



Конструктор содержит все компоненты, которые используются в реальных современных **электромобилях**, в том числе и **роботизированных**.

При работе с данным набором можно использовать 4 языка программирования — от самого простого графического **Graph Z** до применяемого профессионалами **C++**.



Три способа управления электромобилем от смартфона или по закачанной в его «мозги» программе, например, автоматической парковки в гараж или объезд препятствия. Конструктор будет интересен не только детям и их родителям, но и педагогам.

Это реальная помощь в образовании.

Пожалуйста, прежде чем приступить к выполнению проектов, внимательно прочитайте информацию в разделах «ВВЕДЕНИЕ», «ОПИСАНИЕ ДЕТАЛЕЙ» и «УСТАНОВКА ПРОГРАММ».

Это поможет избежать поломок, сэкономит много времени и нервов. Предлагаемые QR-коды позволят увидеть работу схемы или лучше понять принципы работы.



Хотелось бы выразить глубокую признательность людям, вложившим душу в осуществлении этого проекта: Анатолию Вожадеву, Марии Коршуновой, Владиславу Зыбенко, Чжану Тяньго.

A handwritten signature in black ink, appearing to read 'Андрей Бахметьев'.

С уважением,
Андрей Бахметьев

ВВЕДЕНИЕ	4	• Сигналы спецтехники	36
ПЕРЕЧЕНЬ ДЕТАЛЕЙ	6	• Поворотники	38
ОПИСАНИЕ ДЕТАЛЕЙ		• Сигнализация на разрыв / замыкание	40
• Провода.....	10	• Сигнализация на свет	42
• Фоторезистор	11	• Сигнализация на движение.....	44
• Монтажная плата.....	11	• Тревожная кнопка	46
• Светодиоды	12	• Управление фарой.....	48
• Резисторы	12	• Управление зуммером.....	50
• Аккумулятор.....	13	• Одноклавишное пианино.....	52
• Мотор-редуктор.....	14	• Адаптивная фара	54
• Серво-привод	15	• Дистансвокс с кнопкой	56
• Драйвер электродвигателей	16	• Парктроник	58
• Кнопки	16	• Датчик удара	60
• Ультразвуковой дальномер	17	• Регулировка звуковых сигналов.....	62
• Модуль Bluetooth	18	• Регулировка звуковых и световых сигналов.....	64
• Переменный резистор	19	• Ручное управление световыми сигналами.....	66
• Модуль Ардуино	20	• Способы управления.....	68
ОПИСАНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	22	• Смешанное управление	70
УСТАНОВКА ПРОГРАММ	24	• Блок управления	72
ПРОЕКТЫ		2. УПРАВЛЕНИЕ МОТОРАМИ	
1. УПРАВЛЕНИЕ СВЕТОВЫМИ И ЗВУКОВЫМИ СИГНАЛАМИ		• Управление мотор-редуктором.....	74
• Автоматическое управление светодиодом.....	30	• Вариатор	76
• Управление двумя светодиодами.....	32	• Индикатор к вариатору.....	77
• Простая музыка.....	34	• Калибровка серво-привода.....	78
		• Ручное управление серво-приводом.....	80
		• Аварийная разблокировка дверей	81
		• Автомобильные дворники	82
		• Семь режимов	84

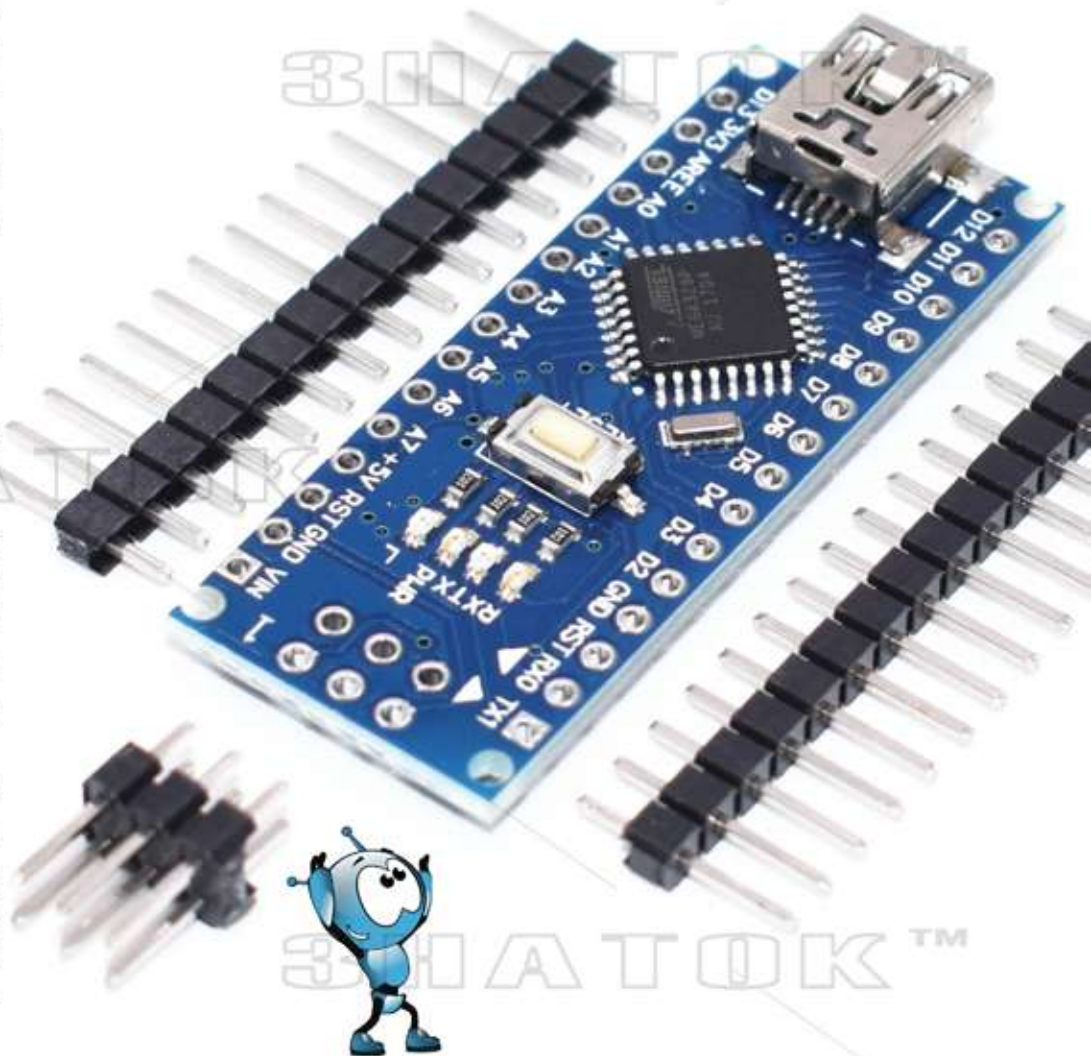
3. ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ ПО BLUETOOTH

- Установка программы управления на мобильное устройство Android86
- Настройка модуля Bluetooth для работы.....86
- Дистанционное управление светодиодом87
- Дистанционное управление пьезоизлучателем88
- Дистанционное управление несколькими устройствами89
- Дистанционная сигнализация на разрыв.....90
- Дистанционная сигнализация на свет.....91
- Дистанционная сигнализация на движение92
- Дистанционное управление серво-приводом.....93

4. СБОРКА ЭЛЕКТРОМОБИЛЯ94

- **УПРАВЛЕНИЕ ЭЛЕКТРОМОБИЛЕМ.**
Установка программы управления на мобильное устройство Android.....95
- Настройка электромобиля96
- Указатели поворота98
- Указатели поворота со звуковым сигналом99
- Автоматические фары.....100
- Безопасная езда.....101
- Парковка в гараж.....102
- Лабиринт.....103
- Дистанционное управление электромобилем.....104
- Электромобиль с поворотниками и звуковым сигналом.....105
- Электромобиль с поворотниками и фарой.....106
- Мобильный измеритель расстояния.....107

ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ C++108



ВВЕДЕНИЕ

Первый электромобиль появился раньше привычных для нас автомобилей с бензиновыми двигателями. В 1828 году была сделана игрушечная тележка на электрической тяге, а уже в 1839 году создана электро-тележка, которая могла перевозить взрослого человека. Автомобиль на бензине смог это сделать только в 1870 году.

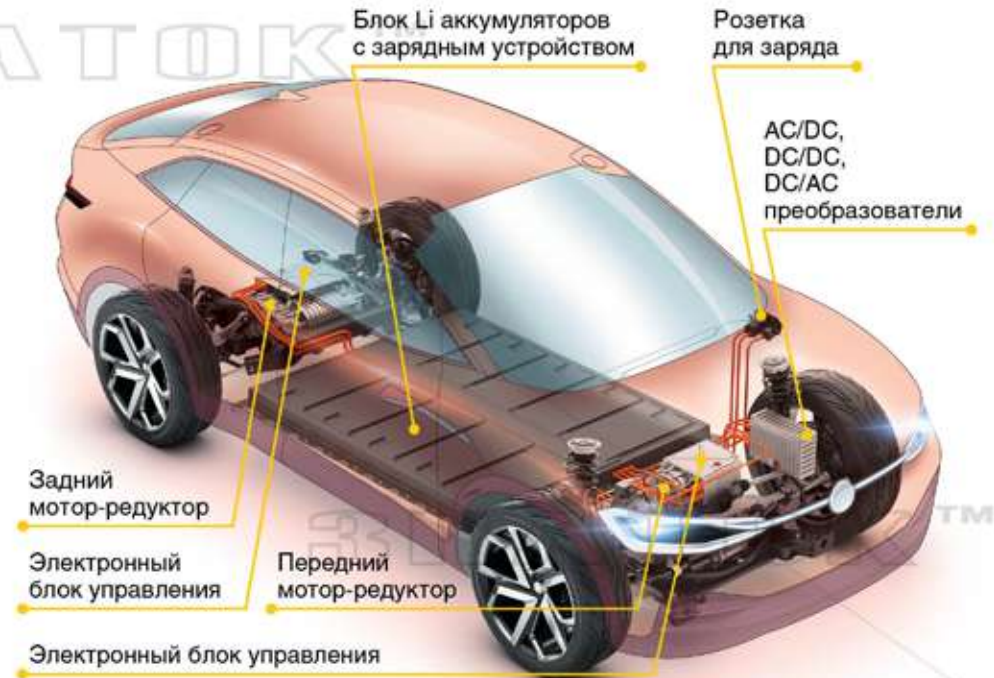


В 1899 году электромобиль преодолел скорость 100 км/ч, а автомобиль с бензиновым двигателем 19 км/ч. Но потом автомобиль на электротяге надолго уступил пальму первенства бензиновым и дизельным машинам. Сейчас, с появлением новых технологий, интерес к электромобилям вырос многократно.

Почти все электромобили состоят из следующих основных блоков: мощные аккумуляторные батареи со схемой зарядки, мотор-редукторы (от 1 до 4), серво-приводы, электронные «мозги», которые управляют скоростью, торможением, системами активной безопасности, автономными режимами езды, звуком, светом, зарядом батарей и пр. Ну и, естественно, колёса, подвеска, рулевое управление, сиденья, панель приборов, фары... Практически всё это есть в нашем наборе.

Несмотря на множество достоинств электромобиля, еще есть ряд сложностей, ограничивающих их применение: ограниченный пробег без подзарядки, скорость заряда аккумуляторов и их невысокий срок службы, количество зарядных станций, проблемы с утилизацией аккумуляторов, содержащих ядовитые вещества. Отдельная история с источниками электроэнергии для заряда — это электростанции, которые, в отличие от самого электромобиля, очень даже портят окружающую среду.

Если вас заинтересует эта тема, поверьте, тут есть над чем работать!



КОНСТРУКТОР АБСОЛЮТНО БЕЗОПАСЕН И ПРОСТ В ОБРАЩЕНИИ, НО ЧТОБЫ ОН ПРОСЛУЖИЛ ВАМ КАК МОЖНО ДОЛЬШЕ, СЛЕДУЕТ СОБЛЮДАТЬ НЕКОТОРЫЕ ПРАВИЛА:

- **Соблюдайте полярность!** Ряд элементов имеют в своей маркировке знаки «+» и «-». Несоблюдение полярности может сделать схему неработоспособной или повредить электронные компоненты!
- При сборе схемы надавливайте не на середину детали, а по краям — в точках крепления. А когда монтажная плата уже находится на стойках, при установке на неё деталей, придерживайте край монтажной платы снизу (как указано на фотографиях).
- Собирайте только те схемы, которые описаны в данной инструкции.
- Никогда не подсоединяйте схемы конструктора к электрическим сетям в вашем доме.
- Большинство проблем в электрических цепях связано с неправильной сборкой, всегда внимательно проверяйте, что ваша цепь действительно соответствует тому, что изображено на рисунке. Удостоверьтесь, что все соединения надёжно защелкнуты.
- Всегда отключайте аккумулятор, если он, или какой-то элемент схемы стал сильно нагреваться!

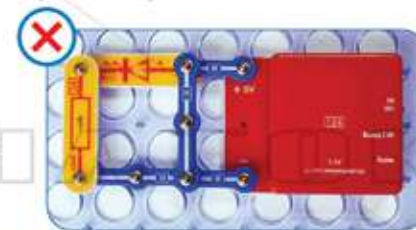


НЕ ДОПУСКАЙТЕ КОРОТКОГО ЗАМЫКАНИЯ БАТАРЕЙ!

Несмотря на то, что аккумулятор 124 из нашего набора имеет защиту от короткого замыкания, никогда не делайте так, как

указано на примерах ниже и других похожих ситуациях — когда выводы «+» и «-» соединяются напрямую при помощи про-

водов или выключателя. Всегда должна быть какая-то нагрузка — светодиод, резистор, электродвигатель.










Производитель не несёт ответственности за повреждения деталей, вызванные несоблюдением техники безопасности и неправильным их подключением.

ПЕРЕЧЕНЬ ДЕТАЛЕЙ

Дизайн деталей может отличаться от приведённых в таблице, но это никак не отражается на работе схем

Код	Наименование элемента	Кол-во	Изображение
1	Провод с 1 контактом	8	
2	Провод с 2 контактами	16	
3	Провод с 3 контактами	5	
4	Провод с 4 контактами	3	
5	Провод с 5 контактами	2	
6	Провод с 6 контактами	2	
7	Провод с 7 контактами	1	
11	Пьезоизлучатель	1	
16	Фоторезистор	1	

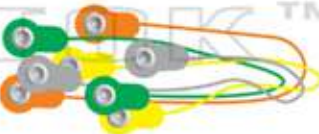


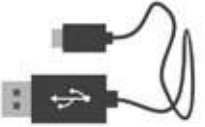


Код	Наименование элемента	Кол-во	Изображение
30	Резистор 100 Ом	2	
31	Резистор 1 кОм	1	
32	Резистор 5.1 кОм	1	
33	Резистор 10 кОм	1	
34	Резистор 100 кОм	1	
47	Фара	1	
89	Ультразвуковой дальномер	1	

Код	Наименование элемента	Кол-во	Изображение	Код	Наименование элемента	Кол-во	Изображение
98	Bluetooth модуль	1		124	Li-ion аккумулятор 7.4V/5V	1	
101	Драйвер электродвигателей	1			Серво-привод	1	
111	Модуль Arduino Nano	1			Мотор-редуктор	1	
113	Желтый светодиод	2			Платформа	1	
118	Переменный резистор 5 кОм	1			Монтажная плата	1	

Код	Наименование элемента	Кол-во	Изображение
	Стойки	4	
	Винт M2.5 x 10мм	8	
	Винт M3 с потайной головкой	2	
	Винт M3 x 8мм	22	
	Винт M3 x 13мм	4	
	Подшипник фланцевый	2	
	Гайка M3	20	
	Контргайка M2.5	2	
	Контргайка M4	2	

Код	Наименование элемента	Кол-во	Изображение
	Уголок-крепление серво-привода	2	
	Шестерня большая и малая	2	
	Стойка металлическая h22xM3	4	
	Площадка крепления переднего колеса	2	
	Качалка серво-привода (с винтом)	1	
	Рычаги	2	

Код	Наименование элемента	Кол-во	Изображение
	Несущая скоба	1	
	Ось задняя	1	
	Отвертка	1	
	Ключ шестигранный	1	
	Ключ универсальный	1	
	Колесо заднее (в сборе)	2	
	Колесо переднее (в сборе)	2	

Код	Наименование элемента	Кол-во	Изображение
	Провода гибкие с двумя клеммами (оранжевый, зелёный, серый, жёлтый)	4	
	USB-кабель для заряда аккумулятора*	1	
	Кабель подключения аккумулятора к схемам	1	
	Кабель USB-miniUSB для подключения к компьютеру	1	
	USB-флэш-накопитель (флэшка)	1	
	Кнопка малая	2	

* AC-DC адаптер в комплект не входит.
Для заряда аккумулятора вы можете воспользоваться любым сетевым адаптером с USB выходом 5V/1A или 2A.

ПРОВОДА



Электрические провода предназначены для соединения элементов схемы между собой и не влияют на характеристики цепи.

В данном наборе используются три типа проводов.

- **ЖЕСТКИЕ** синие провода имеют обозначение, указывающее на длину этого провода и количество контактов на нём.

Одиночные провода-шайбы предназначены для электрического соединения между слоями схемы. В более редких случаях, для придания конструкции устойчивости, как «ножки».

На схемах такие провода-шайбы не видны и поэтому места, куда они должны быть помещены, указываются вот таким образом:



- **ГИБКИЕ** провода, которые с обеих сторон имеют контакт-кнопку для подсоединения к деталям схемы и жестким проводам.



Чем меньше сопротивление провода, тем лучше, поэтому провода делают из металлов. Но разные металлы имеют разное сопротивление. Наименьшее сопротивление имеет серебро, но делать из него провода очень дорого. Лучшими считаются медные провода, но и это не совсем дешевый металл. Очень часто провода делают из алюминия. Несмотря на страшную дороговизну провода делают и из золота — оно, в отличие от других металлов, не окисляется, что очень важно, например, в космической и военной технике. Но эти провода совсем тонкие.

Из таблицы понятно, почему провода не делают из железа. Чем длиннее провод, тем выше его сопротивление. Сопротивление провода ещё зависит и от его

Металл	Удельное сопротивление
Серебро	0,015
Медь	0,018
Золото	0,023
Алюминий	0,029
Вольфрам	0,055
Железо	0,098



толщины — чем тоньше провод, тем больше у него сопротивление. Именно поэтому провода для запуска автомобиля от чужого аккумулятора делают не очень длинными, толстыми и из меди.

ФОТОРЕЗИСТОР



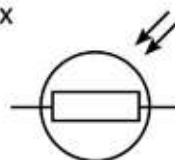
У фоторезистора из данного набора сопротивление в темноте — 20 МОм, а при очень ярком освещении — 150 Ом.

Его ещё называют светочувствительный резистор — полупроводниковый элемент, сопротивление которого зависит от освещенности. Под действием света сопротивление фоторезистора уменьшается. При этом сила тока в цепи возрастает, достигая значения, достаточного для включения или работы какого-либо устройства.

Область применения

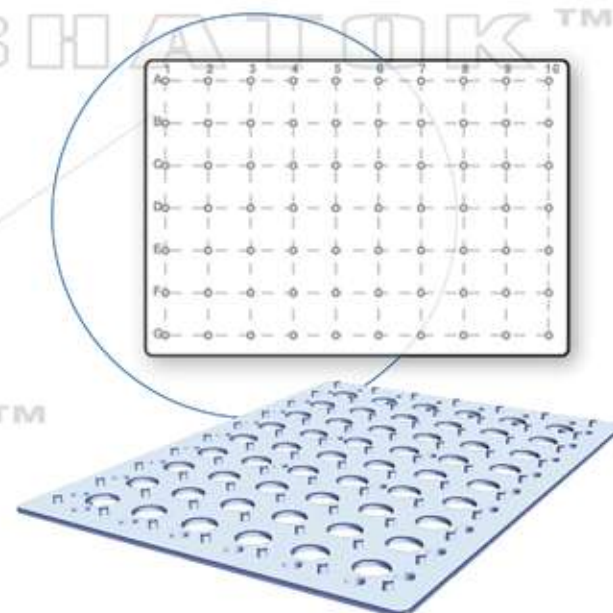
Датчики прохождения человека, противопожарные системы, измерители температуры (по яркости накала), автоматические фонари, охранная сигнализация, измерители уровня освещенности и т.д.

На электрических схемах фоторезистор обозначается вот так:



Если вы хотите узнать больше про резисторы, то это можно сделать, приобретя набор «Для школы и дома», **Практическое занятие №17 «Фоторезистор»** или «Супер набор №1» (подробности на сайте www.znatok.ru)

МОНТАЖНАЯ ПЛАТА



Монтажная плата — платформа для сборки на ней схем из деталей. В отличие от монтажных плат из других наборов ЗНАТОК, здесь специальные крепления имеются с двух сторон. На верхней части находятся выступы для крепления элементов, а на нижней — держатели для мотор-редукторов и переходника для крепления заднего колеса.

СВЕТОДИОДЫ

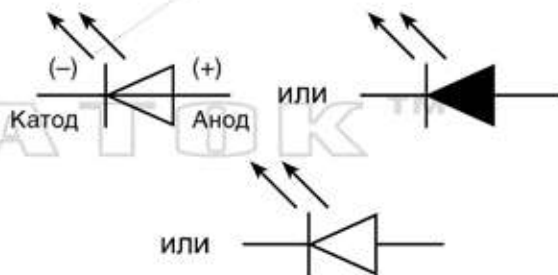


Светодиод (светоизлучающий диод или LED — Light Emitting Diode) — это не лампочка, это электронный полупроводниковый прибор с p-n переходом, который начинает светиться при прохождении через него электрического тока.

При подключении необходимо соблюдать полярность!

При включении светодиода необходимо использовать токоограничивающий резистор. В нашем светодиоде такой защитный резистор уже встроен в корпус, поэтому в некоторых схемах возможно подключение к источнику питания напрямую.

На принципиальных схемах светодиод обозначается так:



РЕЗИСТОРЫ



Резистор (от латинского resisto — сопротивляюсь) — элемент оказывающий сопротивление электрическому току, устанавливается с целью регулирования тока и напряжения. Чем выше сопротивление резистора, тем меньший ток по нему протекает.

Меняйте резисторы в схеме, и вы увидите, что чем меньше сопротивление резистора, тем ярче горит светодиод. При подключении резистора 100k светодиод будет еле «тлеть».

Сопротивление измеряется в омах (Ом или Ohm) с приставкой-множителем «кило», «мега», «гига» и т.д.

Ом, кОм, МОм, ГОм... = Ω , k, M, G...

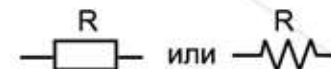
1000 Ом = 10^3 Ом = 1 кОм,

1000 кОм = 10^6 Ом = 10^3 кОм = 1 МОм

Основные характеристики: номинал сопротивления, точность, рассеиваемая мощность, температурный коэффициент, способ монтажа на плате.

Область применения: практически во всех электрических цепях. Если не в виде отдельного элемента, то внутри интегральной схемы.

На электрических схемах резисторы могут обозначаться двумя способами:



АККУМУЛЯТОР



В набор входит Li-ion аккумулятор, который имеет два выходных напряжения: стабилизированные 5V и силовые 7.4V. Для подключения напряжения 7.4V используется специальный кабель.

Аккумулятор имеет встроенную защиту от короткого замыкания. При случайном замыкании «плюса» и «минуса» ток перестает течь. После устранения короткого замыкания, напряжения тоже не будет. Чтобы вернуть аккумулятор в рабочий режим, выключите его (OFF) и на несколько секунд подключите к зарядному устройству.

Зарядка аккумулятора. При заряде аккумулятора переключатель должен находиться в положении OFF. В наборе имеется кабель для заряда от AC-DC адаптера (сам адаптер в набор не входит, но вы можете воспользоваться любым сетевым адаптером, имеющим выход 5V/1A или 2A). Подключите кабель к модулю 124 (разъем Заряд). Включите зарядное устройство в сеть 220V. Когда индикатор заряда горит красным — идёт зарядка, когда индикатор погаснет — аккумулятор полностью заряжен. Время зарядки около 2-х часов. После чего рекомендуется отключить зарядное устройство от сети.

Во время зарядки аккумулятор использовать нельзя!

ВНИМАНИЕ!

- Ещё раз обращаем внимание на недопустимость короткого замыкания аккумулятора! Нельзя напрямую соединять «+» и «-» аккумулятора.
- Так как для зарядки аккумулятора требуется напряжение сети 220V, рекомендуем эту операцию осуществлять под присмотром взрослых.
- Условия хранения: беречь от огня и высокотемпературного воздействия, избегать попадания прямых солнечных лучей, хранить полностью заряженным в сухом прохладном месте. Перед длительным хранением обязательно зарядить! Если Li-ion аккумулятор оставить на хранение с напряжением, ниже 2.5 V сроком на три месяца или более, происходит невосстанавливаемая потеря его емкости!
- Содержит вредные вещества. Утилизация аккумулятора должна происходить по всем правилам, призванным сохранять окружающую среду!

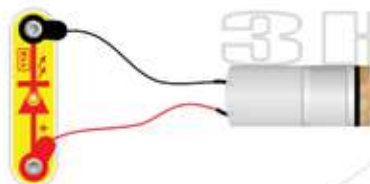
МОТОР-РЕДУКТОР



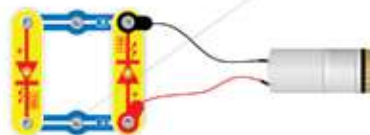
Мотор-редуктор представляет собой устройство, состоящее из мотора (электродвигателя) и редуктора. Такое сочетание применяется для уменьшения частоты вращения вала и увеличения крутящего момента при передаче движения от двигателя на рабочее оборудование, например, колеса.

ПРОЧИТАЙТЕ – ПОПРОБУЙТЕ – ЗАПОМНИТЕ!

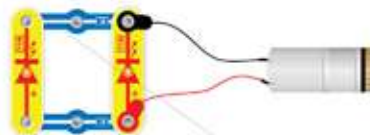
Установите на вал мотор-редуктора большую шестерню. Последовательно соберите все нижеприведённые схемы. Возьмите мотор-редуктор в одну руку, а другой рукой вращайте шестерню. Чем быстрее крутится вал двигателя, тем большее напряжение вырабатывается, но эти значения не беспредельны.



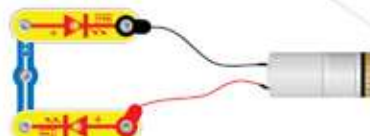
- Мотор-редуктор, как и обычный электродвигатель, может вырабатывать напряжение. Это напряжение имеет полярность, зависящую от направления вращения. Вращайте шестерню в разных направлениях.



- Светодиод тоже имеет полярность и её надо соблюдать при подключении. Вращая вал мотор-редуктора в разных направлениях, можно убедиться, что каждый раз будет гореть только один из двух светодиодов.



- Для свечения двух параллельно включенных светодиодов требуется такое же напряжение, что и в предыдущей схеме, но немного больший ток, что практически не заметно на усилки, т.к. ток потребления светодиодов совсем маленький.



- Для свечения двух последовательно включенных светодиодов требуется гораздо большее напряжение, чем в предыдущей схеме и, следовательно, усилия, прикладываемые для вращения вала мотор-редуктора.



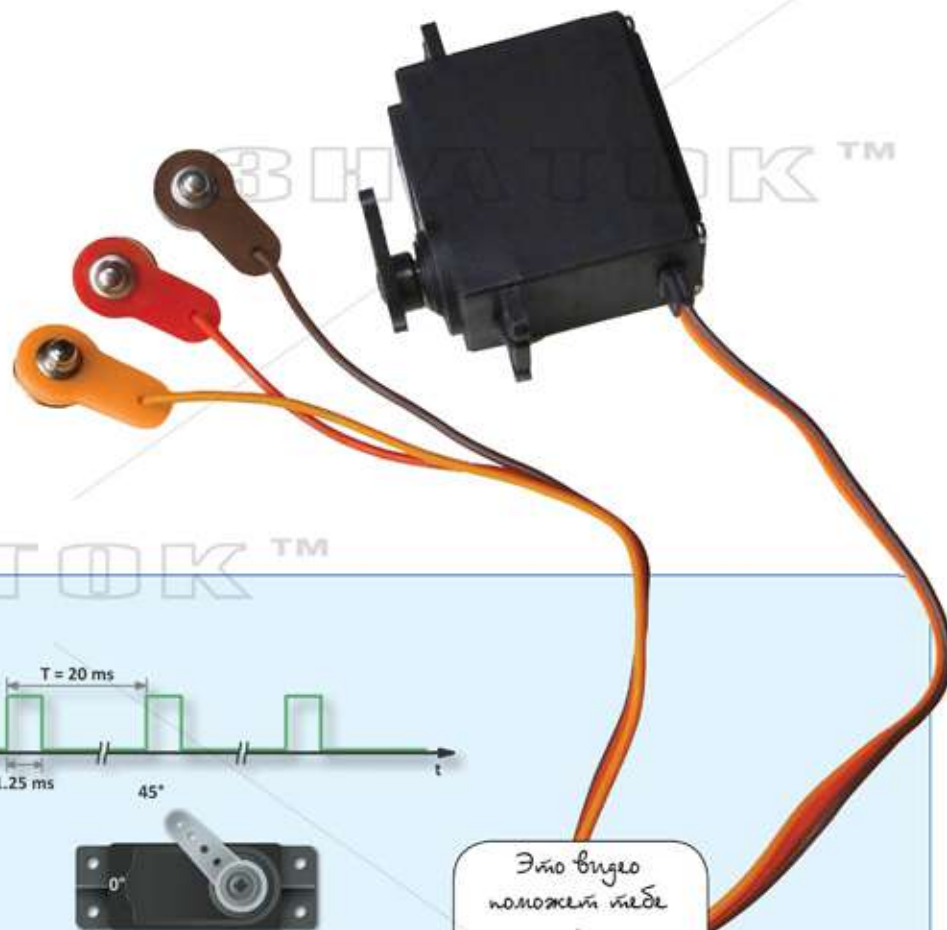
- Два светодиода, включенные «навстречу» друг другу, не проводят электрический ток, поэтому вращение мотор-редуктора в любом направлении не приведет к их включению.

СЕРВО-ПРИВОД



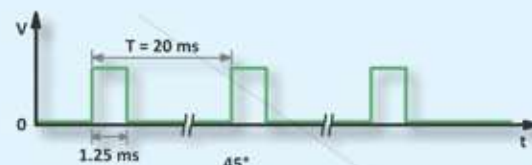
Его часто называют просто **серво** (servo). В зависимости от сигнала, приходящего на вывод «Управление» вал серво-привода может поворачиваться и фиксироваться в определённом положении.

В нашем случае сигналом управления является импульс заданной длительности. Вал нашего серво-привода может поворачиваться на 180° .



КАК ЭТО РАБОТАЕТ

Для полного поворота на 180° необходимо на вход управления подать импульс, длительность которого t меняется от 1 до 2 миллисекунд (ms) при постоянном периоде T . Когда на вход управления приходят импульсы длительностью 1ms, рычаг, закрепленный на валу, находится в положении 0° . По мере увеличения длительности импульса рычаг, с небольшой задержкой, будет поворачиваться. Так, при длительности импульса 1.25ms рычаг повернется на 45° , при 1.5ms — 90° , при 2ms — 180° .

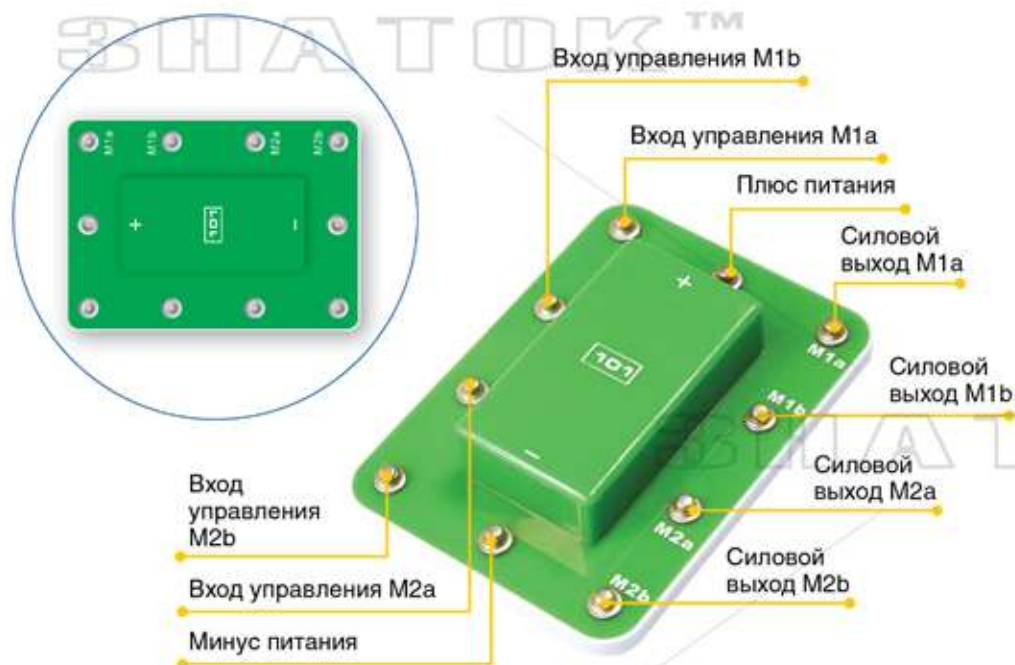


УПРАВЛЕНИЕ ()
 ПИТАНИЕ (+4,8 ... 7,2V)
 ОБЩИЙ (GND)

Это видео
поможет тебе
понять, как
он работает



ДРАЙВЕР ЭЛЕКТРОДВИГАТЕЛЕЙ



Модуль Arduino может выдавать у себя на выходах максимальный ток, не более 40 мА, чего недостаточно для работы электродвигателя и мотор-редукторов. Для увеличения мощности используются специализированные драйверы. В зависимости от входных сигналов управления, поступающих от микроконтроллера, электродвигатели, подключенные к силовым выходам драйвера, могут менять скорость и направление вращения.

Выводы M1 относятся к первому мотор-редуктору, выводы M2 относятся ко второму мотор-редуктору.

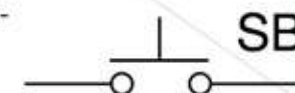
КНОПКИ



Кнопки могут иметь разную форму и размер, но все они пропускают ток, только когда на них нажимают — как в дверном звонке или лифте.

Кнопки не влияют на характеристики электрической цепи, но могут изменять её функции, отключая или подключая к схеме элементы.

На электрических схемах кнопки обозначаются так:



УЛЬТРАЗВУКОВОЙ ДАЛЬНОМЕР



Несмотря на весёлую внешность нашего дальномера, у него очень серьёзная начинка.

Вид на излучатель и приёмник



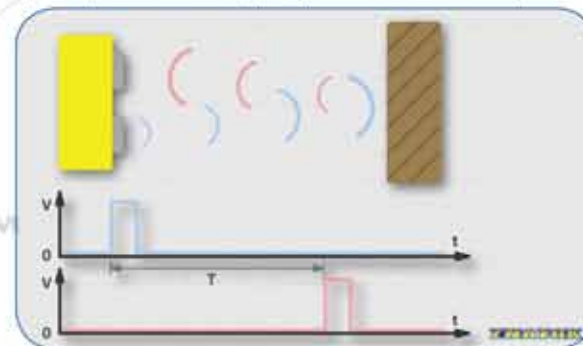
спереди

сзади

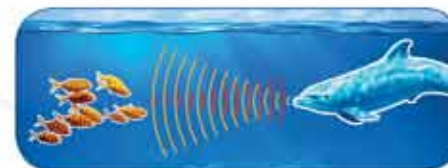
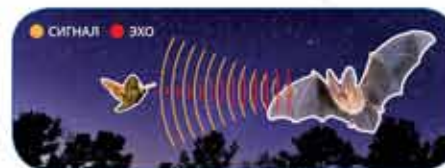
Принцип работы ультразвукового дальномера



Излучатель посылает ультразвуковой сигнал, который отражается от объекта, возвращается назад и улавливается приемником. По разнице во времени между посылаемым сигналом и отраженным (эхо) рассчитывают расстояние. Чем больше разница во времени, тем больше расстояние.



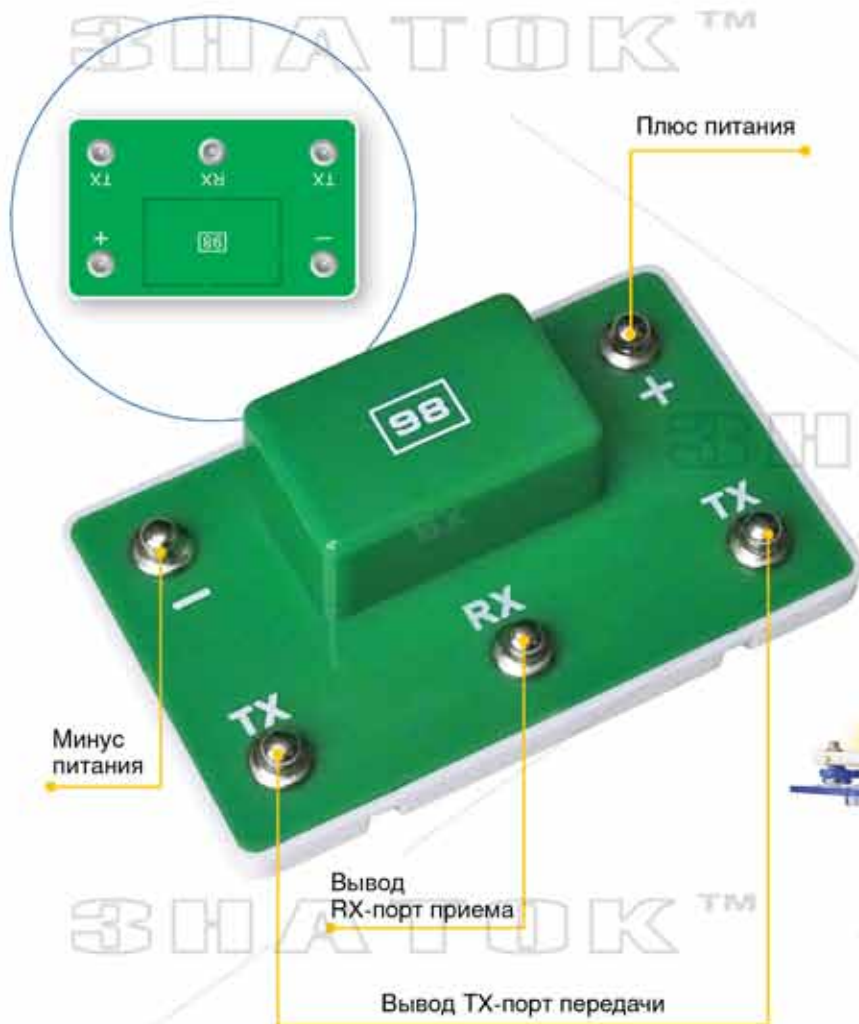
Принцип работы был подсмотрен у летучих мышей и дельфинов, которые даже в полной темноте отлично ориентируются, не натываются на препятствия и находят добычу.



Похожий принцип используется в эхолотах, гидролокаторах, радарах, автономных роботах, автомобильных парктрониках.



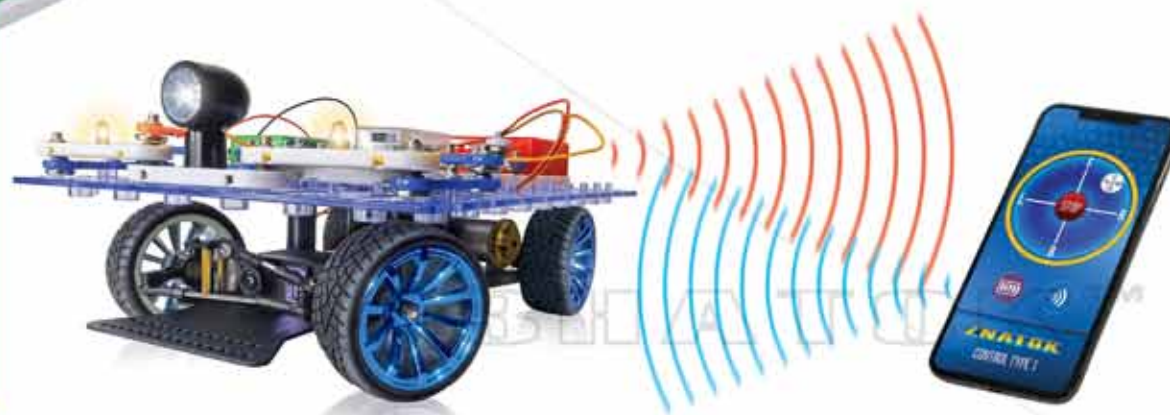
МОДУЛЬ BLUETOOTH



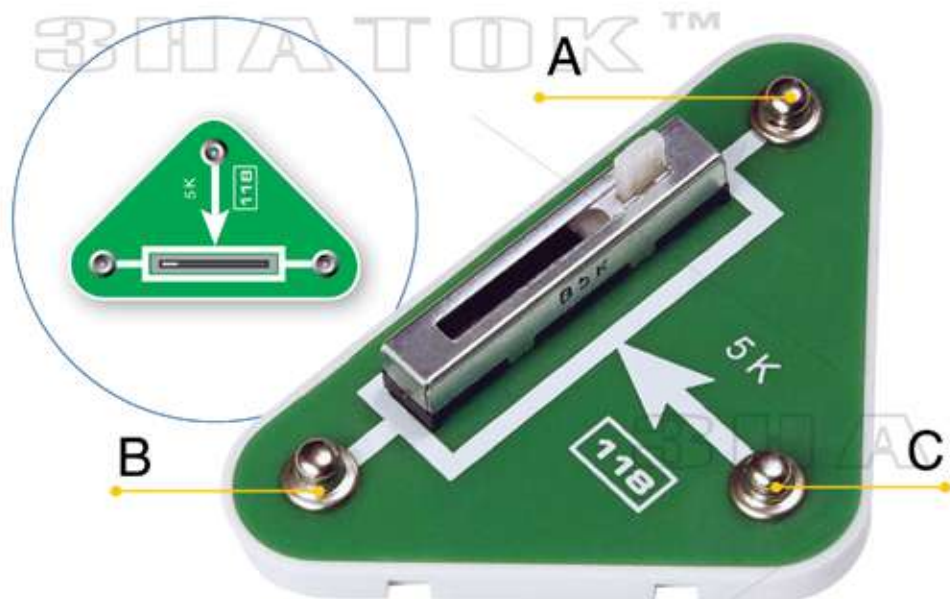
В нашем наборе **модуль Bluetooth 98** предназначен для обеспечения обмена информации между модулем Arduino **111** и вашим смартфоном, у которого должен быть активирован режим связи по Bluetooth. При помощи этого модуля смартфон может получать информацию о вашем устройстве и передавать команды управления на устройство.

Следует обратить внимание, что в наших проектах вывод TX (передатчик) модуля Bluetooth **98** должен всегда соединяться с выводом RX (приемник) модуля Arduino **111**, и наоборот.

Радиус действия **50 метров**.
Рабочая частота **2.4 ГГц**.



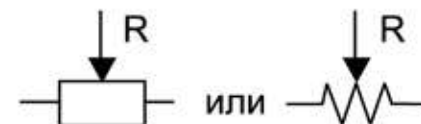
ПЕРЕМЕННЫЙ РЕЗИСТОР



Переменный резистор — предназначен для регулирования силы тока или напряжения в электрической цепи. Имеет три вывода **A–B–C**. Сопротивление между точками **A–B** постоянное (в нашем случае 5 кОм). При перемещении ползунка (движка) сопротивление между точками **A–C** и **B–C** меняется. Когда ползунок находится у точки **A**, то сопротивление **A–C** равно 0, а сопротивление **B–C** максимально — 5 кОм. Когда ползунок у точки **B** — наоборот.

Сопротивление измеряется в омах (см. раздел «Резисторы»).

На электрических схемах переменные резисторы могут обозначаться двумя способами:



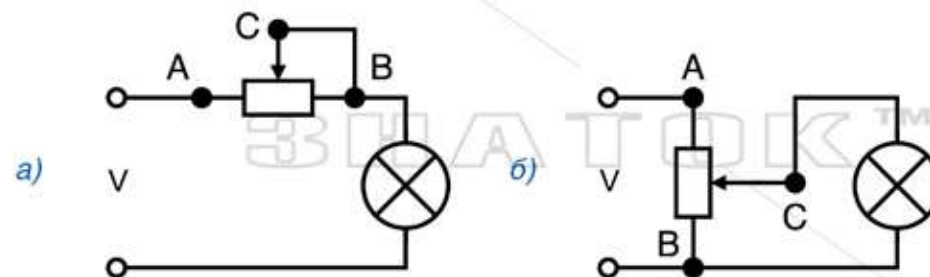
Основные характеристики

Номинал сопротивления, разрешающая способность, рассеиваемая мощность, температурный коэффициент, способ изменения сопротивления.

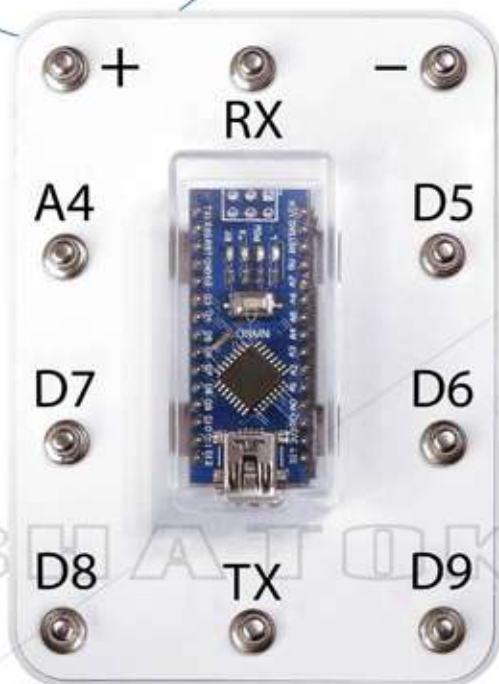
Область применения

Переменные резисторы применяются для регулировки яркости света, громкости звука, скорости вращения и т.п.

Переменный резистор имеет две основные схемы включения: **потенциометрическую** (рис. а) и **потенциометрическую** (рис. б). В качестве нагрузки показана лампа.



МОДУЛЬ ARDUINO



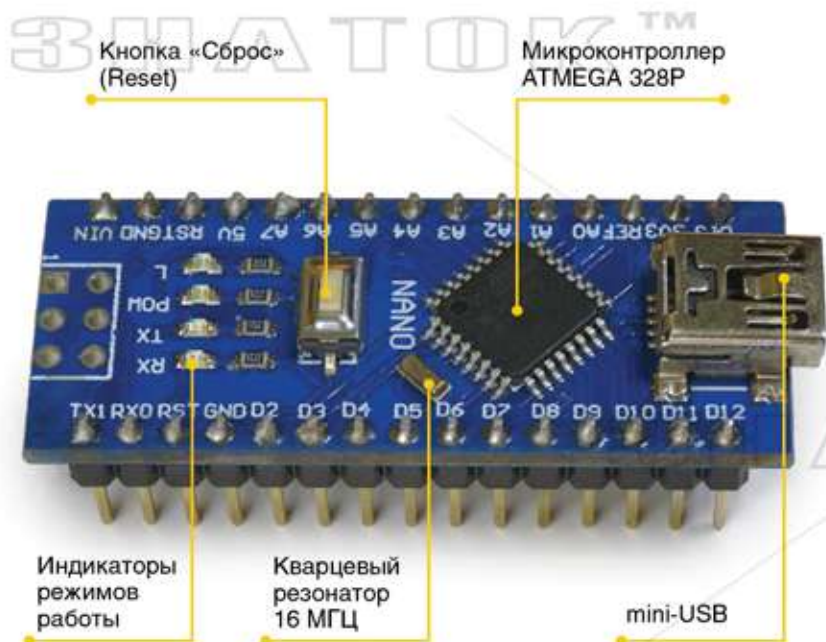
Модуль **111** построен на базе **Arduino Nano**, который, в свою очередь, состоит из микроконтроллера ATMEGA 328P семейства AVR и дополнительных элементов, необходимых для работы.

Микроконтроллеры бывают разные и могут выполнять задачи, как маленький компьютер. Микроконтроллер из нашего набора может выполнять следующие функции:

- Выдавать информацию. По заданной программе управлять внешними устройствами — светодиодами, лампами, электродвигателями. Воспроизводить звуки, показывать время.
- Принимать информацию. По заданной программе обрабатывать информацию от внешних устройств и воспроизводить её на индикаторах. В этом режиме возможно построение различных измерительных приборов.
- Принимать информацию, анализировать её и, на основе анализа, выдавать информацию. На этом принципе построены многие бытовые приборы и роботы.

Описание выводов модуля 111:

«+»	Плюс питания
«-»	Минус питания
TX	Передача асинхронных выходных данных
RX	Приём асинхронных входных данных
A4	Аналоговый вход. По сути, это вход аналого-цифрового преобразователя (АЦП), который преобразует входное напряжение в коды.
D5	Цифровой вход/выход с функцией ШИМ
D6	Цифровой вход/выход с функцией ШИМ
D7	Цифровой вход/выход
D8	Цифровой вход/выход
D9	Цифровой вход/выход с функцией ШИМ
mini-USB	Разъём для соединения с компьютером



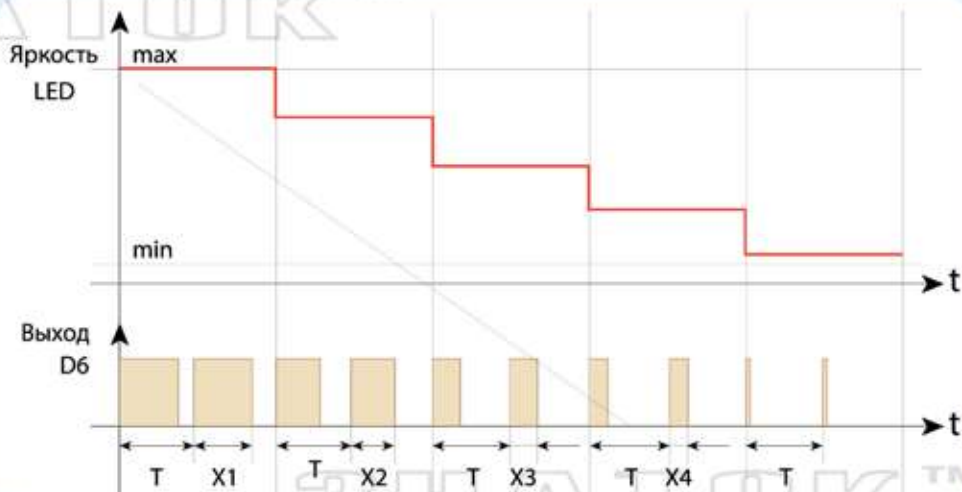
Как видно из фотографии, в модуле **111** используются далеко не все выводы **Arduino Nano**. Но освоив и поняв принцип работы тех выводов, что имеются на модуле **111**, вы легко сможете в своих проектах использовать все остальные.

КАК ЭТО РАБОТАЕТ



Широтно-Импульсная Модуляция (**ШИМ**) на английском Pulse-Width Modulation (**PWM**) — процесс управления мощностью, подводимой к нагрузке, путём изменения скважности импульсов, при постоянной частоте.

На графике приведен пример управления яркостью светодиода (LED). Из графика видно, что период импульсов **T** на выходе **D6** постояен, а длительность импульсов **X** меняется. Чем короче импульсы, тем меньшая мощность поступает на светодиод и яркость его уменьшается. Аналогично управляются лампа накаливания или электродвигатель, у которого будет меняться скорость вращения.



Скважность (S) — отношение периода импульсов **T** к длительности импульса **X**, т.е. $S=T/X$.

У сигнала типа «меандр» скважность равняется 2, т.к. период в два раза больше длительности импульса.



Самый простой вариант, разработанный компанией «ЗНАТОК», для перехода от визуального программирования к программированию проектов уже на «серьезном» языке C++\Arduino. Составив программу из простых блоков-действий можно увидеть, как эта же программа выглядит на C++.



Выполнив ряд проектов, вы сможете понимать, как изменение в алгоритме влияет на текст программы и сможете уже самостоятельно писать код программ.

Для получения инструкции по работе с языком Graph Z в данном наборе пройдите по ссылке: <http://znatok.ru/link/?mobile-graphz> или отсканируйте этот QR-код:



Для программирования на языке C++ предлагается использование среды Arduino IDE, включающую в себя библиотеки для работы с микроконтроллерами. Большая часть программ, написанных для различных устройств, была написана на языках C\C++, а использование модулей Arduino может быть первым серьезным шагом в изучении программирования и микроконтроллеров.

```
01_Control_LED-10
1 #define BUTTON_PIN 7
2 #define LED_RED 5
3
4 void setup()
5 {
6   pinMode(LED_RED, OUTPUT);
7   pinMode(BUTTON_PIN, INPUT_PULLUP);
8 }
9
10 void loop()
11 {
12   if (digitalRead(BUTTON_PIN) == LOW)
13   {
14     digitalWrite(LED_RED, HIGH);
15   }
16   else
17   {
18     digitalWrite(LED_RED, LOW);
19   }
20 }
21
```

Все программы предоставленные в этом пособии, которые вы можете найти на флэшке, изначально написаны именно для Arduino IDE, и в книге вы можете найти примеры именно для этого варианта.

ПРИМЕЧАНИЕ: Компания «ЗНАТОК» не занимается разработкой и поддержкой языка программирования C++ и среды программирования Arduino IDE

Для получения инструкции по работе с языком C++ в данном наборе пройдите по ссылке: <http://znatok.ru/link/?mobile-cpp> или отсканируйте этот QR-код:



SCRATCH

Для визуального программирования можно использовать графический язык Scratch и его производные платформы для робототехники, например, S4A (Scratch for Arduino) и mBlock. Данный вариант подойдет для тех, кто только начинает узнавать, что такое алгоритмы, условия, циклы — все конструкции представлены наглядно в виде блоков. Не все проекты в данном наборе могут быть запрограммированы при помощи Scratch, однако имеющиеся позволят разобраться в азах программирования и двигаться дальше.



ПРИМЕЧАНИЕ: Компания «ЗНАТОК» не занимается разработкой и поддержкой языка программирования Scratch, а так же платформ S4A и mBlock

Для получения инструкции по работе с языком Scratch в данном наборе пройдите по ссылке: <http://znatok.ru/link/?mobile-scratch> или отсканируйте этот QR-код:



python™

Python не является основным языком для программирования микроконтроллеров, но его все же можно использовать для управления устройствами. Если вы хотя бы немного знаете данный язык, то это может стать хорошим стартом для изучения работы микроконтроллеров. Не для всех проектов из данного набора можно создать программу на языке Python, но вы всегда можете создать что-то своё!

```
while 1: #Do this forever

    var = raw_input() #get input from user
    print "you entered", var #print the input for confirmation

    if (var == '1'): #if the value is 1
        ArduinoSerial.write('1') #send 1
        print ("LED turned ON")
        time.sleep(1)

    if (var == '0'): #if the value is 0
        ArduinoSerial.write('0') #send 0
        print ("LED turned OFF")
```

Для получения инструкции по работе с языком Python в данном наборе пройдите по ссылке: <http://znatok.ru/link/?mobile-python> или отсканируйте этот QR-код:



УСТАНОВКА ПРОГРАММ

Эти программы, их обновлённые версии, а так же новые программы к данному набору, вы можете найти и бесплатно скачать на сайте www.znatok.ru

USB-флэш-накопитель. Многие называют его просто флэшка. На флэшке из этого набора находятся программы и драйверы, необходимые для подключения модуля 111 (Arduino Nano) к вашему компьютеру. В одной папке находятся файлы для **Windows**, в другой — для **macOS**. Следуйте ниже-приведённым инструкциям.



Для пользователей Windows

1. Вставьте прилагаемую флэшку в USB-разъем своего компьютера.
2. Если у вас автоматически не открылось окно запуска, то откройте ручную папку флэшки (**Этот компьютер — Znatok Arduino**) при помощи Проводника Windows или другого файлового менеджера (напр., TotalCommander).
3. Запустите программу **ZnatokIntro.exe**. Данная программа поможет вам в установке всех необходимых компонентов.

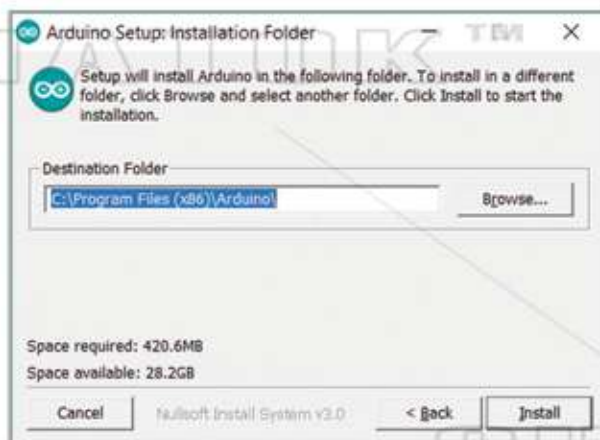


4. Для установки среды программирования **Arduino IDE** нажмите кнопку **Установить Arduino IDE**, если Arduino IDE нужной версии уже была установлена на вашем компьютере, то этот шаг необязателен.

- 4.1. Следуйте инструкциям установщика: на первом появившемся экране нажмите **I agree** («Я согласен»).
- 4.2. Обязательно отметьте пункт **Install USB driver**. Нажмите **Next** («Далее»).



4.3. Выберите путь установки (рекомендуем предложенный). Нажмите **Install** («Установить»).



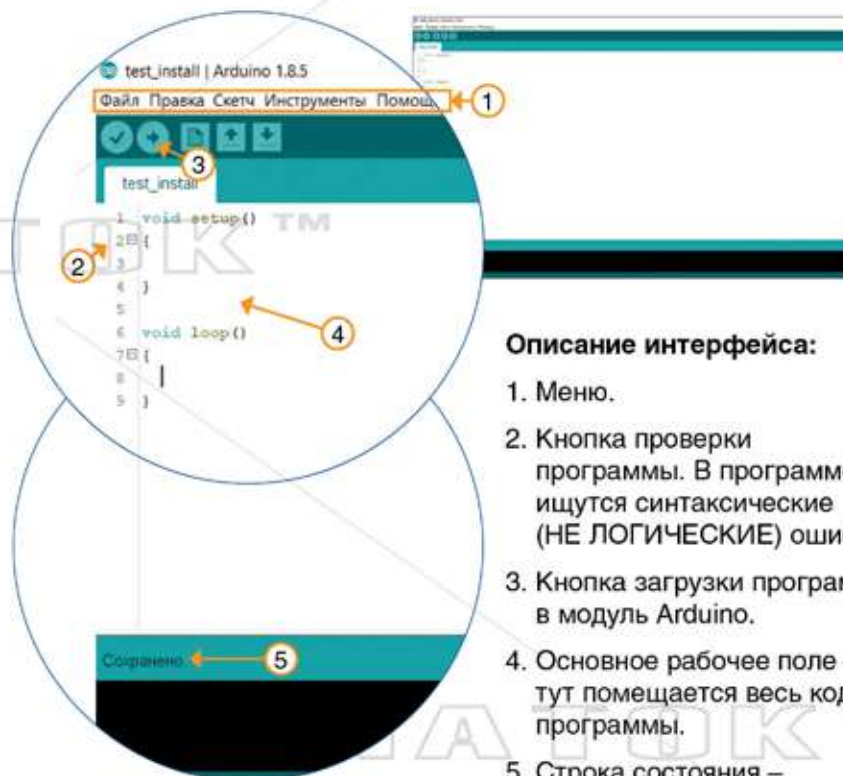
4.5. Дождитесь конца установки, появится надпись **Completed**, нажмите **Close** («Закреть»), после чего на рабочем столе появится ярлык Arduino.

5. Для корректного выполнения лабораторных работ необходимо создать рабочую папку у себя на компьютере, а также скопировать необходимые файлы — для этого нажмите кнопку **Скопировать рабочие файлы**. Вам будет предложено место создания папки, по умолчанию — **Рабочий стол**. Нажмите кнопку **Выбрать**, появится папка **Лабораторные работы Znatok Arduino**, в которой будут находиться все файлы программ. Если вы хотите вручную скопировать файлы программ, то можете найти их на прилагаемой флэшке в папке LaboratoryProjects.

ВНИМАНИЕ: Не рекомендуется открывать файлы программ непосредственно с флэшки!

6. Подключите при помощи USB-кабеля модуль **1 1 1** к компьютеру. Произойдет установка оборудования. Если у вас отключена функция автоматического поиска драйверов, то в ручном режиме их можно найти на прилагаемой флэшке в папке **drivers**.

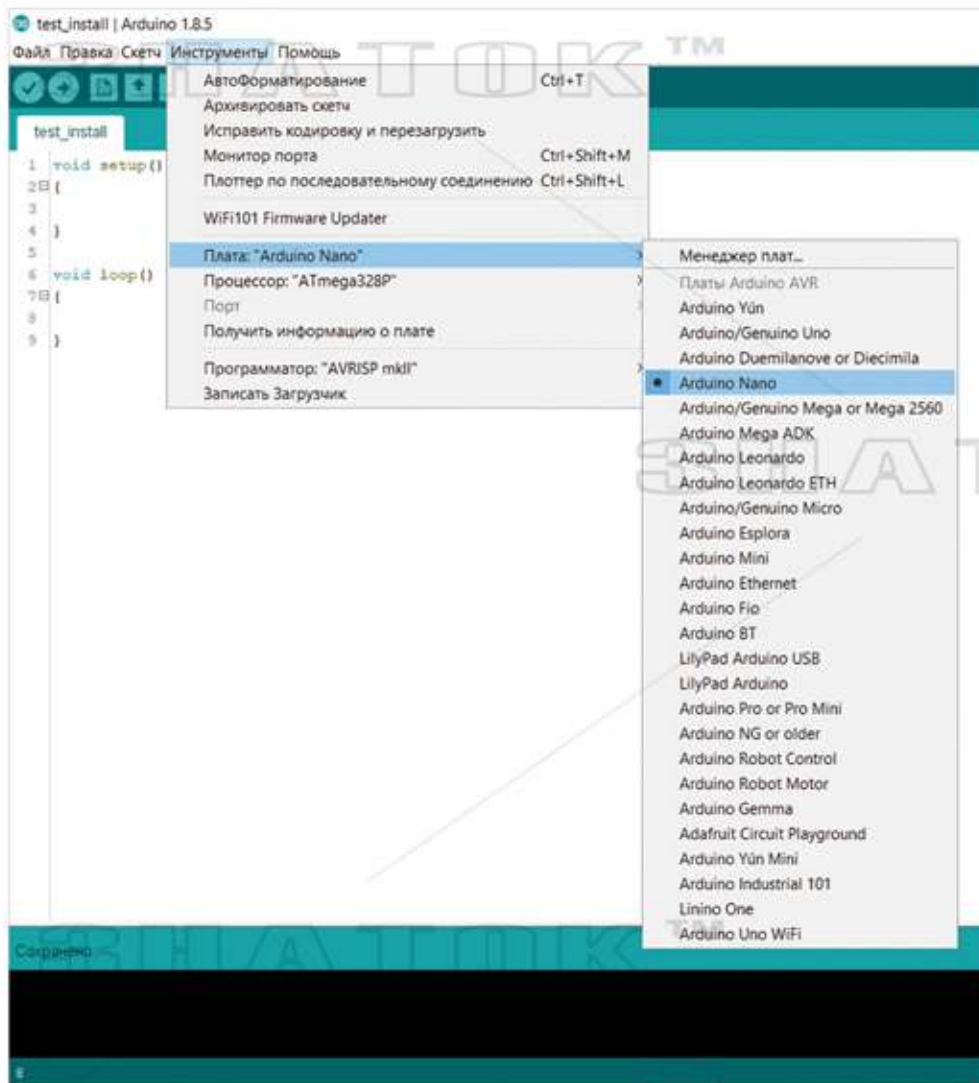
6.1. Запустите Arduino IDE (при помощи ярлыка на рабочем столе).



Описание интерфейса:

1. Меню.
2. Кнопка проверки программы. В программе ищутся синтаксические (НЕ ЛОГИЧЕСКИЕ) ошибки.
3. Кнопка загрузки программы в модуль Arduino.
4. Основное рабочее поле — тут помещается весь код программы.
5. Строка состояния — показывает стадию загрузки программы.

6.2. В меню **Инструменты** выберите подпункт **Плата** и в выпадающем списке выберите **Arduino Nano**.



6.3. В меню **Инструменты** выберите подпункт **Порт** и в выпадающем списке выберите любой подпункт (часто, бывает один), например, COM6.

6.4. В меню **Скетч** выберите подпункт **Загрузка**, после чего начнется загрузка программы на модуль **111** (в данный момент программа тестовая), если все шаги были выполнены верно, то в строке состояния появится сообщение **Загрузка завершена**. Если возникла проблема загрузки в модуль **111**, то попробуйте изменить Порт (см. п.6.3).

7. Для выполнения лабораторной работы, откройте папку **Лабораторные работы Znatok Arduino** на вашем компьютере. Выберите нужную лабораторную работу и откройте программу — у вас откроется Arduino IDE с текстом программы. Если Arduino IDE уже открыта, то вы можете в меню **Файл** выбрать подпункт **Открыть** и выбрать нужную программу.

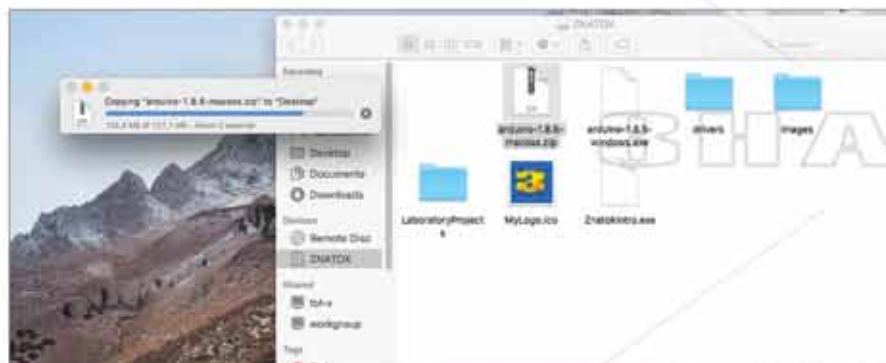
7.1. Для загрузки программы в Arduino необходимо выбрать в меню **Скетч** подпункт **Загрузка** или кликнуть по соответствующей пиктограмме. **НЕ ЗАБУДЬТЕ:** при ошибке загрузки проверьте правильность подключения модуля **111** при помощи USB-кабеля к вашему компьютеру, а так же правильность выбора порта (см. п.6.3).

Для пользователей macOS

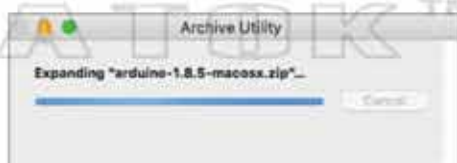
1. Вставьте прилагаемую флэшку в USB-разъем своего компьютера, на рабочем столе у вас появится ярлык флэшки:



2. Скопируйте архив **arduino-1.8.5-macosx.zip**, например, на рабочий стол.



3. Разархивируйте файл **arduino-1.8.5-macosx.zip** в нужное вам место (рекомендуем «Application»), после чего приложение станет доступным для запуска.



4. Скопируйте папку с лабораторными работами **Laboratory-Projects** в любое удобное вам место.

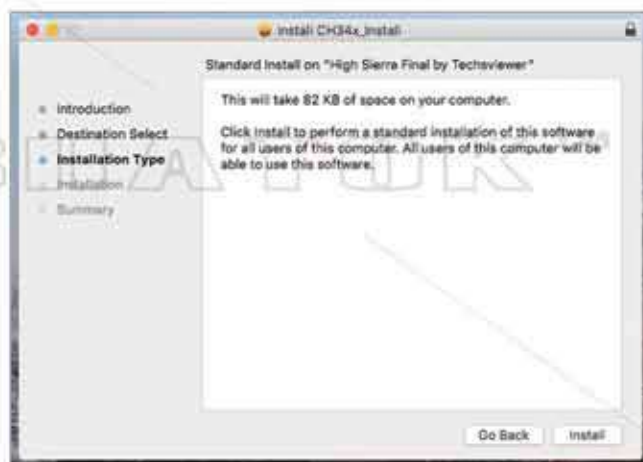


5. В папке **drivers\CH341SER_MAC**, на прилагаемой флэшке, откройте файл установщика:

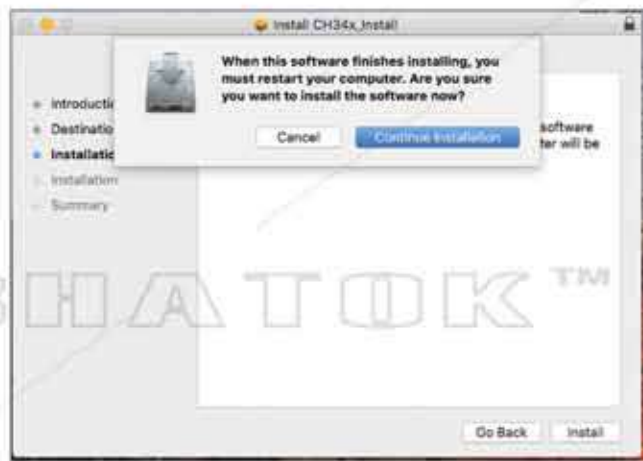


6. Следуйте инструкциям установщика:

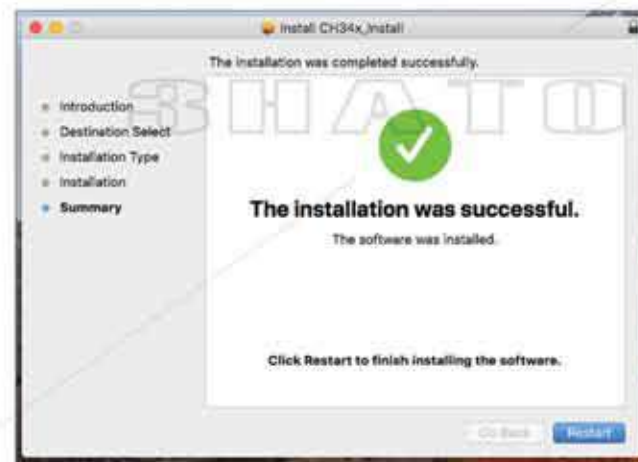




В процессе установки потребуется ввести пароль пользователя системы:



После установки, компьютер требуется перезагрузить.

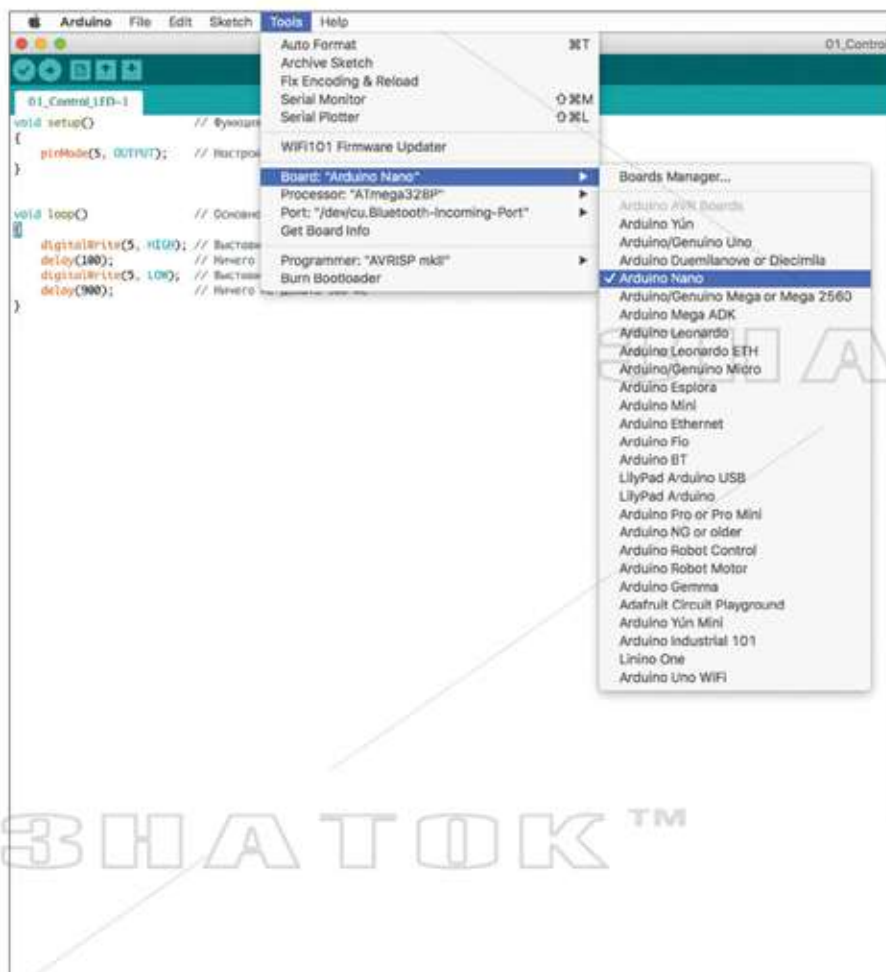


7. В настройках **Security&Privacy** вашего компьютера необходимо нажать кнопку **Allow** для корректной работы драйверов.

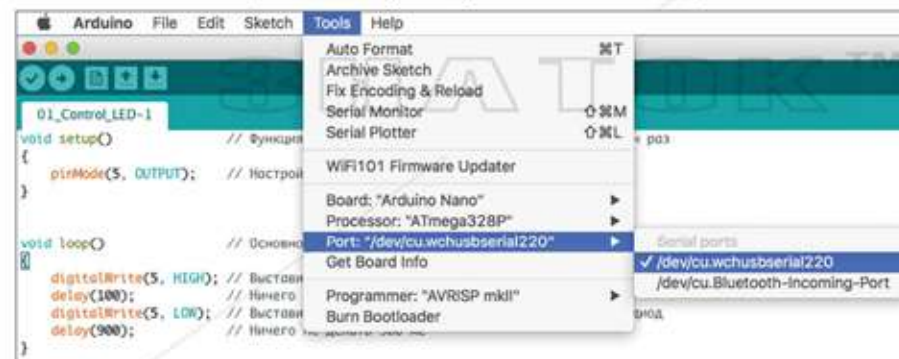


8. Запустите **Arduino IDE** (из папки **Applications**, если вы установили программу туда).

8.1. В меню **Tools** выберите подпункт **Board** и в выпадающем списке выберите **Arduino Nano**.



8.2. В меню **Tools** выберите подпункт **Port** и в выпадающем списке выберите подпункт, указанный на картинке.



8.3. В меню **Sketch** выберите подпункт **Upload**, после чего начнется загрузка программы в модуль **111** (в данный момент программа тестовая), если все шаги были выполнены верно, то в строке состояния появится сообщение «**Загрузка завершена**». Если возникла проблема загрузки в модуль **111**, то попробуйте изменить Порт (п.8.2).

9. Для выполнения лабораторной работы, откройте папку **Лабораторные работы Znatok Arduino** на вашем компьютере. Выберите нужную лабораторную работу и откройте программу — у вас откроется **Arduino IDE** с текстом программы.

Если **Arduino IDE** уже открыта, то вы можете в меню **File** выбрать подпункт **Open** и выбрать нужную программу.

9.1. Для загрузки программы в **Arduino** необходимо выбрать в меню **Sketch** подпункт **Upload** или кликнуть по соответствующей пиктограмме. **НЕ ЗАБУДЬТЕ**: при ошибке загрузки проверьте правильность подключения модуля **111** при помощи **USB-кабеля** к вашему компьютеру, а так же правильность выбора порта (п.8.2).

АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ СВЕТОДИОДОМ

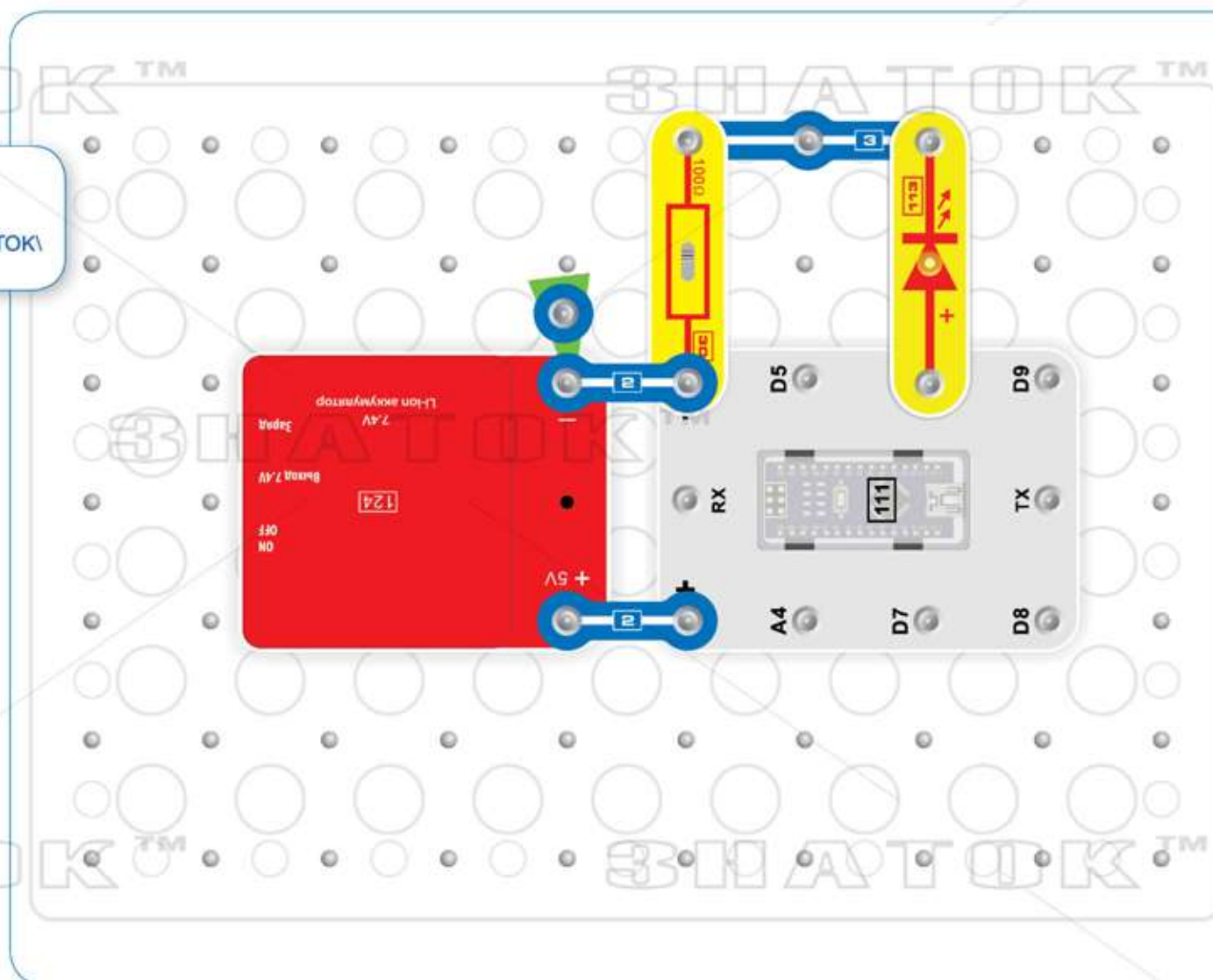


Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

- 01_LightAndSound-01** —
Изменение частоты включения.
- 01_LightAndSound-02** —
Плавное изменение яркости.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите программу
01_LightAndSound-01. После за-
грузки программы (по завершении
загрузки перестают моргать све-
тодиоды на модуле **111**, и внизу
экрана компьютера в строке состо-
яния появится «Загрузка завер-
шена») отсоедините USB-кабель
и включите питание — выключа-
тель на аккумуляторе **124** в по-
ложение ON.



П Р О Г Р А М М Ы

01_LightAndSound-01

Данная программа позволяет светодиоду работать в режиме «Вспышка» — включает его на короткое время. Изменяя численные значения в программе, можно регулировать частоту

Длительность включения

Увеличьте данное значение до **300** и убедитесь в том, что светодиод стал гореть дольше

Частота включения

Уменьшите данное значение до **500** и убедитесь в том, что светодиод стал включаться чаще

```
void setup()
{
  pinMode(6, OUTPUT);
}
void loop()
{
  digitalWrite(6, HIGH); // Выставить «высокий» уровень на D6 – включить светодиод
  delay(100);           // Ничего не делать 100 мс
  digitalWrite(6, LOW); // Выставить «низкий» уровень на D6 – выключить светодиод
  delay(1500);          // Ничего не делать 1500 мс
}
```

включения светодиода. Ниже приведен полный текст программы с комментариями и выделены параметры, изменяющие режим работы.

01_LightAndSound-02

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **01_LightAndSound-02**. После загрузки программы отсоедините USB-кабель.

Данная программа позволяет светодиоду, изменяя яркость, плавно включаться и выключаться. Изменяя численные значения в программе, можно регулировать максимальную яркость и скорость включения. Ниже приведен фрагмент программы с комментариями и выделены параметры, изменяющие режим работы. Полный текст программы можно посмотреть на прилагаемой флэшке.

```
...
int shDelay = 10;
int maxBrightness = 255;
...
```

Скорость включения

Уменьшите данное значение до **5** и убедитесь в том, что светодиод стал набирать максимальную яркость в 2 раза быстрее

Максимальная яркость

Уменьшите данное значение до **100** и убедитесь в том, что светодиод стал включаться с меньшей яркостью

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

УПРАВЛЕНИЕ
ДВУМЯ СВЕТОДИОДАМИ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается 3 программы:

01_LightAndSound-03 —

Синхронное включение.

01_LightAndSound-04 —

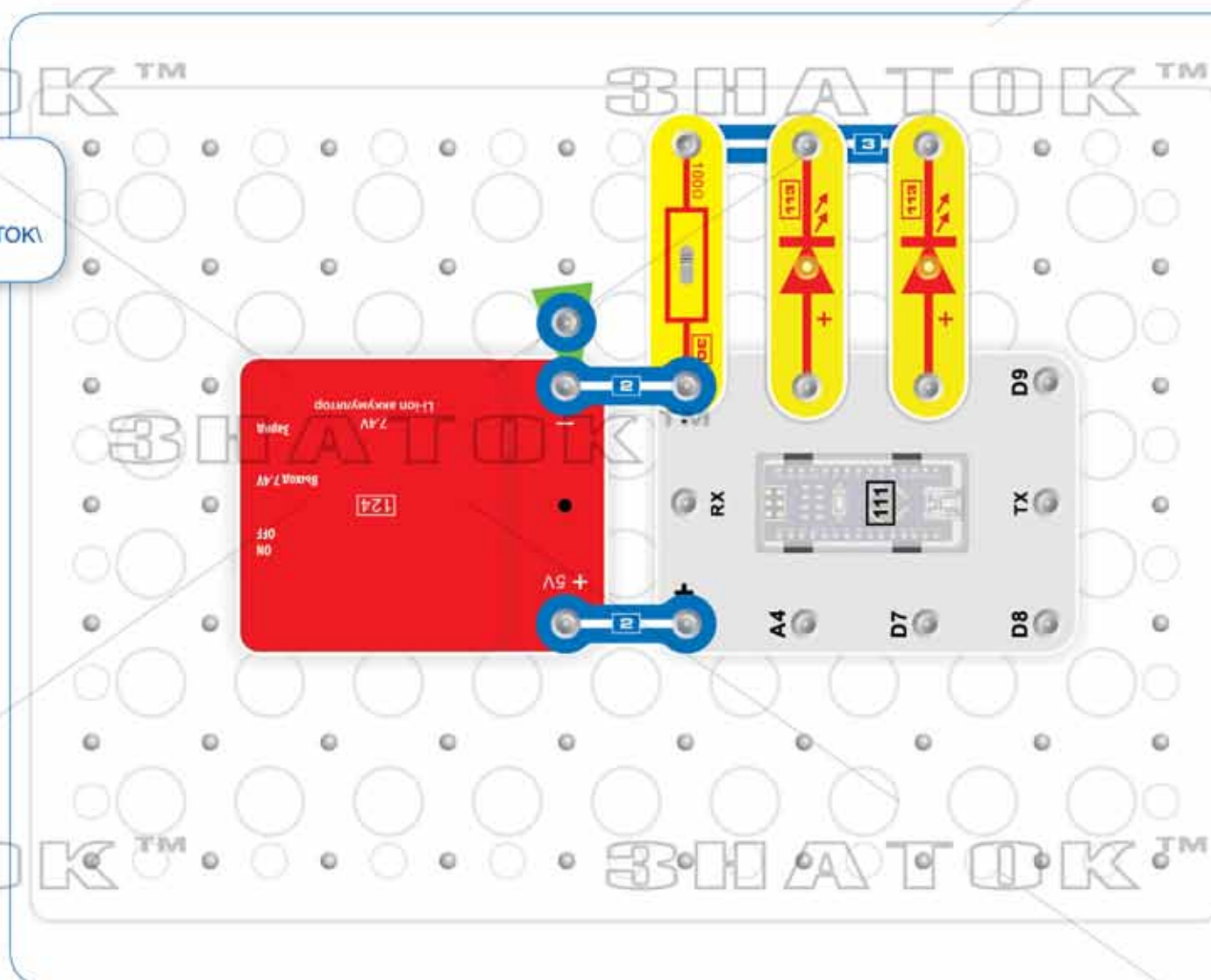
Попеременное включение.

01_LightAndSound-05 —

Плавное попеременное включение.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **01_LightAndSound-03**. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-03

Данная программа позволяет двум светодиодам включаться и выключаться синхронно. Изменяя численные значения в программе, можно регулировать частоту включения светодиодов. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Длительность включения

Увеличьте данное значение до **300** и убедитесь в том, что светодиоды стали гореть дольше

```
...
void loop()
{
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  delay(100);

  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  delay(1000);
}
...
```

Частота включения

Увеличьте данное значение до **2000** и убедитесь в том, что светодиоды стали включаться реже

01_LightAndSound-04

Данная программа позволяет двум светодиодам включаться попеременно. Изменяя численные значения в программе, можно регулировать время работы каждого светодиода. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Время работы первого светодиода

Увеличьте данное значение до **1000** и убедитесь в том, что один светодиод стал гореть дольше

```
...
void loop()
{
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  delay(500);

  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(500);
}
...
```

01_LightAndSound-05

Данная программа позволяет двум светодиодам работать в режиме, когда при увеличении яркости одного светодиода, яркость другого уменьшается и наоборот. Изменяя численные значения в программе, можно регулировать скорость изменения яркости. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Скорость включения

Уменьшите данное значение до **5** и убедитесь в том, что светодиоды стали изменять яркость в 2 раза быстрее

```
...
int shDelay = 10;
int maxBrightness = 255;
...
```

Максимальная яркость

Уменьшите данное значение до **100** и убедитесь в том, что светодиоды стали включаться с не максимальной яркостью

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

ПРОСТАЯ МУЗЫКА



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

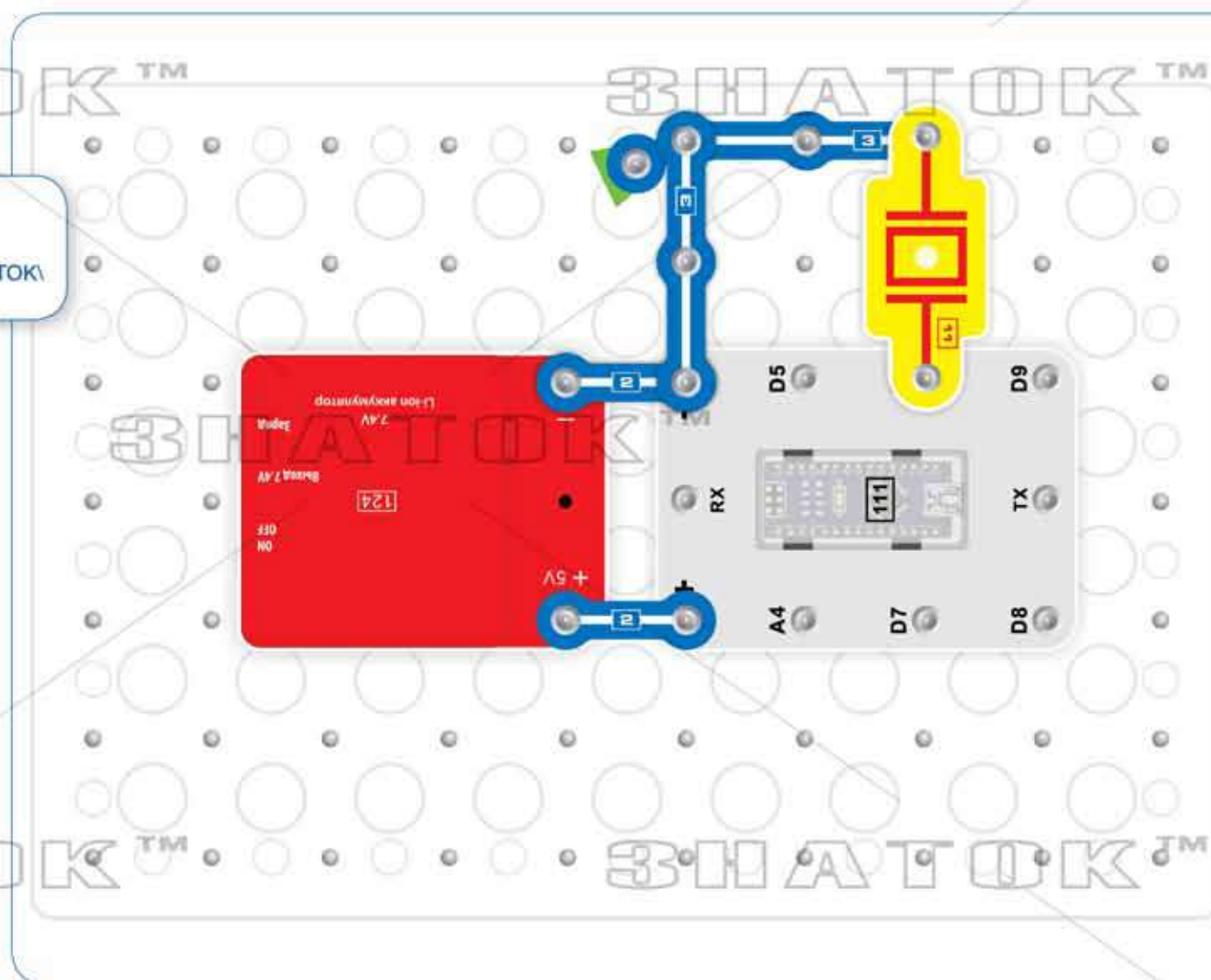
Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:

01_LightAndSound-06 —

Воспроизведение мелодии.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите программу. По-
сле загрузки программы отсо-
едините USB-кабель и включите
питание — выключатель на ак-
кумуляторе **124** в положение
ON.



01_LightAndSound-06

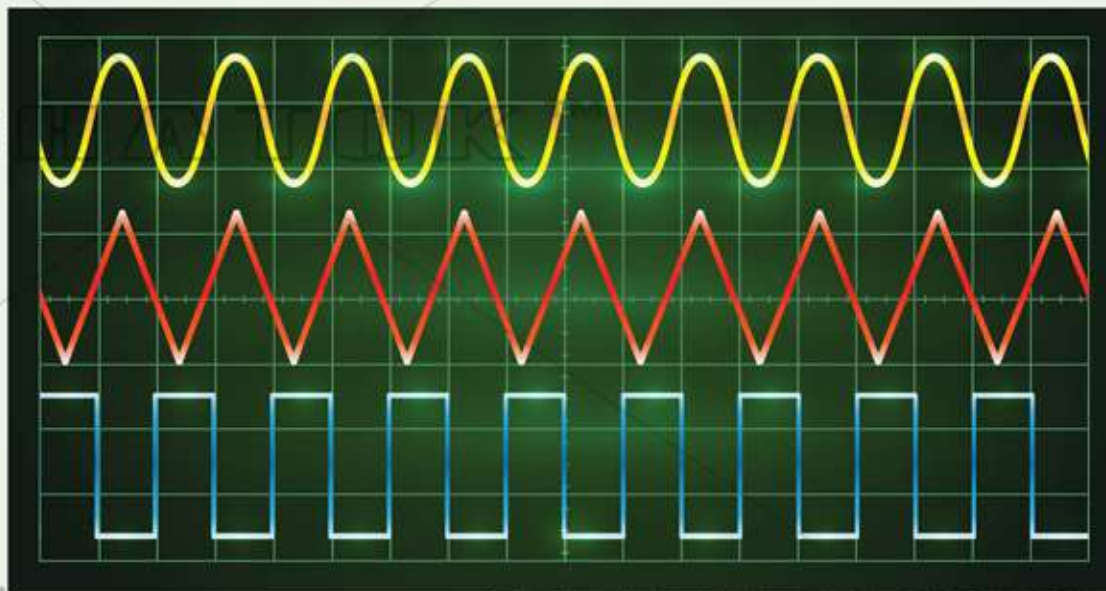
ЗНАТОК™

ЗНАТОК™

Данная программа позволяет пьезоизлучателю **11** воспроизвести простую мелодию.

КАК ЭТО РАБОТАЕТ

Звук — это колебания воздуха, которые доносятся до наших ушей или микрофона. Но колебания могут быть различными по форме. Если мы ударим по ножке камертона, то она будет плавно колебаться — синусоидальная форма (**желтый**). Каждая труба органа (музыкального инструмента) создает колебания треугольной формы (**красный**). В цифровой технике чаще всего используются прямоугольные колебания (**синий**) — их легче всего генерировать. В Arduino для генерации подобного сигнала используется функция `tone` (9,440). Первое число в скобках указывает номер выхода, на котором будет генерироваться сигнал, а второе — частоту этого сигнала. Такой сигнал называется меандр.



ЗНАТОК™

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

СИГНАЛЫ СПЕЦТЕХНИКИ



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается 2 программы:

01_LightAndSound-07 —

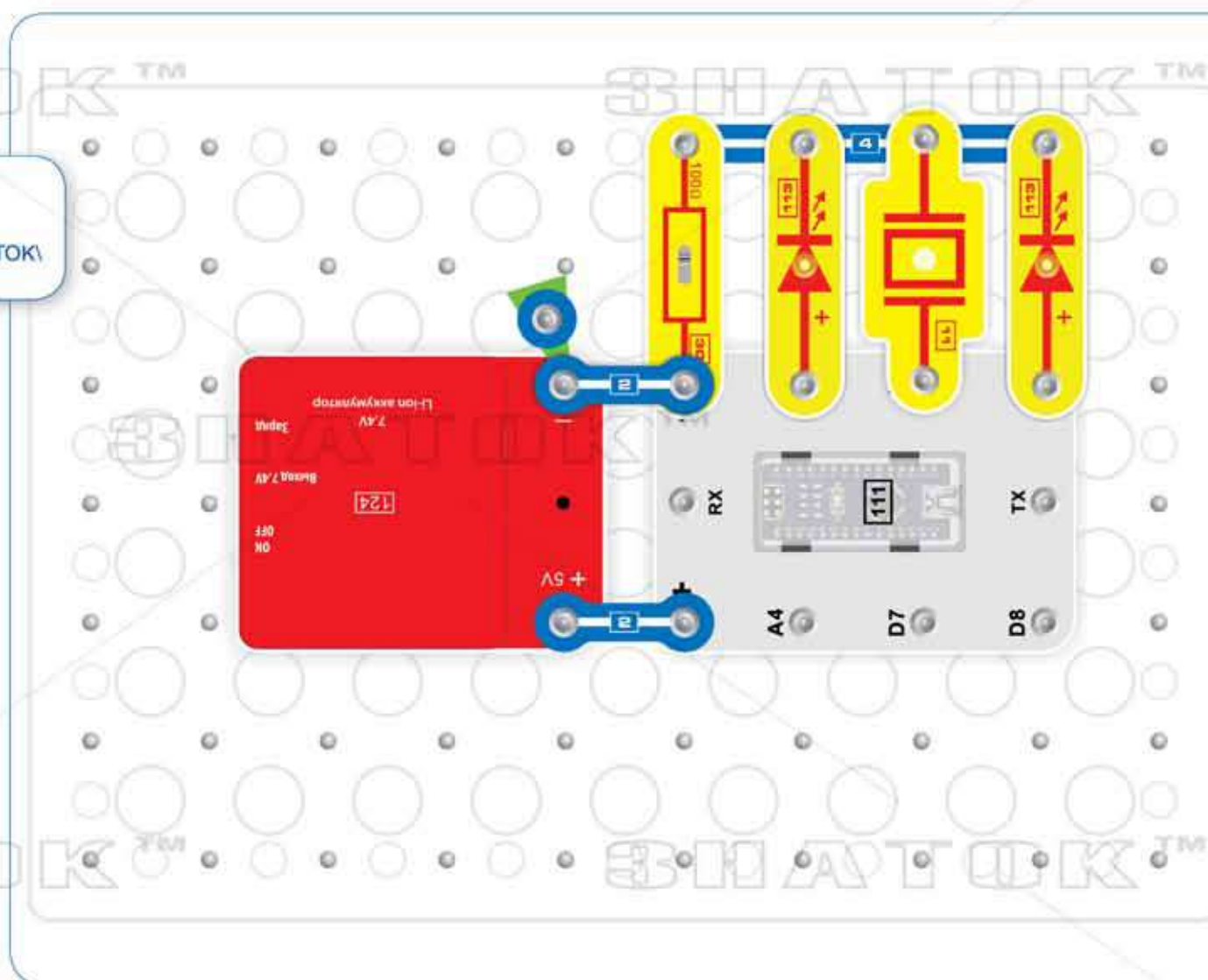
Задний ход.

01_LightAndSound-08 —

Дорожные работы.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **01_LightAndSound-07**. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



01_LightAndSound-07

Данная программа позволяет имитировать световые и звуковые сигналы, издаваемые автомобилем при движении задним ходом. Изменяя численные значения в программе, можно регулировать частоту включения светодиодов. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Частота звукового сигнала

Уменьшите данное значение до **440** и убедитесь в том, что тон звукового сигнала изменился соответственно

```
...
int toneFrequency = 1200;
int shDelay = 1500;
...
```

Частота включения

Уменьшите данное значение до **700** и убедитесь в том, что световой и звуковой сигнал стал срабатывать чаще

01_LightAndSound-08

Данная программа позволяет имитировать световые и звуковые сигналы спецтранспорта при дорожных работах. Изменяя численные значения в программе, можно регулировать частоту работы спецсигнала. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

```
int shDelay = 500;
```

```
...
```

Частота включения

Увеличьте данное значение до **1000** и убедитесь в том, что световой и звуковой сигнал стал работать медленнее

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

ПОВОРОТНИКИ



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE.

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

01_LightAndSound-09 —

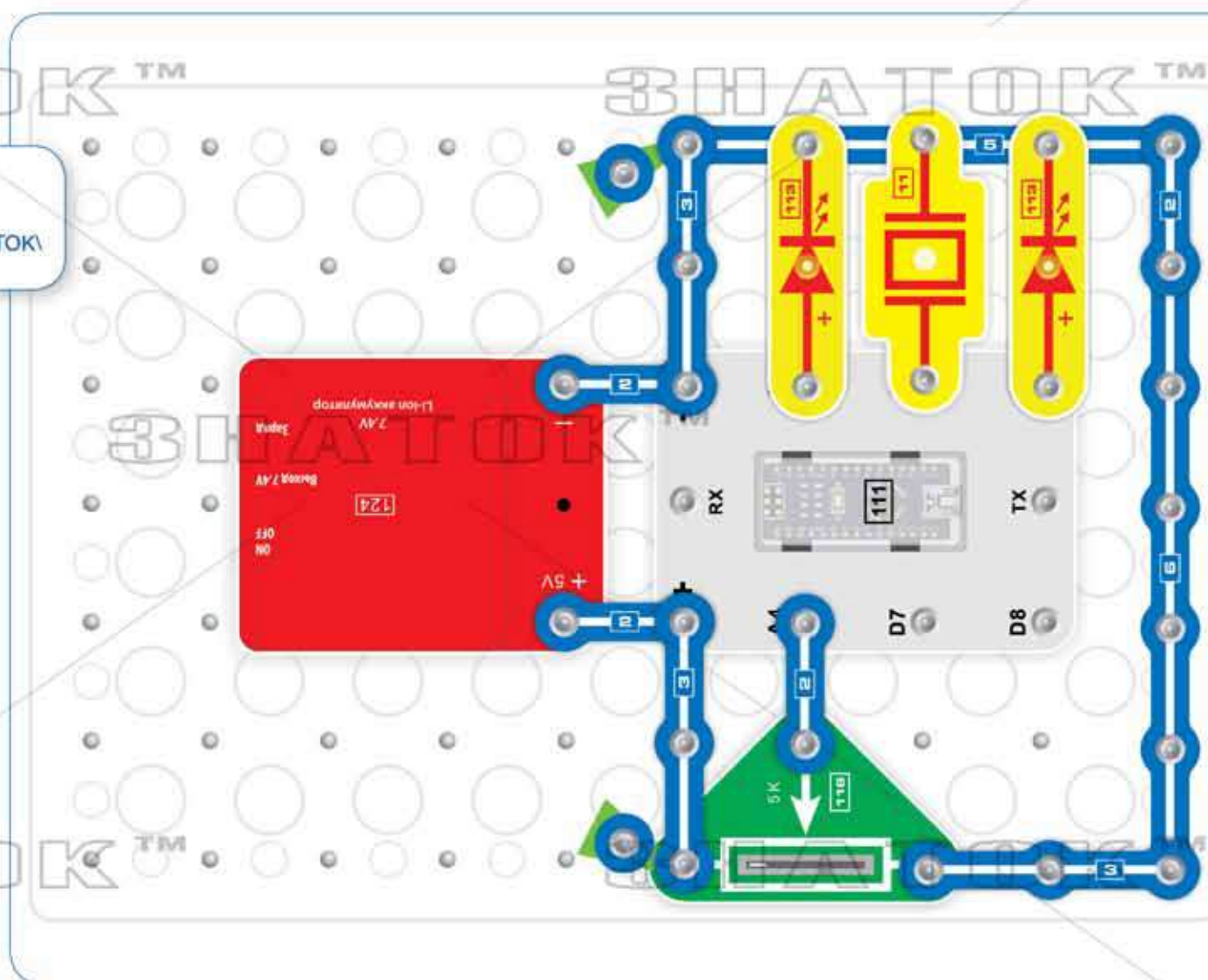
Поворотные огни.

01_LightAndSound-10 —

Забота о других водителях.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите программу
01_LightAndSound-09. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-09

ЗНАТОК™

Данная программа позволяет имитировать работу поворотников в автомобиле. В центральном положении ползунок переменного резистора — ничего не происходит, в крайнем левом — начинает моргать левый поворотник и издаваться сигнал, в крайнем правом — начинает моргать правый. Изменяя численные значения в программе, можно регулировать частоту моргания поворотников. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

```
...
int shDelay = 500;
...
```

Частота включения

Уменьшите данное значение до **250** и убедитесь в том, что поворотники при срабатывании стали моргать чаще

ЗНАТОК™

01_LightAndSound-10

ЗНАТОК™

ЗАМЕНИТЕ ПЬЕЗОИЗЛУЧАТЕЛЬ 11 фарой 47
и загрузите программу **01_LightAndSound-10**

Данная программа позволяет имитировать «Умные поворотники»: при включении любого из поворотников яркость фары уменьшается для того, чтобы их было лучше видно. Изменяя численные значения в программе, можно регулировать насколько изменится яркость фары. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Измененная яркость

Уменьшите данное значение до **0** и убедитесь в том, что фара, при срабатывании поворотников, стала выключаться совсем

```
...
int fadeBrightness = 100;
...
```

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.
Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

СИГНАЛИЗАЦИЯ
НА РАЗРЫВ / ЗАМЫКАНИЕ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

01_LightAndSound-11 —

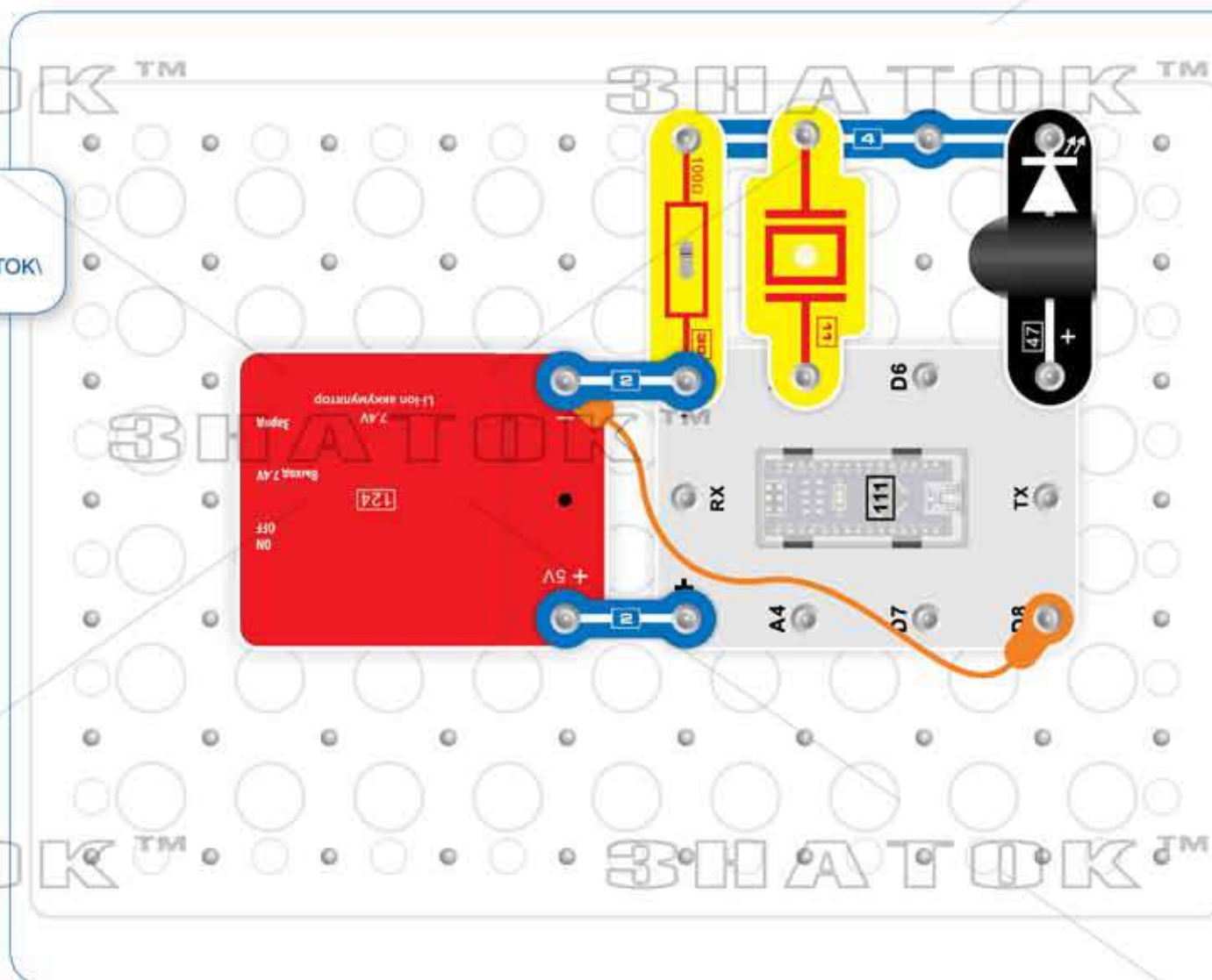
На разрыв.

01_LightAndSound-12 —

На замыкание.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите программу
01_LightAndSound-11. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-11

Данная программа позволяет схеме работать в режиме «Сигнализация на разрыв провода» — Если после запуска схемы отсоединить оранжевый провод от вывода D8, то пьезоизлучатель начнет воспроизводить предупреждающие звуки.

01_LightAndSound-12

Данная программа позволяет схеме работать в режиме «Сигнализация на замыкание провода» — перед запуском схемы, отсоедините провод от вывода D8. Если после запуска схемы им коснуться вывода D8, то пьезоизлучатель начнет воспроизводить предупреждающие звуки.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.
Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

СИГНАЛИЗАЦИЯ НА СВЕТ



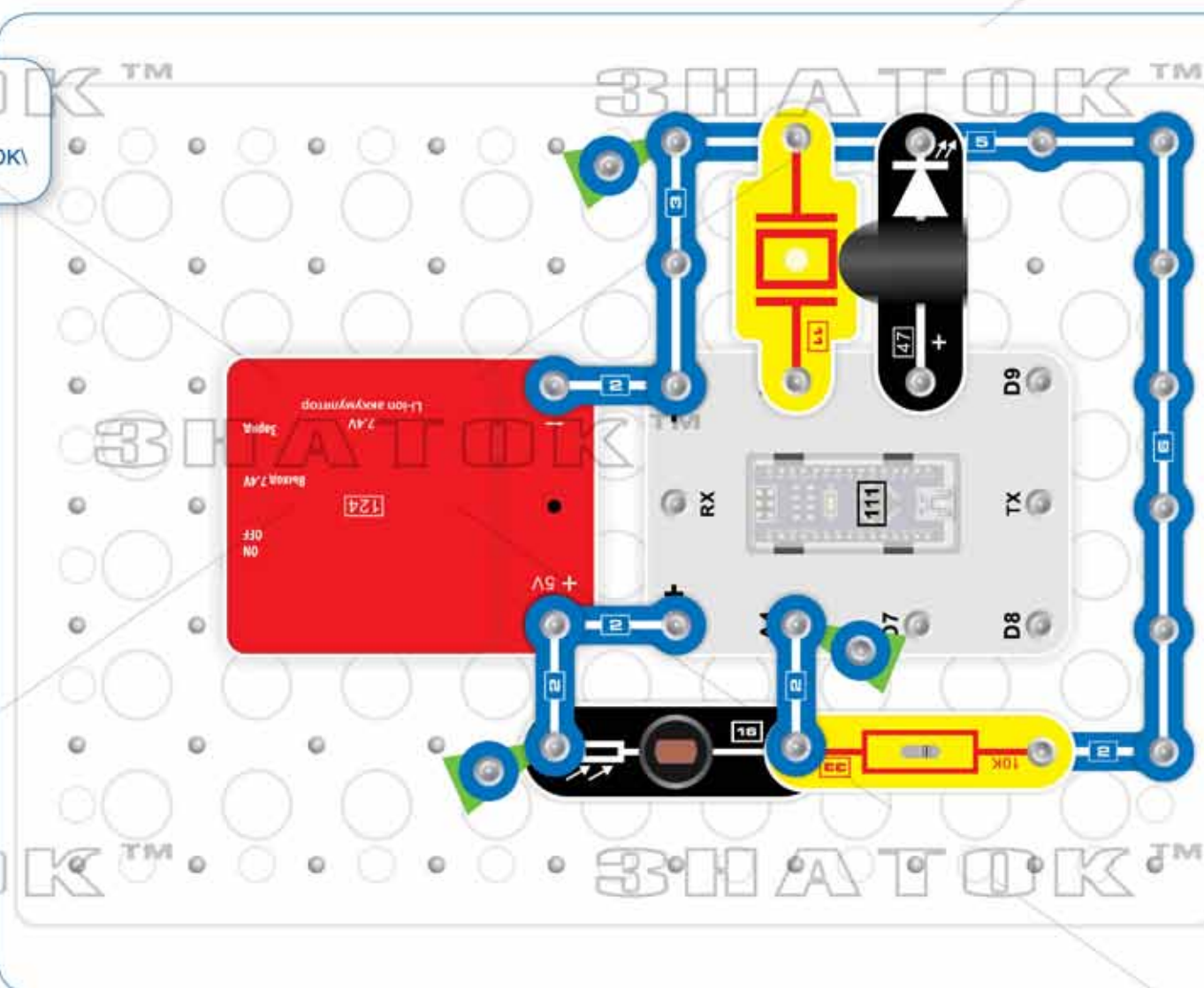
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается 4 программы:

- 01_LightAndSound-13** —
Сигнализация, реагирующая на свет.
- 01_LightAndSound-14** —
Сигнализация, реагирующая на темноту.
- 01_LightAndSound-15** —
Звуки света.
- 01_LightAndSound-16** —
Умный фонарь.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите одну из вышеперечисленных программ. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-13

ЗНАТОК™

«Сигнализация, реагирующая на свет». В данной программе сигнализация сработает, если на фоторезистор попадет свет.

01_LightAndSound-14

ЗНАТОК™

«Сигнализация, реагирующая на темноту». В данной программе сигнализация сработает, если на фоторезистор перестанет попадать свет

ЗНАТОК™

01_LightAndSound-15

«Звуки света». Закрывая рукой от света фоторезистор можно изменять звук в диапазоне от 200Гц до 3кГц.

01_LightAndSound-16

«Умный фонарь». Закрывая рукой от света фоторезистор изменяется яркость фары.

ЗНАТОК™

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.
Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

СИГНАЛИЗАЦИЯ
НА ДВИЖЕНИЕ

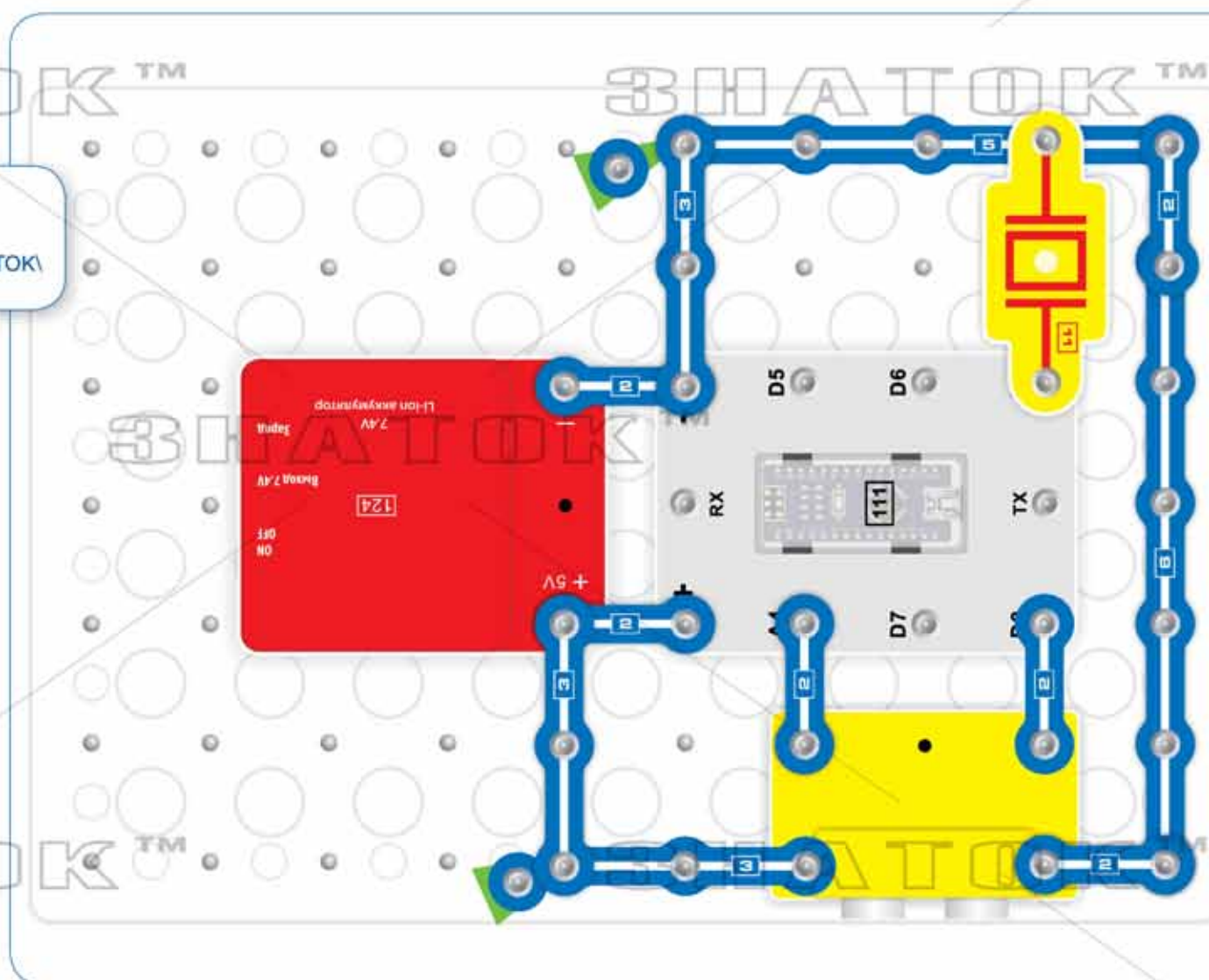
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

- 01_LightAndSound-17 —
Сигнализация на движение.
- 01_LightAndSound-18 —
Датчик движения. Звук.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль 111 при
помощи USB-кабеля к компью-
теру и загрузите программу
01_LightAndSound-17. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе 124 в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-17

ЗНАТОК™

«Сигнализация на движение». Через 1 секунду после включения раздастся звуковой сигнал, говорящий о том, что сигнализация включилась. Если после этого перед датчиком расстояния произойдет какое-либо движение, то сработает сигнализация. Изменяя численные значения в программе, можно регулировать время установки сигнализации, во время которого она не будет срабатывать. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Время установки

Увеличьте данное значение до 3000 и у вас появится дополнительное время для установки сигнализации

...

```
int setupDelay = 1000;
```

...

ЗНАТОК™

01_LightAndSound-18

ЗНАТОК™

«Датчик движения. Звук.». В данной программе если перед датчиком расстояния происходит какое-либо движение, раздастся звуковой сигнал. Когда движение прекращается — звук затихает.



ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

ТРЕВОЖНАЯ КНОПКА



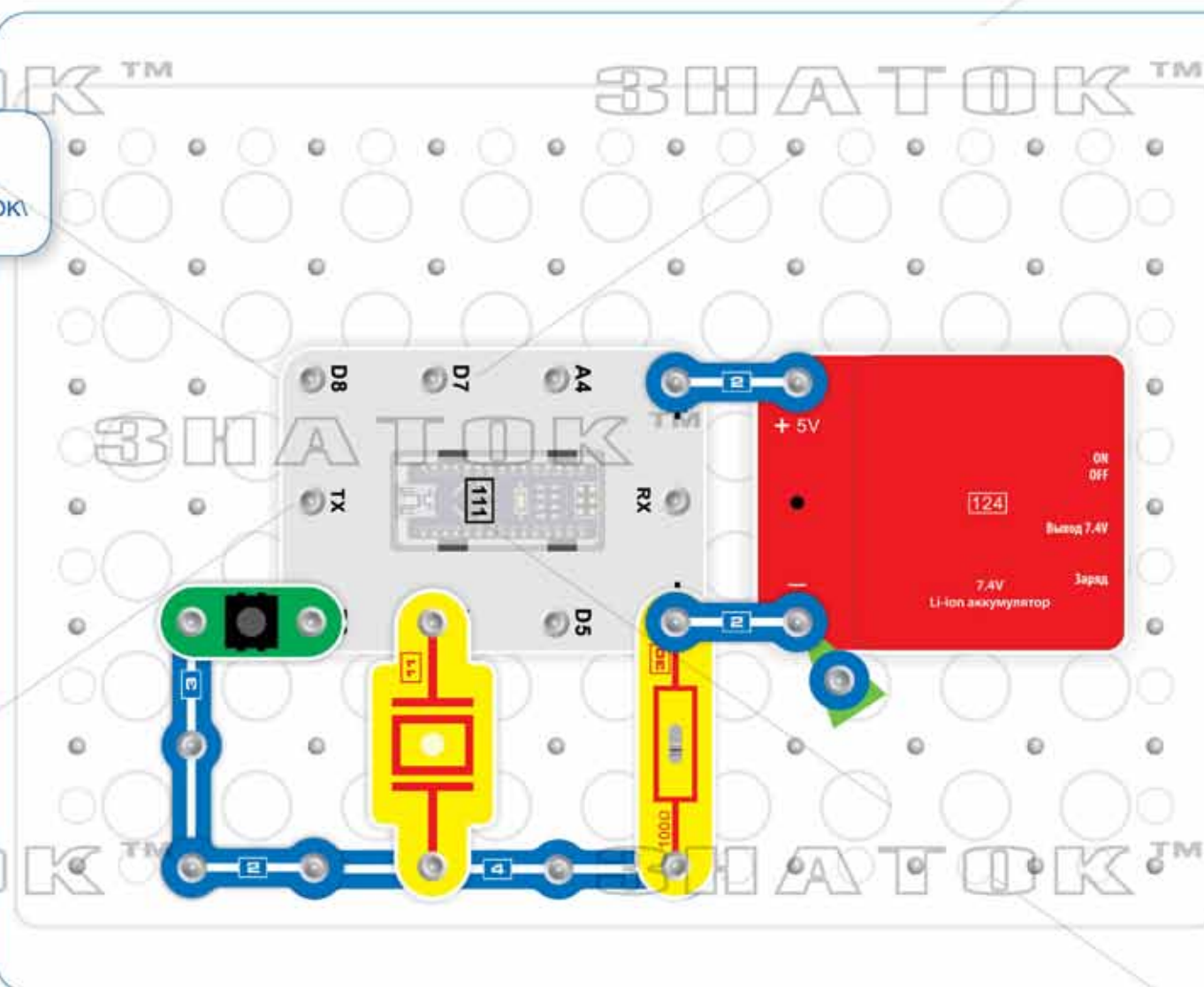
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 3 программы:

- 01_LightAndSound-19 —
Тревожная кнопка.
- 01_LightAndSound-20 —
Ручная сирена.
- 01_LightAndSound-21 —
Клаксон.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
01_LightAndSound-19. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-19

«Тревожная кнопка». В данной программе сигнализация сработает, если нажать на кнопку, подключенную к выводу D9. Сигнализация будет работать до момента выключения питания схемы.



01_LightAndSound-20

«Ручная сирена». В данной программе при нажатии на кнопку — плавно увеличивается частота звука, воспроизводимого пьезоизлучателем, если кнопку отпустить — частота уменьшается. Изменяя численные значения в программе, можно регулировать скорость изменения тона. Ниже приведен фрагмент текста программы с комментариями и выделены параметры, изменяющие режим работы.

Скорость изменения

Уменьшите данное значение до 5 и убедитесь в том, что тон звука стал изменяться быстрее

...

```
int fadeDelay = 10;
```

...

01_LightAndSound-21

«Клаксон». При нажатии на кнопку раздается звуковой сигнал автомобильного клаксона.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

УПРАВЛЕНИЕ ФАРОЙ



Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/ Электронные конструкторы ЗНАТОК/ Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагаются 4 программы, регулирующие яркость фары по-разному:

01_LightAndSound-22 —

Линейная зависимость.

01_LightAndSound-23 —

V-образная зависимость.

01_LightAndSound-24 —

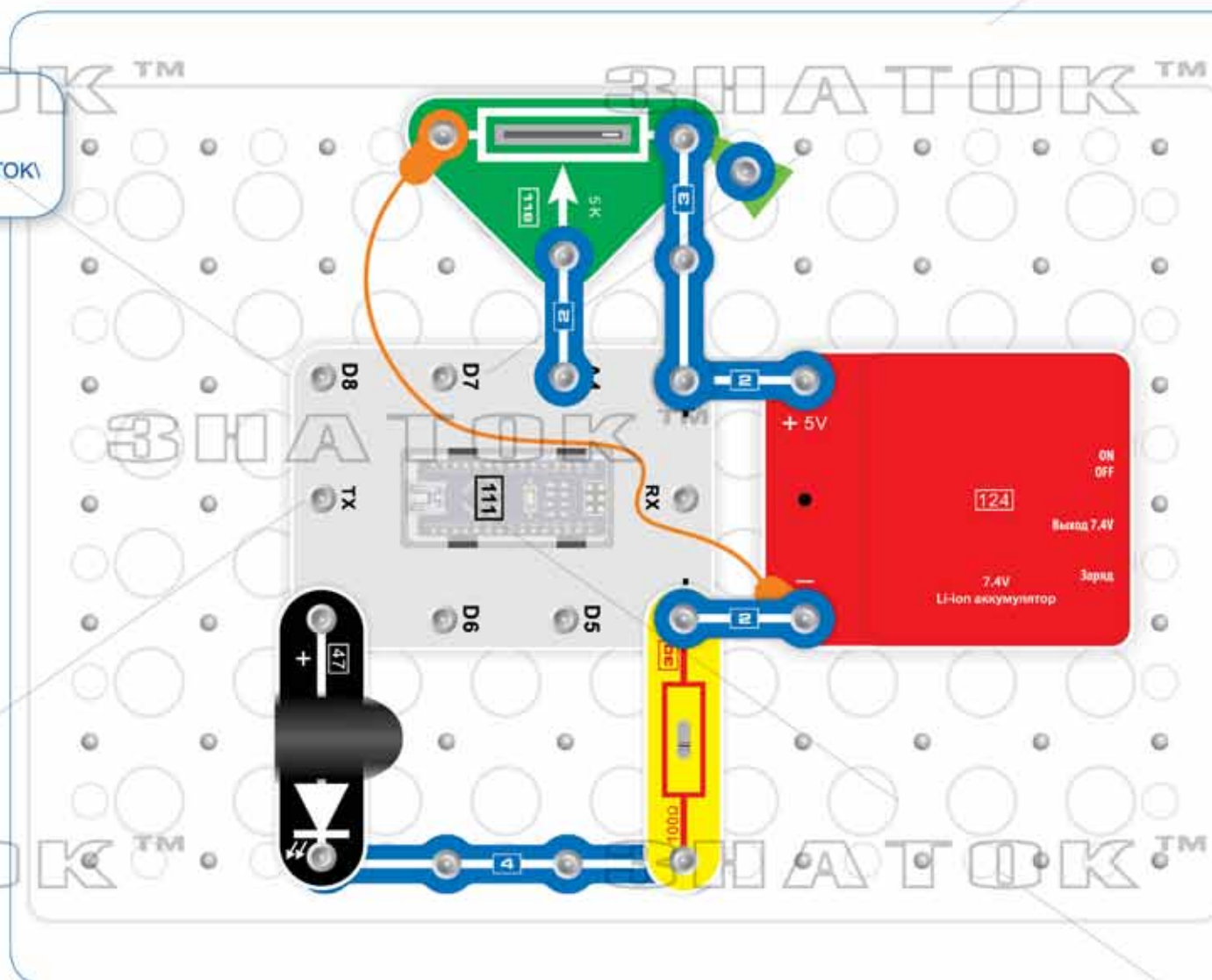
Колоколообразная зависимость.

01_LightAndSound-25 —

Граничное значение.

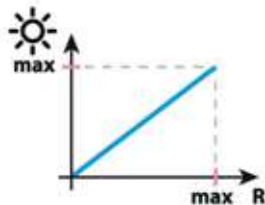
Полный текст программ можно посмотреть на прилагаемой флешке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите одну из вышеперечисленных программ. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



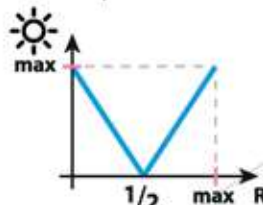
01_LightAndSound-22

«Линейная зависимость». При работе данной программы, изменяя положение потенциометра яркость фары изменяется линейно: в крайнем левом положении фара выключена, в крайнем правом — работает на максимальной мощности.



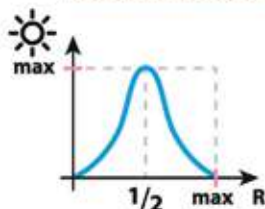
01_LightAndSound-23

«V-образная зависимость». При работе данной программы, изменяя положение потенциометра, яркость фары изменяется следующим образом: в крайних положениях фара включена с максимальной яркостью, а со смещением ползунка переменного резистора к центру — её яркость сначала уменьшается, а после прохождения центра — увеличивается.



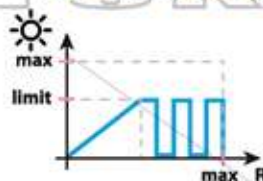
01_LightAndSound-24

«Колоколообразная зависимость». При работе данной программы, изменяя положение потенциометра яркость фары изменяется следующим образом: в крайних положениях фара выключена, а со смещением ползунка переменного резистора к центру — её яркость сначала увеличивается, а после прохождения центра — уменьшается.



01_LightAndSound-25

«Граничное значение». Данный режим позволяет определить наступление пика какого-либо значения. При перемещении потенциометра слева направо, в начале яркость фары увеличивается, а по достижении определенного уровня фара начинает моргать.



ДЛЯ ПРОДВИНУТЫХ

В данной схеме при линейном изменении напряжения на входе A4 яркость светодиода будет меняться нелинейно. Попробуйте написать программу, которая будет изменять яркость светодиода линейно.

Подсказка:

L — желаемая яркость от 0.0 (0%) до 1.0 (100%)

Y — масштабируемое значение ШИМ от 0.0 (0%) до 1.0 (100%)

$$Y = \begin{cases} (100 \cdot L) / 902.3, & \text{если } L \leq 0.08 \\ ((100 \cdot L + 16.0) / 116.0)^3, & \text{иначе} \end{cases}$$

Необходимо помнить, что данная формула работает для значений от 0.0 до 1.0, а в Arduino для ШИМ используются значения от 0 до 255

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.

Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

УПРАВЛЕНИЕ ЗУММЕРОМ



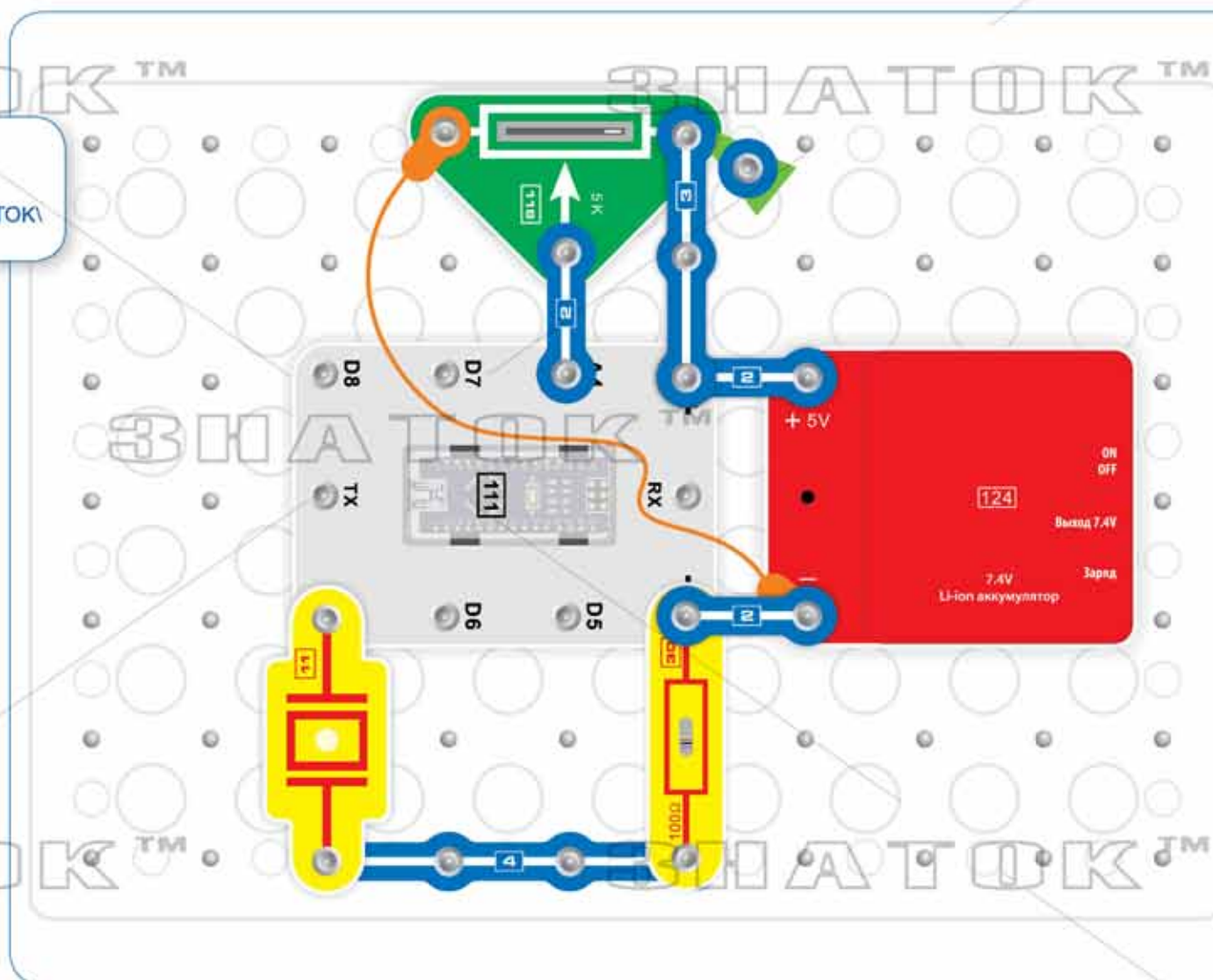
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется **3** программы:

- 01_LightAndSound-26** —
Регулировка тона.
- 01_LightAndSound-27** —
Три мелодии.
- 01_LightAndSound-28** —
Ручной проигрыватель.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
01_LightAndSound-26. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-26

В данной программе при помощи переменного резистора регулируется частота звука, которую испускает пьезоизлучатель. Ниже приведен фрагмент программы, в котором показано как можно изменять минимальную и максимальную частоту воспроизводимого звука.

Максимальная частота

Увеличьте данный параметр до **6000** и убедитесь, что при перемещении потенциометра в крайнее положение звук стал *значительно выше*

Минимальная частота

Уменьшите данный параметр до **100** и убедитесь, что в одном из крайних положений потенциометра звук стал ниже

```
...
const int MIN_FREQUENCY = 400;
const int MAX_FREQUENCY = 3000;
```

01_LightAndSound-27

Данная программа позволяет при помощи потенциометра выбирать одну из трех воспроизводимых мелодий — выбор происходит при перемещении ползунка потенциометра в крайнее левое положение, крайнее правое или центральное положение. Программу целиком можно посмотреть на прилагаемой к набору флэшке.

01_LightAndSound-28

«Ручной проигрыватель». Каждая мелодия — это последовательность нот с паузами между ними. В данной программе каждому положению ползунка переменного резистора соответствует одна из нот заданной мелодии: левое крайнее положение — первая нота, правое крайнее положение — последняя нота. Вы можете перемещать ползунок слева направо (или справа налево, если хотите послушать мелодию «наоборот») с разными скоростями и послушать, как изменится мелодия.

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

ОДНОКЛАВИШНОЕ ПИАНИНО

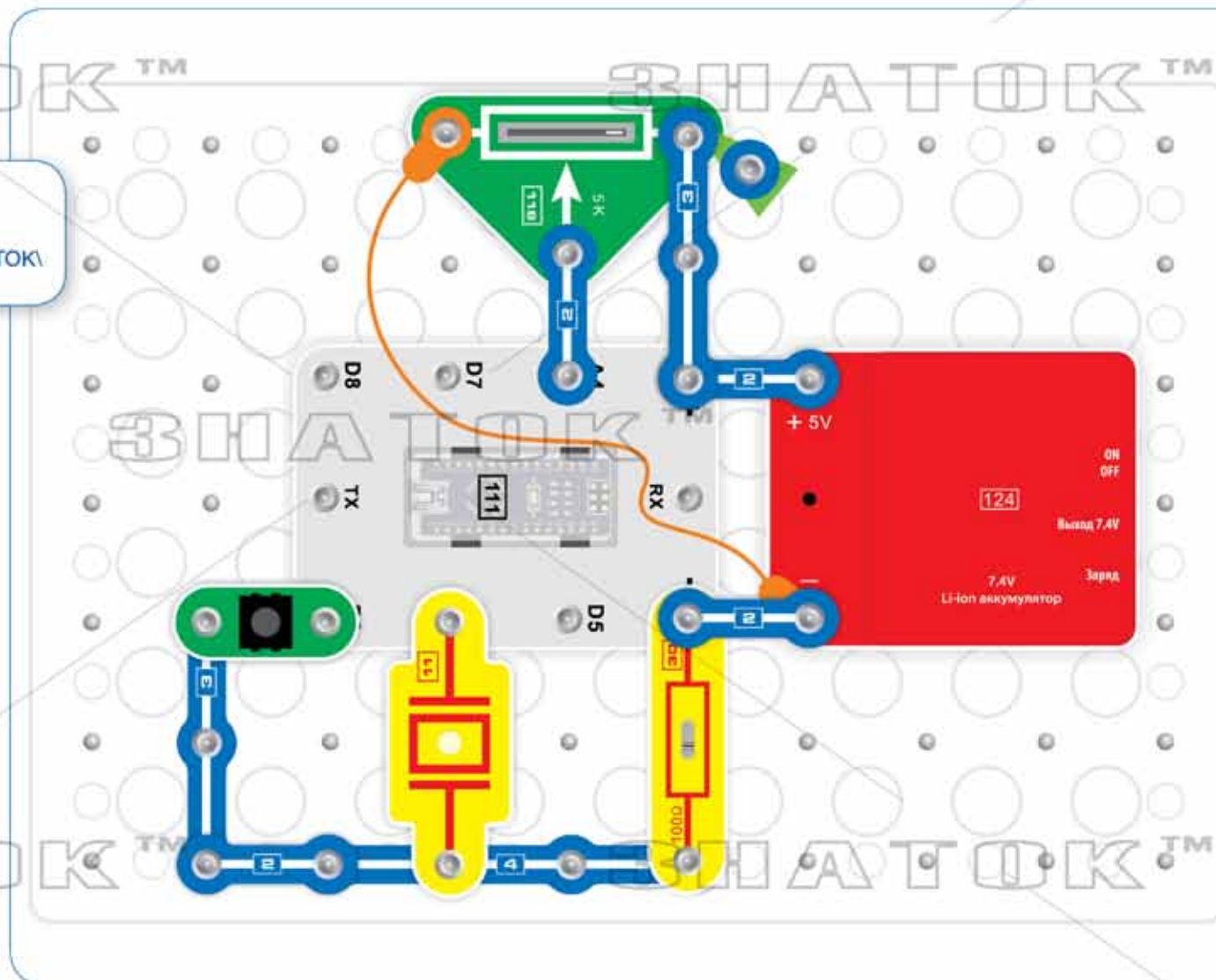
Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/Электронные конструкторы ЗНАТОК/Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается **2** программы:

- 01_LightAndSound-29** — Одноклавишное пианино.
- 01_LightAndSound-30** — Настраиваемый звонок.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **01_LightAndSound-29**. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-29

ЗНАТОК™

«Одноклавишное пианино». В данной программе каждому из положений потенциометра соответствует одна из 10 нот первой октавы, нажатие на кнопку позволяет сыграть эту ноту.

01_LightAndSound-30

ЗНАТОК™

«Настраиваемый звонок». В данной программе, при нажатии на кнопку раздается дверной звонок. Изменяя положение переменного резистора можно изменять частоту сигнала звонка.



ЗНАТОК™

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

АДАПТИВНАЯ ФАРА



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

01_LightAndSound-31 —

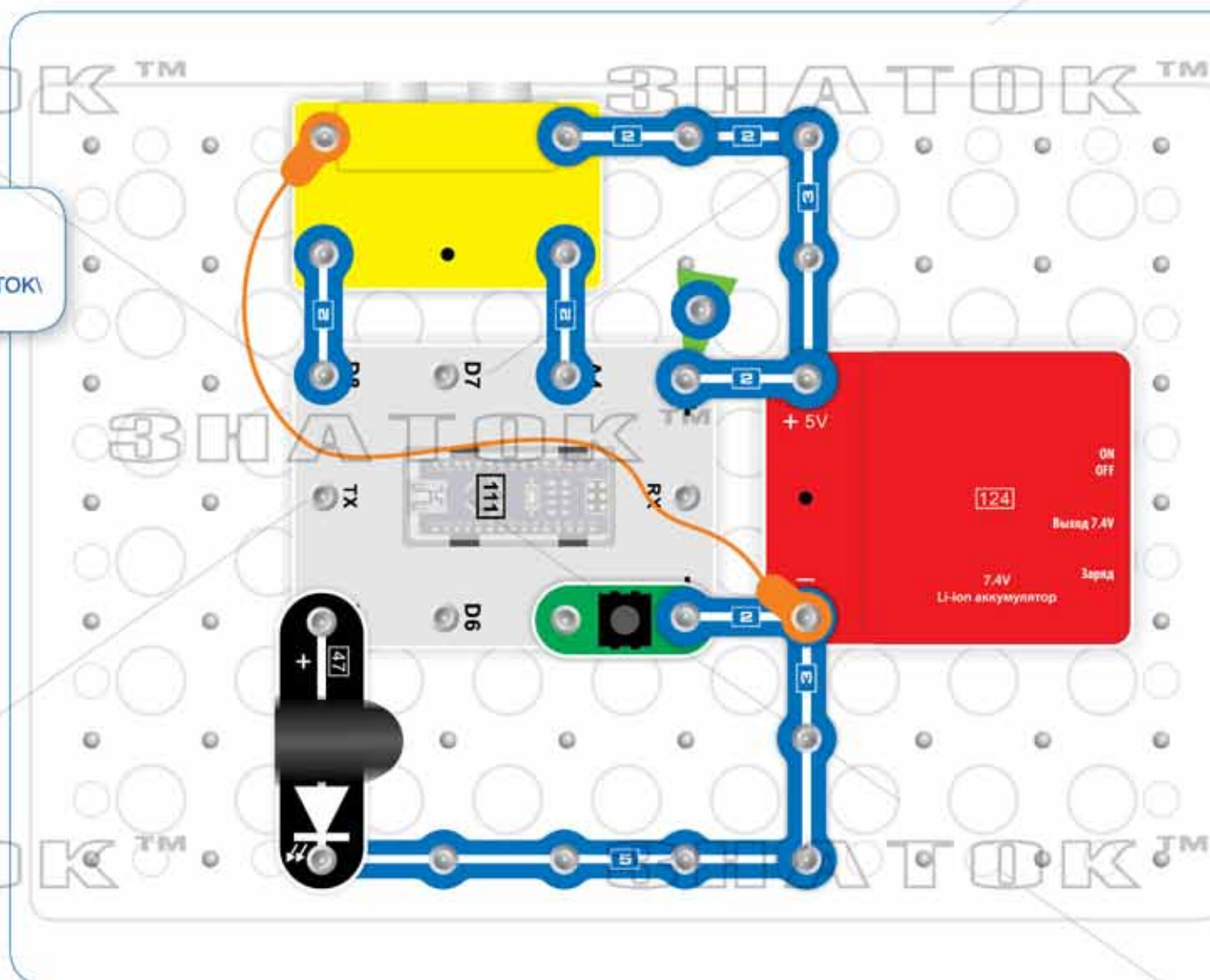
Яркость от расстояния.

01_LightAndSound-32 —

Датчик движения. Свет.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
01_LightAndSound-31. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-31

В данной программе яркость фары устанавливается автоматически в зависимости от удаленности препятствия от ультразвукового дальномера 89: чем ближе к дальномеру находится какой-либо объект, тем слабее светит фара 47.



01_LightAndSound-32

«Датчик движения. Свет». В данной программе, если перед ультразвуковым дальномером 89 происходит какое-либо движение, включается фара 47. Когда движение прекращается — фара гаснет.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

ДИСТАНСВОКС
С КНОПКОЙ

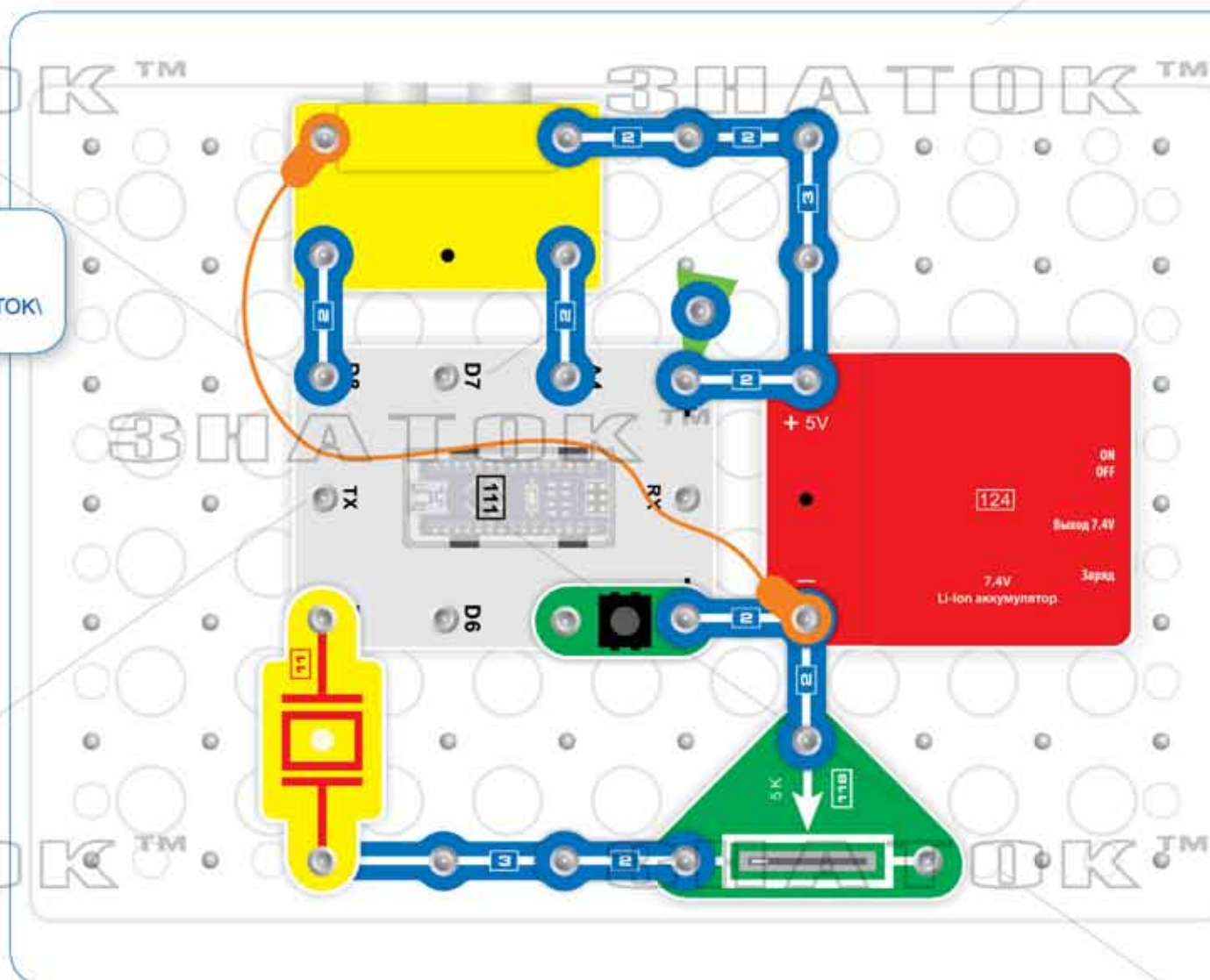
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:

01_LightAndSound-33 —
Дистансвокс с кнопкой.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компьюте-
ру и загрузите программу. После
загрузки программы отсоеди-
ните USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



01_LightAndSound-33

На каких только инструментах не играют люди! Предлагаем вам ещё один — мы его назвали «Дистансвокс с кнопкой» потому, что нота, которую он выдает, соответствует определенному расстоянию до ультразвукового датчика. Можно использовать руку, но лучший результат достигается при использовании плоской поверхности, например, книги с твердой обложкой.

Ультразвуковой датчик настроен таким образом, что все ноты первой октавы и «До» второй октавы расположены в промежутке от 3 до 27 см с шагом 3 см. Это означает, что вы можете, меняя расстояние от вашей руки до датчика, и нажимая на кнопку, воспроизводить различные мелодии.

Это усовершенствованная модель Дистансвокса, по сравнению с описанной в наборе Arduino BASIC.

для продвинутых

Попытайтесь собрать схему и написать программу свето-звуковой сигнализации, основываясь на знаниях, полученных в двух предыдущих проектах.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

ПАРКТРОНИК



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 2 программы:

01_LightAndSound-34 —

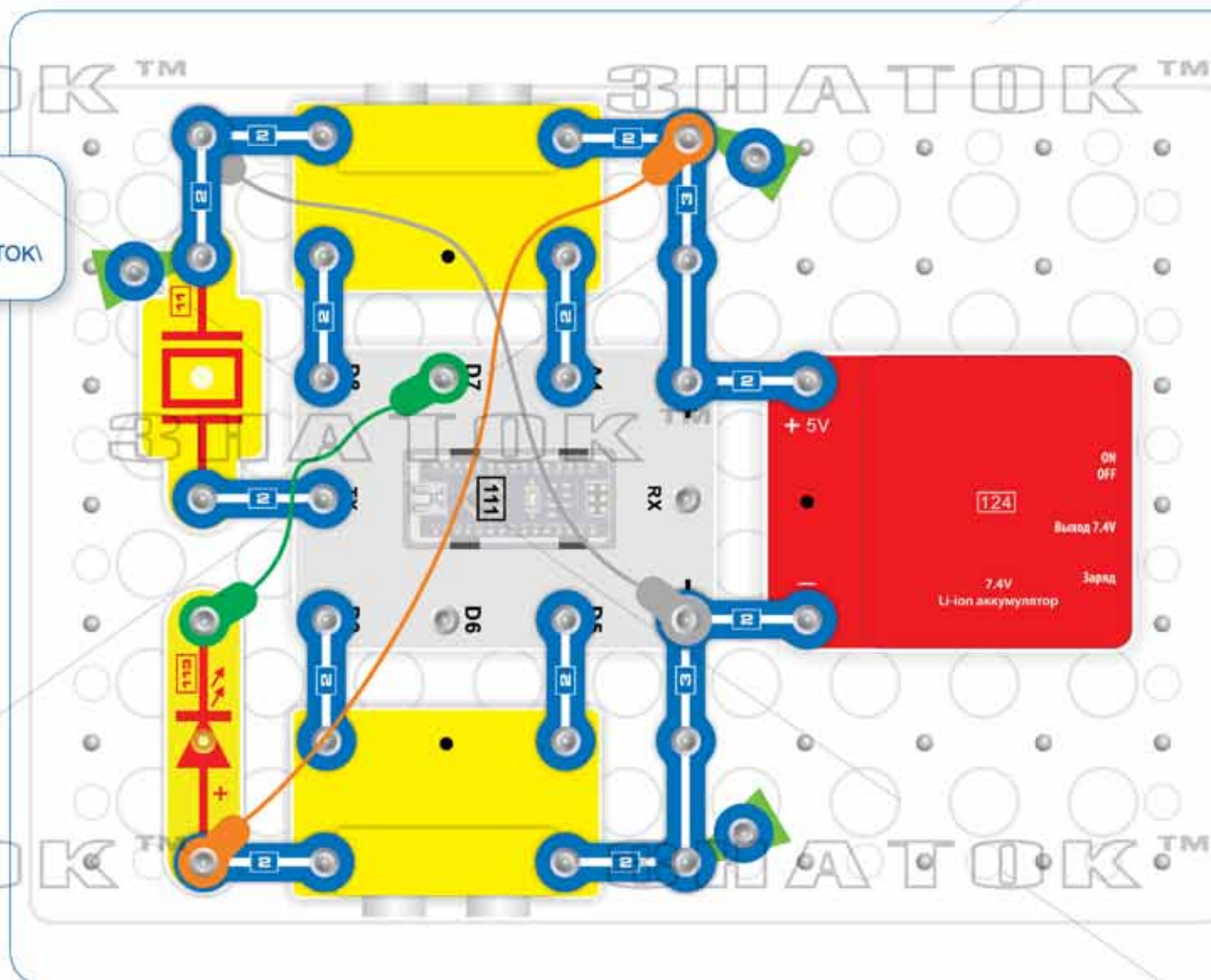
Парктроник.

01_LightAndSound-35 —

Датчик посещения.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
01_LightAndSound-34. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-34

ЗНАТОК™

«Парктроник». Данная программа позволяет имитировать работу переднего и заднего датчика парковки. При приближении объекта к одному из ультразвуковых датчиков раздается специальный световой и звуковой сигнал, разный для переднего и заднего датчика.

01_LightAndSound-35

ЗНАТОК™

«Датчик посещения». Подобная схема может использоваться и там, где два датчика удалены друг от друга на большое расстояние, например, в магазине. Один датчик, расположенный на входе фиксирует и сигнализирует о входящих в помещение посетителей, а другой — о выходящих.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

ДАТЧИК УДАРА



Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/ Электронные конструкторы ЗНАТОК/ Arduino MOBILE

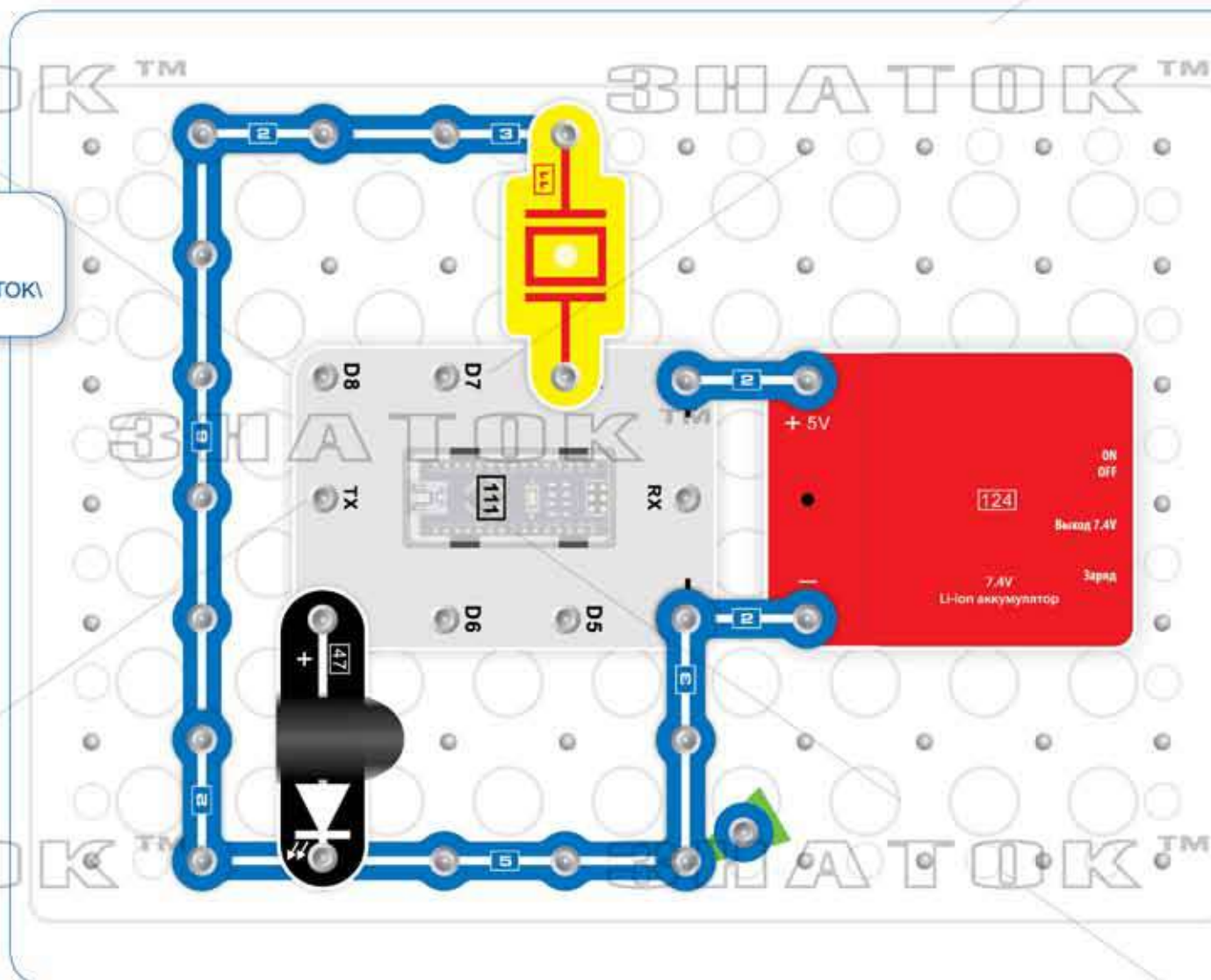
Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:

01_LightAndSound-36 —

Датчик удара.

Полный текст программы можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

01_LightAndSound-36

ЗНАТОК™

ЗНАТОК™

«Датчик удара». Данная программа показывает, что пьезоизлучатель **11** может не только воспроизводить звуки, но и одновременно быть датчиком. Если после запуска про-

граммы ударить с умеренной силой по пьезоизлучателю, то сработает светозвуковая сигнализация.



ЗНАТОК™

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.
Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

РЕГУЛИРОВКА
ЗВУКОВЫХ СИГНАЛОВ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

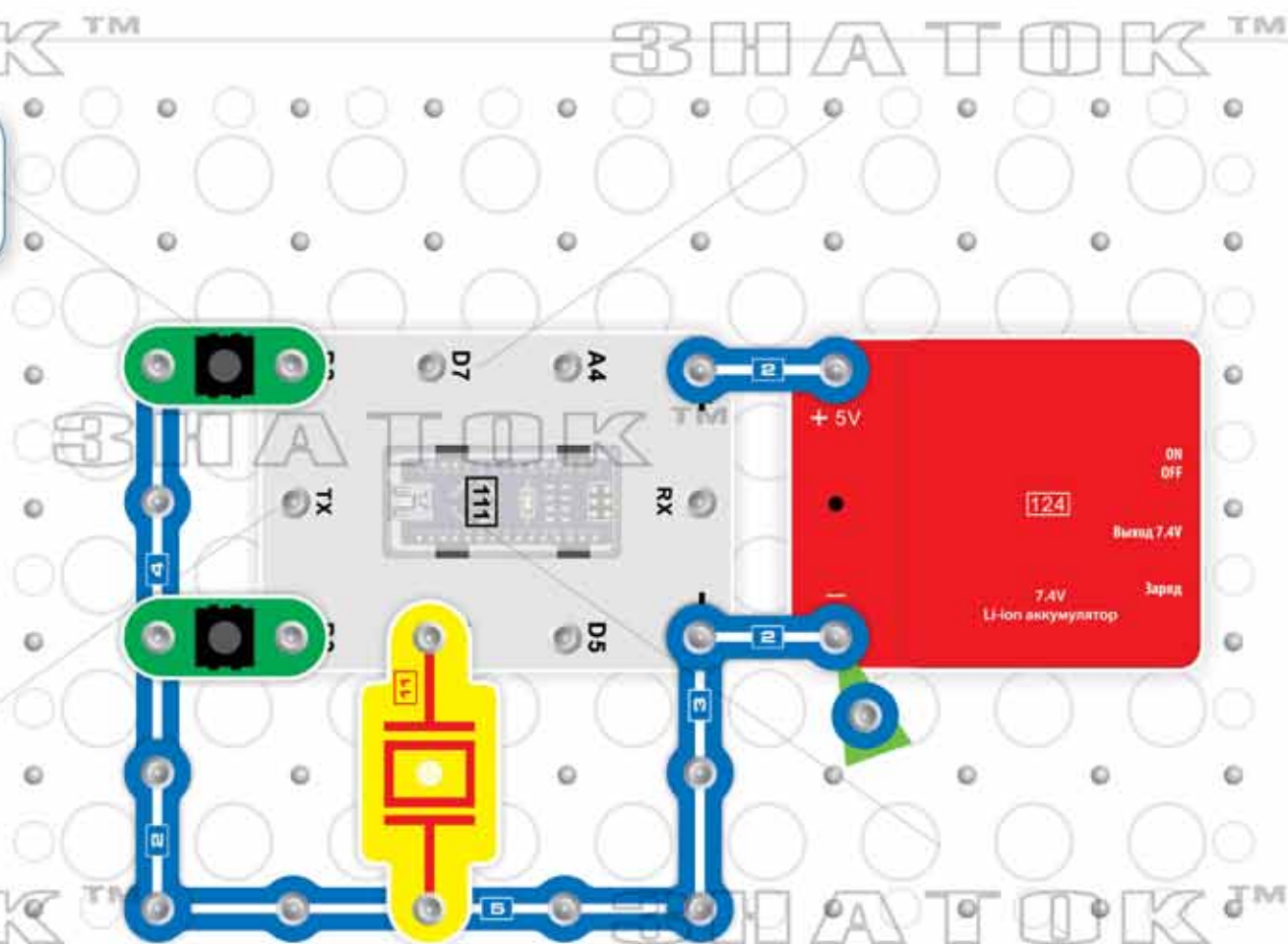
Соберите схему, как показано
на рисунке. К этой схеме прила-
гается 2 программы:

01_LightAndSound-37 —
Управление звуком двумя кноп-
ками. Вариант 1.

01_LightAndSound-38 —
Управление звуком двумя кноп-
ками. Вариант 2.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
01_LightAndSound-37. После за-
грузки программы отсоедините
USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-37

ЗНАТОК™

Данная программа позволяет при помощи двух кнопок изменять высоту тона прерывистых звуковых сигналов: одна кнопка её увеличивает, другая уменьшает.

01_LightAndSound-38

ЗНАТОК™

Данная программа показывает другой способ управления при помощи двух кнопок. При нажатии на одну кнопку выбирается одна из заранее заданных частот следования звуковых сигналов, а нажатие на другую — высоту тона.

← Настройки > Аудио

Баланс/Микшер

Эквалайзер

Громкость/Скорость

Смещ. громкости AUX



Басы 4

Сред. 1

Высок. 7

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.

Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

РЕГУЛИРОВКА
ЗВУКОВЫХ И СВЕТОВЫХ
СИГНАЛОВ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

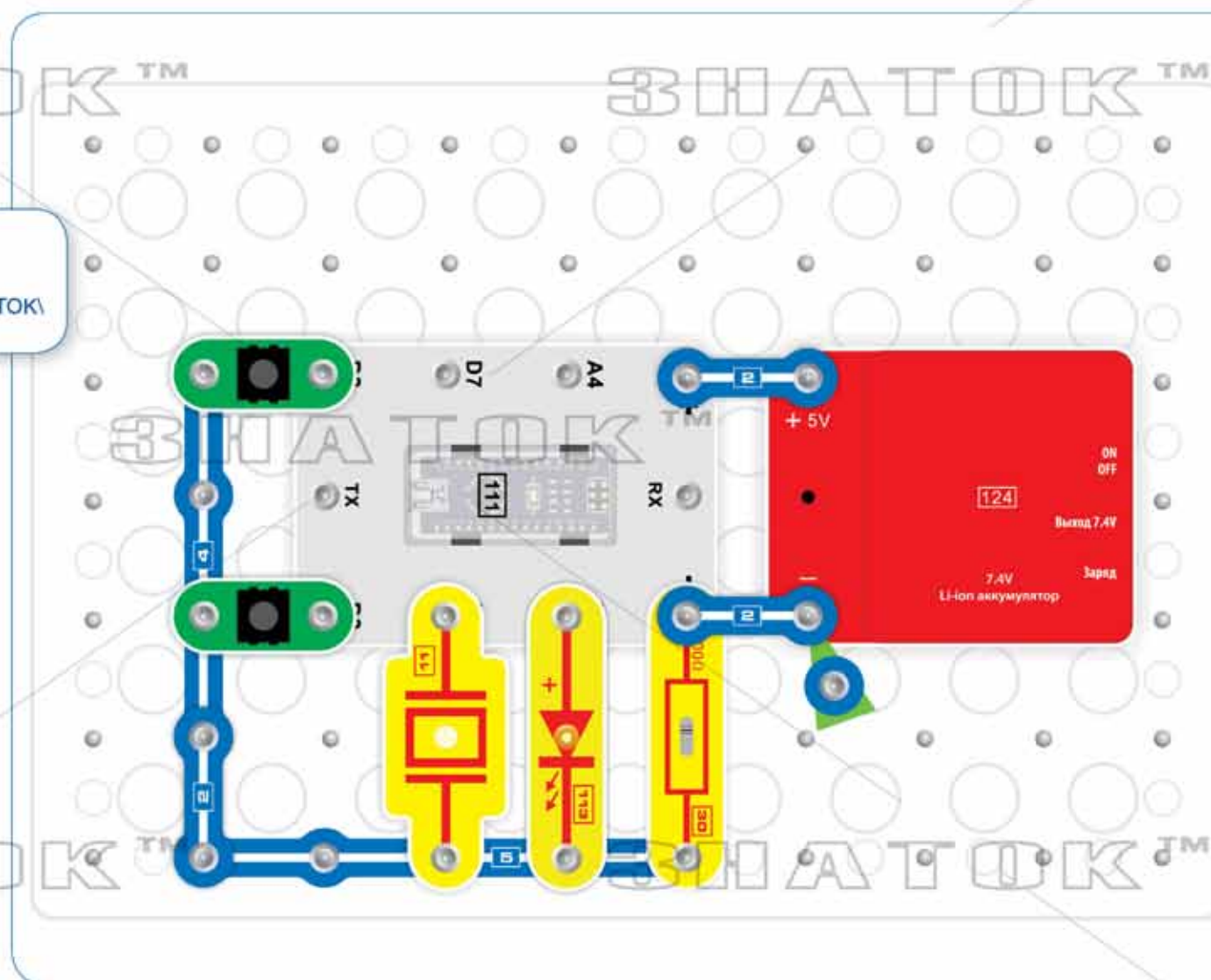
Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:

01_LightAndSound-39 —

Звук и свет.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите программу. По-
сле загрузки программы отсо-
едините USB-кабель и включите
питание — выключатель на ак-
кумуляторе **124** в положение
ON.



П Р О Г Р А М М А

01_LightAndSound-39

Данная программа реализует одновременное управление звуковыми и световыми сигналами. При нажатии на одну кнопку увеличивается частота следования звуковых сигналов и одновременно увеличивается яркость светодиода, другая же кнопка выполняет обратные действия.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

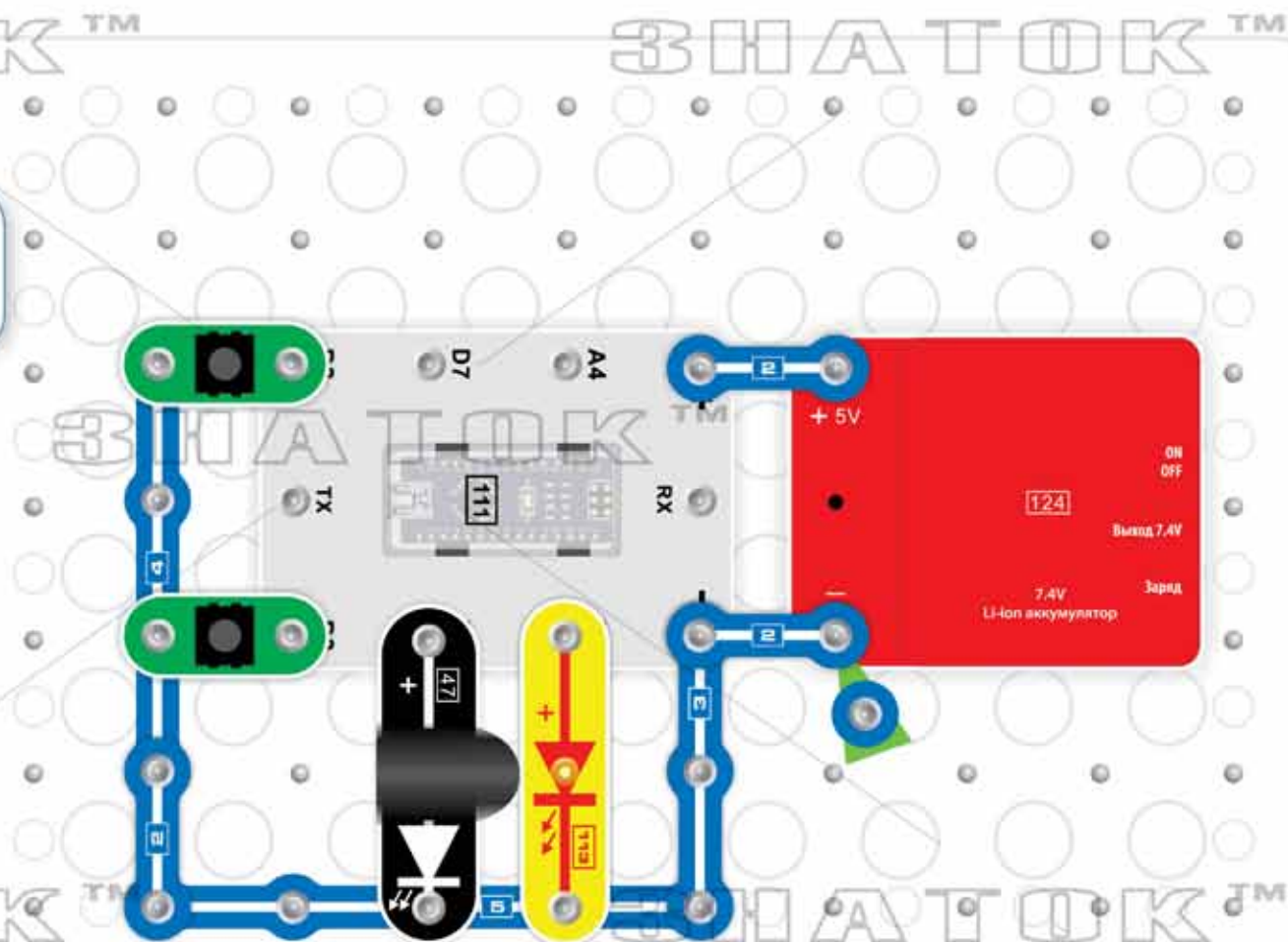
РУЧНОЕ УПРАВЛЕНИЕ
СВЕТОВЫМИ
СИГНАЛАМИ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:
01_LightAndSound-40 —
Ручной свет.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компьюте-
ру и загрузите программу. После
загрузки программы отсоеди-
ните USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



П Р О Г Р А М М А

01_LightAndSound-40

Данная программа позволяет последовательным нажатием верхней кнопки выбрать один из 5 режимов работы светодиода и фары, при этом работать они будут только если нажать и удерживать нижнюю кнопку. Список режимов:

1. Синхронное включение-выключение фары и светодиода.
2. Синхронное плавное включение-выключение фары и светодиода.
3. Попеременное включение-выключение фары и светодиода.
4. Попеременное плавное включение-выключение фары и светодиода.
5. Стробоскоп — фара быстро включается и выключается. Светодиод при этом не горит.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

СПОСОБЫ УПРАВЛЕНИЯ



Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/Электронные конструкторы ЗНАТОК/Arduino MOBILE

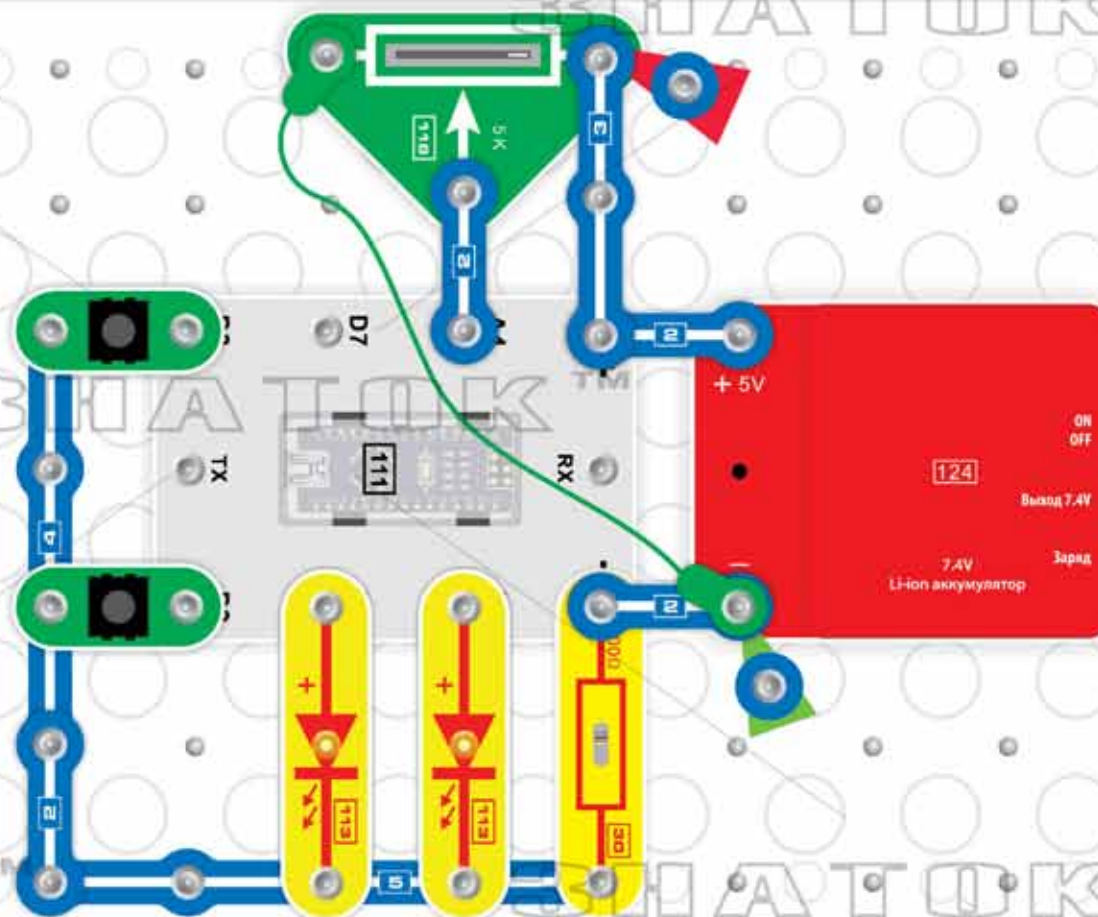
Соберите схему, как показано на рисунке. К этой схеме прилагается 2 программы:

01_LightAndSound-41 — Управление светом двумя кнопками. Вариант 1.

01_LightAndSound-42 — Управление светом двумя кнопками. Вариант 2.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **01_LightAndSound-41**. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М Ы

01_LightAndSound-41

Данная программа показывает, как разные органы управления могут отвечать за одинаковые функции. Нажатие на кнопки увеличивает или уменьшает яркость левого светодиода, а двигая ползунок переменного резистора можно изменять яркость правого.

01_LightAndSound-42

В данной программе те же самые органы управления будут выполнять другую функцию, по сравнению с предыдущей: нажатие на кнопки увеличивает или уменьшает частоту моргания левого светодиода, а двигая ползунок переменного резистора можно изменять частоту моргания правого светодиода.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

СМЕШАННОЕ
УПРАВЛЕНИЕ

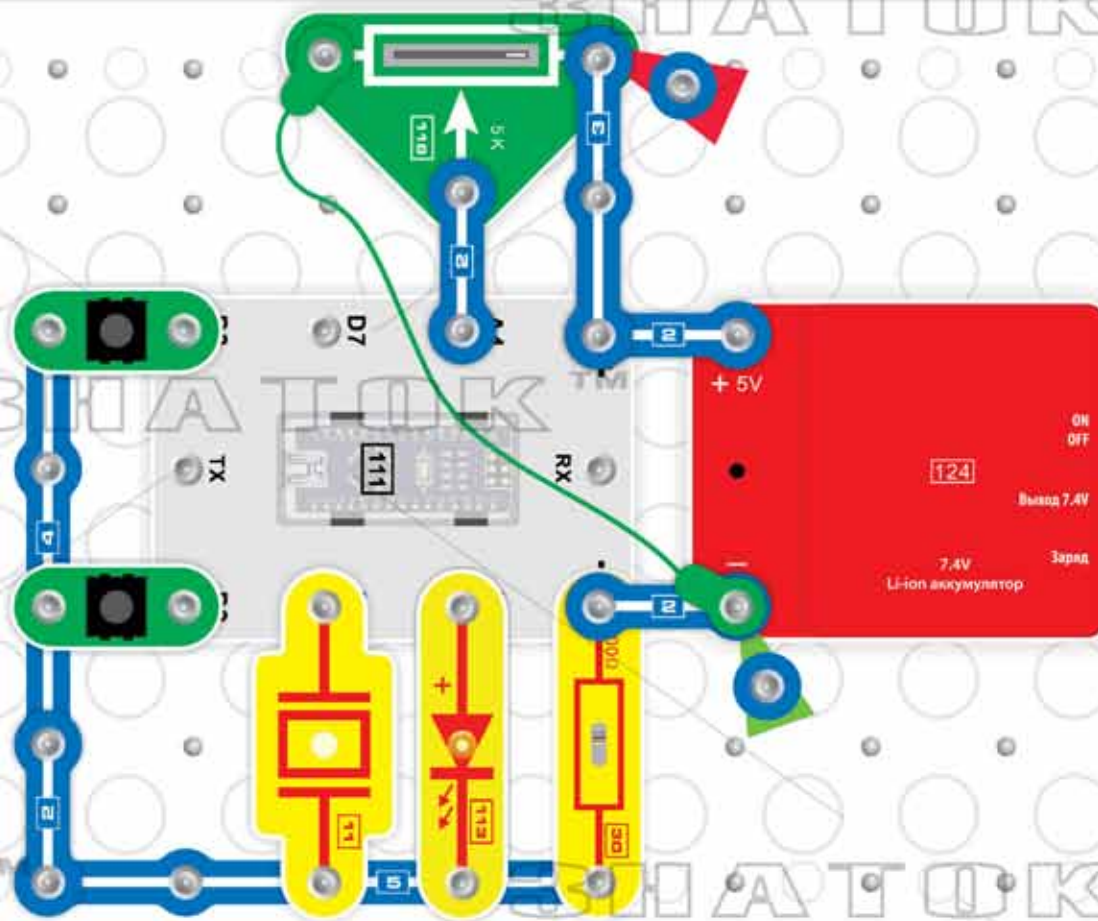
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прила-
гается 1 программа:

01_LightAndSound-43 —
Смешанное управление.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компьюте-
ру и загрузите программу. После
загрузки программы отсоеди-
ните USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



01_LightAndSound-43

«Смешанное управление». Данная программа позволяет последовательным нажатием верхней кнопки выбрать один из 4 режимов работы светодиода и фары, при этом работать они будут, только если нажать и удерживать нижнюю кнопку. Ползунком резистора **115** можно изменять один из параметров работы данного режима. Список режимов:

1. **Постоянный звуковой сигнал.**
Перемещением ползунка изменяется высота тона.
2. **Прерывистый звуковой сигнал.**
Перемещением ползунка изменяется частота сигналов.
3. **Работающий светодиод.**
Перемещением ползунка изменяется яркость.
4. **Мигающий светодиод.**
Перемещением ползунка изменяется частота моргания.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

БЛОК УПРАВЛЕНИЯ



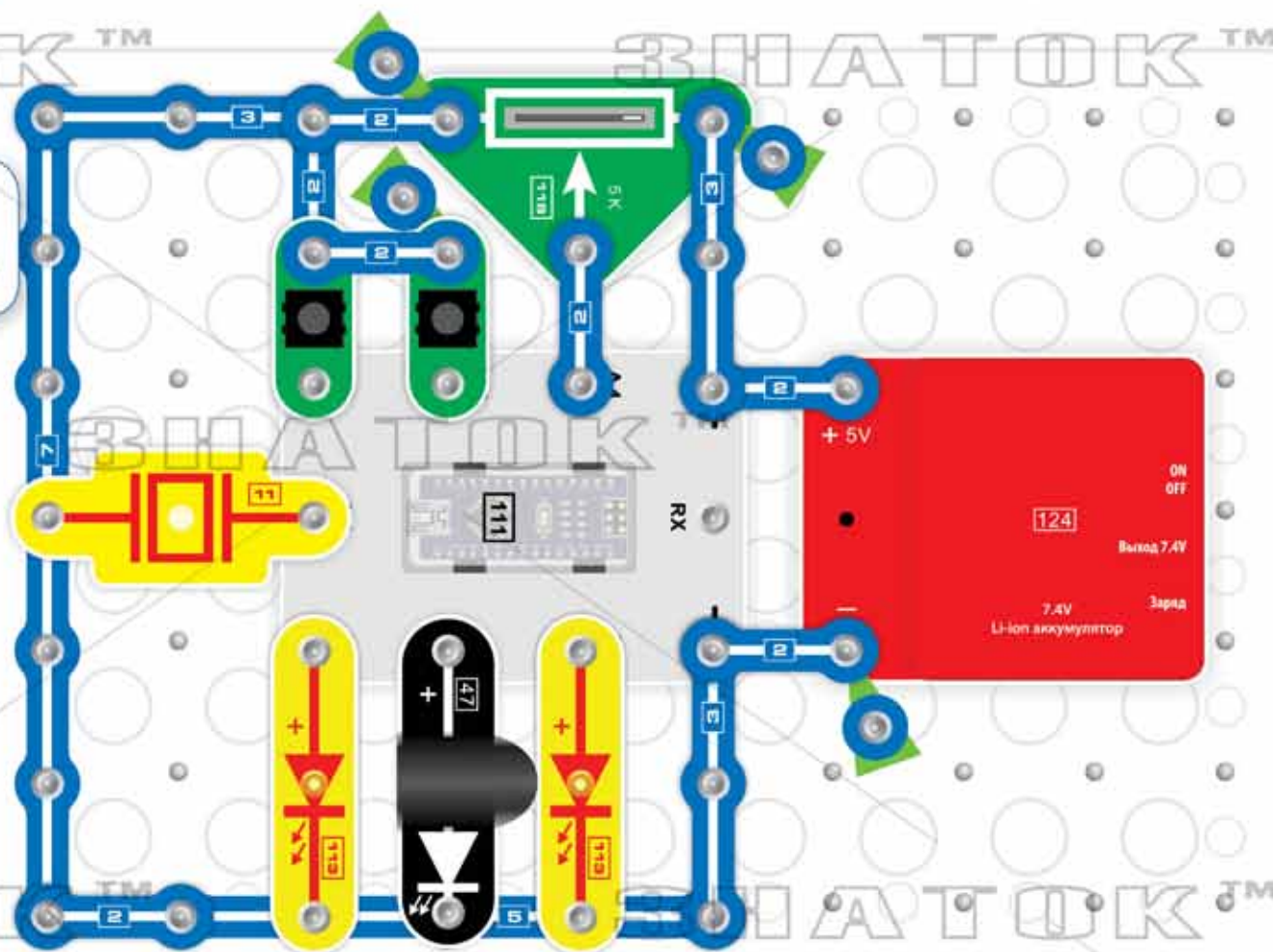
Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/ Электронные конструкторы ЗНАТОК/ Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:

01_LightAndSound-44 — Блок управления.

Полный текст программы можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

01_LightAndSound-44

ЗНАТОК™

ЗНАТОК™

«Блок управления». Данная программа имитирует работу блока управления, в котором каждому действию соответствует отдельный орган управления: светодиоды **113**, имитирующие поворотники, включаются при выставлении

крайних положений переменного резистора **118**, фара **47** работает от выключателя **15**, а нажатие на кнопку позволяет издавать звуковой сигнал.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

УПРАВЛЕНИЕ
МОТОР-РЕДУКТОРОМ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 4 программы:

02_Motors-01 — Старт-Стоп.

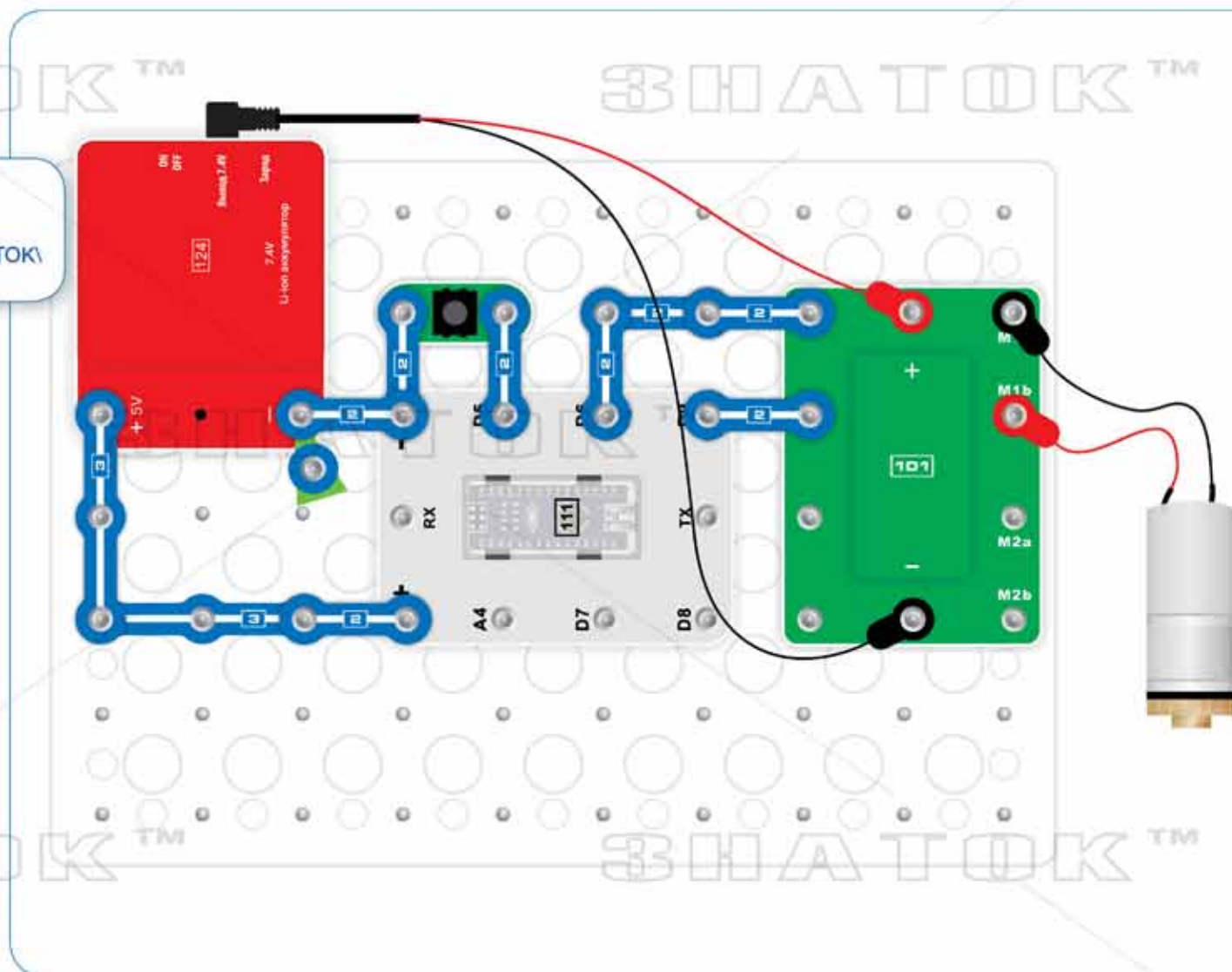
02_Motors-02 — Тормоз.

02_Motors-03 — Газ.

02_Motors-04 — Переключение
скоростей.

Полный текст программ мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111** при
помощи USB-кабеля к компью-
теру и загрузите одну из выше-
перечисленных программ. После
загрузки программы отсоедини-
те USB-кабель и включите пита-
ние — выключатель на аккумуля-
торе **124** в положение ON.



02_Motors-01

Данная программа позволяет осуществлять простое управление мотором: последовательное нажатие на кнопку включает и выключает мотор.

02_Motors-02

В данной программе показано, как может быть реализован тормоз в электромобиле.

Внимание! После загрузки программы мотор сразу начинает крутиться. При нажатии на кнопку скорость вращения постепенно падает, если кнопку отпустить — мотор снова начнет вращаться с изначальной скоростью.

02_Motors-03

Данная программа имитирует работу электродвигателя при нажатии на педаль газа в электромобиле: при нажатии — скорость быстро увеличивается, если кнопку отпустить — скорость начнет медленно уменьшаться.

02_Motors-04

«Переключение скоростей». Электродвигатель может использоваться не только в электромобилях. Например, в данной программе имитируется работа многоскоростного вентилятора: последовательное нажатие на кнопку циклически изменяет скорость работы электродвигателя (0:1:2:3 - 0:1:2:3 - 0...).

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.

Не забывайте загружать обновленную программу в модуль **111** (Arduino)!



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

ВАРИАТОР

Соберите схему, как показано на рисунке. К этой схеме прилагается 2 программы:

02_Motors-05 —

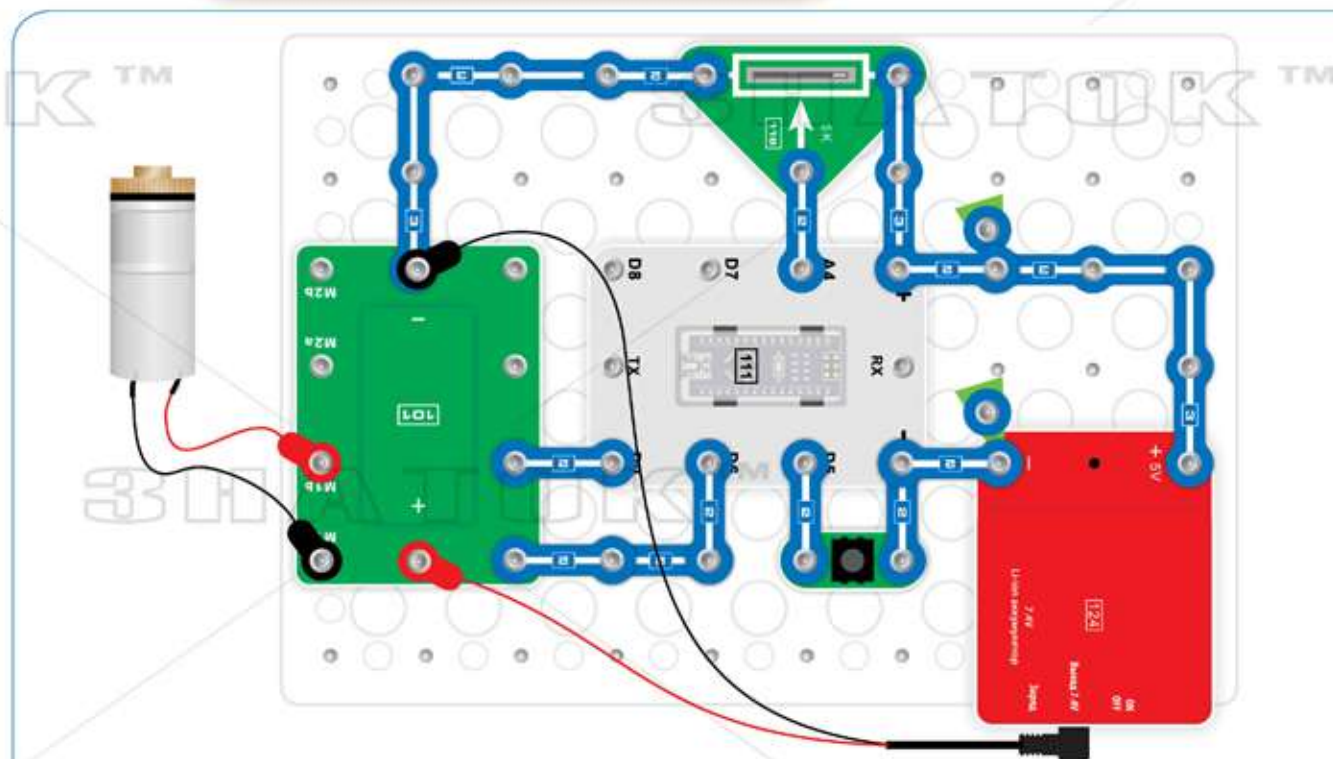
Изменение скорости.

02_Motors-06 —

Изменение скорости и направления.

Полный текст программ можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите одну из вышеперечисленных программ. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М Ы

02_Motors-05

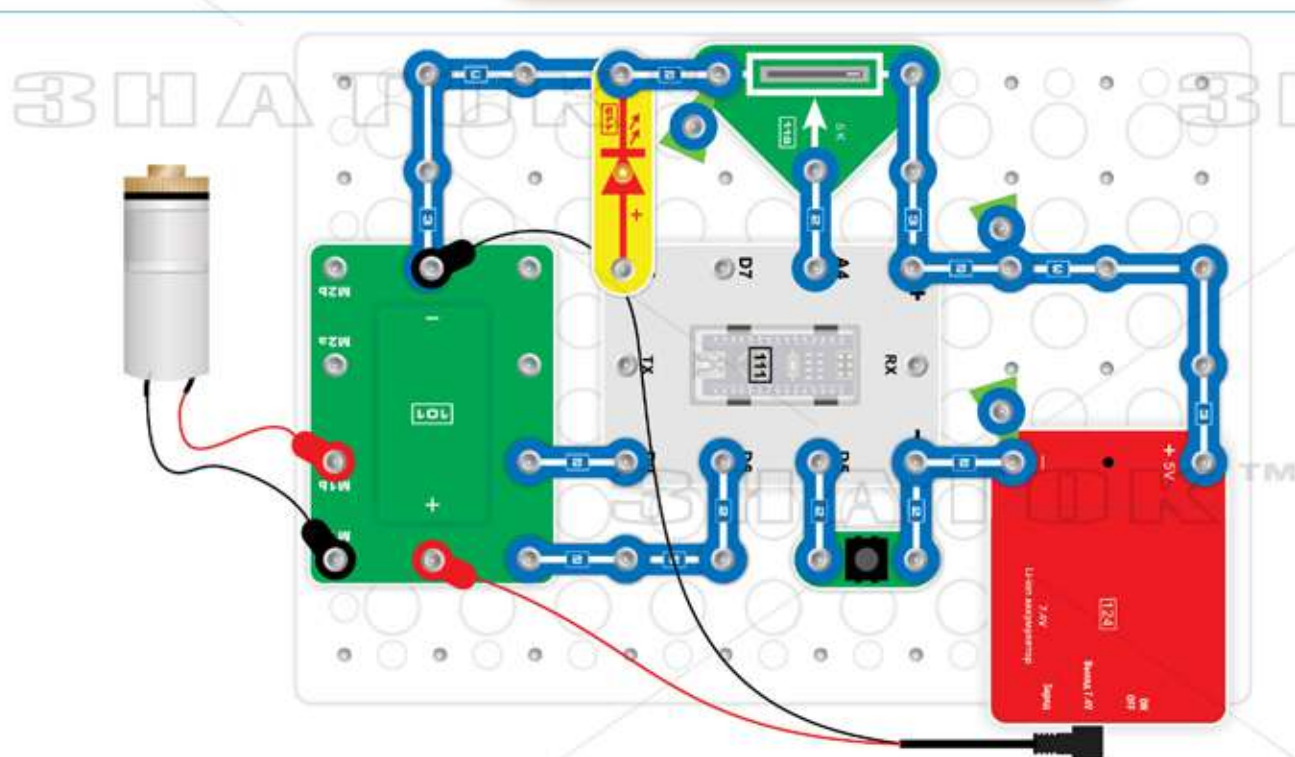
В данной программе при помощи кнопки можно включать и выключать мотор, а изменение положения ползунка переменного резистора позволяет изменить его скорость.

02_Motors-06

В данной программе при помощи кнопки можно включать и выключать мотор. Положение ползунка переменного резистора отвечает за направление вращения и скорость электродвигателя. В крайнем положении мотор вращается с максимальной скоростью, при смещении ползунка к центру его скорость уменьшается, а при переходе к другому крайнему положению, мотор начинает вращаться в противоположном направлении.



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE



ИНДИКАТОР К ВАРИАТОРУ

Соберите схему, как показано на рисунке. К этой схеме прилагается **1** программа:

02_Motors-07 —

Индикатор изменения скорости и направления вращения.

Полный текст программы можно посмотреть на прилагаемой флэшке.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **02_Motors-07**. После загрузки программы отсоедините USB-кабель и включите питание — выключатель на аккумуляторе **124** в положение ON.

П Р О Г Р А М М А

02_Motors-07

В данной программе при помощи кнопки можно одновременно включать и выключать мотор и светодиод. Положение же ползунка пе-

ременного резистора отвечает за направление вращения и скорость электродвигателя. В крайнем положении ползунка мотор вращается с максимальной скоростью, а светодиод горит с максимальной яркостью, при смещении ползунка к центру скорость мотора уменьшается, а при переходе к другому крайнему положению, мотор начинает вращаться в противоположном, а светодиод моргать.

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

КАЛИБРОВКА
СЕРВО-ПРИВОДА

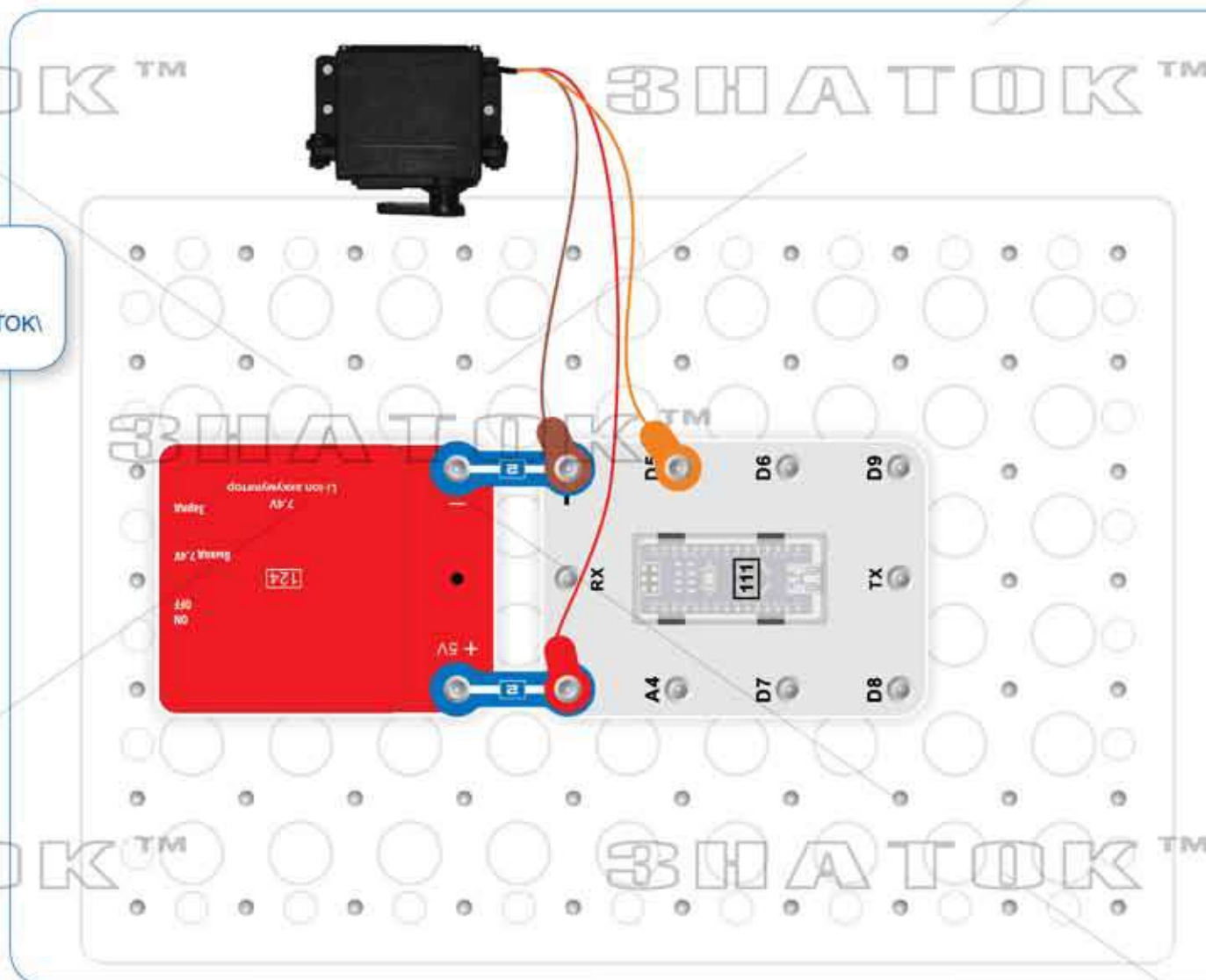
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:

02_Motors-08 — Калибровка.

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
02_Motors-08. После загрузки
программы отсоедините USB-
кабель и включите питание —
выключатель на аккумуляторе
124 в положение ON.



02_Motors-08

ЗНАТОК™

ЗНАТОК™



Одна из важнейших программ! Данная программа позволяет проверить работоспособность серво-привода и откалибровать крайние положения его вала. Научиться это делать совершенно необходимо для управления положением колес, когда серво-привод будет установлен на платформу и подсоединен к колесам рулевыми тягами (рычагами). Ниже приведен фрагмент программы, позволяющей установить крайние положения вала серво-привода.

**Крайнее правое положение**

Выставить значение от 0 до 180 (градусов).

Обратите внимание, что для корректной работы данное значение должно быть больше значения крайне-левого положения

...

```
const int servo_right_pos = 180;
```

```
const int servo_left_pos = 0;
```

...

Крайнее левое положение

Выставить значение от 0 до 180 (градусов).

Обратите внимание, что для корректной работы данное значение должно быть меньше значения крайне-правого положения

ЗНАТОК™

ЗНАТОК™

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

РУЧНОЕ УПРАВЛЕНИЕ СЕРВО-ПРИВОДОМ

Соберите схему, как показано на рисунке.
К этой схеме прилагается 2 программы:

02_Motors-09 —

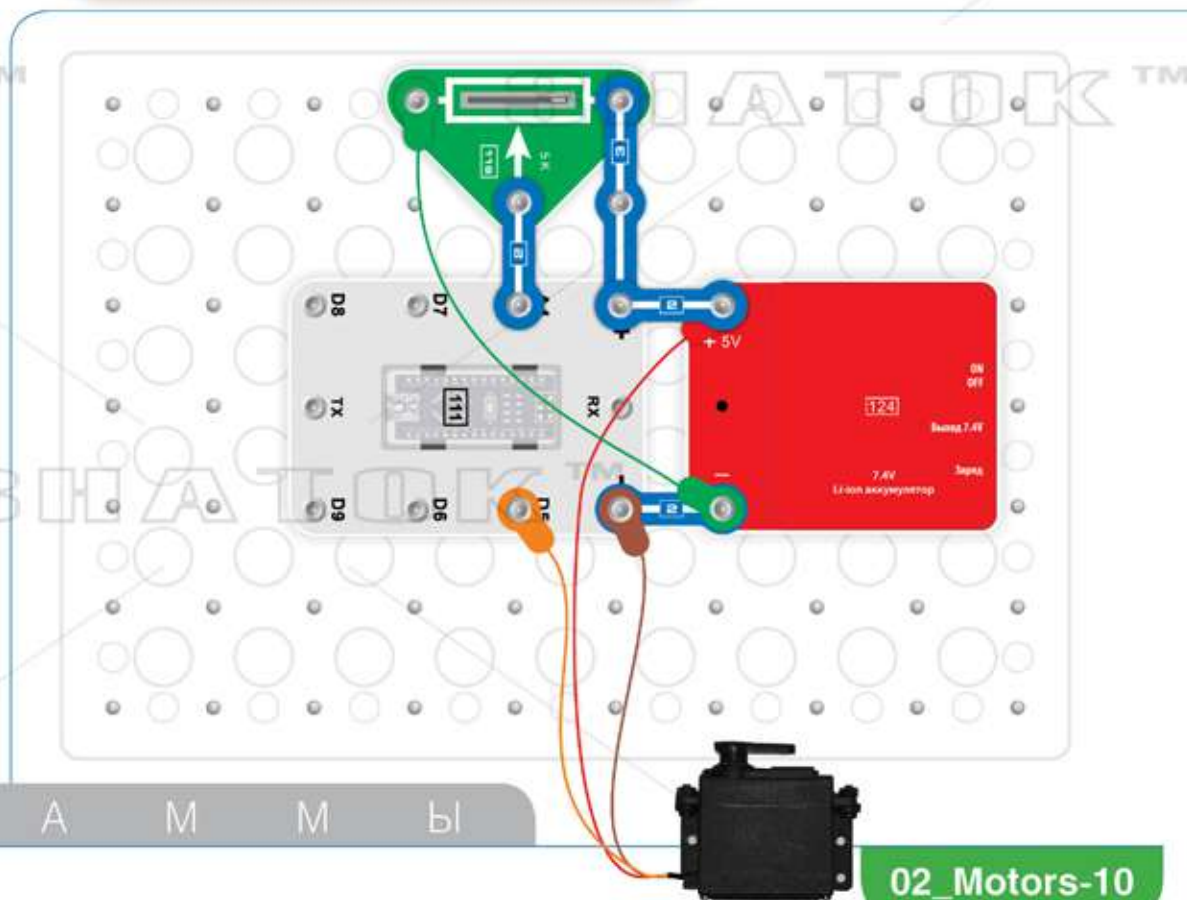
Плавное изменение положения.

02_Motors-10 —

Ступенчатое изменение положения.

Полный текст программ можно посмотреть
на прилагаемой флэшке.

Подключите модуль **111** при помощи
USB-кабеля к компьютеру и загрузите одну
из вышеперечисленных программ. После за-
грузки программы отсоедините USB-кабель
и включите питание — выключатель на акку-
муляторе **124** в положение ON.



П Р О Г Р А М М Ы

02_Motors-09

В данной программе возможно ручное управление положением вала серводвигателя. В зависимости от положения ползунка переменного резистора плавно изменяется угол поворота.

02_Motors-10

В данной программе возможно ручное управление положением вала серводвигателя, но, в отличие от предыдущей программы, здесь имеется всего три заранее заданных положения вала серво-привода. В крайнем левом положении ползунка переменного резистора вал находится в положении 0° , в среднем положении — 90° , а в крайнем правом — 180° .

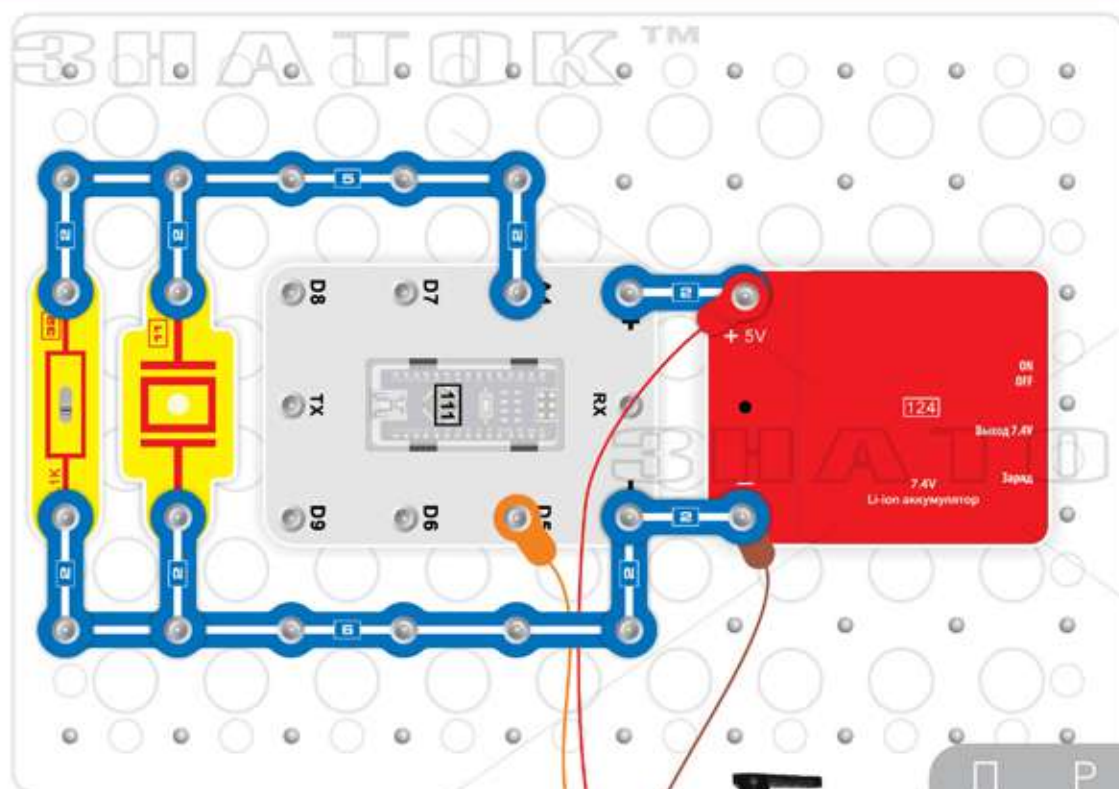


Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

АВАРИЙНАЯ РАЗБЛОКИРОВКА ДВЕРЕЙ

Соберите схему, как показано на рисунке.
К этой схеме прилагается **1** программа:
02_Motors-11 — Разблокировка дверей

Подключите модуль **111** при помощи
USB-кабеля к компьютеру и загрузите про-
грамму **02_Motors-11**. После загрузки про-
граммы отсоедините USB-кабель и вклю-
чите питание — выключатель на аккумуляторе
124 в положение ON.



После каждого изменения в программе,
сохраняйте обновленную версию
с новым именем в библиотеке ваших
проектов.

Не забывайте загружать обновленную
программу в модуль **111** (Arduino)!

П Р О Г Р А М М А

02_Motors-11

После запуска данной программы, если ударить по пьезоизлучателю **11**, имитируя столкновение, то произойдет поворот вала серво-привода, имитирующее открывание замка блокировки дверей.

АВТОМОБИЛЬНЫЕ
ДВОРНИКИ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

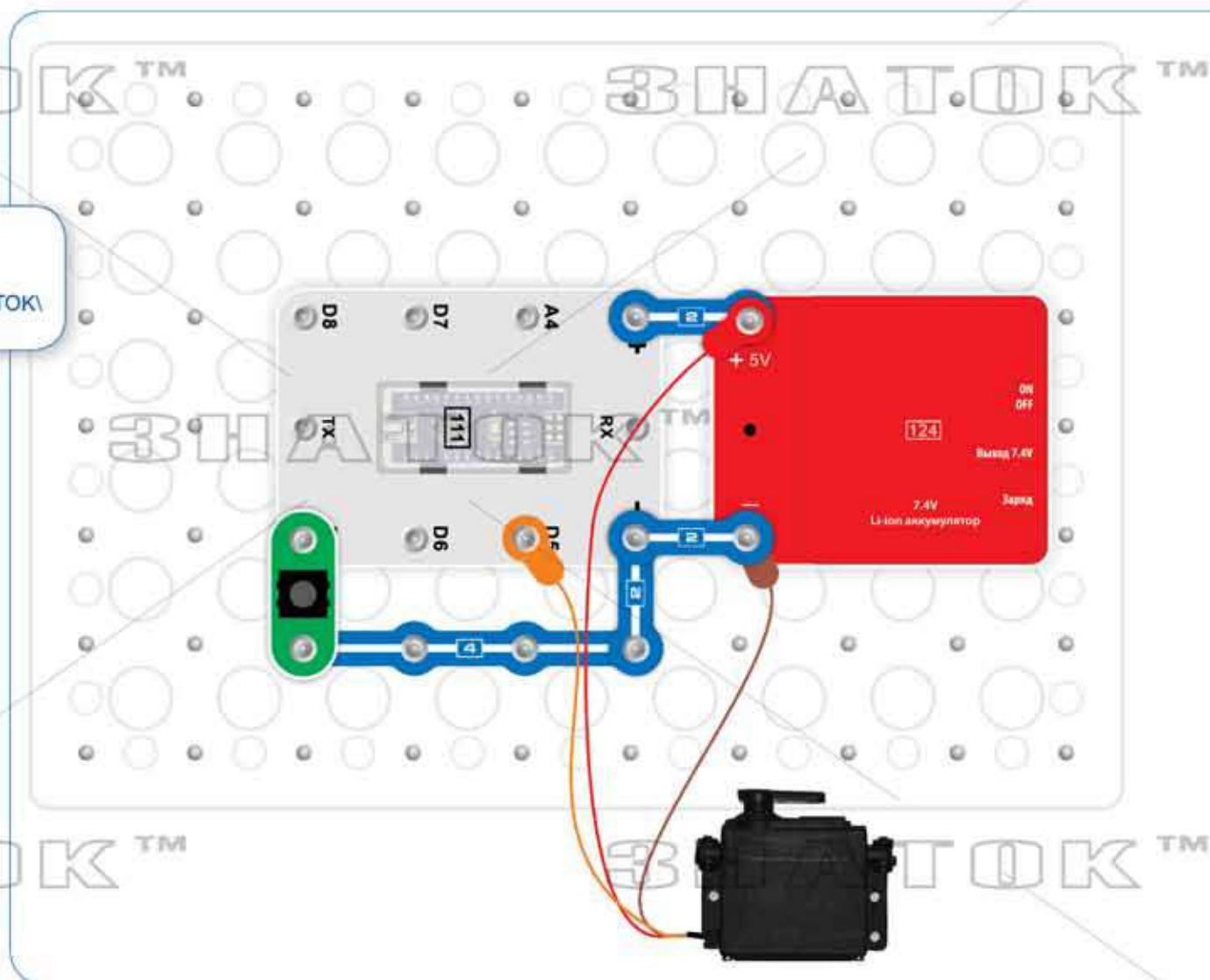
Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется **1** программа:

02_Motors-12 —

Управление дворниками

Полный текст программы мож-
но посмотреть на прилагаемой
флэшке.

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
02_Motors-12. После загрузки
программы отсоедините USB-
кабель и включите питание —
выключатель на аккумуляторе
124 в положение ON.



02_Motors-12

Данная программа позволяет имитировать выбор режима работы дворников при помощи одной кнопки. Двойное нажатие на кнопку циклично переключает режимы работы:

1. Если в данном режиме один раз нажать на кнопку — дворники сработают один раз.
2. Если в данном режиме один раз нажать на кнопку — дворники сработают три раза.
3. Нажатие на кнопку в данном режиме включает/выключает постоянную работу дворников.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

СЕМЬ РЕЖИМОВ

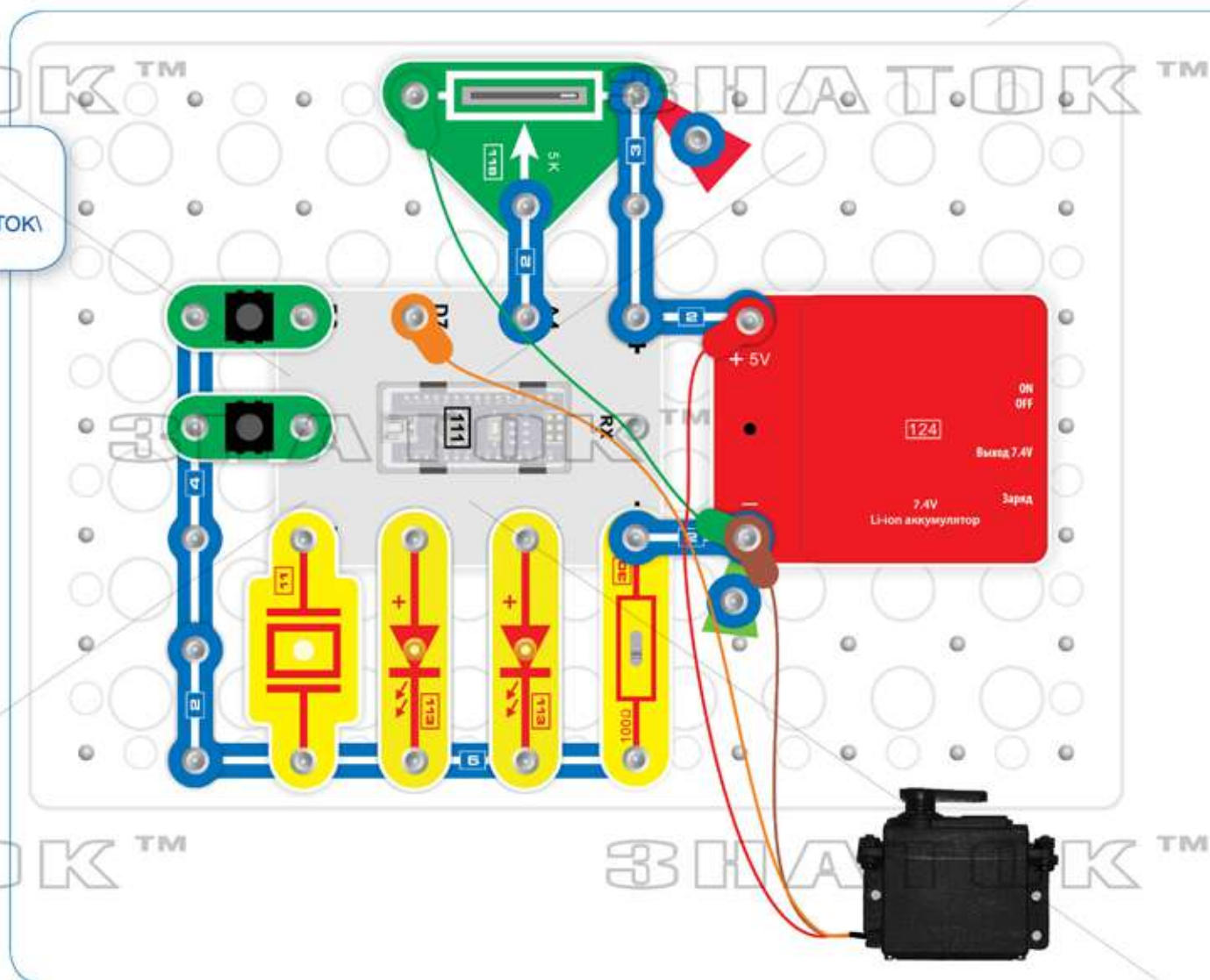


Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Соберите схему, как показано
на рисунке. К этой схеме прилага-
ется 1 программа:

02_Motors-13 — Семь режимов

Подключите модуль **111**
при помощи USB-кабеля к ком-
пьютеру и загрузите программу
02_Motors-13. После загрузки
программы отсоедините USB-
кабель и включите питание —
выключатель на аккумуляторе
124 в положение ON. По умол-
чанию будет выставлен первый
из семи нижеперечисленных ре-
жимов. Для выбора другого ре-
жима нажимайте на верхнюю кноп-
ку. Далее, одной рукой нажмите
и удерживайте нижнюю кнопку,
а другой рукой меняйте поло-
жение ползунка резистора **115**.



02_Motors-13

Данная программа позволяет последовательным нажатием верхней кнопки выбрать один из 7 режимов работы светодиодов, пьезоизлучателя и серво-привода, при этом работать они будут только если нажата нижняя кнопка. Ползунком резистора можно изменять один из параметров работы данного режима. Список режимов:

1. **Постоянный звуковой сигнал.**
Ползунком потенциометра изменяется высота тона.
2. **Прерывистый звуковой сигнал.**
Ползунком потенциометра изменяется частота сигналов.
3. **Работающие светодиоды.**
Ползунком потенциометра изменяется яркость.
4. **Мигающие светодиоды.**
Ползунком потенциометра изменяется частота моргания.
5. **Ручное управление серводвигателем.**
Ползунком потенциометра выбирается положение вала серводвигателя.
6. **Скорость серводвигателя.**
Ползунком потенциометра выбирается скорость вращения вала серводвигателя.
7. **Мигающие огни.**
Ползунком потенциометра выбирается частота переключения двух попеременно мигающих светодиодов.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов.
Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

УСТАНОВКА ПРОГРАММЫ УПРАВЛЕНИЯ НА МОБИЛЬНОЕ УСТРОЙСТВО ANDROID

В данной главе в схемах используется Bluetooth модуль **98**, и для его функционирования необходимо установить на своё мобильное устройство (смартфон или планшет) приложение Znatok.Mobile.

Для этого, считайте приведенный здесь QR-код или перейдите на сайт http://znatok.ru/link/?mobile_app и следуйте указанным там инструкциям для установки.



После установки приложения у вас на мобильном устройстве должен появиться такой значок:



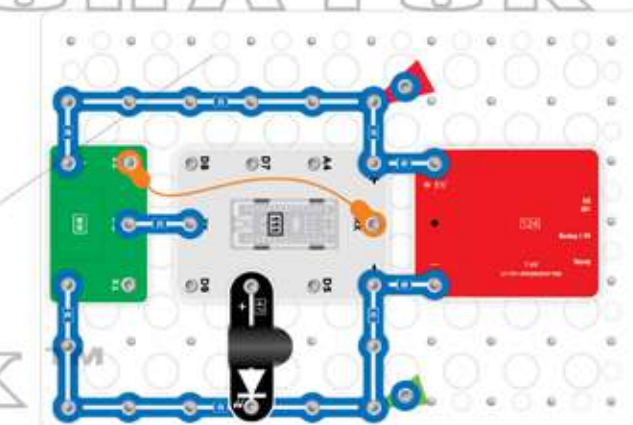
После этого необходимо произвести настройку модуля **98** Bluetooth.

НАСТРОЙКА МОДУЛЯ BLUETOOTH ДЛЯ РАБОТЫ

Активируйте на вашем мобильном устройстве Bluetooth. При поиске устройств Bluetooth на вашем мобильном устройстве отобразится новое устройство с именем «MLT-BT05» — это имя вашего модуля **98**.

Подсоедините модуль **111** Arduino при помощи USB-кабеля к компьютеру и откройте на нем программу **03 Bluetooth 0000** в редакторе Arduino IDE. Найдите в тексте программы этот фрагмент:

```
...  
String uartName = "ZNATOK_MOBILE";  
...
```



Новое имя модуля **98**

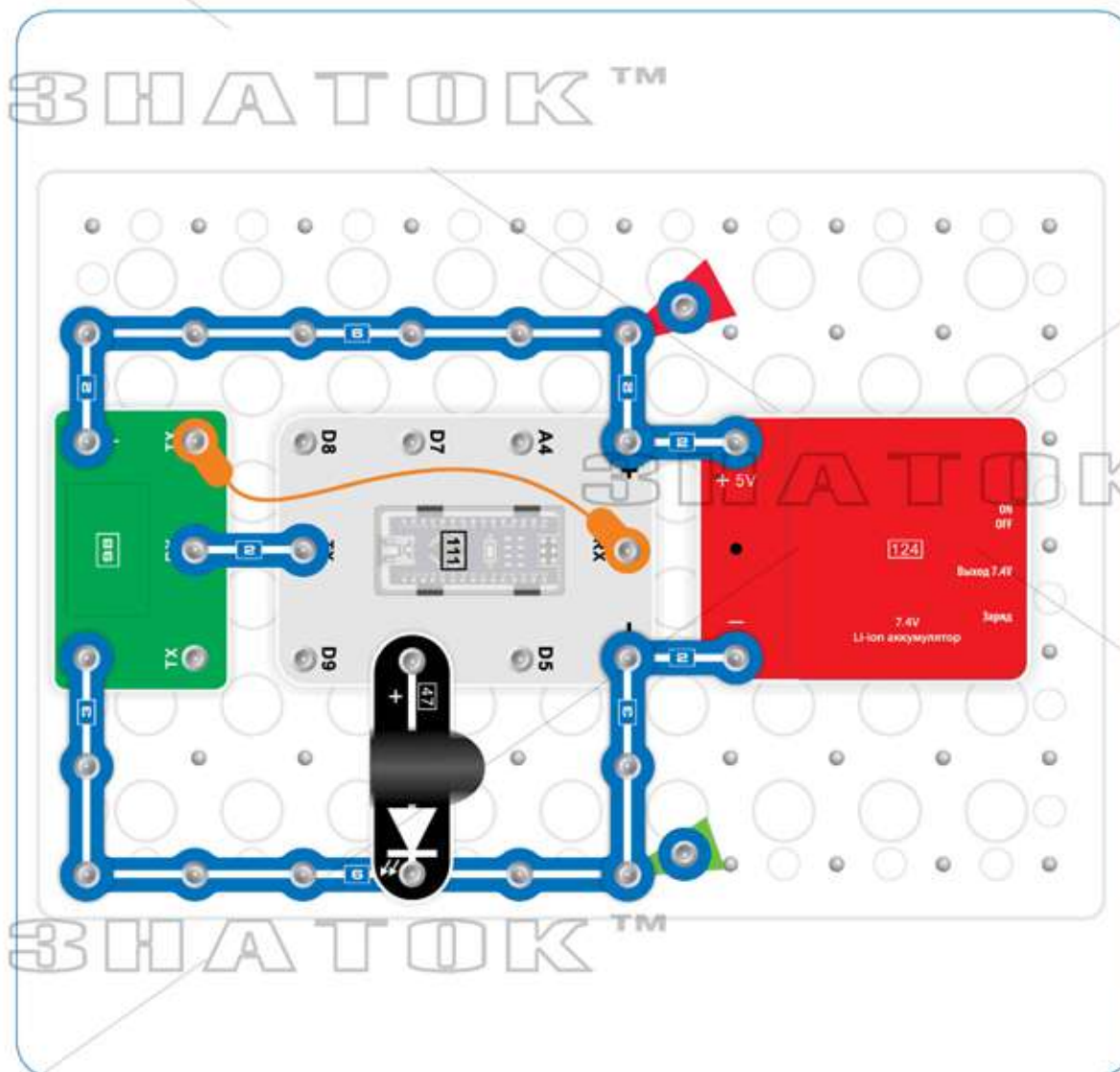
Измените данное имя на новое (сохранив кавычки). Длина имени может составлять максимум 32 символа и не должна содержать буквы русского алфавита.

Измените указанный фрагмент на новое имя и загрузите программу. Соберите данную схему («Дистанционное управление светодиодом») и включите питание — выключатель на аккумуляторе **124** в положение ON.

После этого на мобильном устройстве запустите приложение, дотронувшись до значка:



Далее следуйте инструкциям, появляющимся на вашем смартфоне или планшете, с учётом заданного вами нового имени модуля **98**.

ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ
СВЕТОДИОДОМ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **03_Bluetooth-01** — «**Дистанционное управление светодиодом**». После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. Открывшийся пульт управления позволит дистанционно переводить фару в режим «маяка» или изменять её яркость.

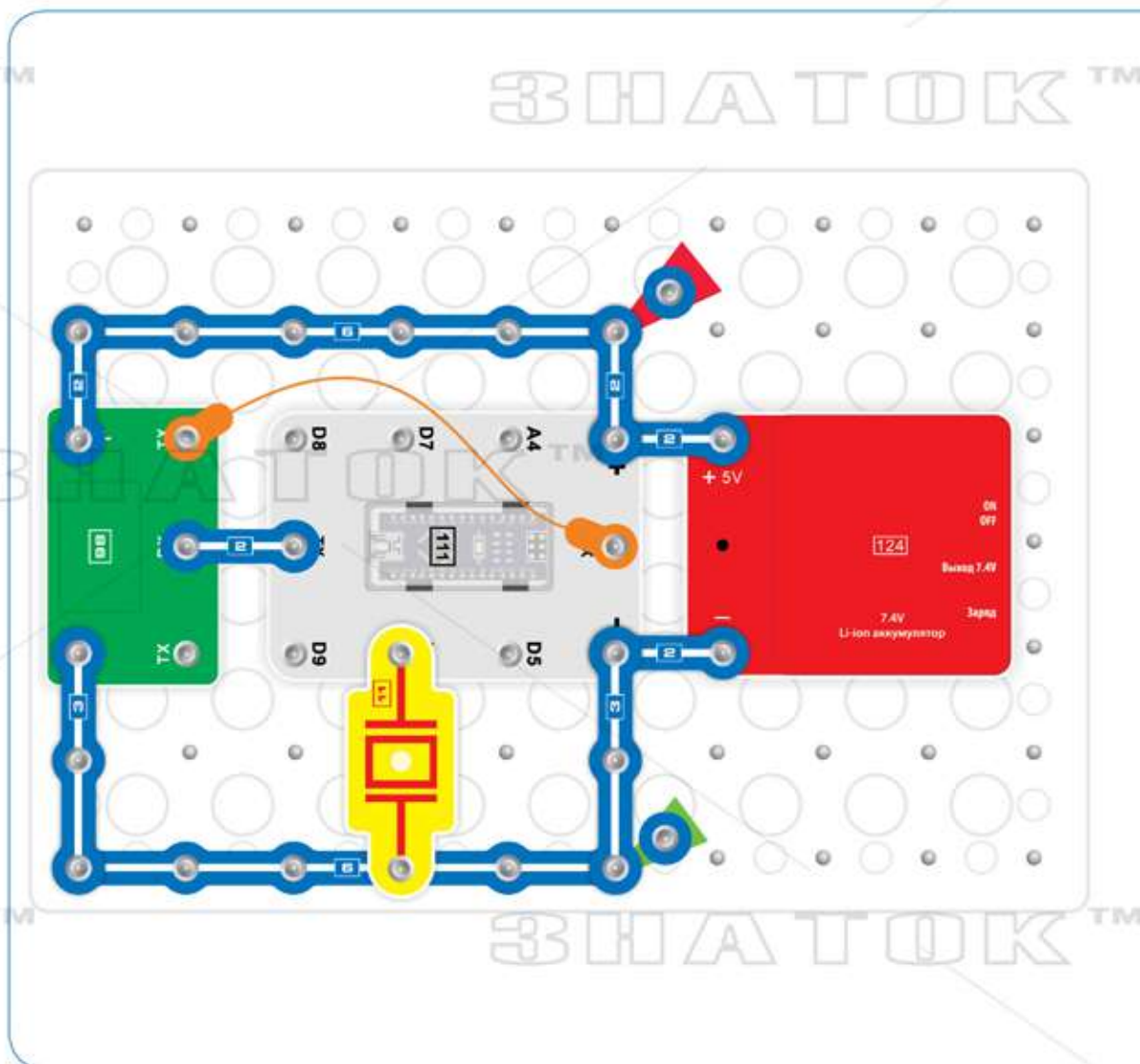
Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.

ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ
ПЬЕЗОИЗЛУЧАТЕЛЕМ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **03_Bluetooth-02** — Дистанционное управление пьезоизлучателем. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. Открывшийся пульт управления позволит включать пьезоизлучатель, а так же изменять высоту тона работы.

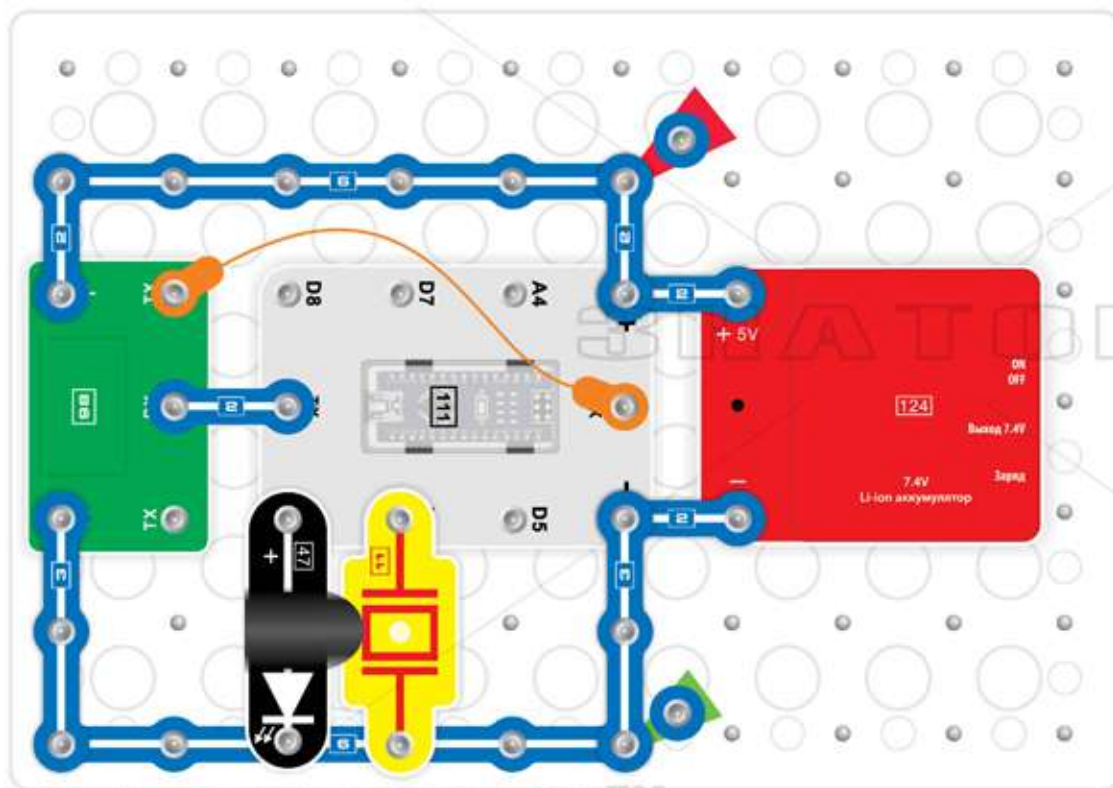
Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ
НЕСКОЛЬКИМИ УСТРОЙСТВАМИ™

ЗНАТОК™

ЗНАТОК™



Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **03_Bluetooth-03** — Комбинированное управление. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. Открывшийся пульт управления позволит изменять высоту тона звука пьезоизлучателя и яркость фары.

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.

ЗНАТОК™

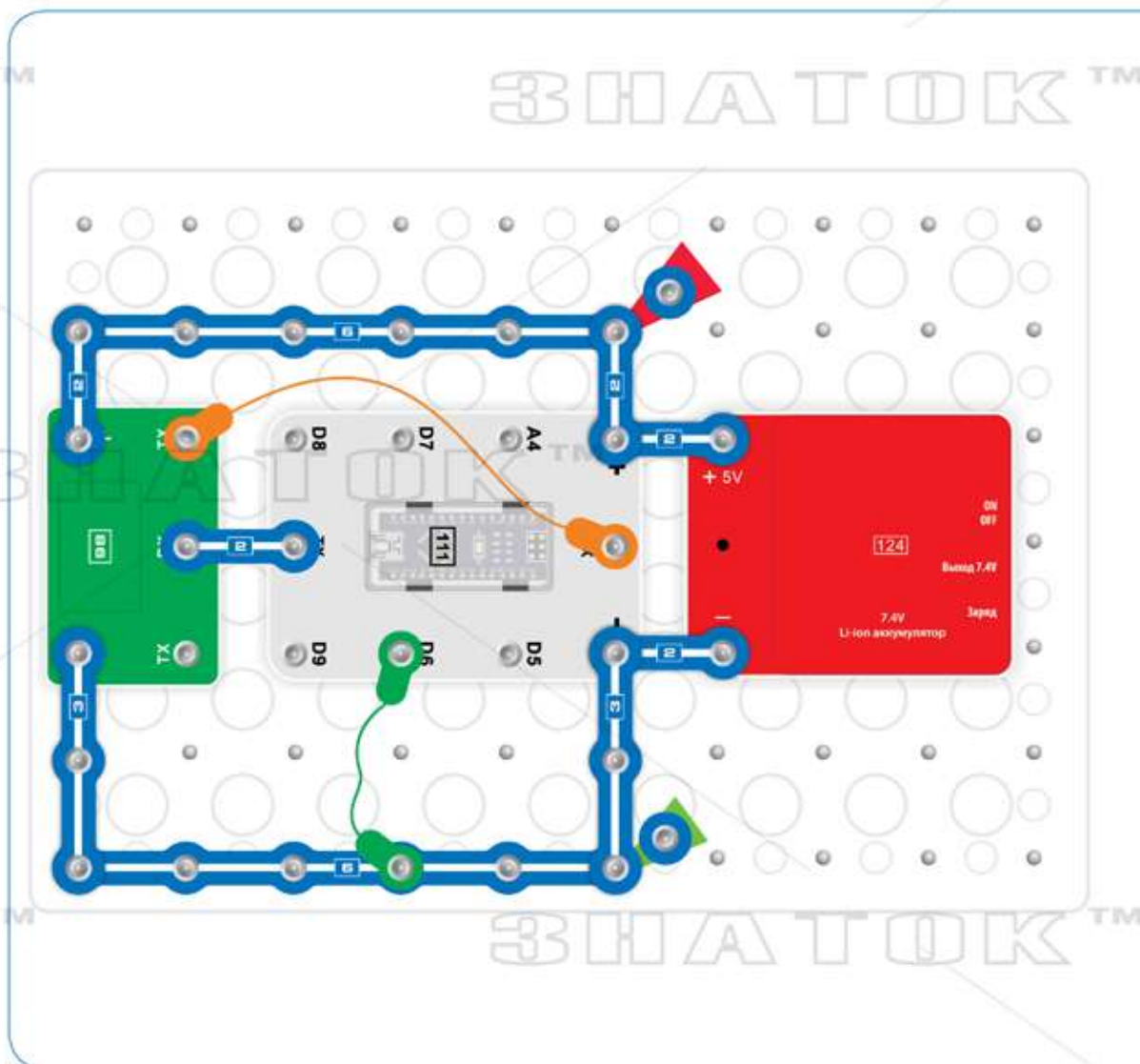
ЗНАТОК™

ДИСТАНЦИОННАЯ СИГНАЛИЗАЦИЯ НА РАЗРЫВ

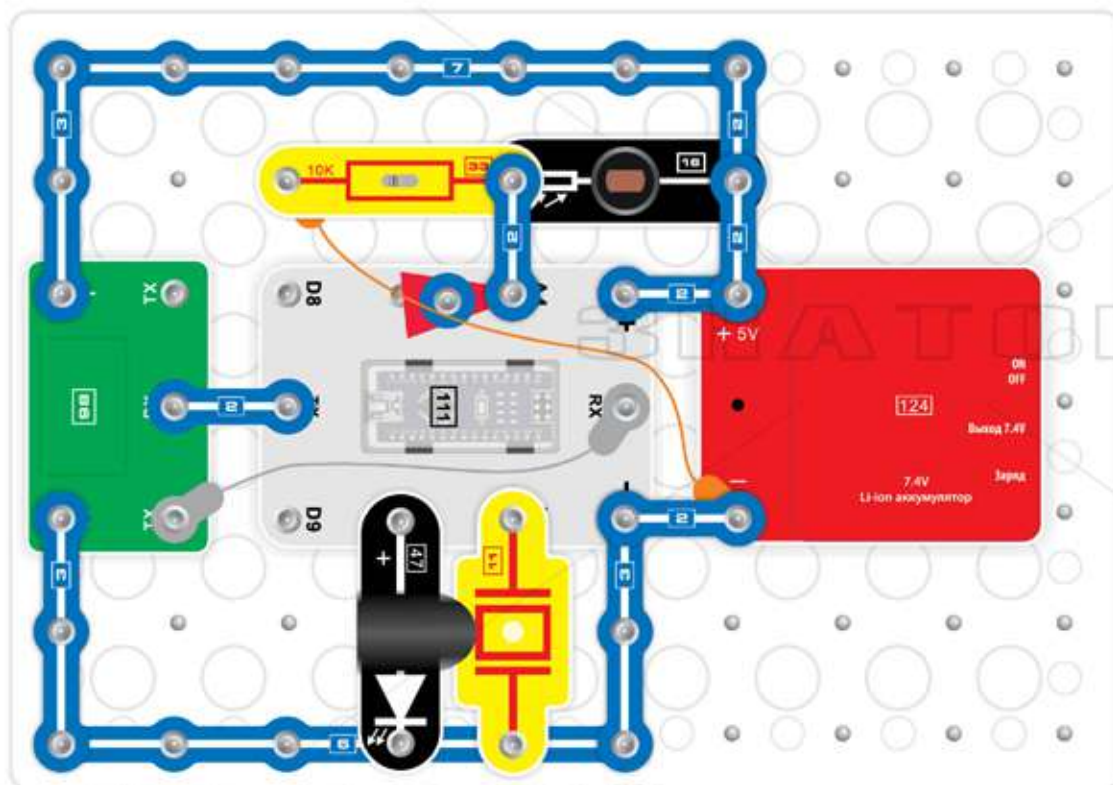
Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу **TX**. Загрузите программу **03_Bluetooth-04** — Дистанционная сигнализация на разрыв. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. После включения сигнализации на смартфоне (ON), сигнализация переходит в рабочий режим. Если всё нормально, то индикатор горит зеленым (OK). При отключении зеленого провода на схеме, подключенного к выводу D6, сработает сигнализация, и это отразится на вашем мобильном устройстве — индикатор изменит цвет на красный (ALARM) и раздастся звуковой сигнал (при включенном звуке на мобильном устройстве).

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



ЗНАТОК™



ЗНАТОК™

ДИСТАНЦИОННАЯ СИГНАЛИЗАЦИЯ НА СВЕТ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **03_Bluetooth-05** — Дистанционная сигнализация на свет. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. На появившемся пульте управления выберите условие срабатывания сигнализации — на свет (**СОЛНЫШКО**) или на темноту (**МЕСЯЦ**). После включения сигнализации на смартфоне (ON), сигнализация переходит в рабочий режим. Если всё нормально, то индикатор горит зеленым (OK), если происходит срабатывание, то индикатор изменит цвет на красный (ALARM) и раздастся звуковой сигнал (при включенном звуке на мобильном устройстве). Кроме того, соответствующие звуковые и световые сигналы будет издавать сама схема.

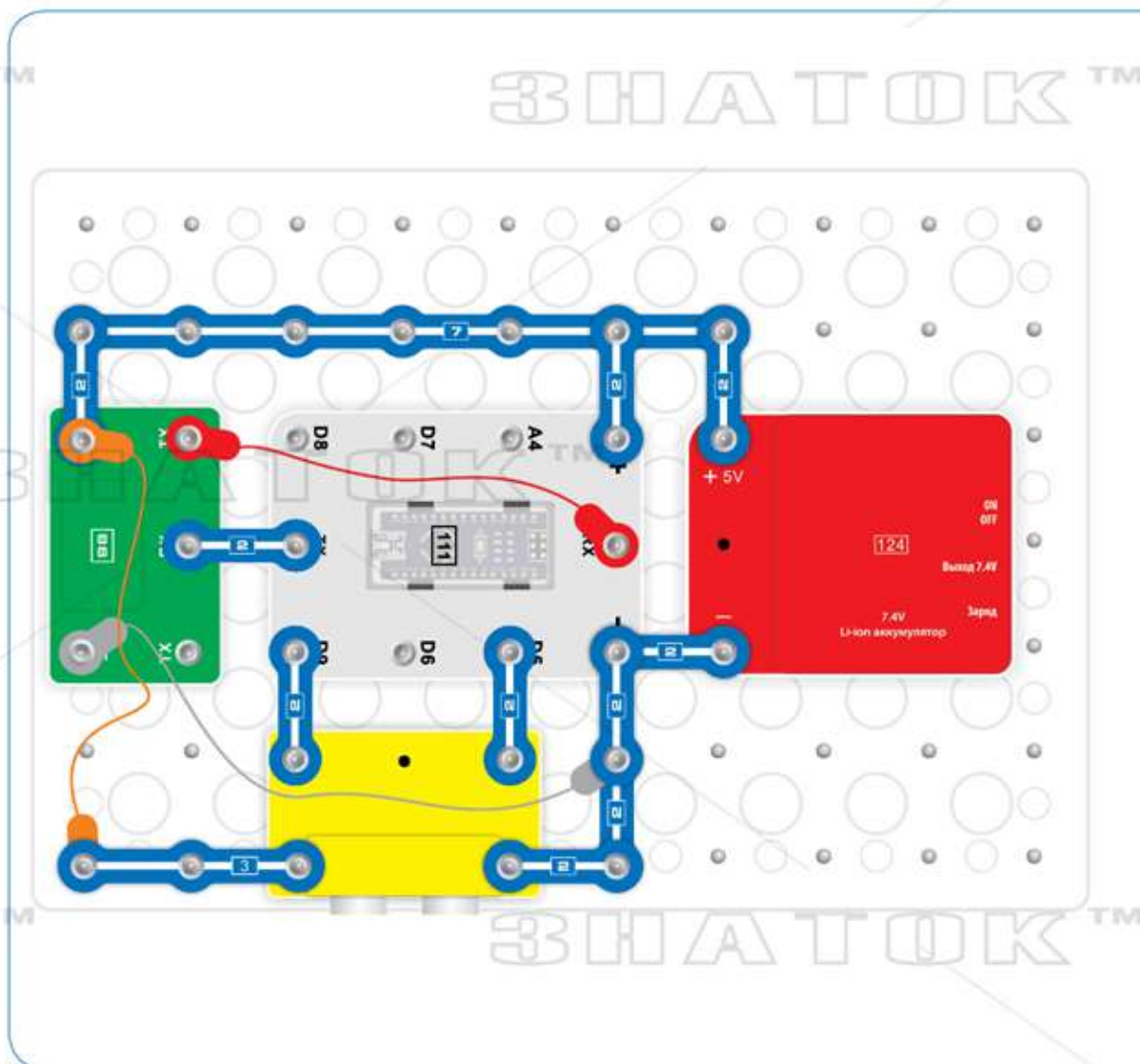
Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.

ДИСТАНЦИОННАЯ СИГНАЛИЗАЦИЯ НА ДВИЖЕНИЕ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу **TX**. Загрузите программу **03_Bluetooth-06** — Дистанционная сигнализация на движение. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. После включения сигнализации на смартфоне (ON), сигнализация переходит в рабочий режим. Если всё нормально, то индикатор горит зеленым (OK). Если перед ультразвуковым датчиком 89 произойдет движение, то сигнализация сработает и это отразится на вашем мобильном устройстве — индикатор изменит цвет на красный (ALARM) и раздастся звуковой сигнал (при включенном звуке на мобильном устройстве).

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



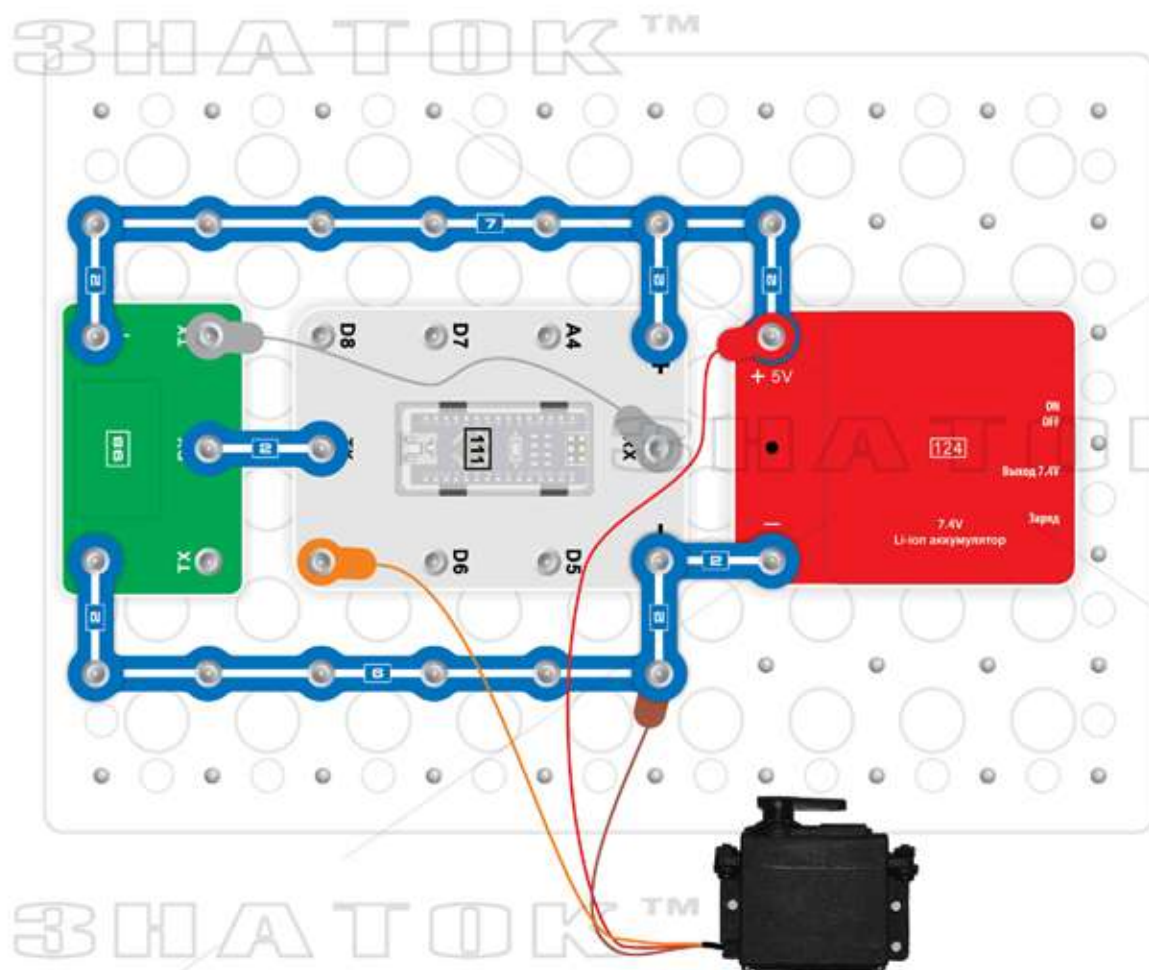
ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ
СЕРВО-ПРИВОДОМ

Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу **TX**. Загрузите программу **03_Bluetooth-07** — Дистанционное управление серво-приводом. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile**. Открывшийся пульт управления позволит дистанционно изменять положение вала серво-привода.

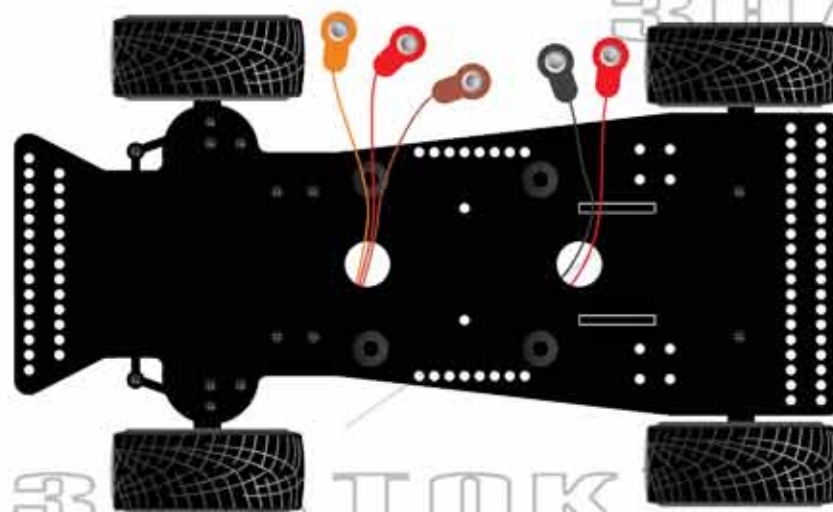
Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



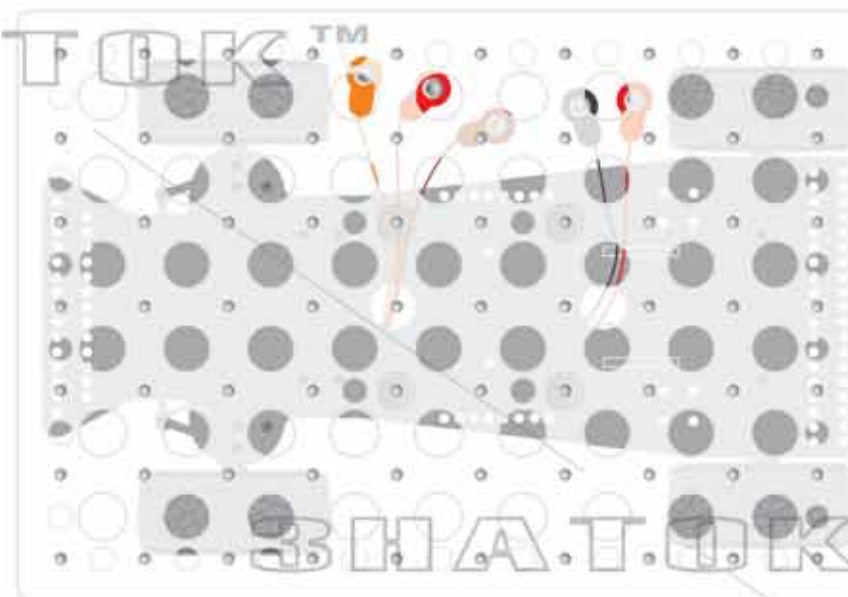


В данном наборе используется видео-инструкция по сборке ходовой части нашего электромобиля. Для просмотра вы можете воспользоваться тремя способами:

- 1.** Отсканируйте данный QR-код и просмотрите инструкцию на мобильном устройстве:
- 2.** Перейдите на сайт http://znatok.ru/link/?Arduino-Mobile_Chassis_Finish
- 3.** На прилагаемой флешке откройте файл [Mobile_Video-instruction.mp4](#)



БЕЗ ПЛАТЫ



С УСТАНОВЛЕННОЙ ПЛАТОЙ

УСТАНОВКА ПРОГРАММЫ УПРАВЛЕНИЯ НА МОБИЛЬНОЕ УСТРОЙСТВО ANDROID

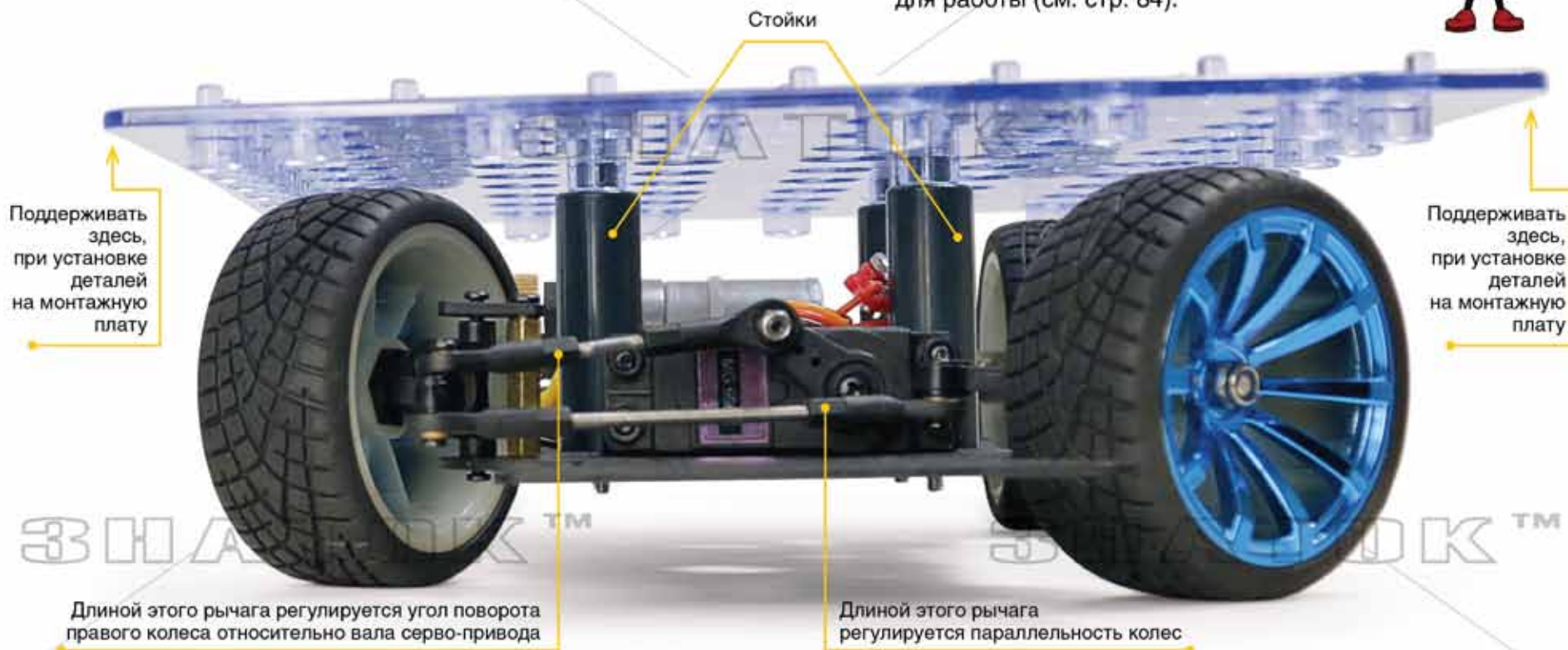
В данной главе в схемах используется Bluetooth модуль **98**, и для его функционирования необходимо установить на своё мобильное устройство (смартфон или планшет) приложение Znatok Mobile. Для этого, считайте приведенный здесь QR-код или перейдите на сайт http://znatok.ru/link/?mobile_app и следуйте указанным там инструкциям для установки.

После установки приложения у вас на мобильном устройстве должен появиться такой значок:



После этого необходимо произвести настройку модуля **98** Bluetooth для работы (см. стр. 84).

Здесь
можно скачать
приложение

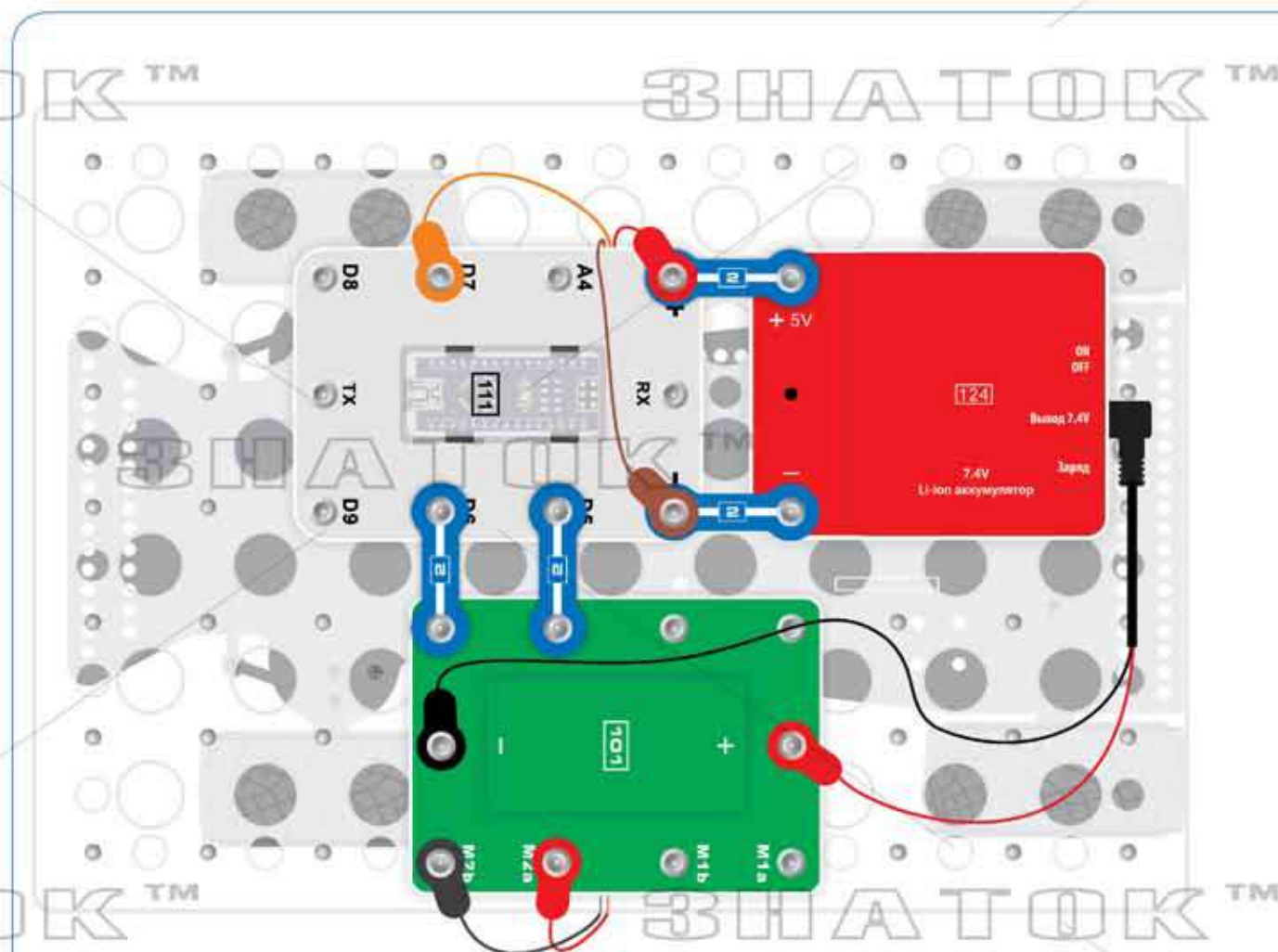


НАСТРОЙКА
ЭЛЕКТРОМОБИЛЯ

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:

04_Mobile-01 — Настройка электромобиля.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-01**. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!

04_Mobile-01

Данная программа позволяет настроить и проверить работу установленных на платформу мотор-редуктора и серво-привода. Используя программу «Калибровка серво-привода», вы можете отрегулировать крайнее правое и крайнее левое положение передних колес. После включения, платформа начинает циклично выполнять следующие движения (вы можете поставить платформу на пол или держать колеса на весу):

- Выставляет передние колеса строго для движения прямо
- 1 секунду едет вперед
- 1 секунду едет назад
- Поворачивает передние колеса в крайнее правое положение
- Едет по дуге, по часовой стрелке около 2 секунд
- Поворачивает передние колеса в крайнее левое положение
- Едет по дуге, против часовой стрелки около 2 секунд

Если одно из этих движений выполняется некорректно, то проверьте правильность выставления параметров программы из проекта «Калибровка серво-привода», а также проверьте правильность сборки схемы.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль 111 (Arduino)!

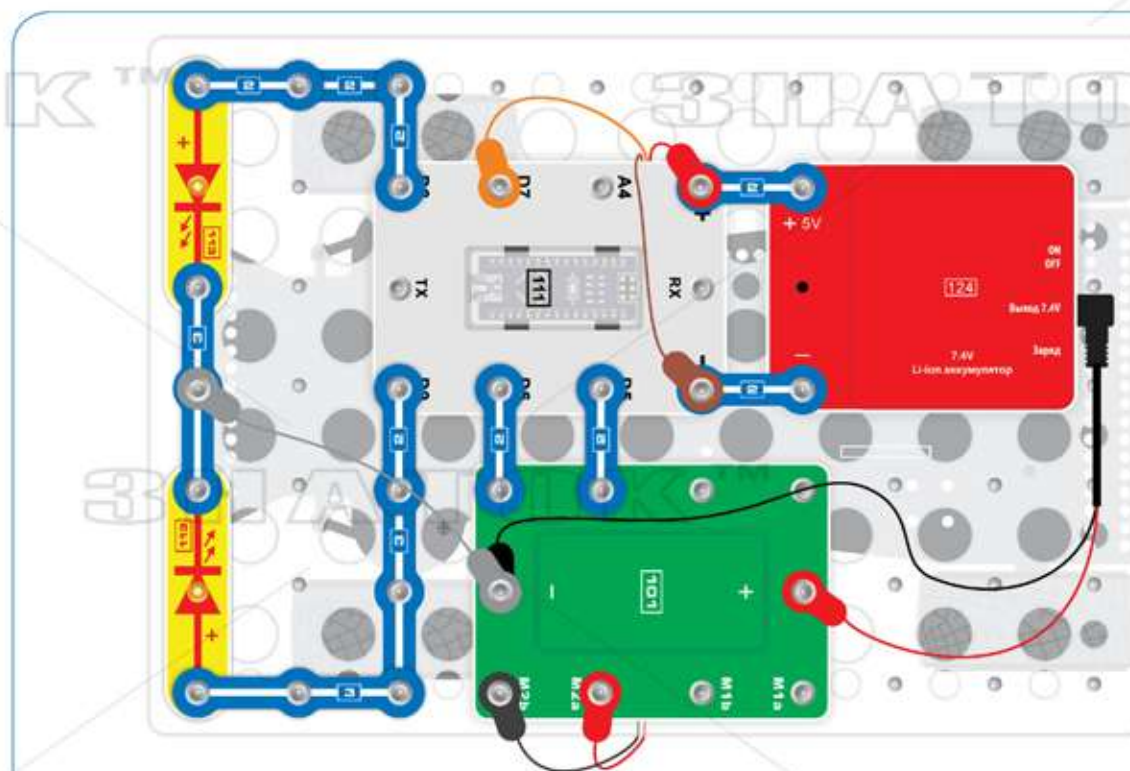
УКАЗАТЕЛИ ПОВОРОТА

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:

04_Mobile-02 —

Указатели поворота.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-02**. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



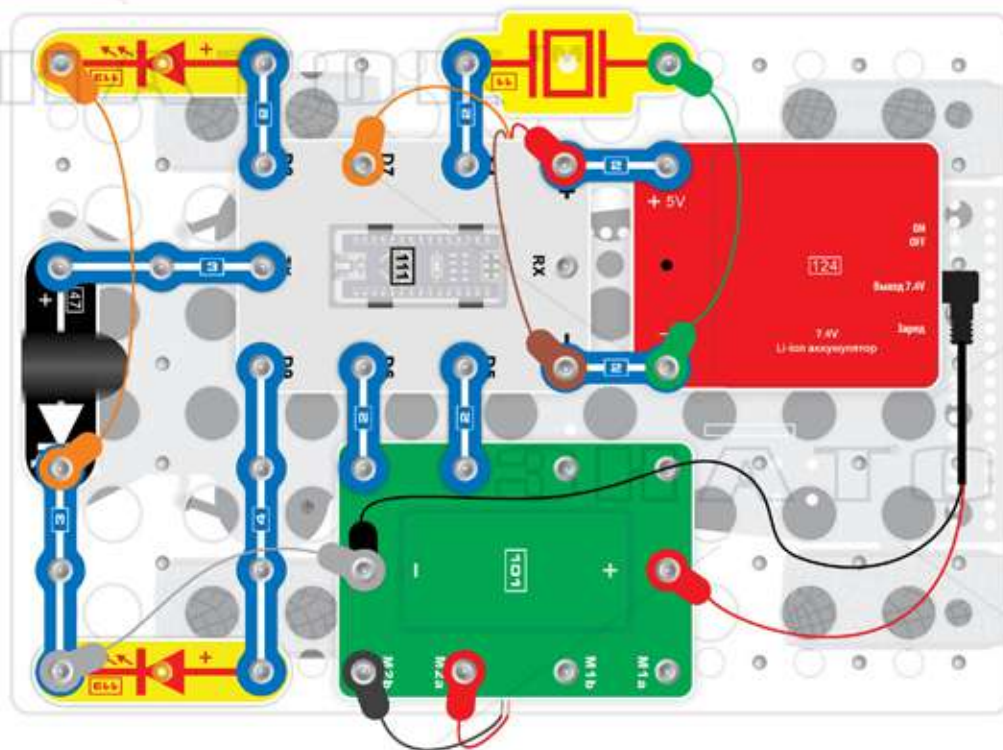
П Р О Г Р А М М А

04_Mobile-02

После включения питания, электромобиль начнет движение по прямой, затем он включит правый поворотный

сигнал, остановится, повернет направо, продолжит движение, выключит поворотный сигнал. Затем электромобиль продолжит движение по прямой, включит левый поворотный сигнал, остановится, повернет налево, продолжит движение, выключит поворотный сигнал, остановится. После остановки оба сигнала начнут синхронно моргать, имитируя сигнал «аварийная остановка».

Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!



УКАЗАТЕЛИ ПОВОРОТА СО ЗВУКОВЫМ СИГНАЛОМ

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:
04_Mobile-03 —
Поворотники со звуковым сигналом.

Временно удалите провод **3**, идущий к выводу TX модуля **111**. Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-03**. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **3** и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.

П Р О Г Р А М М А

04_Mobile-03

После включения питания, электромобиль начнет движение по прямой, затем он включит правый поворотный сигнал, остановится, по-

вернет направо, продолжит движение, выключит поворотный сигнал. Затем электромобиль продолжит движение по прямой, включит левый поворотный сигнал, остановится, повернет налево, продолжит движение, выключит поворотный сигнал, остановится. После остановки оба сигнала начнут синхронно моргать, имитируя сигнал «аварийная остановка». Каждое действие, при этом, будет сопровождаться соответствующим звуковым сигналом из пьезоизлучателя.

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

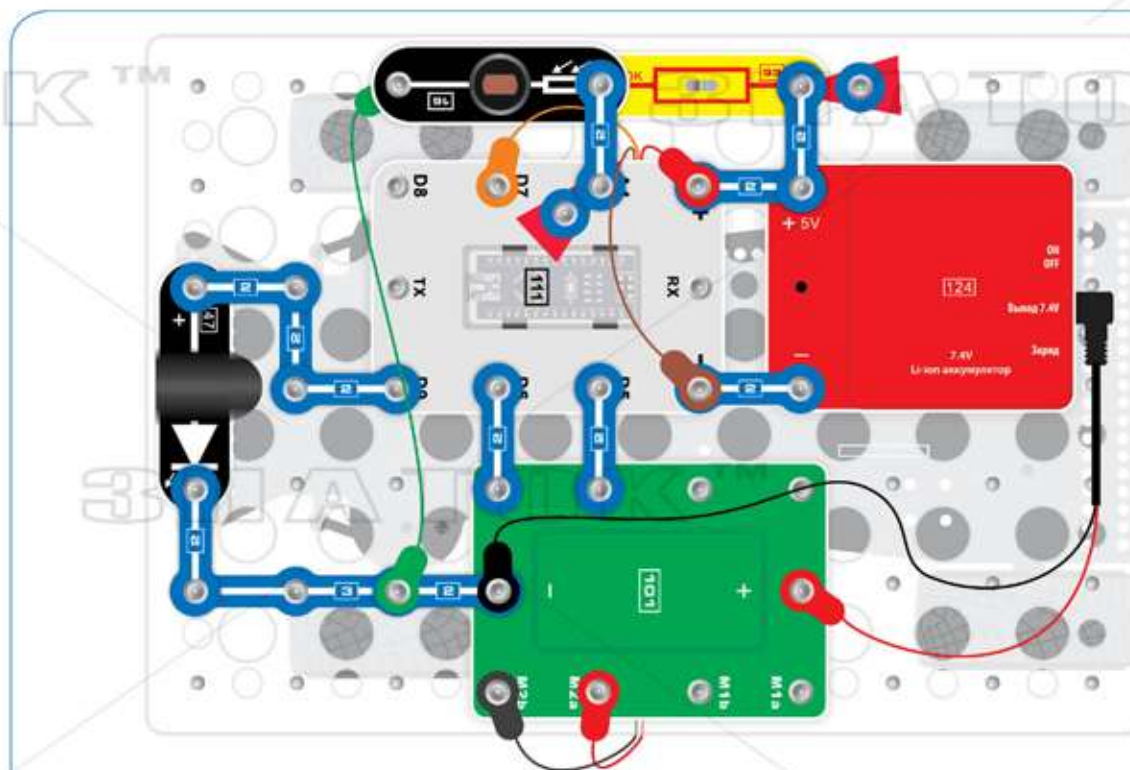
АВТОМАТИЧЕСКИЕ ФАРЫ

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа:

04_Mobile-04 —

Автоматические фары.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-04**. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

04_Mobile-04

После включения питания, электромобиль начнет медленно двигаться по кругу по часовой стрелке. Если во время движения закрыть резистор **16** рукой или выключить

свет в помещении, то включится фара, расположенная в передней части платформы.

Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!

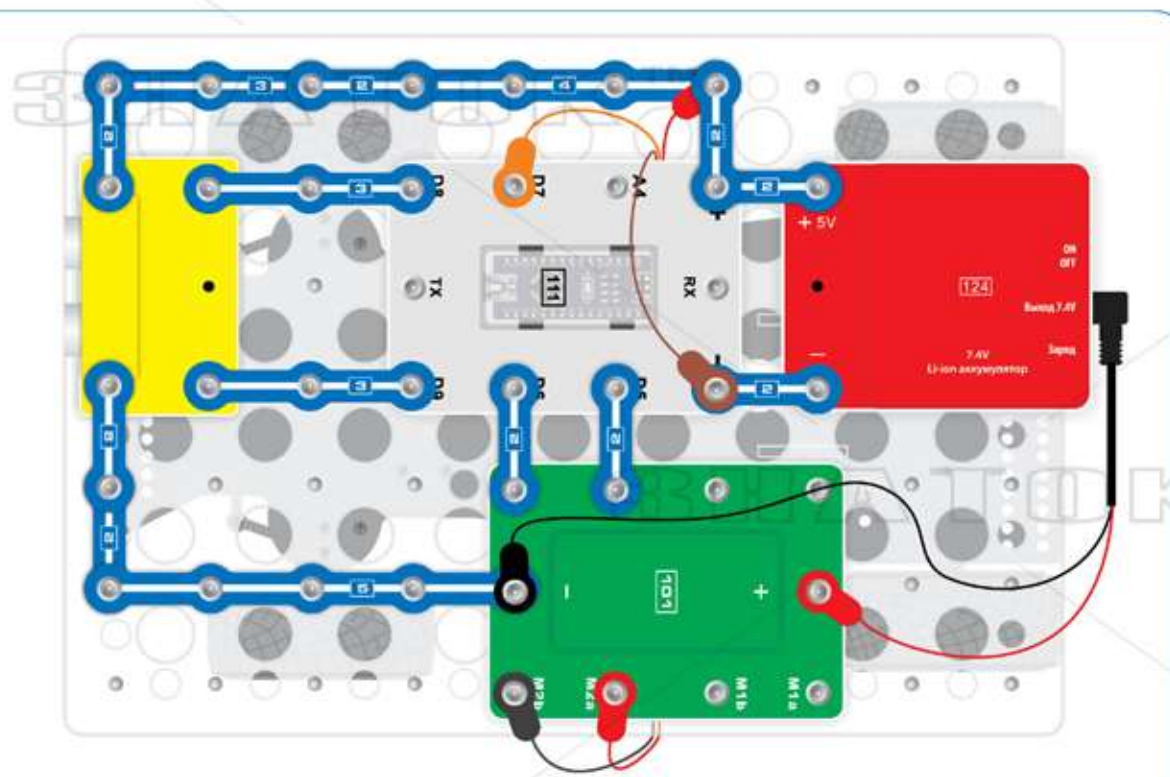
БЕЗОПАСНАЯ ЕЗДА



Это видео доступно и на сайте www.znatok.ru в разделе Конструкторы/Электронные конструкторы ЗНАТОК/Arduino MOBILE

Соберите схему, как показано на рисунке. К этой схеме прилагается 1 программа: **04_Mobile-05** — Соблюдение дистанции.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-05**. После загрузки программы (по завершении загрузки перестанут моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

04_Mobile-05

После включения питания, электромобиль начнет движение: если расстояние до находящегося впереди препятствия более 30 см машина будет ехать

вперед, если равно 30 ± 1 см — машина остановится, если меньше 30 см, то поедет назад, пока не отъедет на безопасное расстояние.

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!

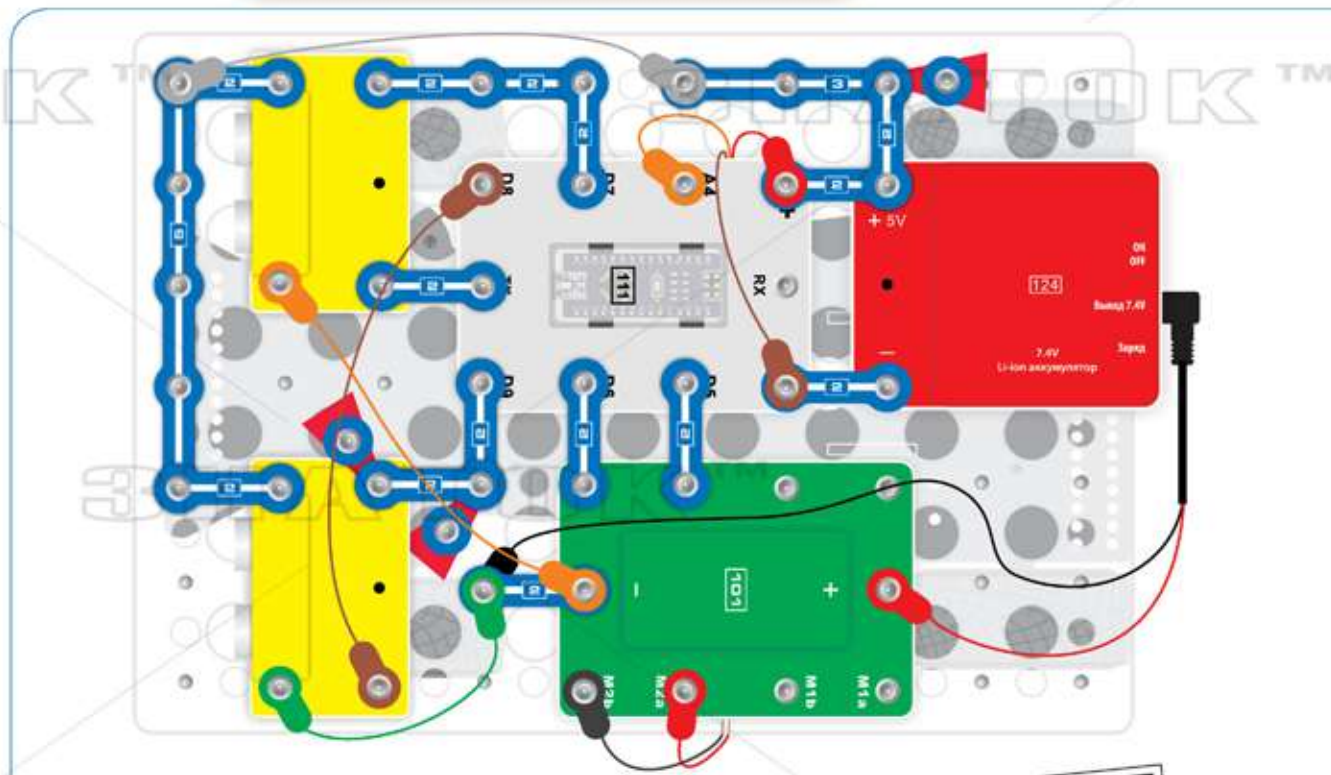


Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

ПАРКОВКА В ГАРАЖ

Соберите схему, как показано на рисунке. К этой схеме прилагается **1** программа:
04_Mobile-06 — Парковка в гараж.

Временно удалите провод **2**, идущий к выводу TX модуля **111**. Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-06**. После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните провод **2** на место и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

04_Mobile-06

Данная программа позволяет имитировать парковку электромобиля в гараж: после включения, начинается

плавное движение вперед. При этом, если по пути, с одной стороны, встретится какое-либо препятствие, например, боковая стенка гаража, то машина попытается скорректировать свое движение, чтобы не врезаться. Парковка заканчивается, когда электромобиль видит обоими своими сенсорами препятствие – заднюю стенку гаража.



Если проблема.
Посмотри здесь



Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!



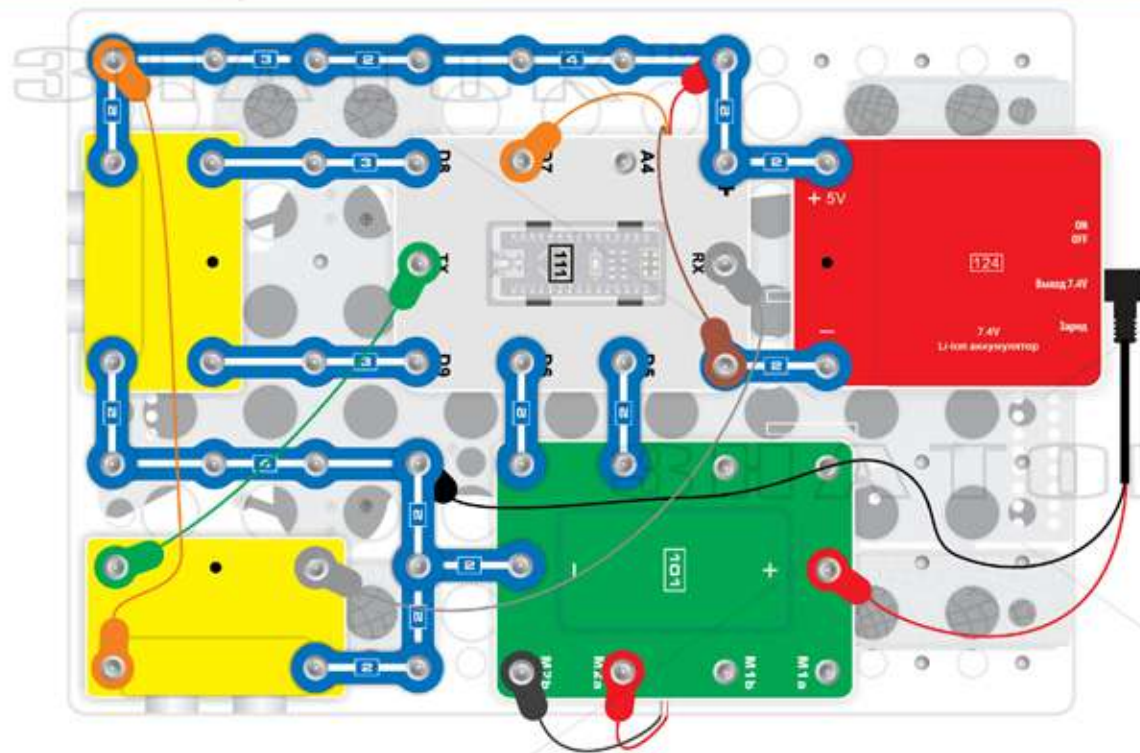
Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

ЛАБИРИНТ

ЗНАТОК™

Соберите схему, как показано на рисунке.
К этой схеме прилагается 1 программа:
04_Mobile-07 — Лабиринт.

Подключите модуль **111** при помощи USB-кабеля к компьютеру и загрузите программу **04_Mobile-07**. Если для подключения мешает зелёный провод, его можно временно отключить. После загрузки программы (по завершении загрузки перестанут моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, подключите зелёный провод к выводу TX и, когда будете готовы, включите питание — выключатель на аккумуляторе **124** в положение ON.



П Р О Г Р А М М А

04_Mobile-07

Данная программа позволяет имитировать поиск выхода из лабиринта, придерживаясь давно известного правила «левой

руки»*. Увидеть этот эффект можно разместив какой-то большой предмет, например, чемодан или коробку посередине комнаты – электромобиль будет все время ехать вокруг него. Если на пути следования поместить препятствие, то электромобиль попытается объехать его справа, а затем продолжит движение.

*ПРАВИЛО ЛЕВОЙ РУКИ

Считается, что выйти из любого лабиринта можно, если сразу при заходе в него все время будете касаться стены левой рукой.

Аналогично можно использовать правило «правой руки».



У нас это
получилось.



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

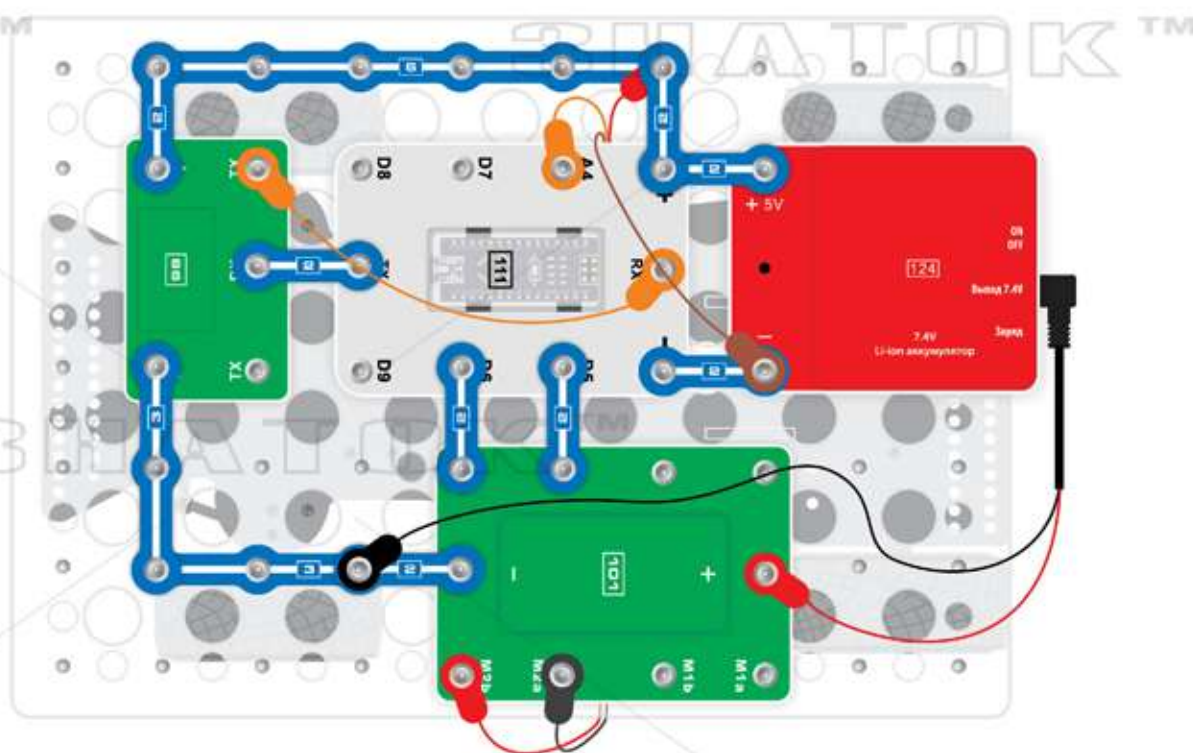
ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ ЭЛЕКТРОМОБИЛЕМ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **04_Mobile-08** — «Дистанционное управление электромобилем». После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

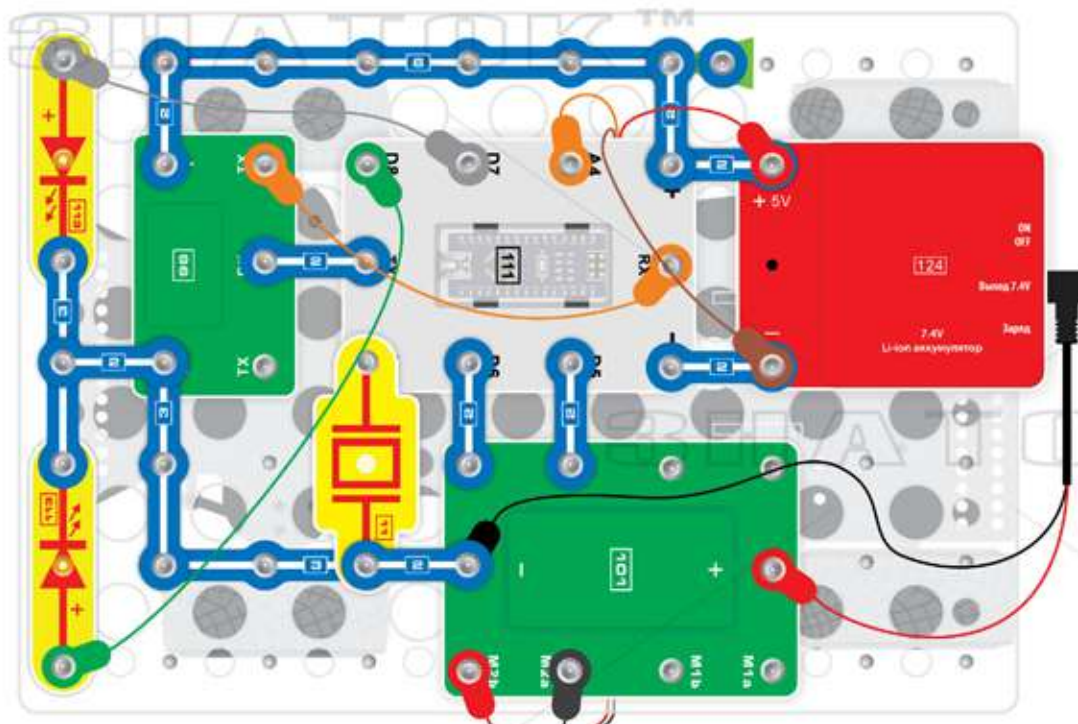


Откройте на вашем мобильном устройстве приложение **Znatok.Mobile** и выберите в меню «Электромобиль» один из вариантов управления. Открывшийся пульт управления позволит дистанционно управлять электромобилем: выбрать скорость и направление движение.

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!



ЭЛЕКТРОМОБИЛЬ С ПОВОРОТНИКАМИ И ЗВУКОВЫМ СИГНАЛОМ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **04 Mobile-09** — «Электромобиль с поворотниками и звуковым сигналом». После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile** и выберите в меню «Электромобиль» один из вариантов управления. Открывшийся пульт управления позволит дистанционно управлять электромобилем: выбирать скорость и направление движение, включать поворотники, а так же включать спецсигнал или клаксон.

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!



Это видео доступно
и на сайте www.znatok.ru
в разделе Конструкторы/
Электронные конструкторы ЗНАТОК/
Arduino MOBILE

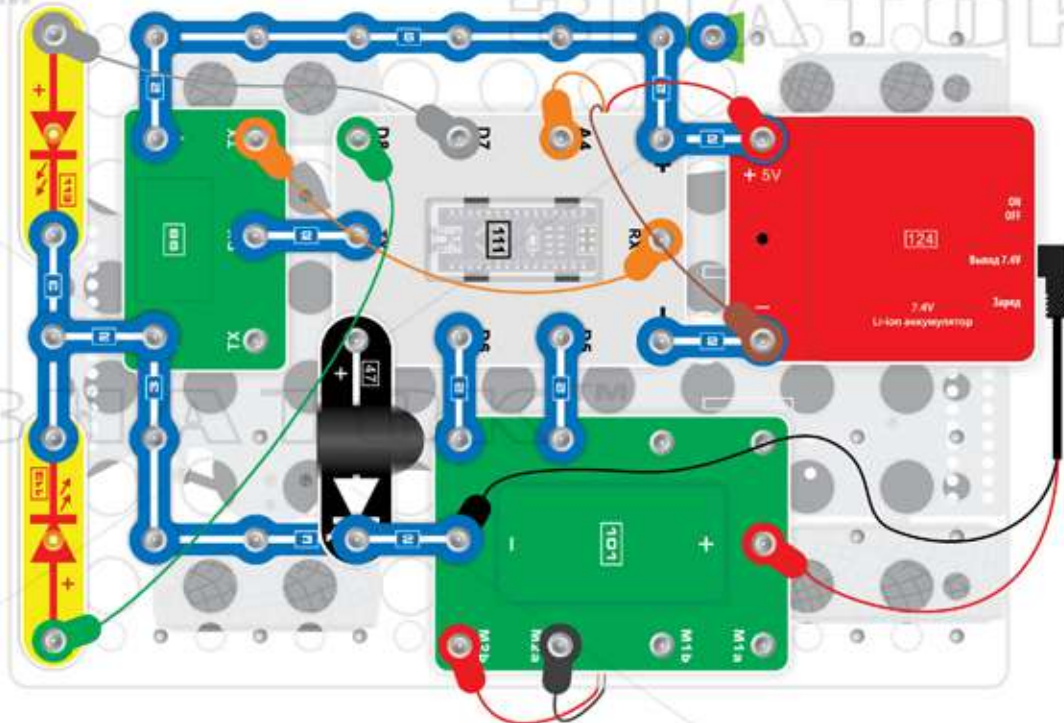
ЭЛЕКТРОМОБИЛЬ С ПОВОРОТНИКАМИ И ФАРОЙ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **04_Mobile-10** — «Электромобиль с поворотниками и фарой». После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.



Откройте на вашем мобильном устройстве приложение **Znatok.Mobile** и выберите в меню «Электромобиль» один из вариантов управления. Открывшийся пульт управления позволит дистанционно управлять электромобилем: выбирать скорость и направление движение, включать поворотники, а так же включать и выключать фару.

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.



Осторожно! Если ваша платформа находится на столе, то после включения она может поехать прямо и упасть с него!

МОБИЛЬНЫЙ ИЗМЕРИТЕЛЬ РАССТОЯНИЯ

Выключите аккумулятор **124** (OFF). Соберите схему, как показано на рисунке. Подключите модуль **111** при помощи USB-кабеля к компьютеру, для этого вам понадобится снять провод **2**, идущий к выводу TX. Загрузите программу **04_Mobile-11** — «Мобильный измеритель расстояния». После загрузки программы (по завершении загрузки перестают моргать светодиоды на модуле **111**, и внизу экрана компьютера в строке состояния появится «Загрузка завершена») отсоедините USB-кабель, верните на место провод **2** и включите питание — выключатель на аккумуляторе **124** в положение ON.

Откройте на вашем мобильном устройстве приложение **Znatok.Mobile** и выберите в меню «Электромобиль» один из вариантов управления. Открывшийся пульт управления позволит дистанционно управлять электромобилем: выбирать скорость и направление движения. Кроме того, на экране мобильного устройства будет отображаться информация, полученная с датчика, расположенного в передней части платформы.

Подробную информацию о работе приложения **Znatok.Mobile** с разными схемами можно найти на сайте http://znatok.ru/link/?mobile_app.

После каждого изменения в программе, сохраняйте обновленную версию с новым именем в библиотеке ваших проектов. Не забывайте загружать обновленную программу в модуль **111** (Arduino)!



ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ C++

Для выполнения проектов информация, приведенная в этом разделе, может и не потребоваться, но для общего понимания она будет полезна.

Структура программы

Язык Arduino — это язык C/C++ со специальными функциями-командами, предназначенными для работы с микроконтроллерами.

- Программа для Arduino состоит минимум из двух обязательных блоков-функций: **setup** и **loop**. Код, находящийся в блоке **setup()** выполняется один раз при запуске микроконтроллера, код внутри блока **loop()** выполняется циклически вплоть до отключения питания микроконтроллера.
- Все команды выполняются последовательно!
- На выполнение любой команды (включение светодиода, снятие показания с датчика) тратится определенное время, иногда достаточно большое (сотни миллисекунды).
- Каждую команду надо писать с новой строки и в конце ставить символ ; (точка с запятой).
- Любой блок выражений или функций должен находиться внутри фигурных скобок { }. Отсутствие скобок — одна из самых частых ошибок начинающего программиста.

Комментарии в программе

Даже опытным программистам, часто, сложно понять, что происходит, когда выполняется та или иная программа — для этого, в коде программы можно оставлять комментарии. Комментарии — это записи, предназначенные не для выполнения микроконтроллером, а для удобства чтения самой программы человеком. Комментарии могут быть двух видов: однострочные и многострочные.

Если требуется оставить какое-либо длинное пояснение, то это можно сделать при помощи двух специальных последовательностей символов: **/*(начало комментария)** и ***/(конец комментария)**, поместив любой текст между ними.

Если комментарий не занимает много места и влезает на одну строку, то отделить текст программы от комментария можно при помощи двух символов **//** — весь текст после этих символов будет воспринят как комментарий. В среде программирования Arduino комментарии выделяются зелёным цветом.

Часто бывает, что для пробного запуска необходимо убрать одну-две строки из программы, которые потом вернуться — такие строки можно «закомментировать» — они не будут восприняты микроконтроллером как команды для выполнения, а когда их нужно будет вернуть — символы комментариев можно будет убрать.

Все вышесказанное можно увидеть на примере данной программы:

```
void setup() // Функция установки — команды внутри выполняются один раз
{
    pinMode(5, OUTPUT); // Настройка цифрового вывода D5 на выход
}
void loop() // Основной цикл — команды выполняются до выключения
{
    digitalWrite(5, HIGH); // Выставить «высокий» уровень на D5 — включить светодиод
    delay(100); // Ничего не делать 100 мс
    digitalWrite(5, LOW); // Выставить «низкий» уровень на D5 — выключить светодиод
    delay(900); // Ничего не делать 900 мс
}
```


Что такое функция

Функция — это блок кода, выполняющий определенные заданные действия, и имеющий собственное имя. Существуют стандартные функции, выполняющие определенные действия — «подать напряжение на вывод», «считать значение с АЦП». Кроме того блоки кода `loop()` и `setup()` так же являются функциями. Существуют целые наборы функций, называемые библиотеками, упрощающие работу с различными устройствами.

Любая функция в программе либо **выполняет определенное действие**, либо вычисляет какое то значение и возвращает его в программу для дальнейшей работы с ним, их называют **функции, возвращающие значение**.

Процедура (Функции действия). Например, если написать `digitalWrite(9, HIGH)`; — будет вызвана данная функция — на цифровой вывод D9 модуля 111 будет подан высокий уровень напряжения (HIGH — 5 вольт) и светодиод загорится. Если написать `digitalWrite(9, LOW)`; то на вывод D9 будет подан низкий уровень напряжения (0В) и светодиод погаснет.

Где `digitalWrite` — это имя стандартной функции. Различные числа и слова в скобках после имени функции — это **входные параметры функции**. Если мы будем их менять, то функция будет работать по-другому. Еще пример, заменив 9 на 8, мы будем подавать уровни HIGH и LOW уже не на вывод D9, а на вывод D8.

Функции, возвращающие значение. Например, в модуле Arduino есть аналого-цифровой преобразователь (АЦП), который преобразует входное напряжение от 0 до 5 вольт, поступающее на вывод A4, в число от 0 до 1023. Если стоит задача считать число на выходе АЦП и сохранить его, то можно написать:

```
int a = analogRead(A4);
```

Где `analogRead` — имя стандартной функции, которая принимает один входной параметр, в нашем случае вывод A4, и возвращает значение — число от 0 до 1023 в программу, для принятия дальнейших решений. В приведенном примере, если на вывод A4 подавать напряжение 5В, то после выполнения функции, указанная строчка превратится в

```
int a = 1023;
```

Где `1023` — это значение, которое функция «вернула» в процессе выполнения.

Программист может написать собственную функцию, а не только использовать стандартные. Например, для написания функция вычисления среднего значения двух чисел a и b, необходимо создать конструкцию такого вида:

Тип возвращаемого значения — тип переменной, которая будет возвращена как ответ (см. Переменные). Если функция ничего не возвращает, то стоит писать void (как это делается в случае с функциями loop() и setup())

Имя функции — имя, придуманное программистом для собственной функции

```
float mean(float a, float b)
{
    float sum = a + b;
    return sum / 2;
}
```

Входные параметры — параметры, необходимые для работы функций

Всё, что находится между фигурными скобками, называется **телом функции**.

Число, идущее после служебного слова `return` — возвращаемое значение.

Переменные

Кроме выполнения операций по управлению своими выводами, микроконтроллер Arduino может сохранять и обрабатывать информацию. Самым простым способом сохранения информации для дальнейшего использования является использование **переменных**. Переменная — это область в памяти микроконтроллера, у которой есть собственное имя.

```
int a;
```

```
a = analogRead(A4);
```

В данном примере внутри программы создается переменная с именем «a» (выделяется память внутри микроконтроллера и этому кусочку присваивается имя), после чего в эту ячейку-переменную «a» записывается число, полученное с выхода АЦП. Имя переменной придумывает сам программист и оно может быть практически любым — хорошим тоном считается именование переменных по назначению их использования. Так по имени переменной «distanceSensorValue» становится понятно, что внутри нее хранится значение, полученное с датчика расстояния.

Из-за того, что для хранения любых значений, в т.ч. чисел, в компьютерах используется двоичное кодирование, то программистам необходимо явно указывать с каким **типом переменных** ведется работа — одна и та же последовательность бит (1000101...) в зависимости от **типа** может являться чем угодно — целым числом, дробным числом или символом.

Ниже приведен список основных типов переменных, используемых в языке Arduino, а так же объем памяти в битах, занимаемый каждой переменной.

boolean (8 бит) — логическая переменная — принимает значение true/false

byte (8 бит) — unsigned (беззнаковое) число в диапазоне от 0 до 255

char (8 бит) — signed (знаковое — может быть отрицательным и положительным) число в диапазоне от -128 до 127. Чаще всего, микроконтроллер воспринимает переменные данного типа не как числа, а как символы (например число 64 — это символ «@»)

unsigned char (8 бит) — то же самое, что и byte

word (16 бит) — unsigned число в диапазоне от 0 до 65535

unsigned int (16 бит) — то же самое, что и «word»

int (16 бит) — signed число в диапазоне от -32768 до 32767

unsigned long (32 бита) — unsigned число в диапазоне от 0–4,294,967,295. Используется в случаях, когда диапазона int может не хватать — например при использовании встроенной функции millis() (см.ниже)

long (32 бита) — signed число в диапазоне от -2,147,483,648 до 2,147,483,647

float (32 бита) — signed число в диапазоне от -3.4028235E38 до 3.4028235E38

Любая переменная имеет границы видимости — область программы, внутри которой переменная может «работать». Границы видимости любой переменной — это фигурные скобки, внутри которых эта переменная была создана. Если нужно создать переменную, которую «видно» отовсюду — её можно создать перед блоком setup() — такие переменные называются глобальными.

Арифметические операции

С переменными можно выполнять математические операции:

Сложение: `int a = 3+2;` Вычитание: `int b = a - 3;`

Умножение: `int c = a*b;` Деление: `int d = c / a;`

`int e = c % a;`

Обратите внимание, что для целых чисел (типа int) существует 2 операции для деления: целая часть от деления (/) и остаток от деления (%). Это связано с тем, что в переменной типа **int** могут храниться только целые значения, и если нам нужно получить дробное число, то мы должны использовать тип **float**:

`float f = 2.5/1.2;`

По умолчанию, переменные воспринимаются микроконтроллером как переменные того типа, которым они объявлены, поэтому выражение с созданными выше переменными:

`f = a/b;`

будет воспринято, как попытка записать в переменную f целую часть от деления a на b, и если мы захотим разделить эти числа «по настоящему», то мы должны сказать воспринимать переменную a, как переменную не целую, а дробную и правильно строчка должна выглядеть как:

`f = (float)a/b;`

Данная операция называется приведением типов.

Кроме указанных выше способов написания арифметических операций, существуют так же краткие способы записи — смешанное присваивание:

`i++; // увеличение i на единицу. Тоже самое что i = i+1`

`i--; // уменьшение i на единицу. Тоже самое что i = i-1`

`i += x; // Тоже самое что i = i + x;`

`i -= x; // Тоже самое что i = i - x;`

`i *= x; // Тоже самое что i = i * x;`

`i /= x; // Тоже самое что i = i / x;`

Логические выражения

Когда необходимо сравнить значение одной переменной со значением другой используются операторы сравнения:

```
a == b // a равняется b
a != b // a не равняется b
a > b // a больше b
a < b // a меньше b
a >= b // a больше или равно b
a <= b // a меньше или равно b
```

Данные операторы могут быть скомбинированы при помощи трех логических операторов: **И**, **ИЛИ**, **НЕ**. Данные конструкции могут быть использованы, например, при написании условий (if):

```
if (a > 0 && a < b) // a больше нуля И a меньше b
if (a == 1 || a == 2) // a равно 1 ИЛИ a равно 2
if (!a == 0) // НЕ (a равно нулю) — тоже самое, что и a не равно нулю
```

Конструкции управления программой

Иногда в программах необходимо выполнять те или иные команды в зависимости от каких либо **условий**: например включить вентилятор, если температура выше 25 градусов. Это можно реализовать при помощи конструкции if..else

```
if (curTemperature > 25)
{
digitalWrite(6, HIGH);
}
else
{
digitalWrite(6, LOW);
}
```

В данном примере сравнивается значение переменной curTemperature с величиной 25 и если переменная больше 25, то на цифровой вывод D6 подается ВЫСОКИЙ уровень напряжения, иначе подается НИЗКИЙ.

Когда требуется повторить некоторое действие несколько раз или осуществлять это действие до наступления какого-либо условия используются **циклы**.

```
int i = 0;
while (i < 10)
{
digitalWrite(9, HIGH);
delay(500);
digitalWrite(9, LOW);
delay(500);
i++;
}
```

В данном примере создается переменная i, она приравняется к нулю и «пока значение переменной i меньше 10», выполняется тело цикла — то, что находится внутри фигурных скобок, а именно — включение и выключение устройства, подключенного к цифровому выводу D9, например, светодиода, который будет мигать. В конце каждого цикла происходит увеличение значения переменной i на единицу. Таким образом, данное действие выполнится 10 раз, до момента, когда i станет равным 10 и условие не будет истинным — светодиод погаснет.

Для подобных циклов, можно использовать более компактную запись **for**

```
for (int i = 0; i < 10; i++)
{
digitalWrite(9, HIGH);
delay(500);
digitalWrite(9, LOW);
delay(500);
}
```

Стандартные функции, используемые в проектах

Функции работы с цифровыми и аналоговыми выводами

pinMode(pin, mode)

Устанавливает режим работы вывода Arduino

- **pin** — номер вывода (например, 6 или A4)
- **mode** — настраиваемый режим
 - *INPUT* — вход, для дальнейшего считывания информации с вывода
 - *OUTPUT* — вывод
 - *INPUT_PULLUP* — вход с встроенным подтягивающим резистором

digitalRead(pin)

Считывает значение с вывода pin и возвращает результат HIGH (высокий уровень) или LOW (низкий уровень)

digitalWrite(pin, value)

Выставляет на вывод pin уровень напряжения value: высокий (HIGH) или низкий (LOW).

analogRead(pin)

Считывает значение при помощи АЦП с вывода pin – полученное значение лежит в пределах от 0 (0В) до 1023 (5В)

analogWrite(pin, value)

Выдает аналоговую величину (ШИМ) на вывод pin. Ширина положительных импульсов задается значением value, лежащим между 0 и 255.

Функции работы со временем

millis()

Возвращает количество миллисекунд, прошедших с момента запуска модуля Arduino. Тип возвращаемого значения — unsigned long, поэтому данное значение «переполнится» через 50 дней и для дальнейшей корректной работы потребуется перезагрузка микроконтроллера.

micros()

Возвращает количество микросекунд, прошедших с момента запуска модуля Arduino. Тип возвращаемого значения — unsigned long, поэтому данное значение «переполнится» через 70 минут и для дальнейшей корректной работы потребуется перезагрузка микроконтроллера.

delay(ms)

Останавливает выполнение программы на ms миллисекунд.

delayMicroseconds(us)

Останавливает выполнение программы на us микросекунд.

Функции работы со звуком

tone(pin, frequency)

tone(pin, frequency, duration)

Генерирует на выводе pin прямоугольный сигнал заданной частоты frequency в течение duration миллисекунд (если указано). Для воспроизведения звука, к выводу pin следует подключить динамик.

noTone(pin)

Останавливает генерацию прямоугольного сигнала на выводе pin.

Математические функции

map(value, fromLow, fromHigh, toLow, toHigh)

Преобразовывает значение переменной из одного диапазона в другой. Если переменная value будет равняться fromLow, то функция вернет значение toLow, если переменная value будет равняться fromHigh, то функция вернет значение toHigh, все промежуточные значения будут отмасштабированы относительно нового диапазона.

abs(x)

Возвращает модуль числа x

min(x, y)

Возвращает наименьшее из двух чисел x и y

max(x, y)

Возвращает наибольшее из двух чисел x и y

random(max)

random(min, max)

Возвращает псевдо-случайно целое число в диапазоне от 0 (min, если указано) до max-1