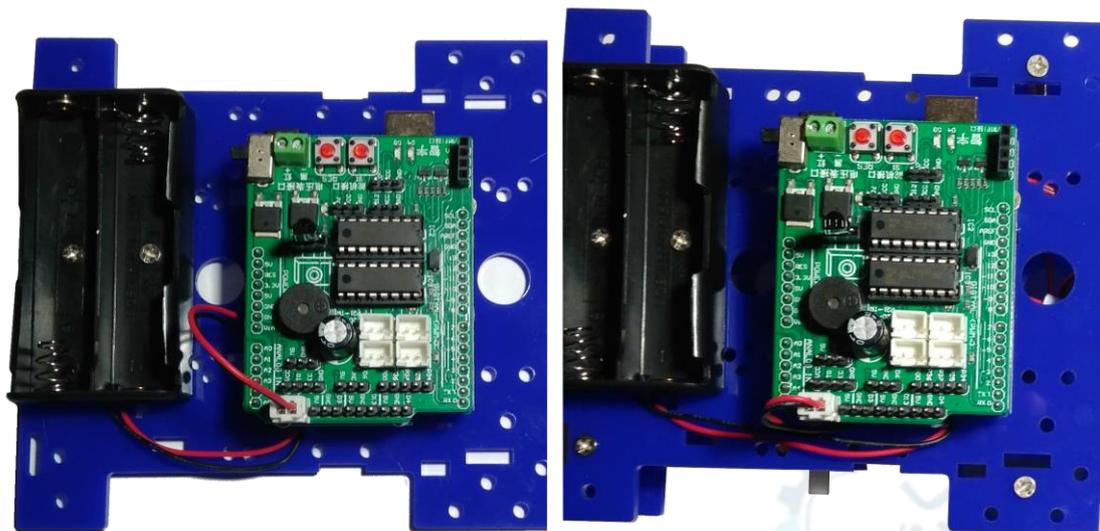
Закрепляем стойки на Arduino и соединяем плату с платформой с помощью винтов. На Arduino можно насадить Motor Shield.



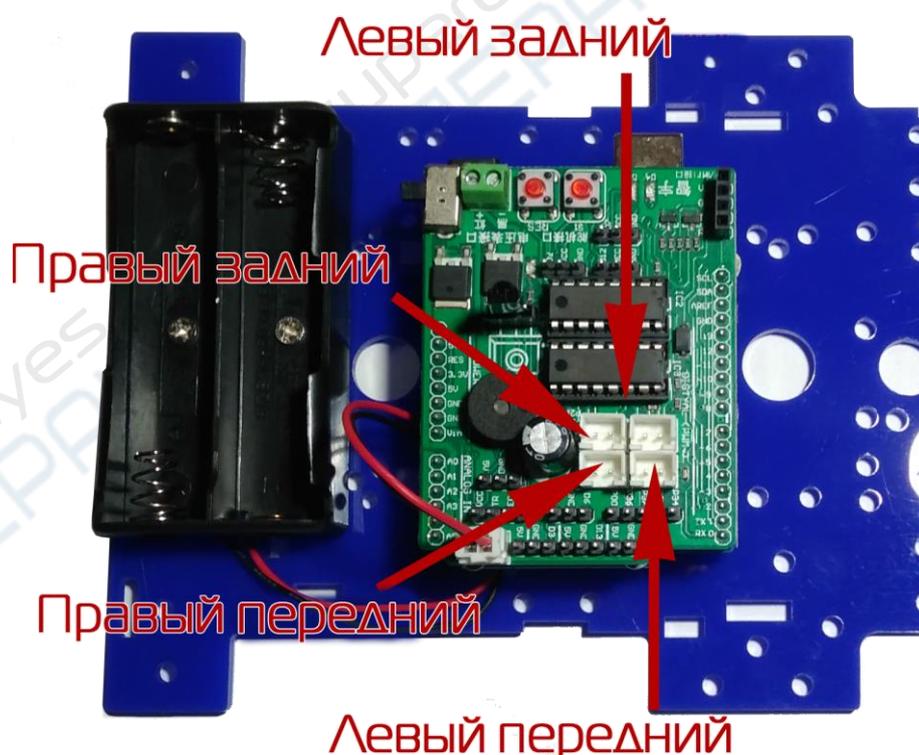
Теперь установим отсек для батареек. Крепиться он будет в эти отверстия с помощью двух винтов.



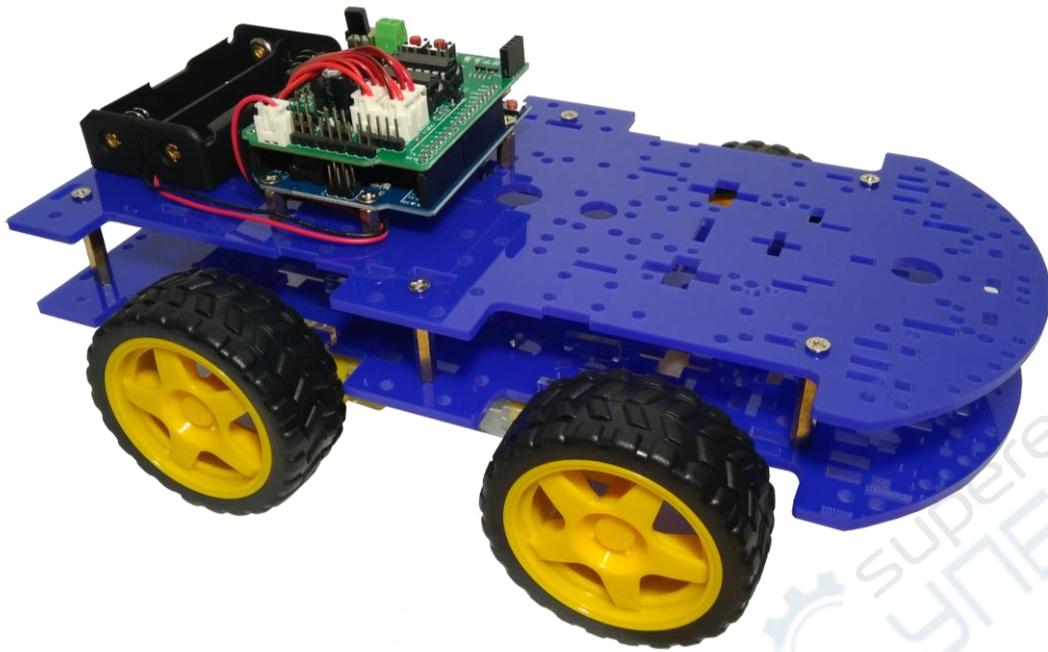
Соединяем две платформы с помощью креплений винтов. В процессе соединения не забываем протянуть провода моторов через круглые отверстия - потом будет сложнее.



После этого подключаем провода. Вот схема какой мотор подключать куда. На самом деле конкретные места подключения моторов не важны. В программе можно без проблем переназначить номера подключённых моторов.



Базовая модель собрана. Уже можно идти программировать. Далее будут приведены инструкции по подключению датчиков и сборке остальных частей. В базовой конструкции дальнейшие шаги можно пропустить.

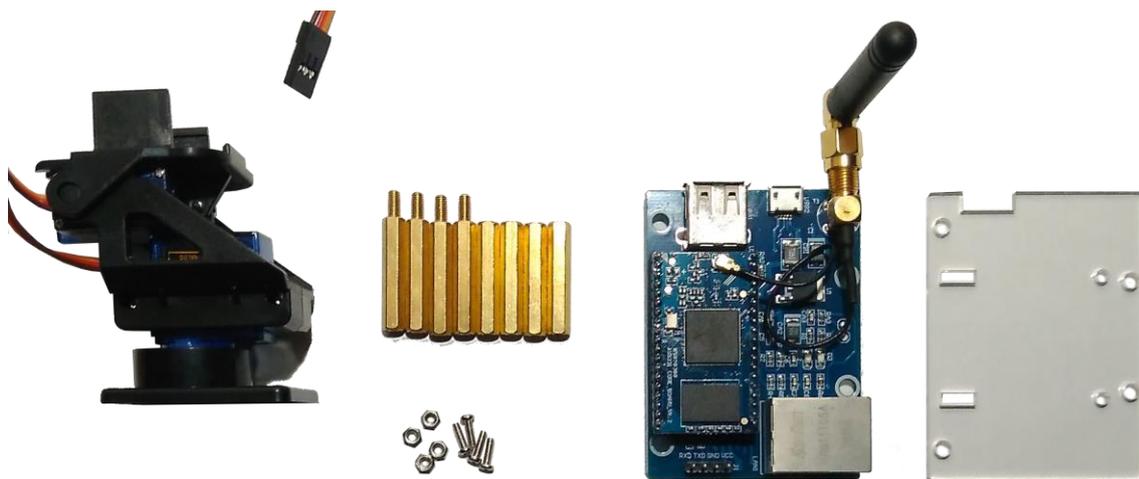


Ещё можно подключить Bluetooth модуль, чтобы управлять роботом с телефона.

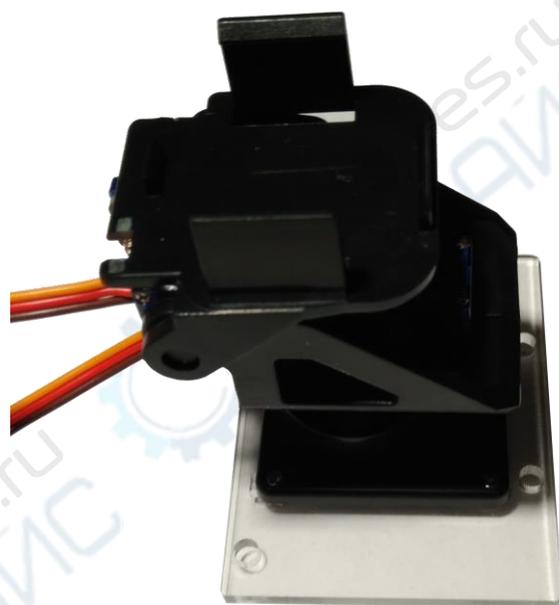


Сборка башни с камерой

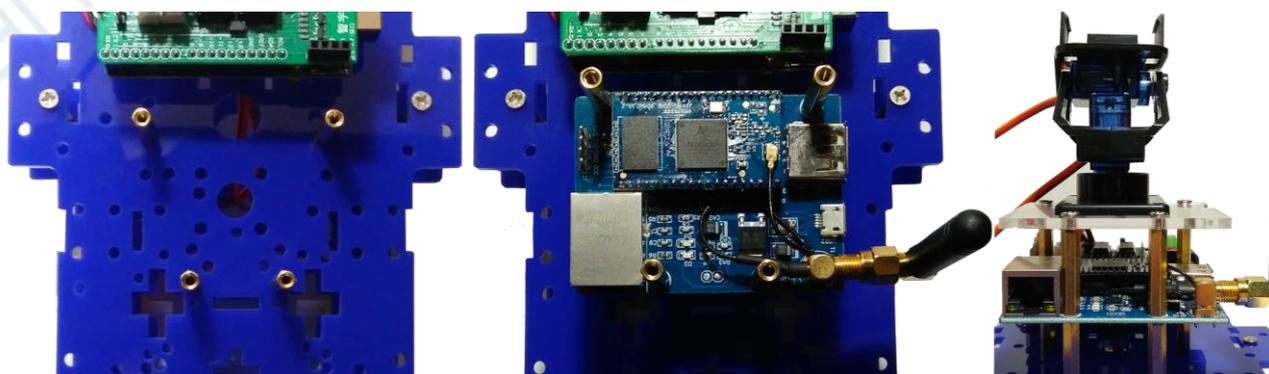
Нам понадобятся:



Соединяем башню и пластину таким образом, чтобы задняя часть крепления смотрела в сторону отверстий с большим расстоянием на пластине. Можно и наоборот - зависит от того, куда камера будет смотреть. Представленное положение подходит, если камера будет в основном смотреть вперёд и вверх.



После этого крепим стойки к платформе. Сверху крепим WiFi модуль и стягиваем его стойками с резьбой. Сверху крепим пластину с башней.

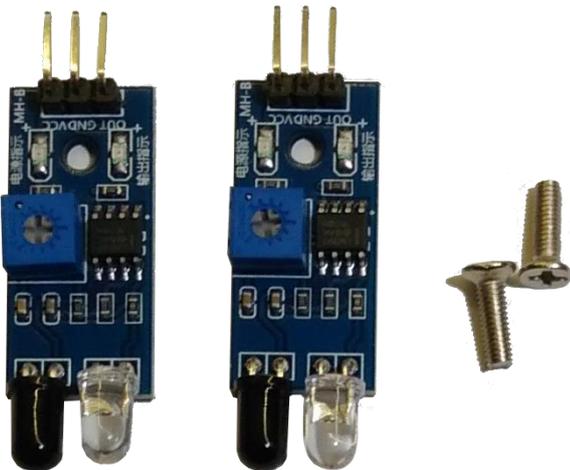


На конце башни защёлкиваем камеру, проводим провода до платы и вставляем их в разъемы для сервомоторов. WiFi модуль соединяем с Arduino через шлейф Папа-Мама с 4 проводами.



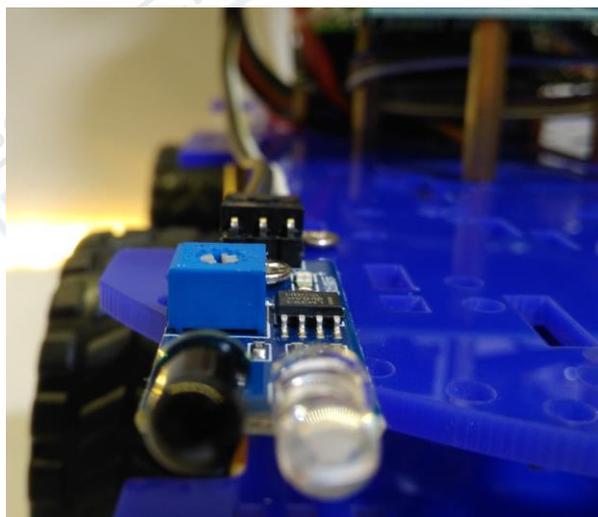
Подсоединение ИК датчиков

Нам понадобятся:



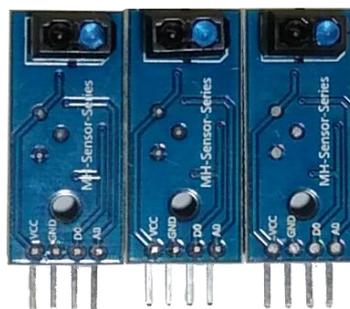
Прикручиваем куда заблагорассудится - отверстий много. Исходим из предназначения датчиков. В примере прикручено по бокам верхней платформы, чтобы можно было отслеживать препятствия перед роботом. Подсоединяем каждый датчик шлейфами из 3 проводов Мама-Мама.

Контакты ИК датчиков

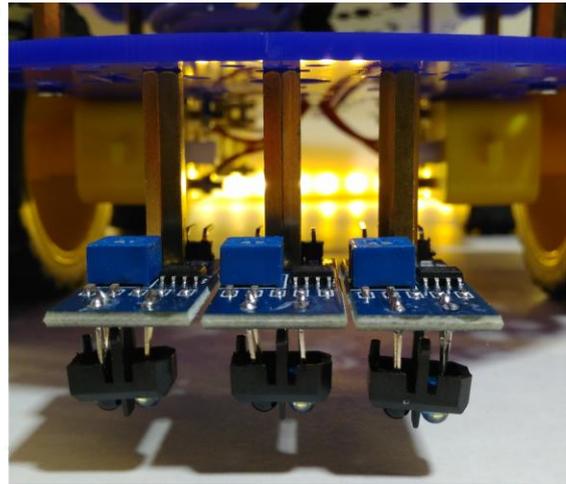
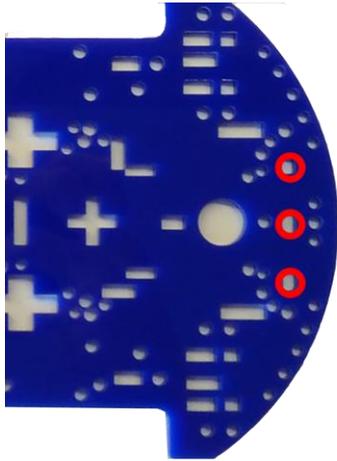


Подсоединение датчиков линии

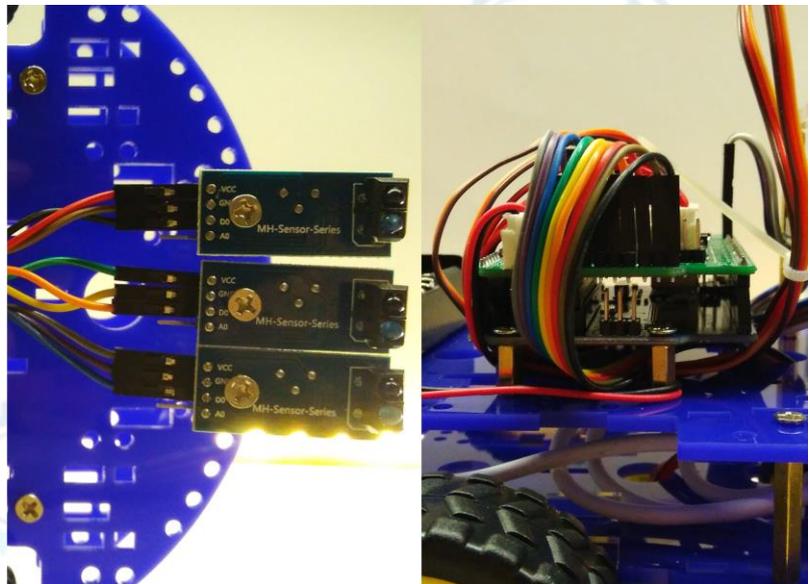
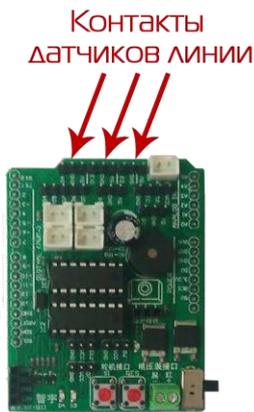
Нам понадобятся:



Накручиваем стойки на каждый датчик и прикручиваем их к данным отверстиям на платформе.

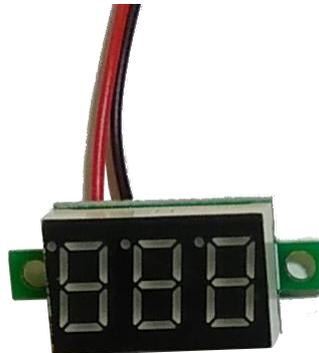


После этого подключаем большой шлейф, состоящий из трёх трёхпроводных шлейфов.

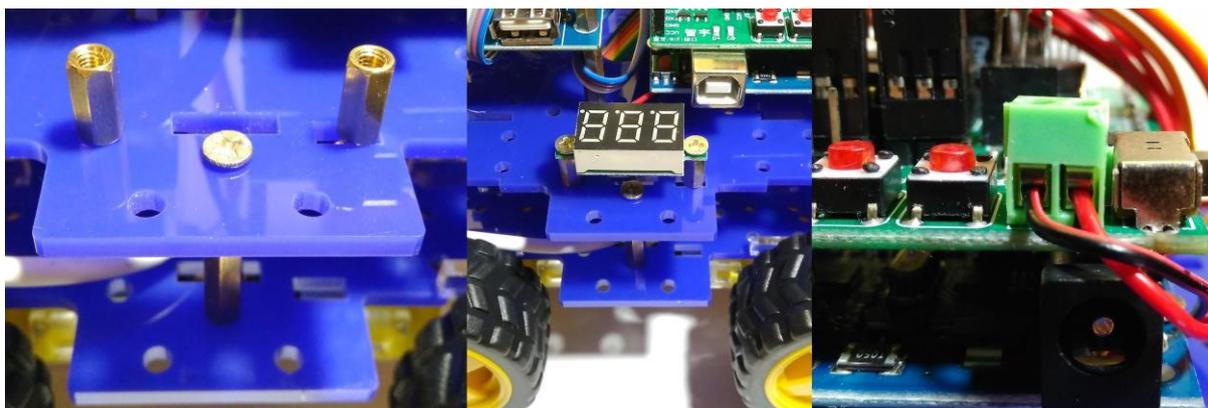


Подсоединение Вольтметра

Нам понадобятся:

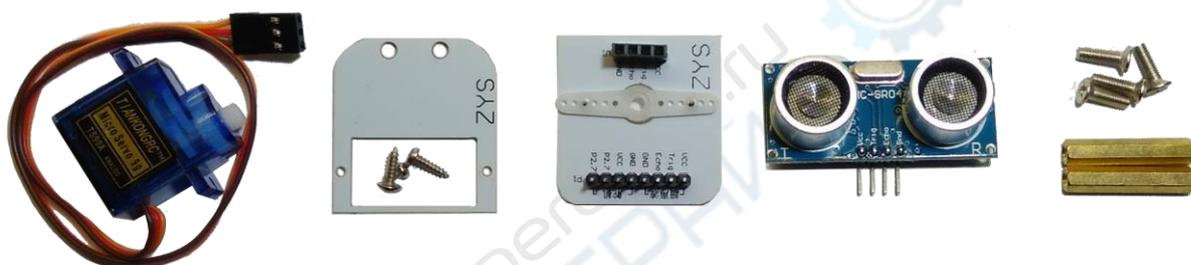


С помощью стоек крепим вольтметр к платформе. Красный провод подсоединяем к "+" клеммника, чёрный - к "-".

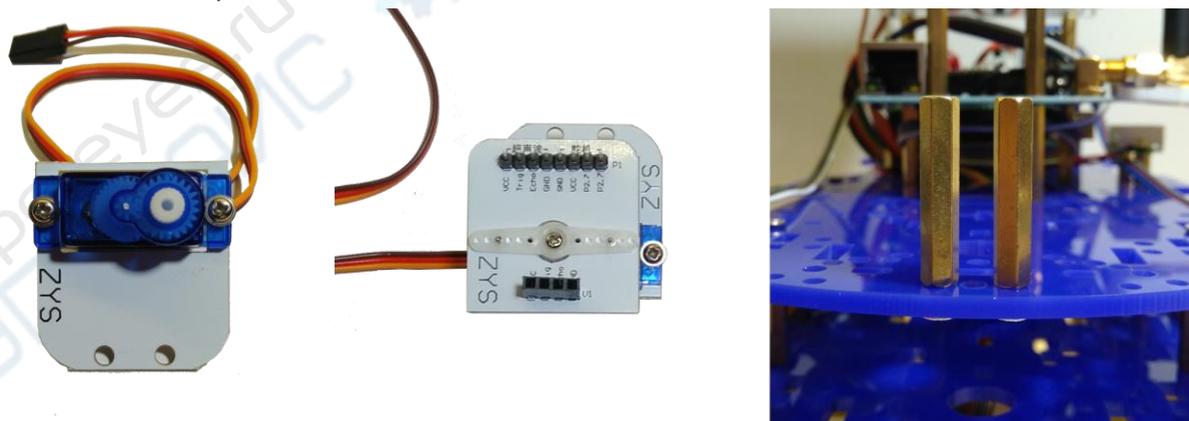


Сборка крепления ультразвукового дальномера

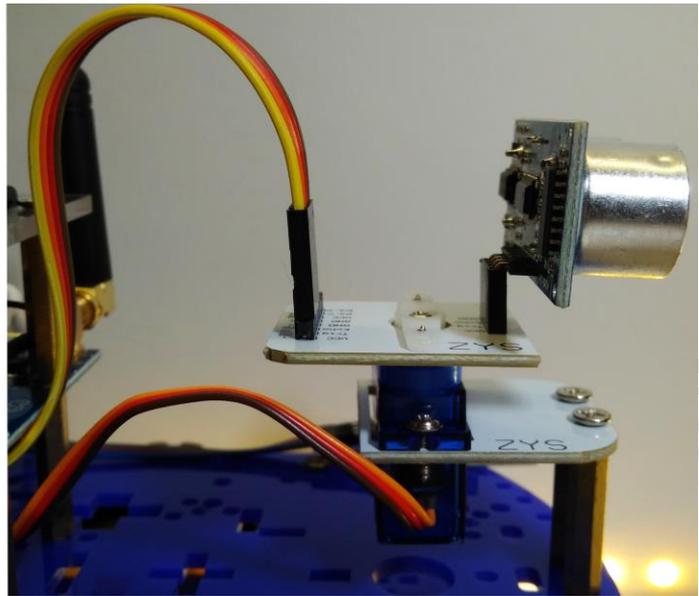
Нам понадобятся:



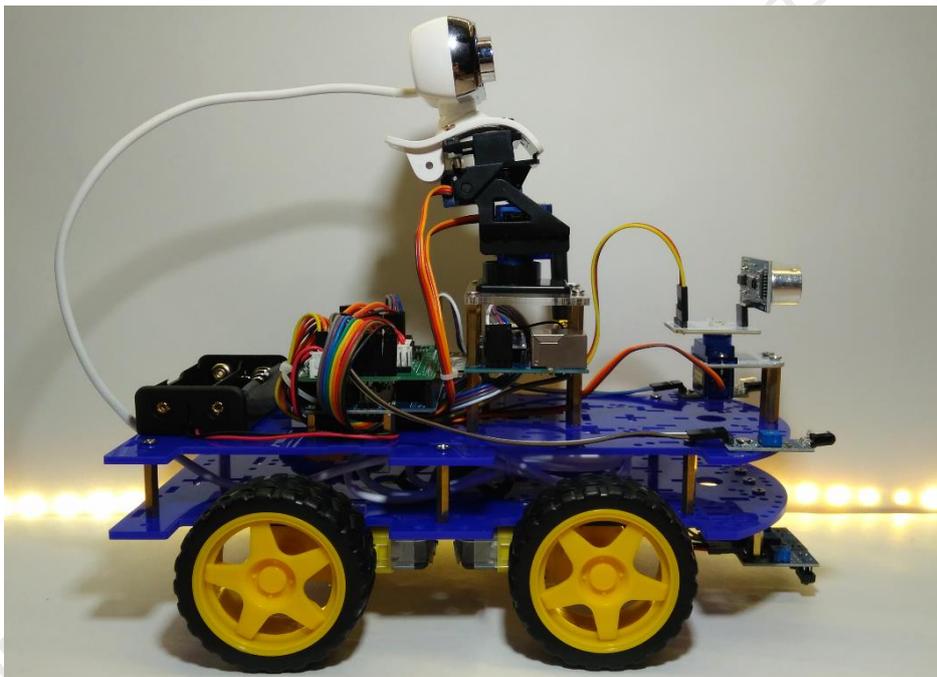
Прикручиваем сервомотор к рамке, на сервомотор надеваем крепление для ультразвукового дальномера. В отверстия на носу платформы вкручиваем стойки с помощью винтов.

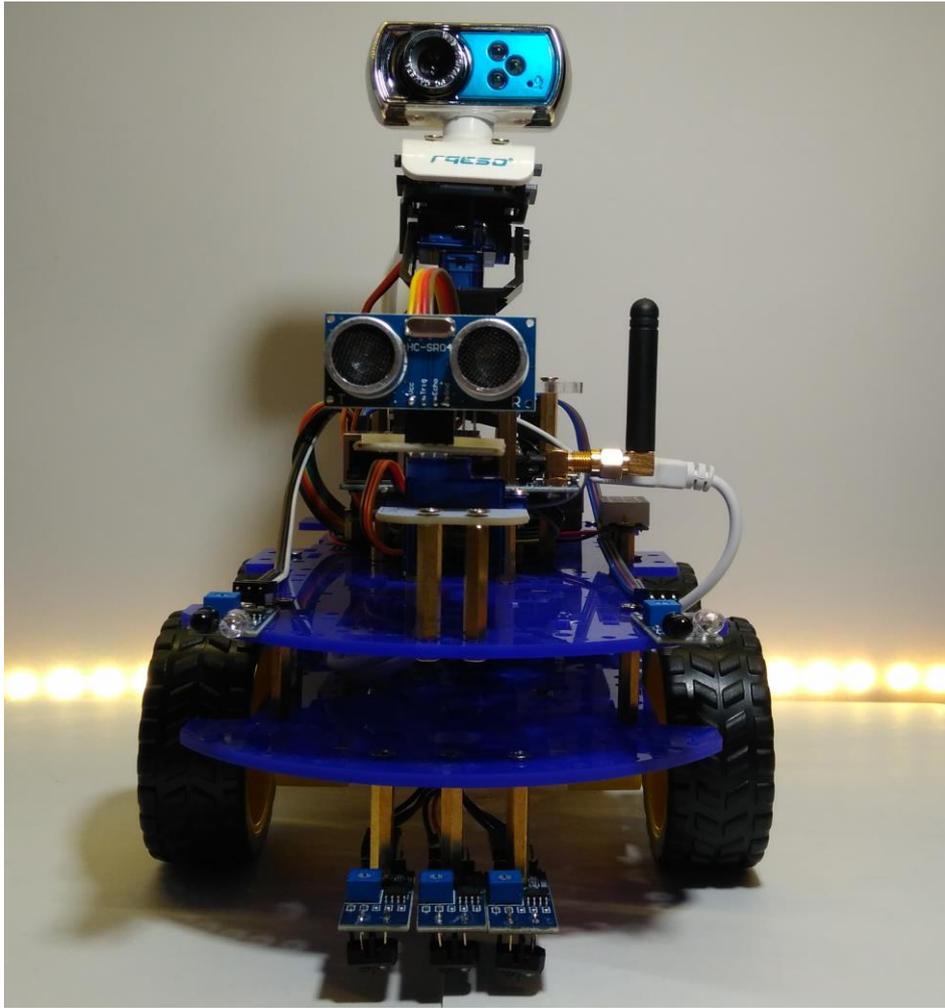


Протягиваем кабели. Для дальномера нужен 4 проводной кабель Мама-Мама.



Внешний вид полностью собранного робота:





supereyes.ru
СУПЕРПАЙС

supereyes.ru
СУПЕРПАЙС

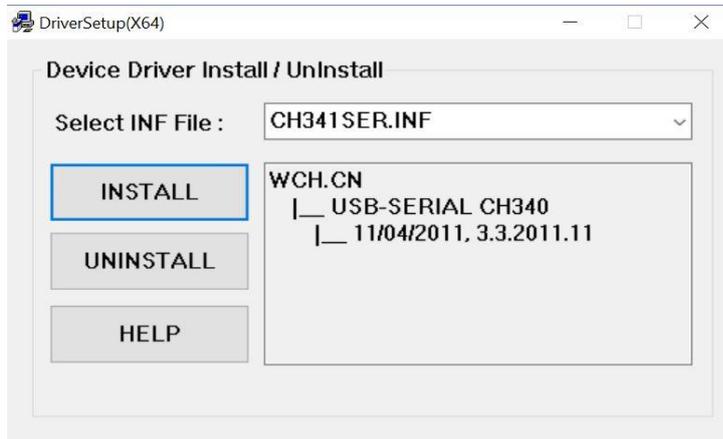
ЗАПУСК ГОТОВЫХ ПРИМЕРОВ КОДА

Подключение Arduino UNO.

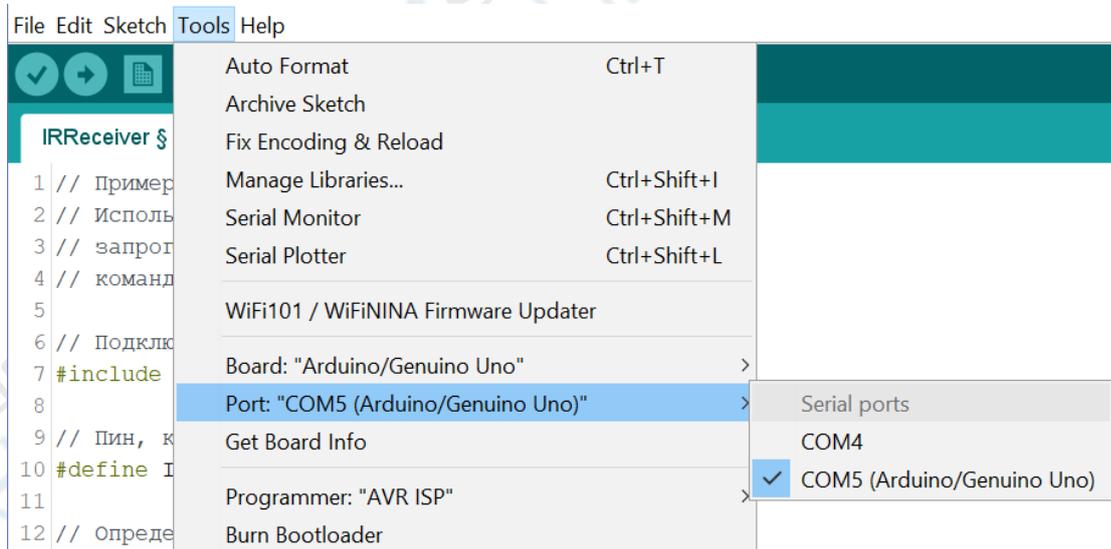
Для начала:

Скачиваем интегрированную среду разработки [Arduino IDE](#).

После этого скачиваем и устанавливаем [драйвер](#) для Arduino.



Подключаем Arduino UNO к компьютеру, открываем Arduino IDE. Проверяем подключение Arduino UNO. Для этого открываем вкладку Tools, в ней находим пункт Port, в Port должна находиться Arduino UNO. Не обязательно будет написано COM*(Arduino/Genuino Uno). Вполне возможно будет просто COM*. В пункте Board выбираем Arduino/Genuino Uno, если выбрано что-то другое.



Если в Port ничего нет, то нужно проверить нашлась ли Arduino в системе. Открываем менеджер устройств. В Windows 10, например, его можно найти в поиске вбив "Device Manager". Во вкладке Порты и COM устройства в менеджере устройств находим Serial Device - это и есть наша Arduino. Если найти Arduino не получилось, проверяем физическое подключение, пробуем другой USB порт или кабель, переустанавливаем драйвер.

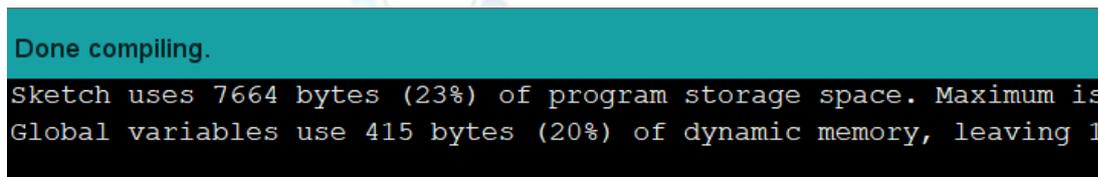
- > Network adapters
- > Other devices
- ▼ Ports (COM & LPT)
 - USB Serial Device (COM5)
- > Print queues
- > Processors
- > Security devices

После удачного подключения платы открываем программу по выбору. Кнопка с галочкой “Verify” компилирует код, но не загружает его на плату. Кнопка со стрелочкой “Upload” сначала компилирует, а потом загружает его в Arduino. Загружаем код на плату с помощью кнопки Upload.

File Edit Sketch Tools Help



О текущем статусе загрузки программы можно узнать из панели внизу.



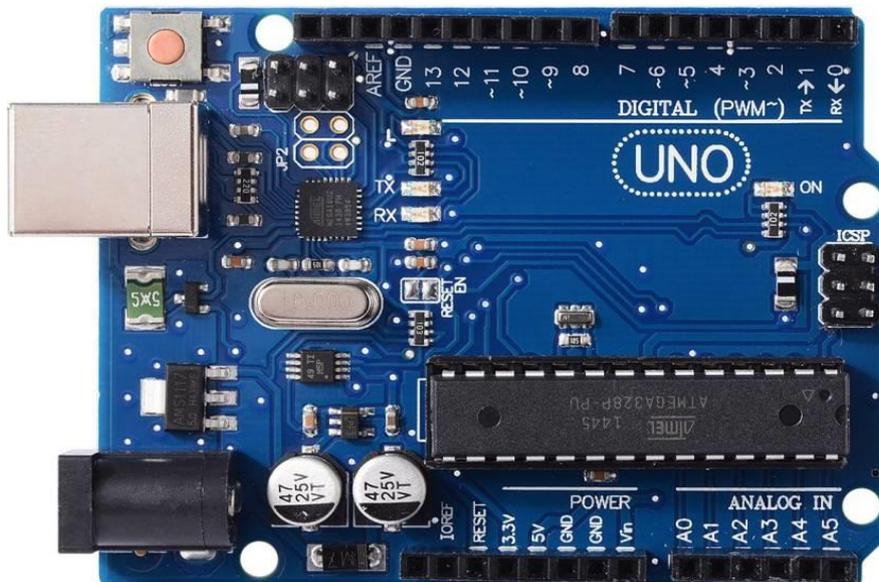
Интерфейс Arduino IDE можно перевести на русский в настройках. Настройки находятся во вкладке File -> Preferences

НАЗНАЧЕНИЕ КОМПЛЕКТУЮЩИХ

Платы

В этом разделе можно подробнее узнать про каждую из плат, входящих в состав комплекта.

Arduino UNO

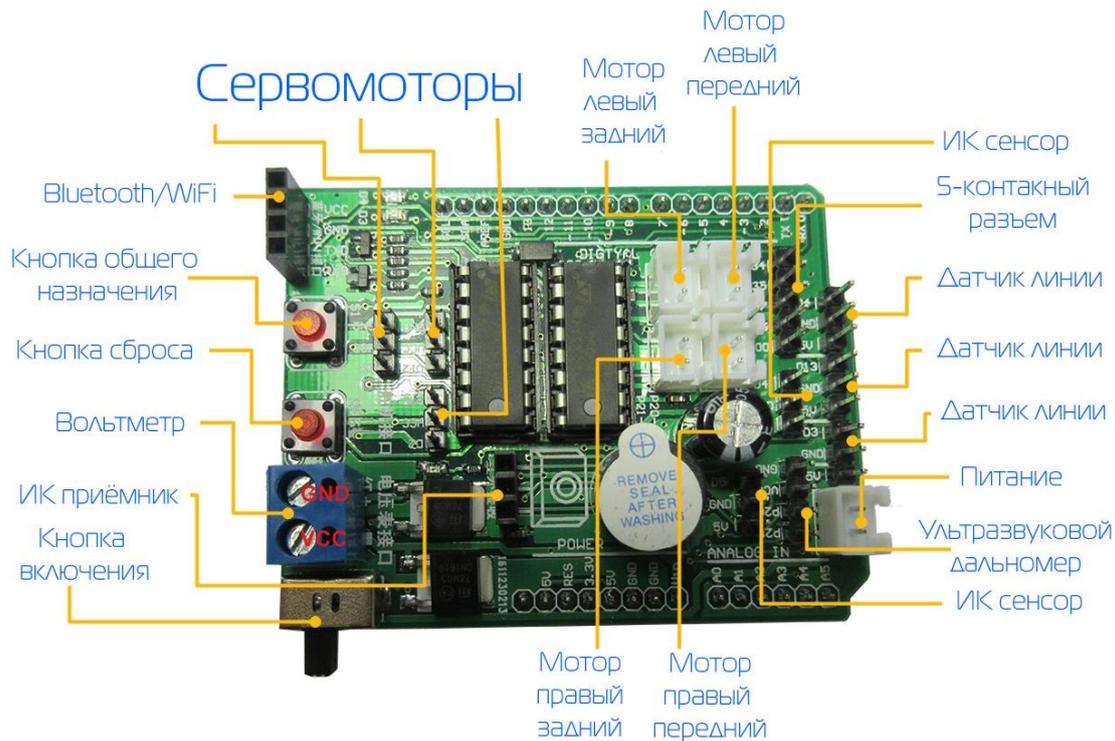


Роботом управляет микроконтроллер Arduino UNO. Специализация Arduino UNO - включать и выключать питание в нужный момент. 14 цифровых и 6 аналоговых контактов, или пинов по-другому, работают на благо пользователя в соответствии с программой.

В нашем случае Arduino UNO играет роль управляющего. UNO опрашивает датчики о состоянии окружения, слушает команды пользователя с Bluetooth или WiFi и отдаёт команды моторам и сенсорам.

Motor Shield

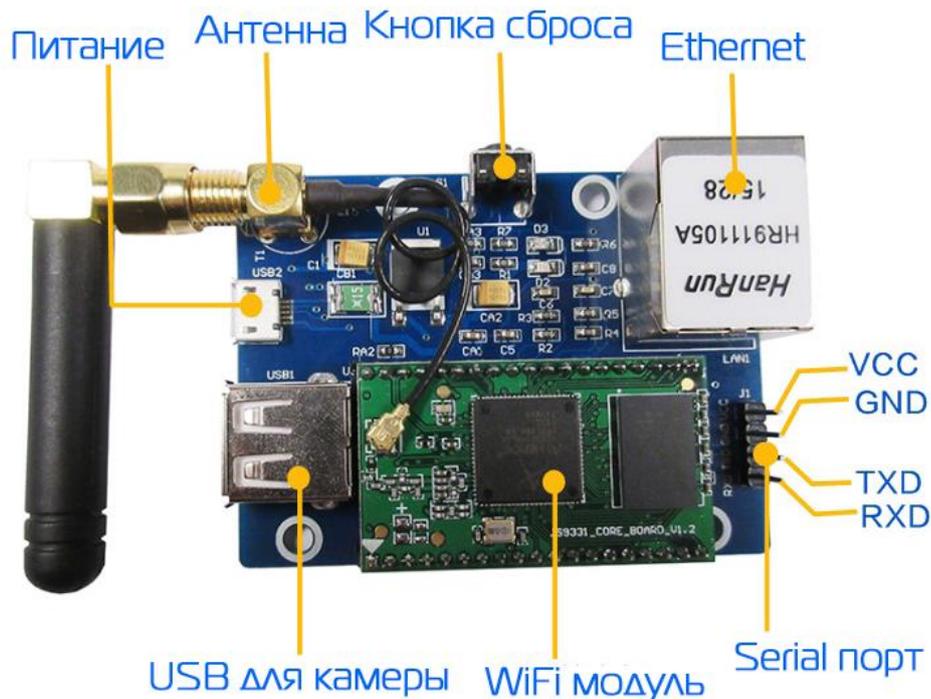
Для Arduino существуют так называемые шилды, или платы расширения. Плата расширения - это печатная плата, которая выполняет какую-нибудь функцию. У нас в распоряжении есть Motor Shield. Из названия можно предположить, что эта штука как-то связана с моторами. Действительно, Motor Shield берёт на себя обязанности по обслуживанию моторов, а также предоставляет несколько дополнительных функций. Подключение моторов к Arduino UNO требует дополнительных деталей. Не получится просто подключить мотор как светодиод. Первая проблема - в потреблении тока. Первая проблема рассматривается в разделе "Моторы". Вторая проблема - для робота-машинки недостаточно одного направления вращения колёс. Motor Shield решает эти проблемы с помощью драйверов двигателей L293D и нескольких электронных компонентов типа конденсаторов и резисторов.



Наш Motor Shield объединяет в себе функции драйвера двигателей и платы подключения сенсоров. Все необходимые разъемы выведены наружу, остаётся только подключить контакты. Обратите внимание: нажатие кнопки сброса останавливает бесконечное вращение моторов при подаче питания, после чего вы можете приступить к программированию контроллера Arduino UNO.

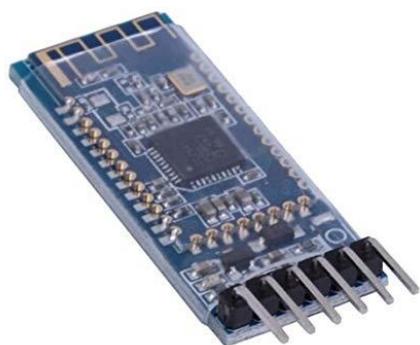
Wi-Fi модуль

С помощью этого модуля можно подсоединить робота к WiFi. Имея доступ к WiFi можно передавать видео и управлять роботом в реальном времени.



Bluetooth модуль

Bluetooth модуль даёт возможность принимать и отправлять данные по Bluetooth. Подключаем его к Arduino UNO. Теперь данные, которые мы отсылаем и принимаем в Serial мониторе будут транслироваться по Bluetooth.



Датчики

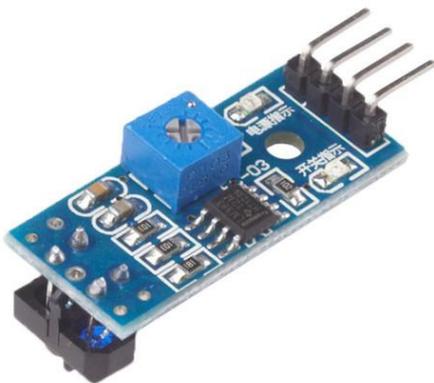
Датчики помогают роботу исследовать мир вокруг. Существует огромное количество различных датчиков. Основная суть в том, что датчики извлекают и передают информацию, которую потом можно обработать и извлечь пользу. Датчики можно разделить на цифровые и аналоговые. Цифровой датчик может ответить на простой(да-нет) вопрос. Например, цифровой датчик света около включенной лампы может ответить на вопрос “Есть свет?”. Ответом будет “Да”. Аналоговый датчик света может ответить на тот же вопрос по-другому. Он может сказать “Лампочка горит на 70% от максимума”. В общем, цифровые датчики передают 0 или 1, а аналоговые - диапазон значений.

Обычно у датчиков три контакта - Vcc, Gnd и S. Vcc - это питание датчика. На Vcc подаём 5В с Arduino. Gnd - земля. К Gnd подсоединяем землю с Arduino. S(сигнал) - выход датчика с полезной информацией. В зависимости от датчика, подключаем его к аналоговому или цифровому пину Arduino. Ещё бывают более сложные датчики типа ультразвукового дальномера или камеры. Таким датчикам может не хватить одной сигнальной линии, чтобы передать всю информацию. В этом случае придумывают более сложные способы передачи информации: через несколько проводов или с помощью протоколов. Производитель датчика обычно описывает способ считывания информации в инструкции к датчику (Datasheet). Подключая новый датчик, полезно ознакомиться с даташитом к нему, чтобы избежать всевозможных неприятностей, да и просто быстрее разобраться что к чему.

Немного подробнее о нашем случае. На роботе есть несколько цифровых и аналоговых датчиков. Данные с датчиков, кроме камеры, передаются на Arduino UNO и там обрабатываются. Данные представляют собой уровни напряжения, обычно от 0 до 5 вольт. Arduino преобразует напряжение в сигнал. На Arduino UNO стоит 10 битный преобразователь входного напряжения. Это

означает то, что Arduino UNO способна различать 1024 уровней напряжения между 0 и 5 Вольт. Аналоговые датчики на выходе дают сигнал между 0 и 5 Вольт, а Arduino UNO преобразует их в диапазон от 0 до 1023. Цифровые датчики также выдают сигнал от 0 до 5 Вольт, но обычно не передают промежуточные значения типа 2.6 Вольт. Идея в том, чтобы любое напряжение выше какого-нибудь порога, считать высоким (HIGH) уровнем, а напряжение ниже этого порога - низким. Возникает вопрос - а что если напряжение будет как раз на пороге? Ответ зависит от ситуации. Порог - это всего лишь договорённость. Договорённость можно и поменять, однако она играет роль при работе с устройствами, которые сделали другие люди. Arduino UNO имеет порог в 1.5V и если цифровой датчик выдаёт 1.5 В, то сигнал будет осциллировать между HIGH и LOW так как не получится идеально держать 1.5 В в каждый момент времени. Вывод - чем сильнее отличаются HIGH и LOW выходы цифрового датчика, тем он надёжнее.

Датчик линии

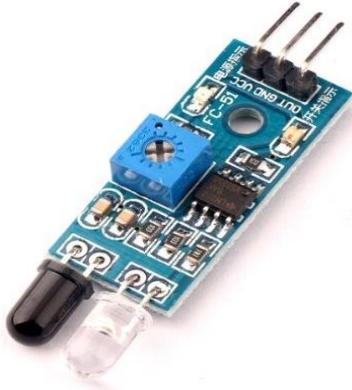


Датчик линии помогает определить наличие линии под ним. Работает он на принципе отражения света. Когда на предмет падает луч света, одну часть света он отражает, другую - поглощает. Яблоко красное потому, что оно отразило красную часть(спектр) света и поглотило все остальные. В датчике установлено два светодиода. Один испускает свет, второй - принимает. Когда перед датчиком линии тёмный цвет, количество света, принимаемого вторым диодом меньше. Этим датчик пользуется и выдаёт различное напряжение в зависимости от количества принятого света.

В этом наборе датчик может работать как в аналоговом, так и в цифровом режиме. У каждого датчика есть 4 пина: питание Vcc, земля Gnd, цифровой выход DO и аналоговый выход AO. Подключаем Vcc и Gnd к соответствующим пинам на Arduino UNO, подключаем AO или DO по ситуации.

Пример с кодом LineSensor вы можете скачать [здесь](#).

ИК датчик препятствий



Позволяет определить наличие препятствия перед сенсором. Работает на принципе отражения света. Если перед датчиком есть препятствие, то света, излучаемого светодиодом, обратно отражается больше. Датчик цифровой. Потенциометром можно регулировать дистанцию срабатывания.

Подключаем Vcc и Gnd к соответствующим пинам на Arduino UNO, подключаем OUT к одному из цифровых пинов.

Пример с кодом IRDistanceSensor вы можете скачать [здесь](#).

ИК ресивер + пульт



ИК (инфракрасный канал) связь – это беспроводная связь, которую мы применяем для передачи данных в прямой видимости и на небольшой скорости. Переключая каналы телевизора, мы пользуемся этой самой ИК связью. У каждого ИК пульта есть ИК светодиод, который включаясь и выключаясь посылает информацию о нажатой кнопке. Эту информацию можно принять и обработать с помощью ИК детектора.

Инфракрасный диапазон не виден глазу, но виден камере. Если хочется увидеть, что же всё-таки передаётся - наведите камеру на светодиод пульта. Этот способ поможет узнать самую частую причину того, что телевизор не переключает каналы – сели батарейки.

Подключаем ИК ресивер как цифровой датчик. Проблема в том, что данные с ресивера вручную считать достаточно сложно, но нам повезло - проблема решена. Есть классная библиотека. Называется IRemote. Скачать ее можно [ТУТ](#) или через

мастер библиотек в Arduino IDE. После установки библиотеки запускаем пример с кодом.

Пример IRReceiver вы можете скачать [здесь](#).

Можно побегать по дому и разыскать всякие пульты, научиться считывать их показания и использовать для управления роботом. Также можно воспользоваться пультом из комплекта.

Ультразвуковой дальномер



Нужно узнать расстояние до чего-нибудь в пределах 4 метров? Ультразвуковой дальномер идёт на помощь. Работает по принципу “летучей мыши”. Он посылает ультразвуковую волну и считает время, за которое волна возвратится. Зная скорость звука и время, за которое волна пришла обратно, можно рассчитать расстояние до объекта.

Датчик имеет 4 вывода:

Vcc – На этот контакт подается питание в 5В.

Trig – На этот контакт нужно подать логическую единицу на 10 мкс, чтобы дальномер испустил ультразвуковую волну.

Echo – После того, как ультразвуковая волна вернется обратно, на этот контакт будет подана логическая единица на время, пропорциональное расстоянию до объекта

Gnd – Этот контакт подключается к земле.

Пример с кодом UltrasonicDistanceSensor вы можете скачать [здесь](#).

Вольтметр



Нужно измерить заряд аккумулятора? Не проблема. Подсоединяем вольтметр к клеммнику на Motor Shield и наблюдаем за дисплеем. Никаких дополнительных действий не требуется.

Камера



Камера тоже датчик, только более сложный. Такой сложный, что Arduino UNO не может эффективно работать с камерой. К UNO камеру так просто не подключить, а работе с изображениями люди посвящают карьеры. Компьютерное зрение называется, кому интересно.

Нам камера нужна, чтобы следить за обстановкой вокруг робота. Эта задача вполне выполнима с помощью WiFi модуля, который передаст видео с камеры на устройство, с которого мы управляем роботом.

Моторы

В распоряжении имеются два вида моторов.

DC мотор



Мотор постоянного тока (DC мотор) - знакомый всем моторчик из игрушек. На клеммы подаётся напряжение - вал крутится. Разные уровни напряжения - разная скорость. У наших моторов есть понижающий редуктор. Редуктор - это набор шестерёнок, который преобразует мощность мотора в полезную работу. Для применения в качестве моторов для колёс количество оборотов в минуту у наших моторов очень

велико, а крутящий момент слаб. Понижающий редуктор позволяет снизить обороты и поднять крутящий момент.

Сервомотор



Сервомотор - это DC мотор с примочками. К обычному мотору прикреплен микросхема с потенциометром. Такая обвязка позволяет контролировать угол поворота мотора. Сервомотору можно дать команду встать в любой угол из возможного интервала. У наших сервомоторов интервал от 0 до 180 градусов.

Питание

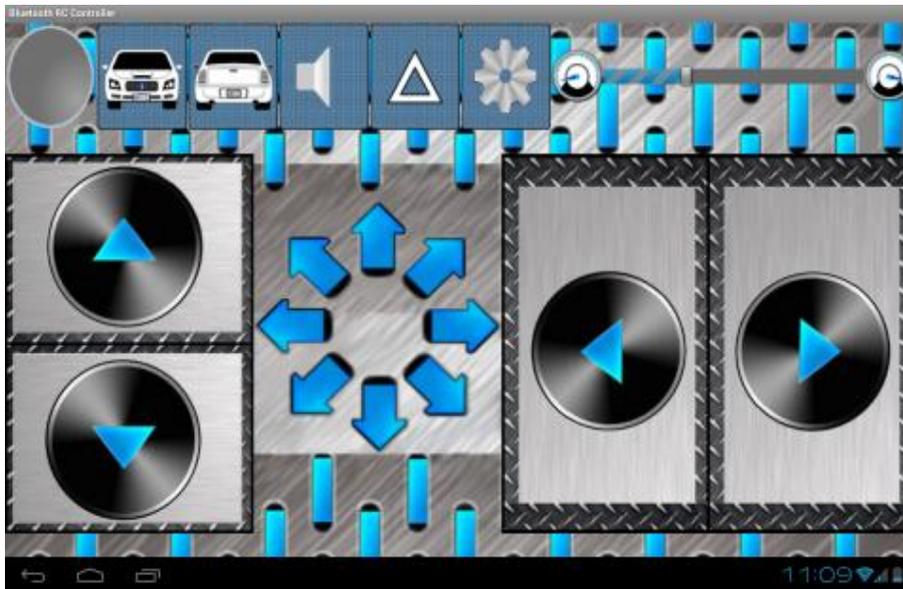
Ток, который поддерживает Arduino на своих выходах до 40 мА. Моторчик без нагрузки потребляет 80 мА, а при блокировке 1600 мА (для разных моторчиков потребляемый ток может варьироваться), что определенно больше максимальных 40 мА. Поэтому, если моторчик просто подключить к пину Arduino, то велик шанс, в лучшем случае повредить мотор или сжечь пин, в худшем – вывести из строя микроконтроллер.

Подключать моторы нужно с помощью драйверов моторов. У нас есть Motor Shield, на котором установлены драйверы. Кроме того, на Motor Shield выведены все необходимые контакты для подключения.

Несколько слов об аккумуляторах. В комплекте идут два аккумулятора 3.7 Вольт на 2000 мАч каждый, что при последовательном подключении даёт 7.4 Вольт и 2000 мАч. Что это значит? 7.4 Вольт достаточно, чтобы запитать Arduino и двигатели. 2000 мАч говорят о том, что батареи способны выдавать ток в 2000 миллиампер или 2 Ампера в течение часа. Много ли это? Смотря для чего. Моторы - товарищи прожорливые. Наши берут примерно 200-500 мАч каждый, в зависимости от нагрузки. Arduino UNO берёт около 50 мАч. Делаем соответствующие выводы.

УПРАВЛЕНИЕ ПО BLUETOOTH

С помощью Bluetooth модуля можно управлять роботом с телефона. На Arduino загружаем скетч [BluetoothCar](#). Для связи Arduino и смартфона через Bluetooth есть множество различных приложений. В поиске ключевыми словами будут: Arduino, Bluetooth, RC.



Пример приведен с приложением под названием Bluetooth RC Controller. Оно идеально подойдет для нашей цели. При нажатии на кнопку приложение отправляет значение типа char на Bluetooth модуль, который, в свою очередь, передает пришедшее значение Arduino. Значение, отправляемое при нажатии кнопки, устанавливается самостоятельно.



Чтобы установить соединение с роботом, нужно нажать на шестеренку и выбрать пункт “Settings” В “Settings” нужно убедиться, что кнопки соответствуют этим посылаемым символам, либо изменяем код Arduino.



После настройки символов можно устанавливать соединение с Bluetooth модулем. Ждем на шестеренку и заходим в “Connect to car” Открывается окно сопряженными устройствами. В нем выбираем BT04-A. Если его нет, то ищем с помощью “Scan for devices”. Если устройство найдено, но не хочет сопрягаться, то заходим в Bluetooth на смартфоне и сопрягаем, как обычно. Пароль стандартный – 1234. После этого заходим в “Scan for devices” и соединяемся.

Машинкой управлять можно в двух режимах - кнопками или акселерометром. Выбираем тот, что больше нравится.

СТРИМИНГ ВИДЕО ПО WIFI

Используем WiFi модуль, чтобы передать видео с камеры на телефон и управлять роботом с телефона.

Для этого нужно:

1. Устанавливаем приложение «WiFiCarApp.apk», находящееся в папке [WiFiCar](#).
2. Заливаем код WiFiCar.ino, находящийся в папке [WiFiCar](#) и запускаем Arduino.
3. Включаем WiFi на телефоне и подключаемся к zywifibot.
4. Открываем приложение, управляем роботом, смотрим видео с камеры.

ЗАКЛЮЧЕНИЕ

На этом инструкция по сборке заканчивается, а начинается самая интересная и важная часть - самостоятельное изучение.

Напоследок несколько советов, которые помогут в дальнейшем:

- Начиная какой-либо проект стоит установить конкретную цель, при достижении которой проект можно считать выполненным. Ещё лучше - составить план на какое-то время, а потом его придерживаться.
- Не скупиться на документацию. Сейчас-то всё понятно, но пройдет месяц и всё забудется. Приятно открыть готовый код и с помощью документации вновь разобраться в нём. И портфолио потом проще будет составлять.
- Собираем работа так, будто он на Луну завтра полетит. Временные вещи стремятся стать постоянными. Для заправки - [вот как](#) спаять два провода, чтобы соответствовать стандартам NASA.
- Бывает, что вроде есть целый день на занятие делом, а через некоторое время понимаешь, что день уже закончился, а дела ещё не сделаны. Куда уходит время? Это можно отследить. [Toggl](#) идёт на помощь. Полезное приложение для оценки времени, которое мы тратим.
- При разработке программ бывает утеря кода. Свет пропал в доме, или сохранилось не то, что хотели - много чего может случиться. Чтобы минимизировать последствия неприятностей можно пользоваться системами контроля версий. [Тут](#) можно почитать про git.
- Если проект задуман сложный, то можно поискать чего на этот счёт думают другие люди. Пользуемся поисковиком и [Google Scholar](#) в качестве тяжёлой артиллерии.