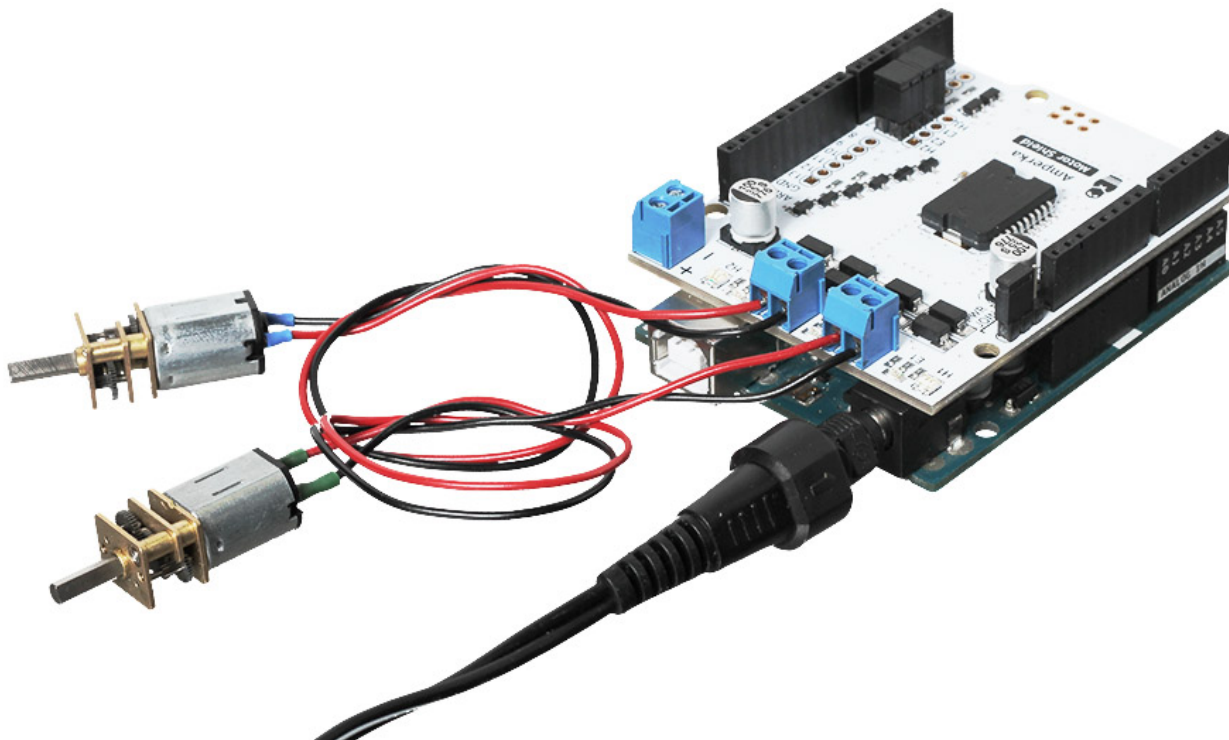


## Motor Shield

Хотите подключить мотор к Arduino [[https://amperka.ru/collection/arduino?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/collection/arduino?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)] или Iskra JS [[https://amperka.ru/product/iskra-js?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/product/iskra-js?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)]?! К сожалению микроконтроллер установленный на управляющих платформах не сможет без посторонней помощи управлять большой нагрузкой. Motor Shield [[https://amperka.ru/product/arduino-motor-shield?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/product/arduino-motor-shield?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)] поможет микроконтроллеру управлять коллекторными моторами и шаговыми двигателями.



Motor Shield — это плата расширения, предназначенная для двухканального управления скоростью и направлением вращения коллекторных двигателей постоянного тока, напряжением 5–24 В и максимальным током до двух ампер на канал.

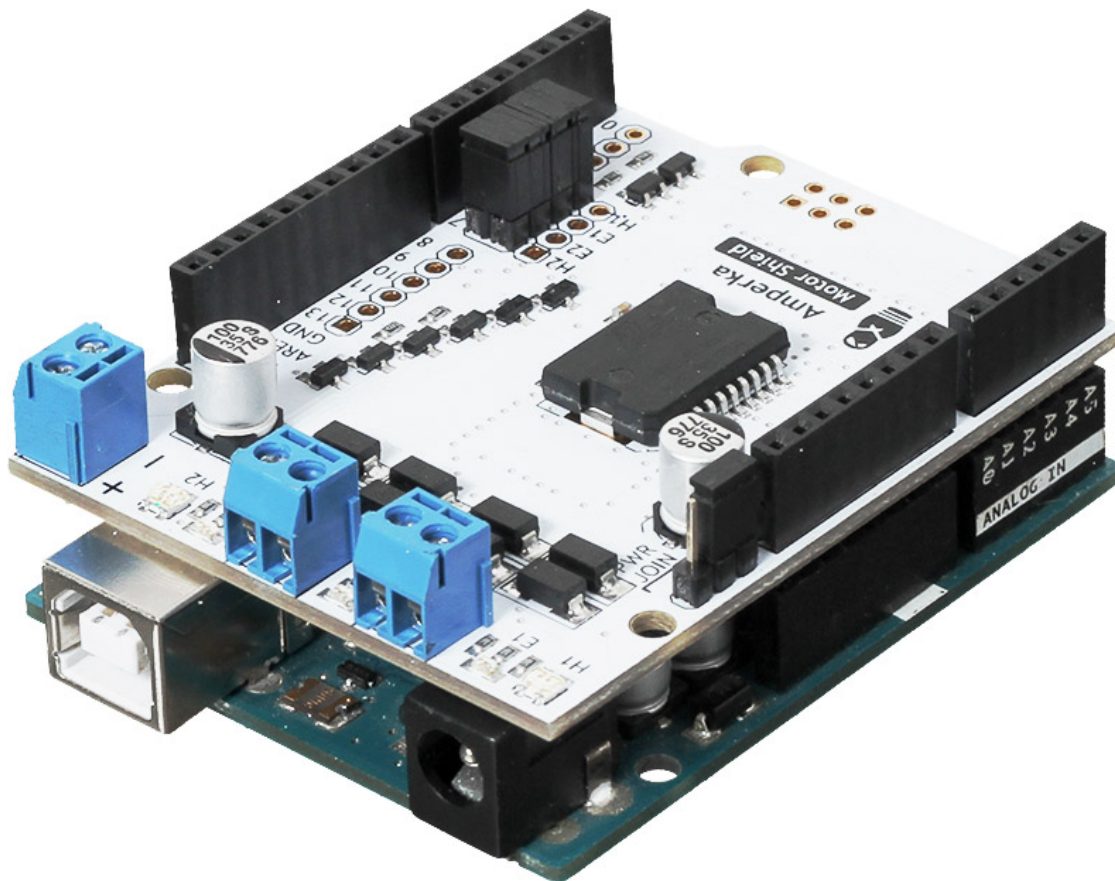
Плата расширения также сможет управлять одним биполярным шаговым двигателем.

## Видеообзор

## Подключение и настройка

1.

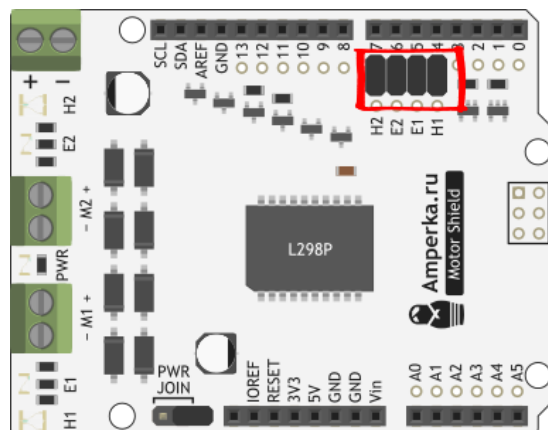
Установите Motor Shield сверху на управляющую платформу, например Arduino Uno [[https://amperka.ru/product/arduino-uno?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/product/arduino-uno?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)] или Iskra JS [[https://amperka.ru/product/iskra-js?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/product/iskra-js?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)].



2.

Убедитесь в наличии и правильности соединения джамперов на плате Motor Shield:

- H1 с 4 пином;
- E1 с 5 пином.
- E2 с 6 пином;
- H2 с 7 пином.



3.

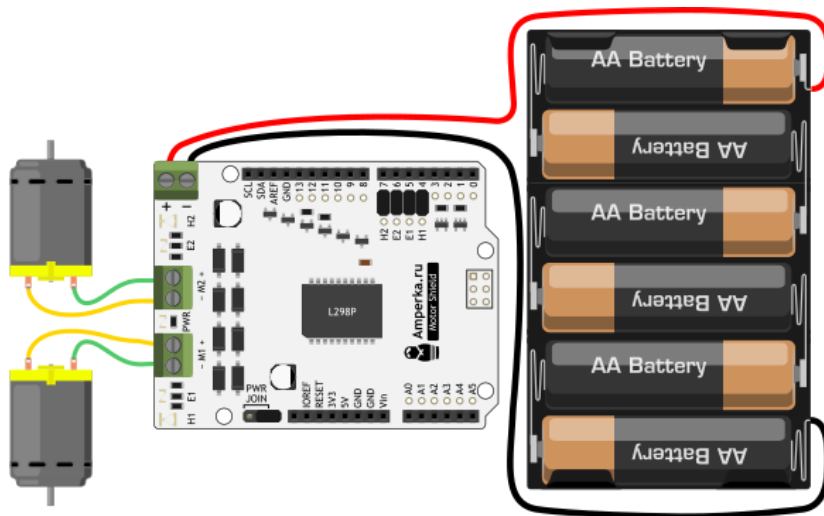
Подключите внешнее питание для моторов через клеммник "PWR".

## Примеры работы для Arduino

### Управление коллекторными двигателями

Подключите два коллекторных мотора [https://amperka.ru/collection/dc-motors?utm\_source=man&utm\_campaign=arduino-motor-shield&utm\_medium=wiki] к клеммникам M1 и M2 соответственно.

#### Схема устройства



#### Код программы

Для начала покрутим каждый мотор в одну, а затем другую сторону.

##### dc\_motors.ino

```
// подключите один мотор к клемме: M1+ и M1-
// а второй к клемме: M2+ и M2-
// Motor shield использует четыре контакта 4, 5, 6, 7 для управления моторами
// 4 и 7 — для направления, 5 и 6 — для скорости
#define SPEED_1      5
#define DIR_1        4

#define SPEED_2      6
#define DIR_2        7

void setup() {
  // настраиваем выводы платы 4, 5, 6, 7 на вывод сигналов
  for (int i = 4; i < 8; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  // устанавливаем направление мотора «M1» в одну сторону
  digitalWrite(DIR_1, LOW);
  // включаем мотор на максимальной скорости
  analogWrite(SPEED_1, 255);
  // ждём одну секунду
  delay(1000);

  // устанавливаем направление мотора «M1» в другую сторону
  digitalWrite(DIR_1, HIGH);
  // ждём одну секунду
  delay(1000);
  // выключаем первый мотор
  analogWrite(SPEED_1, 0);

  // устанавливаем направление мотора «M2» в одну сторону
  digitalWrite(DIR_2, LOW);
  // включаем второй мотор на максимальной скорости
  analogWrite(SPEED_2, 255);
  // ждём одну секунду
  delay(1000);

  // устанавливаем направление мотора «M2» в другую сторону
  digitalWrite(DIR_2, HIGH);
  // ждём одну секунду
  delay(1000);

  // выключаем второй мотор
  analogWrite(SPEED_2, 0);
  // ждём одну секунду
  delay(1000);
}
```

**Код программы**

Усложним задачу. Будем плавно увеличивать скорость первого мотора до максимальной скорости, а потом понижать до полного выключения. Аналогично проделываем со вторым мотором.

**dc\_motors\_pwm.ino**

```
// подключите один мотор к клемме: M1+ и M1-
// а второй к клемме: M2+ и M2-
// Motor shield использует четыре контакта 4, 5, 6, 7 для управления моторами
// 4 и 7 — для направления, 5 и 6 — для скорости
#define SPEED_1      5
#define DIR_1        4

#define SPEED_2      6
#define DIR_2        7

void setup() {
  // настраиваем выходы платы 4, 5, 6, 7 на вывод сигналов
  for (int i = 4; i < 8; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  // устанавливаем направление мотора «M1» в одну сторону
  digitalWrite(DIR_1, LOW);
  // медленно разгоняем мотор
  for (int i = 0; i <= 255; i++) {
    analogWrite(SPEED_1, i);
    Serial.println(i);
    delay(20);
  }

  // устанавливаем направление мотора «M1» в другую сторону
  digitalWrite(DIR_1, HIGH);
  // медленно тормозим мотор
  for (int i = 255; i >= 0; i--) {
    analogWrite(SPEED_1, i);
    Serial.println(i);
    delay(20);
  }

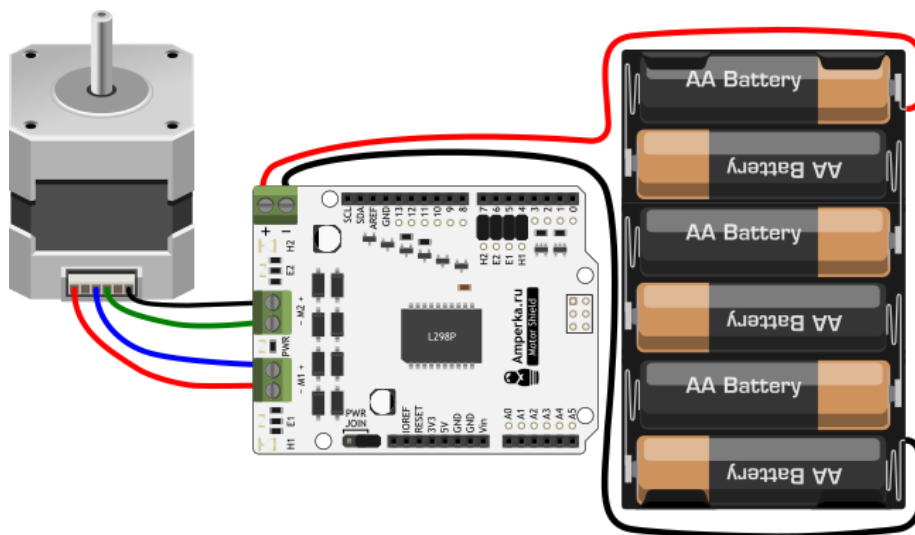
  // устанавливаем направление мотора «M2» в одну сторону
  digitalWrite(DIR_2, LOW);
  // медленно разгоняем мотор
  for (int i = 0; i <= 255; i++) {
    analogWrite(SPEED_2, i);
    delay(20);
  }

  // устанавливаем направление мотора «M2» в другую сторону
  digitalWrite(DIR_2, HIGH);
  // медленно тормозим мотор
  for (int i = 255; i > 0; i--) {
    analogWrite(SPEED_2, i);
    delay(20);
  }
}
```

**Управление биполярным шаговым двигателем**

Драйвер моторов может на себя также взять управления шаговым двигателем. В качестве примера подключим шаговый двигатель 42STH47-0406A [[https://amperka.ru/product/stepper-motor-nema17-42sth47-0406a?utm\\_source=man&utm\\_campaign=arduino-motor-shield&utm\\_medium=wiki](https://amperka.ru/product/stepper-motor-nema17-42sth47-0406a?utm_source=man&utm_campaign=arduino-motor-shield&utm_medium=wiki)].

**Схема устройства**



### Управление без библиотек

Motor Shield поддерживает три режима управления биполярным шаговым двигателем:

- Однофазный полношаговый «wave drive mode»
- Двухфазный полношаговый «full step mode»
- Полушаговый «half step mode»

### Код программы

Протестируем по очереди три режима управления.

#### step-motor-switch-mode.ino

```
// пины для управления Motor Shield
#define E1      5
#define E2      6
#define H1      4
#define H2      7

// три режима управления шаговым мотором
// однофазный режим
#define WAVE_DRIVE 0
// двухфазный режим
#define FULL_STEP 1
// полушаговый режим
#define HALF_STEP 2

// задержка между переключением обмоток
// определяет скорость вращения
int delayTime = 10;

unsigned long startTime;

// функция выставляет значение на пинах
void doStep(bool E1State, bool H1State, bool E2State, bool H2State) {
  digitalWrite(E1, E1State);
  digitalWrite(H1, H1State);
  digitalWrite(E2, E2State);
  digitalWrite(H2, H2State);
  delay(delayTime);
}

// определяем комбинации управляющих импульсов в зависимости от режима
void rotate(byte rotateMode) {
  switch (rotateMode) {
    case WAVE_DRIVE:
      doStep(1, 1, 0, 0);
      doStep(0, 0, 1, 1);
      doStep(1, 0, 0, 0);
      doStep(0, 0, 1, 0);
      break;

    case FULL_STEP:
      doStep(1, 1, 1, 1);
      doStep(1, 0, 1, 1);
      doStep(1, 0, 1, 0);
      doStep(1, 1, 1, 0);
      break;

    case HALF_STEP:

```

```

        doStep(1, 1, 0, 0);
        doStep(1, 1, 1, 1);
        doStep(0, 0, 1, 1);
        doStep(1, 0, 1, 1);

        doStep(1, 0, 0, 0);
        doStep(1, 0, 1, 0);
        doStep(0, 0, 1, 0);
        doStep(1, 1, 1, 0);
        break;
    }
}

void setup() {
    // настраиваем ножки на выход
    for (int i = 4; i < 8; i++) {
        pinMode(i, OUTPUT);
    }
    // запоминаем текущее время
    startTime = millis();
}

void loop() {
    // узнаём время прошедшее с начала работы в миллисекундах
    // и переводим его в секунды
    unsigned long loopTime = (millis() - startTime) / 1000;
    // меняем режим каждую секунду
    rotate(loopTime % 3);
}

```

Скорость вращения шагового двигателя очень сильно влияет на развиваемый мотором момент. Убедитесь сами. Запустите этот же пример с разными значениями `delayTime`.

Обратите внимание, что двигатель в однофазном полношаговом режиме позволяет развить гораздо меньший момент, чем в двухфазном полношаговом режиме.

### Управление через готовую библиотеку

Для лёгкого и быстрого управления шаговым двигателем мы написали библиотеку `AmperkaStepper` [<https://github.com/amperka/AmperkaStepper>]. Она скрывает в себе все тонкости работы с мотором и предоставляет удобные методы.

### Пример работы в различных режимах

#### step-motor-test.ino

```

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

// создаём объект для работы с шаговым двигателем
// передаём фиксированное количество шагов за полный оборот
// смотреть в характеристиках двигателя
AmperkaStepper motor(200);

// при использовании Motor Shield на нестандартных пинах
// необходимо передавать управляющие пины
// Например, для пинов 8, 9, 10, 11
// AmperkaStepper motor(200, 8, 9, 10, 11);

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);
}

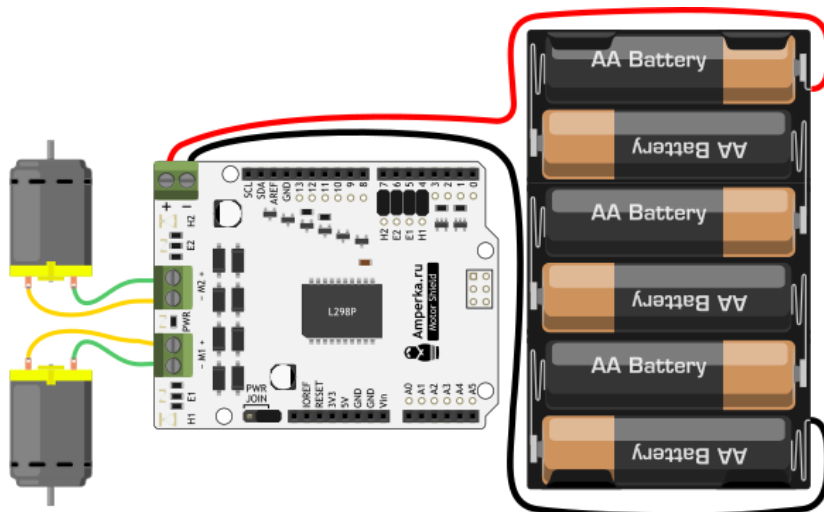
```

## Пример работы для Espruino

### Управление коллекторными двигателями

Подключите два коллекторных мотора [https://amperka.ru/collection/dc-motors?utm\_source=man&utm\_campaign=arduino-motor-shield&utm\_medium=wiki] к клеммникам M1 и M2 соответственно.

### Схема устройства



### Код программы

Для начала покрутим каждый мотор в одну, а затем другую сторону.

#### dc\_motors.js

```
// подключаем библиотеку «motor»
var Motor = require('@amperka/motor');
// подключаем первый мотор канала M1 на Motor Shield
var motorOne = Motor.connect(Motor.MotorShield.M1);
// подключаем второй мотор канала M2 на Motor Shield
var motorTwo = Motor.connect(Motor.MotorShield.M2);

// Альтернативный способ подключения с указанием пинов
// var myMotor = Motor.connect({phasePin: P4, pwmPin: P5, freq: 100});

// интервал времени
var time = 1000;
// счётчик
var state = 0;

// каждую секунду меняем режим работы
setInterval(() => {
  // останавливаем моторы
  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);
```

### Код программы

Усложним задачу. Будем плавно увеличивать скорость первого мотора до максимальной скорости, а потом понижать до полного выключения. Аналогично проделываем со вторым мотором.

#### dc\_motors\_pwm.js

```
// подключаем библиотеку «motor»
var Motor = require('@amperka/motor');
// подключаем первый мотор канала M1 на Motor Shield
var motorOne = Motor.connect(Motor.MotorShield.M1);
// подключаем второй мотор канала M2 на Motor Shield
var motorTwo = Motor.connect(Motor.MotorShield.M2);

// создаём объект анимация
// для плавного изменения параметров вращения мотора
var animMotor = require('@amperka/animation').create({
  // начальное значение
  // при «0» мотор стоит
```

```

from: 0,
// конечное значение
// при «1» мотор вращается с максимальной скоростью в одну сторону
to: 1,
// продолжительность полного перехода
// за 5 секунд мотор пройдёт диапазон значений от нуля до единицы
duration: 5,
// шаг обновления: каждые 20 мс
updateInterval: 0.02
}).queue({
// после завершения перехода, выполняем ещё одну операцию
// начальное значение
// при «-1» мотор вращается с максимальной скоростью в обратную сторону
from: -1,
// конечное значение
// при «0» мотор стоит
to: 0,
// продолжительность полного перехода
// за 5 секунд мотор пройдёт диапазон значений от минус единицы до нуля
duration: 5
});

// номер мотора
var motorNum = motorOne;
// время вращения работы мотора
var time = 10000;

// обработчик анимации
animMotor.on('update', function(val) {
  motorNum.write(val);
});

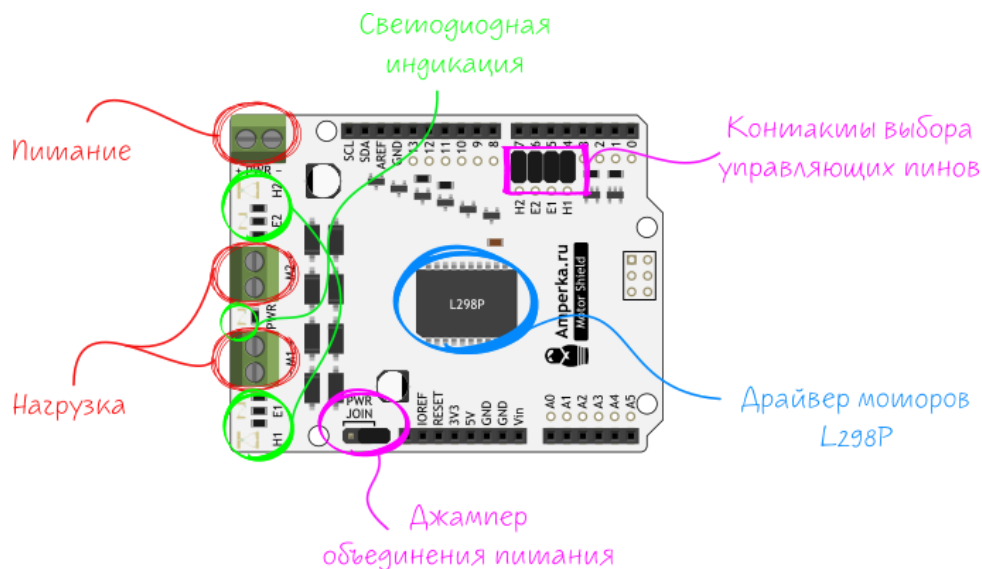
// функция работы первого мотора
var startMotorOne = () => {
  motorNum = motorOne;
  animMotor.play();
  setTimeout(() => {
    startMotorTwo();
  }, time);
};

// функция работы второго мотора
var startMotorTwo = () => {
  motorNum = motorTwo;
  animMotor.play();
  setTimeout(() => {
    startMotorOne();
  }, time);
};

// запускаем функцию работы первого мотора
startMotorOne();

```

## Элементы платы

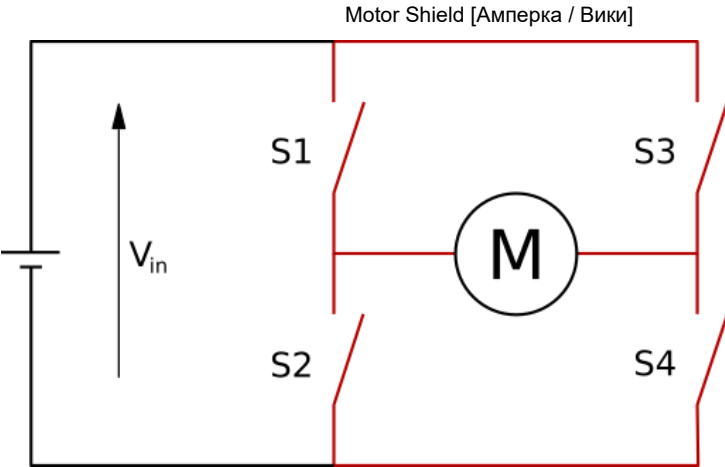


## Драйвер двигателей

Сердце и мускулы платы — микросхема двухканального Н-моста L298P.

Термин «Н-мост» появился благодаря графическому изображению схемы, напоминающему букву «Н». Н-мост состоит из четырёх ключей.





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 транзисторы для регулировки скорости мотора с помощью ШИМ-сигнала.

Н-мост с силовыми ключами — основная начинка микросхемы L298P для управления скоростью и направлением двигателей.

Питание

На плате расширения Motor Shield два контура питания.

- Силовой контур — напряжение для питания моторов от силовой части микросхемы Н-моста L298P. При отсутствии питания на силовом контуре, светодиод `PWR` не горит.
- Цифровой контур — для питания вспомогательной цифровой логики управления микросхемой L298P и светодиодов индикации. Цифровое питание поступает на плату расширения от пина `5v` управляющей платы. Если по какой-то причине напряжения на этом пине не оказалось, светодиоды индикации гореть не будут.

Если отсутствует хотя бы один из контуров питания — Motor Shield работать не будет.

Силовое питание подключается через клеммник `PWR`. Диапазон входного напряжения:

- 5–24 В при снятом джампере `PWR JOIN`
- 5–12 В при установленном джампере `PWR JOIN`

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

Джампер объединения питания

Motor Shield использует два контура питания: силовое и цифровое. По умолчанию для питания всей конструкции необходимо два источника напряжения:

- Питания для силовой части через клеммник `PWR`
- Питания для цифровой части с пина `5v` управляющей платформы.

При установки джампера в положение `PWR JOIN`, происходит объединение контакта `Vin` управляющей платформы и положительного контакта клеммника `PWR`. Режим объединённого питания позволяет запитывать управляющую платформу и силовую часть драйвера от одного источника питания.

При объединённом режиме напряжение может быть подано двумя способами:

- на клеммник PWR драйвера моторов
- на внешний разъём питания управляющей платы

При работе двигателей по цепи питания может проходить очень большой ток, на который цепь vin управляющей платформы не рассчитана. Поэтому выбор для питания клеммника PWR предпочтительнее.

В режиме совместного питания управляющей платформы и Motor Shield, используйте входное напряжение в диапазоне 7–12 вольт. Напряжение более 12 вольт убьёт управляющую платформу. Если вы хотите работать с Motor Shield в диапазоне 5–24 вольт, используйте два отдельных источника питания.

Источник питания должен быть способен обеспечить стабильное напряжение при резких скачках нагрузки. Даже кратковременная просадка напряжения может привести к перезагрузке управляющей платформы. В итоге программа начнётся сначала и поведения двигателей будет неадекватным.

При объединённом питании используйте литий-ионные [https://ru.wikipedia.org/wiki/Литий-ионный\_аккумулятор] и никель-металлгидридные аккумуляторы [https://ru.wikipedia.org/wiki/Никель-металлгидридный\_аккумуляторы]. Если вы используете другие источники питания, лучше воспользуйтесь отдельной схемой питания управляющей платформы и Motor Shield.

Нагрузка

Нагрузка разделена на два независимых канала. К каждому каналу можно подключить один коллекторный мотор. Первый канал на плате обозначен шёлком m1, второй канал — m2.

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

Светодиодная индикация

Имя светодиода	Назначение
PWR	Индикация состояния силового питания. Есть питание — светодиод горит, нет питания — не горит.
H1	Индикация состояния направления первого канала m1. При высоком логическом уровне светится зелёным светом, при низком — красным.
H2	Индикация состояния направления второго канала m2. При высоком логическом уровне светится зелёным светом, при низком — красным.
E1	Индикация скорости первого канала m1. Яркость светодиода пропорциональна скорости вращения двигателя.
E2	Индикация скорости второго канала m2. Яркость светодиода пропорциональна скорости вращения двигателя.

Пины управления скоростью вращения двигателей

Для запуска двигателя на первом или втором канале установите высокий уровень на пинах скорости E1 или E2 соответственно. Для остановки моторов установите на соответствующих пинах низкий уровень.

Управление скоростью происходит при помощи ШИМ, за счёт быстрого включения и выключения нагрузки.

Пины управления направлением вращения двигателей

Пины направления h1 или h2 отвечают за направление вращения двигателей. Смена направления вращения коллекторных двигателей достигается за счёт изменения полярности приложенного к ним напряжения.

- Если выставить на пин направления высокий уровень, то полярность напряжения на клеммниках нагрузки будет соответствовать обозначению «+» и «-» на плате.
- При подаче на пин низкого уровня, напряжение на клеммнике изменится на противоположное.

Контакты выбора управляющих пинов

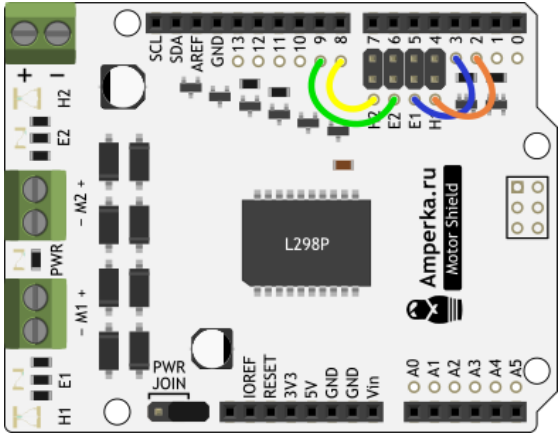
По умолчанию Motor Shield для управления скоростью и направлением вращения моторов использует пины управляющей платы:

Назначение	Канал 1	Канал 2
Скорость	5	6
Направление	4	7

Если в вашем устройстве эти пины уже заняты, например используются для управления Relay Shield [https://amperka.ru/product/arduino-relay-shield?utm\_source=man&utm\_campaign=arduino-motor-shield&utm\_medium=wiki]), вы можете использовать другой свободный пин.

Контакты отвечающие за направления — n1 и n2 можно заменять на любые другие. А вот контакты управляющие скоростью моторов — e1 и e2, можно перебрасывать только на пины с поддержкой ШИМ.

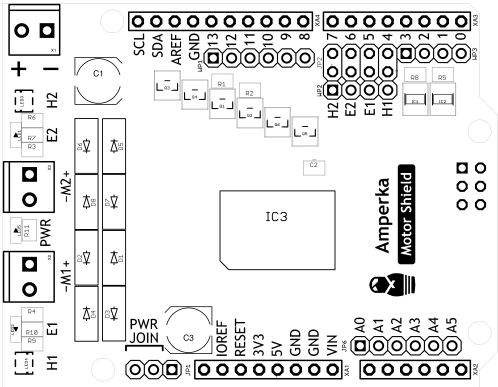
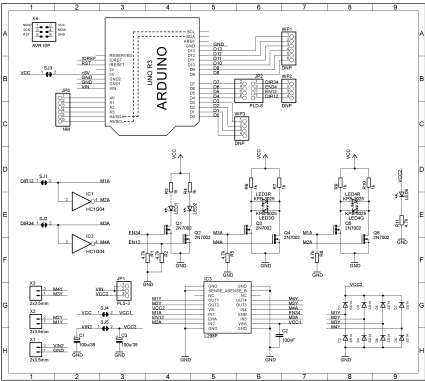
Для переброски пинов снимите джампер напротив занятого пина и припаяйте проводок между луженым отверстием рядом со снятым джампером и луженым отверстием напротив нужного пина.



На этой картинке мы перекинули:

- управляющие пины первого канала n1 и e1: с 4 и 5 пина — на 2 и 3 пин соответственно;
- управляющие пины второго канала n2 и e2: с 7 и 6 — на 8 и 9 пин соответственно.

Принципиальная и монтажная схемы



Характеристики

- Драйвер моторов: L298P
- Количество подключаемых моторов: 2
- Напряжение логической части: 3,3—5 В
- Напряжение силовой части:
  - 5—12 с PWR JOIN
  - 5—24 без PWR JOIN
- Продолжительный ток нагрузки на канал: до 2 А

- 
- Пиковый ток нагрузки на канал: до 3 А в течении 100 мкс

Для достижения максимума используйте дополнительное охлаждение для микросхемы L298P.

## Ресурсы

- Motor Shield [[https://amperka.ru/product/arduino-motor-shield?utm\\_source=man&utm\\_campaign=motor-shield&utm\\_medium=wiki](https://amperka.ru/product/arduino-motor-shield?utm_source=man&utm_campaign=motor-shield&utm_medium=wiki)] в магазине.
- Векторное изображение драйвера моторов [<https://github.com/amperka/hardware-drawings/blob/master/arduino-motor-shield.svg>]
- Datasheet на микросхему L298P
- Библиотека для работы с шаговым двигателем при помощи Motor Shield [<https://github.com/amperka/AmperkaStepper>]
- Библиотека для Espruino по работе с коллекторными моторами