

Драйвер сервоприводов Multiservo Shield v2

Используйте драйвер сервомоторов [Multiservo Shield v2](#) для управления до 18 сервомашинками одновременно, на базе которых вы можете создавать гексаподов, роботов манипуляторов и других моторизованных устройств.



Предыдущие версии

Первая ревизия модуля



[Multiservo Shield v1](#). Отличие v1 и v2:

- В версии v2 присутствует защита от переплюсовки по питанию через силовой клеммник.
- В версии v2 распаяна электронная обвязка для считывания показаний силового питания и потребляемого тока.

Примеры работы для Arduino

В качестве мозга для работы с Multiservo Shield v2 рассмотрим платформы из семейства [Arduino](#), например [Uno](#).

[Как начать работу с Arduino?](#)

Подключение и настройка

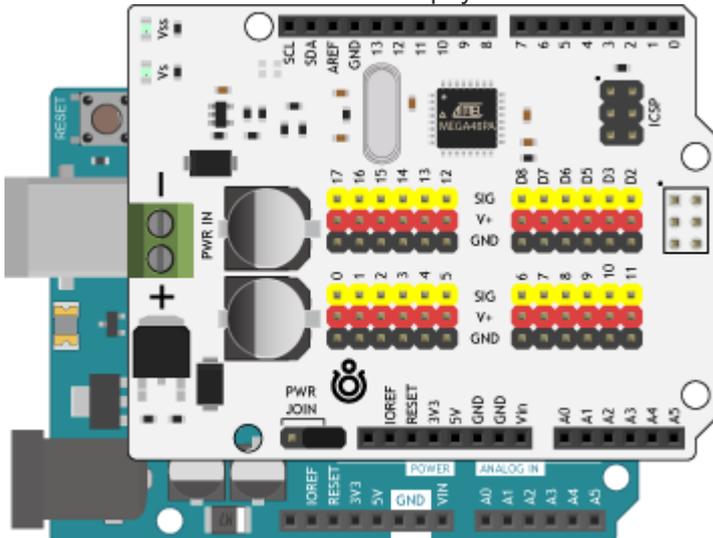
Для старта необходимо [подключить железо](#) и [выполнить программную настройку](#).

Аппаратная часть

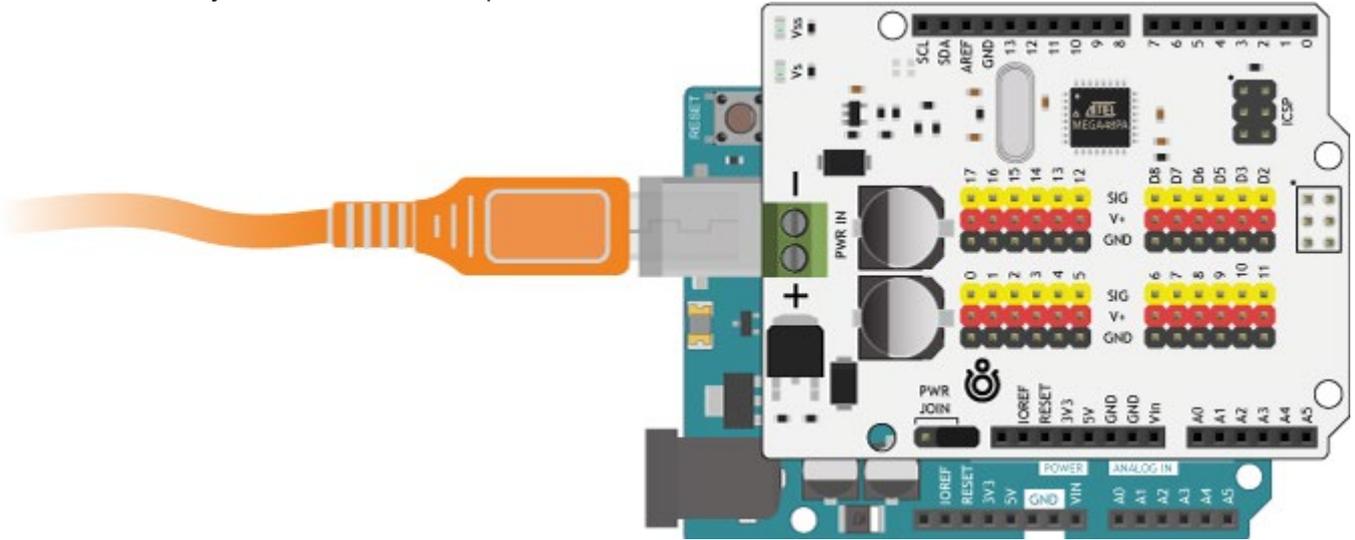
На низком уровне драйвер сервоприводов общается с управляющей электроникой по интерфейсу I²C.

[Подробнее про I²C в Arduino](#)

1. Установите Multiservo Shield сверху на Arduino Uno методом бутерброда.

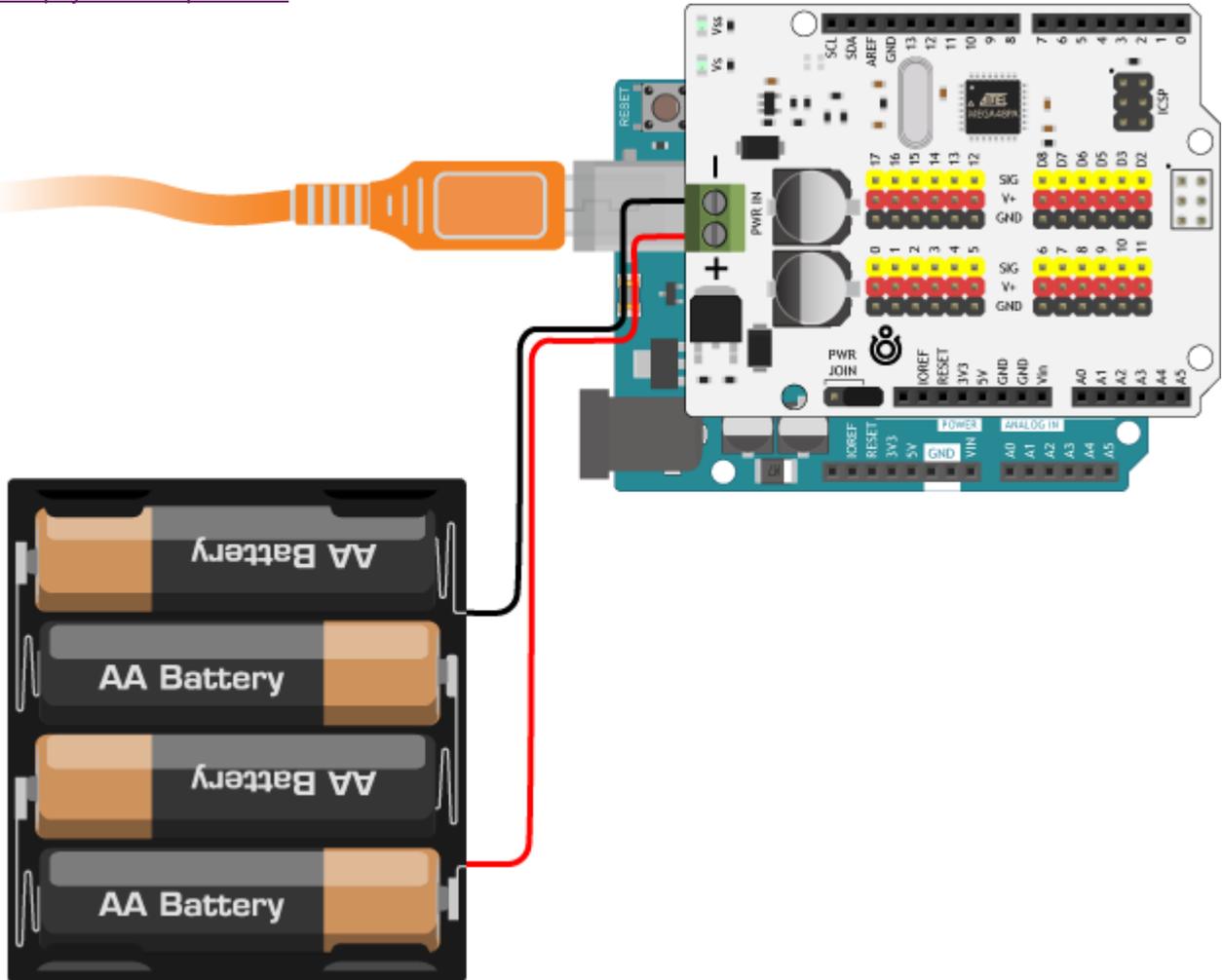


2. Соедините плату Arduino с компьютером по USB.

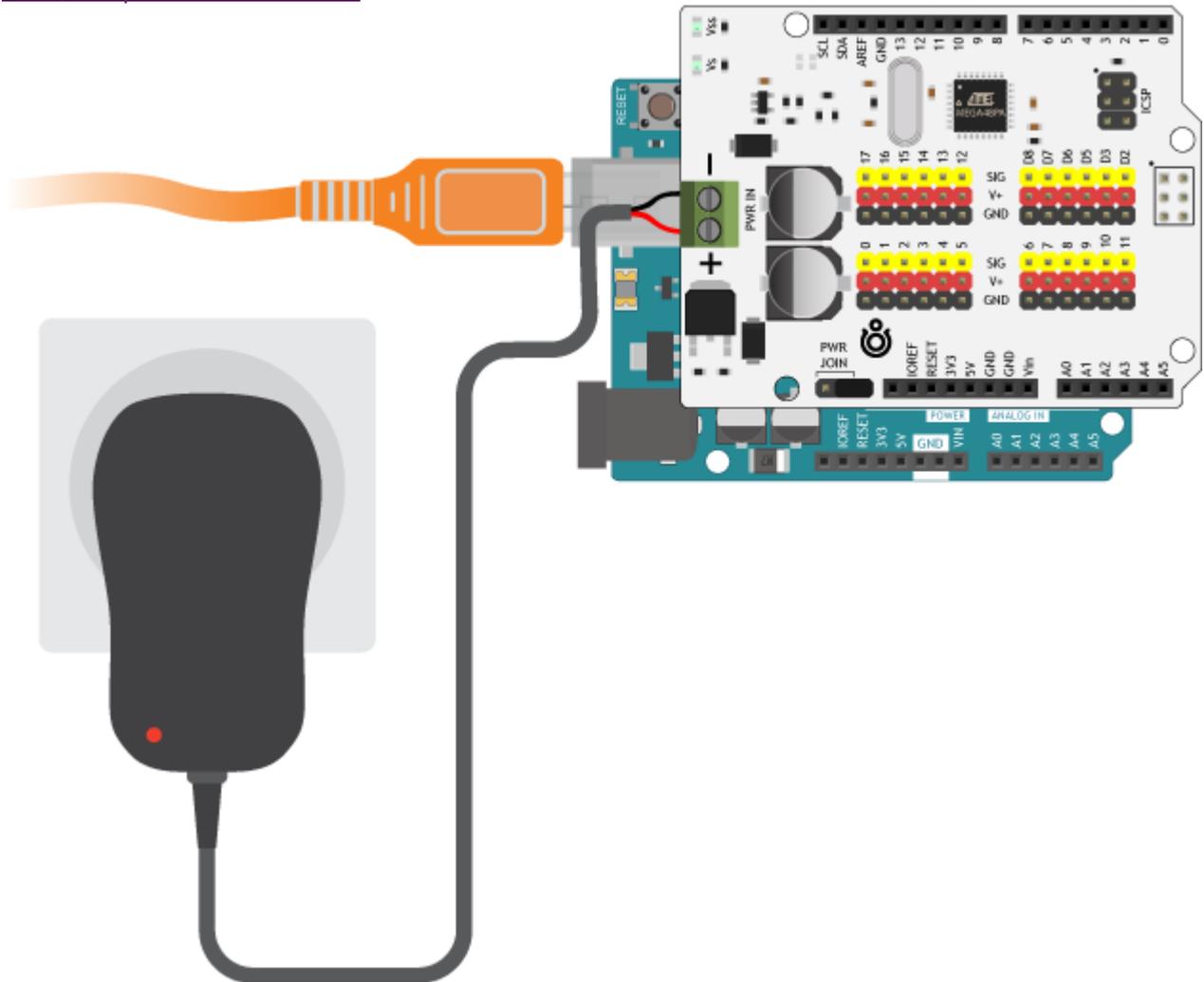


3. Подключите силовое питание к драйверу сервоприводов через силовой клеммник. Диапазон входного напряжения должен соответствовать рабочему напряжению будущих сервоприводов, т.е. сколько приложили на силовой клеммник, столько и поступит на линию питания сервоприводов. В качестве источника питания рекомендуем взять:

- [Сборку из батареек AA:](#)



- [Стационарный блок питания:](#)



Программная часть

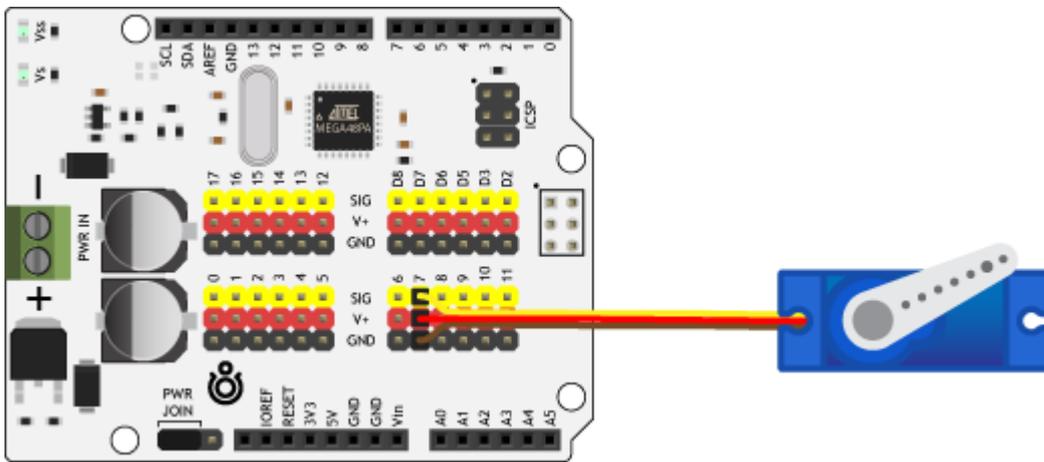
1. [Заведите Arduino.](#)
2. [Установите библиотеку Multiservo.](#)
3. [Протестируйте устройство I²C-сканером.](#) Адрес устройства по умолчанию: 0x47.

На этом установка закончена, теперь смело переходите к экспериментам.

Управление одним сервоприводом

Рассмотрим базовый пример — подключим один сервопривод к 7 пину Multiservo Shield и заставим его плавно вращаться от 0 до 180 градусов.

Схема устройства



Код для Arduino IDE

[multiservo-sweep.ino](#)

```
// Библиотека для работы с Multiservo Shield
// https://github.com/amperka/Multiservo
#include <Multiservo.h>

// Создаём объект для работы с сервомоторами
Multiservo multiservo;

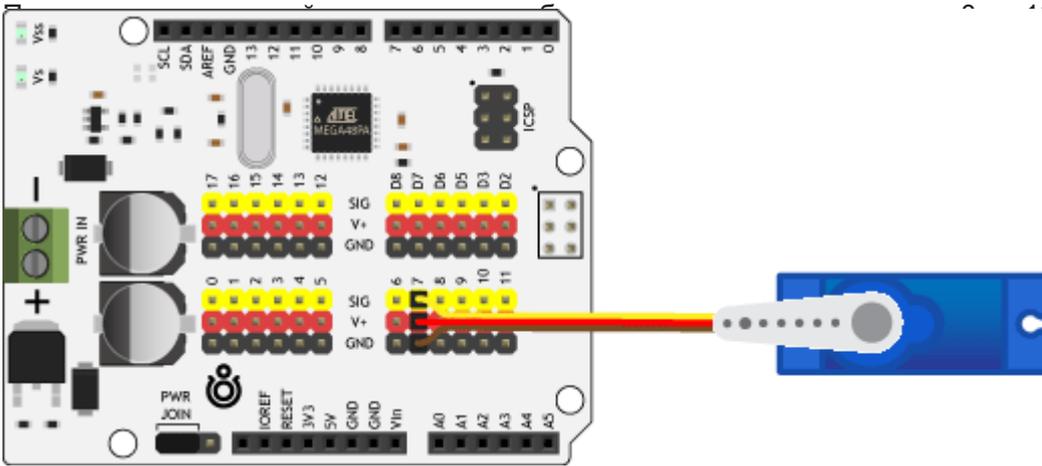
// Задаём имя пина к которому подключён сервопривод
constexpr uint8_t MULTI_SERVO_PIN = 7;

// Переменная для хранения текущей позиции сервомотора
int pos = 0;

void setup() {
  // Подключаем сервомотор
  multiservo.attach(MULTI_SERVO_PIN);
}

void loop() {
  // Перебираем значения счётчика от 0 до 180
  for (pos = 0; pos <= 180; pos++) {
    // Отправляем текущий угол на серво
    multiservo.write(pos);
    // Ждём 15 мс
    delay(15);
  }
  // Перебираем значения счётчика от 180 до 0
  for (pos = 180; pos >= 0; pos--) {
    // Отправляем текущий угол на серво
    multiservo.write(pos);
    // Ждём 15 мс
    delay(15);
  }
}
```

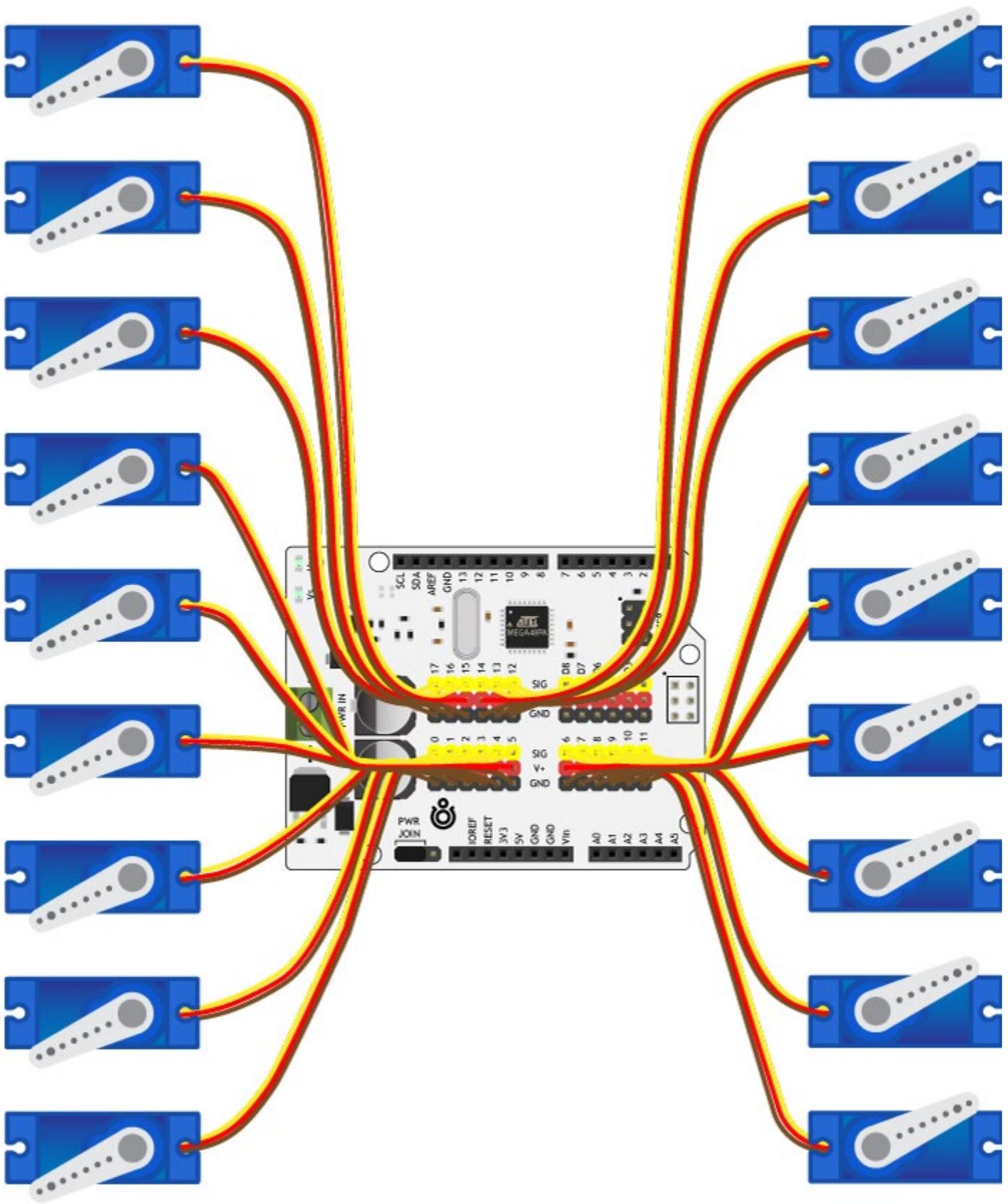
130 градусов и так по кругу.



Управление 18 сервоприводами

Выжмем максимум из драйвера — подключим 18 сервоприводов к пинам 0–17 Multiservo Shield и заставим их плавно по очереди вращаться от 0 до 180 градусов.

Схема устройства



Код для Arduino IDE

[multiservo-multiple-sweep.ino](#)

```
// Библиотека для работы с Multiservo Shield
// https://github.com/amperka/Multiservo
#include <Multiservo.h>

// Задаём количество сервоприводов
```

```

constexpr uint8_t MULTI_SERVO_COUNT = 18;

// Создаём массив объектов для работы с сервомоторами
Multiservo multiservo[MULTI_SERVO_COUNT];

// Переменная для хранения текущей позиции сервомотора
int pos = 0;

void setup() {
  // Перебираем значения моторов от 0 до 17
  for (int count = 0; count < MULTI_SERVO_COUNT; count++) {
    // Подключаем сервомотор
    multiservo[count].attach(count);
  }
}

void loop() {
  // Перебираем значения моторов от 0 до 17
  for (int count = 0; count < MULTI_SERVO_COUNT; count++) {
    // Перебираем значения счётчика от 0 до 180
    for (pos = 0; pos <= 180; pos++) {
      // Отправляем текущий угол на серво
      multiservo[count].write(pos);
      // Ждём 15 мс
      delay(15);
    }
    // Перебираем значения счётчика от 180 до 0
    for (pos = 180; pos >= 0; pos--) {
      // Отправляем текущий угол на серво
      multiservo[count].write(pos);
      // Ждём 15 мс
      delay(15);
    }
  }
}

```

После прошивки устройства вал каждого мотора будет плавно по очереди перемещаться от 0 до 180 градусов и так по кругу.

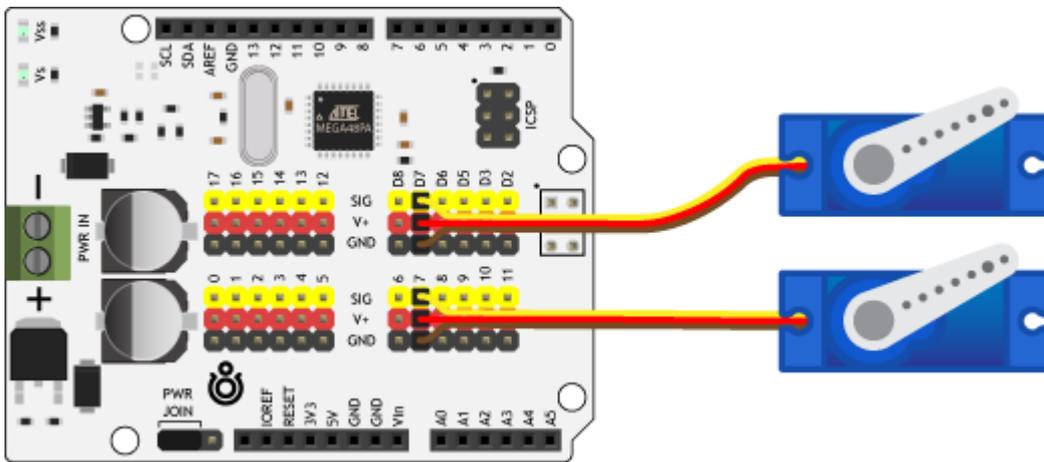
[Скачать демонстрационный пример.](#)

Servo и Multiservo

А теперь рассмотрим симбиоз — подключим один сервопривод к 7 пину Multiservo Shield, а второй к 7 пину платформы Arduino Uno. А затем заставим их плавно вращаться от 0 до 180 градусов.

Если запутались в подключении, смотрите подробности в распиновке платы.

Схема устройства



Код для Arduino IDE

[multiservo-and-servo-sweep.ino](#)

```
// Библиотека для работы с Servo
#include <Servo.h>

// Библиотека для работы с Multiservo Shield
// https://github.com/amperka/Multiservo
#include <Multiservo.h>

// Создаём объект для работы с сервомоторами на Arduino
Servo servo;

// Создаём объект для работы с сервомоторами на Multiservo Shield
Multiservo multiservo;

// Задаём имена пинов к которым подключены сервоприводы
constexpr uint8_t SERVO_PIN = 7;
constexpr uint8_t MULTI_SERVO_PIN = 7;

// Переменная для хранения текущей позиции сервомотора
int pos = 0;

void setup() {
  // Подключаем сервомотор на Arduino
  servo.attach(SERVO_PIN);
  // Подключаем сервомотор на Multiservo Shield
  multiservo.attach(MULTI_SERVO_PIN);
}

void loop() {
  // Перебираем значения счётчика от 0 до 180
  for (pos = 0; pos <= 180; pos++) {
    // Отправляем текущий угол на серво
    servo.write(pos);
    // Ждём 15 мс
    delay(15);
  }
  // Перебираем значения счётчика от 180 до 0
  for (pos = 180; pos >= 0; pos--) {
    // Отправляем текущий угол на серво
    servo.write(pos);
  }
}
```

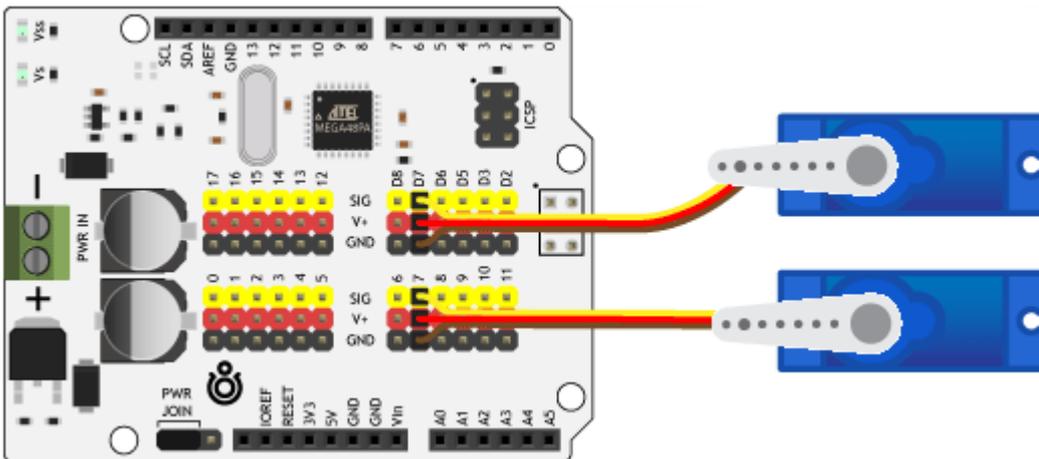
```

// Ждём 15 мс
delay(15);
}

// Перебираем значения счётчика от 0 до 180
for (pos = 0; pos <= 180; pos++) {
  // Отправляем текущий угол на серво
  multiservo.write(pos);
  // Ждём 15 мс
  delay(15);
}
// Перебираем значения счётчика от 180 до 0
for (pos = 180; pos >= 0; pos--) {
  // Отправляем текущий угол на серво
  multiservo.write(pos);
  // Ждём 15 мс
  delay(15);
}
}

```

После прошивки устройства сначала вал одного мотора будет плавно перемещаться от 0 до 180 градусов, а затем вал другого мотора от 0 до 180 градусов и так по кругу.



Примеры работы для Espruino

В качестве мозга для работы с Multiservo Shield v2 рассмотрим платформы из семейства [Espruino](#), например [Iskra JS](#).

[Как начать работу с Iskra JS?](#)

Подключение и настройка

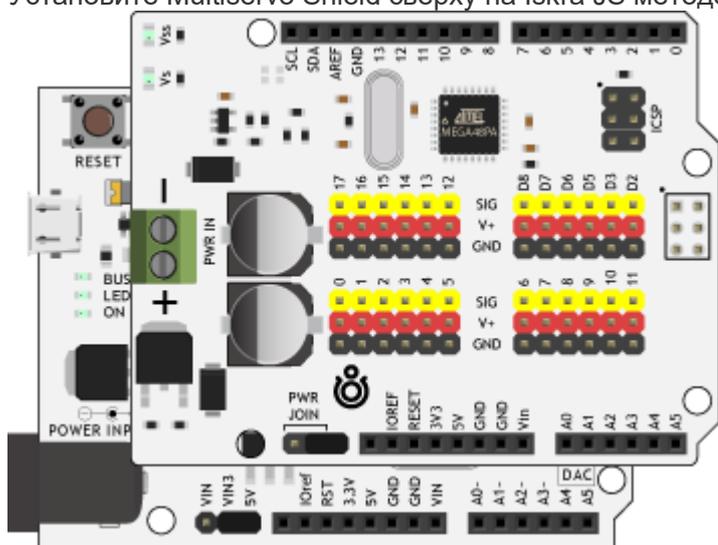
Для старта необходимо [подключить железо](#) и [выполнить программную настройку](#).

Аппаратная часть

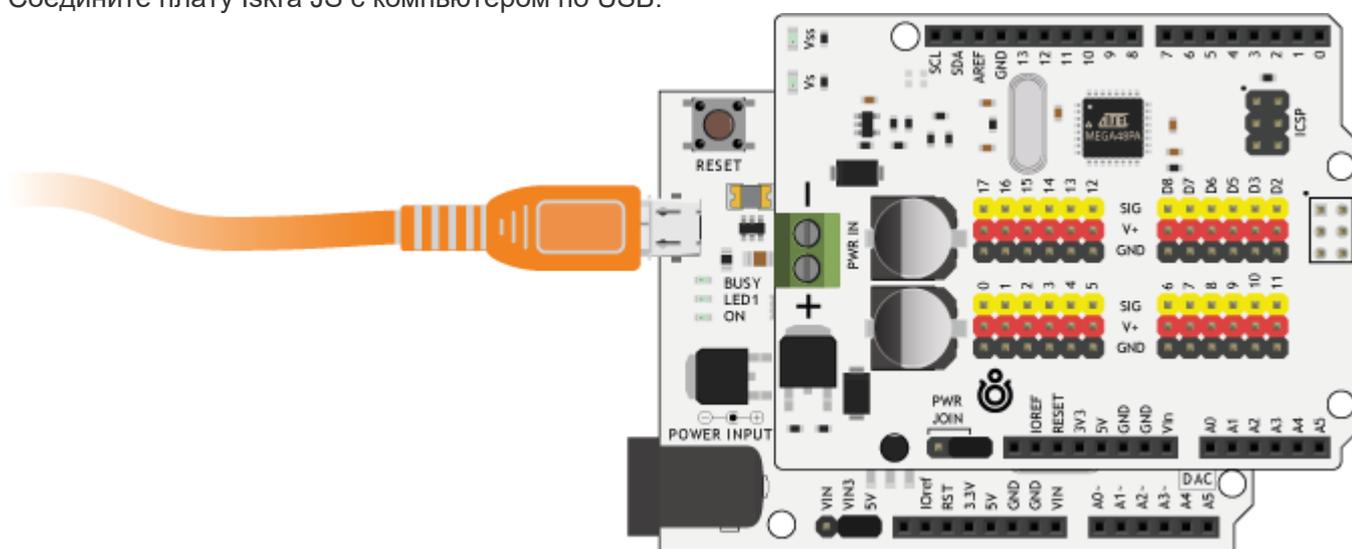
На низком уровне драйвер сервоприводов общается с управляющей электроникой по интерфейсу I²C.

[Подробнее про I²C в Espruino](#)

1. Установите Multiservo Shield сверху на Iskra JS методом бутерброда.

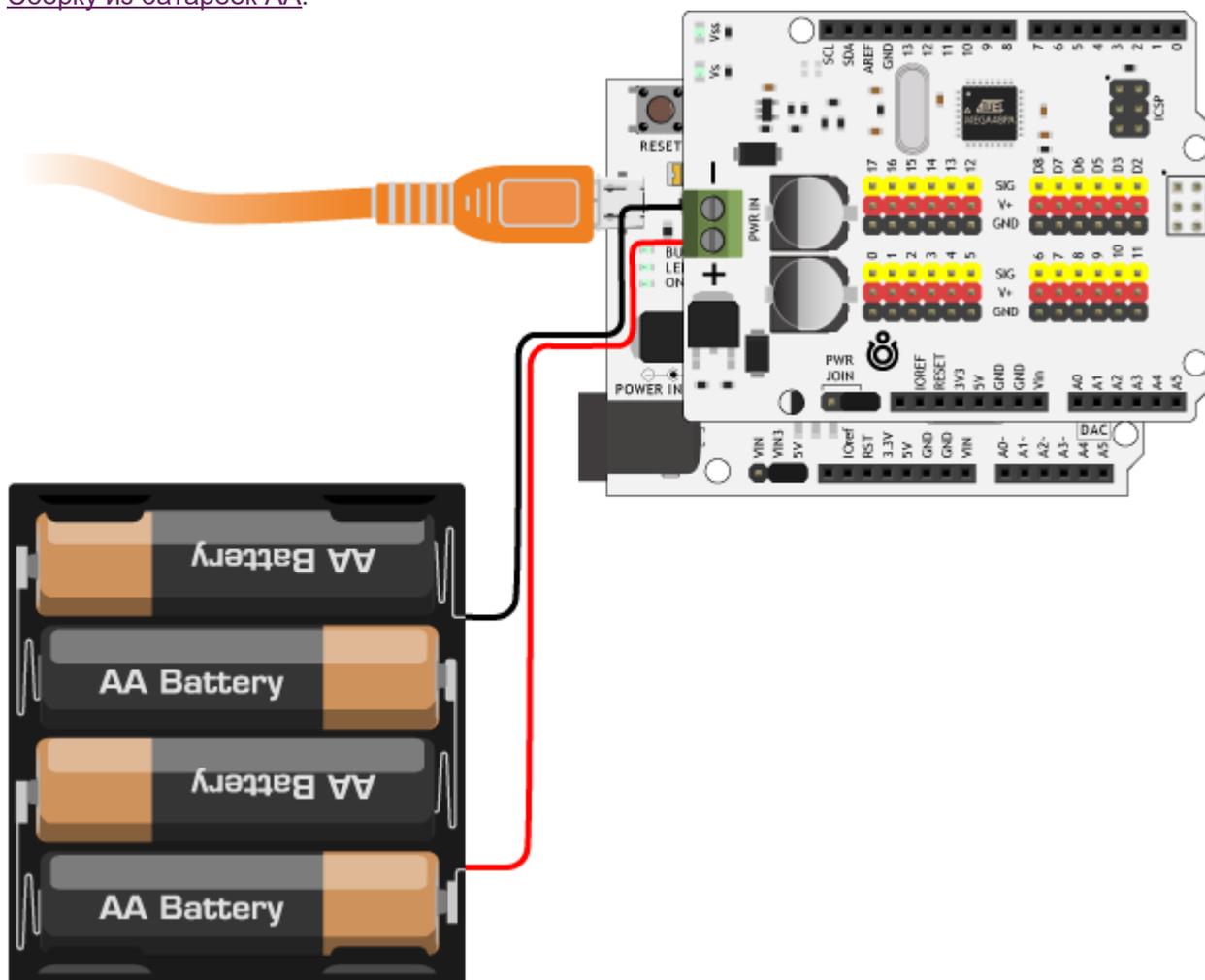


2. Соедините плату Iskra JS с компьютером по USB.

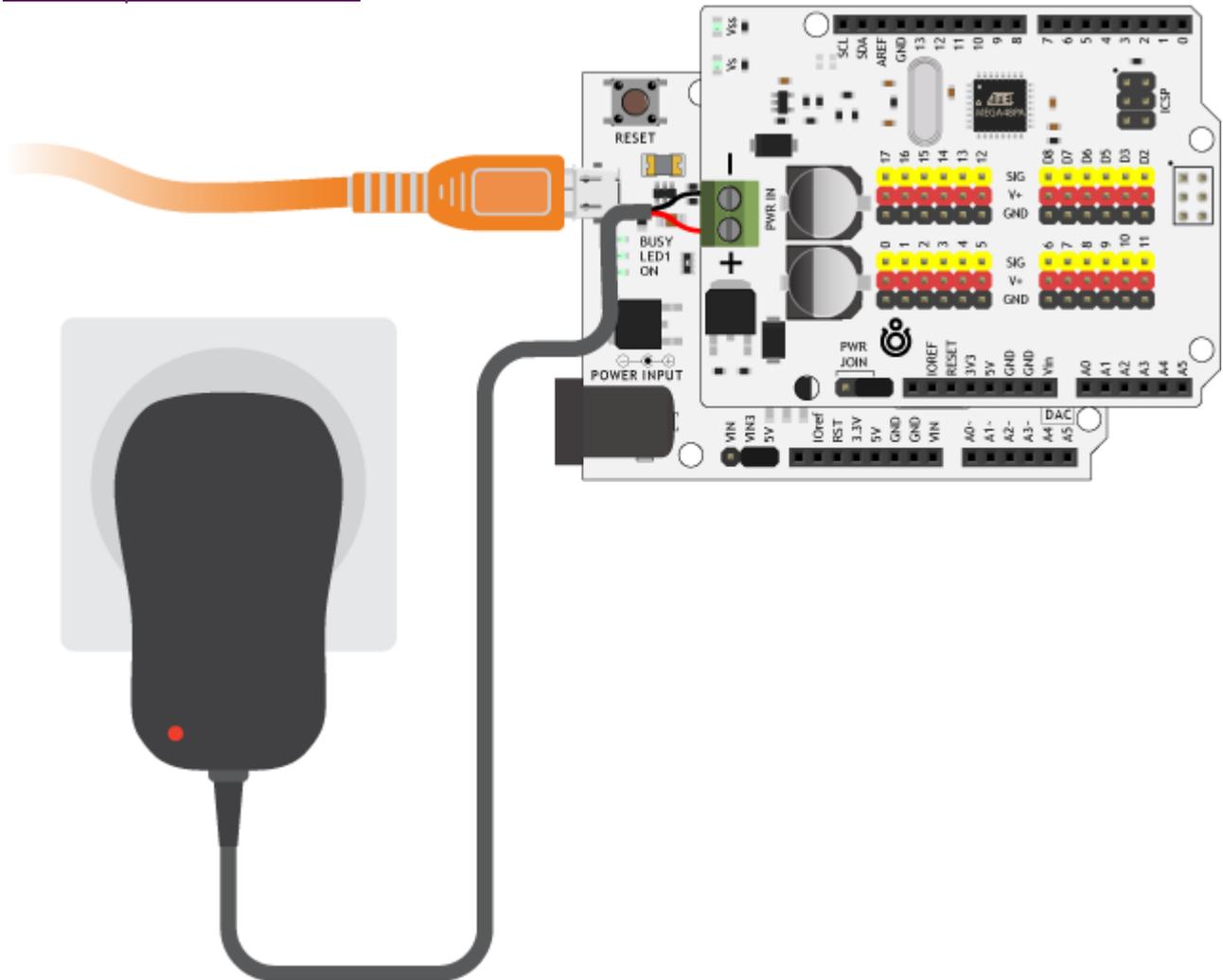


3. Подключите силовое питание к драйверу сервоприводов через силовой клеммник. Диапазон входного напряжения должен соответствовать рабочему напряжению будущих сервоприводов, т.е. сколько приложили на силовой клеммник, столько и поступит на линию питания сервоприводов. В качестве источника питания рекомендуем взять:

- [Сборку из батареек AA:](#)



- [Стационарный блок питания:](#)



Программная часть

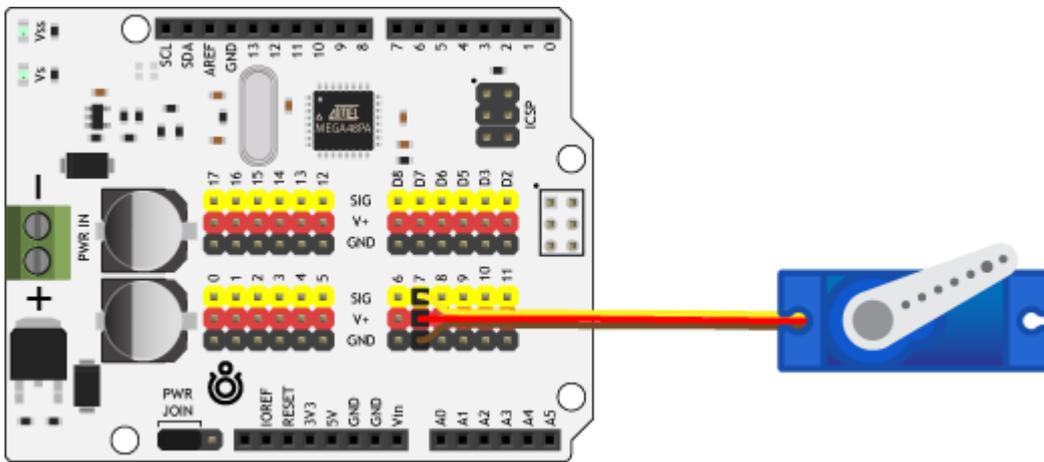
1. [Заведите Iskra JS.](#)
2. [Протестируйте библиотеку Multiservo.](#)

На этом установка закончена, теперь смело переходите к экспериментам.

Управление одним сервоприводом

Рассмотрим базовый пример — подключим один сервопривод к 7 пину Multiservo Shield и заставим его плавно вращаться от 0 до 180 градусов.

Схема устройства



Код для Iska JS

[multiservo-sweep.js](#)

```
// Настраиваем шину I2C
PrimaryI2C.setup({sda: SDA, scl: SCL, bitrate: 400000});

// Создаем новый объект Multiservo
var multiservo = require('@amperka/multiservo').connect(PrimaryI2C);

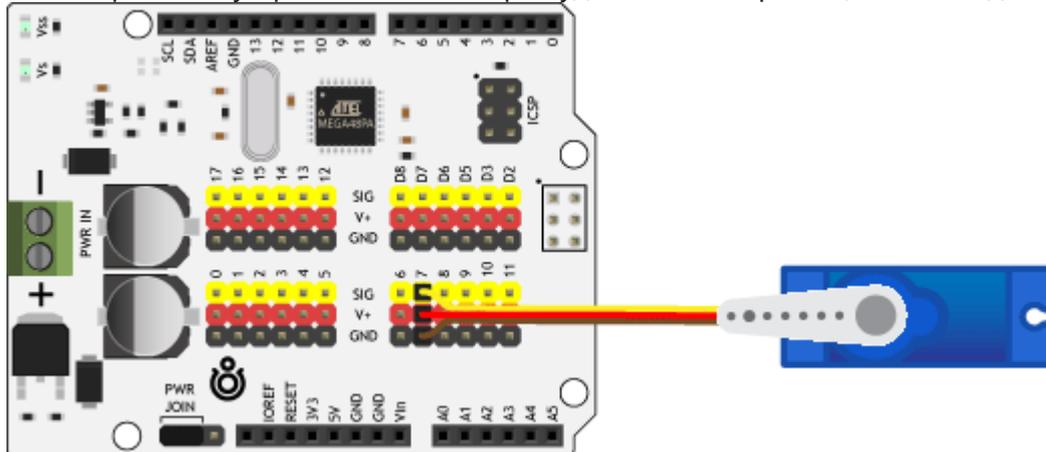
// Создаем новый объект сервопривода
// подключены к контакту 7
var servo = multiservo.connect(7);

// Создаём объект анимация
// для плавного изменения параметров вращения мотора
var animServo = require('@amperka/animation').create({
  // Начальное значение в градусах
  from: 0,
  // Конечное значение в градусах
  to: 180,
  // Продолжительность полного перехода
  // за 5 секунд мотор пройдёт диапазон значений от 0 до 180
  duration: 5,
  // Шаг обновления: каждые 20 мс
  updateInterval: 0.02
}).queue({
  // Начальное значение в градусах
  from: 180,
  // Конечное значение в градусах
  to: 0,
  // продолжительность полного перехода
  // за 5 секунд мотор пройдёт диапазон значений от 180 до 0
  duration: 5
});

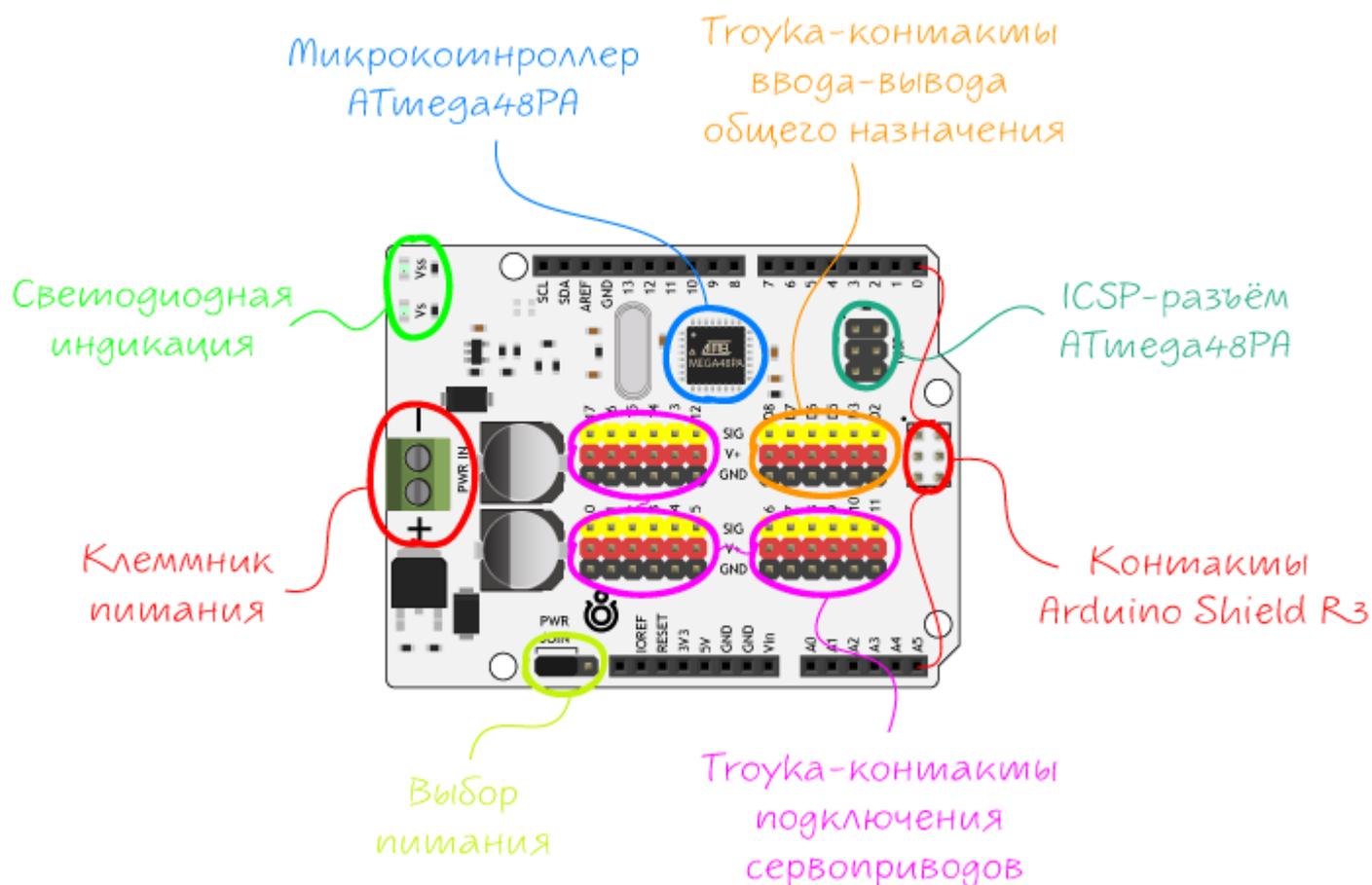
// Обработчик анимации
animServo.on('update', function(val) {
  servo.write(val);
});
```

```
// Запускаем анимацию
setInterval(() => {
  animServo.play();
}, 1000);
```

После прошивки устройства вал мотора будет плавно перемещаться от 0 до 180 градусов и так по кругу.



Элементы платы



Микроконтроллер ATmega48PA

Плата Multiservo Shield выполнена на микроконтроллере [ATmega48PA](#) с прошивкой управления сервоприводами от Амперки. Чип принимает команды по I²C интерфейсу от внешней управляющей платы, например [Arduino Uno](#) или [Iskra JS](#), и рулит до 18 сервоприводами в одно время.

Силовой клеммник питания

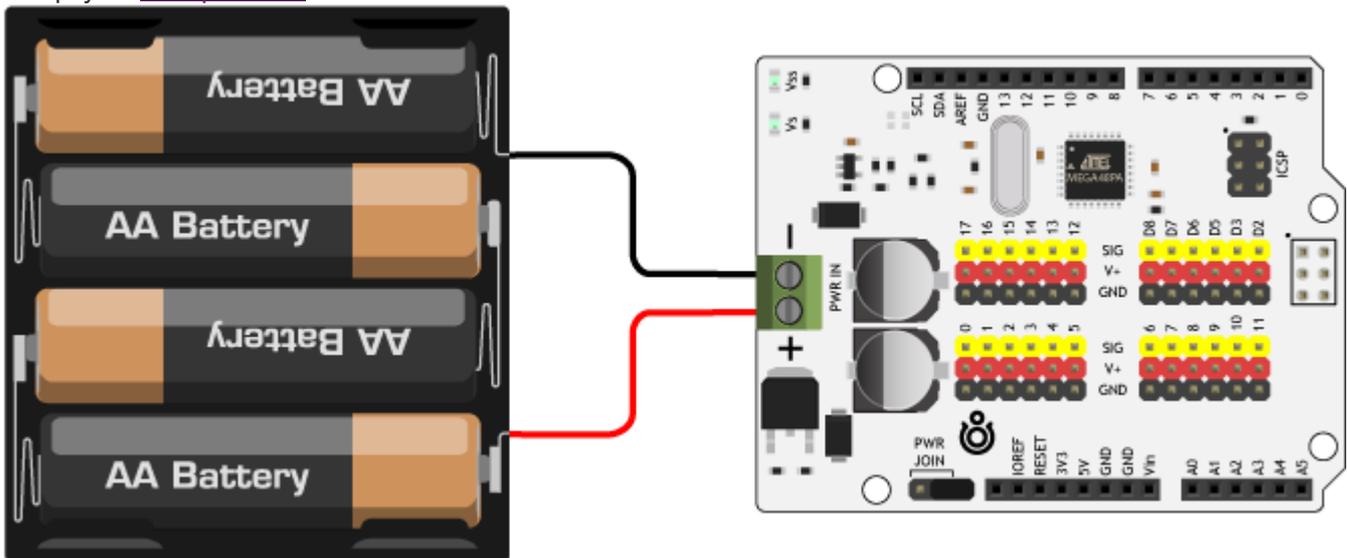
Для питания сервомоторов используйте клеммник под винт PWR IN.

Силовой клеммник	Подключение
PWR +	Силовое питание
PWR -	Земля

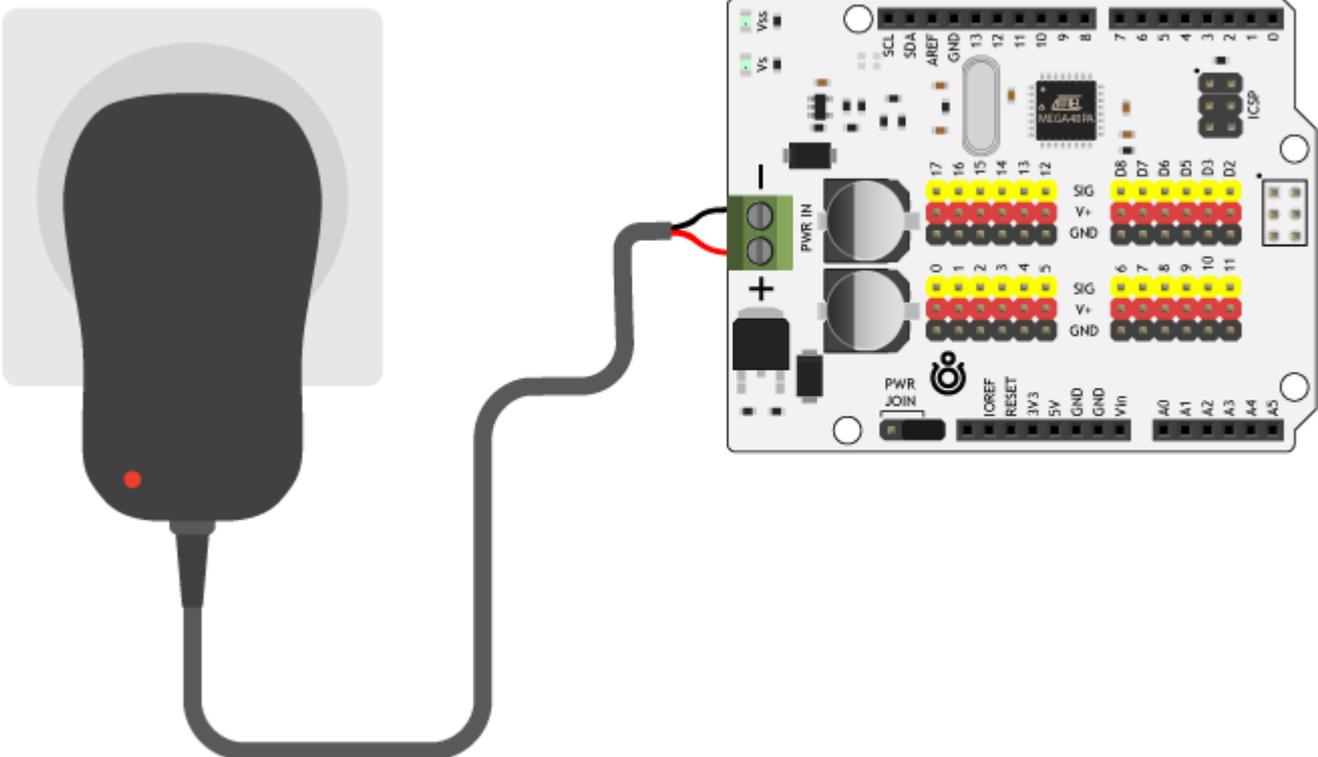
Диапазон входного напряжения должен соответствовать рабочему напряжению подключаемых сервоприводов, т.е. сколько приложили на силовой клеммник, столько и поступит на линию питания моторов. Номинальное напряжения большинства хобби сервоприводов не выходит за рамки диапазона от 5 до 12 вольт.

В качестве источника питания рекомендуем взять:

- Сборку из [батареек AA](#):



- [стационарный блок питания](#):



- И другие источники напряжения.

Контуры питания

На плате расширения MultiServo Shield присутствует два контура питания.

- **Силовой контур Vs.** Напряжение питания сервомоторов, которое поступает от [силового клеммника](#). Диапазон входного напряжения должен соответствовать номинальному питанию моторов, а суммарный максимальный ток потребления не должен превышать 10 А.
- **Цифровой контур Vss.** Напряжение питания микроконтроллера и другой вспомогательной логики. Цифровое питание поступает через пин 5V от внешней управляющей платы, например от USB. Входное напряжение соответственно равно 5 вольт, а максимальный ток потребления не более 50 мА.

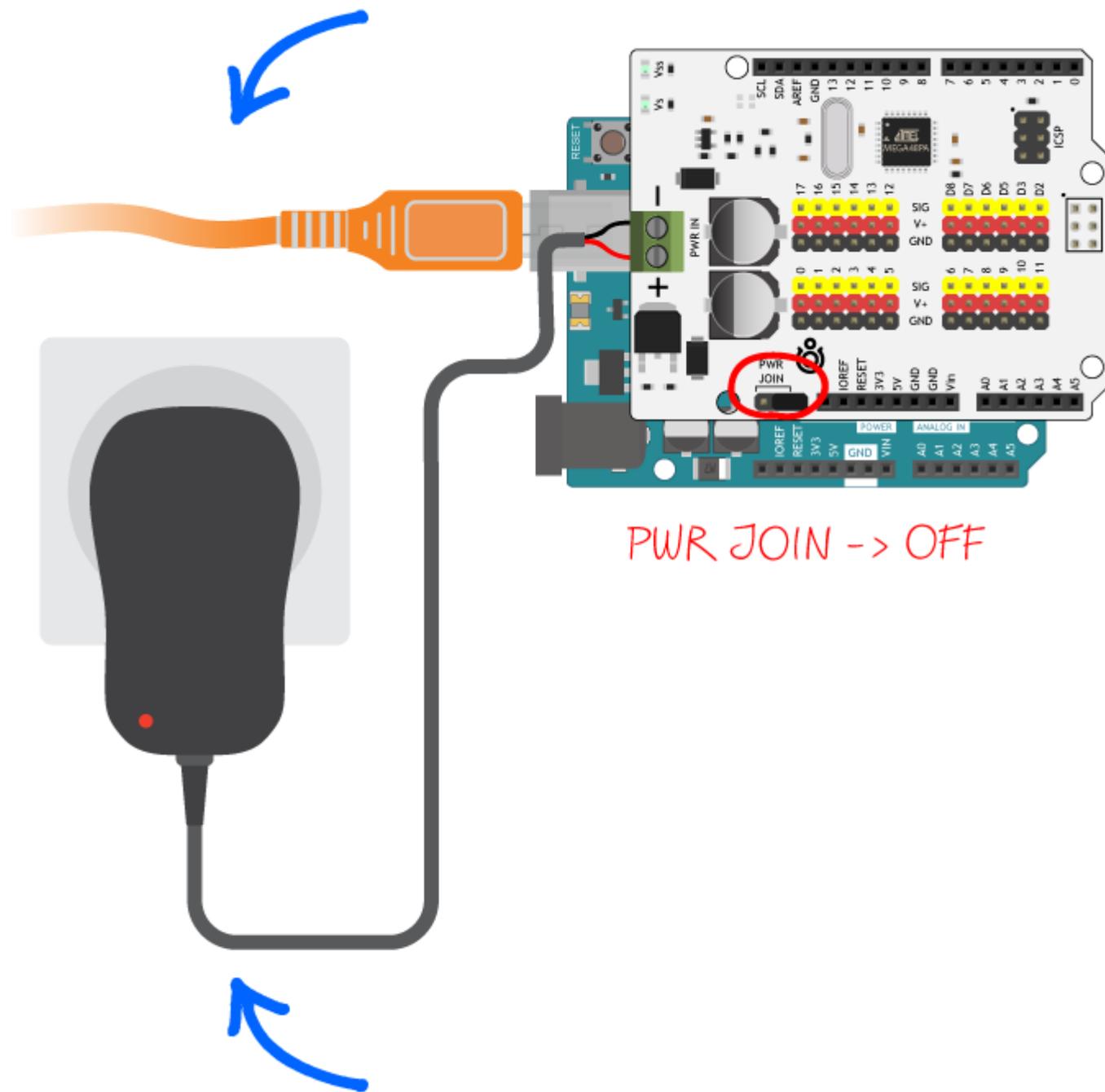
Если отсутствует хотя бы один из контуров питания Vs или Vss — Multiservo Shield работать не будет.

Для информации о текущем состоянии каждого контура [используйте светодиодную индикацию](#).

Джаммер объединения питания

На плате расширения MultiServo Shield присутствует два контура напряжения, т.е. для работы схемы

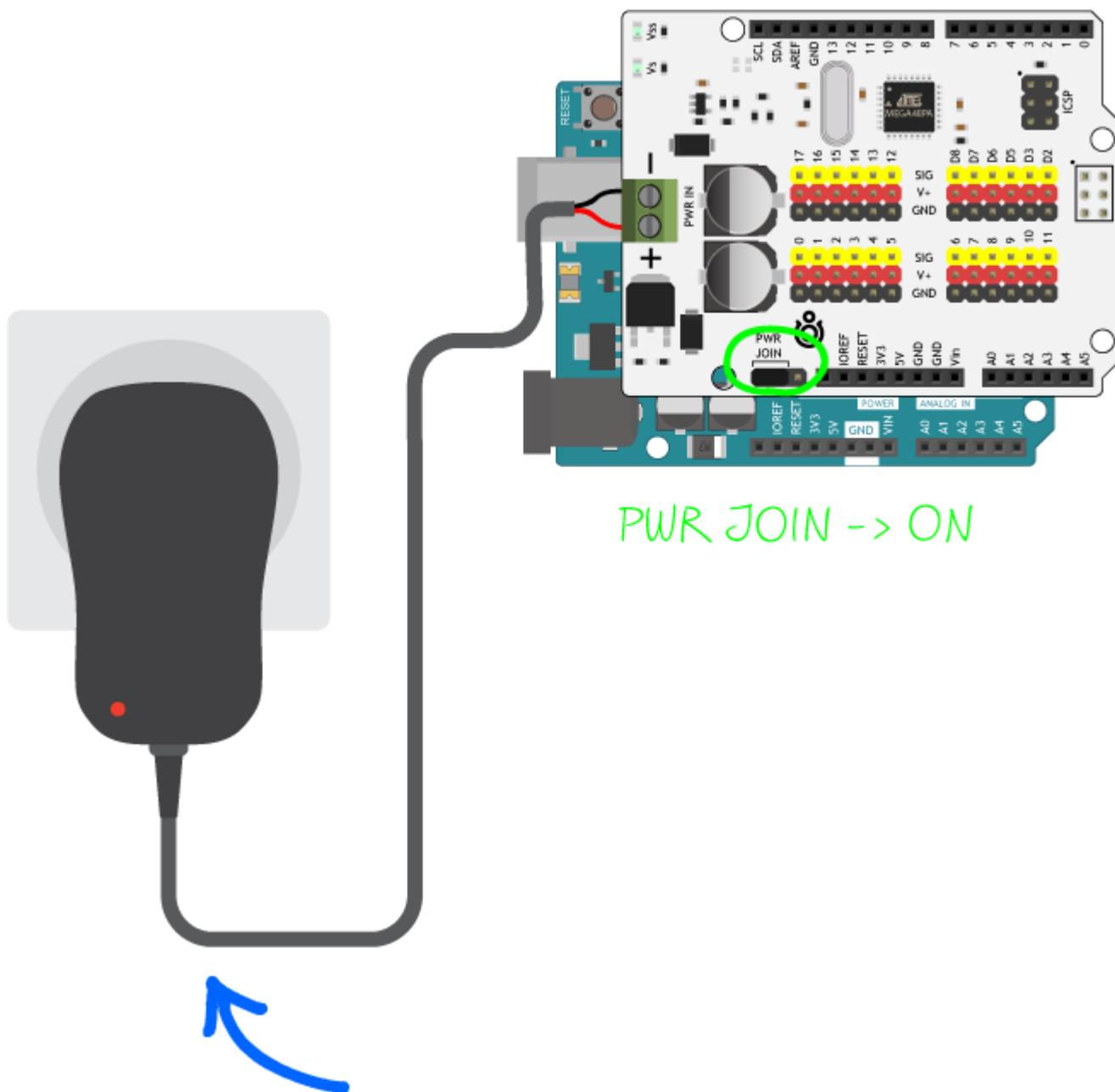
Vss: источник цифрового питания



Vs: источник силового питания

При установке джампера в положение PWR JOIN, происходит объединение положительного контакта + силового клеммника PWR IN с пином Vin управляющей платформы. Режим объединённого

питания позволяет запитывать всё устройство от одного источника напряжения.



Vs: источник силового питания

Выбор питания

При объединённом режиме **PWR JOIN**, напряжение на устройство может быть подано двумя способами:

- На драйвер сервомоторов через клеммник **PWR IN**.
- На управляющую плату через внешний DC-разъём.

Правила

При объединённом режиме **PWR JOIN**, важно знать:

- При работе двигателей по цепи питания может проходить большой ток, на который цепь **Vin** управляющей платформы может быть не рассчитана. Поэтому выбор питания через силовой клеммник **PWR IN** предпочтительнее.

- 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.

Светодиодная индикация

Имя светодиода	Назначение
V _S / PON	Индикатор подачи силового питания. Горит — напряжение есть, не горит — напряжение нет.
V _{SS} / цON	Индикатор подачи цифрового питания. Горит — напряжение есть, не горит — напряжение нет.

Тройка-контакты подключения сервоприводов

Сервоприводы подключаются к плата Multiservo Shield через контактные штыри S-V-G, где:

- S — сигнал с номером от 0 до 18.
- V — питание сервомоторов. Берется от силового клеммника.
- G — земля.

На линии V будет присутствовать не логическое питание платформы 3,3 / 5 В, а напряжение V_s приложенное к [силовому клеммнику](#).

Тройка-контакты ввода-вывода общего назначения

На плате доступны шесть Тройка-контактов ввода-вывода внешнего контролера, которые можно задействовать в дополнение к основным. Контакты пронумерована S-V-G, где:

- S — сигнал с номером: D2, D3, D5, D6, D7 и D8.
- V — питание от силового клеммника.
- G — земля.

На линии V будет присутствовать не логическое питание платформы 3,3 / 5 В, а напряжение V_s приложенное к [силовому клеммнику](#).

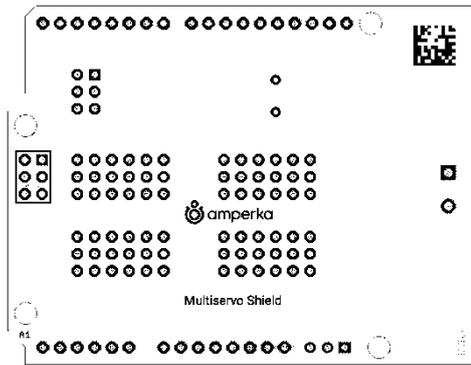
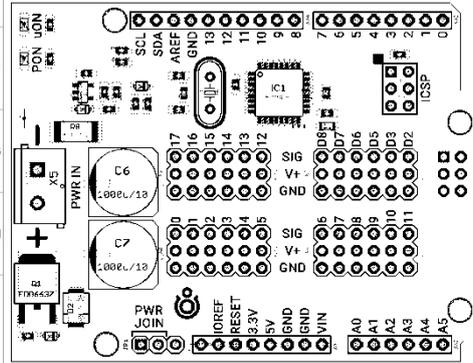
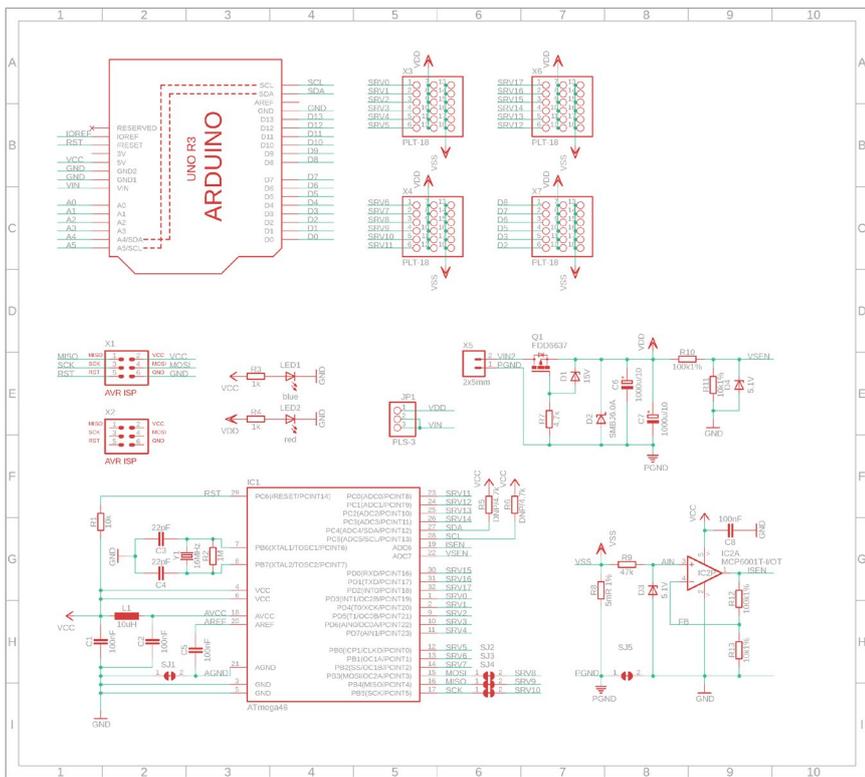
ICSP-разъём ATmega48PA

На плате расположен ICSP-разъём, который предназначен для загрузки прошивки в микроконтроллер ATmega48PA через внешний программатор. В нашем случае — это мост, который получает команды по I²C и рулит 18 сервоприводами.

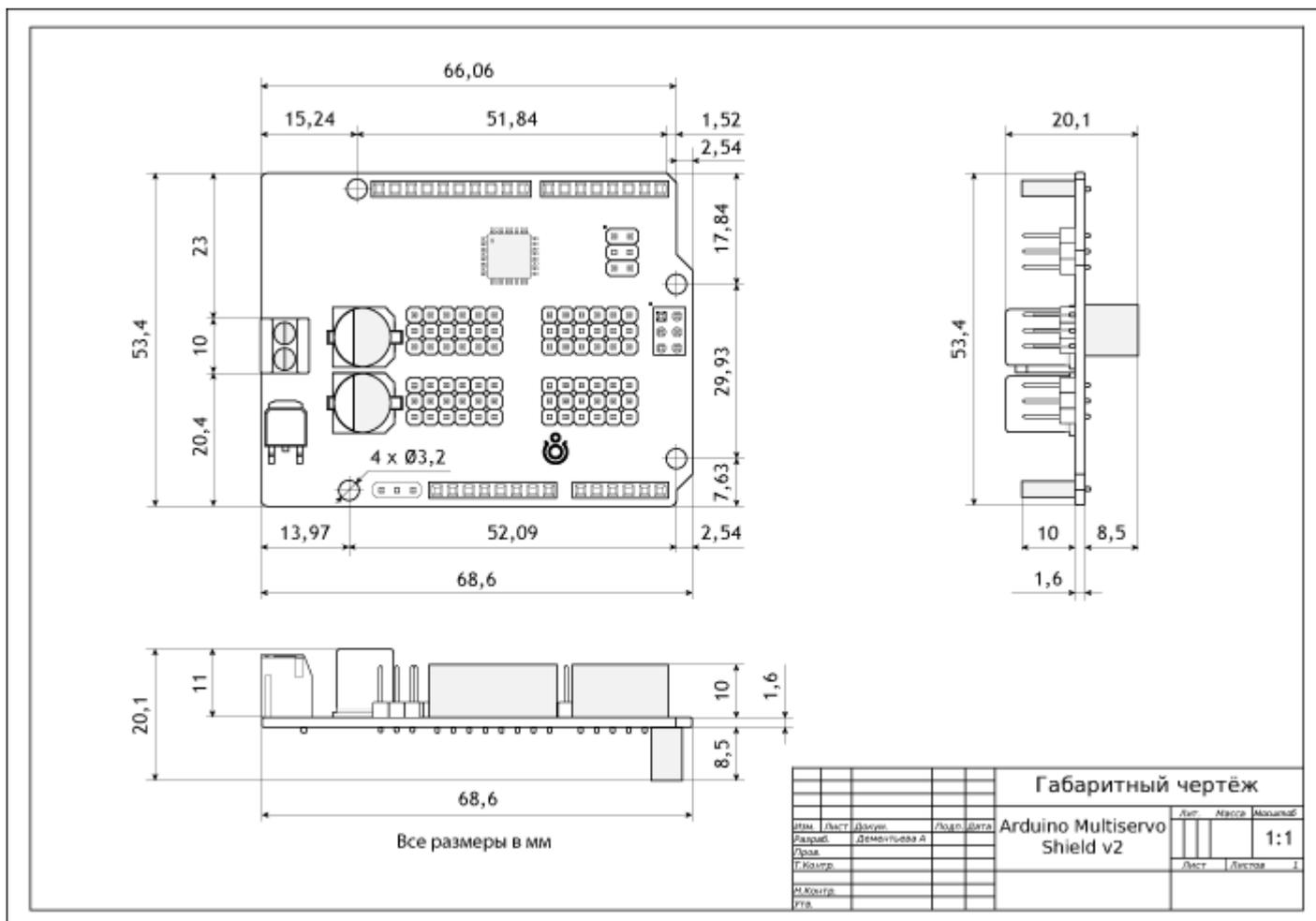
Контакты Arduino Shiled R3

Плата Multiservo Shield выполнена в форм-факторе Arduino Shield R3. а это значит расширение просто одевается сверху на управляющую платформу форм-фактора Arduino R3 методом бутерброда без дополнительных проводов и пайки. В итоге вам остаются доступны все физические контакты вашего контроллера для дальнейшего использования.

Принципиальная и монтажная схемы



Габаритный чертёж



Характеристики

- Модель: Драйвер сервоприводов Multiservo Shield v2 AMP-B201
- Драйвер: микроконтроллер ATmega48PA
- Аппаратный интерфейс: контактные штыри
- Программный интерфейс: I²C
- I²C-адрес: 0x47
- Контактных подключения сервоприводов: 18
- Контактных ввода-вывода общего назначения от внешнего контроллера: 6
- Напряжение питания силовой части: 5–9 В
- Потребляемый ток силовой части: до 16 А
- Напряжение питания цифровой части: 5 В
- Потребляемый ток цифровой части: до 50 мА
- Напряжение логических уровней: 3,3–5 В
- Размеры модуля: 68,6×53,4×20,1 мм

Ресурсы

- [Multiservo Shield v2](#) в магазине.
- [Векторное изображение Multiservo Shield v2](#)
- [Библиотека для Arduino](#)
- [Библиотека для Espruino](#)
- [Прошивка для Multiservo v2 в контроллер ATmega48PA](#)