

СхемCAR - Android-управление машинкой по Bluetooth

После приобретения недорогого 7" китайского планшета и экспериментами с взаимодействием с Arduino по USB и Bluetooth захотелось сделать что-то более серьезное и полезное для своих детей. Так родилась идея сделать машинку с управлением от акселерометра Android-устройства и связи по Bluetooth каналу. До этого я никогда не увлекался робототехникой или РУ управлением, но желание было. Были поставлены три цели:

1. Сделать максимально простое для повторяемости устройство, которое сможет повторить любой начинающий радиолюбитель, айтишник, программист, домохозяйка. Т.е. минимум пайки, [ЛУТ](#) и [фоторезиста](#) (да простят меня радиолюбители), а основной упор сделать на готовые, дешевые и главное доступные модули, которые без проблем можно приобрести. Хотя конечно ничто не мешает спаять собственную отладочную плату или драйвер двигателей.
2. Проект полностью должен быть Open Source и мультиплатформенным. В качестве железной части использовать все популярные и современные аппаратно-программные платформы: Arduino (Processing), STM32, MSP430, PIC, AVR, [.NET Micro Framework](#) и может быть даже [Raspberry Pi](#).
3. Алгоритм управления полностью разработать самому с нуля, не читая при этом заумных книг по роботостроительству, а также сделать его максимально простым. Ну а в дальнейшем, уже можно будет повысить свой скилл читая специализированную литературу и сделать более совершенное управление.

<http://youtube.com/v/6ZL4zX8A3I8>

Управление машинкой (или гусеничной платформой) осуществляется путем наклона Android-устройства. Т.е. наклонили вперед - машинка едет вперед, наклонили влево - машинка поворачивает на лево, назад - машинка едет назад. Причем скорость движения или поворота зависит от степени наклона устройства. Чувствительность и величина наклона устанавливается в настройках Android-приложения. Данный способ управления достигается за счет доступа из программы к встроенному акселерометру (который сейчас имеется во всех современных Android смартфонах и планшетах).

Реализован также гибридный способ управления: газ - при помощи ползунка на экране, а поворот модельки - при помощи поворота Android-устройства (как виртуальный руль). Для заднего хода - отдельная кнопка. Предусмотрен и обычный способ управления от кнопок на экране, но он менее функционален и в основном служит для отладки и проверки работоспособности.

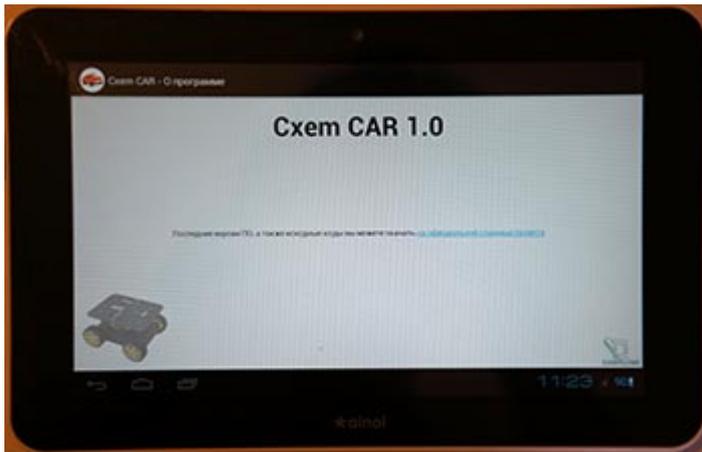
В дополнении ко всему, я ради спортивного интереса реализовал и тач управление, т.е. на экране отображается круг, и чем выше двигаете в нем маркер, тем быстрее едет машинка, чуть повели пальцем влево - машинка поворачивает.

Данные обрабатываются и передаются по Bluetooth каналу на контроллер машинки, который в свою очередь через драйвер двигателей управляет моторчиками машинки.

Компоненты

1. Android устройство

Итак, первое что прежде всего потребуется - это любое Android устройство: смартфон, планшет, часы и т.п., желательно с датчиком наклона (акселерометром) и Bluetooth (или возможностью подключения внешнего модуля через USB OTG). Я использовал китайский Ainol Aurora с внешней USB Bluetooth флэшкой. Цена такого устройства на сегодняшний день составляет менее 100\$.



2. Шасси для машинки

Также понадобится любое шасси с 2-мя или 4-мя моторчиками. Это может быть как шасси для DIY проектов, так и шасси от старой б/у радиоуправляемой машинки.

Я купил готовое шасси на eBay. Кто еще не знаком с интернет-аукционом eBay, рекомендую для прочтения эту статью: [покупка радиодеталей на eBay](#). Гарантирую, что сэкономите немало денег покупая там, а не здесь у перекупов. Также, можно воспользоваться и другими китайскими магазинами: www.aliexpress.com, dx.com и др. Найти шасси довольно легко, достаточно в поисковую строку вбить одно из словосочетаний: Robot chassis, Robot platform, DIY Car chassis и др. Стоимость варьируется от 20\$ до 60\$.



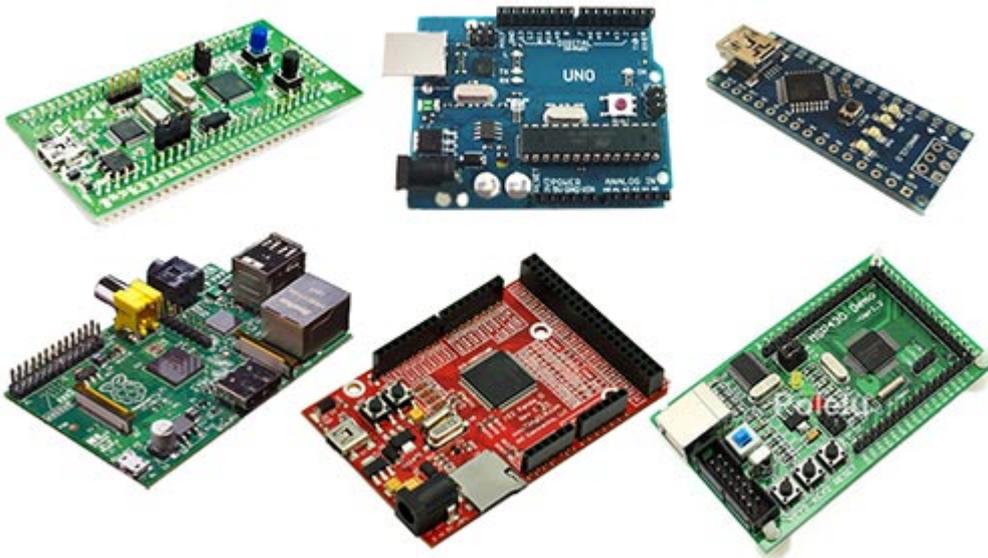
При выборе шасси обращайте внимание на питание и мощность двигателей, а также на обороты моторчиков и наличие редуктора. Хотя большинство платформ содержат стандартный 6В моторчик с редуктором.

Для принципа управления описанного в данном проекте, больше всего подходит гусеничная платформа, но т.к. у меня пока что ее нет, я реализовал проект на основе 4WD платформы.

3. Контроллер машинки

Как я уже говорил, проект планируется сделать мультиплатформенным. На текущий момент проект СхемCAR реализован для следующих аппаратных платформ:

1. STM32
2. Arduino
3. .NET Micro Framework

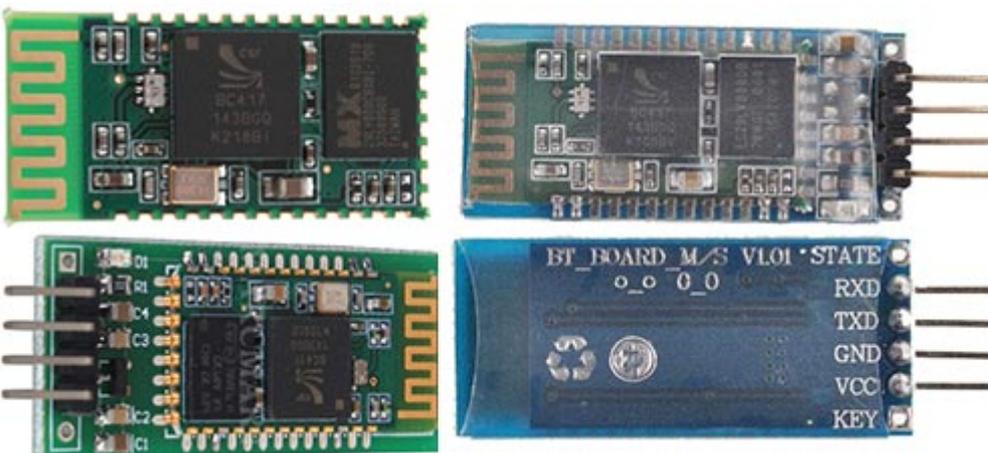


"Мозги" робота не требовательны к быстродействию МК и количеству периферии, необходимый минимум, который должен поддерживать микроконтроллер: 2 ШИМ и UART.

Если вы никогда не имели дело с микроконтроллерами, то я советую вам для начала собрать этот проект на платах Arduino, т.к. во первых они достаточно дешевы (10-15\$), а во вторых в сети и на этом сайте полно примеров, мануалов и т.п. Контроллеры STM32 и MSP430 более функциональны, но для новичка будут сложнее в освоении. Ну а для программистов, понравится один из вариантов данного проекта на [FEZ Panda II](#) с .NET Micro Framework, где в качестве среды программирования используется Microsoft Visual C# Express. Но нужно учесть, что сами платы FEZ не сильно распространены и купить их проблематично, хотя существует .NET вариант под Arduino форм-фактор под названием Netduino.

4. Bluetooth модуль

В качестве Bluetooth модуля использован дешевый китайский UART модуль HC-06. Подойдут модули вида HC-03, HC-04, HC-05, HC-06, да и вообще любые Serial Bluetooth. Лучше брать с готовыми штыревыми выводами, чтобы не пришлось паять, т.к. расстояние между выводами очень маленькое (см. 1-ой на фото внизу). Стоимость такого модуля на eBay составляет в среднем 5-10\$.

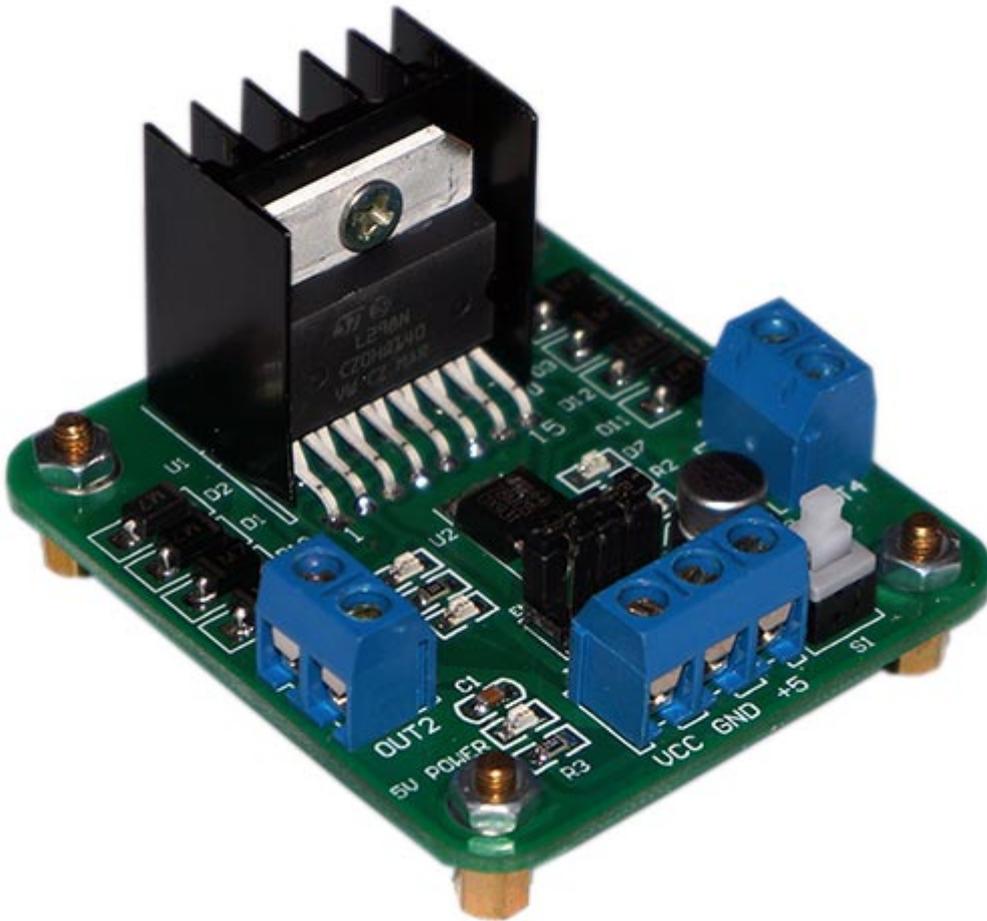


Bluetooth модули и работа с ними применительно к Arduino, неплохо описаны в [этой статье](#). Для других контроллеров все практически тоже самое, [вот к примеру статья](#) с описанием связи по Bluetooth между

STM32 и Android.

5. Драйвер двигателя

В качестве драйвера я использовал специализированную микросхему L298N, которая представляет собой двоянный мостовой драйвер двигателей и предназначена для управления DC и шаговыми двигателями. На eBay продаются готовые платы со всей необходимой обвязкой, цена платы составляет 4-5\$ и выглядит она приблизительно так:



Подключение к Arduino достаточно подробно описано в [этой статье](#). В нашем проекте для плавного изменения скорости вращения двигателей, мы будем использовать ШИМ (широтно-импульсную модуляцию).

6. Остальные комплектующие

Помимо вышеперечисленных компонентов понадобятся:

- батарейный отсек (4-5 батареек AA) или аккумуляторы, можно использовать к примеру готовые блоки Li-Po аккумуляторов на 7.4В
- соединительные провода
- выключатель питания
- термоусадочная трубка, хомуты и др.



Все это конечно опционально и можно заменить тем, что есть под рукой, к примеру вместо термоусадки использовать изоленту и т.д.

Как видите, себестоимость машинки не так уж и велика, если использовать свое шасси с моторчиками, то выходит около 20-25\$ на все, если покупать и шасси, то выйдет уже 45-60\$ в зависимости от типа шасси (т.к. диапазон цен на них очень широк).

Вот, что получилось у меня:



Принцип работы

В Android устройстве формируются команды перемещения машинки в зависимости от наклона смартфона/планшета, либо от нажатой кнопки. Все расчеты производятся в Android-приложении, и сразу же вычисляются значения ШИМ для левого и правого двигателей. Приложение обладает гибкими настройками,

такими как диапазон ШИМ, чувствительность наклона, минимальный порог ШИМ и др. По Bluetooth передаются команды вида:

L-255\rR-120\r

L - команда для левого двигателя, R - для правого
 минус обозначает вращение двигателя для движения назад
 255 - число ШИМ, для Arduino это максимальная скорость вращения
 \r - конец команды.

По данной команде машинка будет двигаться назад и немного поворачивать в правую сторону, т.к. правый двигатель будет вращаться медленнее левого.

L255\rR-255\r

По данной команде левый двигатель будет вращаться вперед, а правый назад, что заставит машинку вращаться вокруг своей оси против часовой стрелки.

H1\r

Команда включения дополнительного канала, к которому например можно подключить фары, звуковой сигнал и т.п. В качестве примера, приведен только один дополнительный канал, однако ПО легко модифицировать, чтобы задействовать большее количество дополнительных каналов.

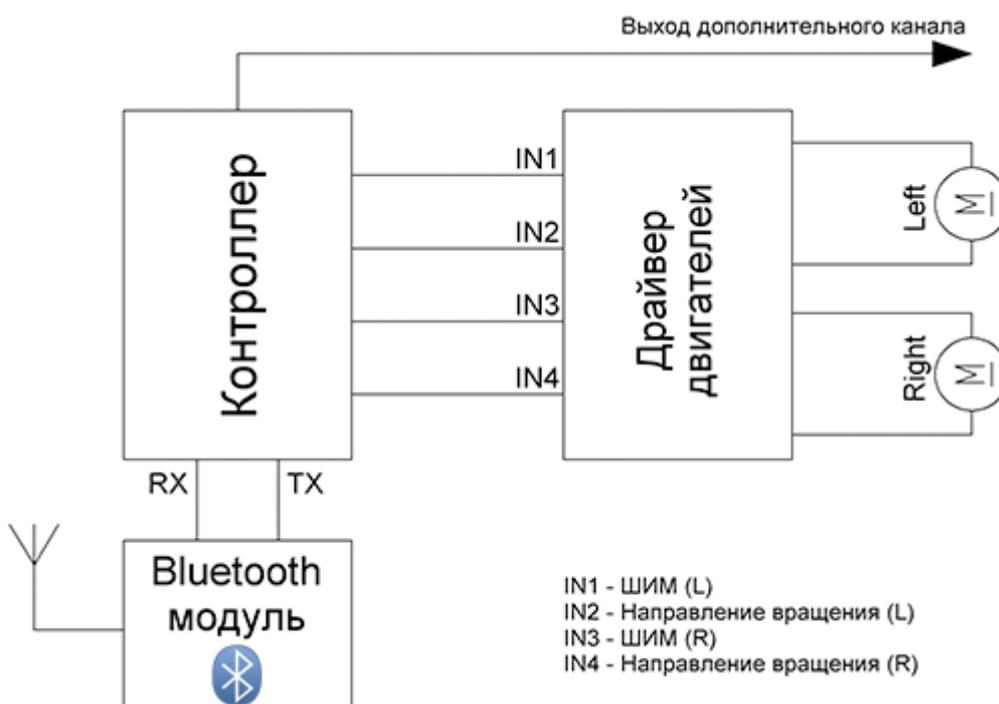
Символы команд L, R и H можно задавать в настройках Android-приложения.

В программе контроллера предусмотрен таймер, который отключает двигатели, если последняя команда была получена более, чем n-секунд назад. Настройка количества секунд хранится в EEPROM памяти контроллера и может быть изменена с Android устройства. Диапазон данной настройки составляет от 0.1 сек до 99.9 секунд. Также, настройку можно совсем отключить. Но тогда, при потере связи машинка будет ехать, пока не будет выключено питание.

Для работы с памятью микроконтроллера предусмотрены команды **Fr** - чтение значений и **Fw** - запись значений.

Электронная начинка

Структурная схема SchemCAR представлена ниже:

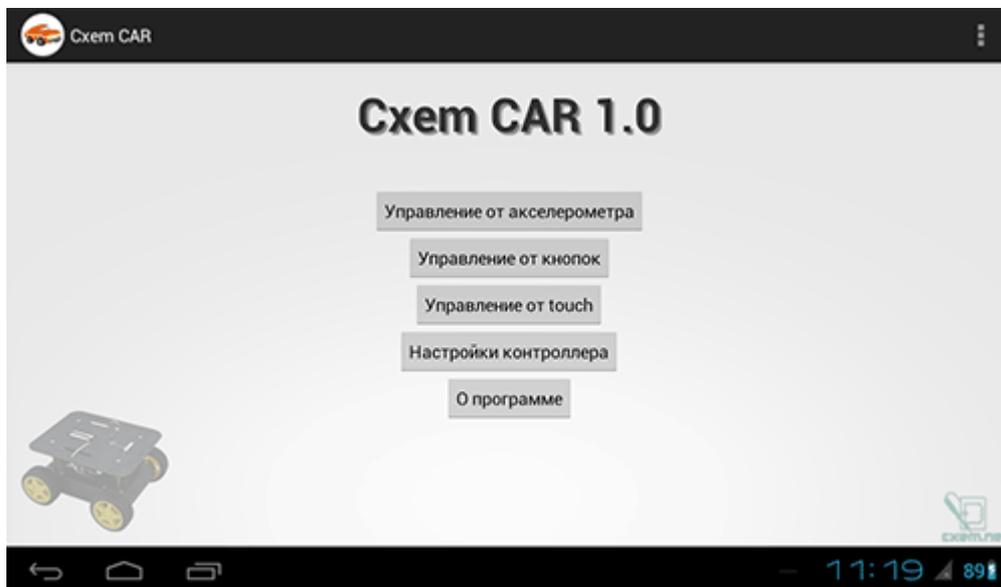


Как видим, к контроллеру (Arduino, STM32 и др. неважно какой) подключается Bluetooth модуль и драйвер двигателей, к которому в свою очередь подключается 2 или 4 моторчика Bluetooth-управляемой модели. На схеме изображен один выход дополнительного канала (включение звукового сигнала, фар и т.п.), но путем небольшой правки программы число дополнительных каналов может быть легко увеличено.

Работа с Android приложением

Приложение под Android писалось в среде Eclipse. Все исходники проекта и проект для Eclipse вы можете скачать ниже. Я не специалист в Java программировании и это мое первое приложение под Android поэтому кое-где код не совсем удачен. Именно на разработку Android приложения ушло основное время при разработке данного проекта. Версия Android должна быть не ниже 3.0, я все писал и тестировал под планшет с версией 4.0.3.

Приложение содержит несколько активити. Главное активити представляет собой начальный экран с кнопками запуска различных режимов управления и настройками:



Предусмотрено 3 режима управления Bluetooth-моделью:

Управление от акселерометра - основной способ управления. Управление движением Bluetooth-модели осуществляется за счет наклона Android-устройства (планшет, телефон и др.)

Виртуальный руль - гибридное управление. Газ - при помощи ползунка, повороты - при помощи поворота устройства. Задний ход - отдельной экранной кнопкой.

Управление от кнопок - на экране приложения выводятся 4 кнопки управления: вперед, назад, влево и вправо. При нажатии кнопки "вперед" машина едет вперед пока держите кнопку, при нажатии "назад" тоже самое, но едет назад. При нажатии кнопок "влево" или "вправо" машинка крутится вокруг своей оси в одну или в другую сторону. При этом значение скорости фиксировано (по умолчанию стоит максимальная скорость), но в настройках можно изменить данный параметр.

Управление от touch - данный способ управления я подсмотрел в игре DeathRally и ради спортивного интереса решил повторить. Честно сказать поучилось не очень удобно, но может кому-нибудь пригодится. На экране рисуется круг, внутри которого и происходит процесс управления. Повели пальцем вверх относительно центра - машинка едет вперед, чуть левее - машинка начинает поворачивать влево. Насчет этого способа управления, есть в дальнейшем идея усовершенствования с помощью компаса, т.е. использовать круг не как задатчик скорости и поворота, а задавать с помощью него направление движения.

Настройки приложения

Скриншот настроек Android приложения SchemCar версии 1.0:

ОБЩИЕ НАСТРОЙКИ	
MAC адрес MAC-адрес Bluetooth устройства	
Команда для левого двигателя Символ команды для левого двигателя	
Команда для правого двигателя Символ команды для правого двигателя	
Дополнительная команда Символ дополнительной команды (к примеру звуковой сигнал)	
Отладочная информация Отображение отладочной информации	<input checked="" type="checkbox"/>
ПАРАМЕТРЫ ПРИ УПРАВЛЕНИИ ОТ АКСЕЛЕРОМЕТРА	
Предел по оси X Предел наклона (чувствительность) по оси X	
Точка разворота для мотора (ось X) Точка разворота для мотора по оси X, выше которой один мотор начинает вращаться в другую сторону	
Предел по оси Y Предел наклона (чувствительность) по оси Y	
Минимальное значение ШИМ Минимальное значение ШИМ (порог ниже которого не вращается двигатель)	
Максимальное значение ШИМ Максимальное значение ШИМ (для Arduino 255)	
ПАРАМЕТРЫ ПРИ УПРАВЛЕНИИ ОТ ЭКРАННЫХ КНОПОК	
Значение ШИМ при кнопочном управлении (левый) Постоянное значение ШИМ при кнопочном управлении для левого двигателя	
Значение ШИМ при кнопочном управлении (правый) Постоянное значение ШИМ при кнопочном управлении для правого двигателя	
ПАРАМЕТРЫ ПРИ УПРАВЛЕНИИ ОТ ТАЧА	
Точка разворота для мотора по оси X (в процентах) Точка разворота для мотора по оси X, выше которой один мотор начинает вращаться в другую сторону	

К каждой настройке внизу есть небольшое пояснение, поэтому с их предназначением думаю не должно возникнуть вопросов. Однако на некоторых из них хотелось бы остановиться более подробно.

Точка разворота для мотора (ось X)

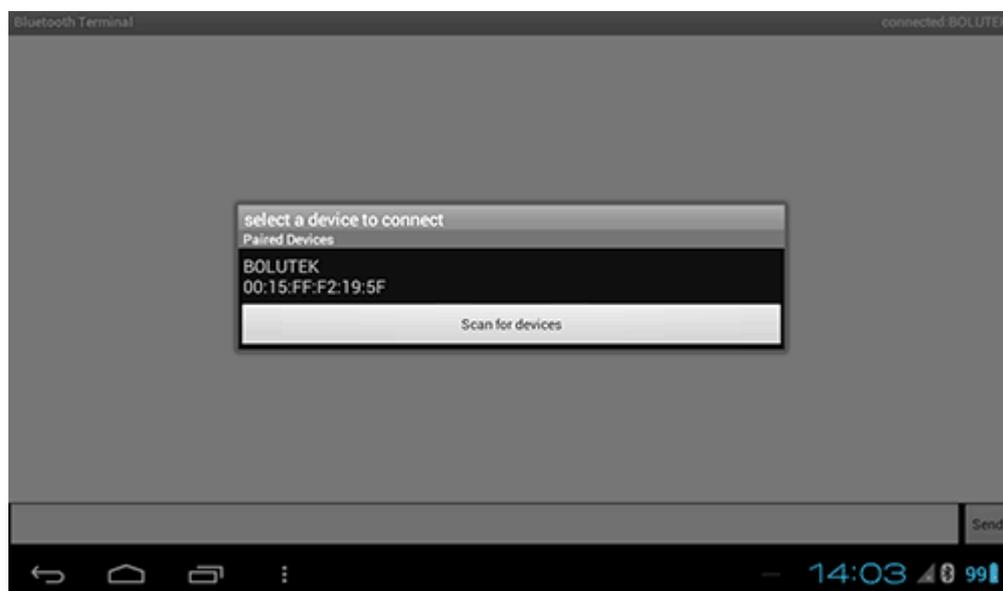
При наклоне Android-устройства влево или вправо программа притормаживает тот двигатель, в сторону которого наклонено устройство, т.о. осуществляется поворот. Однако, когда значение наклона доходит до заданной в настройках точки разворота, то двигатель начинает вращаться в другую сторону. Если наклонить

устройство максимально вбок, то один двигатель будет вращаться с максимальной скоростью в одну сторону, а другой в другую и соответственно машинка будет крутиться вокруг своей оси на месте.

MAC адрес

Для установления связи с Bluetooth модулем машинки, в настройках приложения необходимо задать MAC-адрес. Предварительно необходимо настроить сопряжение устройств в настройках вашего Android-устройства. Для этого переходим в Настройки -> Bluetooth и нажимаем "Поиск устройств", телефон находит наш Bluetooth-модуль, нажимаем по нему и вводим пароль (как правило 1234).

Узнать Bluetooth адрес модуля можно из какого-нибудь приложения, к примеру [Bluetooth Terminal](#). Для этого внизу нажимаем "Connect a device - Secure" и в появившемся окошке нажимаем кнопку "Scan for devices". ПО сканирует Bluetooth устройства и отобразит их MAC-адреса:



Этот MAC-адрес и необходимо прописать в настройках приложения SchemCAR.

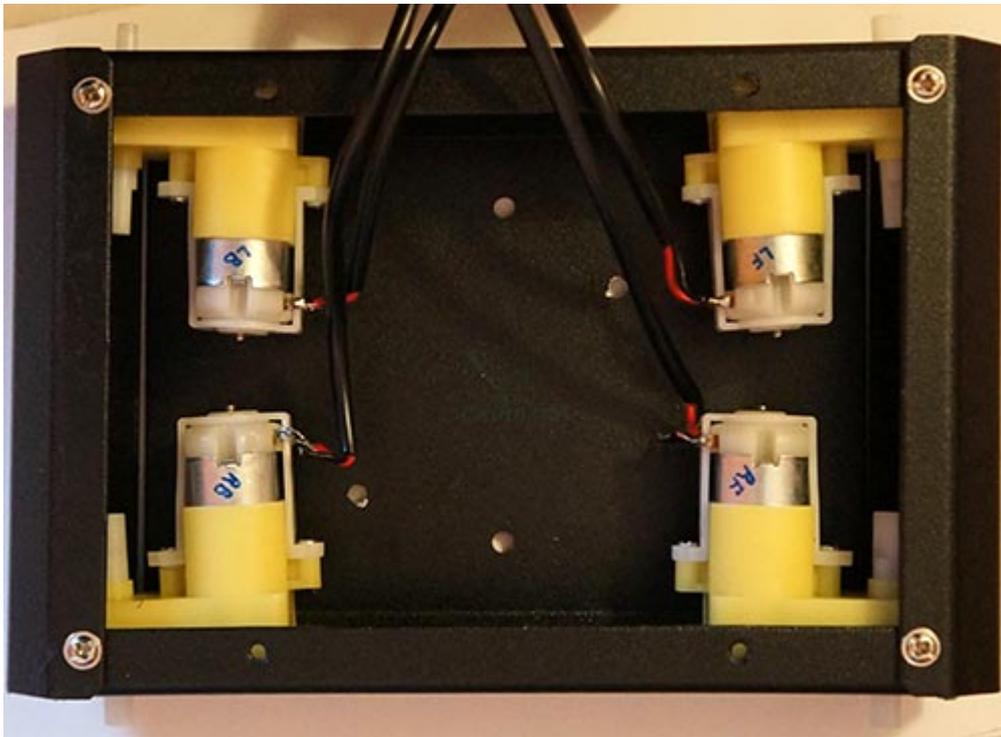
Само ПО под Android я не буду расписывать, т.к. оно довольно таки громоздкое и поэтому если у вас возникнут какие-либо вопросы по нему, то обращайтесь тему поддержки данного проекта на форуме.

Сборка платформы

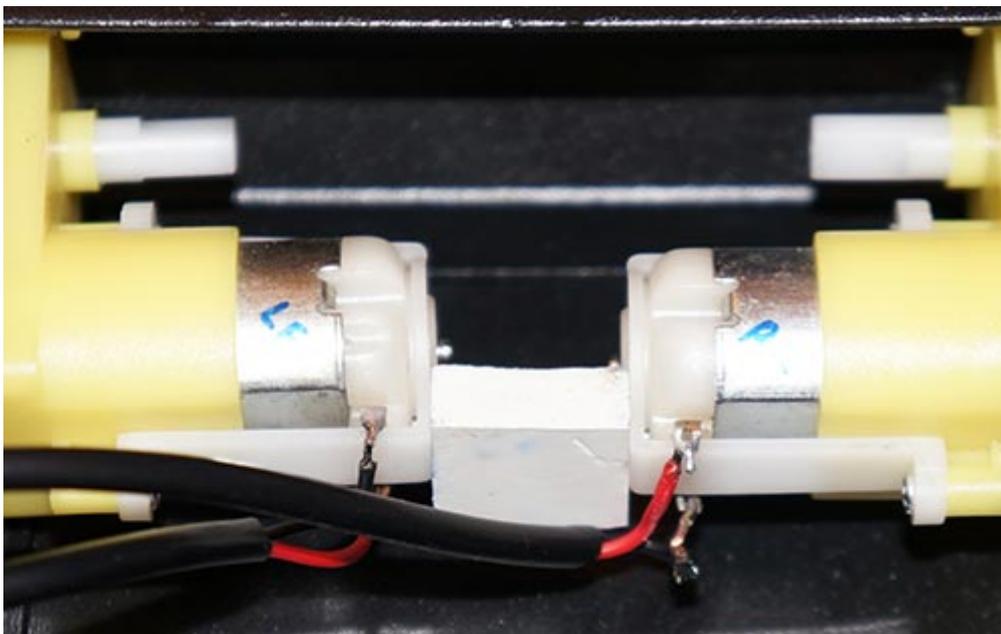


Как я уже говорил выше, в качестве платформы я выбрал шасси Pirate-4WD от производителя DFRobot (SKU:ROB0003). Это одно из самых популярных шасси, вот его [описание](#) и [видео по сборке](#). Хотя там все интуитивно понятно и сборка очень проста.

После того, как была собрана основа шасси с моторами, припаиваем провода и на всякий случай маркируем моторчики:

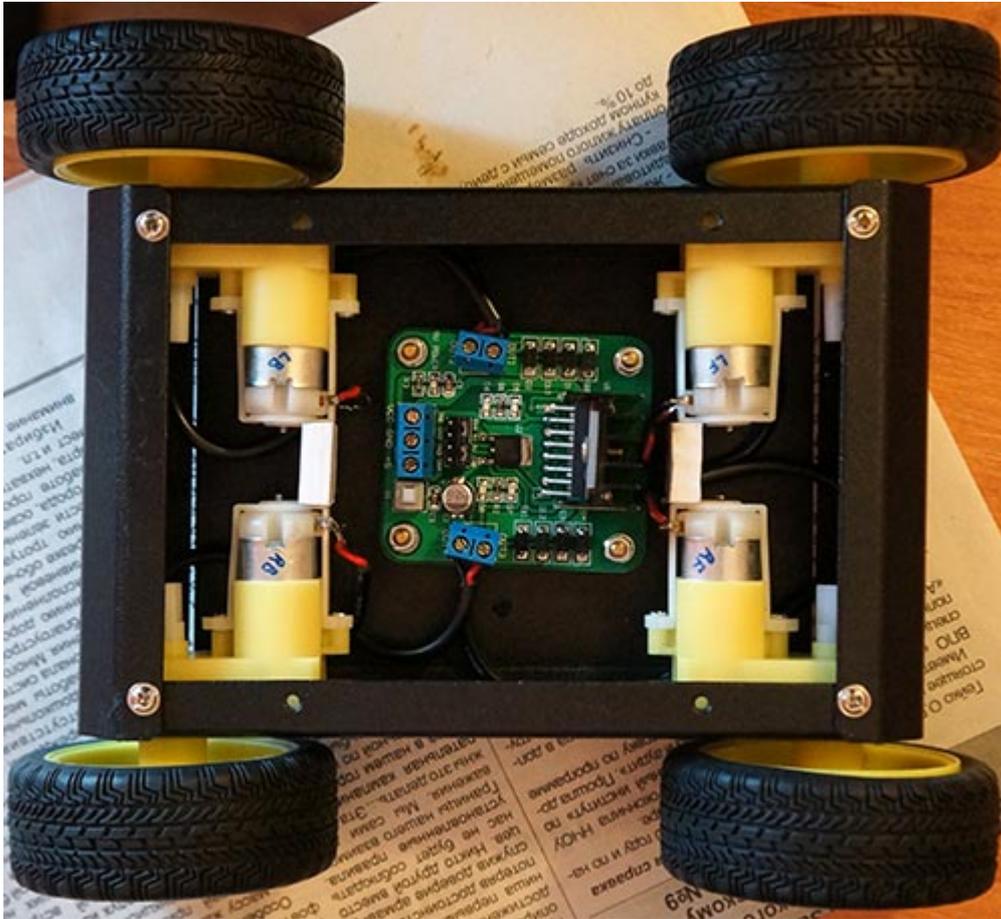


На верхнем фото хорошо видно, что левые и правые двигатели относительно друг-друга расположены несоосно из-за непродуманности китайской конструкции, и из-за этого колеса немного проскальзывали при движении. Поэтому потребовалась небольшая доработка, для этого я вырезал распорки из пластика и вставил их между двигателями.



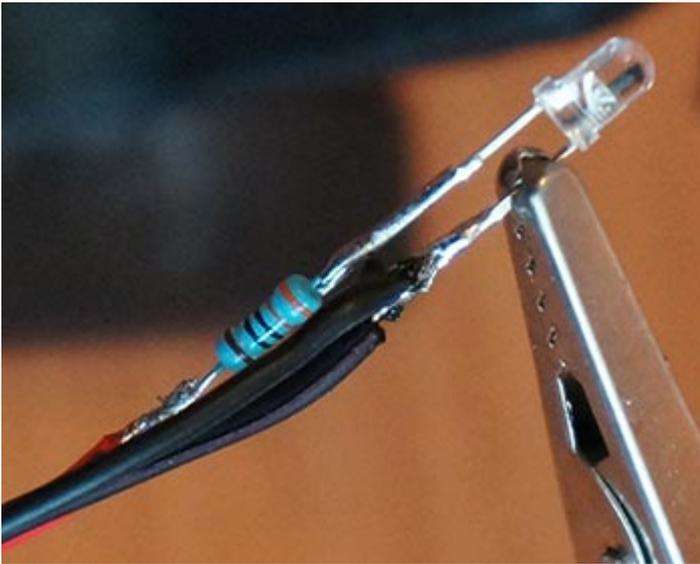
Стало немного лучше, но все равно остался небольшой градус у колес. Однако, как показали дальнейшие испытания, это не сильно влияло на ходовые качества платформы. Поэтому не стал углубляться в доработку шасси и оставил как есть с распорками.

Следующим этапом является установка колес. На вал они налезают очень туго, кое-где даже пришлось подпилить катэром.



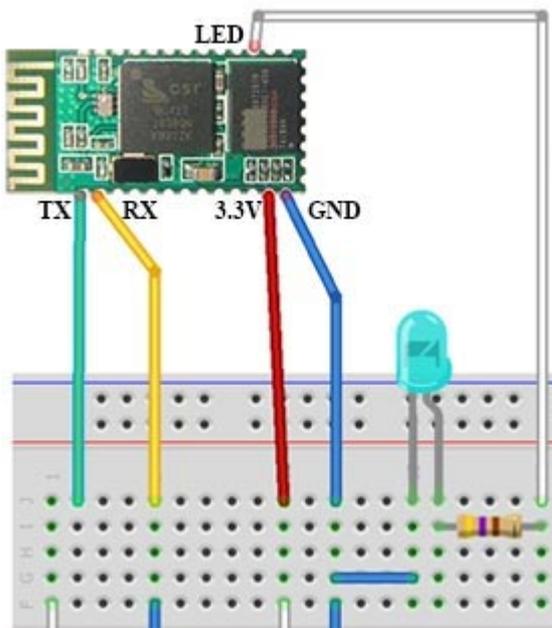
Провода от 4-х моторов подключаем к плате драйвера двигателей L298N, левые 2 мотора просто запаралеленны, тоже самое и правые.

Для того, чтобы можно было видеть состояние Bluetooth соединения, у модуля HC-06 (да и у других тоже) присутствует возможность подключения светодиода состояния. Я решил его также подключить и вывести на видное место. Для этого, я использовал токоограничительный резистор номиналом 470 Ом и термоусадочную трубку.

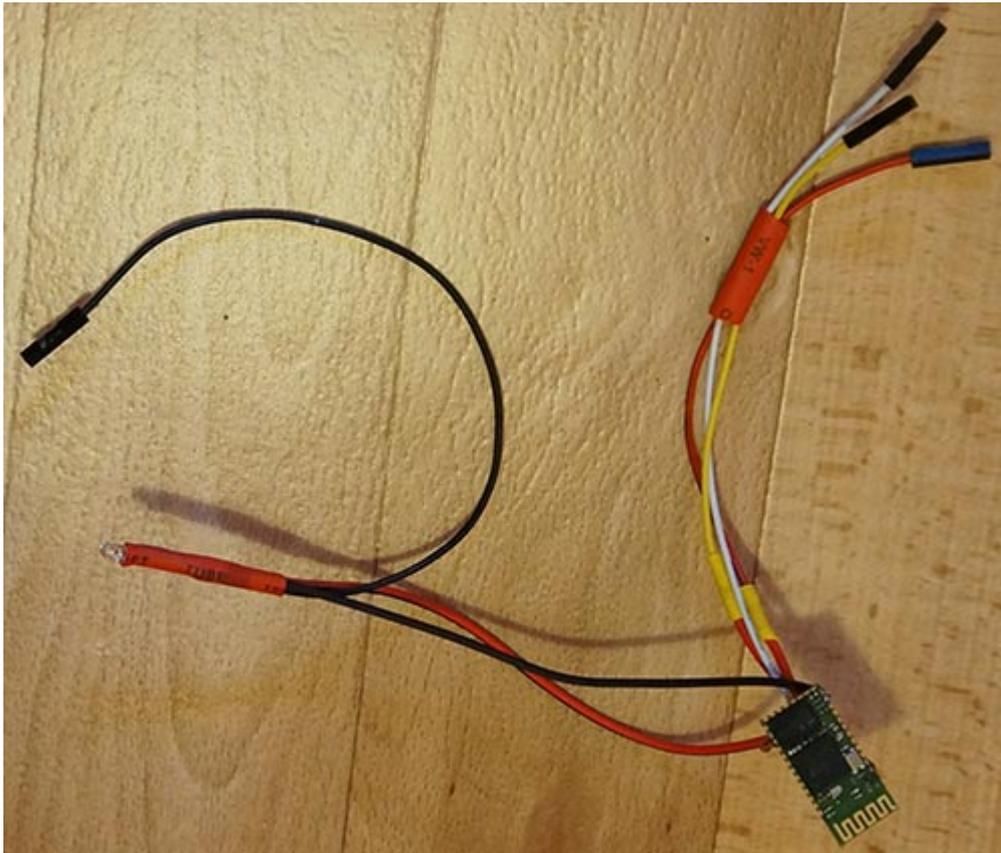


К модулю Bluetooth подключаем или подпаиваем все необходимые провода, помимо светодиода это питание, GND, TX и RX. Смотрите документацию на ваш Bluetooth модуль. Чтобы не мучаться с пайкой, я вам советую сразу взять модуль с подпаянными штыревыми выводами (см. фото выше). И лучше брать модули HC-03/HC-05.

Я же использовал модуль HC-06, который у меня был до этого. Схема распайки следующая:

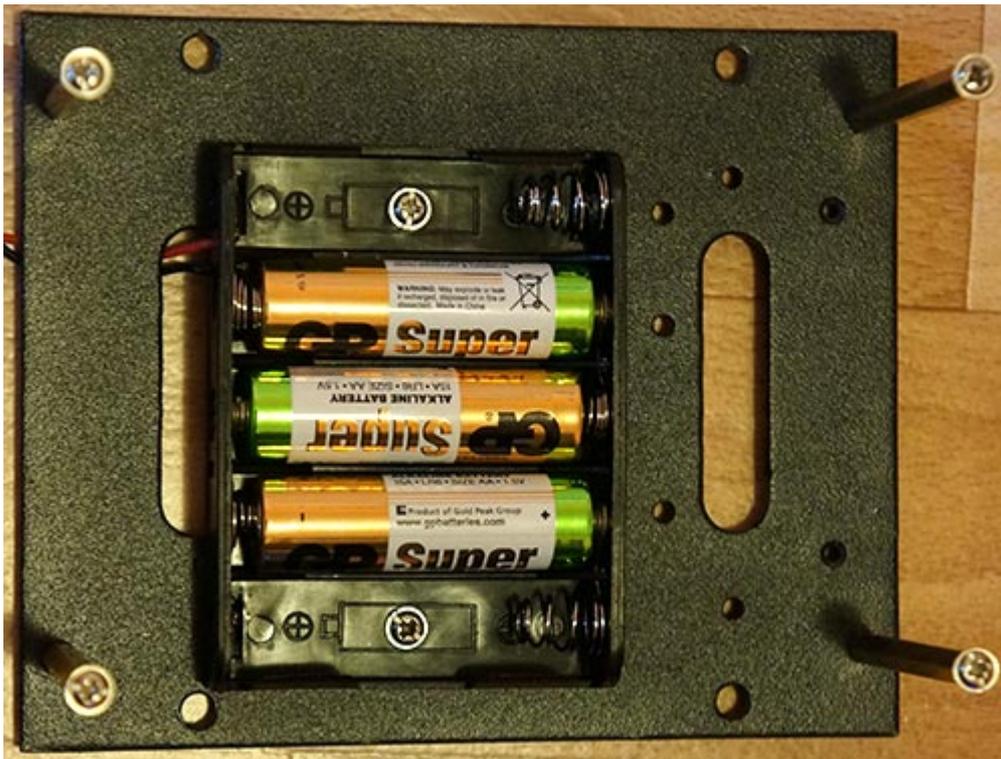


В распаянном виде модуль выглядит так:



Его я также разместил в термоусадочной трубке большого диаметра.

Для питания двигателей я использовал батарейный отсек на 5 элементов типа АА, т.е. напряжение питания двигателей составило 7.5 Вольт. Можно использовать LiPo и другие аккумуляторы. Для крепления батарейного отсека в крышке шасси просверлил два отверстия и закрепил при помощи двух болтов.



История изменений ПО для Android:

Версия 1.1 (от 28.01.2013) - в классе cBluetooth изменения для более стабильного подключения по Bluetooth. Теперь не нужно вводить код для pairing (связывания устройств)

Версия 1.2 (от 15.02.2013) - дополнен класс cBluetooth на предмет установки соединения с устройством (полезно при отладке). Изменения в классе Handler (исправлены все ошибки с static). Реализован новый вид управления - "виртуальный руль".

Версия 1.3 (от 20.07.2013) - мелкие правки в активити. Исправление ошибки с инверсией координат на смартфонах

[Проект на GitHub](#)

Схема подключения "мозгов" робота, т.е. основного контроллера, для каждой платформы будет своя, поэтому подключение и другие нюансы будут описаны в отдельных статьях:

[Аппаратная реализация СхемCAR на платформе STM32](#)

[Аппаратная реализация СхемCAR на платформе Arduino](#)

[Аппаратная реализация СхемCAR на платформе .NET Micro Framework](#)

<http://youtube.com/v/tRXcGxlb1EI>

<http://youtube.com/v/-28a1hSGIVQ>

URL статьи на сайте: [СхемCAR - Android-управление машинкой по Bluetooth](#)

Автор: [talibanich](#)

Дата публикации: 27.12.2012

Комментарии:

Максим 27.12.2012 15:18

С wi-fi работали? Было бы неплохо на нем, у блютуза радиус действия мизерный.

talibanich 27.12.2012 16:04

В планах есть и с Wi-Fi сделать, но Wi-Fi модуль в 8-10 раз дороже Bluetooth. Ну и плюс у меня флэшка 20м, а видел в продаже Bluetooth-флэшки с увеличенным радиусом действия что-то под 100 метров. Ну и плюс большая дальность, без камеры с real-time картинкой, не имеет смысла.

Максим 27.12.2012 16:20

Ну тут согласен. Просто нужна дальность свыше 100м. Правда модули нарыл, но вот управление надо поучить.

MaFiX 27.12.2012 22:58

Будет ли машина камера с real-time картинкой. Можете сделать?

talibanich 28.12.2012 07:32

Смотря что имеется ввиду по real-time. Если аналоговые FPV камеры для RC, то есть куча готовых решений, смысла изобретать велосипед нет, а установить на платформу может каждый.

А если подключение камеры к контроллеру и передача real-time IP видео на Android, то платки типа Arduino, STM32, MSP430 такое не потянут, можно сделать на базе Wi-Fi маршрутизатора, но не все USB-камеры можно успешно подключить к Linux, чтобы заставить их выдавать вмняемый поток. Даже если заюзать Raspberry Pi с аппаратным кодером, и то наблюдается проблема с драйверами для многих камер (надеюсь временно).

У меня есть одно тестовое рабочее решение с передачей IP видео на Android, но там вопрос в цене, т.к. к существующему набору добавляется еще 60-80\$. Об этом также будет статья, но позже.

Sergey 09.01.2013 12:30

Проект создавался в какой версии Eclipse?

talibanich 09.01.2013 12:36

Juno 4.2

Joni 12.01.2013 16:45

Какой Вы использовали Bluetooth адаптер для планшета? Дополнительного ПО не устанавливали чтобы планшет его увидел? На своем планшете подключал два разных адаптера, но планшет их не видит Android 4.0.3

talibanich 12.01.2013 17:02

Адаптер использовал самый дешевый, который был в магазине, китайский понаме. Они все на одних чипах построены. Фото адаптера есть в [этой статье](#)

Работал на 4.0.4, 4.1.2 Cyanogen Mod 9 и 10. Другие версии не пробовал.

Дмитрий 16.02.2013 17:49

Здравствуйте! А вы не можете добавить в программу управления поддержку более ранних версий android например версии 2.3, а то есть замечательный аппарат Sony Ericsson xperia play, с игровой клавиатурой, но для него нету возможности обновиться до последних версий прошивки.

talibanich 16.02.2013 18:30

Все исходники открыты, можете это сами сделать при необходимости. Для поддержки старых версий придется очень много переделывать в приложении, те же настройки и т.п. У меня на это нет времени, лучше двигаться вперед, чем назад.

Эльдар 04.03.2013 00:40

У меня 2 вопроса:

1) Какого быстродействие управления? От чего оно больше зависит: от типа контроллера или

производительности смартфона? Вопрос вот в чем: хочу сделать блютуз-управляемую машинку полноприводную для RC-дрифта, но вот встал вопрос: в дрифте главную роль играет точность управления, не будет ли машинка запоздало реагировать на управление? Допустим при использовании конт-ра Arduino Nano v3.0 AVR ATmega328 и смарта samsung galaxy s 2 с Android 4

2) Насколько сложно будет переделать софт для смарта при изменении управления, с мостовой схемы на обычную (1-ый шим на сервопривод руля, 2-ой шим на обороты двигателя)?

talibanich 04.03.2013 08:56

1. Зависит от пропускной способности канала передачи данных, bluetooth-канала вполне хватает. Производительности любого одноядерного смарта и любой ардуинки хватает более чем.

2. Тут можно поступить двумя вариантами:

а) через одно место (это если не хотите копаться в исходниках под андроид). В контроллере машинки пересчитывать общую скорость и разницу для сервопривода поворота машинки.

б) правильный вариант - немного переделать логику в ПО под андроид. Там ничего сложного нет если разобраться. Была бы такая машинка сделал бы, но увы DIY платформ с серво для руля нет, а готовые решения дороговаты.

Эльдар 04.03.2013 20:05

Спасибо за ответы! Если я решусь на свой проект, могу ли я рассчитывать на вашу помощь в вопросах перепрограммирования вашего андроидного софта?

talibanich 04.03.2013 20:46

При наличии моего свободного времени - да.

Sem54 23.03.2013 18:30

Всем привет хотелось бы по подробней по прошивке у меня arduino nano выдаёт ошибку:

```
E:\arduino-1.5.2-windows\arduino-1.5.2\hardware\tools\avr\bin\avr-ar: unable to rename 'C:\Users\Sem\AppData\Local\Temp\build8768165817093499422.tmp\core.a'; reason: File exists
```

```
Binary sketch size: 3.782 bytes (of a 30.720 byte maximum) - 12% used
```

```
avrdude: stk500_getsync(): not in sync: resp=0x00
```

Как я понимаю нехватка памяти?

Василий 12.04.2013 19:50

Подскажите пожалуйста какую модель аino1 вы использовали?

talibanich 13.04.2013 08:45

В статье указана модель

Александр 30.04.2013 21:31

Можете помочь как добавить еще дополнительных каналов, например на pin 10 и 11, не могу разобраться с кодом.

talibanich 01.05.2013 08:07

Для андроид приложения нигде пины не задаются. Посмотрите проект в эклипсе, там есть пример доп. канала, сделать по аналогии и все.

Тщерук 24.05.2013 16:13

Подскажите, а можно ли сделать что-нибудь подобное используя ДВА android-телефона? Один чтобы был "мозгом" машинки, а другой пультом управления? И сможет ли андроид телефон управлять колесами через разъем micro-usb?

talibanich 25.05.2013 07:14

Можно

Тщерук 29.05.2013 16:19

Просто преследуются цели оставить только корпус, колеса, аккумулятор и моторы подключенные к android-телефону(мозгу) через micro-usb, и программное обеспечение, ну и соответственно второй android-телефон.

Или потребуются еще какие-либо микросхемы?

Евгений 13.10.2013 20:08

Подскажите пожалуйста как осуществляется поворот и разворот на 4x4 платформе? Левые и правые колеса крутятся в разные стороны? Или просто достаточно того, что одна сторона будет крутить?

[talibanich](#) 14.10.2013 08:17

И так и так. Только в случае останова одной стороны, колеса будут скользить по полу, что на шершавой поверхности может доставить некоторые неудобства.

Евгений 14.10.2013 20:53

Спасибо! А частота вращения одинаковая должна быть у разных сторон?

[talibanich](#) 15.10.2013 08:15

В статье же все есть.