

# ARDUINO ROBOT CAR С ВИДЕОКАМЕРОЙ И WI-FI

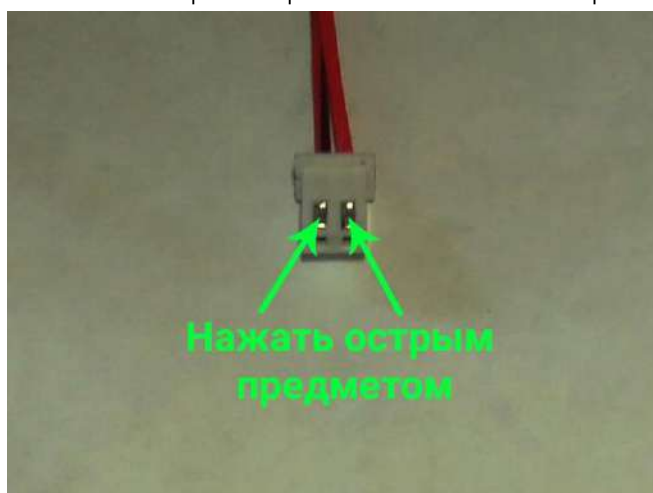
Проекты



## КАК ПЕРЕСТАВИТЬ ПРОВОДА В ДВИГАТЕЛЕ

При подключении моторов может случиться такое, что моторы робота крутятся в противоположные стороны, и робот никуда не едет. Первым делом стоит проверить правильность подключения моторов, но если они точно подключены правильно придётся переставить провода моторов. Для этого нужен какой-либо небольшой острый предмет типа ножа или плоской отвёртки.

Отключаем мотор, провода которого мы хотим переставить. На коннекторе есть пазы, в которых видно наконечник провода. Туда нужно нажать острым предметом и одновременно потянуть за провод.



Вытащив оба провода, вставляем их обратно, поменяв местами. На концах проводов есть небольшие зацепки. Их нужно отогнуть, чтобы провода не вылезали. Вставлять необходимо в той же ориентации, в какой провода доставались.

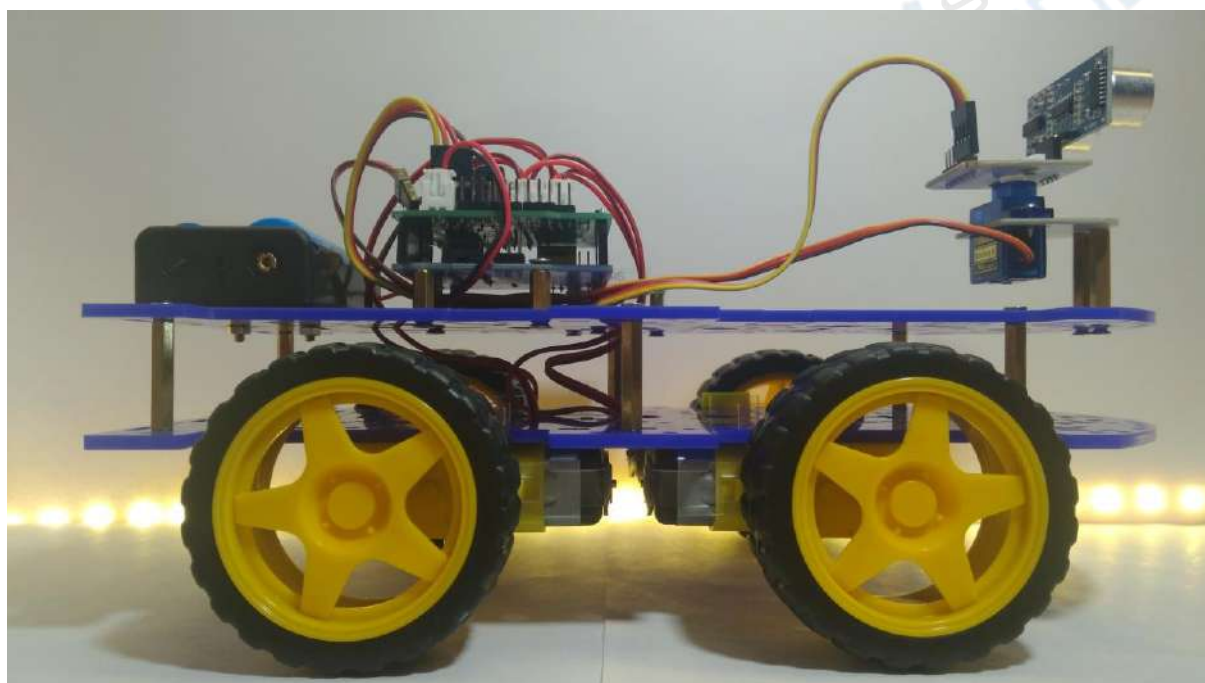


## УЛЬТРАЗВУКОВОЙ ПОСЛЕДОВАТЕЛЬ

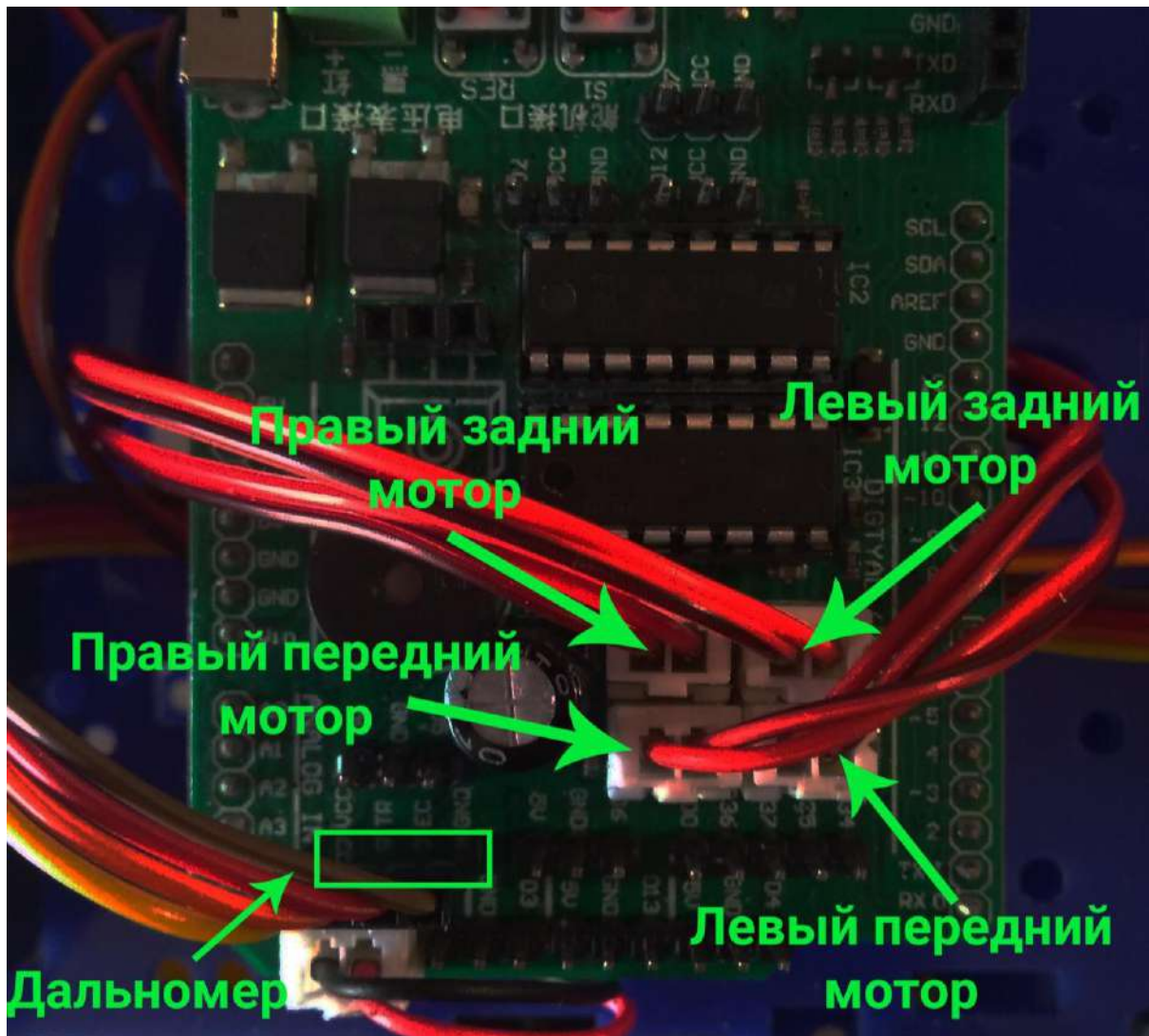
В этом проекте мы используем ультразвуковой дальномер, чтобы держать заданное расстояние до объекта слежки. Предварительно стоит ознакомиться с ультразвуковым дальномером в разделе Датчики и собрать базовую конфигурацию робота в разделе Сборка.

### Детали

Нам понадобится базовая конфигурация робота и ультразвуковой дальномер.



Подключаем моторы и дальномер, как представлено на изображении ниже. Получится, что левые моторы подключены к пинам 8 и 9, правые моторы - к пинам 10 и 11, дальномер - к пинам A0 и A1. Пины 8 и 11 отвечают за направление движения моторов. Если на них подать LOW, то робот поедет вперёд, если подать HIGH - назад. Пины 9 и 10 отвечают за скорость движения моторов. Хотя пины цифровые, выход на них можно подавать как на аналоговые. Это достигается путём использования широтно-импульсной модуляции (ШИМ). ШИМ может имитировать разные уровни напряжения, включая и отключая цифровой пин с различной частотой.



## Идея

Перед роботом будет помещаться какой-нибудь объект, например, книга. Робот должен удерживать заданное расстояние до объекта, подъезжая, если объект удаляется, и отъезжая, если объект приближается. Расстояние до объекта будем определять с помощью ультразвукового дальномера. Первый алгоритм, который приходит в голову это:

- 1) Считываем расстояние до объекта.
- 2) Если расстояние до объекта меньше желаемого - едем назад.
- 3) Если расстояние до объекта больше желаемого - едем вперед.
- 4) Если расстояние до объекта равно желаемого – стоим.

Давайте его и попробуем реализовать. Внимание на код `FollowerNaive`, который вы можете скачать [здесь](#).

## Пояснения

Код работает, однако, что-то движения резковатые, да и робот часто переезжает нужную точку, и ему приходится отъезжать назад. Это происходит из-за того, что наш алгоритм имеет всего три состояния: полный вперед, остановка, полный назад. Проблема в том, что даже если мы на сантиметр отделились от требуемого расстояния, робот включит моторы на полную мощность и не успеет вовремя остановиться. Нужно придумать что-то, что сможет решить эту проблему. Кому интересно - может попробовать придумать решение, а самые нетерпеливые читают дальше.

## Улучшенный алгоритм

Давайте улучшим наш алгоритм. Вот идея поинтереснее. У нас есть два расстояния: желаемое расстояние до объекта и реальное расстояние до объекта. Можно ввести так называемую функцию ошибки. За функцию ошибки возьмём разницу между желаемым и реальным расстоянием. Интересно то, что функция ошибки тем больше по модулю, чем дальше робот от желаемой цели. Это можно использовать: чем ближе мы к цели, тем меньше будет скорость робота.

Новый алгоритм

- 1) Считываем расстояние до объекта.
- 2) Считаем функцию ошибки, скорость моторов устанавливаем пропорционально ошибке.
- 3) Если ошибка отрицательна - едем вперед.
- 4) Если ошибка положительна - едем назад.

Обновленный код [Follower](#).

## Пояснения

Уже лучше. Робот реагирует на изменение расстояния до цели плавнее и больше не переезжает нужную точку. В коде есть какая-то странная переменная  $k$ .  $k$  - это пропорциональный коэффициент. Функцию ошибки нужно как-то перевести в скорость робота. Скорость робота может принимать значения от 0 до 255, а функция ошибки, теоретически не ограничена ни сверху, ни снизу. На практике функция ошибки принимает значения от -400 до 400 из-за ограничения

дальномера в 4 метра. Кроме этого, маленькой скорости может не хватить, чтобы сдвинуть моторы с места. Пропорциональный коэффициент решает проблему нехватки скорости - мы увеличиваем скорость, а взятие абсолютного значения от функции ошибки помогает избавиться от минуса. Коэффициент подбирается вручную, путём изменения и наблюдения за результатами. Алгоритм, который мы запрограммировали - это частичка более совершенного алгоритма. Можно улучшить наш алгоритм, беря во внимание ускорение робота и прошлые значения ошибки. Кому интересно - гуглим PID регулятор.

## Ответы на возможные вопросы

В: У меня робот в другую сторону едет. Что делать?

О: Возможно моторы подключены наоборот. Либо переподключить моторы, либо в коде поменять пины подключения левых и правых моторов местами.

В: А можно изменить расстояние, которое нужно удерживать?

О: Можно. В коде нужно поменять значение DESIRED\_DISTANCE на желаемое.

В: Робот не едет и что-то пищит. Что это такое?

О: Это пищат моторы. Либо села батарейка, либо скорости, которая подана на моторы, не хватает, чтобы преодолеть силу трения и сдвинуться с места. Пробуем добавить скорость, увеличивая SPEED и увеличивая k.

## Заключение

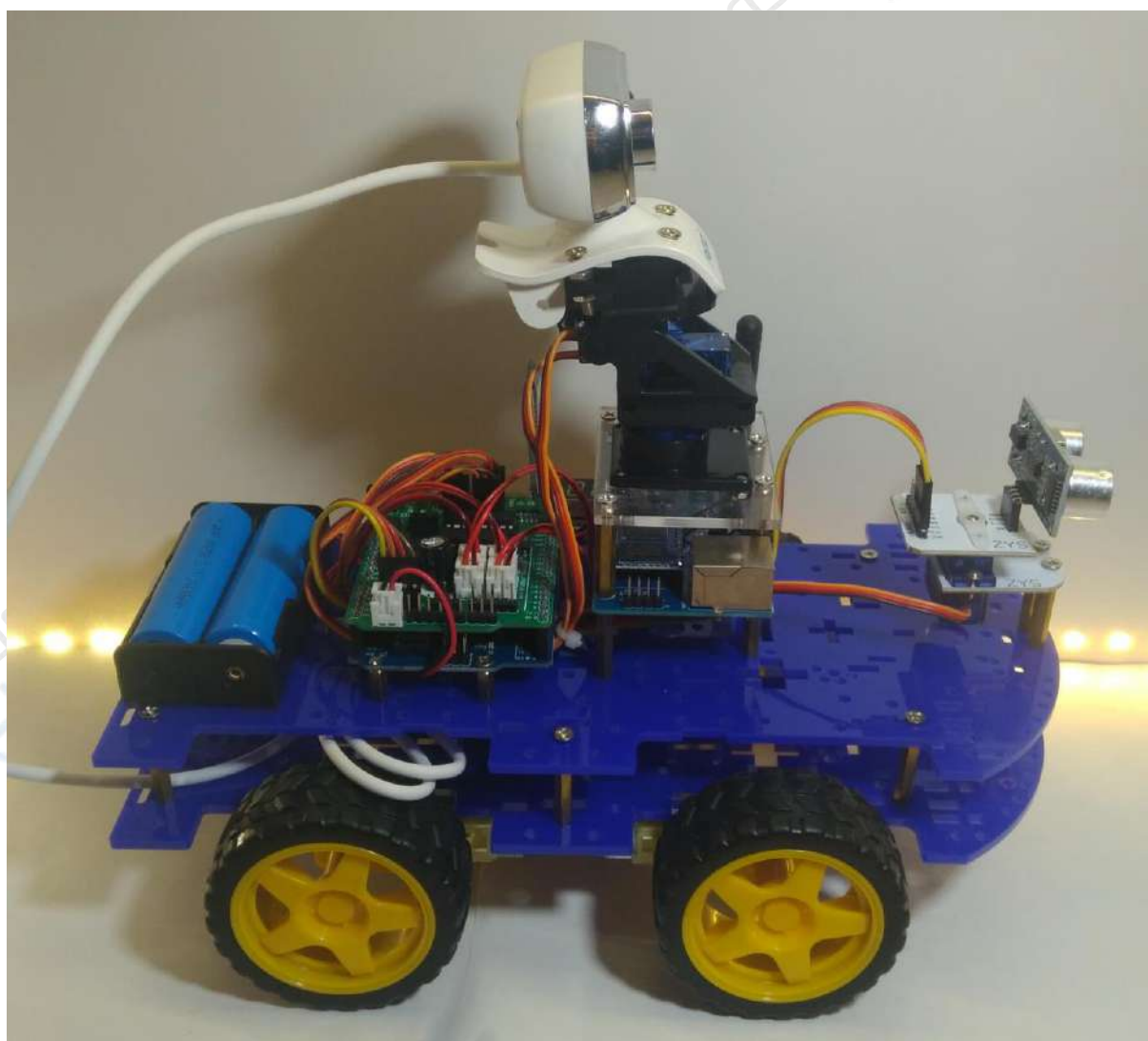
В этом проекте мы сделали робота, который может удерживать заданное расстояние до объекта, познакомились с простым и более сложным решениями поставленной проблемы. Для дальнейшего развития можно почитать про ШИМ и PID-регуляторы и сделать робота-оператора, который будет следовать за целью и снимать её на камеру.

## ВСЕВИДЯЩЕЕ ОКО

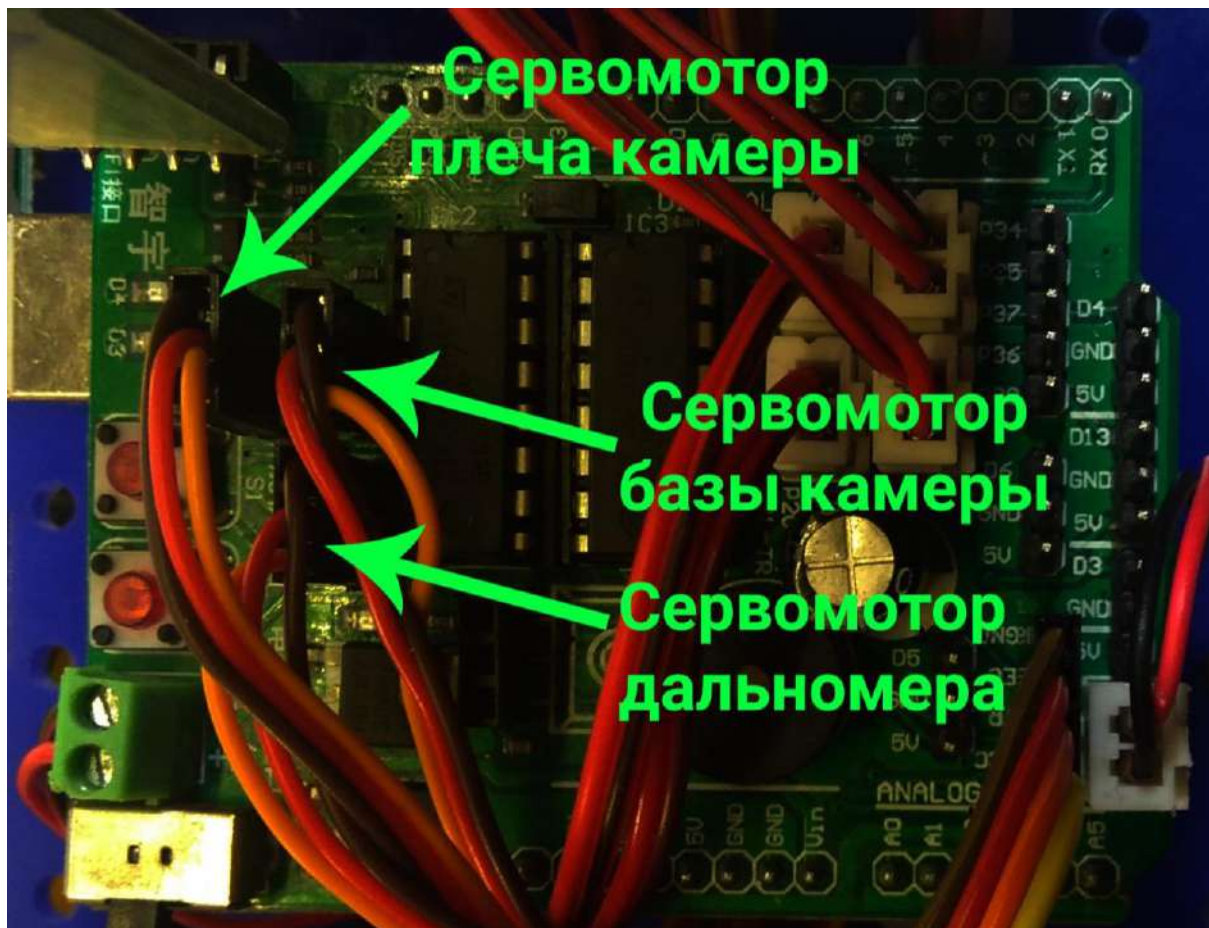
В этом проекте мы научимся использовать башню с камерой для наблюдения за окружающим пространством. Ни один хоббит не останется незамеченным! Предварительно можно почитать про сервомоторы в разделе Моторы и собрать базовую версию робота с башней и ультразвуковым датчиком из раздела Сборка.

### Детали

Нам понадобится башня для камеры и ультразвуковой датчик, прикрученные к базовой конфигурации. Для части с Bluetooth управлением нужно подключить Bluetooth модуль и раздобыть телефон с приложением Bluetooth RC Controller.



Подключаем сервомоторы, как на картинке (можно подключить по-своему и поменять номера пинов в коде). Сервомотор дальномера подключён к пину номер 2, сервомотор базы камеры - к пину номер 12, сервомотор плеча камеры - к пину 7.



### Идея

Давайте рассмотрим работу сервомоторов и используем их для наблюдения за окружением робота. В дополнение сделаем управление сервомотором по Bluetooth.

Управление сервомотором осуществляется с помощью широтно-импульсной модуляции (ШИМ). Электроника сервомотора ждёт приказ в виде импульса определённой ширины и устанавливает сервомотор в позицию, которая соответствует пришедшему импульсу. Чтобы не усложнять себе жизнь, мы воспользуемся встроенной библиотекой Arduino - Servo. С помощью Servo можно отправлять приказ сервомотору в виде угла, в который нужно встать. Нашим



сервомоторам можно задать угол от 0 до 180 градусов. После отправки команды, сервомотор будет удерживать заданную позицию. Чтобы проверить это, нужно легонько попробовать сдвинуть сервомотор. Он будет сопротивляться. Только не переусердствуйте с попытками - мотор может сгореть. Давайте рассмотрим код для управления башней камеры и базой дальномера.

Код [ServoTower](#)

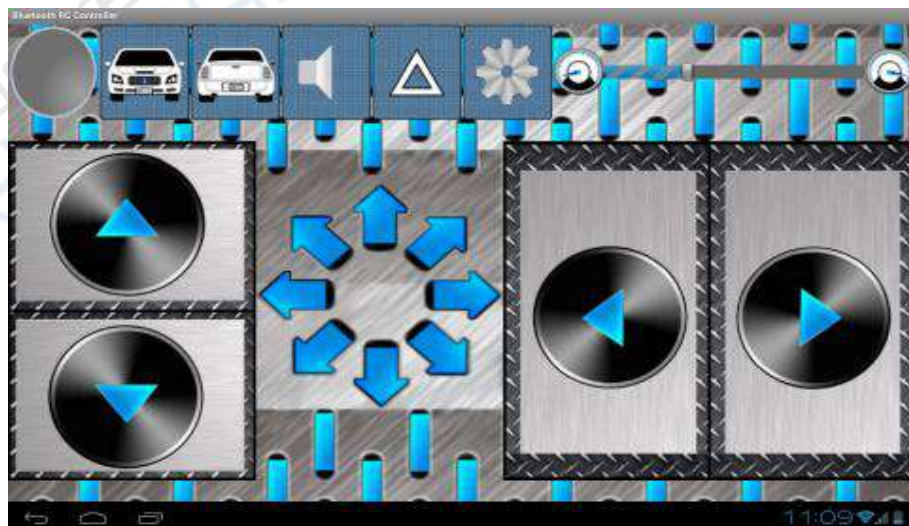
## Пояснения

Сервомоторами мы управляем с помощью библиотеки Servo. Для начала создаём объекты сервомоторов. К каждому объекту прикрепляем пин соответствующего сервомотора. После прикрепления можно подавать команды с помощью команды write. Вот общий вид команды:

`имя_мотора.write(угол);`

Чтобы поворачивать башни непрерывно был создан цикл, который постепенно изменяет угол, в который нужно встать сервомотору. Используя этот пример можно сделать систему наблюдения за периметром и радар на дальномере, чтобы ни один нарушитель спокойствия не остался незамеченным.

## Bluetooth управление



Теперь сделаем ручное управление сервомотором по Bluetooth. В нашем примере используется приложение Bluetooth RC Controller. В нём есть ползунок, которым обычно регулируется скорость моторов. Мы найдём ему другое применение.

Код [ServoBluetooth](#)

## Пояснения

Подключившись к роботу по Bluetooth можно управлять положением ультразвукового дальномера с помощью ползунка. В зависимости от положения ползунка телефон посылает символы от 0 до 9. Когда приходит команда с Bluetooth, мы расшифровываем её и, если это была цифра от 0 до 9, устанавливаем сервомотор в соответствующее положение. Положение сервомотора варьируется от 0 до 180 градусов, а пришедшая команда от 0 до 9. Чтобы перевести команду с Bluetooth в команду сервомотору, умножаем число с Bluetooth на 20.

## Возможные вопросы

Ответы на возможные вопросы

В: Что-то у меня ничего не двигается. Что делать?

О: Проблем может быть несколько. Первым делом стоит проверить, подключено ли питание робота. Сервомоторы не такие прожорливые как DC моторы, но втроём они расходуют значительную часть энергии и могут не работать с питанием от USB. Далее стоит проверить подаются ли команды сервомотору в принципе. Можно написать простейшую программу с одной командой сервомотору. Ещё можно попробовать подключить другой сервомотор - вдруг этот сгорел.

В: Зачем из пришедшего сообщения вычитается '0'?

О: Сообщения приходят в виде символа. Если пришло сообщение 'З', то это не цифра три, а символ три. Символы приходят в кодировке ASCII. Каждому символу соответствует число от 0 до 255. '0' соответствует число 48, а 'З' - число 51. Если из 'З' вычесть '0', то мы получим нужное число 3. Вот таким нехитрым способом можно перевести ASCII символ три в число три.

В: Захотел подключить сервомотор, подключил - машинка больше не работает. Что это такое?

О: Библиотека Servo отключает ШИМ на пинах 9 и 10, как раз тех пинах, куда подключены наши моторы. Решение - либо забыть про ШИМ и управлять в режиме включить-отключить, либо переносить моторы на другие пины. Во втором случае нужен другой Motor Shield.

## Заключение

Мы научились управлять сервомоторами по Bluetooth и программировать сервомоторы на автономное движение. С полученными знаниями можно раскрыть потенциал механических частей робота на полную.

## БЕЗОПАСНОЕ ВОЖДЕНИЕ

В этом проекте мы используем ИК датчики и ультразвуковой дальномер чтобы сделать систему помощи водителю. Перед тем как приступить к проекту стоит прочитать про ультразвуковой дальномер и ИК сенсоры в разделе Датчики.

### Детали

Нам понадобятся два ИК сенсора и ультразвуковой дальномер в дополнение к базовой сборке. Крепим датчики куда заблагорассудится. Я решил закрепить по одному ИК сенсору сзади и спереди, а дальномер закрепить на стандартную позицию.



Подключаем ИК сенсоры и ультразвуковой дальномер. Передний ИК сенсор подключается к пину 5, задний ИК сенсор - к пину 6, дальномер - к пинам A0 и A1.



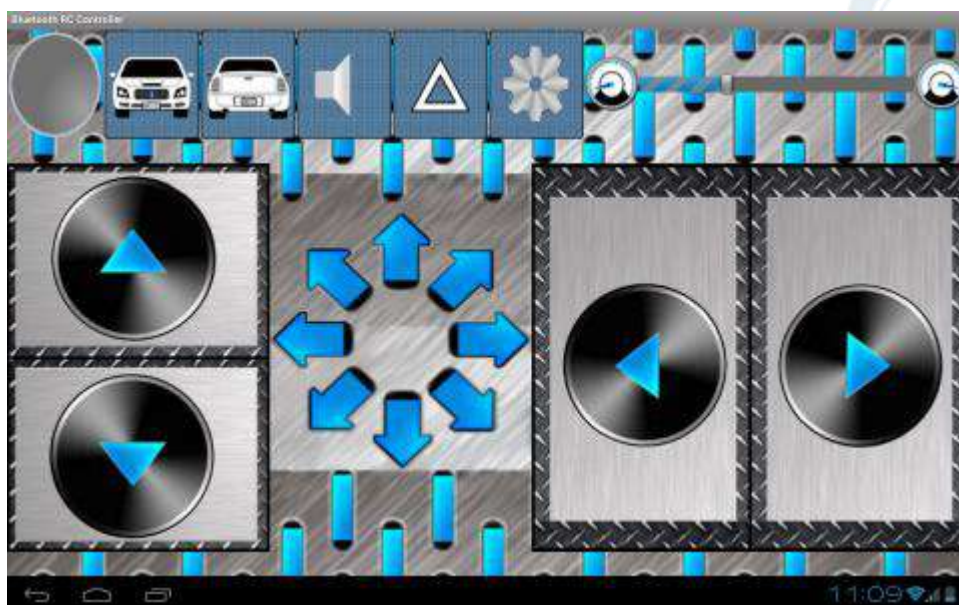
## Идея

Водитель робота должен следить за обстановкой вокруг робота, чтобы никуда не врезаться. Давайте немного снимем нагрузку с водителя и обеспечим робота системой объезда и избежания препятствий.

Звуковой дальномер будет служить детектором препятствий впереди робота для объезда, а ИК датчики - сигнализировать о необходимости экстренной остановки.

Код [DriveAssist](#)

## Пояснения



Роботом мы управляем по Bluetooth в приложении Bluetooth RC Controller. Ползунок отвечает за регулирование скорости моторов, кнопками осуществляется управление. Нажатие на кнопку посылает какую-либо команду по Bluetooth.



Для начала нам нужно узнать команды, посылаемые при нажатии каждой из кнопок. Для этого нажимаем на шестерёнку и выбираем пункт Settings. Зная команды кнопок, можно написать обработку входящих команд.

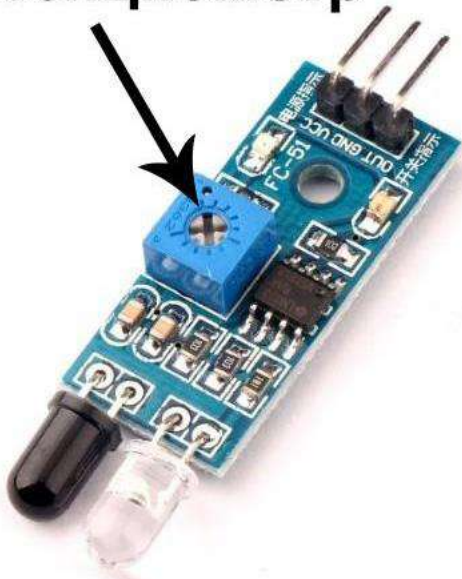
В коде реализовано несколько функций, которые будут выполняться в зависимости от пришедшей команды. Например, если пришла команда 'F', то вызывается функция

```
forward(speed);
```

Она командует моторам ехать вперёд со скоростью speed. Скорость моторов устанавливается ползунком в приложении. Чтобы обработать команду изменения скорости, которая пришла с ползунка нужно перевести пришедший ASCII символ в число. Для этого вычитаем '0' из символа. Каждому символу соответствует число от 0 до 255. '0' соответствует число 48, а '3' - число 51. Если из '3' вычесть '0', то мы получим нужное число 3.

С управлением роботом разобрались. Теперь перейдём к подключению датчиков. Главная задача дальномера и ИК сенсоров - замечать приближающиеся препятствия и либо сигнализировать об этом, либо посылать команду объезда или экстренной остановки.

## Потенциометр



Перед поездкой ИК сенсоры нужно настроить на желаемое расстояние до препятствия, которое будет считаться опасным. Для этого крутим потенциометр на сенсоре и проверяем показания датчика по светодиоду на плате. Как только ИК сенсоры замечают препятствие, движение в сторону препятствия блокируется во избежание столкновения.

Ультразвуковой сенсор отслеживает препятствия, которые расположены дальше, чем ИК сенсоры могут заметить, поэтому его реакцией на препятствие является попытка объезда. Как только было замечено препятствие, включается режим объезда, и робот сворачивает в сторону от препятствия и останавливается, ожидая дальнейших указаний.



Ещё наша система издаёт звуковой сигнал в случае обнаружения препятствия ИК сенсорами и попытке проехать ещё ближе к препятствию. Для этого мы используем пьезоэлемент, или по-простому пищалку. Чтобы издать звук используем функцию tone. Вот функция в общем виде

**tone(пин пищалки, частота, продолжительность);**

Получилась следующая система. Водитель управляет роботом с телефона по Bluetooth. В случае обнаружения препятствия дальномером или ИК сенсорами робот автономно предпринимает действия по предотвращению столкновения и звуком уведомляет водителя о препятствии. Получился нулевой уровень из шести уровней автоматизации беспилотных автомобилей. Нулевой уровень означает то, что машина может выдавать предупреждения и иногда вмешиваться в процесс вождения. Для начала неплохо.

## **Возможные вопросы**

**В:** Когда нажимаю вперёд, робот постепенно уезжает в сторону. Как это исправить?

**О:** Это происходит из-за физических различий моторов. Не получится создать полностью идентичные моторы. Даже если на моторы подключить к одному и тому же источнику питания, скорость вращения может отличаться. Чтобы решить проблему можно переставлять моторы местами для нахождения более похожих моторов. Другой способ - это программная коррекция. Опытным путём устанавливается какой из моторов крутится быстрее, а какой медленнее. На основе этих данных вводятся подстроечные коэффициенты, которые корректируют скорость, подаваемую на конкретный мотор. Коэффициенты подбираются вручную путём проведения экспериментов.

**В:** Можно как-то изменить поведение робота при обнаружении препятствия перед ним?

**О:** Да. Для этого необходимо изменить код в условных операторах на свой.



В: Можно изменить безопасную дистанцию до препятствия?

О: Да, можно. За неё отвечает переменная MIN\_DISTANCE.

В: Захотел подключить сервомотор, подключил - машинка больше не работает. Что это такое?

О: Библиотека Servo отключает ШИМ на пинах 9 и 10, как раз тех пинах, куда подключены наши моторы. Решение - либо забыть про ШИМ и управлять в режиме включить-отключить, либо переносить моторы на другие пины. Во втором случае нужен другой Motor Shield.

## Заключение

В этом проекте мы реализовали мини-систему помощи водителю. Для дальнейшего развития проекта можно добавить дополнительные датчики и камеру заднего вида с парктроником. Ещё можно почитать про уровни автоматизации беспилотных автомобилей и попробовать поднять уровень с нулевого до пятого.