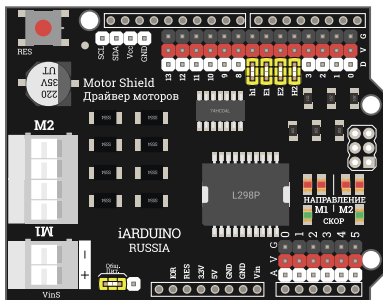


Motor Shield (2 канала)



Общие сведения:

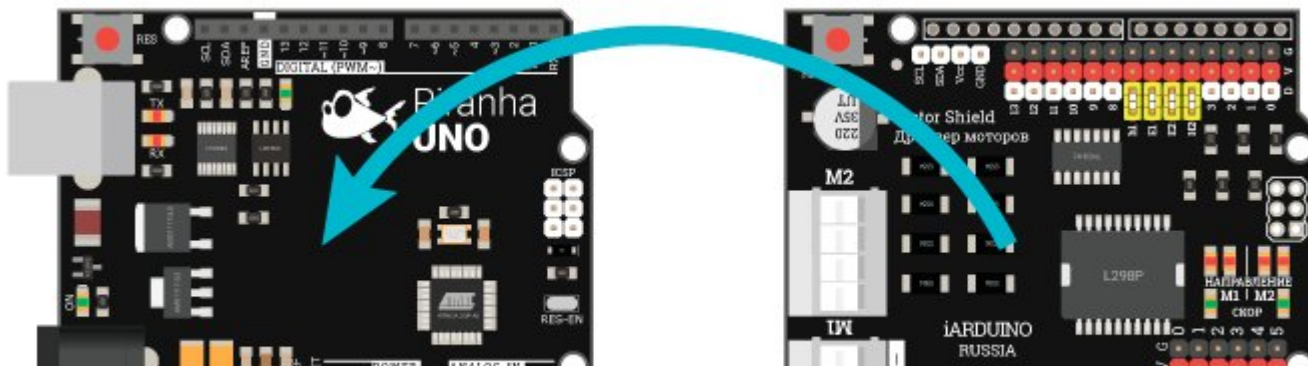
[*Motor Shield \(2 канала\)*](#) - используется для управления до двух [коллекторных моторов постоянного тока](#) или одним биполярным [шаговым двигателем](#).

Спецификация:

- Напряжение питания на входе клеммника Vin: 2,5 ... 35 В.
- Ток потребляемый логической частью: до 36 мА.
- Общий ток нагрузки на выходах: до 2 А (без ШИМ), при этом чип L298 нагревается выше 60 °С.
- Частота коммутаций: до 40 кГц.
- Задержка между сигналами на входе и выходе: до 3 мкс.
- Рабочая температура: -25 ... 130 °С.
- Температура хранения: -40 ... 150 °С.

Подключение:

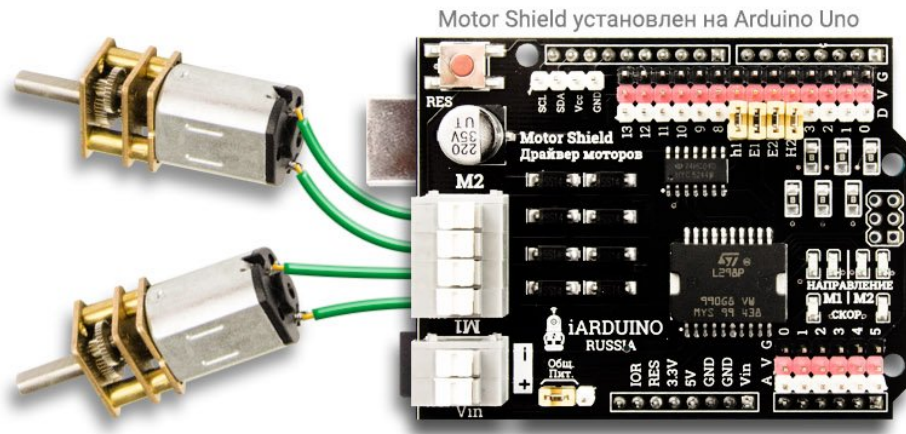
[Motor Shield](#) устанавливается на [Arduino UNO](#) и управляется через 4 входа: N1, E1, E2, N2, они соединены перемычками с цифровыми выводами: D5 - D7. Если требуется использовать другие выводы [Arduino](#), то нужно снять перемычку и соединить освободившийся вход с нужным выводом [Arduino](#).



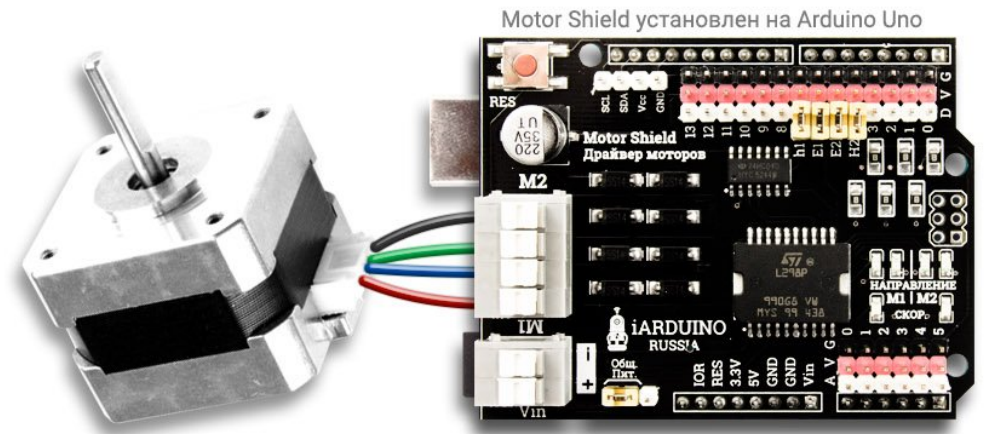


Моторы постоянного тока подключаются следующим образом: первый, к клеммнику M1 (управляется через входы H1 и E1), второй, к клеммнику M2 (управляется через входы H2 и E2). Можно подключить только один мотор, к любому из клеммников.

Биполярный шаговый двигатель подключается следующим образом: выводы одной обмотки к клеммнику M1, выводы второй обмотки к клеммнику M2. (управляется через входы H1 и H2, а на входы E1 и E2 требуется подать уровень логической «1»).



Подключение моторов постоянного тока



Подключение биполярного шагового двигателя

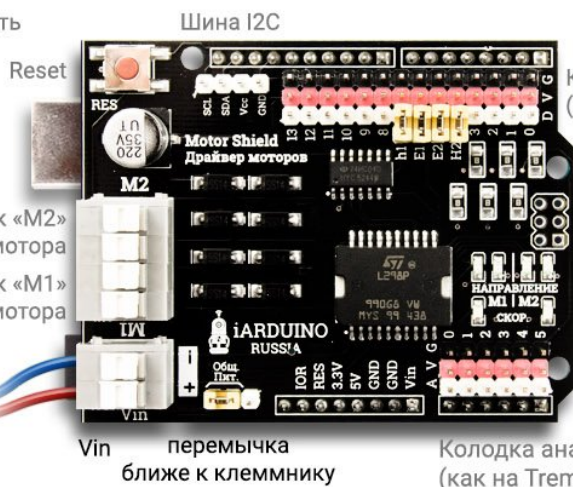
Питание:

Выходы [Motor Shield](#) работают только от внешнего источника питания 2,5 - 35 В, который должен подключаться к клеммнику питания Vin. Рядом с клеммником питания Vin расположены три контакта с переключкой.

- **Общее питание:** если переключка замыкает контакты расположенные ближе к клеммнику «Общ. Пит.», то питание с [Motor Shield](#) подаётся на вход [Arduino](#) «Vin» (общее питание берется с [Motor Shield](#)). Не требуется отдельного питания [Arduino](#).
В данном режиме, напряжение на входе клеммника Vin Motor Shield, не должно превышать 12 В.
- **Раздельное питание:** Если переключка расположена на контактах дальше от клеммника, или отсутствует, то питание должно подаваться и на [Arduino](#) и на [Motor Shield](#) (раздельное питание).

В данном режиме, напряжение на входе клеммника Vin Motor Shield, может достигать 35 В.

К Motor Shield можно подключить два коллекторных мотора, или один шаговый двигатель.



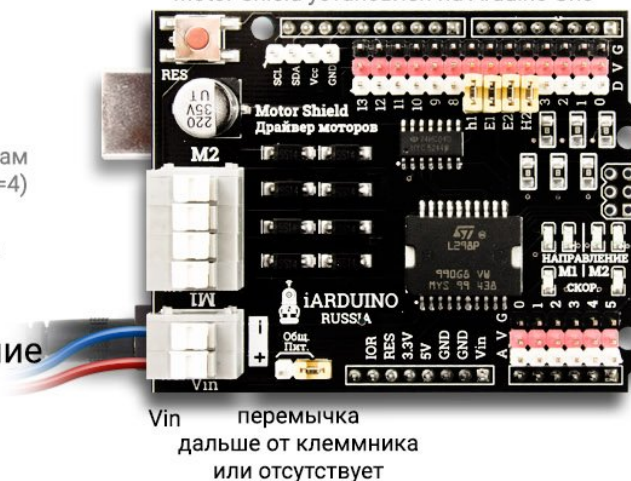
Клеммник «M2» для подключения второго мотора
Клеммник «M1» для подключения первого мотора

Общее питание

Vin переключатель ближе к клеммнику

Колонка аналоговых выводов Arduino (как на Trema Shield)

Motor Shield установлен на Arduino Uno



Раздельное питание

Vin переключатель дальше от клеммника или отсутствует

Подробнее о Motor Shield:

[Motor Shield](#) построен на базе чипа L298, позволяет использовать любые входы [Arduino](#) (по умолчанию D5-D7), а также на нем разведены аналоговые и цифровые выводы, для удобства подключения [других модулей](#), как в [Trema Shield](#). Позволяет управлять, как скоростью движения, так и направлением подключённых двигателей.

Управление моторами постоянного тока: осуществляется при помощи функций: `digitalWrite()` и `analogWrite()`.

- Управление первым [мотором постоянного тока](#) M1, осуществляется через вход H1 (направление) и E1 (скорость).
- Управление вторым [мотором постоянного тока](#) M2, осуществляется через вход H2 (направление) и E2 (скорость).
- Направление устанавливается подачей логического уровня LOW или HIGH.
- Скорость устанавливается подачей сигнала ШИМ, коэффициент заполнения которого, прямо пропорционален скорости.

Управление биполярным шаговым двигателем: осуществляется при помощи стандартной библиотеки «Stepper».

- [Motor Shield](#) использует два вывода, H1 и H2, для управления [шаговым двигателем](#).
- На выводы, E1 и E2, требуется подать уровень логической «1» (программно или аппаратно).

Примеры:

Управление моторами постоянного тока:

```
const uint8_t pinH1   = 7;           // Создаём константу указывая номер вывода H1 MotorShield (он упр
const uint8_t pinE1   = 6;           // Создаём константу указывая номер вывода E1 MotorShield (он упр
const uint8_t pinE2   = 5;           // Создаём константу указывая номер вывода E2 MotorShield (он упр
const uint8_t pinH2   = 4;           // Создаём константу указывая номер вывода H2 MotorShield (он упр
    uint8_t mSpeed    = 0;           // Создаём переменную для хранения скорости моторов
    bool     mDirect   = HIGH;       // Создаём переменную для хранения направления моторов
void setup(){
    pinMode(pinH1, OUTPUT); digitalWrite(pinH1, LOW); // Конфигурируем вывод pinH1 как выход и устанавливаем на нём уро
    pinMode(pinE1, OUTPUT); digitalWrite(pinE1, LOW); // Конфигурируем вывод pinE1 как выход и устанавливаем на нём уро
    pinMode(pinE2, OUTPUT); digitalWrite(pinE2, LOW); // Конфигурируем вывод pinE2 как выход и устанавливаем на нём уро
    pinMode(pinH2, OUTPUT); digitalWrite(pinH2, LOW); // Конфигурируем вывод pinH2 как выход и устанавливаем на нём уро
}
void loop(){
    /*M1*/          digitalWrite(pinH1, mDirect );           // Устанавливаем направление
    while(mSpeed<255){analogWrite(pinE1, mSpeed++); delay(5);} // Увеличиваем скорость
    while(mSpeed> 0){analogWrite(pinE1, mSpeed--); delay(5);} // Уменьшаем скорость
    /*M2*/          digitalWrite(pinH2, mDirect );           // Устанавливаем направление
    while(mSpeed<255){analogWrite(pinE2, mSpeed++); delay(5);} // Увеличиваем скорость
    while(mSpeed> 0){analogWrite(pinE2, mSpeed--); delay(5);} // Уменьшаем скорость
    mDirect=!mDirect;                                       // Меняем направление
}
```

В этом примере, два мотора, поочерёдно, плавно увеличивают и уменьшают свою скорость, после чего меняют направление.

Управление шаговым двигателем:

```
#include <Stepper.h>           // Подключаем библиотеку, для работы с шаговыми двигателями
const uint8_t pinH1          = 7;           // Создаём константу указывая номер вывода H1 MotorShield
```



```

const uint8_t pinE1    = 6;           // Создаём константу указывая номер вывода E1 MotorShield
const uint8_t pinE2    = 5;           // Создаём константу указывая номер вывода E2 MotorShield
const uint8_t pinH2    = 4;           // Создаём константу указывая номер вывода H2 MotorShield
Stepper          motor(200, pinH1, pinH2); // Создаём объект для работы с библиотекой Stepper
                                           // Указывая количество шагов в полном обороте двигателя (200) и н

void setup(){
  pinMode(pinE1, OUTPUT); digitalWrite(pinE1, HIGH); // Конфигурируем вывод pinE1 как выход и устанавливаем на нём уро
  pinMode(pinE2, OUTPUT); digitalWrite(pinE2, HIGH); // Конфигурируем вывод pinE2 как выход и устанавливаем на нём уро
  motor.setSpeed(150); // Указываем скорость вращения двигателя (150 об/мин)
}
void loop(){
  motor.step( 200); delay(1000); // Выполняем поворот ротора на 200 шагов, в одну сторону, после
  motor.step(-200); delay(1000); // Выполняем поворот ротора на 200 шагов, в другую сторону, после
}

```

В этом примере, шаговый двигатель делает полный оборот, в одну и другую сторону, с интервалом в 1 секунду.

Применение:

- Управление [моторами постоянного тока](#)
- Управление биполярным [шаговым двигателем](#)