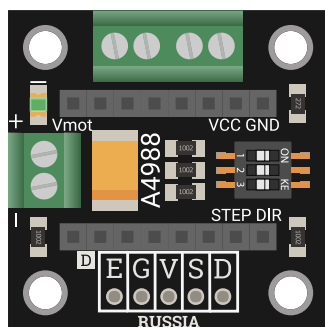


Адаптер драйвера шагового двигателя (Трема-модуль v2.0)



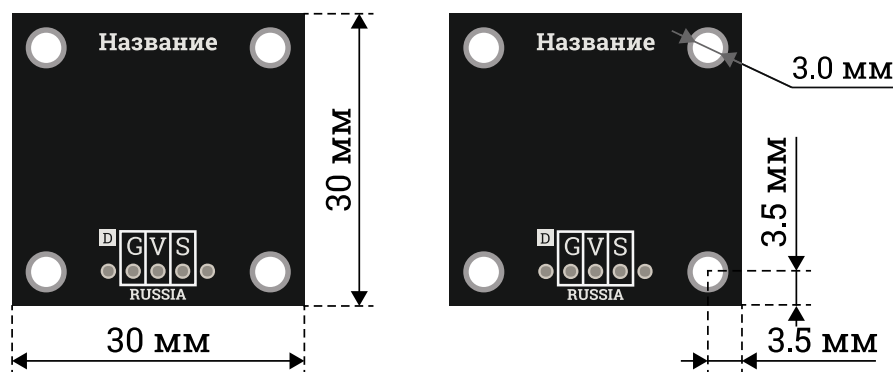
Общие сведения:

[Трема-модуль Адаптер драйвера шагового двигателя](#) — используется для подключения драйвера шагового двигателя ([DRV8825](#) или [A4988](#)) и управления [шаговыми двигателями](#)(ШД).

Спецификация:

- Питание логики: 3.3 - 5В;
- Питание ШД: до 30В;
- Настройка размера шага переключателями на плате;
- Габариты: 30x30 мм.

Все модули линейки "Trema" выполнены в одном формате



Подключение:

[Trema-модуль Адаптер драйвера шагового двигателя](#) входит в линейку [Trema-модулей](#), что позволяет подключить его к [Arduino](#) через [Trema Shield](#) по одному 3-проводному и двум 1-проводным шлейфам (идёт в комплекте) без пайки, без дополнительных проводов и переходников, его можно подключать к цифровому выводу [Arduino](#).

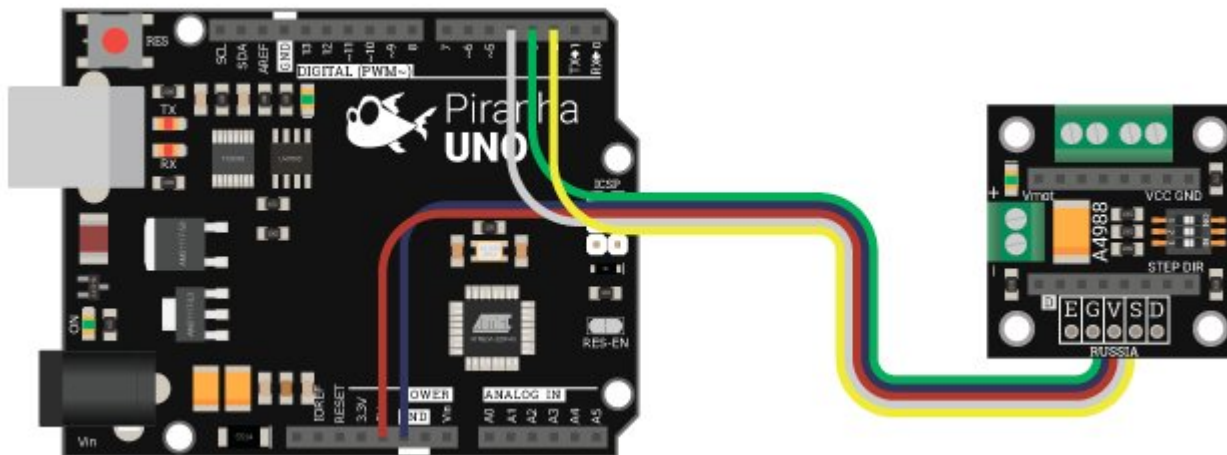
Адаптер драйвера шагового двигателя подключается к 3 любым цифровым выводам [Arduino](#).

| Выводы адаптера | Значение | Выводы Arduino |
|-----------------|--|----------------------|
| E (Enable) | Разрешение чипу начать работу. Инверсный вывод (0 - разрешить, 1 - запретить); | Любой цифровой вывод |
| G (GND) | Земля | GND |
| V (VCC) | Питание логики | 3V3/5V |
| S (STEP) | Подача импульсного сигнала шаговому двигателю для поворота (1 импульс - 1 шаг) | Любой цифровой вывод |
| D (DIR) | Указание направления вращения шагового двигателя (0 или 1) | Любой цифровой вывод |

Модуль удобно подключать 3 способами, в зависимости от ситуации:

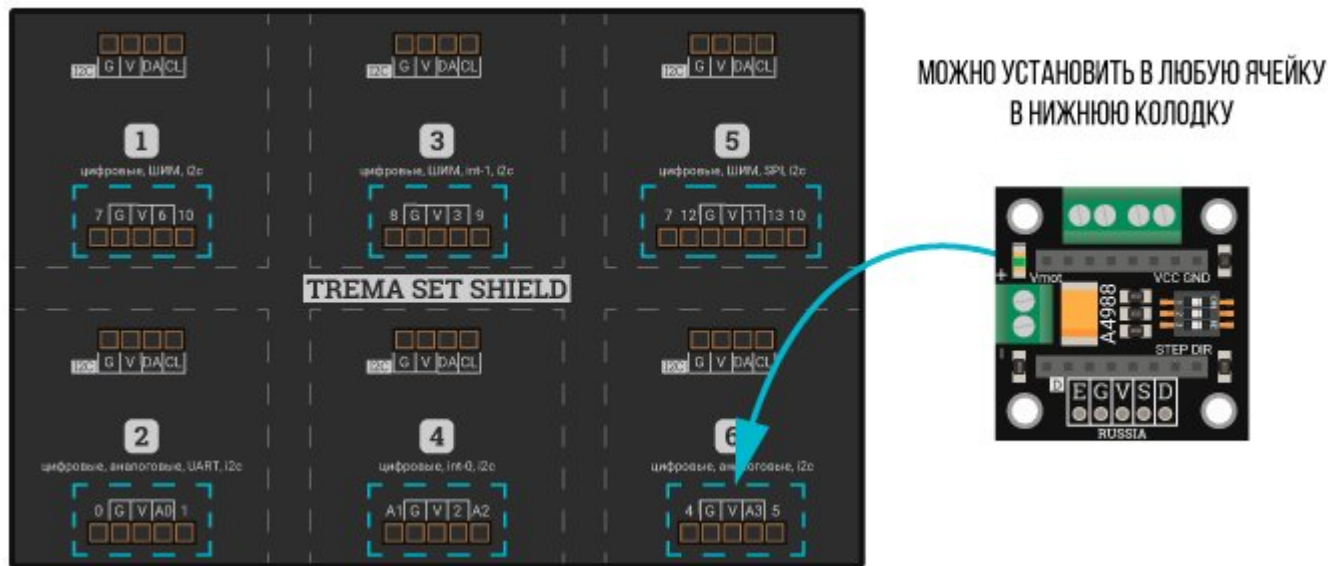
Способ - 1 : Используя проводной шлейф и Piranha UNO

Используя провода «Папа – Мама», подключаем напрямую к контроллеру Piranha UNO



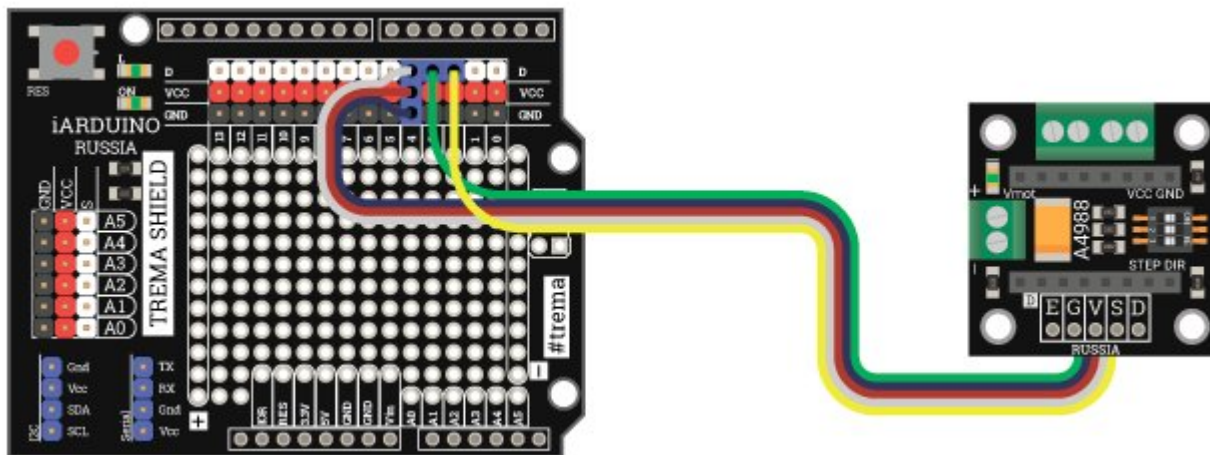
Способ - 2 : Используя Trema Set Shield

Модуль можно подключить к любому из цифровых входов Trema Set Shield.



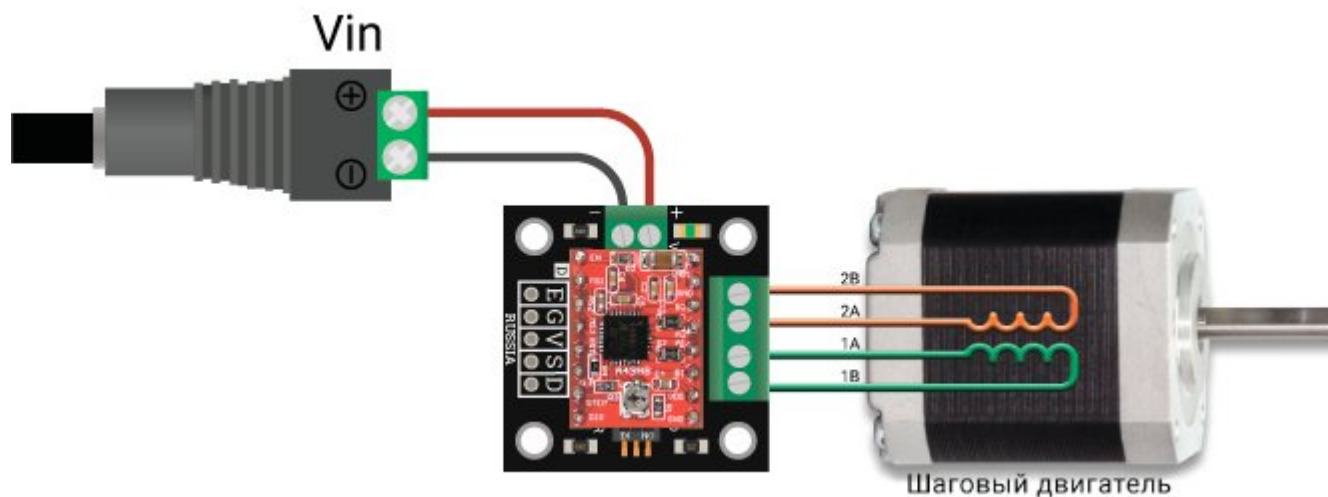
Способ - 3 : Используя проводной шлейф и Shield

Используя 5-проводной шлейф, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.

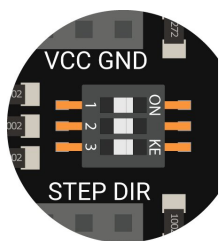


Подключение шагового двигателя к адаптеру:

После установки [драйвера мотора](#) в адаптер подключите питание ШД и сам ШД согласно следующей схеме:



Настройка размера шага двигателя:



На плате имеется блок переключек для регулировки размера шага двигателя.

Для каждого драйвера таблица шагов будет своя, в зависимости от модели драйвера.

Ниже приведены таблицы настройки размера шага для драйверов [DRV8825](#) и [A4988](#).

Драйвер A4988:

| Размер шага | MS-3 | MS-2 | MS-1 |
|-------------|------|------|------|
| Полный шаг | 0 | 0 | 0 |
| 1/2 шага | 0 | 0 | 1 |
| 1/4 шага | 0 | 1 | 0 |

| Размер шага | MS-3 | MS-2 | MS-1 |
|-------------|------|------|------|
| 1/8 шага | 0 | 1 | 1 |
| 1/16 шага | 1 | 1 | 1 |

Драйвер DRV8825:

| Размер шага | MS-3 | MS-2 | MS-1 |
|-------------|------|------|------|
| Полный шаг | 0 | 0 | 0 |
| 1/2 шага | 0 | 0 | 1 |
| 1/4 шага | 0 | 1 | 0 |
| 1/8 шага | 0 | 1 | 1 |
| 1/16 шага | 1 | 0 | 0 |
| 1/32 шага | 1 | 0 | 1 |
| 1/32 шага | 1 | 1 | 0 |
| 1/32 шага | 1 | 1 | 1 |

Питание:

Адаптер драйвера шагового двигателя имеет колодку для подключения внешнего питания (для шагового двигателя) согласно указанной полярности, а так же выводы G и V для питания логики, которые подключаются соответственно к выводам GND и 5V Arduino.

Примеры:

Управление вращением шагового двигателя с помощью потенциометра.

Для управления [шаговым двигателем](#) (ШД) понадобится [драйвер мотора](#). В данном случае используется [драйвер мотора A4988](#).

Устанавливаем его в адаптер, а после этого подключаем адаптер и [потенциометр](#) к Arduino.

Адаптер драйвера шагового двигателя:

- E - цифровой вывод 7;
- G - GND;
- V - 5V;
- S - цифровой вывод 6;
- D - цифровой вывод 10;

Потенциометр:

- G - GND;
- V - 5V;
- S - A0;

```
const int8_t Step    = 6;           // импульсы вращения двигателя
const int8_t Dir     = 10;          // направление вращения двигателя
const int8_t Enable  = 7;           // разрешение работы чипу драйвера ШД
const int8_t Potenc  = A0;          // потенциометр
int16_t move_delay;                 // временная задержка между шагами мотора
int16_t val_potenc;                 // Переменная для хранения значения потенциометра

void setup() {
  pinMode(Potenc, INPUT);           // Настраиваем вывод Potenc на работу в режиме входа
  pinMode(Step, OUTPUT);            // Настраиваем вывод Step на работу в режиме выхода
  pinMode(Dir, OUTPUT);             // Настраиваем вывод Dir на работу в режиме выхода
  pinMode(Enable, OUTPUT);          // Настраиваем вывод Enable на работу в режиме выхода
  digitalWrite(Enable, LOW);        // Устанавливаем состояние на выходе равным 0 (LOW)
}
```

```

void loop() {
  val_potenc = analogRead(Potenc); // считываем значение потенциометра в переменную val_potenc
  if (val_potenc < 470 ) { // Если значение меньше заданного, то
    val_potenc = 470 - val_potenc; // определяем значение val_potenc
    digitalWrite(Dir, LOW); // Устанавливаем направление вращения (против часовой стрелки)
  } else
  if (val_potenc > 554) { // Если значение больше заданного, то
    val_potenc = val_potenc - 554; // определяем значение val_potenc
    digitalWrite(Dir, HIGH); // Устанавливаем направление вращения (по часовой стрелке)
  } else {
    val_potenc = 0; // Если значение не удовлетворяет ни одному условию, то обнуляем
  }
  if (val_potenc) { // Если значение val_potenc отлично от 0, то
    move_delay = map(val_potenc, 1, 470, 5000, 500); // переносим диапазон полученных значений с потенциометра в диапазон времени
    digitalWrite(Step, HIGH); delayMicroseconds(move_delay); // Подаём сигнал на вывод Step в течении времени move_delay;
    digitalWrite(Step, LOW); delayMicroseconds(move_delay); // Снимаем сигнал с вывода Step и ждём время move_delay;
  }
}

```

После загрузки скетча и подключения питания к двигателю, обратите внимание на потенциометр. При плавном повороте ручки потенциометра вправо двигатель будет плавно набирать скорость и вращаться по-часовой стрелке, а при плавном повороте ручки влево двигатель будет плавно набирать скорость и вращаться против часовой стрелки.

Управление вращением 2 шаговых двигателей с помощью джойстика.

Для управления [шаговым двигателем](#) (ШД) понадобится [драйвер мотора](#). В данном случае используется [драйвер мотора A4988](#).

Устанавливаем его в адаптер, а после этого подключаем адаптер и [джойстик](#) к Arduino:

Адаптер драйвера шагового двигателя:

- E - цифровой вывод 7;

- G - GND;
- V - 5V;
- S - цифровой вывод 6;
- D - цифровой вывод 10;

Джойстик:

- X - A2;
- K - не подключен;
- V - 5V;
- G - GND;
- Y - A1;

```

const int8_t Step1    = 6;           // импульсы вращения двигателя оси Y
const int8_t Step2    = 11;          // импульсы вращения двигателя оси X
const int8_t Dir1     = 10;          // направление вращения двигателя оси Y
const int8_t Dir2     = 13;          // направление вращения двигателя оси X
const int8_t Enable1  = 7;           // разрешение работы чипу драйвера ШД оси Y
const int8_t Enable2  = 12;          // разрешение работы чипу драйвера ШД оси X
const int8_t Potenc1  = A1;           // потенциометр оси Y
const int8_t Potenc2  = A2;           // потенциометр оси X
int16_t move_delay1;  // временная задержка между шагами мотора оси Y
int16_t move_delay2;  // временная задержка между шагами мотора оси X
int16_t val_potenc1;  // Переменная для хранения значения потенциометра оси Y
int16_t val_potenc2;  // Переменная для хранения значения потенциометра оси X

void setup() {
  pinMode(Potenc1, INPUT);           // Настраиваем вывод Potenc1 на работу в режиме входа
  pinMode(Potenc2, INPUT);           // Настраиваем вывод Potenc2 на работу в режиме входа
  pinMode(Step1, OUTPUT);             // Настраиваем вывод Step1 на работу в режиме выхода
  pinMode(Step2, OUTPUT);             // Настраиваем вывод Step2 на работу в режиме выхода

```

```

pinMode(Dir1, OUTPUT);
pinMode(Dir2, OUTPUT);
pinMode(Enable1, OUTPUT);
pinMode(Enable2, OUTPUT);
digitalWrite(Enable1, LOW);
digitalWrite(Enable2, LOW);
}

void loop() {
    val_potenc1 = analogRead(Potenc1);
    if (val_potenc1 < 470 ) {
        val_potenc1 = 470 - val_potenc1;
        digitalWrite(Dir1, LOW);
    } else
    if (val_potenc1 > 554) {
        val_potenc1 = val_potenc1 - 554;
        digitalWrite(Dir1, HIGH);
    } else {
        val_potenc1 = 0;
    }
    val_potenc2 = analogRead(Potenc2);
    if (val_potenc2 < 470 ) {
        val_potenc2 = 470 - val_potenc2;
        digitalWrite(Dir2, LOW);
    } else
    if (val_potenc2 > 554) {
        val_potenc2 = val_potenc2 - 554;
        digitalWrite(Dir2, HIGH);
    } else {
        val_potenc2 = 0;
    }
    if (val_potenc1) {
        move_delay1 = map(val_potenc1, 1, 470, 5000, 500);
}
// Настраиваем вывод Dir1 на работу в режиме выхода
// Настраиваем вывод Dir2 на работу в режиме выхода
// Настраиваем вывод Enable1 на работу в режиме выхода
// Настраиваем вывод Enable2 на работу в режиме выхода
// Устанавливаем состояние на выходе Enable1 равным 0 (LOW)
// Устанавливаем состояние на выходе Enable2 равным 0 (LOW)

// считываем значение потенциометра оси Y в переменную val_
// Если значение меньше заданного, то
// инвертируем значение val_potenc1, чтобы оно принимало зн
// Устанавливаем направление вращения (против часовой стрел
//
// Если значение больше заданного, то
// инвертируем значение val_potenc1, чтобы оно принимало зн
// Устанавливаем направление вращения (по часовой стрелке)
// Если значение не удовлетворяет ни одному условию, то
// обнуляем значение переменной val_potenc1

// считываем значение потенциометра оси X в переменную val_
// Если значение меньше заданного, то
// инвертируем значение val_potenc2, чтобы оно принимало зн
// Устанавливаем направление вращения (против часовой стрел
//
// Если значение больше заданного, то
// инвертируем значение val_potenc2, чтобы оно принимало зн
// Устанавливаем направление вращения (по часовой стрелке)
// Если значение не удовлетворяет ни одному условию, то
// обнуляем значение переменной val_potenc2

// Если значение val_potenc1 отлично от 0, то
// переносим диапазон полученных значений с потенциометра о

```

```
digitalWrite(Step1, HIGH); delayMicroseconds(move_delay1); // Подаём сигнал на вывод Step1 в течении времени move_delay1;
digitalWrite(Step1, LOW); delayMicroseconds(move_delay1); // Снимаем сигнал с вывода Step1 и ждём время move_delay1;
}
if (val_potenc2) { // Если значение val_potenc2 отлично от 0, то
  move_delay2 = map(val_potenc2, 1, 470, 5000, 1000); // переносим диапазон полученных значений с потенциометра о
  digitalWrite(Step2, HIGH); delayMicroseconds(move_delay2); // Подаём сигнал на вывод Step2 в течении времени move_delay2;
  digitalWrite(Step2, LOW); delayMicroseconds(move_delay2); // Снимаем сигнал с вывода Step2 и ждём время move_delay2;
}
}
```

После загрузки скетча и подключения питания к двигателю, обратите внимание на джойстик. При повороте ручки джойстика вправо/вверх двигателя оси X/Y будет плавно набирать скорость и вращаться по-часовой стрелке, а при плавном повороте ручки влево/вниз двигателя будут плавно набирать скорость и вращаться против часовой стрелки.

Применение:

- Станки резки/гравировки/3D-печати;