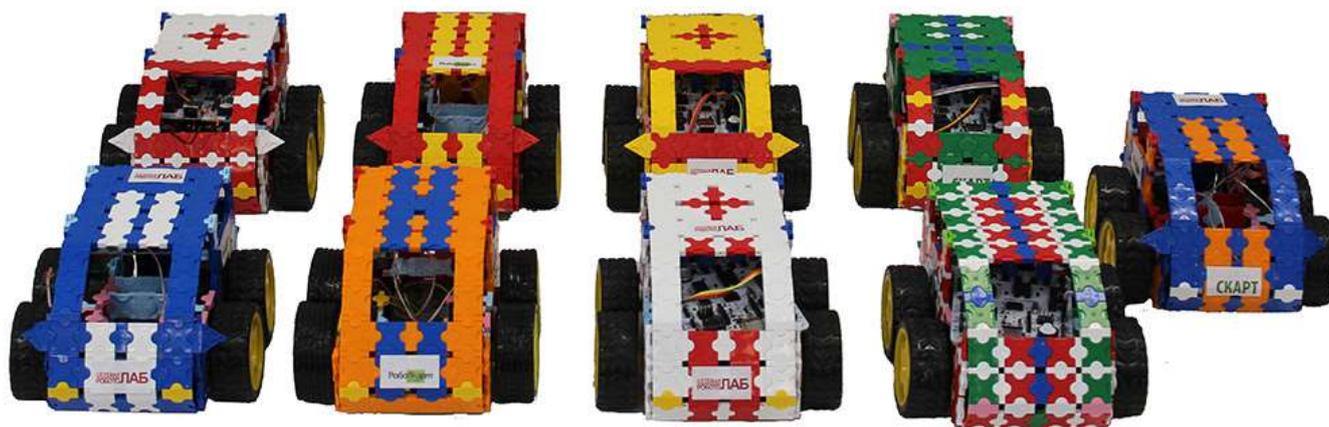


# Универсальные роботы СКАРТ



## Инструкция по эксплуатации

г. Москва  
2016



# 1. Состав конструктора

## Конструкторы и наборы для средней школы

<p>Конструктор СКАРТ- 1</p>	<p>Набор предназначен для обучения сборке и программированию робота-машинки.</p> <p>В комплекте есть цифровые датчики, которые позволят сделать робота, едущего по линии или проходящего лабиринт. Благодаря входящему в комплект модулю Bluetooth, для собранного робота доступно управление с помощью смартфона или планшета.</p>	 The image shows the contents of the SkART-1 kit laid out on a white surface. At the top is a red and white carrying case with the 'СКАРТ' logo and a picture of a yellow robot. Below the case are two large black wheels with yellow hubs, a black motor, a blue USB cable, a black battery pack, a breadboard with electronic components, and several small plastic bags containing various parts like resistors and jumper wires. A multi-colored ribbon cable is also visible.
<p>Конструктор СКАРТ- 2</p>	<p>В дополнение к набору «СКАРТ-1» в наборе «СКАРТ-2», присутствуют датчики шума, температуры, влажности, освещенности, что позволяет проводить с помощью собранного робота различные исследования. Входящих в набор комплектующих достаточно для того что бы начать работу над собственным исследовательским проектом. Это может быть, например, мобильная исследовательская лаборатория. При этом, также, как и в наборе «СКАРТ-1», можно собрать робота для движения по линии или по лабиринту.</p>	 This image shows the contents of the SkART-2 kit. It includes all the components from the SkART-1 kit, plus additional sensors: a blue sound sensor module, a yellow temperature sensor module, a green humidity sensor module, and a red light sensor module. There are also additional jumper wires and small components in plastic bags.
<p>Конструктор СКАРТ- 3</p>	<p>По сравнению с набором «СКАРТ-2», в наборе «СКАРТ-3» увеличено количество датчиков, в комплект добавлены 2-х строчный ЖК-экран, макетная плата, комплекты радиодеталей и мультиметр. Такой конструктор позволит не только построить робота-машину с различным функционалом или «умный дом», но и начать полноценно изучать электронику, создавая проекты на основе макетной платы и проводя необходимые измерения.</p>	 The image displays the contents of the SkART-3 kit. In addition to the sensors from the previous kits, it includes a 2-line LCD display, a breadboard, a multimeter, and various electronic components like resistors and integrated circuits. There are also more jumper wires and a small black module.

Сравнительная таблица комплектаций конструкторов серии «СкАРТ»

№ п/п	Наименование	«СкАРТ-1»	«СкАРТ-2»	«СкАРТ-3»
1	Набор "Роботолаб-конструирование" 500 деталей	1	1	1
2	Плата "АМПЕРКА Стрела"	1	1	1
3	Моторы	4	4	4
4	Колесо	4	4	4
5	Модуль Bluetooth для Arduino	1	1	1
6	Батарейный блок	1	1	1
7	Шлейф из проводов «мама-мама»	1	1	1
8	Шлейф из проводов «папа-папа»	1	1	1
9	Выключатель	1	1	1
10	Шаровая опора	1	1	1
11	Датчик шума (Тройка-модуль)		1	1
12	Барометр (Тройка-модуль)			1
13	Светодиод «Пиранья» (Тройка-модуль)	1	1	1
14	Цифровой датчик температуры и влажности (Тройка-модуль)		1	1
15	Датчик освещённости (Тройка-модуль)		1	1
16	Ультразвуковой дальномер	1	1	1
17	Датчик линии	2	2	2
18	Датчик влажности почвы			1
19	Кнопка			1
20	Потенциометр			1
21	Датчик IMU на 10 степеней свободы			1
22	ЖК-экран двухстрочный			1
23	Мультиметр			1
24	Макетная плата			1
25	Кабель USB A-B	1	1	1
26	Набор для схемотехники			1
30	Винт М3	4	4	5
31	Гайка М3	4	4	5
32	Пластиковая коробка для хранения	1	1	1

**\*Комплектация набора может быть незначительно изменена производителем без предупреждения.**

## 2. С чего начать

Конструктор предназначен для самостоятельной сборки роботов-машинок из входящих в набор деталей.

Конструктив создается на основе фантазии учащихся.

Моторы, входящие в комплект, могут быть закреплены в конструкторе без применения каких-либо дополнительных деталей. Рабочее напряжение моторов 3-8В.

Колёса легко надеваются на ось мотора.

Соединения не требуют пайки и могут быть осуществлены при помощи входящих в состав набора проводов с разъёмами. Дополнительно, для присоединения питания и моторов к плате контроллера, потребуется шлицевая отвертка (не входит в комплект).

Количество входящих в набор деталей конструктора позволяет собрать две разные машинки, оснащенные двумя колесами и шаровой опорой, либо одну машину с четырьмя колесами с полным приводом (левые и правые моторы в этом случае подключаются парами).

В этом руководстве мы подробно рассмотрим создание робота-машинки с управлением с помощью смартфона или планшета на OS Android через интерфейс BlueTooth.

**Примеры сборки машинок:**

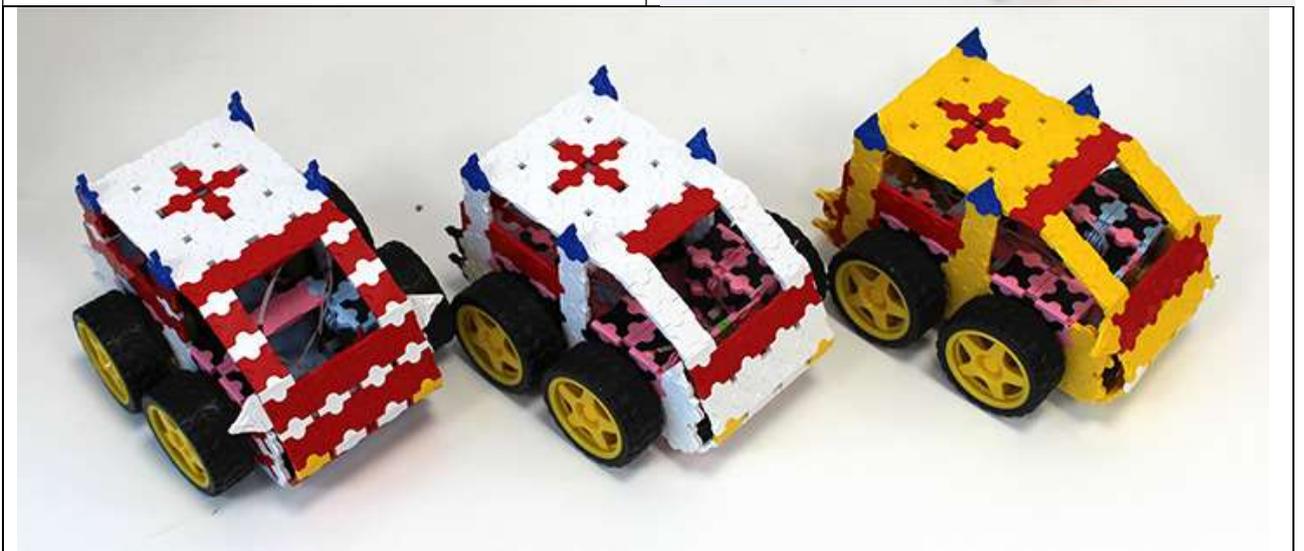
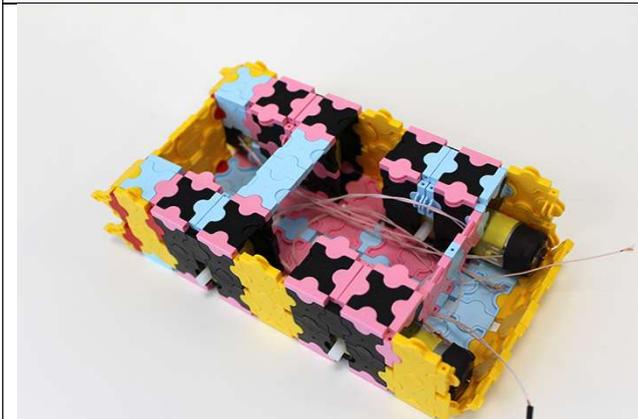
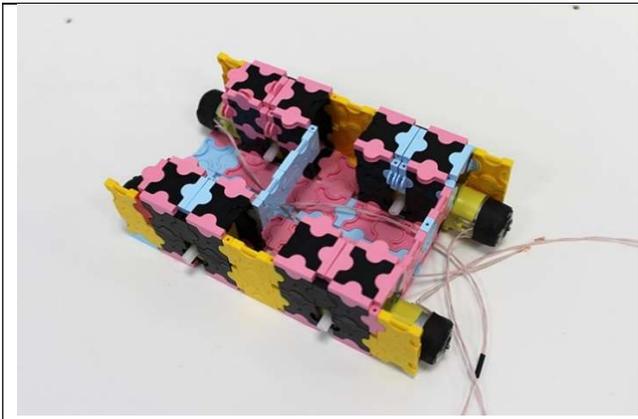






Одна из возможных схем сборки четырехколесного автомобиля



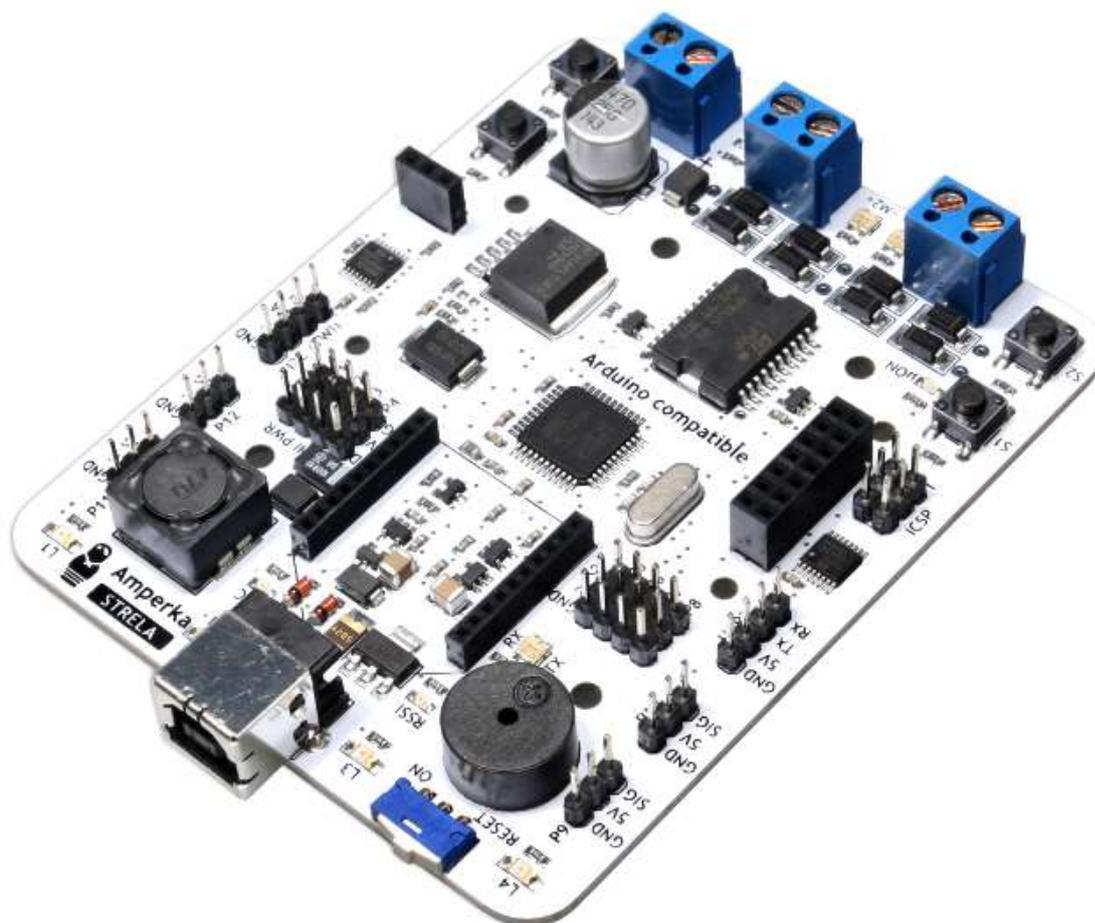


### 3. Контроллер «СТРЕЛА»

Познакомимся с сердцем робота – управляющим контроллером. Arduino-совместимый контроллер «STRELA» разработан и производится компанией «АМПЕРКА» как материнская плата для постройки роботов. Одна эта плата призвана заменить собой Arduino с несколькими шилдами, которые обычно применяются в роботах. А, благодаря библиотеке Strela (<https://github.com/amperka/Strela>), работа с платой не вызовет затруднений даже у начинающих роботостроителей.

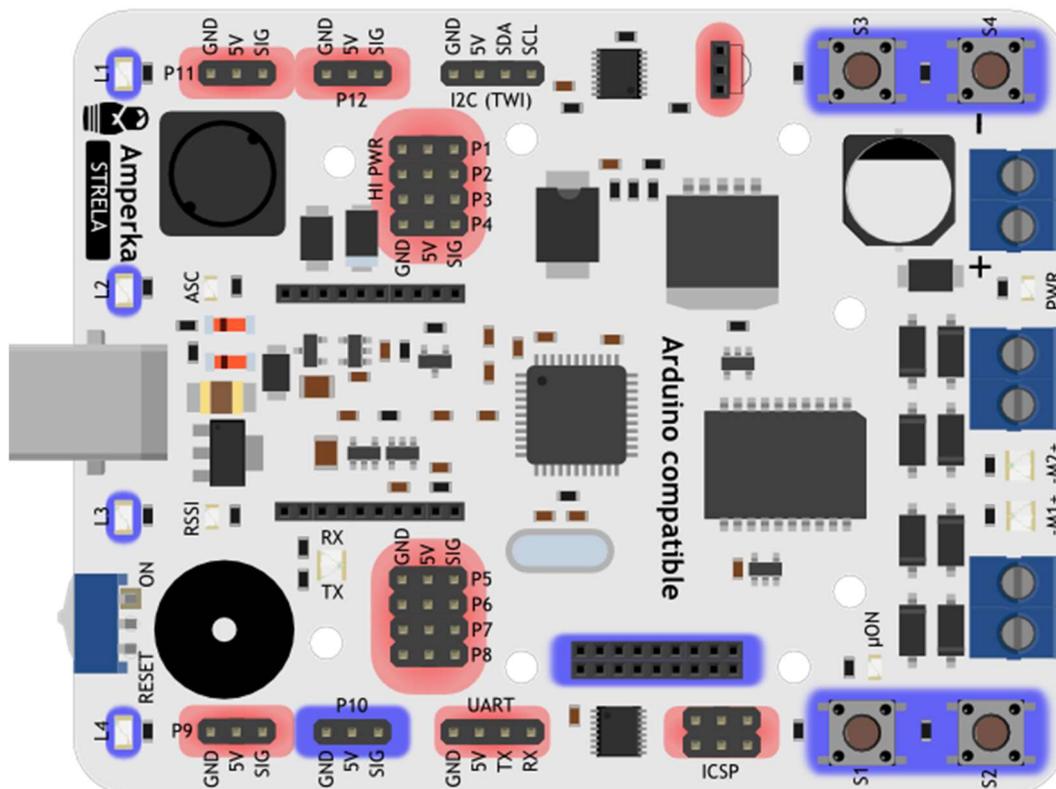
«Стрела» отлично подходит для соревнований. Вы можете сделать на её основе колёсного робота, который отлично себя покажет при езде по линии, в сумо, в лабиринте и во многих других категориях. Эта плата может пригодится не только для строительства роботов. Ничто не мешает использовать её там, где необходимо управление двигателями или сервоприводами. Например, она может управлять шторами на окнах, в зависимости от освещённости на улице, или управлять мотор-колесом на электросамокате.

«Мозг» платы — микроконтроллер ATmega32u4. Это тот же самый микроконтроллер, что установлен в Arduino Leonardo. Этот контроллер используется и в качестве USB-UART преобразователя для прошивки. Смело выбирайте модель Leonardo при работе с платой через Arduino IDE.



# Arduino GPIO

# I<sup>2</sup>C GPIO



Платформа Strela имеет на борту контакты, подключённые к Arduino-микроконтроллеру, и контакты, подключённые к I<sup>2</sup>C-расширителю портов ввода-вывода. На аппаратном уровне доступ к этим портам ввода-вывода очень сильно различается, но вы можете не переживать по этому поводу. Для облегчения работы с платой из Arduino IDE была создана библиотека [Strela](#). Все примеры будут с этой библиотекой.

Плата программируется в [Arduino IDE](#). Для начала работы с платой достаточно выбрать в меню *Инструменты* → *Плата* → *Arduino Leonardo* и соответствующий COM-порт.

## Элементы платы

Теперь рассмотрим подробнее все компоненты и разъёмы, которые можно встретить на плате с точки зрения схемотехники.

### Микроконтроллер ATmega32u4



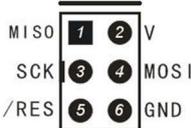
Strela	Arduino Leonardo	Доступные функции
<b>P7</b>	A1	Цифровой вход/выход, аналоговый вход.
<b>P8</b>	A0	Цифровой вход/выход, аналоговый вход.
<b>P9</b>	8, A8	Цифровой вход/выход, аналоговый вход.
<b>P11</b>	A4	Цифровой вход/выход, аналоговый вход.
<b>P12 (IR)</b>	A5	Цифровой вход/выход, аналоговый вход. Strela использует этот же контакт как <b>вход для ИК-приёмника</b> . Управление с бытового ИК-пульта управления возможно с помощью библиотеки <a href="#">IRremote</a> .

### Контакты специального назначения

Некоторые стандартные контакты имеют специальное назначение, поскольку именно они управляют установленным на Strela оборудованием или отвечают за коммуникацию.

Контакты Arduino Leonardo	Конфигурация	Функция
12	OUTPUT	Пищалка — встроенный <a href="#">пьезодинамик</a> , который воспроизводит звук с помощью стандартной функции <a href="#">tone()</a> .
7	INPUT	Этот контакт подключён к I <sup>2</sup> C-расширителям портов, для обнаружения изменения напряжения на их входах. Этот контакт можно использовать для <a href="#">прерывания основной программы по изменению уровня</a> .
9	OUTPUT	Подключён к драйверу двигателей. Используется для задания скорости вращения мотора M1 при помощи <a href="#">ШИМ</a> .
10	OUTPUT	Подключён к драйверу двигателей. Используется для задания скорости вращения мотора M2 при помощи <a href="#">ШИМ</a> .
4	OUTPUT	Подключён к драйверу двигателей. Используется для задания направления вращения мотора M1.
~ (PE2)	OUTPUT	Подключён к драйверу двигателей. Используется для задания направления вращения мотора M2.

### Коммуникация

Порт связи	Контакты Arduino Leonardo	Функция	Использование
<b>Serial</b>	0, 1	RX, TX	Используется для связи по последовательному протоколу с модулем беспроводной связи, если он подключён, или с другими устройствами. Программно передача происходит с помощью <a href="#">Serial1</a> . Если в вашем проекте не используются устройства с последовательным протоколом, эти контакты могут быть использованы как обычные цифровые пины.
<b>SPI</b>	14, 15, 16	MISO, SCK, MOSI	<p>SPI-порт доступен на колодке ICSP.</p>  <p>Используется для связи по протоколу SPI. Передача данных происходит с помощью библиотеки <a href="#">SPI</a>. Если в вашем проекте не используются периферийные устройства с подключением по SPI, эти контакты могут быть использованы как обычные цифровые пины. Контактная колодка ICSP может использоваться для программирования платы с помощью <a href="#">программатора</a>.</p>
<b>I<sup>2</sup>C(TWI)</b>	2, 3	SDA, SCL	Используются для коммуникации по шине I <sup>2</sup> C. По этим пинам управляются встроенные I <sup>2</sup> C-расширители портов, поэтому использовать их в других целях не получится. К ним может быть подключено и другое оборудование, управляемое по I <sup>2</sup> C. Передача данных происходит с помощью библиотеки <a href="#">Wire</a> .
<b>USB</b>	USB-разъём тип В	Виртуальный Serial-порт	Используется для прошивки платы из Arduino IDE и связи по последовательному протоколу с компьютером. Программно передача происходит с помощью <a href="#">Serial</a> .

## Сброс

На плате присутствует переключатель, который бывает полезен при отладке или при подготовке к старту на соревнованиях. В положении «RESET» контакт Reset микроконтроллера замыкается на землю. В таком положении на микроконтроллер подано напряжение, но он не работает. Чтобы запустить микроконтроллер необходимо перевести рычажок в положение «ON».

## Силовая часть



### Питание

Клеммники под винт для подключения питания на плате обозначены как **PWR**. К ним подключается источник питания, который будет использоваться для питания моторов. Напряжение питания должно быть в пределах 7–24 В постоянного тока.



**Внимание!**

При подключении питания соблюдайте полярность. Неправильное подключение может привести к непредсказуемому поведению или выходу из строя платы или источника питания.

На плате три контура питания.

- Первый — *силовой*, напряжение на который приходит с клеммника PWR. От этого контура запитана микросхема **H-моста L298P**, моторы и **DC-DC-преобразователь LM2596-5.0**.



**Важно!**

Не все источники питания подходят для работы с моторами.

Для корректной работы DC-DC преобразователя источники питания должны быть способны обеспечить напряжение выше 7 В при резких скачках нагрузки или при включении моторов на полную мощность. Если источник питания не рассчитан на мощность, необходимую двигателям, просадка напряжения при росте нагрузки ниже 7 В может привести к перезагрузке управляющего контроллера и, связанным с этим, неадекватным поведением платы. Этому требованию соответствуют только **литий-ионные** и **никель-металлгидридные** аккумуляторы или блоки питания необходимой мощности.

Поэтому питать робота от обычной "Кроны" не стоит.

- Источником напряжения для второго контура питания может быть DC-DC преобразователь или USB. Это контур *управления*. Напряжение в этом контуре — 5 В. От него запитаны микроконтроллер, I<sup>2</sup>C-расширители, логика H-моста и преобразователь напряжения на 3,3 В для питания модулей беспроводной связи. Это же напряжение подано на все трёхштырьковые контакты. Напряжение в 5 В может быть подано как с USB, так и с клеммника PWR. При этом при подключении по USB будет работать только контур управления, напряжение на H-мосту будет недостаточным для работы двигателей. При одновременном подключении платы по USB и через колодки PWR питание будет подаваться только от PWR.
- Третий контур питания используется для питания модуля беспроводной связи и его обвязки через площадку для подключения модулей связи. Напряжение питания в этом контуре — 3,3 В.

#### Клеммники для подключения двигателей

Клеммники под винт для подключения нагрузки на плате обозначены как **-M1+ -M2+**.

Нагрузка разделена на 2 канала. Первый канал на плате имеет обозначение «M1», второй канал имеет обозначение «M2». Каждый канал управляется независимо.

Обозначения «+» и «-» показывают воображаемые начало и конец обмотки: если подключить два коллекторных двигателя таким образом, чтобы их одноимённые контакты щётчного узла соответствовали одному и тому же обозначению на плате, то, при подаче на H-мост L298P одинаковых управляющих импульсов, моторы будут вращаться в одну и ту же сторону.

#### Характеристики драйвера двигателей L298p

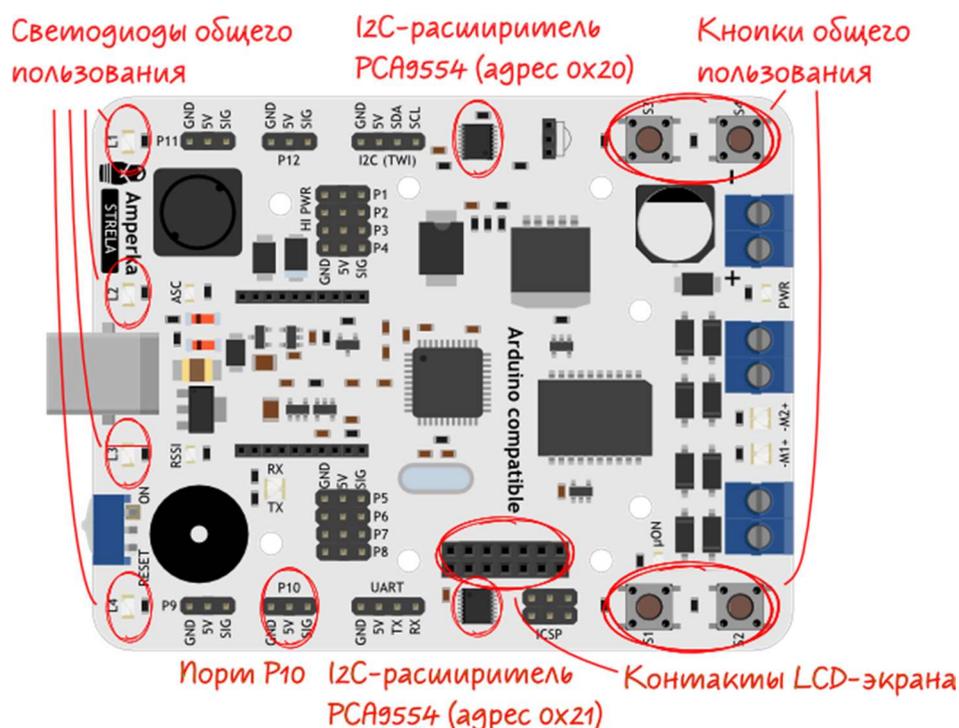
Параметр	Мин.	Макс.	Ед. изм.
Напряжение питания двигателей	7	24	В
Продолжительный ток на канал *	—	2	А
Пиковый прерывистый ток на канал (80% включение, до 10 мс)	—	2,5	А
Пиковый непрерывный ток на канал (до 100 мкс)	—	3	А

\* для достижения максимума необходимо дополнительное охлаждение чипа L298P

#### Характеристики DC-DC преобразователя LM2596-5.0

Параметр	Мин.	Номинал.	Макс.	Ед. изм.
Напряжение питания силовой части	7	—	24	В
Выходное напряжение	4,75	5	5,25	В
Ток нагрузки	—	—	3	А

## Расширители портов ввода-вывода



На плате установлены два I<sup>2</sup>C-расширителя портов ввода-вывода PCA9554.

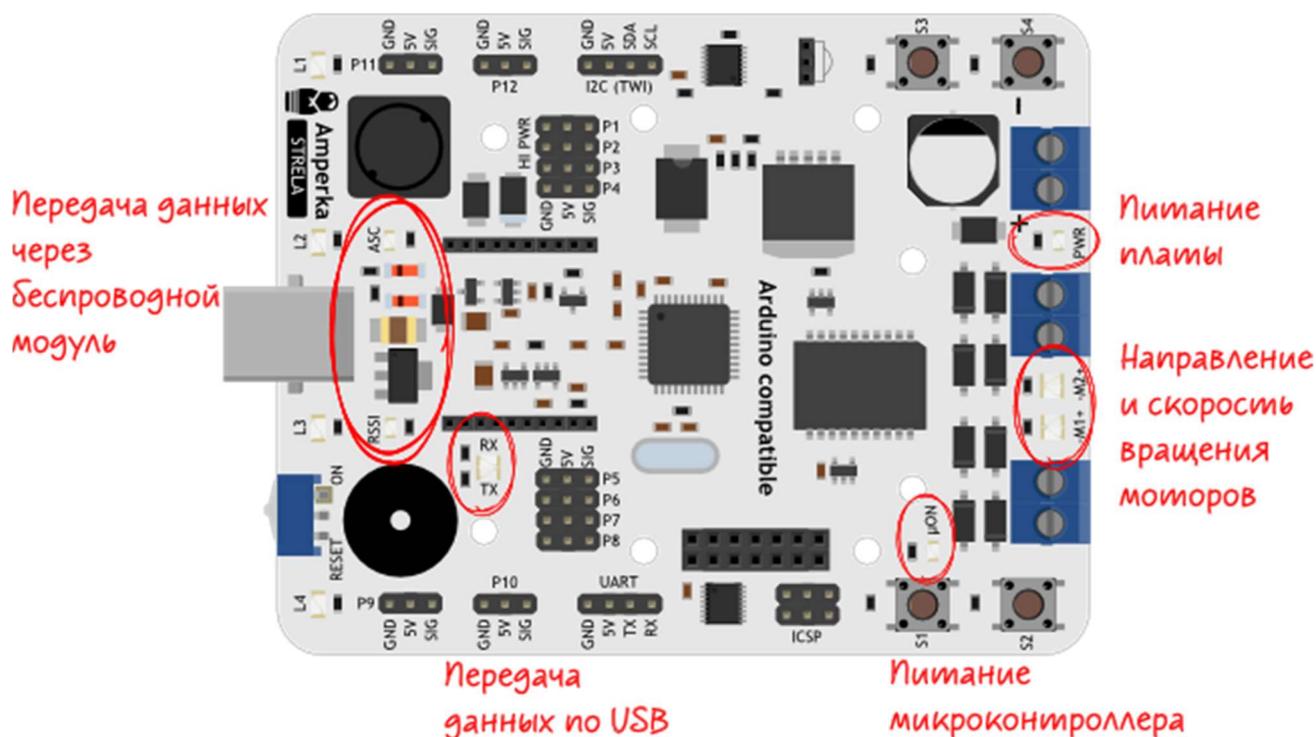
Адрес	Функция
0x20	Управление светодиодами L1-L4, получение информации о нажатии кнопок S1-S4.
0x21	Управление контактом P10 и LCD-экраном MT-08S2A при помощи библиотеки <a href="#">LiquidCrystal I2C</a> . Если LCD-экран не используется, можно использовать свободные контакты как дополнительные порты ввода-вывода.

## Характеристики I<sup>2</sup>C-расширителей портов ввода-вывода PCA9554

Параметр	Номинал.	Ед. изм.
Напряжение питания	5	В
Максимальный входной ток одного контакта	20	мА

Параметр	Номинал.	Ед. изм.
Максимальный выходной ток одного контакта	50	мА
Максимальный ток через контакты питания	85	мА

## Индикация



### Индикация питания платы

При правильном подключении питания через клеммник PWR, загорается светодиод индикации питания. Если полярность питания перепутана или питание по какой-то причине не подано, светодиод гореть не будет.

Из-за большой ёмкости фильтрующего конденсатора, установленного на плате, светодиод индикации питания в некоторых случаях может кратковременно продолжать гореть и после отключения питания. Этот светодиод может тускло гореть при питании платы от USB.

### Индикация питания микроконтроллера

Индикатор питания микроконтроллера горит, если на микроконтроллер ATmega32u4 подано напряжение.

### Индикация направления и скорости вращения двигателей

При высоком логическом уровне на пине управления направлением вращения, т.е. вращении вперёд, индикатор светится зелёным светом. При этом полярность напряжения на клеммниках для подключения двигателей соответствует обозначению «+» и «-» на плате. При низком уровне, т.е. при реверсе, светодиод светится красным цветом. Яркость свечения светодиодов зависит от скорости вращения соответствующих моторов — чем выше скорость, тем ярче светится светодиод.

### Индикатор передачи данных по USB

Светодиоды **RX** и **TX** светятся во время передачи данных через виртуальный последовательный интерфейс. При получении данных светится зелёный светодиод RX, при передаче — красный светодиод TX.

### Индикаторы передачи данных через беспроводной модуль

- **RSSI** — индикатор мощности сигнала XBee-модулей. Чем ярче горит индикатор, тем больше мощность сигнала принимаемого XBee-модулем. Этот светодиод выполняет свою функцию только с модулями XBee.
- **ASC** — индикатор подключения к сети XBee-модулей. Если модуль не подключён, светодиод постоянно горит. Светодиод мигает при подключении к сети. Частота мигания зависит от роли модуля. Подробнее можно узнать из [технического описания XBee](#).

## 4. Соединение электронных компонентов робота

Теперь, когда мы знаем, как устроен контроллер, присоединим к нему двигатели и модуль Bluetooth.

Моторы присоединяются к силовым разъёмам контроллера, соединённым с встроенным двухканальным драйвером. Для этого возьмите отвертку (в комплект не входит), немного ослабив винты в разъёмах, вставьте туда контакты проводов, припаянных к моторам, и затяните винты.

Теперь рассмотрим схему присоединения модуля Bluetooth HC-06, входящего в комплект. Нам понадобятся 4 провода «мама-мама».

Найдите на плате контроллера разъём UART. Напротив контактов разъёма вы видите надписи, соответствующие их назначению.

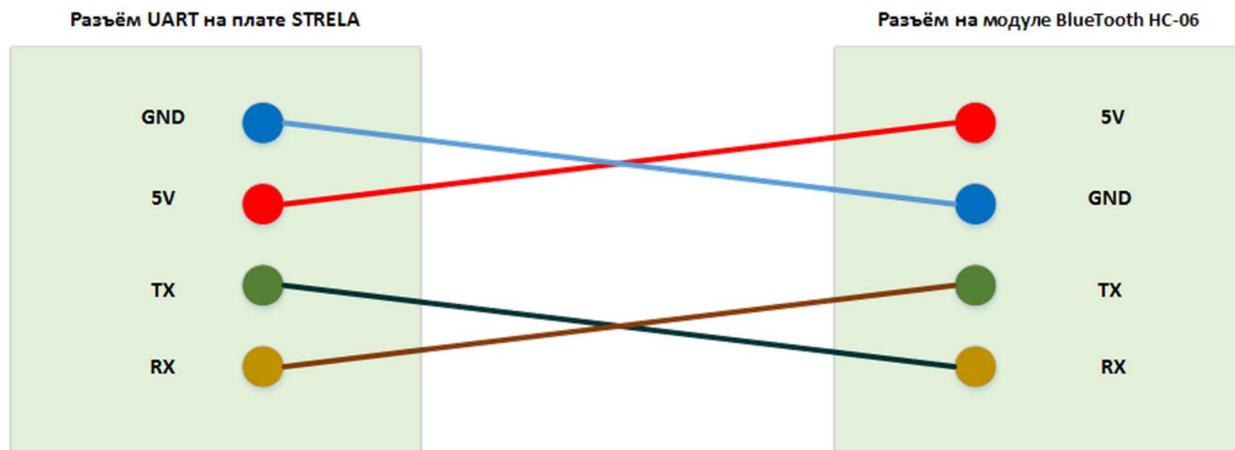
GND – «земля»

5V – питание 5 вольт

RX и TX – контакты прямой и обратной передачи данных.

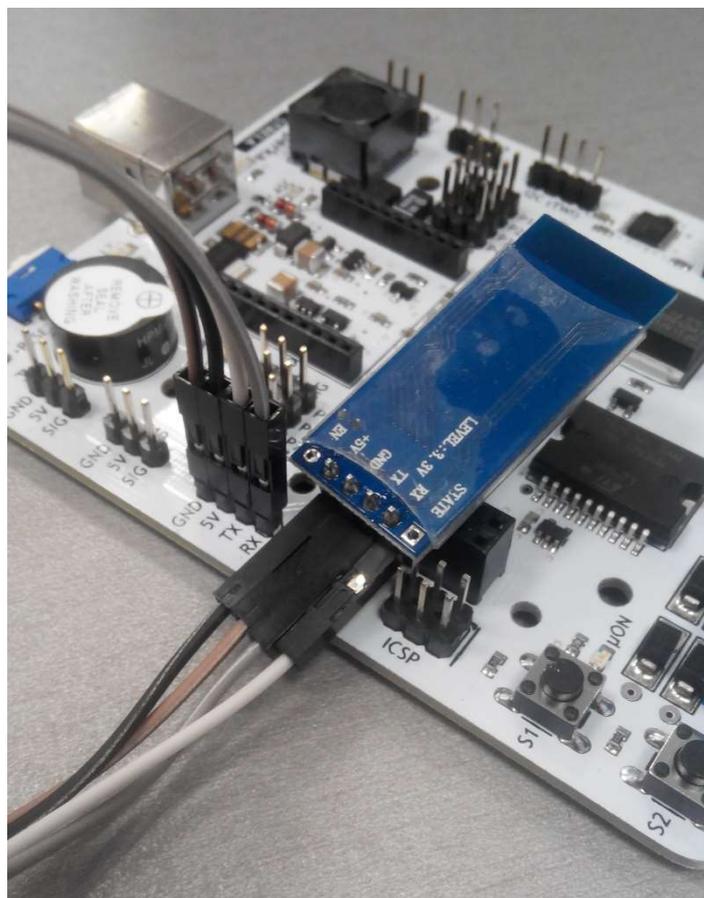
Такие же надписи вы найдёте на модуле Bluetooth. Соедините контакты модуля и контроллера следующим образом:

## Схема присоединения модуля BlueTooth HC-06 к плате STRELA

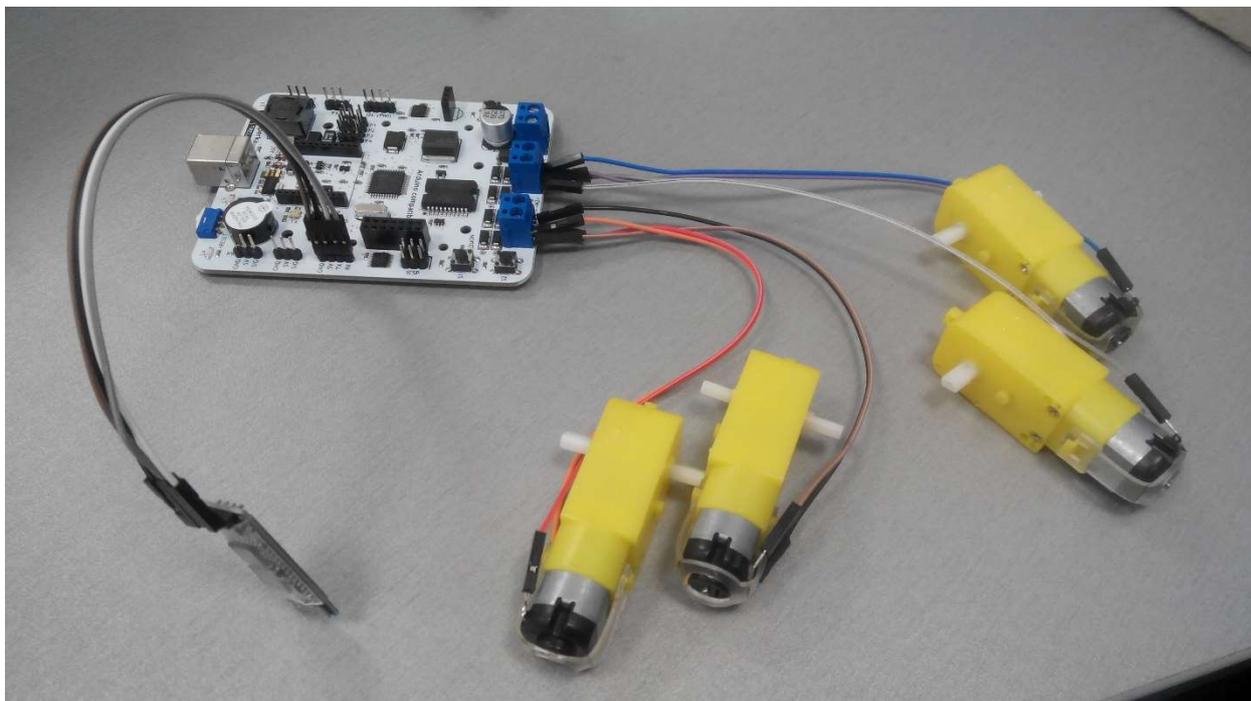


Соблюдая полярность, подключите через выключатель контакты батарейного отсека к клеммам питания контроллера (красный провод «+»).

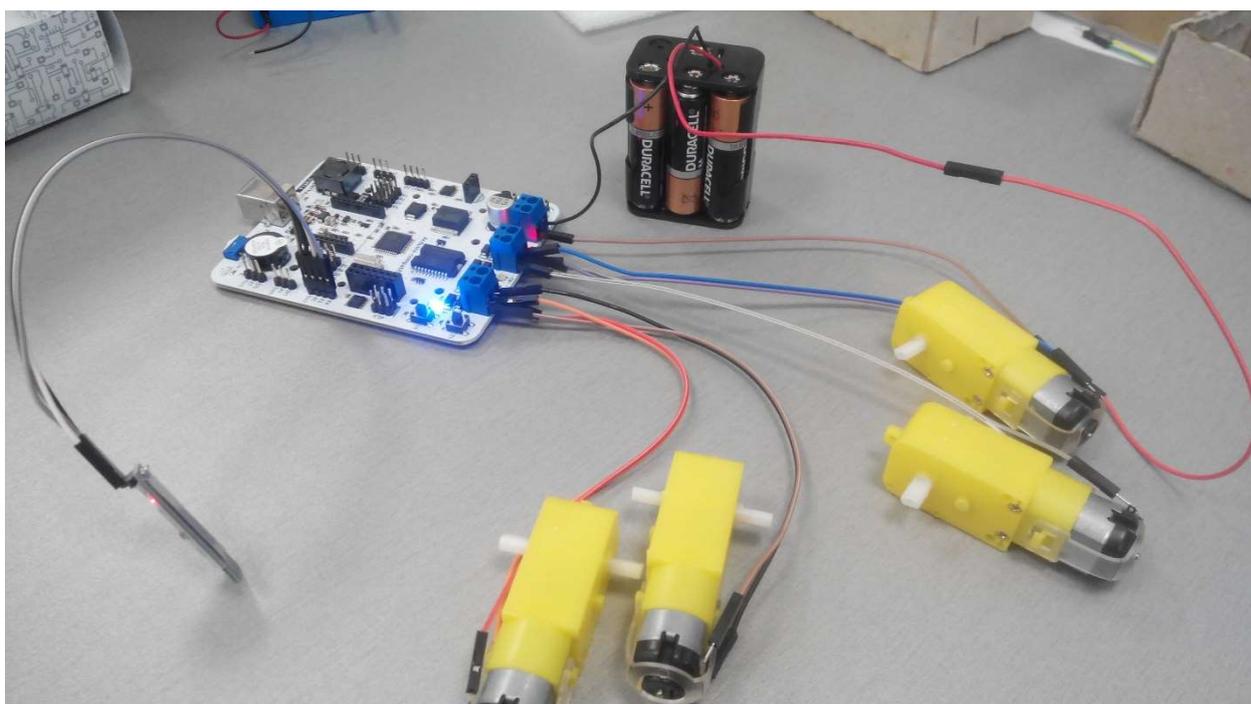
В результате должно получиться примерно так:



Моторы присоединяются попарно с помощью отвертки в разъёмы M1 и M2.



Теперь подключим к плате контроллера батарейный отсек:



На фото видно, как можно с помощью проводов, входящих в комплект, собрать электрическую цепь, которую удобно включать и выключать без выключателя, путём соединения проводов в разъёмах.

На этом сборку электрической части нашего робота можно считать законченной.

Собранную схему нужно поместить в корпус из конструктора, и наш робот, с возможностью управления через BlueTooth со смартфона или планшета, готов.

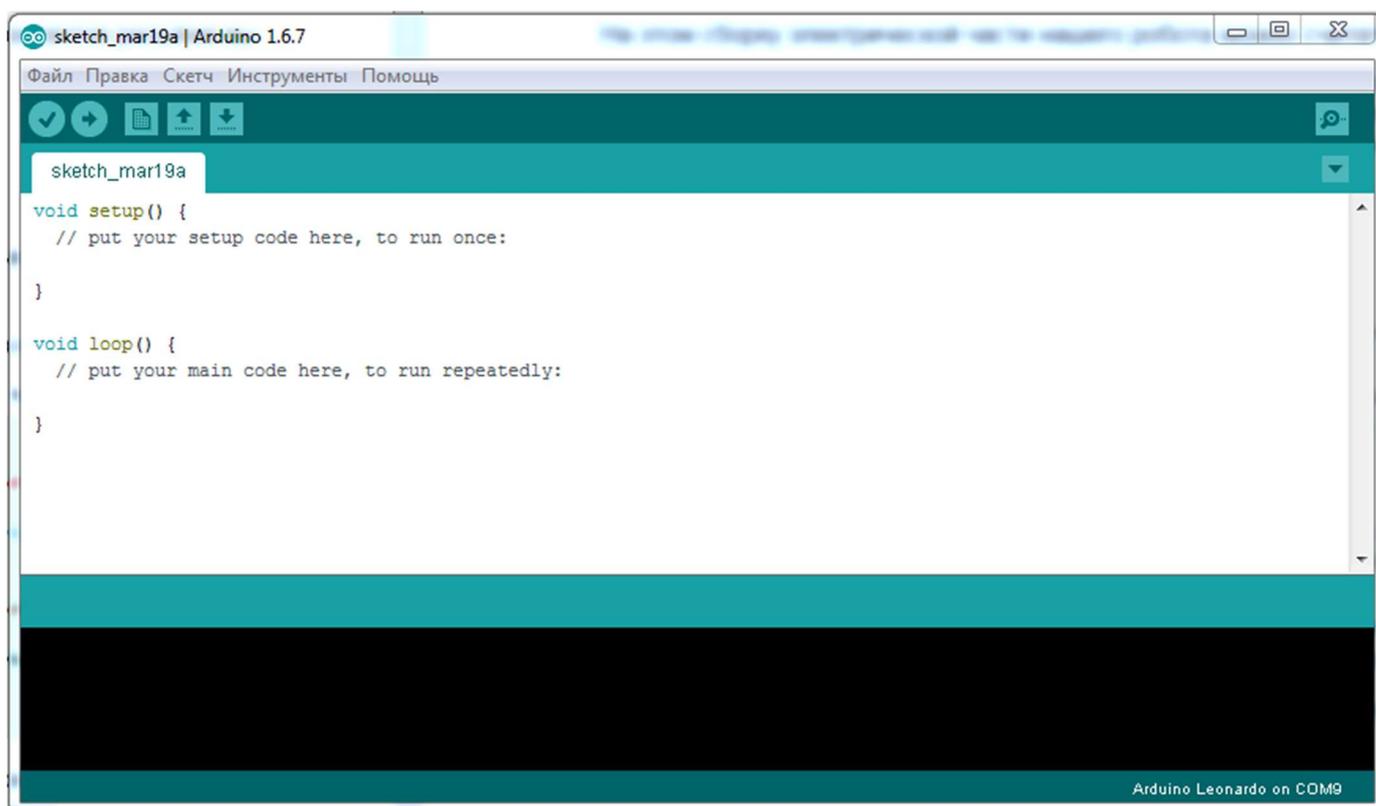
Осталось только ввести в контроллер нужную программу и запустить робота.

Далее мы рассмотрим программирование контроллера Strela в среде Arduino IDE и примеры программ. Для управления роботом со смартфона с помощью BlueTooth, применяется соответствующая библиотека, описанная в данных примерах.

## 5. Программирование в среде Arduino IDE

1. Скачайте Arduino IDE с [официального сайта](https://www.arduino.cc/en/Main/Software) <https://www.arduino.cc/en/Main/Software>.
2. Если у вас Windows и Arduino IDE из zip-файла, установите драйверы из папки `drivers`.
3. Подключите контроллер к компьютеру через USB-кабель, откройте «Диспетчер устройств Windows» и посмотрите, на какой COM-порт подключился контроллер.
4. Запустите Arduino IDE.
5. В меню «Инструменты → Плата» выберите «Arduino Leonardo».
6. В «Инструменты → Порт» выберите порт, куда она подключена.
7. Пишите программу или загружайте готовый пример.
8. Жмите значок с галочкой для проверки скетча на панели инструментов или стрелку вперед для прошивки платы.

Перепрошивать плату можно сколько угодно раз. Программа сохраняется после обесточивания платы.



Для удобства работы необходимо загрузить и установить библиотеку Strela.

Откройте страницу загрузки <https://github.com/amperka/Strela>.

Скачайте архив с библиотекой, нажав на кнопку «Download ZIP».

Скопируйте директории Strela и LiquidCrystal в `$sketchbook/libraries`, где `$sketchbook` — ваша директория со скетчами.

После этого перезагрузите Arduino IDE.

Среда программирования и нужные библиотеки установлены, подключены и готовы к работе.

Для загрузки в контроллер программы управления по BlueTooth выполните следующее:

- Подключите контроллер к компьютеру кабелем USB.
- Запустите Arduino IDE.
- В меню «Файл» выберите «Открыть» и перейдите в папку со скетчами библиотеки Strela, которые установили ранее.
- Откройте папку `ArduinoBluetoothRCCarOnStrela`.
- Выберите и загрузите (двойным щелчком мышки) файл `ArduinoBluetoothRCCarOnStrela`.
- Загрузите файл в контроллер.
- Контроллер готов к работе.

## 6. Примеры программ для различных подключаемых устройств

Простой пример управления контактами, подключёнными к Arduino-микроконтроллеру, и контактами, подключёнными к I<sup>2</sup>C-расширителю портов ввода-вывода:

[StrelaPinOutput.ino](#)

```
// Подключим библиотеку для работы с I2C-расширителем портов
#include <Wire.h>

// Подключим библиотеку Strela
#include <Strela.h>

void setup() {
    pinMode(P9, OUTPUT); // это Arduino GPIO. Мы настроим его на выход
    pinMode(P10, OUTPUT); // а это — I2C-GPIO. Его мы тоже настроим на выход
}

void loop() {
    digitalWrite(P9, HIGH); // подадим на P9 высокий уровень
    digitalWrite(P10, HIGH); // подадим на P10 высокий уровень
}
```

```

delay(1000);
uDigitalWrite(P9, LOW); // подадим на P9 низкий уровень
uDigitalWrite(P10, LOW); // подадим на P10 низкий уровень
delay(1000);
}

```

Легко можно обнаружить, что разница в управлении контактами Arduino и контактами I<sup>2</sup>C-расширителей портов отсутствует, если использовать функции `uPinMode`, `uDigitalWrite` и `uDigitalRead` из библиотеки `Strela` вместо привычных `pinMode`, `digitalWrite` и `digitalRead`. Конечно, можно использовать и привычные функции управления цифровыми контактами, но они будут работать только с Arduino-контактами. Остальные функции для работы с контактами, такие как `analogWrite` или `analogRead`, остаются без изменения и работают только с предназначенными для них Arduino-контактами.

Обратите внимание на обозначение пинов в коде. Вместо привычных цифр, которые используются в Arduino Leonardo, для обозначения номеров контактов используются значения P9 и P10. Это сделано для унификации доступа к контактам с различным способом управления, а также для упорядочивания пинов.

Давайте теперь считаем уровень с ножек:

#### [StrelaPinInput.ino](#)

```

// Подключим библиотеку для работы с I2C-расширителем портов
#include <Wire.h>
// Подключим библиотеку Strela
#include <Strela.h>
void setup() {
    uPinMode(P9, INPUT); // это Arduino GPIO. Мы настроим его на вход
    uPinMode(P10, INPUT); // а это — I2C-GPIO. Его мы тоже настроим на вход
    Serial.begin(9600); //Подготовим к работе Serial-порт
    while (!Serial) { // Так как Strela работает на том же контроллере, что и Arduino Leonardo
        ; // необходимо дождаться подключения виртуального Serial-порта
    }
}
void loop() {
    Serial.print("P9 - ");
    //Считаем значение с P9
    // и отправим его в последовательный порт

```

```

Serial.println(uDigitalRead(P9));
Serial.print("P10 - ");
//Считаем значение с P10
// и отправим его в последовательный порт
Serial.println(uDigitalRead(P10));
delay(500);
}

```

Если посмотреть в терминал, то можно увидеть первое отличие — порт P10 будет возвращать 1, а порт P9 — случайное значение. Это происходит потому, что I<sup>2</sup>C-расширитель портов на своих входах имеет встроенный подтягивающий резистор на 100 кОм. Обычно это нисколько не мешает.

## Встроенные кнопки и светодиоды

Работать со встроенными кнопками и светодиодами можно при помощи тех же функций `uDigitalRead` и `uDigitalWrite`. Так как заранее известно, что светодиоды должны управляться цифровым выходом, а считывать нажатие кнопки нужно с цифрового входа, функция `uPinMode` для этих контактов ни к чему не приводит. Их даже не нужно указывать в секции `setup`, так как библиотека `Strela` инициализирует периферию автоматически.

### [StrelaBlink.ino](#)

```

// Подключим библиотеку для работы с I2C-расширителем портов
#include <Wire.h>
// Подключим библиотеку Strela
#include <Strela.h>
void setup() {
}
void loop() {
  uDigitalWrite(L1, HIGH); // Зажжем первый светодиод
  delay(1000);
  uDigitalWrite(L1, LOW); // И погасим его
  delay(1000);
}

```

В следующем примере мы будем считывать значение с кнопок и зажигать светодиоды. Будем зажигать светодиод под тем же номером, что и нажатая кнопка.

### [StrelaLedsAndButtons.ino](#)

```

// Подключим библиотеку для работы с I2C-расширителем портов

#include <Wire.h>

// Подключим библиотеку Strela

#include <Strela.h>

void setup() {

}

void loop() {

  for (byte i = S1; i <= S4; ++i) // Будем проверять состояние кнопок по очереди
  {

    byte buttonNumber = i - S1; //Определим номер текущей кнопки

    bool buttonState = uDigitalRead(i); //Считаем положение кнопки

    //Подадим считанное значение с кнопки на светодиод с тем же номером

    uDigitalWrite(L1 + buttonNumber, buttonState);

  }

}

```

## Пищалка

Добавим к предыдущему скетчу звуковое сопровождение:

[StrelaLedsButtonsAndBuzzer.ino](#)

```

// Подключим библиотеку для работы с I2C-расширителем портов

#include <Wire.h>

// Подключим библиотеку Strela

#include <Strela.h>

void setup() {

}

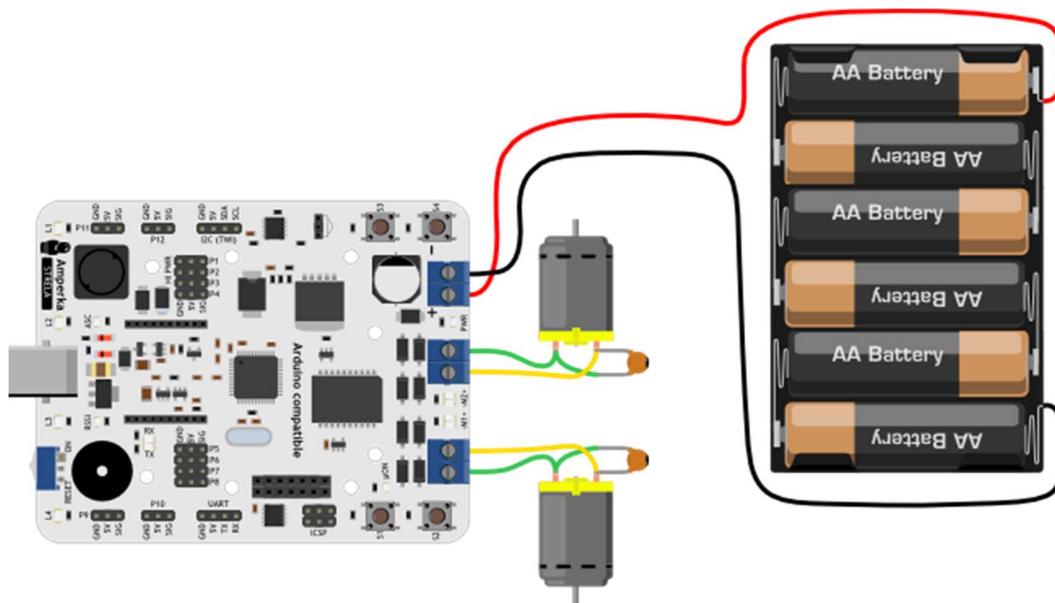
```

```

void loop() {
  for (byte i = S1; i <= S4; ++i) // Будем проверять состояние кнопок по очереди
  {
    byte buttonNumber = i - S1;    //Определим номер текущей кнопки
    bool buttonState = uDigitalRead(i); //Считаем положение кнопки
    if (buttonState)    //Если она была нажата...
      tone(BUZZER, 1000, 50); //...пикнем пищалкой с частотой 1000 Гц, 50 мс
    //Подадим считанное значение с кнопки на светодиод с тем же номером
    uDigitalWrite(L1 + buttonNumber, buttonState);
  }
}

```

## Управление коллекторными двигателями



Теперь перейдём к работе с двигателями.

[StrelaMotors.ino](http://StrelaMotors.ino)

```
// Подключим библиотеку для работы с I2C-расширителем портов
```

```
#include <Wire.h>
```

```
// Подключим библиотеку Strela

#include <Strela.h>

void setup() {

    motorConnection(1, 0); // Я неправильно прикрутил один мотор

    //поэтому, чтобы их не перекручивать

    //можно воспользоваться этой функцией.

    //Направление вращения мотора 1 будет изменено.

}

void loop() {

    //Моторы управляются функцией drive(int m1, int m2).

    //m1 (m2) - это скорость вращения мотора 1 (2).

    //Скорость регулируется в пределах от -255 до 255

    //Если это число положительное - мотор будет вращаться вперёд,

    //если отрицательное - назад.

    //Фаза 1

    digitalWrite(L1, HIGH); // Зажгли светодиод 1

    drive(127, 127); //Средний ход вперёд

    delay(2000); // в течение 2 секунд.

    //Фаза 2

    digitalWrite(L2, HIGH); // Зажгли светодиод 2

    drive(127, 0); //Поворот на право

    delay(1000); // в течение 1 секунды

    //Фаза 3

    digitalWrite(L3, HIGH); // Зажгли светодиод 3
```

```

drive(-255, -255); //Полный назад

delay(1000); // в течение 1 секунды

//Фаза 4

uDigitalWrite(L4, HIGH); // Зажгли светодиод 4

for (int i = 0; i <= 255; ++i)

{

drive(i, -i); //Разворот на месте с ускорением

delay(1);

}

//Фаза 5

drive(0, 0); //Стоп

for (int i = L1; i <= L4; i++)

{

uDigitalWrite(i, LOW); // Гасим все светодиоды

}

// Всё, приехали

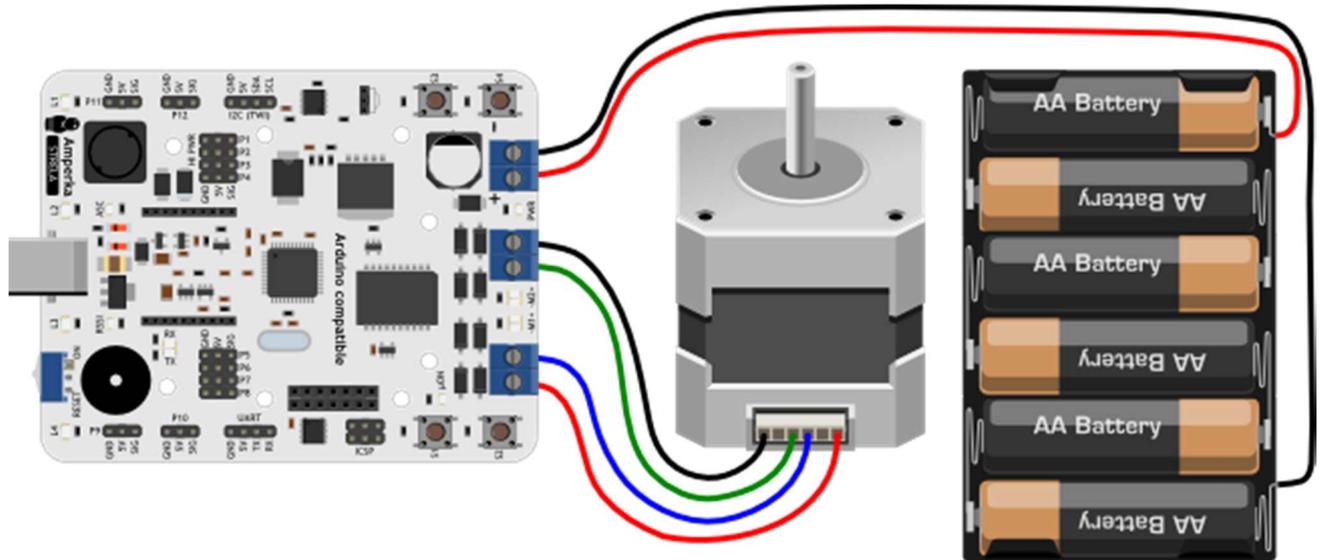
while (true)

;

}

```

**Управление шаговым двигателем**



[StrelaStepper.ino](https://github.com/Amperka/StrelaStepper.ino)

```
// Подключим библиотеку для работы с I2C-расширителем портов
```

```
#include <Wire.h>
```

```
// Подключим библиотеку Strela
```

```
#include <Strela.h>
```

```
void setup() {
```

```
}
```

```
void loop() {
```

```
    // Шаговый двигатель управляется при помощи
```

```
    // функции stepperMotor(int stepsToMove, int stepDelay);
```

```
    // stepsToMove — это количество шагов которые должен
```

```
    // сделать шаговый двигатель.
```

```
    // Если это число положительное — мотор крутится в одну сторону,
```

```
    // если отрицательное - в другую.
```

```
    // stepDelay — это пауза между шагами в миллисекундах.
```

```
    // От этой величины зависит скорость вращения мотора.
```

```
// Чем больше пауза, тем медленнее вращается мотор
```

```
// Функция stepperMotor() управляет вращением мотора только в
```

```
// полушаговом режиме. В этом режиме шаговый двигатель
```

```
// развивает наибольшую мощность и точность позиционирования.
```

```
// Быстро шагаем 200 раз вперёд
```

```
stepperMotor(200, 5);
```

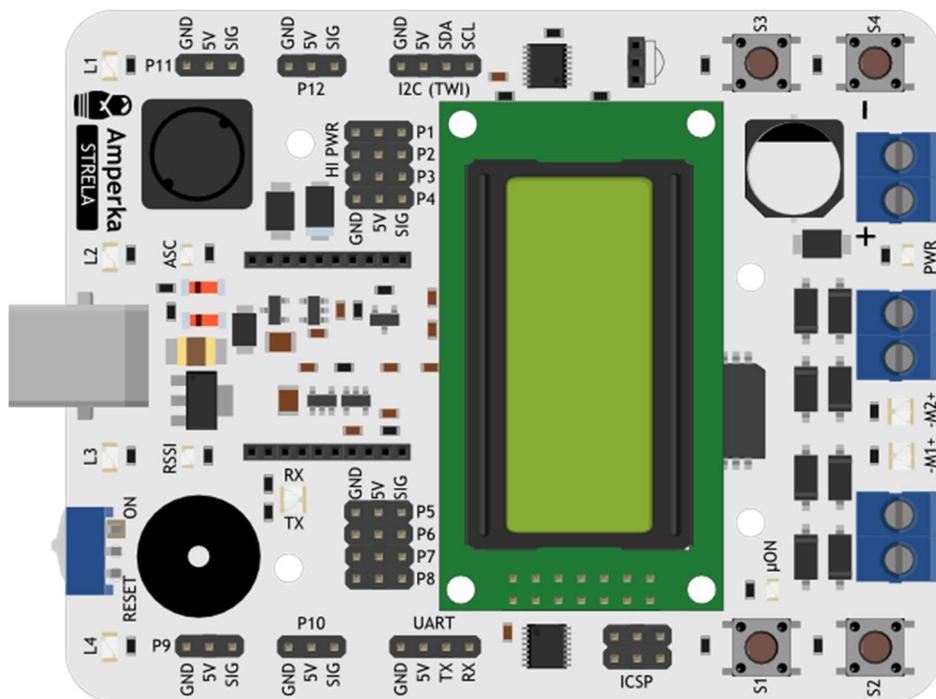
```
// Медленно шагаем 200 раз назад.
```

```
// Мотор вернётся в исходное положение
```

```
stepperMotor(-200, 10);
```

```
}
```

## Жидкокристаллический дисплей



Подключим к плате LCD-экран [MT-08S2A](#). Управлять мы им будем при помощи библиотеки [LiquidCrystal I2C](#). Мы уже добавили гарантированно работающую версию этой библиотеки в папку с библиотекой [Strela](#), поэтому отдельно её устанавливать не нужно.

[StrelaLiquidCrystal.ino](#)

```
// Подключим библиотеку для работы с I2C-расширителем портов
```

```
#include <Wire.h>

// Подключим библиотеку Strela

#include <Strela.h>

// Подключим библиотеку для работы с LCD-эраном через I2C

#include <LiquidCrystal_I2C.h>

// Создадим объект lcd, который будет выводить текст на экран

LiquidCrystal_I2C lcd(LC_ADDR, LCEN, LCRW, LCRS, LC4, LC5, LC6, LC7);

void setup()

{

//Инициализация экрана. У нашего экрана 8 символов 2 строки

lcd.begin(8, 2);

//Переместим курсор в начало первой строки (символ 0, строка 0)

lcd.home();

//Печатаем строку

lcd.print("Hello");

//Переместим строку в начало второй строки (символ 0, строка 1)

lcd.setCursor(0, 1);

//Печатаем строку

lcd.print("World!");

//Пауза, чтобы успеть насладиться результатом

delay(2000);

}

void loop()

{
```

```
// Далее будем просто выводить время прошедшее со старта скетча

//Очищаем экран

lcd.clear();

// выставляем курсор в положение 0, 0

lcd.home();

// Пишем первую строку

lcd.print("Uptime:");

// переместим курсор в начало второй строки

lcd.setCursor(0, 1);

// узнаем время прошедшее со старта скетча в миллисекундах

unsigned long time = millis();

// и выведем его на экран

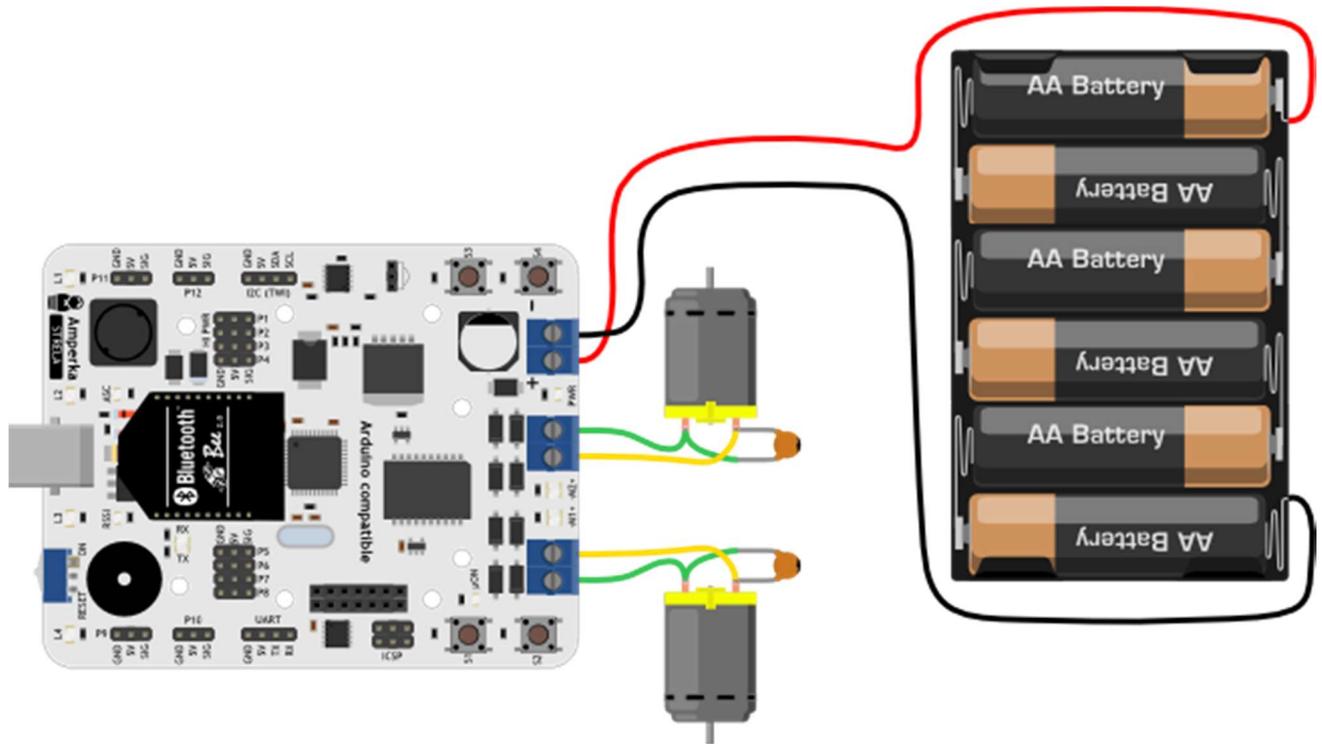
lcd.print(time);

//Пауза, чтобы успеть насладиться результатом

delay(500);

}
```

**Управление роботом с мобильного телефона через Bluetooth-модуль**



Подключим к плате [модуль Bluetooth](#). Будем управлять нашим роботом дистанционно с помощью мобильного телефона на ОС Android.

Для Android существует огромное количество приложений, при помощи которых можно управлять Arduino через Bluetooth. В данном случае для управления Strela мы использовали [Arduino Bluetooth RC Car](#).

[ArduinoBluetoothRCCarOnStrela.ino](#)

//Arduino Bluetooth RC Car in Strela

//Original App:

//<https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller>

#include <Wire.h> // Библиотека для работы с I2C

#include <Strela.h> // Библиотека для работы со Стрелой

int velocity = 0; //Здесь будет храниться значение скорости

int defaultSpeed = 100; // это число мы будем использовать в логике поворотов

void setup()

{

Serial1.begin(9600); //Bluetooth Bee по умолчанию использует эту скорость

motorConnection(1, 0); // Я неправильно прикрутил один мотор

//поэтому, чтобы их не перекручивать

//можно воспользоваться этой функцией.

//Направление вращения мотора 1 будет изменено.

```

}
void loop()
{
  if (Serial1.available() > 0) //Если появились новые команды
  {
    control(); //вызываем функцию управления
  }
  //Здесь можно написать ещё много своего кода
}
void control() // функция управления
{
  char dataIn = Serial1.read(); //Считаем значение пришедшей команды
  if (dataIn == 'F') //Если пришла команда "F"
    drive(velocity, velocity); //едем вперёд
  else if (dataIn == 'B') //или если пришла команда "B"
    drive(-velocity, -velocity); //едем назад
  else if (dataIn == 'L') //или если пришла команда "L"
    drive(-velocity, velocity); //поворачиваем налево на месте
  else if (dataIn == 'R') //или если пришла команда "R"
    drive(velocity, -velocity); //поворачиваем направо на месте
  else if (dataIn == 'I') //или если пришла команда "I", едем вперёд и направо
    drive(defaultSpeed + velocity, defaultSpeed - velocity);
  else if (dataIn == 'J') //или если пришла команда "J", едем назад и направо
    drive(-defaultSpeed - velocity, -defaultSpeed + velocity);
  else if (dataIn == 'G') //или если пришла команда "I", едем вперёд и налево
    drive(defaultSpeed - velocity, defaultSpeed + velocity);
  else if (dataIn == 'H') //или если пришла команда "H", едем назад и налево
    drive(-defaultSpeed + velocity, -defaultSpeed - velocity);
  else if (dataIn == 'S') //или если пришла команда "S", стоим
    drive(0, 0);
  else if (dataIn == 'U') //или если "U", зажигаем "передние фары"
  {

```

```

    uDigitalWrite(L2, HIGH);
    uDigitalWrite(L3, HIGH);
}
else if (dataIn == 'u') //или если "u", гасим "передние фары"
{
    uDigitalWrite(L2, LOW);
    uDigitalWrite(L3, LOW);
}
else if (dataIn == 'W') //или если "W", зажигаем "задние фары"
{
    uDigitalWrite(L1, HIGH);
    uDigitalWrite(L4, HIGH);
}
else if (dataIn == 'w') ///или если "w", гасим "задние фары"
{
    uDigitalWrite(L1, LOW);
    uDigitalWrite(L4, LOW);
}
// если к нам пришло значение от 0 до 9
else if (((dataIn - '0') >= 0) && ((dataIn - '0') <= 9))
    velocity = (dataIn - '0') * 25; //сохраняем новое значение скорости
else if (dataIn == 'q') //если "q" - полный газ!
    velocity = 255;
}

```

## 7. Датчики, рекомендуемые для работы

В работе могут быть использованы любые датчики, совместимые с питанием 5 Вольт и платформой Arduino. Рассмотрим некоторые из них.

## 1. Магнетометр/компас (Тройка-модуль)



Модуль на основе чипа [LIS3MDL](#) от STMicroelectronics.

Микросхема имеет встроенный датчик температуры. Это позволит добиться точной работы датчика даже в экстремальных условиях. Общение модуля с управляющей электроникой, такой как [Arduino](#), происходит по протоколу [I<sup>2</sup>C / TWI](#). А это значит, что для подключения вам понадобится всего два контакта. На модуле присутствует регулятор напряжения и специальный I<sup>2</sup>C-буфер, поэтому можно использовать его с управляющей электроникой напряжением 3,3...5 В.

Подключение этого Тройка-модуля отличается от стандартного: он имеет два трёхконтактных разъёма. Один разъём используется для подачи напряжения. Другой — для подключения к шине I<sup>2</sup>C. Два [3-проводных шлейфа](#) для подключения включены в комплект.

- Чувствительность:  $1.46 \times 10^{-4}$  Гс
- Диапазон измерений:  $\pm 4 / \pm 8 / \pm 12 / \pm 16$  Гс
- Напряжение питания: 3,3–5 В
- Потребляемый ток: менее 10 мА
- Габариты: 25,4×25,4 мм

## 2. Гироскоп (Тройка-модуль)



Модуль на основе чипа [L3G4200D](#) от STMicroelectronics.

Микросхема имеет встроенный датчик температуры. Это позволит добиться точной работы датчика даже в экстремальных условиях. Общение модуля с управляющей электроникой, такой как [Arduino](#), происходит по протоколу [I<sup>2</sup>C / TWI](#). А это значит, что для подключения вам понадобится всего два контакта.

Модуль имеет регулятор напряжения и специальный I<sup>2</sup>C-буфер, поэтому вы смело можете использовать его с управляющей электроникой напряжением 3,3...5 В.

Подключение этого Тройка-модуля отличается от стандартного: он имеет два трёхконтактных разъёма. Один разъём используется для подачи напряжения. Другой — для подключения к шине I<sup>2</sup>C. Два [3-проводных шлейфа](#) для подключения включены в комплект.

- Чувствительность:  $8.7 \times 10^{-3}$  град/сек
- Диапазон измерений:  $\pm 250 / \pm 500 / \pm 2000$  град/сек
- Напряжение питания: 3,3–5 В
- Потребляемый ток: менее 10 мА
- Габариты: 25,4×25,4 мм

### 3. Цифровой датчик температуры и влажности (Тройка-модуль)



Данные температуры и влажности датчик отдаёт по одному проводу в виде [цифрового сигнала](#). Это позволяет передавать данные на расстояние до нескольких десятков метров.

В сердце модуля — популярная среди любителей сенсорная сборка DHT11. Он работает по собственному протоколу, а все функции для удобной работы по этому протоколу с Arduino упакованы в [готовую библиотеку «ТройкаDHT11»](#).

Подключайте модуль к управляющей электронике с помощью [3-х проводов](#). [Шлейф](#) для подключения включён в комплект.

- Напряжение питания: 3–5 В
- Потребляемый ток при запросе данных: 2,5 мА
- Потребляемый ток в ожидании: 100 мкА
- Диапазон температур: 0–50 °C
- Погрешность температуры:  $\pm 2$  °C
- Диапазон влажности: 20–90%
- Погрешность влажности:  $\pm 5\%$
- Габариты: 25×25 мм

Выходом датчика является [цифровой сигнал](#). Температура и влажность передаются по одному сигнальному проводу (S). DHT11 общается с принимающей стороной, такой как [Arduino](#), по собственному протоколу. Коммуникация двунаправлена и в общих чертах выглядит так:

1. Микроконтроллер говорит о том, что хочет считать показания. Для этого он устанавливает сигнальную линию в 0 на некоторое время, а затем устанавливает её в 1.
2. Сенсор подтверждает готовность отдать данные. Для этого он аналогично сначала устанавливает сигнальную линию в 0, затем в 1.

3. После этого сенсор передаёт последовательность 0 и 1, последовательно формирующих 5 байт (40 бит). В первых двух байтах передаётся температура, в третьем-четвёртом — влажность, в пятом — контрольная сумма, чтобы микроконтроллер смог убедиться в отсутствии ошибок считывания.

Благодаря тому, что сенсор делает измерения только по запросу, достигается энергоэффективность: пока общения нет, датчик потребляет ток 100 мкА.

#### 4. Акселерометр (Тройка-модуль)



Модуль на основе чипа [LIS331DLH](#) от STMicroelectronics. Точно такой же акселерометр установлен на iPhone 4S.

Микросхема имеет встроенный датчик температуры. Это позволит добиться точной работы датчика даже в экстремальных условиях. Общение модуля с управляющей электроникой, такой как [Arduino](#), происходит по протоколу [I<sup>2</sup>C / TWI](#), а это значит, что для подключения понадобится всего два контакта.

Мы поместили на модуль регулятор напряжения и специальный I<sup>2</sup>C-буфер, поэтому вы смело можете использовать его с управляющей электроникой напряжением 3,3...5 В.

Подключение этого Тройка-модуля отличается от стандартного: он имеет два трёхконтактных разъёма. Один разъём используется для подачи напряжения. Другой — для подключения к шине I<sup>2</sup>C. Два [3-проводных шлейфа](#) для подключения включены в комплект.

- Чувствительность:  $9.8 \times 10^{-3}$  м/с<sup>2</sup>
- Диапазон измерений:  $\pm 2 / \pm 4 / \pm 8$  g
- Напряжение питания: 3,3–5 В
- Потребляемый ток: менее 10 мА
- Габариты: 25,4×25,4 мм

#### 5. Датчик Хола (Тройка-модуль)



Датчику Холла можно найти очень много разных применений. Помимо определения факта приближения магнита к датчику и определения полюса магнита, датчик Холла, при помощи небольших приспособлений, можно использовать как тахометр, датчик тока, датчик закрытия дверей, датчик положения вала для управления бесколлекторным двигателем, датчик уровня воды. Из двух датчиков Холла можно собрать энкодер. И это только небольшая часть возможных применений датчика Холла.

Датчик Холла выполнен на основе микросхемы SS49E.

Модуль из серии [Troyka Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

Выходным результатом работы сенсора является [аналоговый сигнал](#). В отсутствии магнитного поля датчик выдаёт половину напряжения питания. При появлении магнитного поля значение отклоняется к нулю или напряжению питания, в зависимости от полярности магнитного поля и пропорционально его интенсивности.

Датчик подключается к управляющей электронике через [3 провода](#).

## 6. Светодиод «Пиранья» (Тройка-модуль)



Этот светодиодный модуль содержит в себе очень яркий светодиод, поэтому эти светодиодные модули можно использовать не только для индикации. Из них можно собрать светомузыку, светофор, разнообразные гирлянды для новогодней ёлки или «лампу настроения».

Сверхъяркие светодиоды потребляют ток больший, чем тот, что могут дать микроконтроллеры напрямую, поэтому этот светодиодный модуль имеет на борту транзистор, через который и происходит управление светодиодом. Это же означает, что модуль совместим как с сигналами на уровне 5 вольт, так и на уровне в 3,3 вольта.

Модуль из серии [Тройка Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

**Осторожно!** Эти светодиодные модули очень яркие, поэтому не смотрите в середину светодиода с близкого расстояния - это очень вредно для глаз.

Светодиодный модуль подключается к управляющей электронике через [3 провода](#).

## 7. Барометр (Тройка-модуль)



Модуль на основе чипа [LPS331AP](#) от STMicroelectronics.

Микросхема имеет встроенный датчик температуры. Это позволит добиться точной работы датчика даже в экстремальных условиях. Общение модуля с управляющей электроникой, такой как [Arduino](#), происходит по протоколу [I<sup>2</sup>C / TWI](#), а это значит, что для подключения вам понадобится всего два контакта.

На модуле есть регулятор напряжения и специальный I<sup>2</sup>C-буфер, поэтому вы смело можете использовать его с управляющей электроникой напряжением 3,3...5 В.

Подключение этого Тройка-модуля отличается от стандартного: он имеет два трёхконтактных разъёма. Один разъём используется для подачи напряжения. Другой — для подключения к шине I<sup>2</sup>C. Два [3-проводных шлейфа](#) для подключения включены в комплект.

- Чувствительность:  $2.4 \times 10^{-4}$  мбар
- Диапазон измерений: 1260 мбар
- Напряжение питания: 3,3–5 В
- Потребляемый ток: менее 10 мА
- Габариты: 25,4×25,4 мм

## 8. Датчик освещенности (Тройка-модуль)



С датчиком уровня освещённости устройство сможет отличить день от ночи, солнечную погоду от пасмурной, тень от света. С помощью сенсора освещённости и [лазерного модуля](#) можно сделать датчик для сигнализации, срабатывающий при пересечении лазерного луча, или хитрый фото-триггер для фотоаппарата. Добавив к этому сенсору [светодиодный модуль](#), можно сделать датчик препятствий.

Модуль из серии [Troyka Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

Выходным результатом работы сенсора является [аналоговый сигнал](#). Выходное напряжение датчика обратно пропорционально интенсивности падающего света.

Датчик подключается к управляющей электронике через [3 провода](#).

[Шлейф](#) для подключения включён в комплект.

## 9. ИК-приёмник (Тройка-модуль)



Используя модуль с ИК-приёмником, вы можете организовать дистанционное управление вашим устройством при помощи обычного пульта дистанционного управления от бытовой техники. Если использовать этот приёмник вместе с [инфракрасным светодиодом](#), то можно организовать дешёвую беспроводную двустороннюю полудуплексную связь между двумя вашими устройствами через UART.

В модуле с ИК-приёмником используется микросхема TSOP22.

Модуль из серии [Troyka Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

Выходным результатом работы сенсора является [цифровой сигнал](#).

Приёмник подключается к управляющей электронике через [3 провода](#).

Приёмник может быть использован для получения и декодирования сигнала от обычных пультов бытовой техники. Однако зачастую детали протокола сложно найти, они специфичны для каждой конкретной модели пульта, поэтому, чтобы «подружить» управляющую электронику и ИК-приёмник, потребуется исследование. В качестве отправной точки можно использовать [статью Кена Ширриффа](#).

Несущая частота этого ИК-приёмника — 38 кГц. Philips использует в своих пультах несущую частоту 36 кГц, Sony и Panasonic — 40 кГц, подавляющее большинство других производителей — 38 кГц. Несущая частота ИК-приёмника и пульта управления должны совпадать.

При использовании ИК-приёмника в качестве беспроводного UART-интерфейса, нужно иметь в виду, что с помощью ИК-приёмника можно принимать только небольшие порции данных со скоростью не более 1200 бит в секунду.

- Несущая частота приёмника: 38 кГц
- Напряжение питания: от 2,5 до 5,5 В
- Габариты: 25,4×25,4 мм

## 10. Аналоговый термометр (Тройка-модуль)



Этот датчик температуры можно использовать для измерения температуры воздуха. Если предварительно позаботиться об электрической изоляции модуля, то можно его использовать для измерения температуры воды или масла.

Модуль выполнен на основе микросхемы TMP36. Он работает в диапазоне температур от  $-40\text{ }^{\circ}\text{C}$  до  $+125\text{ }^{\circ}\text{C}$ .

Модуль из серии [Troyka Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

Выходным результатом работы сенсора является [аналоговый сигнал](#) с уровнем напряжения прямо пропорциональным температуре с шагом  $10\text{ мВ}/^{\circ}\text{C}$ . Благодаря этому перевести считанный сигнал в градусы на микроконтроллере крайне просто.

Датчик подключается к управляющей электронике через [3 провода](#). [Шлейф](#) для подключения включён в комплект.

- Диапазон измеряемой температуры:  $-40\dots+125\text{ }^{\circ}\text{C}$
- Точность:  $\pm 1\text{ }^{\circ}\text{C}$  при температуре  $25\text{ }^{\circ}\text{C}$ ,  $\pm 3\text{ }^{\circ}\text{C}$  на всём диапазоне измерения.
- Напряжение питания: от 2,7 В до 5,5 В

- Потребляемый ток: 50 мА
- Габариты: 25,4×25,4 мм

### 11. Зуммер (Тройка-модуль)



Пьезодинамик может воспроизводить звук. Он может подавать тревожные звуки при срабатывании сигнализации или воспроизводить победную мелодию после выигрыша в игре. В паре со светодиодом можно создать индикацию, которая не останется незамеченной. Модуль из серии [Тройка Module](#) можно красиво разместить на лицевой панели вашего устройства, прикрутив его винтами М3.

Модуль с пьезодинамиком подключается к управляющей электронике через [3 провода](#). [Шлейф](#) для подключения включён в комплект.

Для работы со звуком при подключении пьезодинамика к [Arduino](#), можно использовать встроенную функцию [tone](#) или богатую возможностями [библиотеку Tone](#).

- Номинальная частота: 4 кГц
- Интенсивность: 80 дБ
- Номинальное рабочее напряжение: 5 В
- Габариты: 25,4×25,4 мм

### 12. Силовой ключ (Тройка-модуль)



Микроконтроллер, такой как [Arduino](#), не может напрямую управлять мощной нагрузкой. Его выходы могут отдавать только небольшой ток. Если вы хотите управлять мощной нагрузкой, такой как светодиодная лента, [погружная помпа](#) или [электродвигатель](#), вам необходим какой-нибудь ключ. Этот модуль как раз является ключом, выполненным на базе мощного полевого (MOSFET) транзистора.

Для решения задачи вы можете взять и [реле](#), но у полевого транзистора есть перед ним преимущество - он поддерживает [ШИМ](#), а это значит, что вы можете регулировать скорость вращения двигателя или яркость свечения светодиодной ленты. Правда, в отличие от реле, полевой транзистор может коммутировать только цепь с постоянным током.

Модуль основан на N-канальном полевом транзисторе [IRLR8113](#). Он спроектирован таким образом, чтобы как можно лучше рассеивать тепло, образующееся при его работе. Это значит, что транзистор не выйдет из строя от перегрева.

Ключ замыкается, когда на вход модуля подаётся логическая единица. При этом управление производится через встроенный маленький ключ, что даёт возможность использовать в качестве управляющего сигнала любое напряжение от 3 до 5 вольт. Поэтому модулем можно свободно управлять с большинства плат, в том числе с [Raspberry Pi](#)

На модуле присутствует светодиод, показывающий состояние транзистора. Он горит, когда транзистор открыт.

Модуль подключается к управляющей электронике через [3 провода](#).

[Шлейф](#) для подключения включён в комплект.

Клеммник под отвёртку предназначен для подведения проводов цепи, которая будет замыкаться/размыкаться.

- Максимальное напряжение сток-исток: 30 В
- Максимальный ток сток-исток: 12 А (при напряжении на затворе 5 В)
- Сопротивление сток-исток при открытом затворе ( $R_{Dson}$ ): 5,8 мОм
- Габариты: 25,4×25,4 мм

### 13. Датчик линии аналоговый



Датчик линии идеально подходит для установки на днище [мобильной платформы](#), чтобы заставить робота не выезжать за пределы территории обозначенной контуром или чтобы он следовал за нарисованной линией.

Сенсор способен не только отличать чёрную поверхность от белой. В отличие от [цифрового датчика линии](#), он способен отличить оттенки серого. Это даёт вам возможность точно контролировать процесс перехода границы от чёрного к белому и наоборот, ведь результатом является усреднённое значение небольшого пятна под фотоэлементом.

Основной фотоэлемент работает в инфракрасном спектре. Дополнительно на сенсоре установлен светодиод, который загорается, когда поверхность под датчиком светлая. Это удобно для диагностики и настройки.

Переменный резистор, установленный на сенсоре, позволит регулировать чувствительность сенсора в широких пределах. Это позволяет откалибровать датчик под вашу конструкцию, материал покрытия и различные условия внешнего освещения.

Выходным результатом работы сенсора является [аналоговый сигнал](#). Чем светлее поверхность под сенсором, тем меньше его выходное напряжение.

Датчик подключается к управляющей электронике через [3 провода](#).

[Шлейф](#) для подключения включён в комплект.

- Напряжение питания: 3–5 В
- Потребляемый ток: менее 10 мА

#### 14. Датчик линии цифровой



Датчик линии позволяет определять цвет поверхности около него. Выходом является простой [бинарный цифровой сигнал](#): логический 0 или 1, в зависимости от цвета, который он видит перед собой. Единица — чёрный или пустота, ноль — не чёрный.

Основной фотоэлемент работает в инфракрасном спектре. Дополнительно на сенсоре установлен светодиод, который загорается, когда поверхность под датчиком светлая. Это удобно для диагностики и настройки.

Переменный резистор, установленный на сенсоре, позволит регулировать чувствительность сенсора в широких пределах. Это позволяет откалибровать датчик под вашу конструкцию, материал покрытия и различные условия внешнего освещения.

Сенсор может быть закреплён на днище [мобильной платформы](#), чтобы заставить робота не выезжать за пределы территории обозначенной контуром или чтобы он следовал за нарисованной линией.

Также его можно использовать в роли энкодера вращения: разместив на вращающемся объекте нарисованную чёрно-белую мишень, можно понять, с какой угловой скоростью вращается объект, и сколько оборотов он сделал.

Датчик подключается к управляющей электронике через [3 провода](#).

[Шлейф](#) для подключения включён в комплект.

Если вам необходимы градации в измерении яркости цвета объекта, т.е. нужен [аналоговый сигнал](#), обратите внимание на [аналоговый датчик линии](#).

- Напряжение питания: 3–5 В
- Потребляемый ток: менее 10 мА

### 15. Датчик наклона (Тройка-модуль)



Датчик наклона — это капсула с металлическим шариком внутри. Шарик перекачивается в капсуле и замыкает или размыкает цепь. Таким образом, датчик выдаёт простой [цифровой сигнал](#): логический ноль или единицу, в зависимости от того, в какую сторону наклонена капсула.

Модуль подключается к управляющей электронике через [3 провода](#).

### 16. Датчик шума (Тройка-модуль)



Сенсор можно использовать для разработки [замка, открываемого кодовым стуком](#), изучения качества шумоизоляции, в охранных системах или, чтобы измерить интенсивность аплодисментов после вашего выступления.

Для измерения уровня шума вам не хватит простого микрофона. Во-первых, его сигнал, как правило, очень слаб, и его нужно усиливать. Во-вторых, сигнал с микрофона всегда будет представлять собой колебания относительно нуля, ведь так и выглядит звуковая волна. Поэтому вероятность получить нулевое значение высока, даже находясь рядом с работающим перфоратором. Уровень громкости характеризуется интегралом, т.е. площадью под графиком сигнала с микрофона. Усилением и интегрированием сигнала с электретного микрофона как раз

и занимается датчик шума.

Сенсор выдаёт аналоговый сигнал в диапазоне 0–5 В. Выходное напряжение пропорционально средней шумности за последние несколько сотен миллисекунд.

Для регулировки чувствительности на модуле предусмотрен триммер.

Модуль подключается к управляющей электронике с помощью [3-х проводов](#). [Шлейф](#) для подключения включены в комплект.

- Напряжение питания: 3,3–5 В
- Потребляемый ток: менее 10 мА
- Габариты: 25,4×25,4 мм

## 17. Ультразвуковой дальномер HC-SR04



Этот дальномер может служить прекрасным датчиком для робота, благодаря которому он сможет определять расстояния до объектов, объезжать препятствия или строить карту помещения. Его можно также использовать в качестве датчика для сигнализации, срабатывающего при приближении объектов.

Ультразвуковой дальномер определяет расстояние до объектов точно так же, как это делают дельфины или летучие мыши. Он генерирует звуковые импульсы на частоте 40 кГц и слушает эхо. По времени распространения звуковой волны туда и обратно можно однозначно определить расстояние до объекта.

В отличие от [инфракрасных дальномеров](#), на показания ультразвукового дальномера не влияют засветки от солнца или цвет объекта, но могут возникнуть трудности с определением расстояния до пушистых или очень тонких предметов. Именно поэтому высокотехнологичную котловку выполнить на нём будет затруднительно.

В отличие от [ультразвукового дальномера URM37](#), этот дальномер не обладает таким большим выбором интерфейсов и режимов работы. Но этот «недостаток» компенсируется простотой работы с ним. Если вы планируете использовать его с [Arduino](#), вы можете воспользоваться существующими библиотеками:

- [Ultrasonic](#) — самая популярная библиотека для HC-SR04.
- [NewPing](#) — отличается большей точностью и скоростью работы.

Контакты датчика можно соединить с [макетной платой](#) или Arduino [проводами «мама-папа»](#), а с [Troyka Shield](#) - через [провода «мама-мама»](#).

- Vcc — положительный контакт питания.
- Trig — цифровой вход. Для запуска измерения необходимо подать на этот вход логическую единицу на 10 мкс. Следующее измерение рекомендуется выполнять не ранее чем через 50 мс.
- Echo — цифровой выход. После завершения измерения на этот выход будет подана логическая единица на время, пропорциональное расстоянию до объекта.
- GND — отрицательный контакт питания.
- Напряжение питание: 5 В
- Потребление в режиме тишины: 2 мА
- Потребление при работе: 15 мА
- Диапазон расстояний: 2–400 см
- Эффективный угол наблюдения: 15°
- Рабочий угол наблюдения: 30°

## 18. Инфракрасный датчик препятствий



Выводом с сенсора является простой [бинарный цифровой сигнал](#): пока препятствия нет, датчик выдаёт логическую единицу, если препятствие есть — логический ноль.

Расстояние, на котором срабатывает сенсор, можно настроить с помощью триммера на задней части корпуса. Оно может варьироваться от 3 до 80 см.

Датчик размещается в пластиковом корпусе с резьбой. В комплекте с сенсором идёт пара пластмассовых шайб, которые навинчиваются на эту резьбу и могут зафиксировать датчик в [специальном креплении](#). Так вы можете надёжно установить его на одной из [платформ](#) или [шасси](#).

Датчик подключается к управляющей электронике через [3 провода](#).

Шлейф для подключения жёстко присоединён к датчику. Он имеет не совсем стандартную расцветку: красный — питание, зелёный — земля, жёлтый — сигнал.

**Внимание!** Распиновка питания этого сенсора может различаться. Перед включением модуля ознакомьтесь с [особенностями подключения модулей DFRobot](#).

Если вам необходимо знать точное значение расстояния, а не просто факт наличия препятствия, рассмотрите [инфракрасный дальномер](#) или [ультразвуковой дальномер](#).

- Расстояние срабатывания: 3–80 см (настраивается триммером)
- Напряжение питания: 5 В
- Потребляемый ток: 100 мА

### 19. Инфракрасный дальномер Sharp 10 – 80 см.



Выводом является [аналоговый сигнал](#) с уровнем напряжения, зависимым от расстояния до цели в установленном направлении.

Датчик подключается к управляющей электронике через [3 провода](#). При подключении к [Arduino](#) будет крайне удобно использовать [Troyka Shield](#). Шлейф для подключения включён в комплект.

**Внимание!** Распиновка питания этого сенсора может различаться. Перед включением модуля ознакомьтесь с [особенностями подключения модулей DFRobot](#).

Чтобы надёжно установить дальномер куда-либо, существует [специальное крепление](#).

- Напряжение питания: 4,5–5,5 В
- Потребляемый ток: 30–40 мА
- Диапазон расстояний: 10–80 см

Поскольку в основе работы устройства используется свет, сенсор плохо подходит для определения расстояния до светопоглощающих объектов. Для определения расстояний в таком окружении подойдёт ультразвуковой дальномер [URM37](#) или [HC-SR04](#).

Этот инфракрасный дальномер имеет небольшую мёртвую зону перед собой - 10 см. Если необходимо рассматривать препятствия на меньших расстояниях, а предельное расстояние не так важно, рассмотрите [дальномер для дистанций 4-30 см](#) той же линейки. Если же ваше устройство должно видеть дальше, обратите внимание на [дальномер для дистанций 20-150 см](#). Можно добиться большей гибкости, комбинируя датчики с различными диапазонами.

### 20. Инфракрасный дальномер Sharp (4-30 см)



Выводом является [аналоговый сигнал](#) с уровнем напряжения, зависимым от расстояния до цели в установленном направлении.

Датчик подключается к управляющей электронике через [3 провода](#). При подключении к [Arduino](#) будет крайне удобно использовать [Troyka Shield](#). Шлейф для подключения включён в комплект.

**Внимание!** Распиновка питания этого сенсора может различаться. Перед включением модуля ознакомьтесь с [особенностями подключения модулей DFRobot](#).

Чтобы надёжно установить дальномер куда-либо, существует [специальное крепление](#).

- Напряжение питания: 4,5–5,5 В
- Потребляемый ток: 30–40 мА
- Диапазон расстояний: 4–30 см

Поскольку в основе работы устройства используется свет, сенсор плохо подходит для определения расстояния до светопоглощающих объектов. Для определения расстояний в таком окружении подойдёт ультразвуковой дальномер [URM37](#) или [HC-SR04](#).

Этот инфракрасный дальномер хорошо подходит для использования на лёгких роботах, т.к. имеет самую маленькую среди инфракрасных сенсоров расстояния мёртвую зону перед собой - всего 4 см. Если вам необходимо измерять расстояние в большем диапазоне и не так важна ближняя мёртвая зона, рассмотрите [дальномер для дистанций 20-150 см](#) или [10-80 см](#) из той же линейки. Можно добиться большей гибкости, комбинируя датчики с различными диапазонами.

Учебное издание

При описании функционала контроллера использованы материалы сайта [amperka.ru](http://amperka.ru)

Руководство для конструкторов «СКАРТ»

Филимонов А.С.

«Лаборатория Интеллектуальных Технологий ЛИНТЕХ», Москва, 2016г.

Сайты: [www.robotolab.ru](http://www.robotolab.ru)

[www.skartshop.ru](http://www.skartshop.ru)

E-mail: [info@robotolab.ru](mailto:info@robotolab.ru)

Телефон: +7 (499) 110-14-40