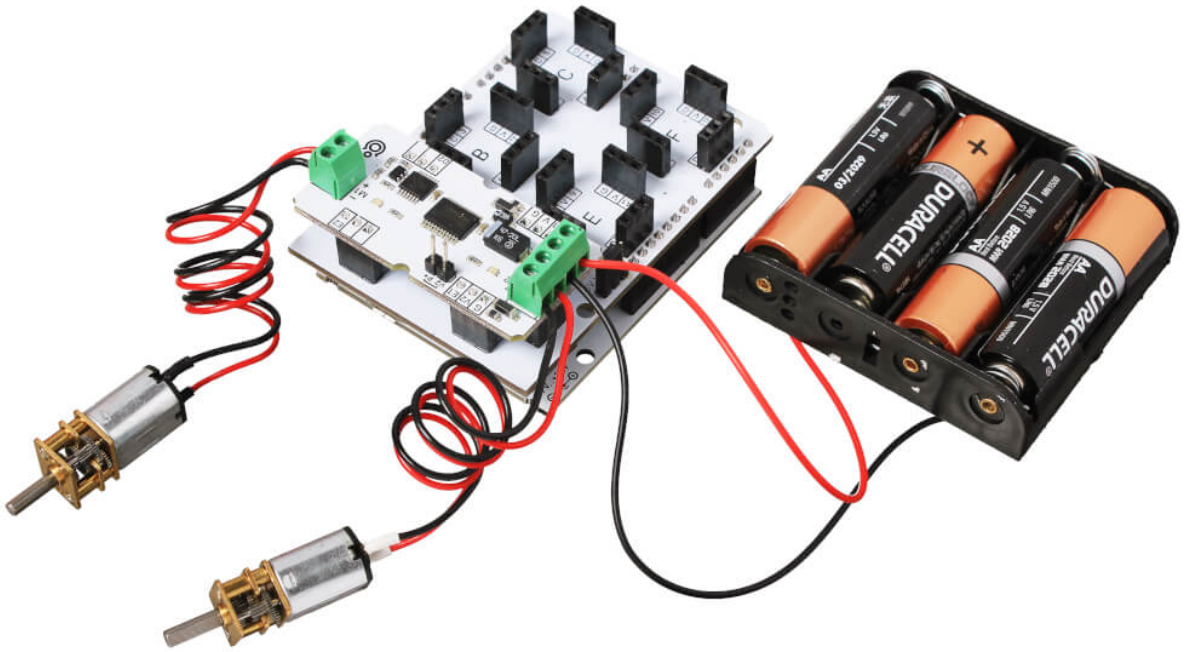
H-мост на 2 канала (Troyka-модуль): инструкция, схемы и примеры использования

Используйте H-мост для управления двумя коллекторными моторами, а точнее скоростью и направлением вращения вала. H-мост также сможет управлять одним биполярным шаговым двигателем.



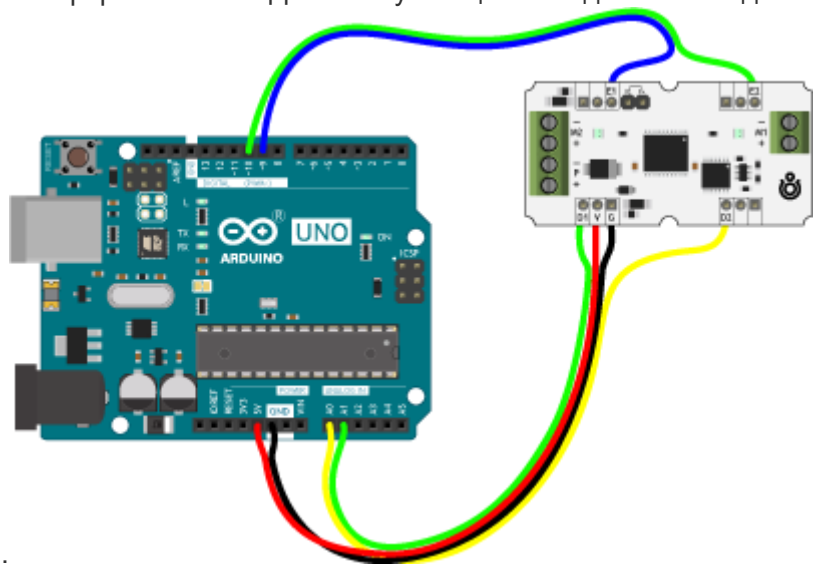
Примеры работы для Arduino и XOD

В качестве мозга для управления моторами рассмотрим платформу из серии Arduino, например Arduino Uno.

Подключение к Arduino

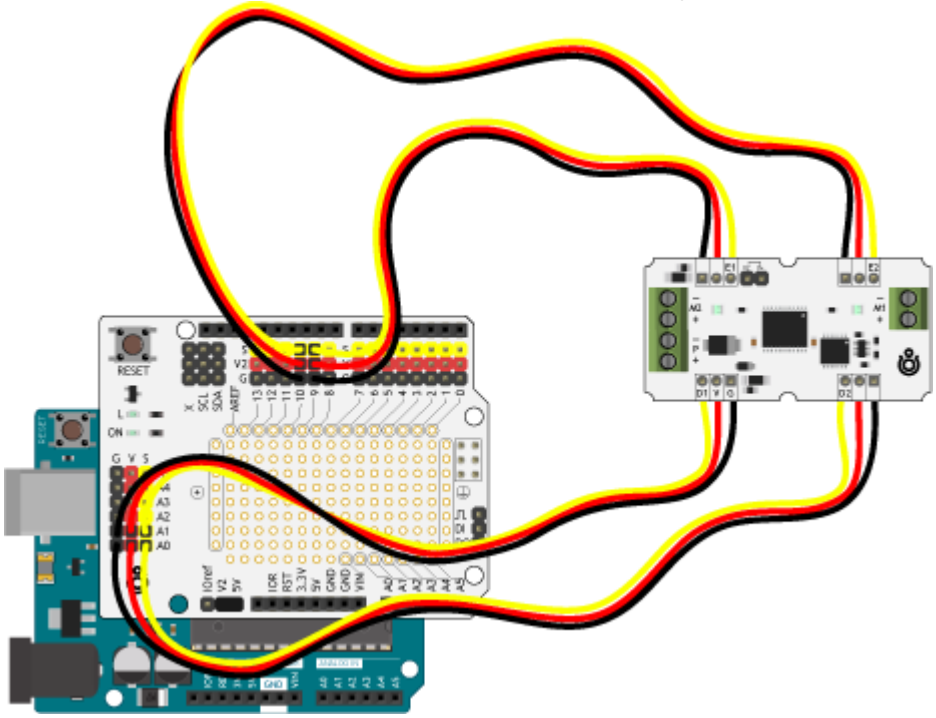
Выберите один из вариантов коммуникации драйвера с внешним микроконтроллером:

- Подключите H-мост к платформе Arduino. Для коммуникации понадобятся соединительные



провода «мама-папа».

- Для быстрой сборки и отладки устройства возьмите плату расширения Troyka Shield, которая одевается сверху на Arduino Uno методом бутерброда. Для коммуникации используйте трёхпроводные шлейфы «мама-мама», который идут в комплекте с модулем.



Управление коллекторными моторами

1. Подключите к драйверу микроконтроллер и логическое питание.
2. Подключите к драйверу коллекторные моторы и силовое питание.

Код для Arduino

Для начала покрутим каждый мотор в одну, а затем другую сторону. Прошейте платформу Arduino скетчем, приведённым ниже.

troyka-h-bridge-dual-example-arduino-dc-motors.ino

```
// Пины управления скоростью и направлением мотора
constexpr auto pinM1Speed = 9;
constexpr auto pinM1Direction = A1;
constexpr auto pinM2Speed = 10;
constexpr auto pinM2Direction = A0;

int pins[] = {pinM1Speed, pinM1Direction, pinM2Speed, pinM2Direction};

void setup() {
  // Настраиваем все пины управление моторами в режим выхода
  for (int i = 0; i < 4; i++) {
    pinMode(pins[i], OUTPUT);
  }
}

void loop() {
  // Крутим мотор M1 в одну сторону в течении 1 секунды
```

```
  motorsDrive(255, 0);
  delay(1000);

  // Крутим мотор M1 в другую сторону в течении 1 секунды
  motorsDrive(-255, 0);
  delay(1000);

  // Крутим мотор M2 в одну сторону в течении 1 секунды
  motorsDrive(0, 255);
  delay(1000);

  // Крутим мотор M2 в другую сторону в течении 1 секунды
  motorsDrive(0, -255);
  delay(1000);

  // Стоим на месте
  motorsDrive(0, 0);
  delay(1000);

}

// Функция управления моторами
void motorsDrive(int M1Speed, int M2Speed) {
  if (M1Speed > 0) {
    digitalWrite(pinM1Direction, HIGH);
  } else {
    digitalWrite(pinM1Direction, LOW);
  }

  if (M2Speed > 0) {
    digitalWrite(pinM2Direction, HIGH);
  } else {
    digitalWrite(pinM2Direction, LOW);
  }
  analogWrite(pinM1Speed, abs(M1Speed));
  analogWrite(pinM2Speed, abs(M2Speed));
}
```

Усовершенствуем эксперимент: заставим каждый мотор по очереди плавно разгоняться и останавливаться в разных направлениях.

troyka-h-bridge-dual-example-arduino-dc-motors-pwm.ino

```
// Пины управления скоростью и направлением мотора
constexpr auto pinM1Speed = 9;
constexpr auto pinM1Direction = A1;
constexpr auto pinM2Speed = 10;
constexpr auto pinM2Direction = A0;

int pins[] = {pinM1Speed, pinM1Direction, pinM2Speed, pinM2Direction};

void setup() {
```

```
  // Настраиваем все пины управление моторами в режим выхода
  for (int i = 0; i < 4; i++) {
    pinMode(pins[i], OUTPUT);
  }
}

void loop() {
  // Медленно разгоняем M1 в одну сторону
  for (int i = 0; i <= 255; i++) {
    motorsDrive(i, 0);
    delay(10);
  }
  // Медленно тормозим мотор
  for (int i = 255; i >= 0; i--) {
    motorsDrive(i, 0);
    delay(10);
  }
  // Медленно разгоняем M1 в другую сторону
  for (int i = 0; i <= 255; i++) {
    motorsDrive(-i, 0);
    delay(10);
  }
  // Медленно тормозим мотор
  for (int i = 255; i >= 0; i--) {
    motorsDrive(-i, 0);
    delay(10);
  }

  // медленно разгоняем M2 в одну сторону
  for (int i = 0; i <= 255; i++) {
    motorsDrive(0, i);
    delay(10);
  }
  // медленно тормозим мотор
  for (int i = 255; i >= 0; i--) {
    motorsDrive(0, i);
    delay(10);
  }
  // медленно разгоняем M2 в другую сторону
  for (int i = 0; i <= 255; i++) {
    motorsDrive(0, -i);
    delay(10);
  }
  // медленно тормозим мотор
  for (int i = 255; i >= 0; i--) {
    motorsDrive(0, -i);
    delay(10);
  }

}
```

```
// Функция управления моторами
void motorsDrive(int M1Speed, int M2Speed) {
  if (M1Speed > 0) {
    digitalWrite(pinM1Direction, HIGH);
  } else {
    digitalWrite(pinM1Direction, LOW);
  }

  if (M2Speed > 0) {
    digitalWrite(pinM2Direction, HIGH);
  } else {
    digitalWrite(pinM2Direction, LOW);
  }
  analogWrite(pinM1Speed, abs(M1Speed));
  analogWrite(pinM2Speed, abs(M2Speed));
}
```

Управление шаговым двигателем

1. Подключите к драйверу микроконтроллер и логическое питание.
2. Подключите к драйверу шаговый двигатель и силовое питание.

Для лёгкого и быстрого управления шаговым двигателем мы написали библиотеку AmperkaStepper, которая скрывает в себе все тонкости работы с мотором и предоставляет удобные методы.

Код для Arduino

troyka-h-bridge-dual-example-arduino-stepper.ino

```
// Библиотека для работы с шаговым двигателем
#include <AmperkaStepper.h>

// Создаём объект для работы с шаговым двигателем
// и передаём фиксированное количество шагов за полный оборот.
// Подробности в характеристиках двигателя
AmperkaStepper motor(200, A0, A1, 9, 10);

void setup() {
  // Устанавливаем скорость вращения 30 оборотов в минуту.
  motor.setSpeed(30);
}

void loop() {
  // 180° по часовой стрелке в двухфазном режиме
  motor.step(100, FULL_STEP);
  delay(1000);

  // 180° против часовой стрелки в однофазном режиме
  motor.step(-100, WAVE_DRIVE);
  delay(1000);
```

```
  // 180° по часовой стрелке в полушаговом режиме
  motor.step(200, HALF_STEP);
  delay(1000);

  // 180° против часовой стрелки в двухфазном режиме
  // этот режим используется по умолчанию, если не передан
  // второй аргумент
  motor.step(-100);
  delay(1000);
}
```

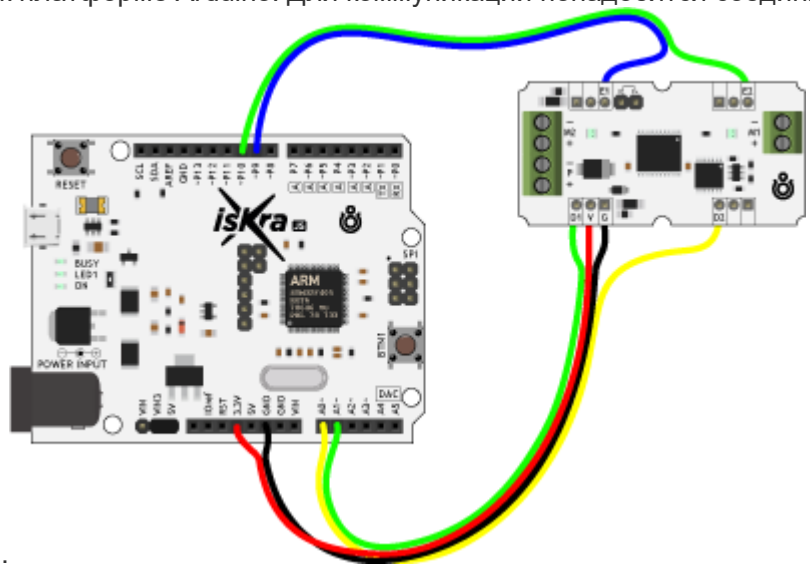
Пример работы для Espruino

В качестве мозга для управления моторами рассмотрим платформу из серии Espruino, например, Iskra JS.

Подключение к Espruino

Выберите один из вариантов коммуникации драйвера с внешним микроконтроллером:

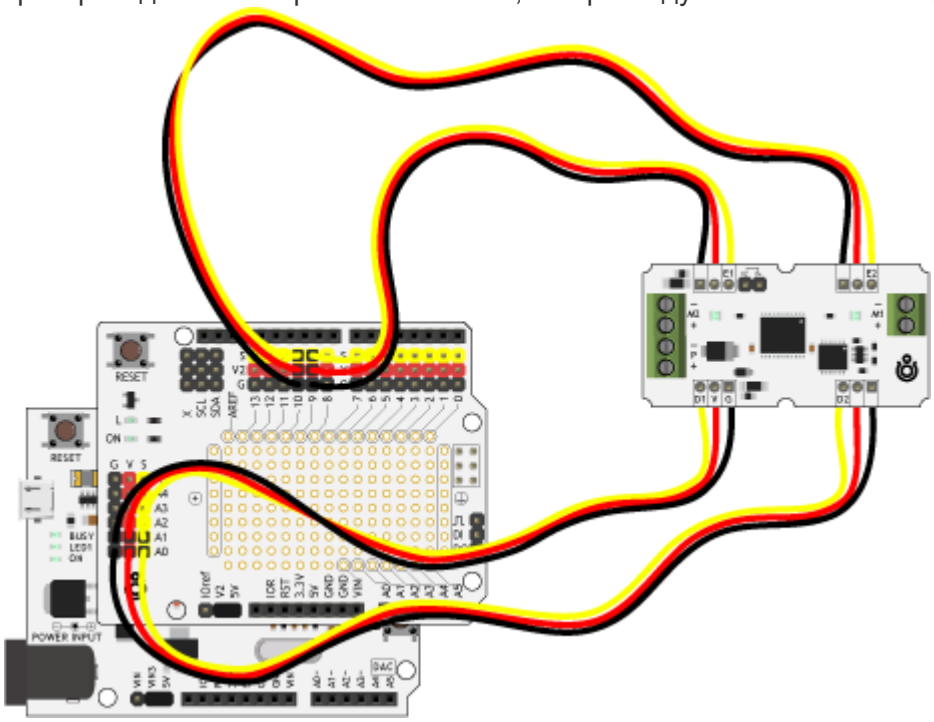
- Подключите драйвер к платформе Arduino. Для коммуникации понадобятся соединительные



провода «мама-папа».

- Для быстрой сборки и отладки устройства возьмите плату расширения Troyka Shield, которая одевается сверху на Arduino Uno методом бутерброда. Для коммуникации используйте

трёхпроводные шлейфы «мама-мама», который идут в комплекте с модулем.



Управление коллекторными двигателями

1. Подключите к драйверу микроконтроллер и логическое питание.
2. Подключите к драйверу коллекторные моторы и силовое питание.

Код для Espruino IDE

Покрутим каждый мотор в одну, а затем другую сторону. Прошейте платформу Espruino скриптом, приведённым ниже.

troyka-h-bridge-dual-example-espruino-dc-motors.js

```
// Подключаем библиотеку «motor»
var Motor = require('@amperka/motor');



// Пины управления скоростью и направлением мотора
var motorOne = Motor.connect({phasePin: A1, pwmPin: P9, freq: 100});
var motorTwo = Motor.connect({phasePin: A0, pwmPin: P10, freq: 100});

// Интервал времени
var time = 1000;
// Счётчик
var state = 0;

// Каждую секунду меняем режим работы
setInterval(() => {
  motorOne.write(0);
  motorTwo.write(0);
  state++;
  if (state === 1) {
    motorOne.write(1);
```

```
  } else if (state === 2) {
    motorOne.write(-1);
  } else if (state === 3) {
    motorTwo.write(1);
  } else if (state === 4) {
    motorTwo.write(-1);
  } else {
    state = 0;
  }
}, time);
```

Управление шаговым двигателем

1. Подключите к драйверу микроконтроллер и логическое питание.
2. Подключите к драйверу шаговый двигатель и силовое питание.

Для лёгкого и быстрого управления шаговым двигателем, используйте библиотеку StepperMotor, которая скрывает в себе все тонкости работы с шаговиком и предоставляет удобные методы.

Код для Espruino

troyka-h-bridge-dual-example-espruino-stepper.js

```
// Подключаем библиотеку «motor»
var StepperMotor = require("StepperMotor");

// Создаём объект для работы с шаговым двигателем
// передаём пины управления
var motor = new StepperMotor({
  pins:[A0, A1, P9, P10],
  pattern:[0b0001,0b0011,0b0010,0b0110,0b0100,0b1100,0b1000,0b1001],
});

// Крутим вал на 100 шагов по часовой стрелке
motor.moveTo(100, 5000, function() {
  // Крутим вал на 100 шагов против часовой стрелке
  motor.moveTo(-100, 5000, function() {
    // Приехали
    console.log("Done!");
  }, true);
});
```

Пример работы для Raspberry Pi

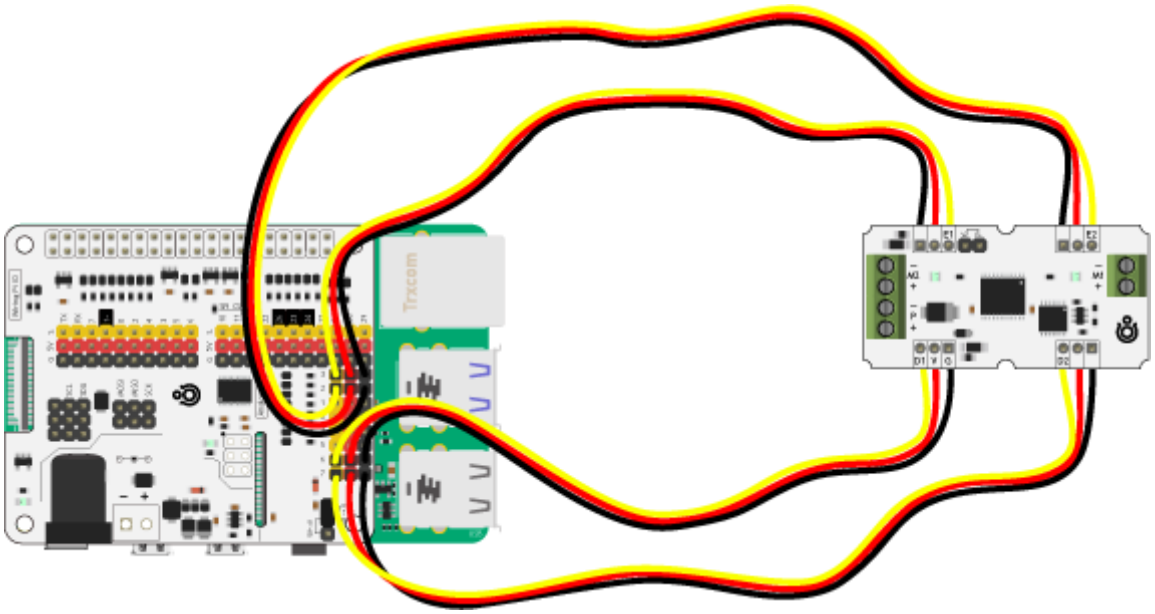
В качестве мозга для управления моторами рассмотрим одноплатные компьютеры Raspberry Pi, например, Raspberry Pi 4.

Подключение к Raspberry Pi

В компьютере Raspberry Pi присутствует только два канала с ШИМ-сигналом, и то которые используются для аналогового звукового выхода. В итоге для регулировки скоростью моторов

придется жертвовать звуком. Используйте плату расширения Troyka Cap, которая добавит малине 9 пинов с поддержкой ШИМ.

Подключите драйвер к компьютеру Raspberry Pi через Troyka Cap. Для коммуникации используйте трёхпроводные шлейфы «мама-мама», который идут в комплекте с модулем.



Управление коллекторными двигателями

1. Подключите к драйверу микроконтроллер и логическое питание.
2. Подключите к драйверу коллекторные моторы и силовое питание.

Код для Raspberry Pi

Для начала покрутим каждый мотор в одну, а затем другую сторону. Запустите скрипт на малине, приведённый ниже.

troyka-h-bridge-dual-example-raspberry-pi-dc-motors.py

```
# библиотека для работы с пинами GPIO
import RPi.GPIO as GPIO
# библиотека для работі с временем
import time

# выбираем имена пинов BCM
GPIO.setmode(GPIO.BCM)
# устанавливаем светодиод в режим выхода
GPIO.setup(24, GPIO.OUT)

try:
    while True:
        # ждём одну секунду
        time.sleep(1)
        # зажигаем светодиод
        GPIO.output(24, GPIO.HIGH)
        # ждём одну секунду
        time.sleep(1)
```

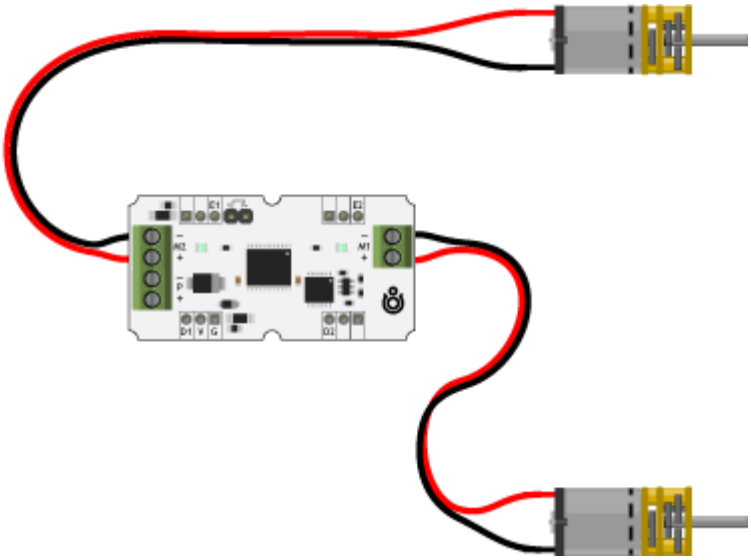
```
        # гасим светодиод
        GPIO.output(24, GPIO.LOW)
except KeyboardInterrupt:
    print('The program was stopped by keyboard.')
finally:
    GPIO.cleanup()
    print('GPIO cleanup completed.')
```

Подключение силового контура

Н-мост может управлять двумя отдельными коллекторными моторами или одним биполярным шаговым двигателем.

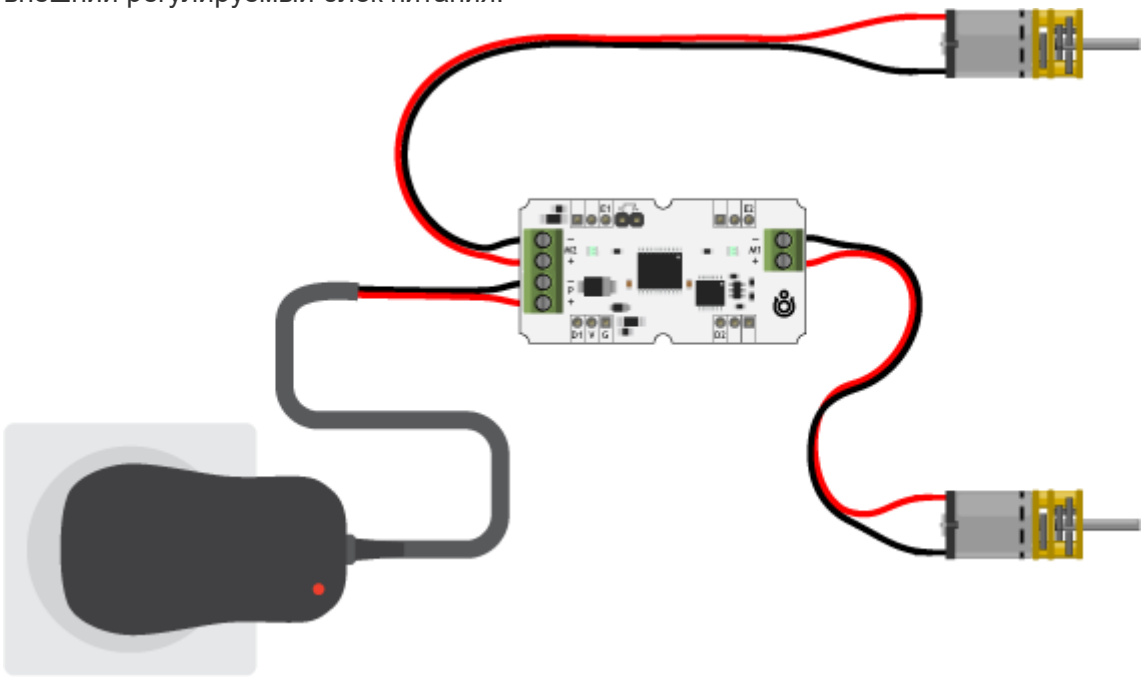
Подключение коллекторных моторов

1. Подключите два коллекторных мотора к клеммникам `M1` и `M2` соответственно.

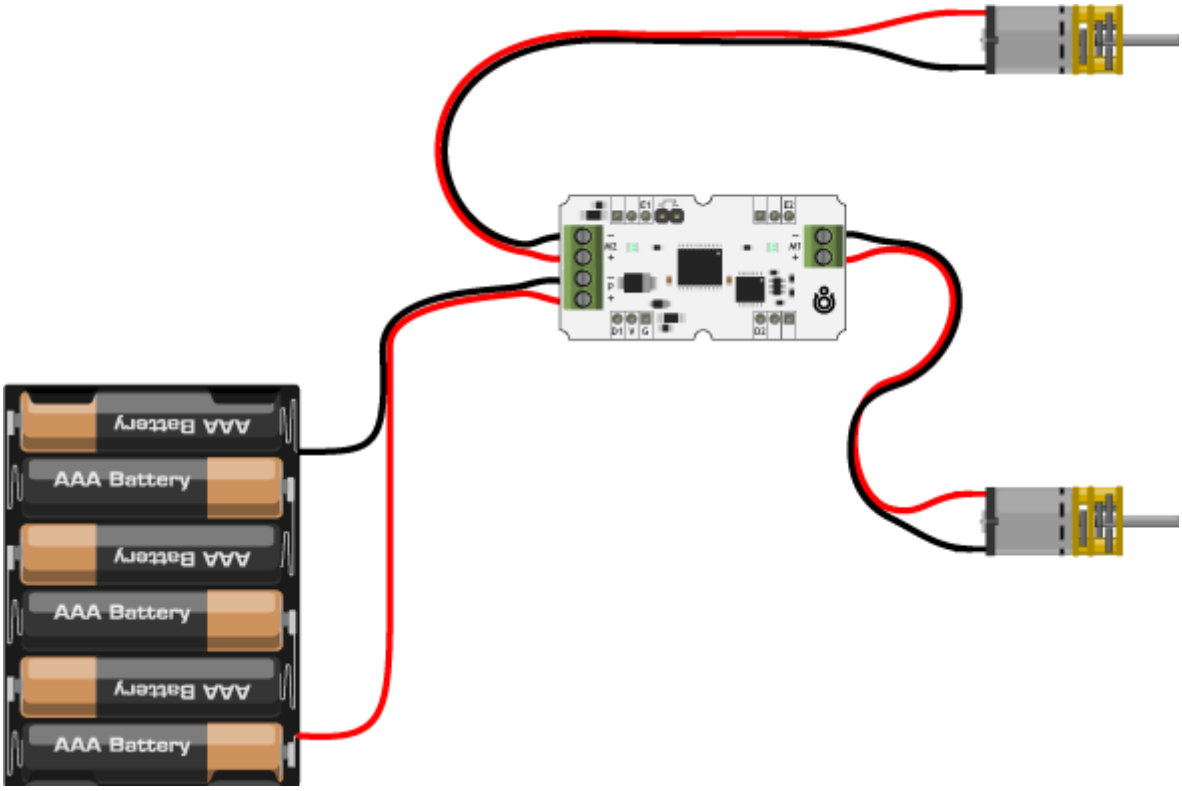


1. Подключите силовое питание для моторов через клеммник `P`.

- В качестве стационарного источника напряжения рекомендуем использовать внешний регулируемый блок питания.



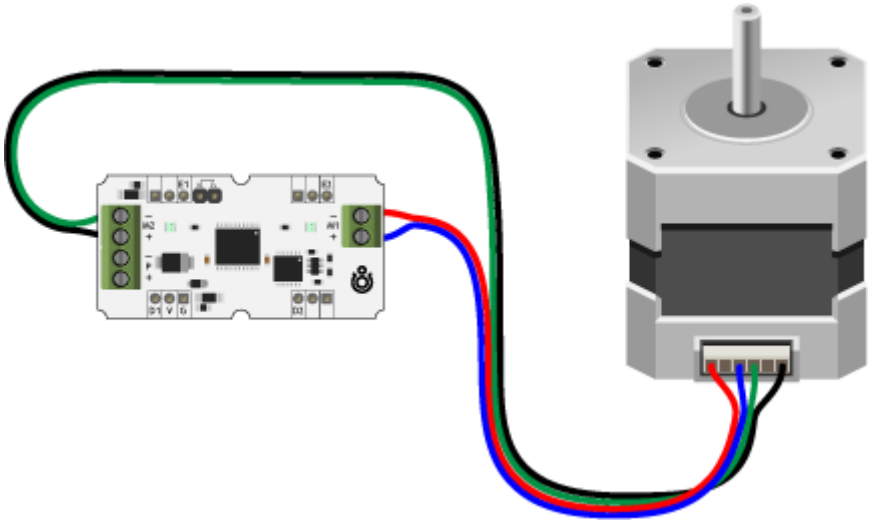
- В качестве автономного источника обратите внимание на батарейный отсек с элементами питания.



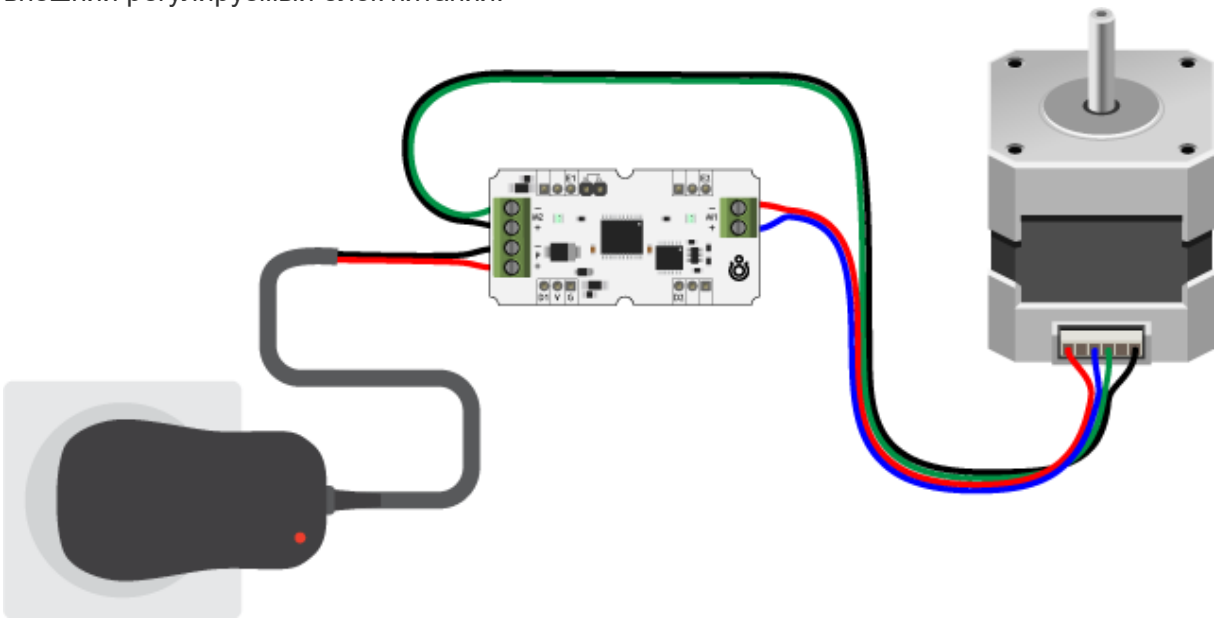
Значение входного силового напряжения зависит от номинального напряжения подключаемых моторов и ограничено диапазоном от 3,3 до 12 вольт.

Подключение шагового двигателя

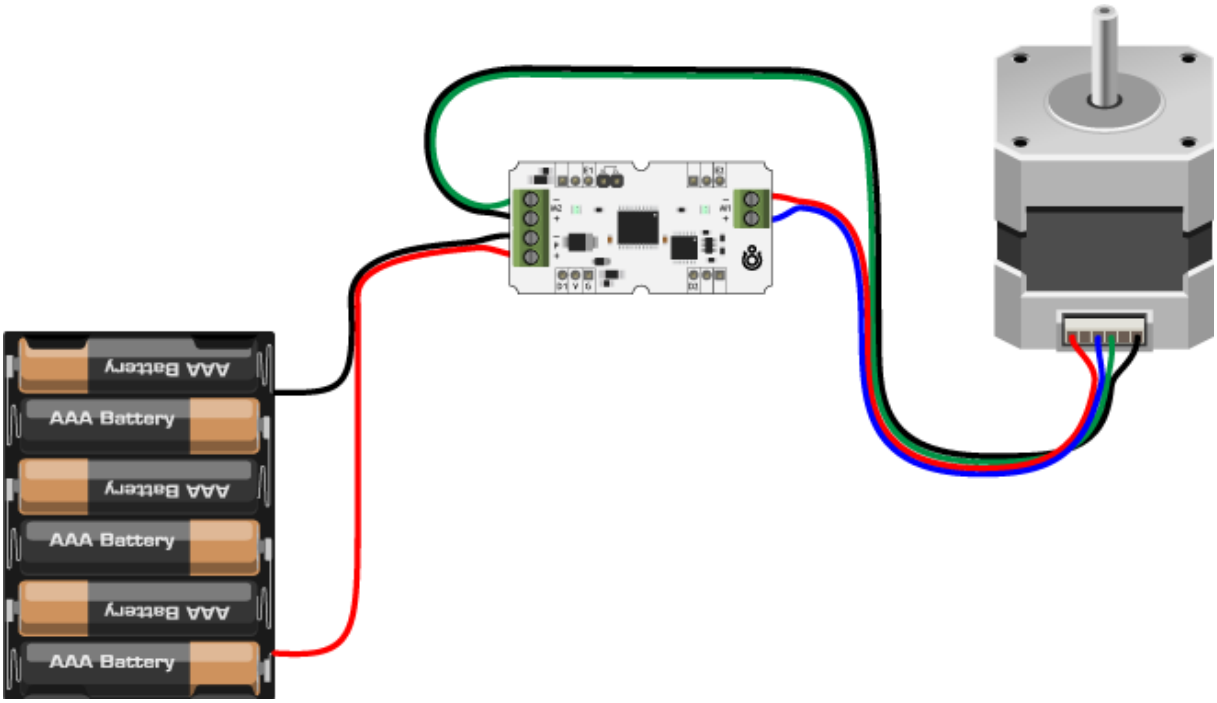
1. Подключите шаговый двигатель к клеммникам `M1` и `M2`.



1. Подключите силовое питание для мотора через клеммник `P`.
    - В качестве стационарного источника напряжения рекомендуем использовать внешний регулируемый блок питания.



- В качестве автономного источника обратите внимания на батарейный отсек с элементами питания.



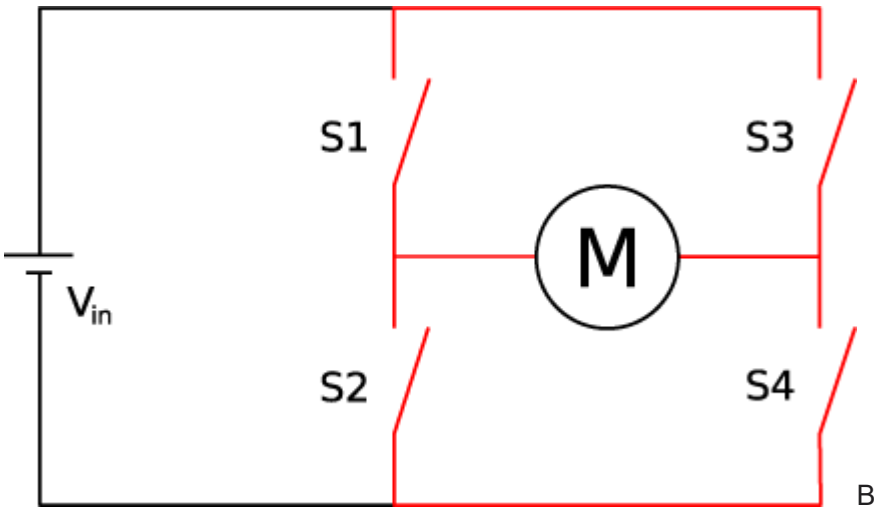
Значение входного силового напряжения зависит от номинального напряжения шагового двигателя и ограничено диапазоном от 3,3 до 12 вольт.

Элементы платы

Драйвер двигателей TB6612FNG

Сердце и мускулы платы — микросхема двухканального H-моста TB6612FNG, которая позволяет управлять двумя коллекторными моторами или одним биполярным шаговым двигателем с помощью внешнего микроконтроллера.

Термин «H-мост» появился благодаря графическому изображению схемы, напоминающему букву «H». Рассмотрим подробнее принцип работы H-моста.



В зависимости от текущего состояние переключателей возможно разное состояние мотора.

S1	S2	S3	S4	Результат
1	0	0	1	Мотор крутится вправо
0	1	1	0	Мотор крутится влево
0	0	0	0	Свободное вращение мотора
0	1	0	1	Мотор тормозит
1	0	1	0	Мотор тормозит
1	1	0	0	Короткое замыкание источника питания
0	0	1	1	Короткое замыкание источника питания

Ключи меняем на MOSFET-транзисторы, а для плавной регулировки скорости вращения вала мотора используем ШИМ-сигнал.

Питание

На плате драйвера моторов присутствует два контура питания: силовое и логическое.

- Силовой контур (VM) — напряжение для питания моторов от силовой части микросхемы TB6612FNG и светодиодов индикации. Силовое питание подключается через клеммник `P` с входным диапазоном напряжения от 5 до 12 вольт.
- Логический контур (Vcc) — питание вспомогательной цифровой логики управления микросхемы TB6612FNG. Логическое питание поступает на плату модуля через контакт `V`. Диапазон входного напряжения от 3,3 до 5 вольт.

Если отсутствует хотя бы один из контуров питания — драйвер H-мост работать не будет.

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

Нагрузка

Нагрузка разделена на два независимых канала. Первый канал на плате обозначен шёлком `M1`, а второй канал — `M2`. К каждому каналу можно подключить по одному коллекторному мотору или объединить каналы для подключения биполярного шагового двигателя.

Обозначения «+» и «−» показывают воображаемые начало и конец обмотки. Если подключить два коллекторных двигателя, чтобы их одноимённые контакты щёточного узла соответствовали одному и тому же обозначению на плате, то при подаче на H-Bridge одинаковых управляющих импульсов, моторы будут вращаться в одну и ту же сторону.

Светодиодная индикация

Имя светодиода	Назначение
DIR1/EN1	Индикация состояния направления и скорости первого канала `M1`. При высоком логическом уровне светится зелёным светом, при низком — красным. Яркость светодиода пропорциональна скорости вращения двигателя.
DIR2/EN2	Индикация состояния направления и скорости первого канала `M2`. При высоком логическом уровне светится зелёным светом, при низком — красным. Яркость светодиода пропорциональна скорости вращения двигателя.

Характеристики

- Драйвер моторов: TB6612FNG
- Количество подключаемых моторов: 2
- Напряжение логической части: 3,3–5 В
- Напряжение силовой части: 3,3—12 В
- Максимальный ток нагрузки: до 1,2 А на канал
- Максимальная частота переключения (ШИМ): 100 кГц
- Габариты модуля: 50,8×25,4×19 мм