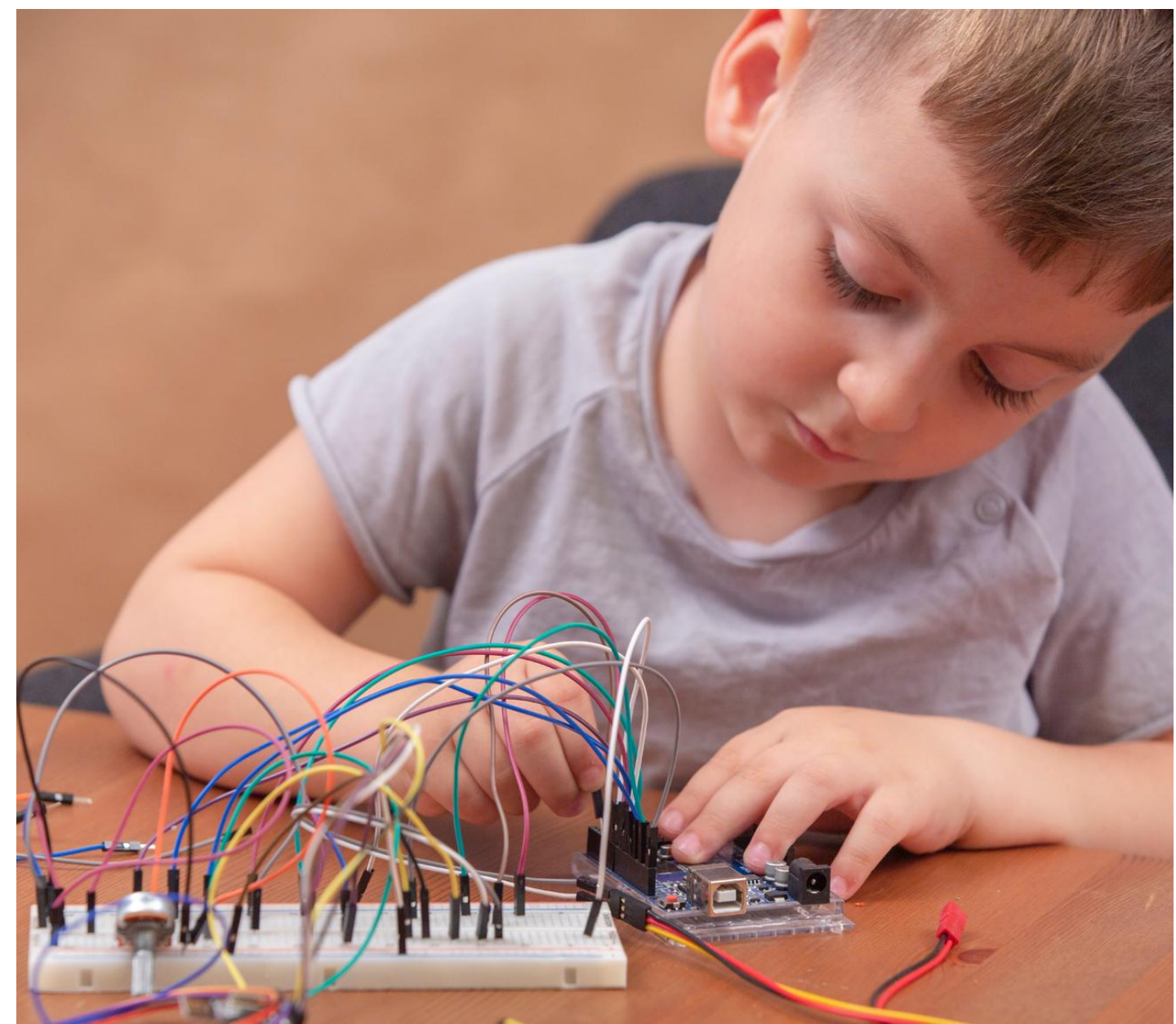
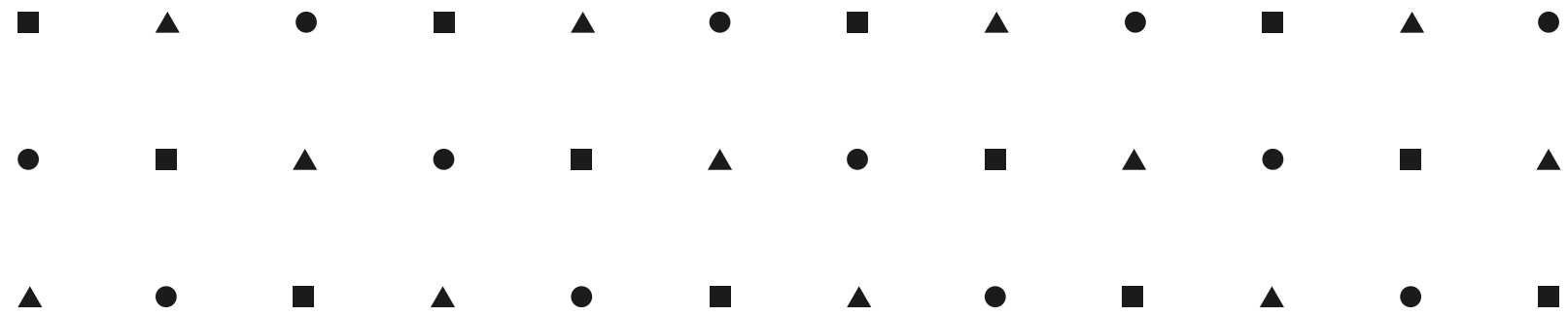




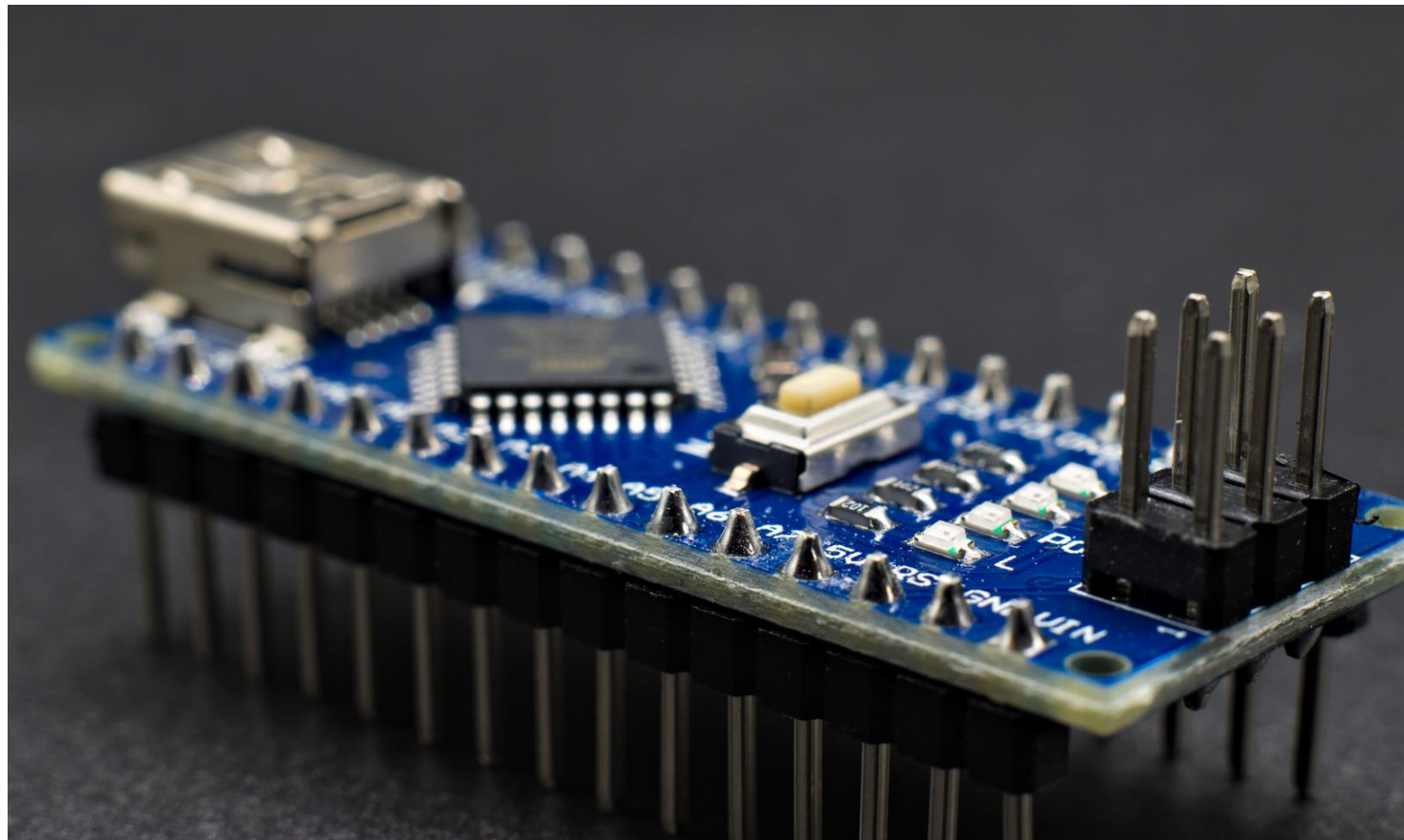
Введение в Arduino и mBlock





Arduino Nano

Программно-аппаратный модуль



Эта отладочная плата небольшого размера, которая входит в тройку лидеров по популярности среди радиолюбителей-программистов. Несмотря на свой скромный размер, она практически ничем не уступает на шумевшей Arduino Uno по функционалу и может использоваться в малогабаритных проектах

Для нас Arduino nano будет мозгом наших проектов, а в дальнейшем и главным модулем нашего робота. Плату мы будем прошивать с помощью программы mBlock, для выполнения нужных нам функций и действий.

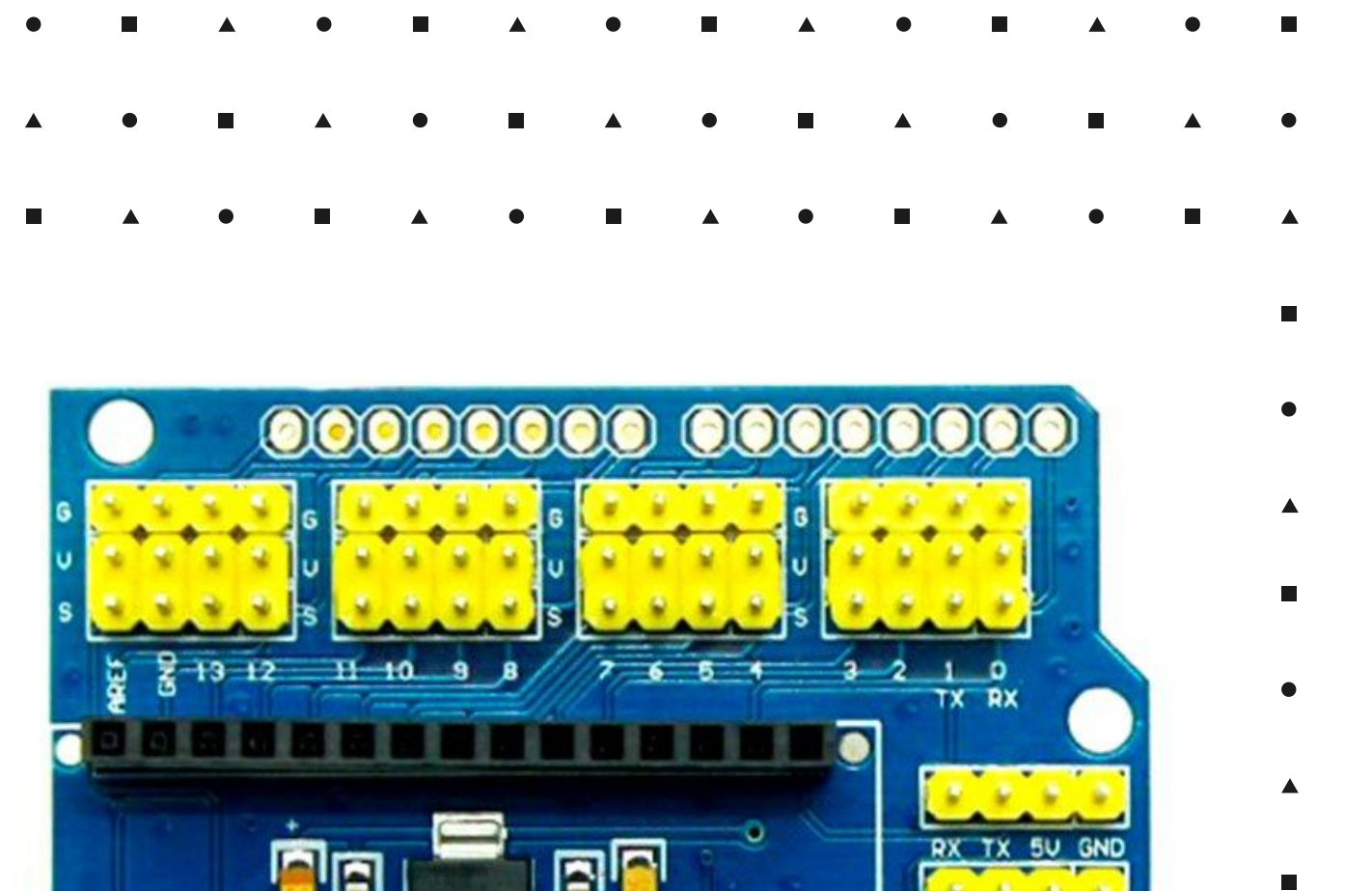
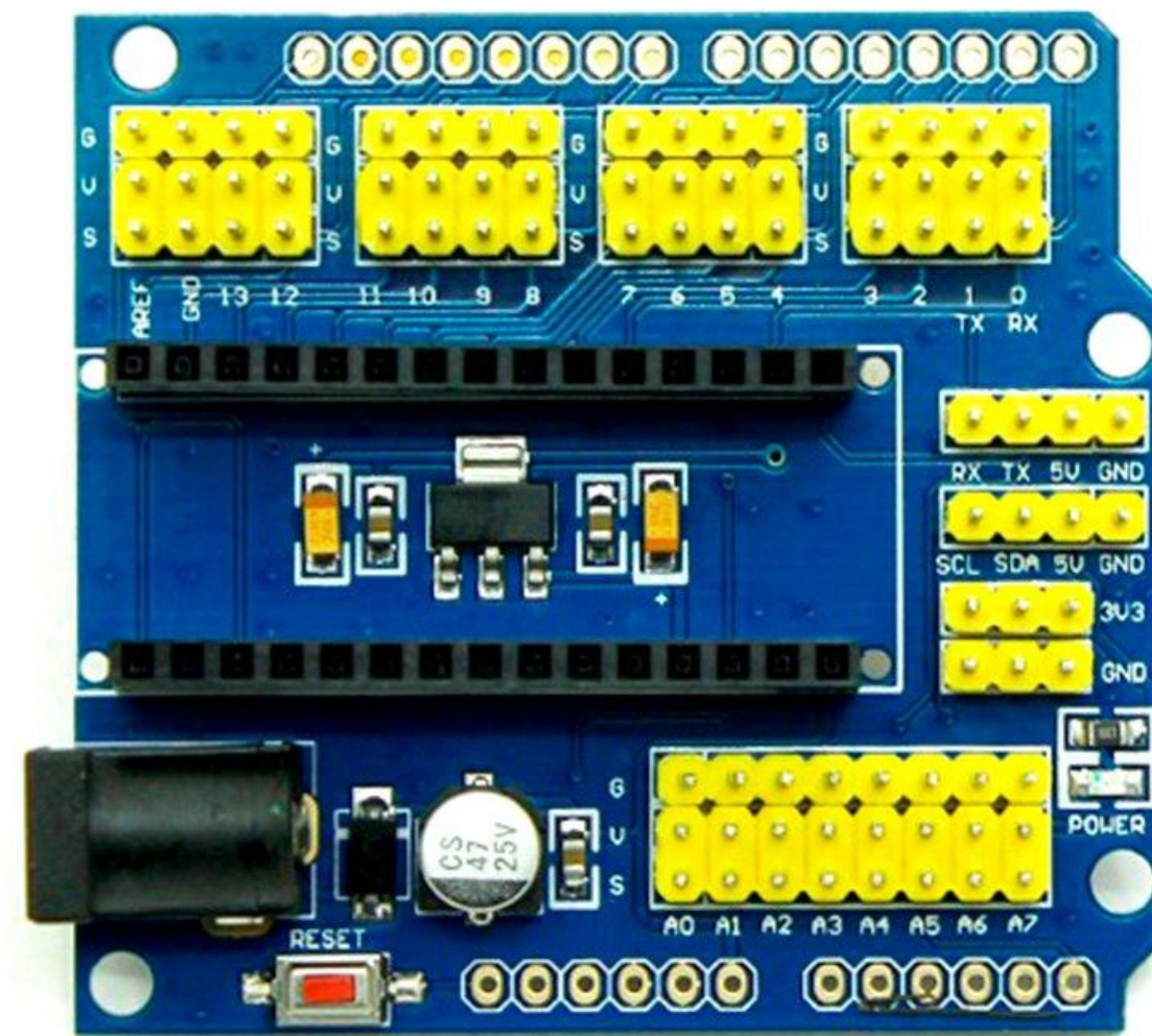


Плата расширения Arduino nano I/O

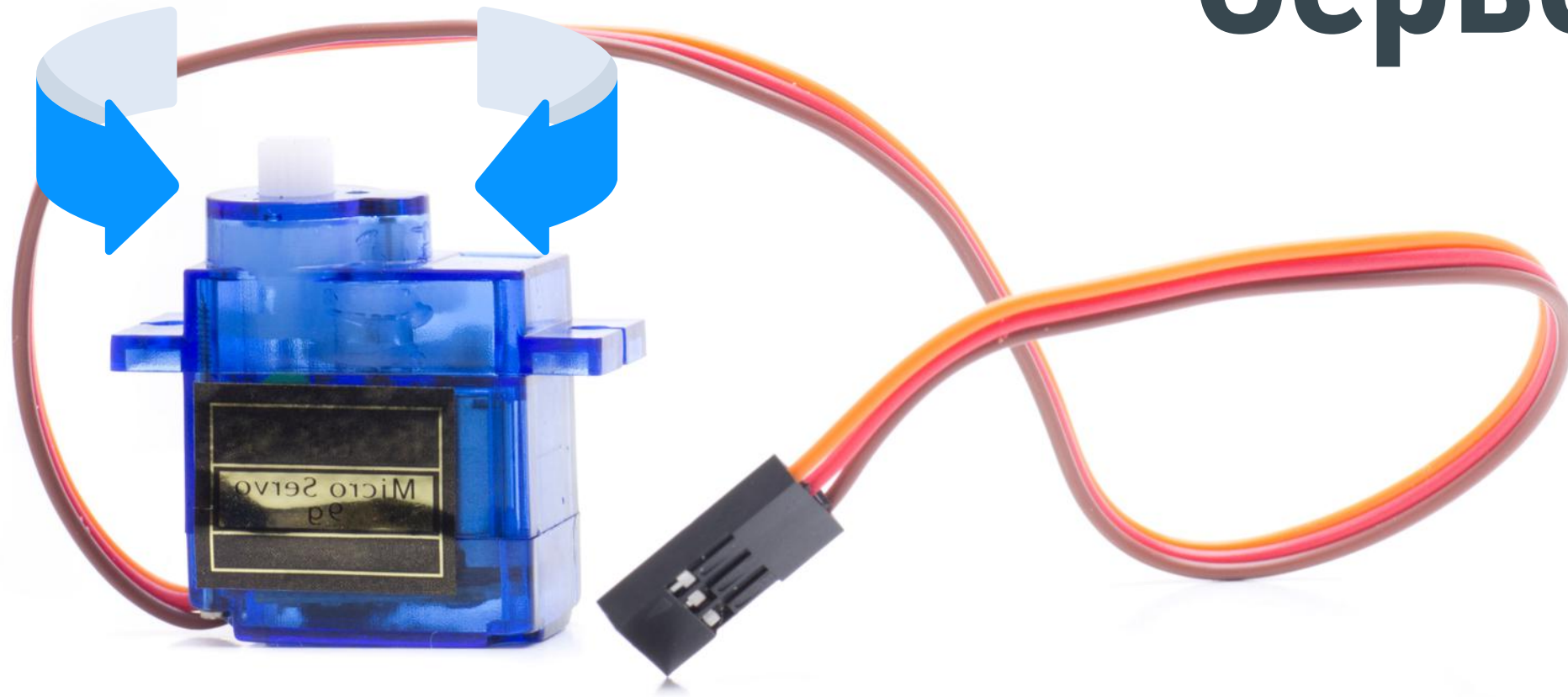
Упрощенное подключение

Добавляя к Arduino Nano специальную плату расширения - "шилд", мы значительно упрощаем работу с контроллером. Зачем тратить время, чтобы проектировать, припаивать и отлаживать то, что можно взять уже в собранном виде, и сразу начать использовать?

Шилд Arduino Nano I/O идеально подходит для работы со всеми электронными компонентами из набора. Модуль предназначен для увеличения количества контактных групп. Одно из важных преимуществ - на разъемы кроме информационных сигналов выведены линии питания (V) и линии земли (G)



Сервопривод SG-90



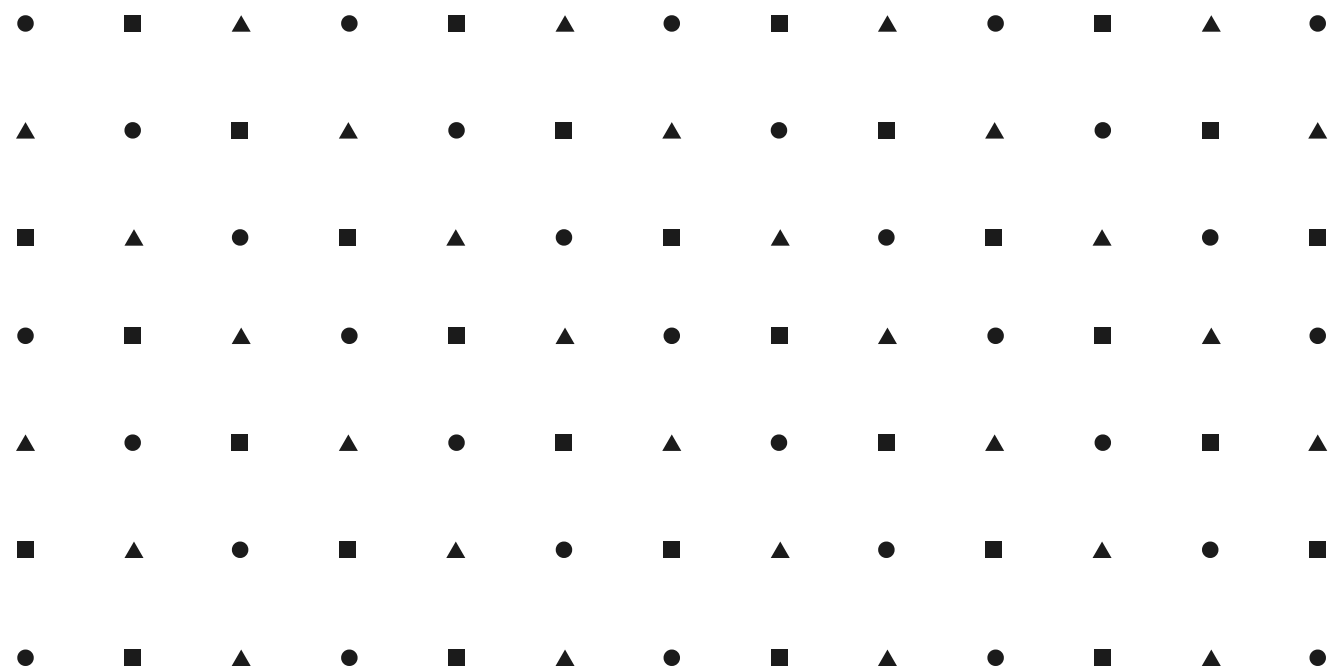
Главный механизм

Под сервоприводом чаще всего понимают механизм с электромотором, которому можно сообщить с Arduino nano определённый сигнал, устанавливающий его в заданный угол и удерживающий это положение

Мы будем включать и выключать электромотор, вращая выходной вал – конечную шестерню сервопривода, к которой можно прикрепить что-то, чем мы хотим управлять

Сервоприводы нашего робота имеют угол поворота от 0 до 180. С помощью mBlock мы будем плавно изменять значения поворота, тем самым приводить нашего робота в движение

Максимальная нагрузка на главный вал составляет 9 грамм



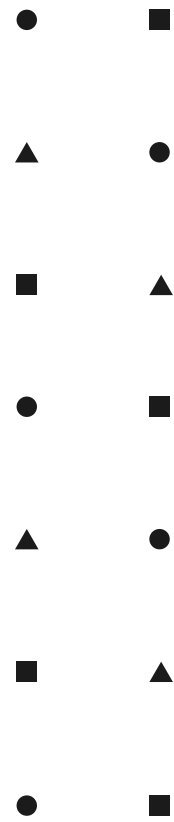
Пьезодинамик



Звуковые сигналы робота

Зуммер, пьезопищалка – все это названия одного устройства. Данный модуль используется для звукового оповещения в тех устройствах и системах, для функционирования которых в обязательном порядке нужен звуковой сигнал. Зуммеры используют в различной бытовой технике и игрушках, использующих электронные платы. Пьезопищалки преобразуют команды, основанные на двухбитной системе счисления 1 и 0, в звуковые сигналы

Принцип действия зуммера основан на открытом братьями Кюри в конце девятнадцатого века пьезоэлектрическом эффекте. Согласно ему, при подаче электричества на зуммер он начинает деформироваться. При этом происходят удары о металлическую пластинку, которая и производит “шум” нужной частоты

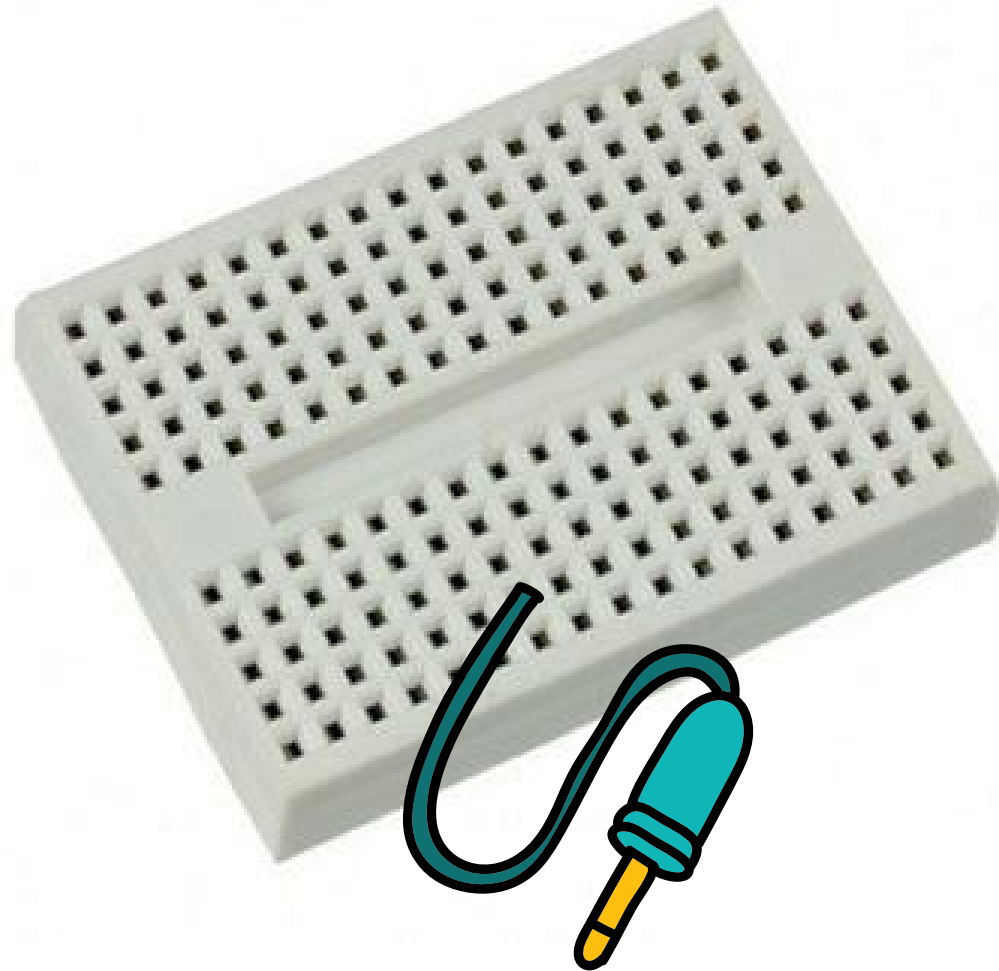
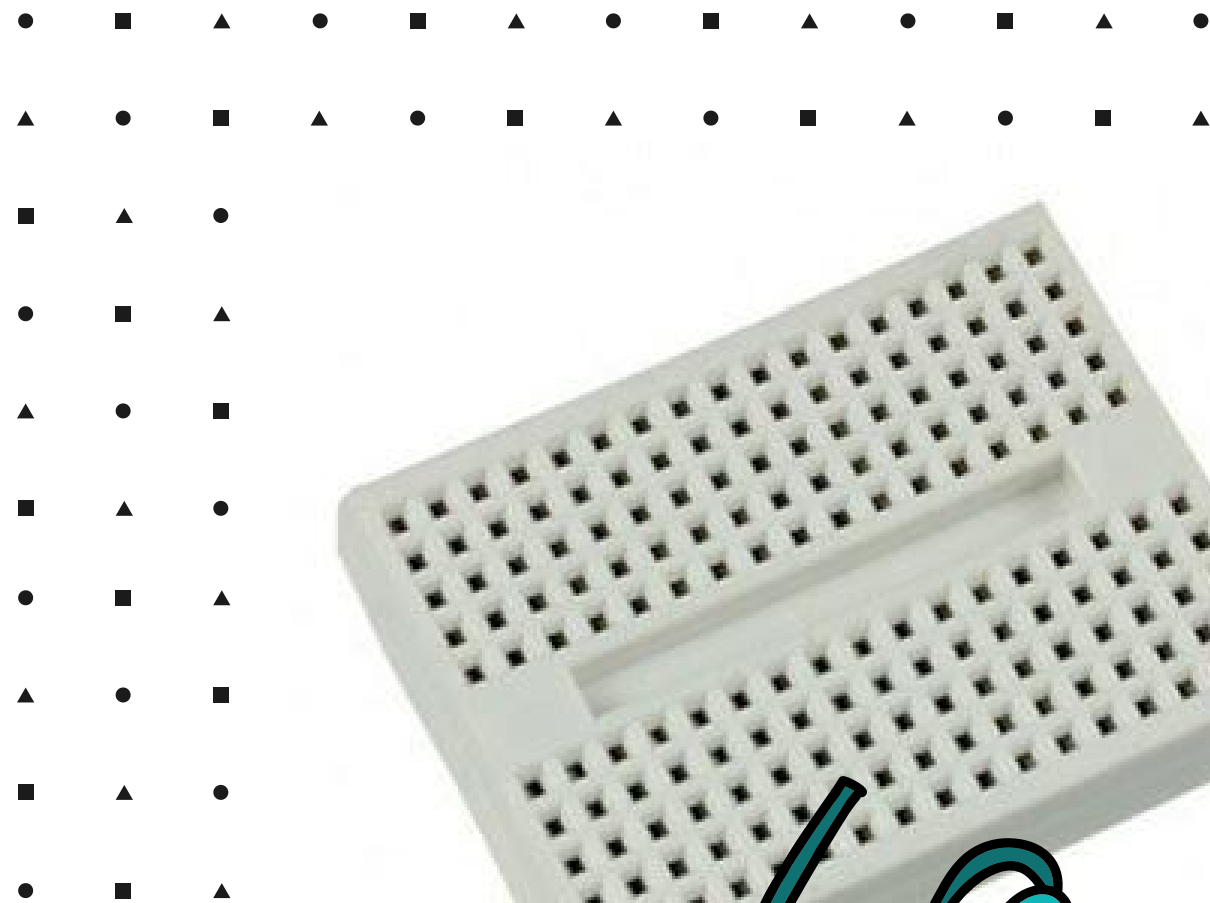


Макетная плата

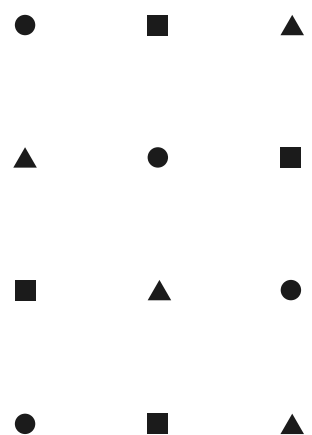
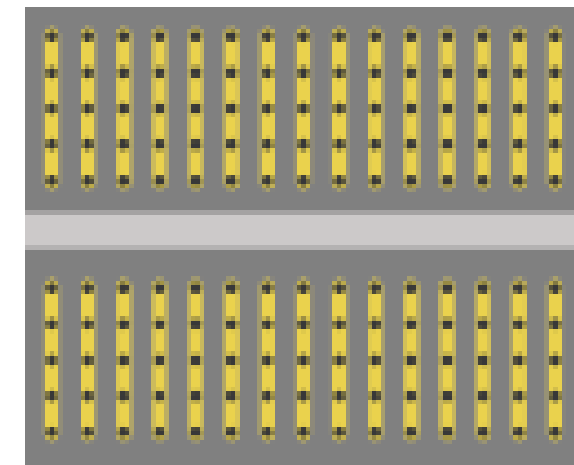
Беспаячная макетная плата на 170 точек

Так называемая breadboard или макетная плата — незаменимая вещь для экспериментов с электроникой. Она позволяет быстро, удобно, без паяльника собирать электрические схемы

При создании чего-то нового, в процессе проб и ошибок почти всегда приходится несколько раз корректировать схему. Если все детали сразу соединять пайкой, изменения становятся проблемой. Макета позволяет не думать об этом и вносить сколько угодно изменений



Контактные площадки макетной платы

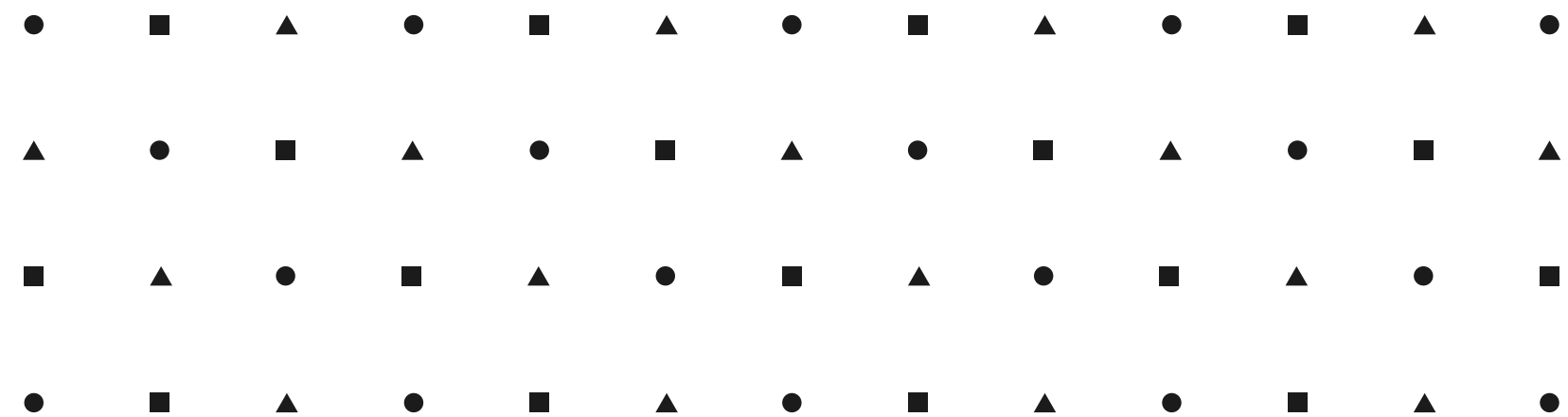
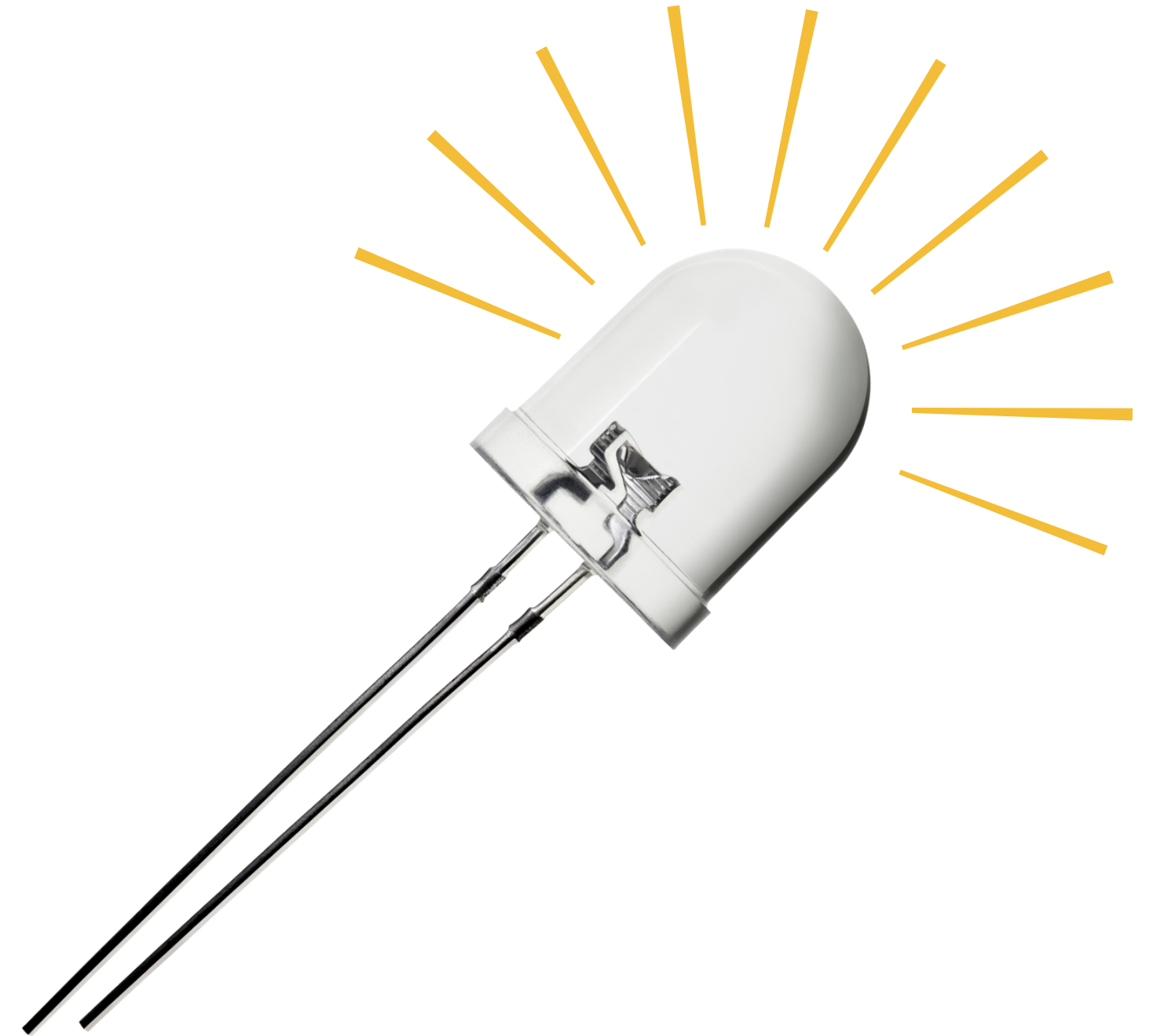


Светодиод

Световая индикация

Светодиод (англ. Light Emitting Diode или просто LED) – энергоэффективная, надёжная, долговечная «лампочка» Светодиод – вид диода, который светится, когда через него проходит ток от анода (+) к катоду (-).

Светодиод имеет разные по длине металлические ножки, именно по ним определяется полярность при его подключении: длинная отвечает за сигнальный вход (+), а короткая будет подключаться к земле (GND) нашей Arduino Nano



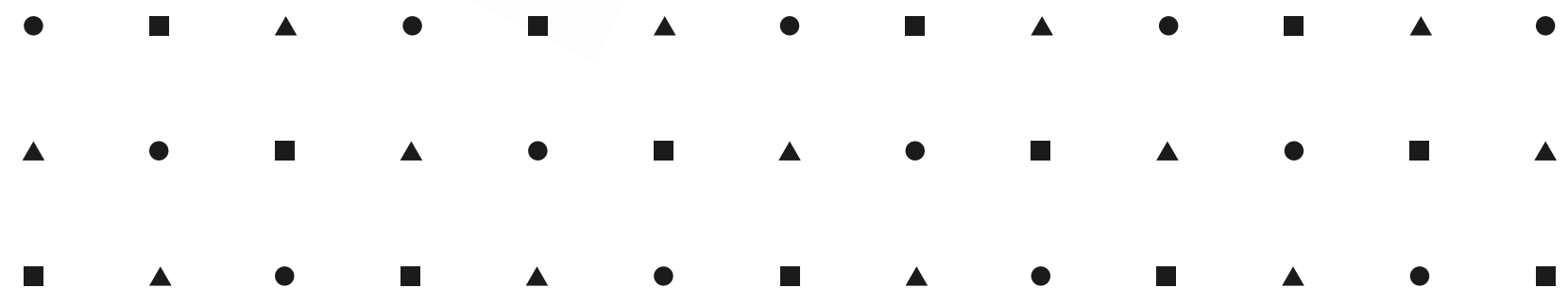
Набор проводов

При прототипировании электронных систем на макетных платах с участием Arduino или какой-либо другой платформы невозможно обойтись без соединительных проводов, по которым будут передаваться сигналы, и протекать питание ко всем датчикам и модулям. Наши провода имеют на обоих концах прочные металлические штыри, которые очень удобно вставлять в отверстия макетных плат



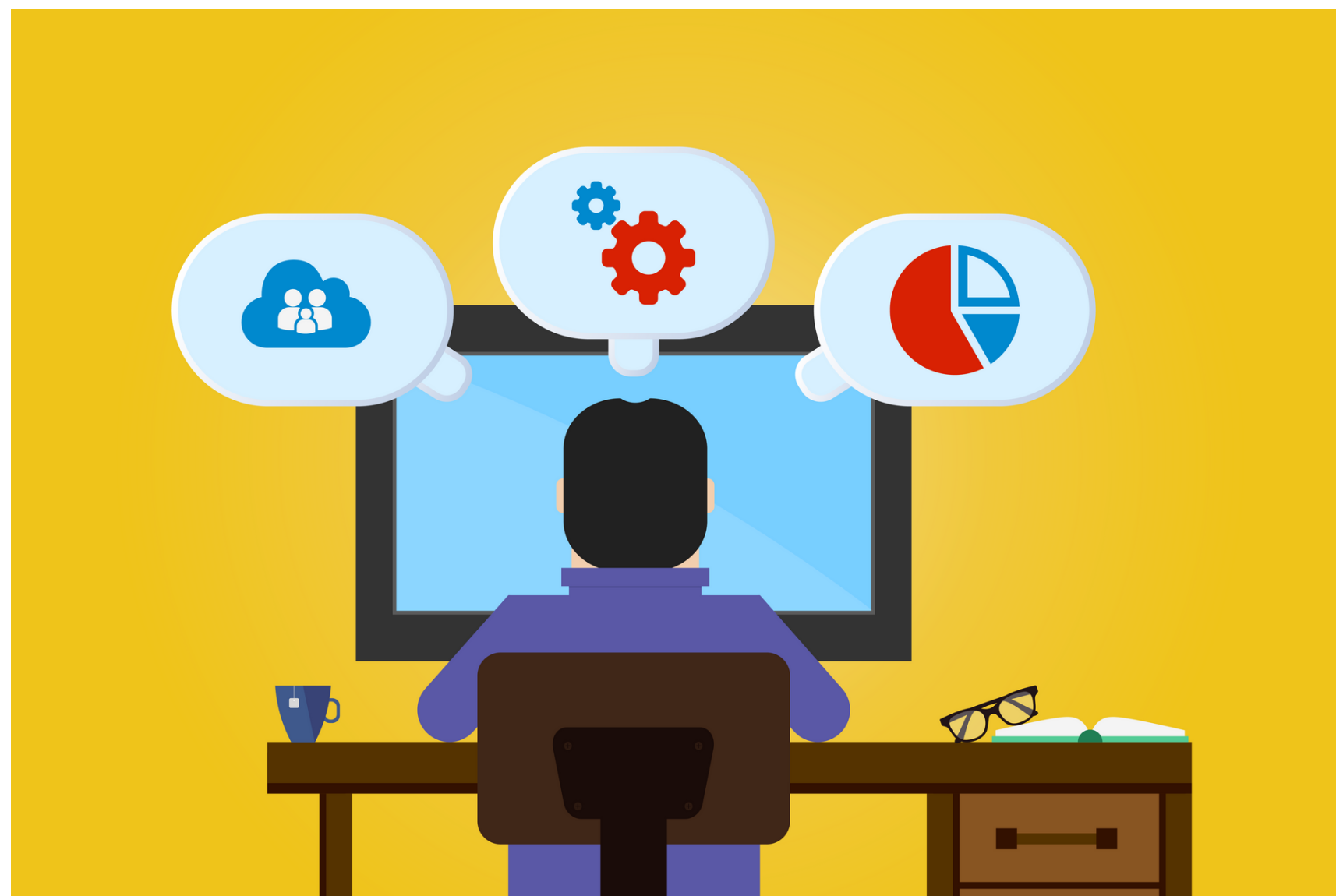
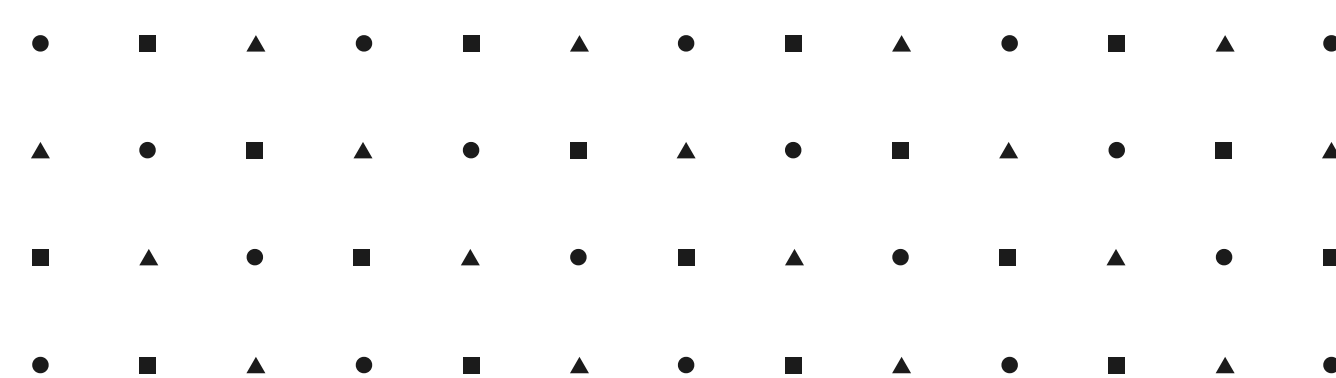
Батарейный отсек

Батарейный отсек на четыре элемента AA - это практически готовый Power Bank для нашего робота. Этот батарейный бокс полностью прячет в своем корпусе 4 батарейки AA и имеет механический выключатель питания. Аккумуляторный блок можно использовать для автономного питания различных устройств на Ардуино, роботов, RC моделей и других устройств требуемых автономного батарейного питания

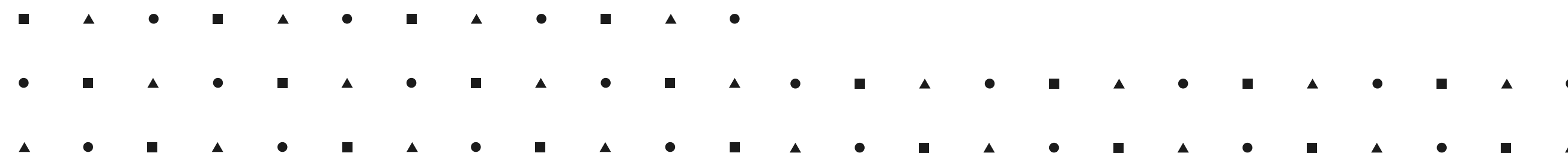


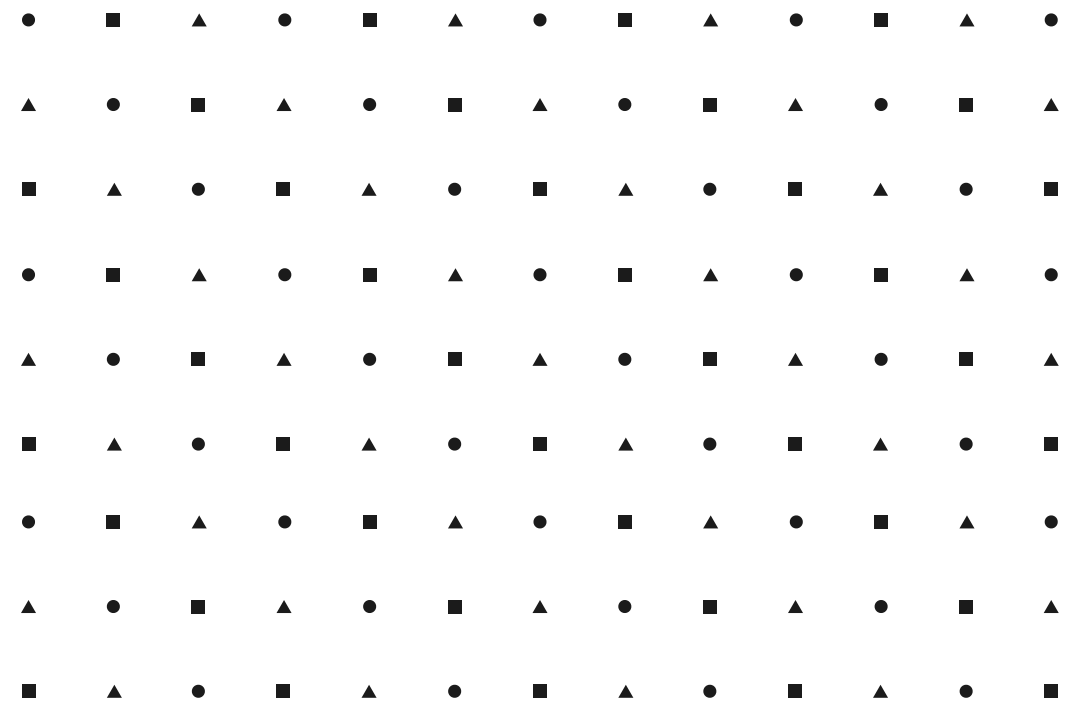
Этапы обучения

План работы



1. Установка mBlock
2. Сборка прототипов схем и программирование базовых проектов Arduino и mBlock
3. Сборка квадропода
4. Калибровка квадропода
5. Программирование квадропода





Блоковая среда программирования

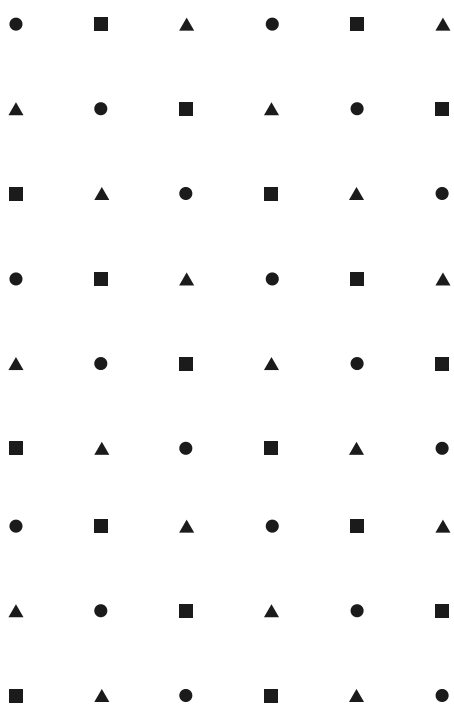
Сегодня она активно используется как для MakeBlock, так и для самого широкого спектра плат Ардуино. Программа русифицирована и базируется на Scratch 3, позволяя с ее помощью создавать собственных роботов и технические проекты произвольной конфигурации на Ардуино

Преимущества mBlock

- поддержка большинства плат Ардуино
- возможность использования собственных блоков с возможностями рекурсии
- простота использования библиотек и наработок в собственных проектах
- удобство создания своих программ; можно изучать синтаксис языка программирования на базе mBlock



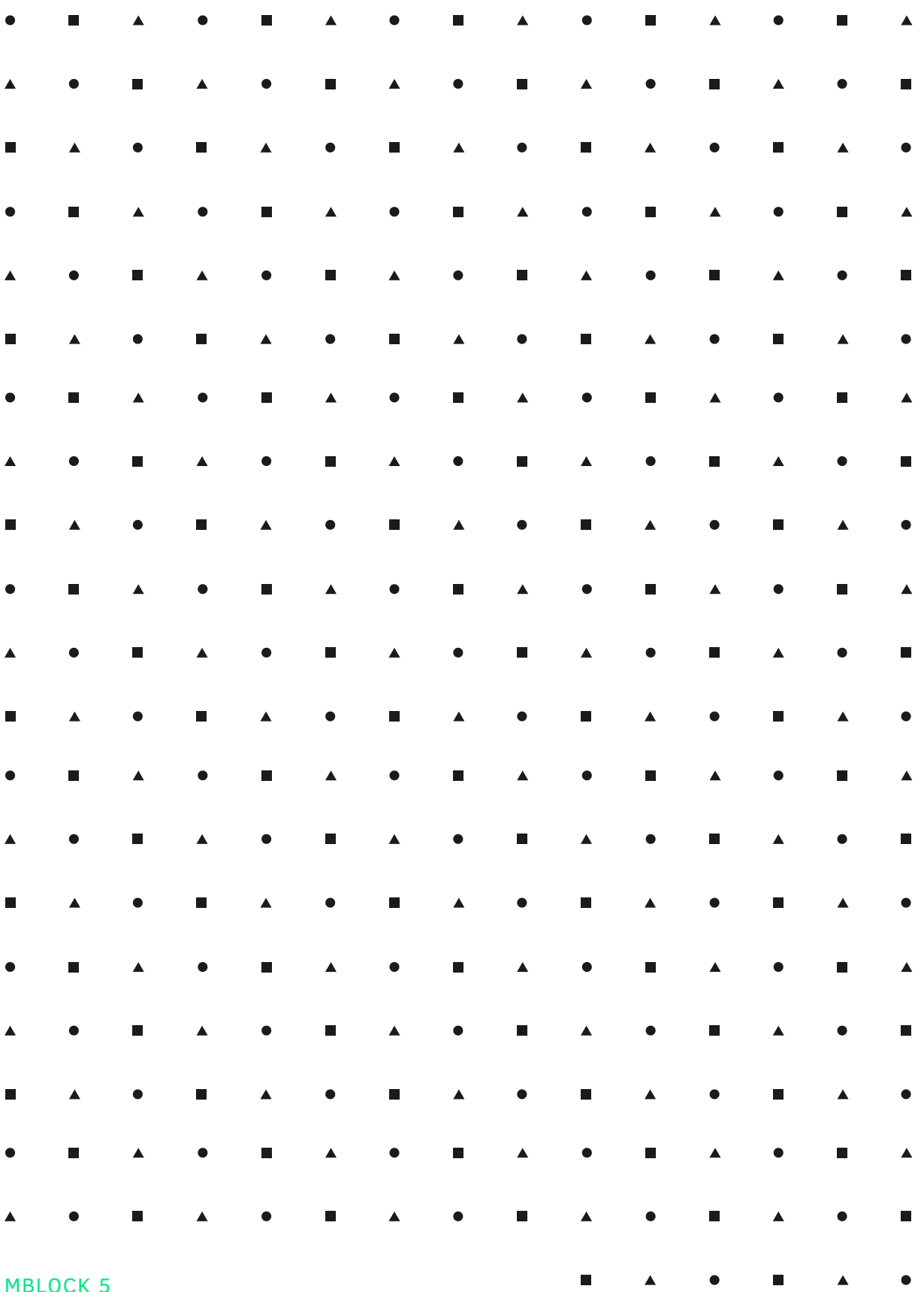
mBlock 5



Установка mBlock


Ссылка для скачивания и установки:

<https://www.mblock.cc/en-us/download>




Download mBlock

One-stop coding platform tailored to coding education, trusted by 15 million educators, and learners



mBlock web version
Chrome browser recommended >>
Support Windows/Mac/Linux/Chromebook


Code with blocks Code with Python




mBlock PC version
Version: V5.2.0
Released: 2020.01.22
Released log >> Previous version >>

Download for Windows Download for Mac


Win7 or Win10 (64-bit recommended) macOS 10.12+



mBlock mobile app
Learn coding in phones and tablets



Android
Android 6.0 +
(ARM-based devices only. X86
Android not supported)

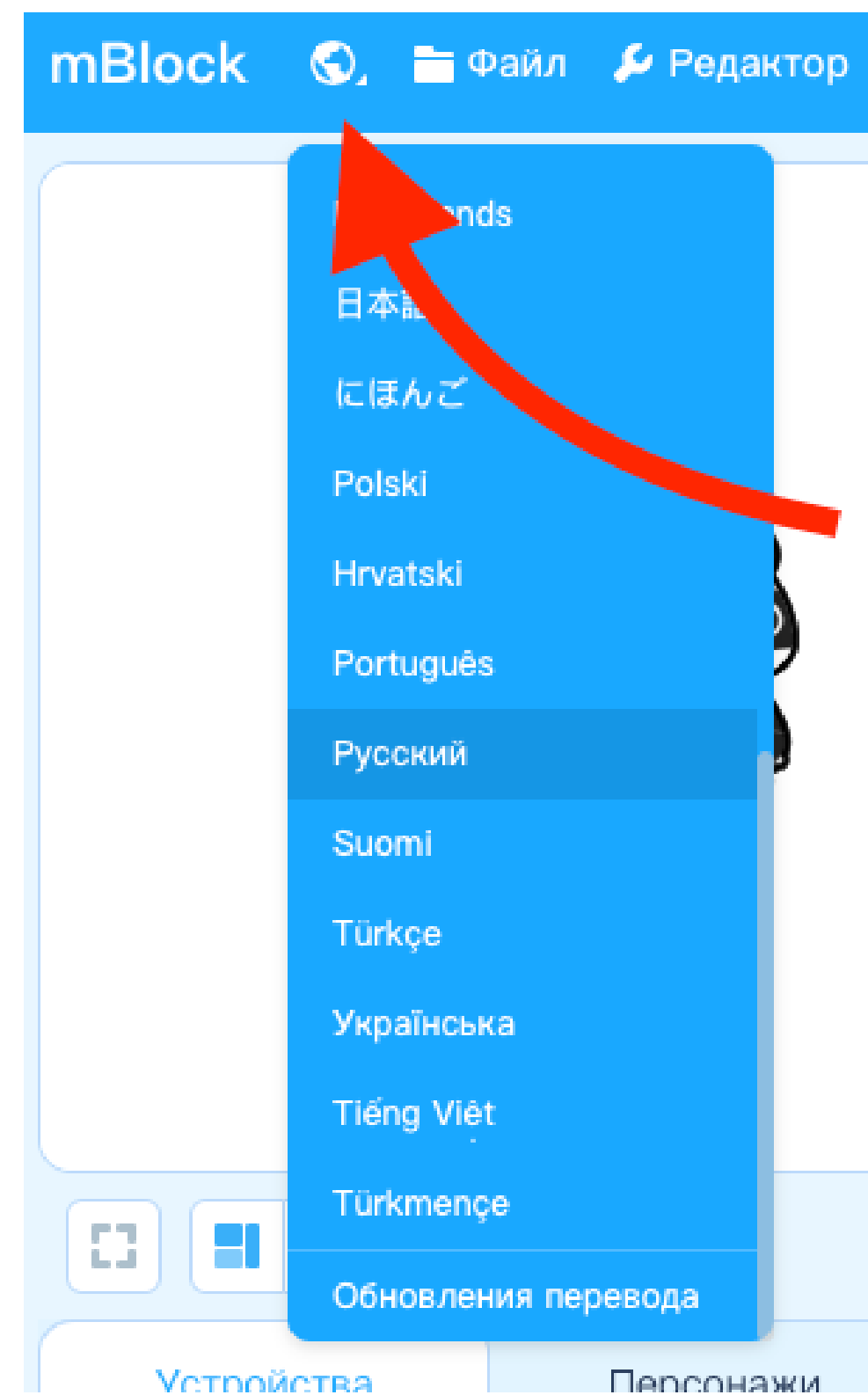


iOS
iOS 10.0 +

Шаг 1

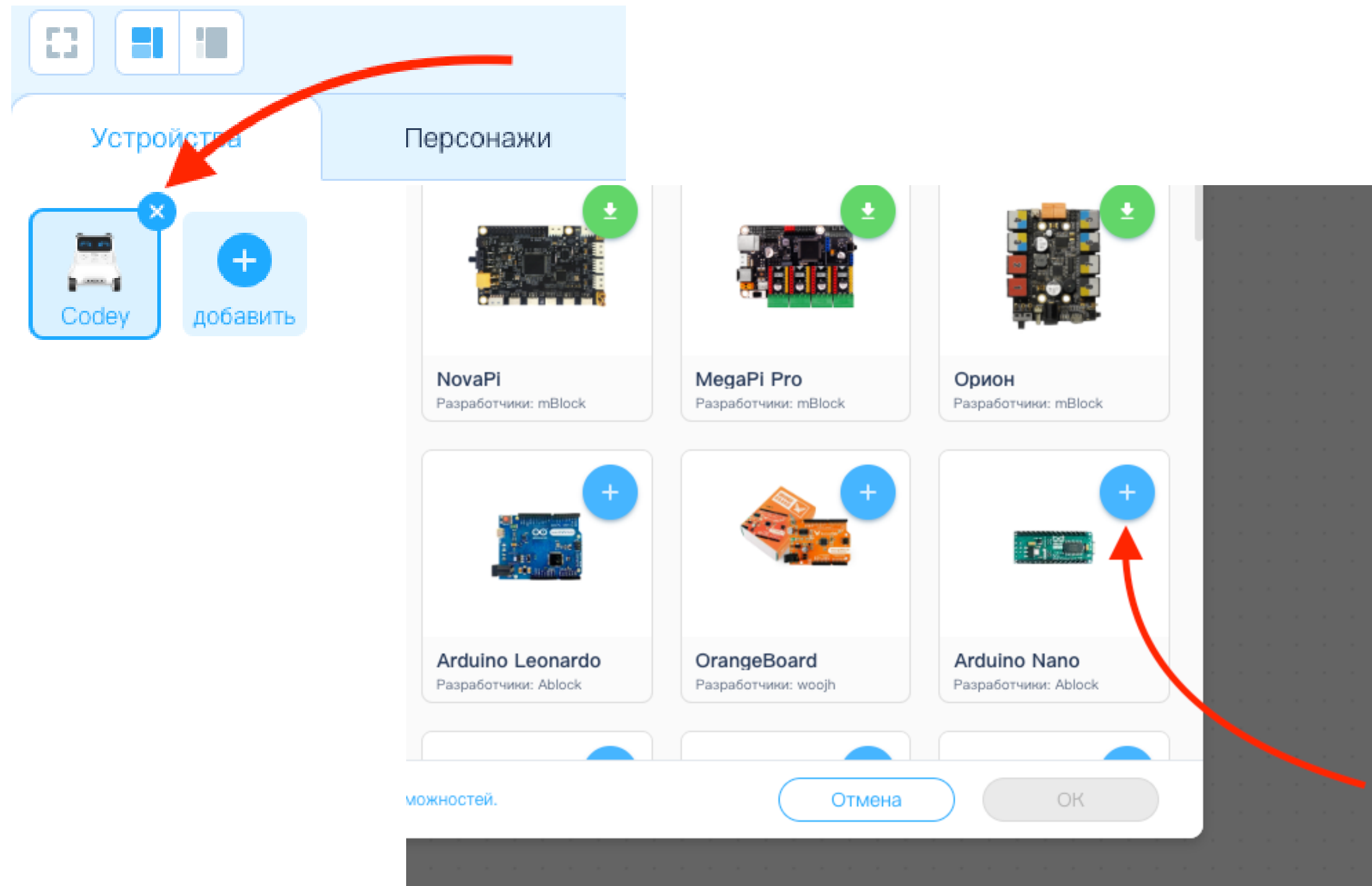
Настройка Mblock

Выберем русский язык



Программирование Arduino

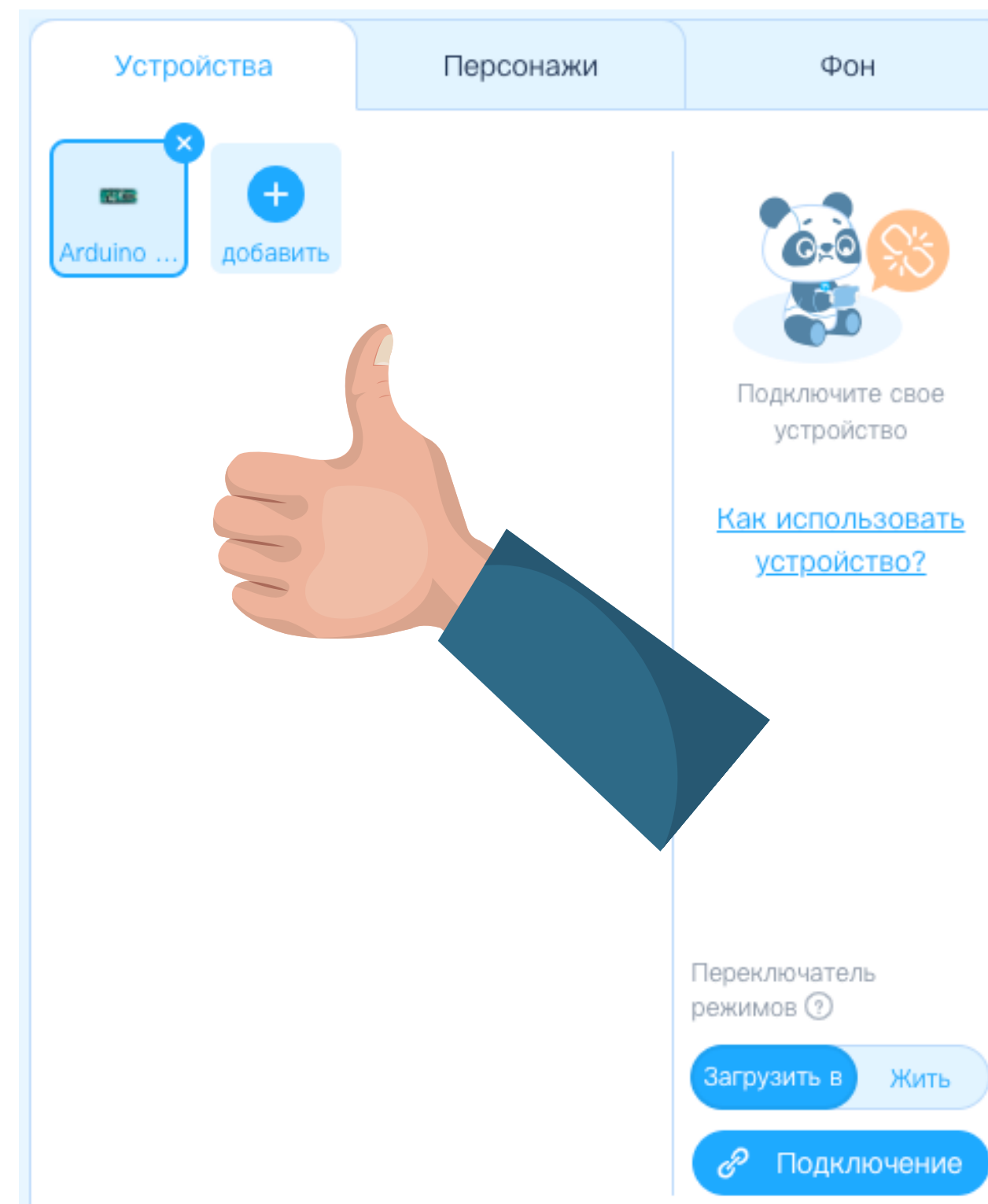
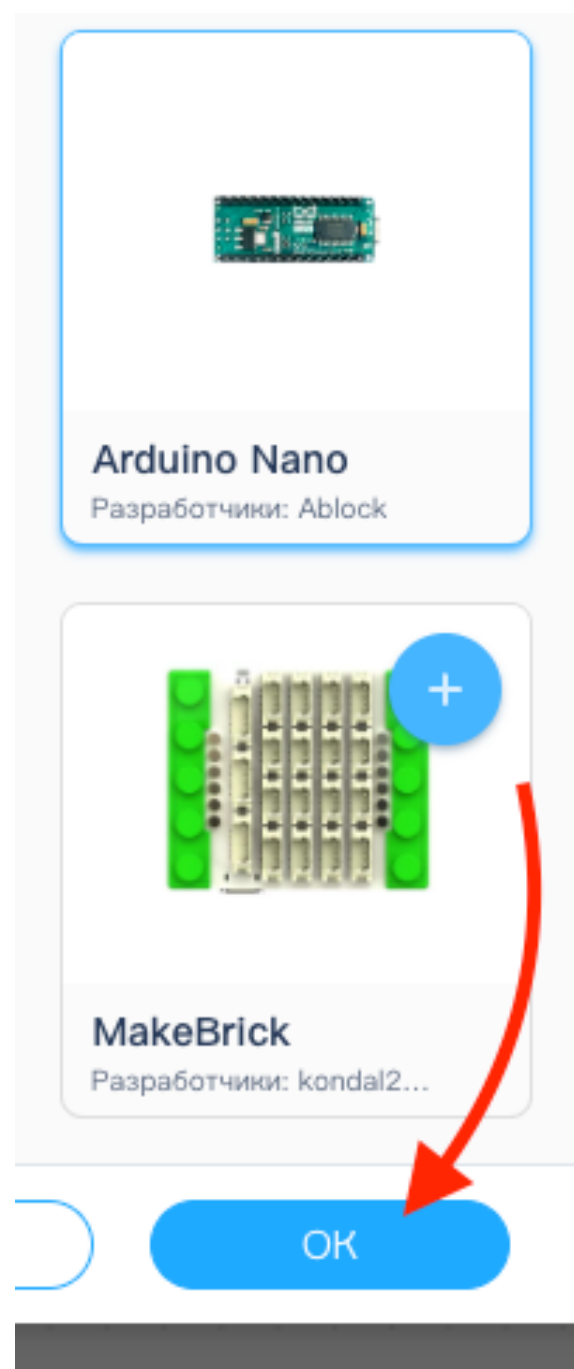
Удалим Codey, добавим Arduino Nano



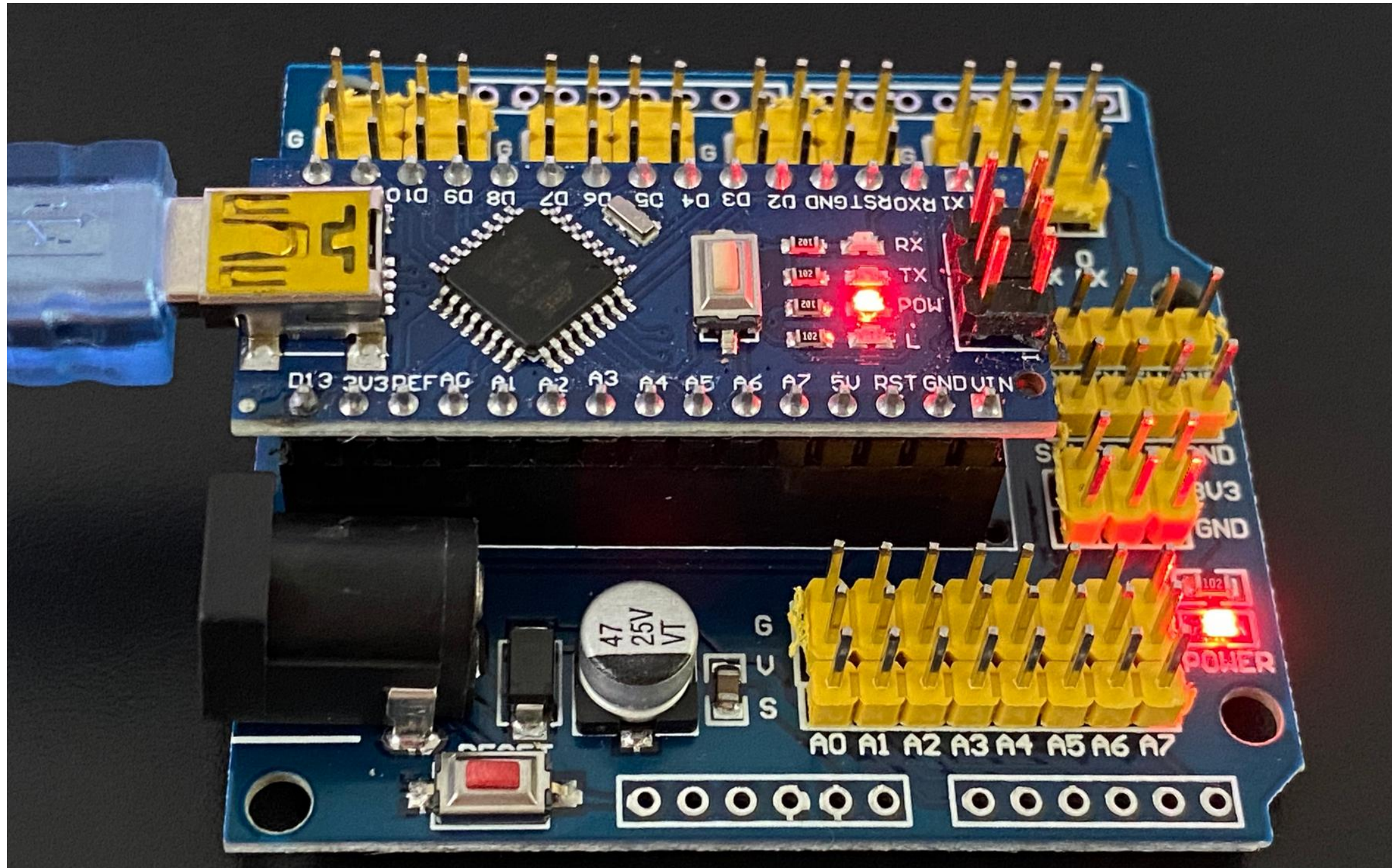
Шаг 2

Шаг 3

Выбрав плату с которой мы будем работать, нажимаем ОК. Она должна появиться во вкладке устройства



Установим Arduino Nano к плате расширения, как показано на рисунке, подключив её кабелем USB-mini из комплекта к компьютеру



Шаг 4



Шаг 5

Подключите свое устройство

[Как использовать устройство?](#)

Переключатель режимов (?)

Загрузить в Жить

Подключение

Перем
Мс
Блс
расш

Соединим плату с компьютером
Название порта может отличаться

USB

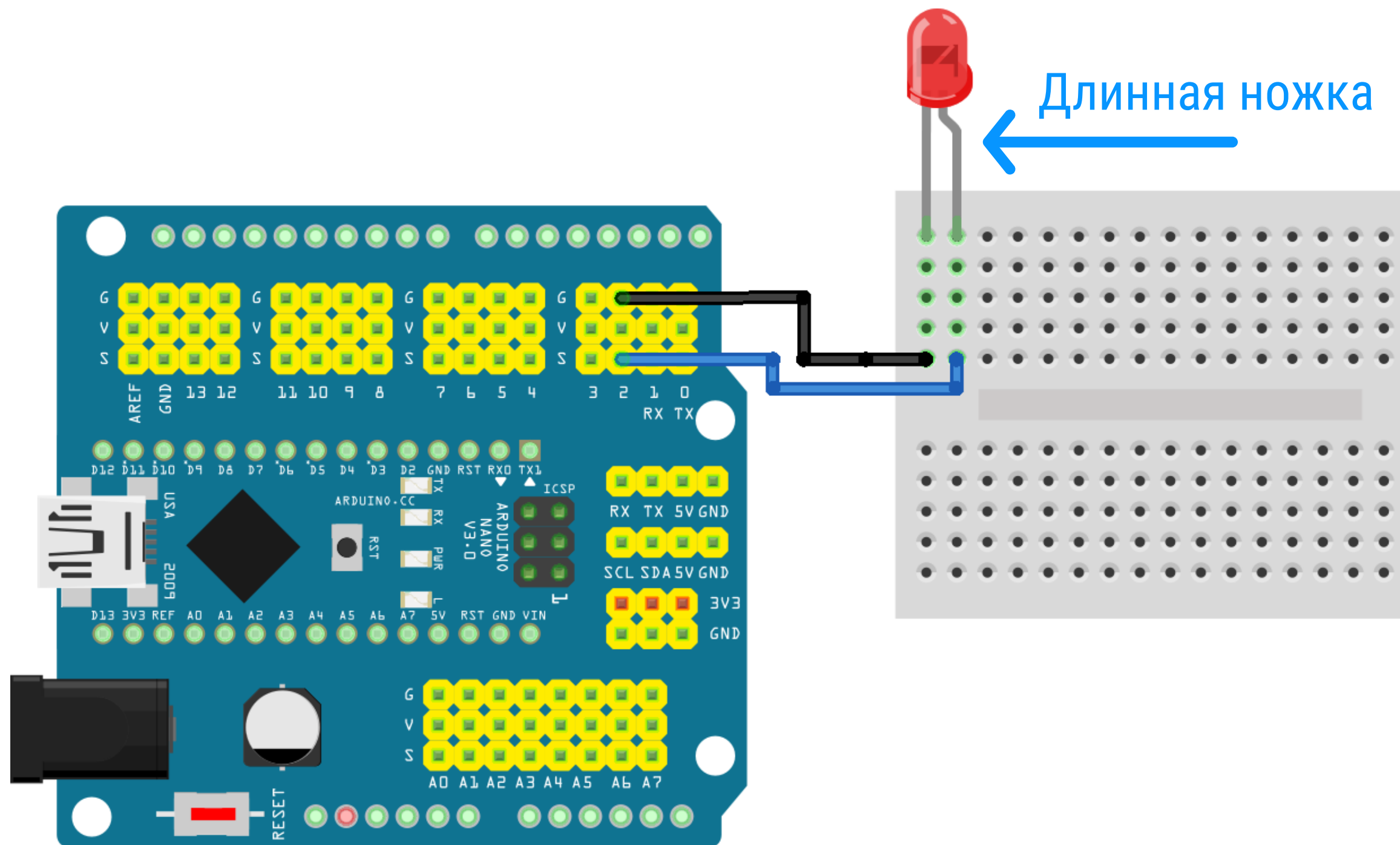
Показать все подключенные устройства

/dev/tty.usbserial-1420

Подключение

- Убедитесь, что кабель USB правильно подключен к устройству.
- Убедитесь, что подключенное устройство включено.
- Только одно устройство может быть подключено одновременно в этой версии. Таким образом, подключение этого устройства приведет к разъединению предыдущего.

Соберем схему с применением макетной платы, пары соединительных проводов, светодиодом и Arduino Nano. Короткую ногу светодиода соединим с GND платы, а длинную с 2 портом

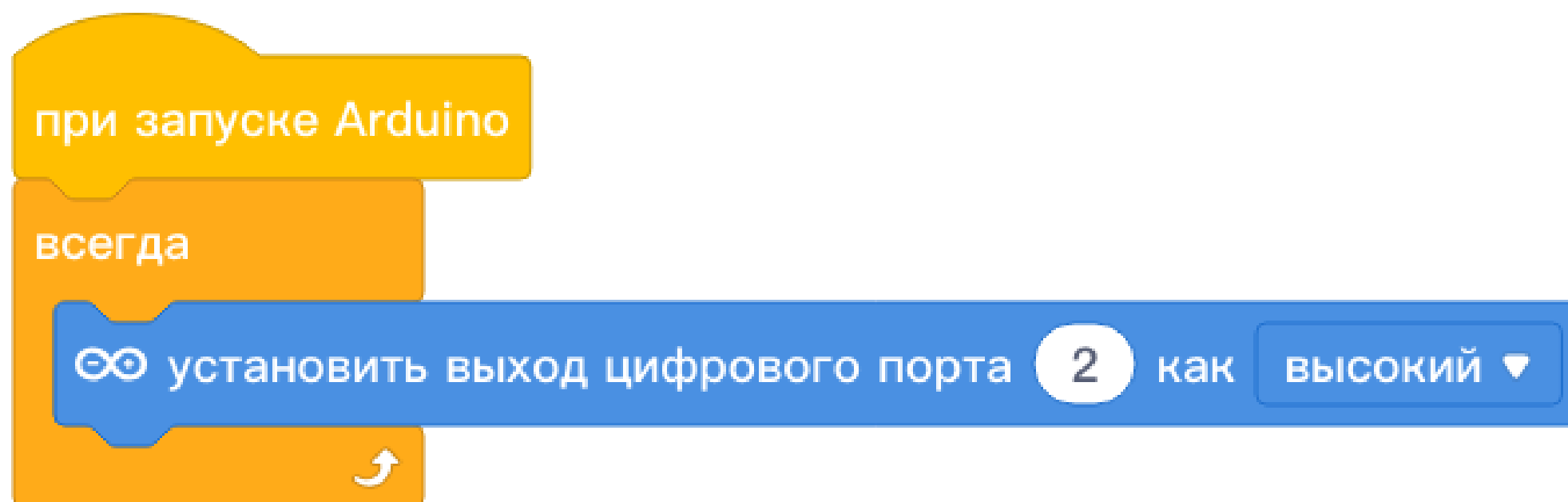


Шаг 6

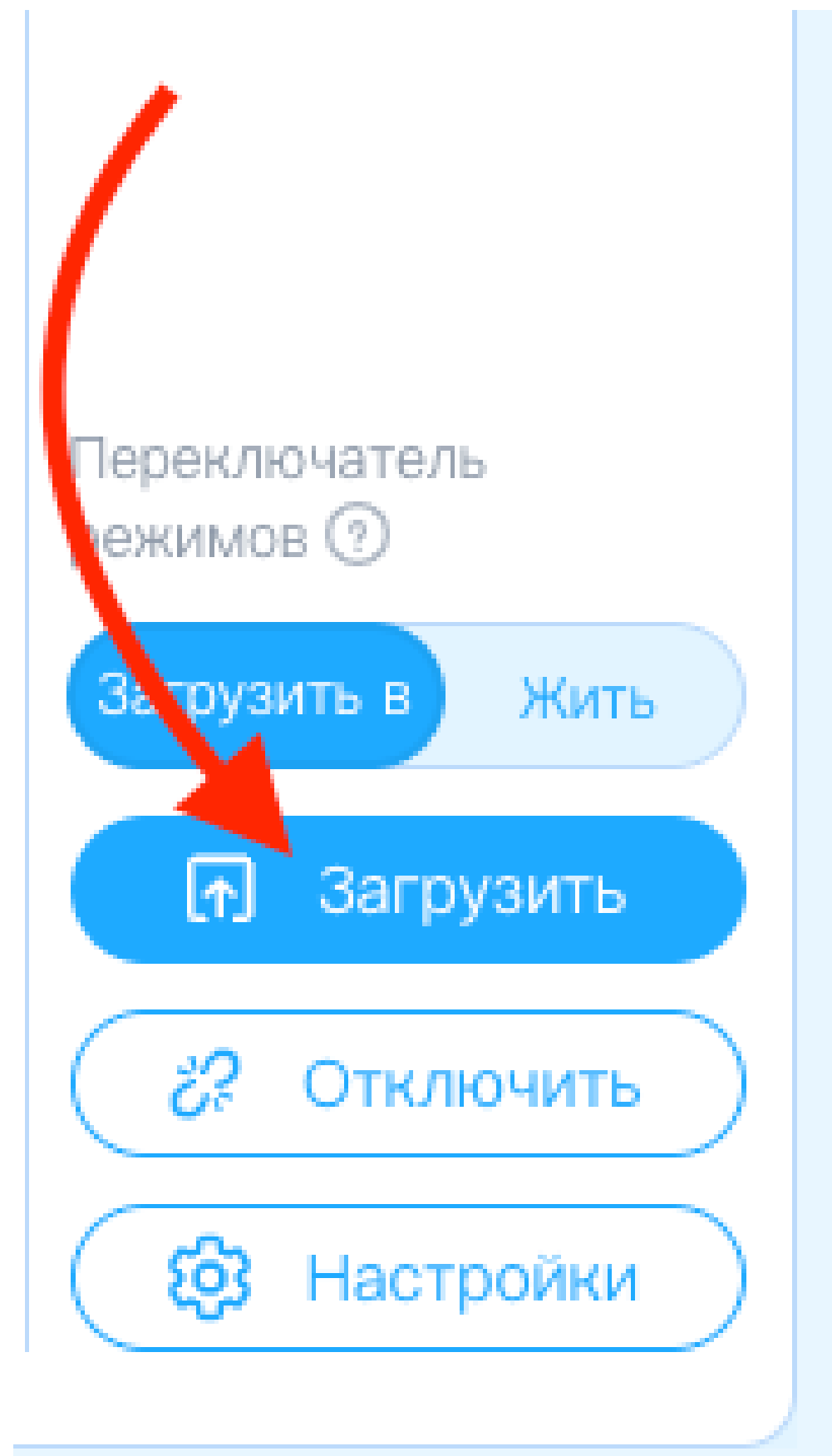


Шаг 7

Напишем блоковый код, который включит светодиод, который мы подключили к 2 порту Arduino Nano



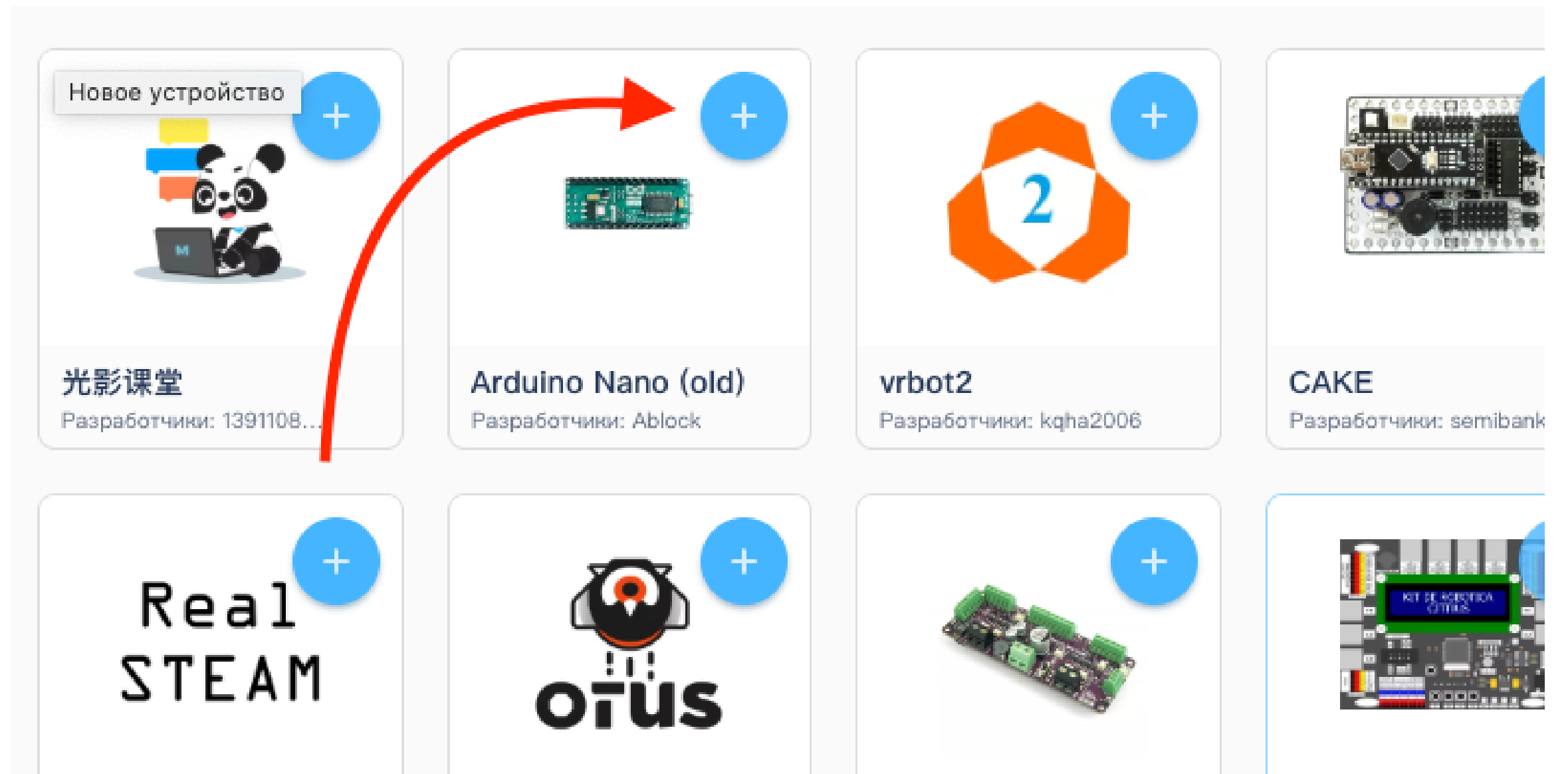
Для загрузки нажимаем на кнопку "Загрузить"



Шаг 8



Может быть такое, что ваша Arduino Nano не подключается к компьютеру, или не загружается код - тогда следует выбрать в качестве используемого устройства другую версию загрузчика Arduino nano (Old)



Шаг 9

Теперь напишем код для мигания светодиодом, с частотой 1 секунда.

при запуске Arduino

всегда

установить выход цифрового порта 2 как ВЫСОКИЙ ▼

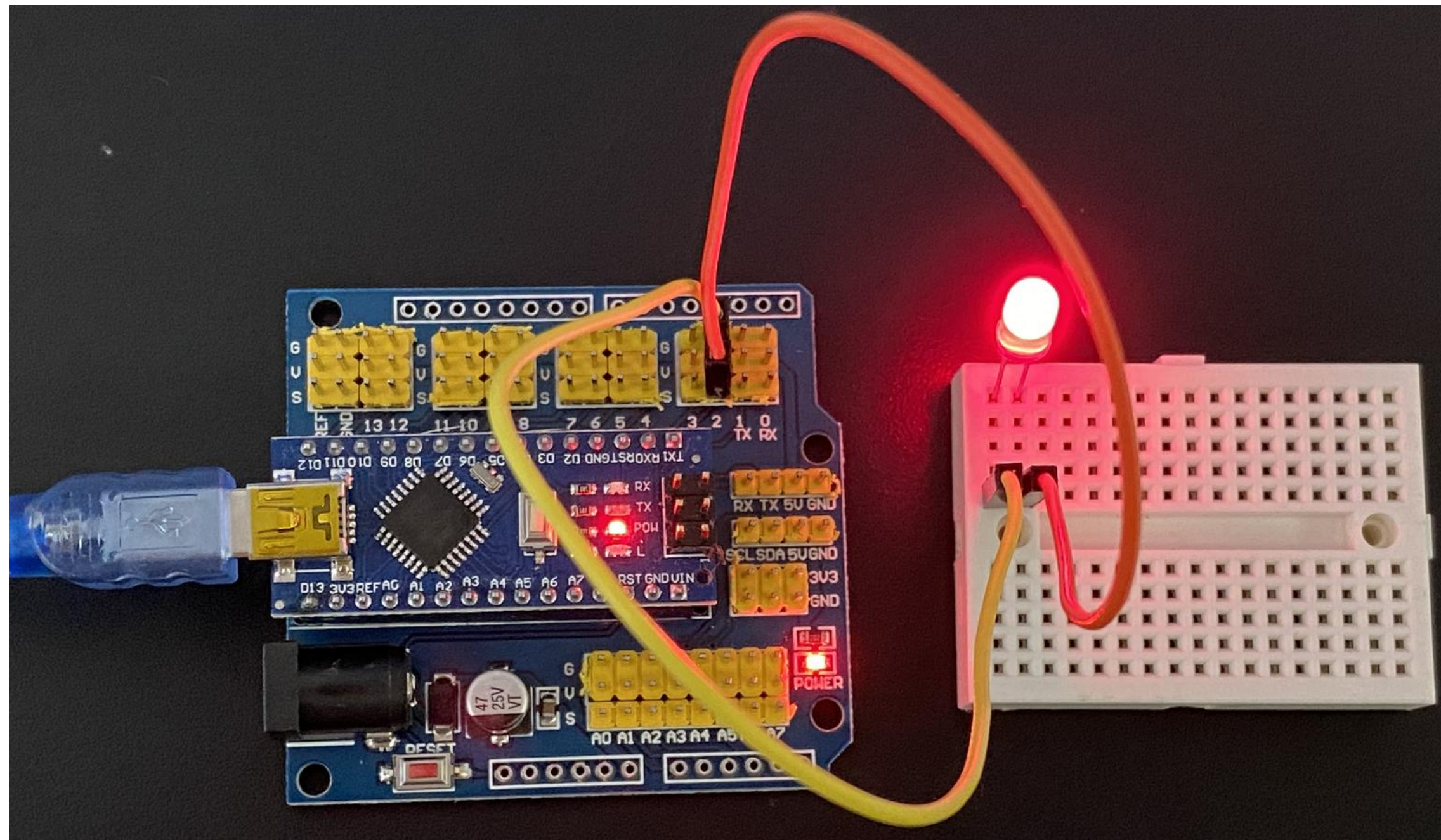
подождать 1 секунд

установить выход цифрового порта 2 как НИЗКИЙ ▼

подождать 1 секунд



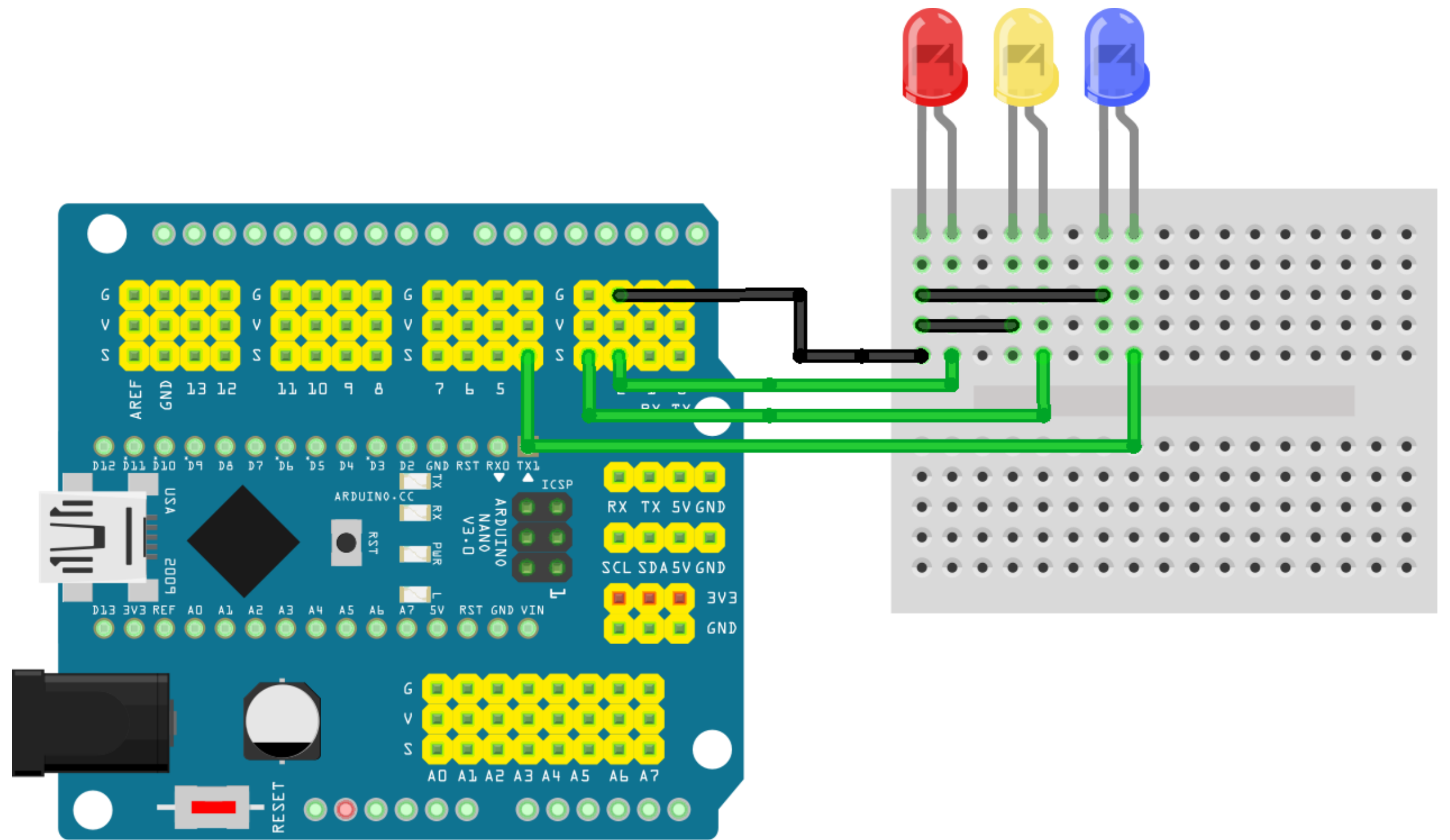
Что бы Arduino выполняла написанный нами новый блокный код, не забывайте загрузить его. Поставьте вместо 1 значения 0.5, 2, 5 и последите как измениться частота мигания

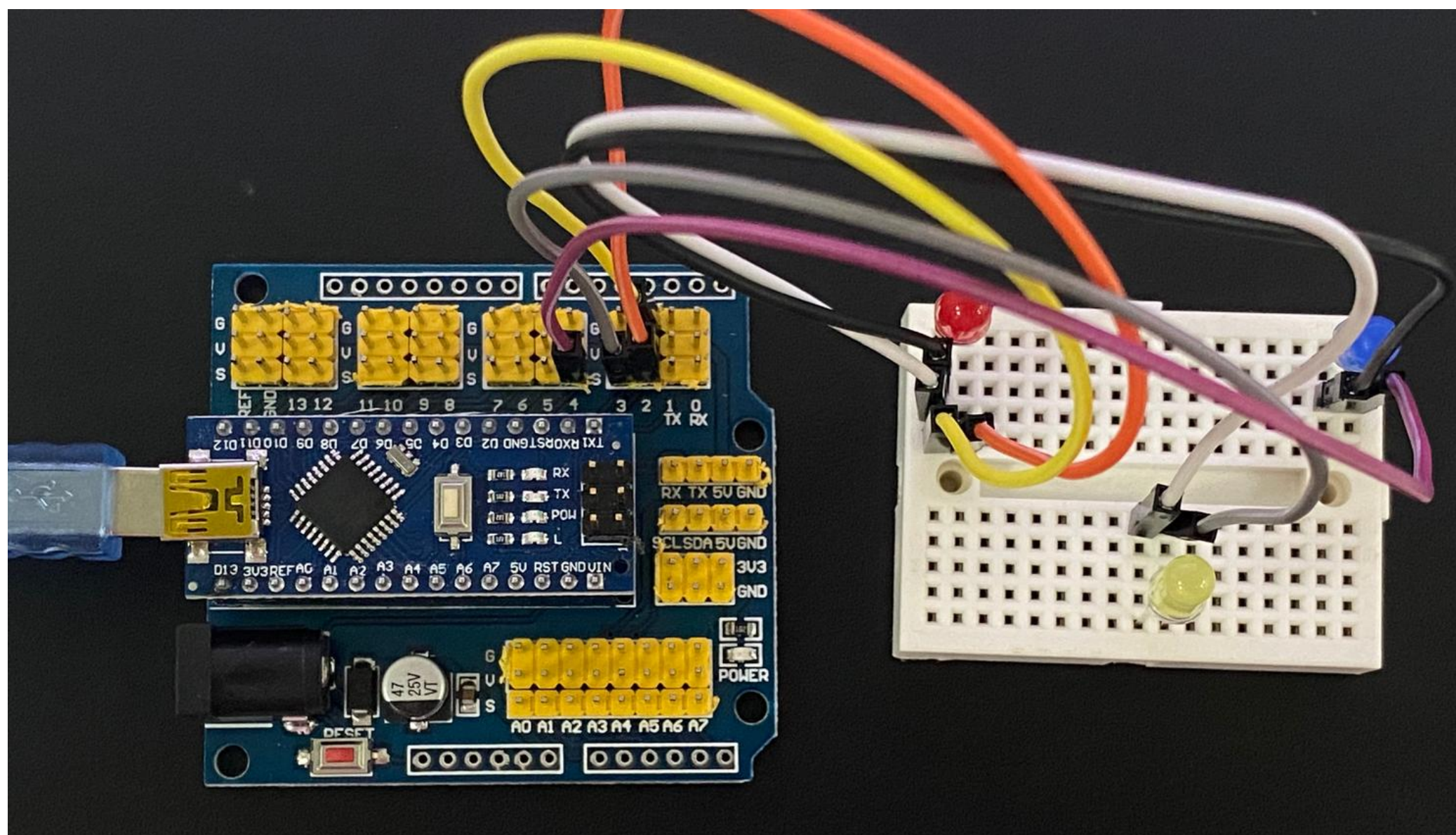
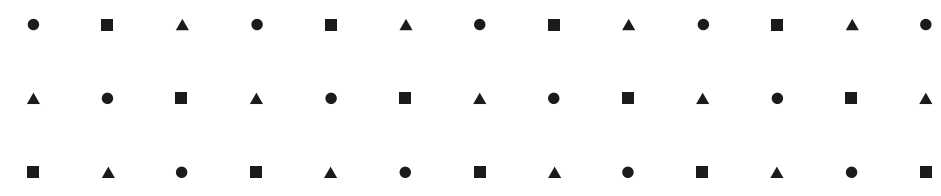


Шаг 10

Шаг 11

Соберем схему с тремя светодиодами

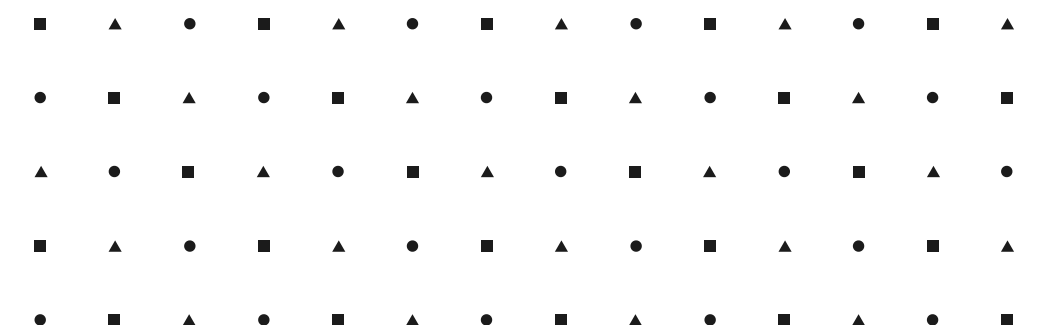




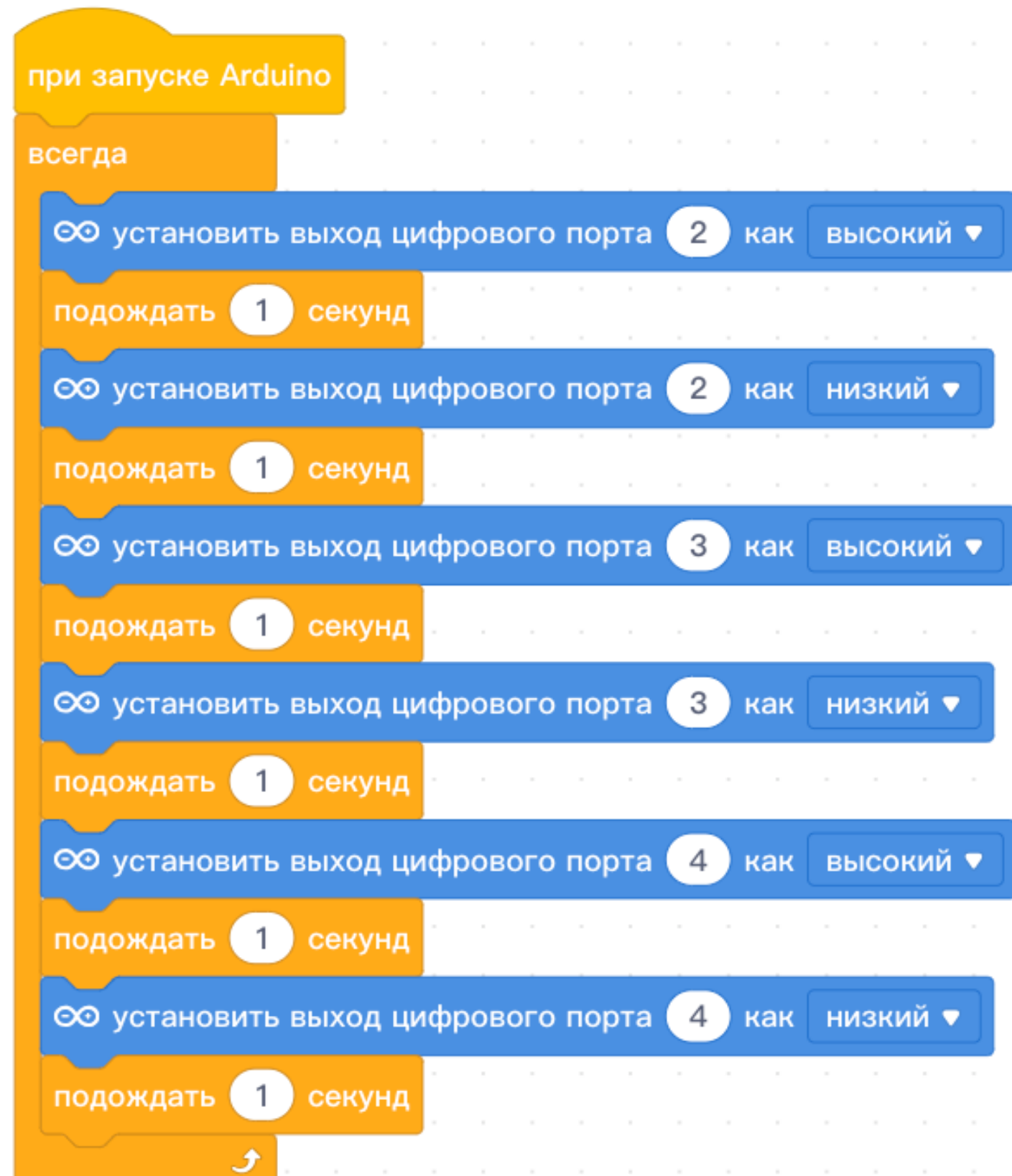
Конечно, расположение элементов на макетной плате не играет никакой роли. Самое главное - сохранять правильность подключений по схеме.

Цвета соединительных проводов никак не влияют на работу электрических схем. Для большего удобства в электронике принято правило обозначать земляные провода (G) черным цветом, сигнальные (S, A0) **желтым** или **синим**, а провода плюсового питания (+5V) **красным**.

Землю можно взять из отдельных металлических контактов - пинов, обозначенных (G). В нашем случае мы пользуемся так называемой "общей" землей



Напишем два программных кода, один из них будет **поочередно зажигать** и выключать светодиоды, а второй - давать команду светодиодам работать **синхронно**. Важно не забывать загружать внесенные в код изменения в плату



```
при запуске Arduino
всегда
  установить выход цифрового порта 2 как высокий
  подождать 1 секунд
  установить выход цифрового порта 2 как низкий
  подождать 1 секунд
  установить выход цифрового порта 3 как высокий
  подождать 1 секунд
  установить выход цифрового порта 3 как низкий
  подождать 1 секунд
  установить выход цифрового порта 4 как высокий
  подождать 1 секунд
  установить выход цифрового порта 4 как низкий
  подождать 1 секунд
```

Шаг 12

Шаг 13

Разная периодичность работы зависит от логического алгоритма, с которым расставлены задержки(подождать)

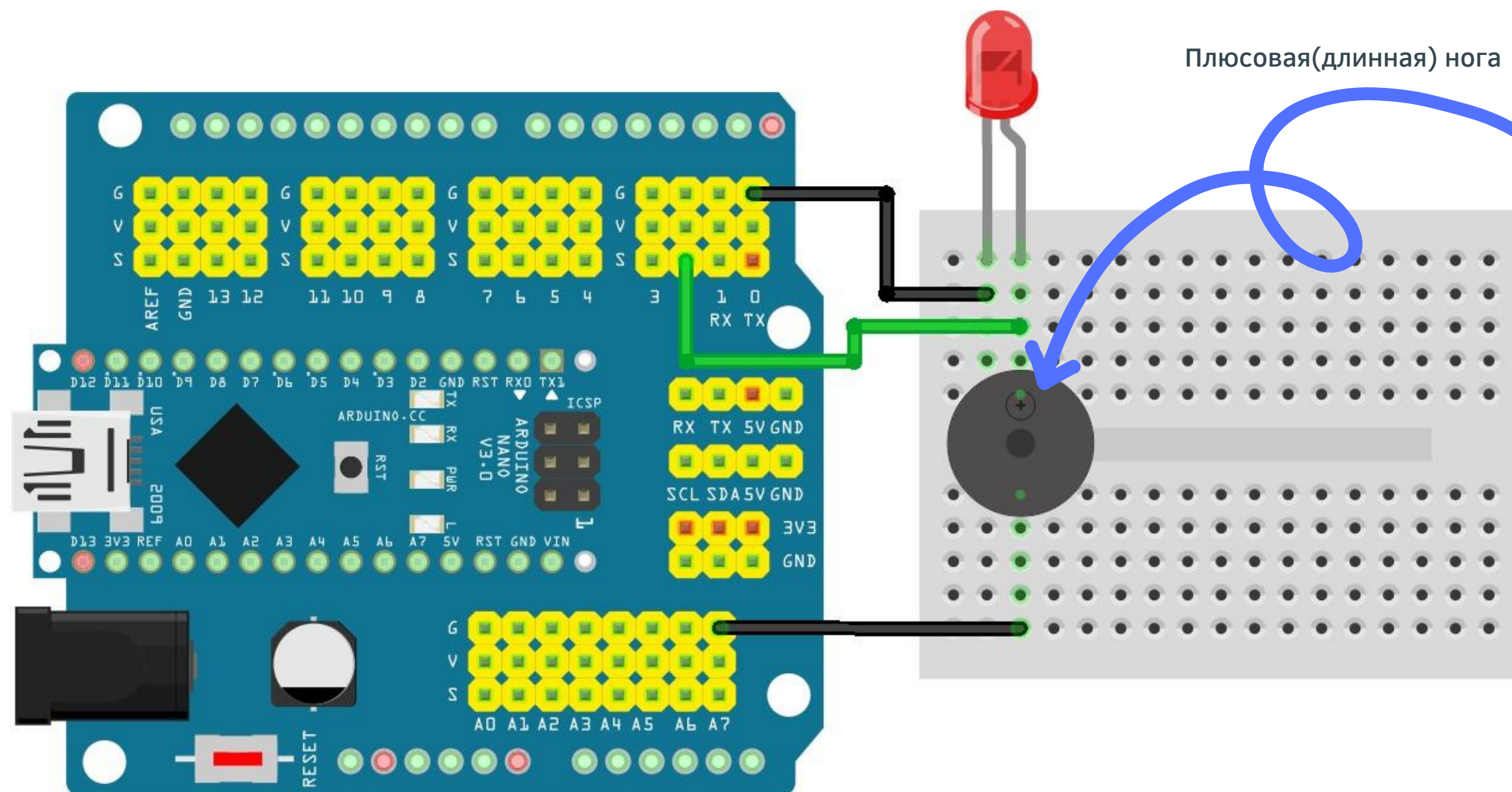
```
при запуске Arduino
  всегда
    установить выход цифрового порта 2 как ВЫСОКИЙ
    установить выход цифрового порта 3 как ВЫСОКИЙ
    установить выход цифрового порта 4 как ВЫСОКИЙ
    подождать 1 секунд
    установить выход цифрового порта 2 как НИЗКИЙ
    установить выход цифрового порта 3 как НИЗКИЙ
    установить выход цифрового порта 4 как НИЗКИЙ
    подождать 1 секунд
    ↺
```

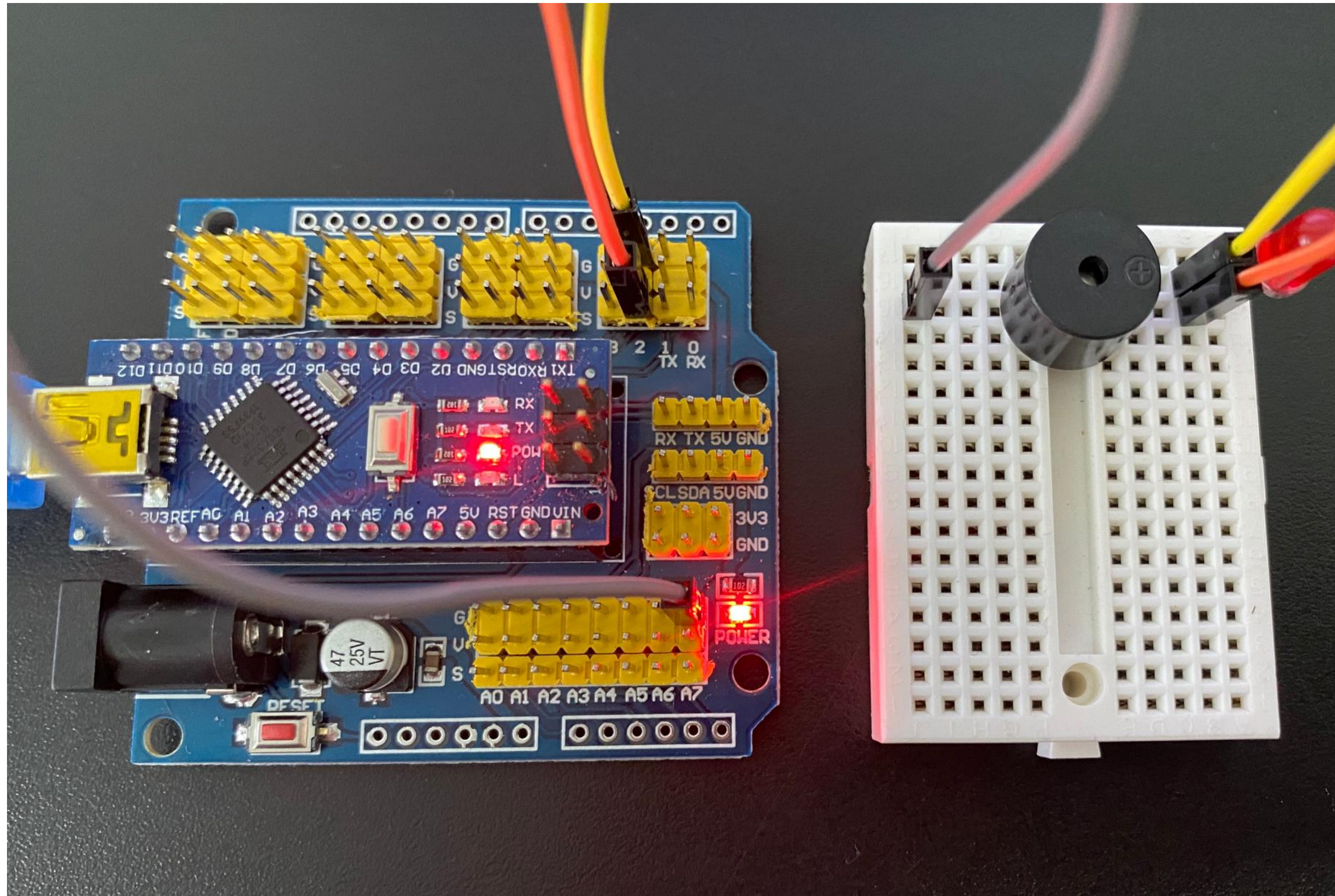
The code block is a vertical sequence of Scratch-style blocks. It starts with a yellow 'when Arduino starts' block, followed by an orange 'always' loop block. Inside the loop, there are three blue 'set digital pin mode' blocks for pins 2, 3, and 4, all set to 'HIGH'. This is followed by an orange 'wait 1 second' block. Then, there are three more blue 'set digital pin mode' blocks for pins 2, 3, and 4, all set to 'LOW'. This is followed by another orange 'wait 1 second' block. The loop ends with a yellow 'repeat' block icon.

Шаг 14

Соберем схему с пьезопищалкой и светодиодом, которая позволит проигрывать ноты различного звучания и частоты.

Светодиод будет выступать в качестве индикации наличия сигнала





В программном коде можно попробовать составить мелодию из различных нот и разной длительности.

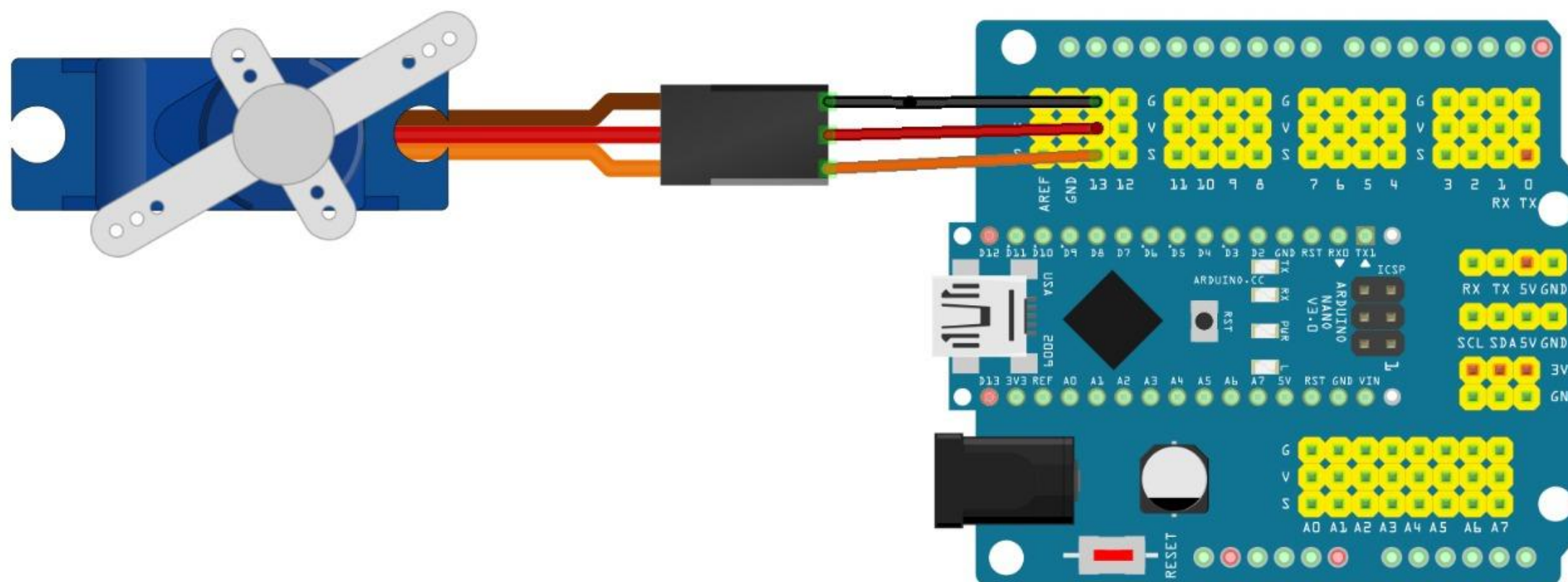
```
при запуске Arduino
  повторить 1
    порт 2 проигрывает ноту C4 0.25 долей
    подождать 0.25 секунд
    порт 2 проигрывает ноту E4 0.5 долей
    порт 2 проигрывает ноту C4 0.25 долей
    подождать 0.25 секунд
    порт 2 проигрывает ноту E4 0.5 долей
    порт 2 проигрывает ноту G4 0.25 долей
    подождать 0.25 секунд
    порт 2 проигрывает ноту C4 1 долей
    подождать 0.25 секунд
```

Шаг 15

Шаг 16

Схема подключения сервопривода очень простая. Но есть 2 варианта подключения: с помощью соединительных проводов или прямого подключения шлейфа сервопривода к плате расширения. При сборке робота сервопривод будет подключаться напрямую, без соединительных проводов

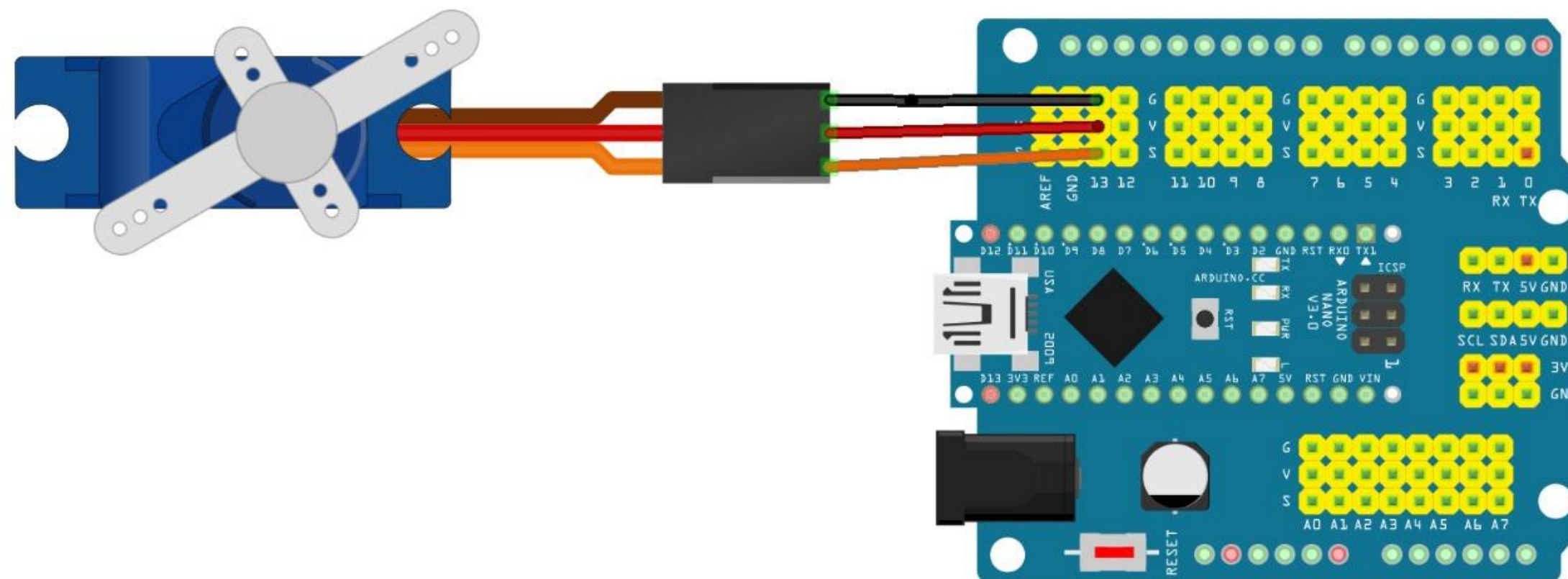
Важно не перепутать полярность подключения - оранжевый провод подключается в S, красный в V, коричневый (черный) – в G



Шаг 17

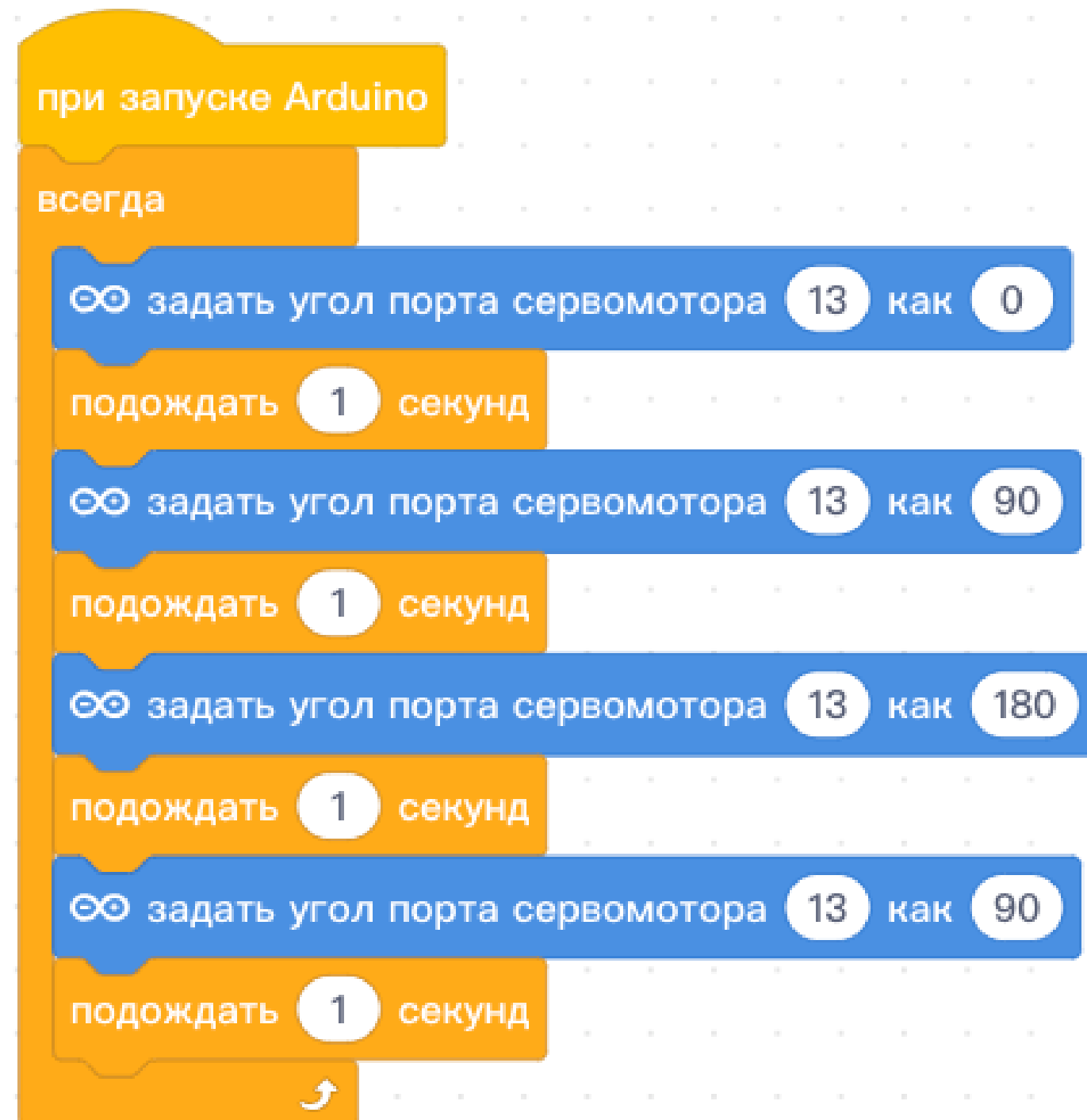
Схема подключения сервопривода очень простая. Но есть 2 варианта подключения: с помощью соединительных проводов или прямого подключения шлейфа сервопривода к плате расширения. При сборке робота сервопривод будет подключаться напрямую, без соединительных проводов.

Важно не перепутать полярность подключения - оранжевый провод подключается в S, красный в V, коричневый (черный) – в G.



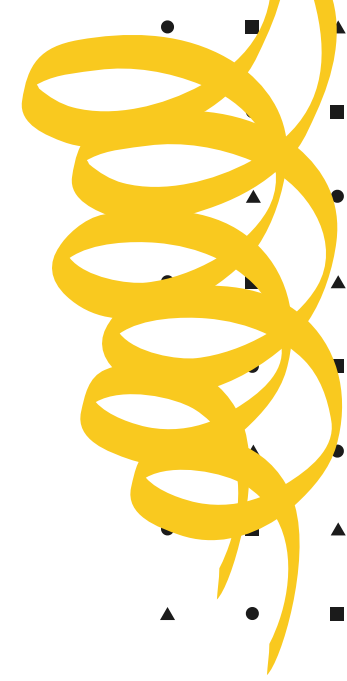
Программный код будет поворачивать вал сервопривода в 3 ключевых положения: 180, 90, 0 градусов.

Обязательно добавляем задержку между углами поворота: без неё сервопривод физически не будет успевать менять положение



```
при запуске Arduino
  всегда
    задать угол порта сервомотора 13 как 0
    подождать 1 секунд
    задать угол порта сервомотора 13 как 90
    подождать 1 секунд
    задать угол порта сервомотора 13 как 180
    подождать 1 секунд
    задать угол порта сервомотора 13 как 90
    подождать 1 секунд
    ↻
```





**Поздравляем!
Можно смело
переходить к
сборке робота
квадропода**

