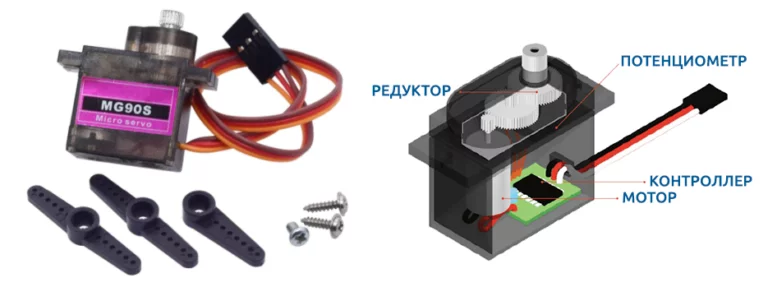
Сервопривод

**Сервопривод** – простейший “модельный” актуатор, использовался в радиоуправляемых моделях ещё до появления Ардуино. Сервопривод поворачивает свой выходной вал в диапазоне ~180 градусов и обладает вполне приличным моментом для перемещения частей механизмов. В маленьком корпусе располагаются:

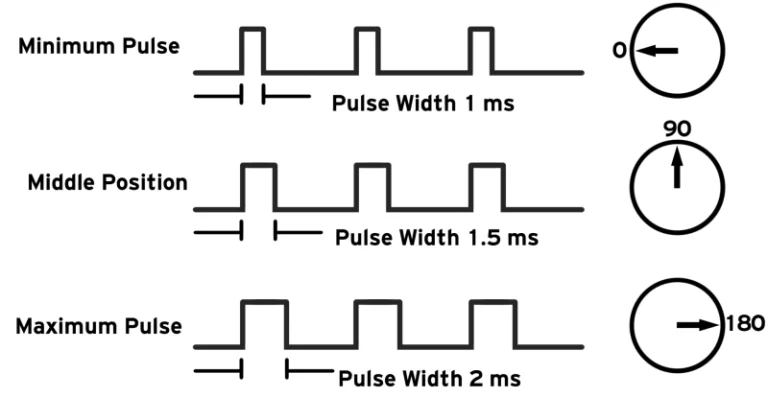
* Коллекторный моторчик;
* Редуктор (пластиковый или металлический);
* Потенциометр обратной связи;
* Контроллер (драйвер мотора, обратная связь, управление по интерфейсу PWM).

Сервопривод комплектуется набором “качалок” и винтами для крепления



## Характеристики 9-граммового сервопривода:

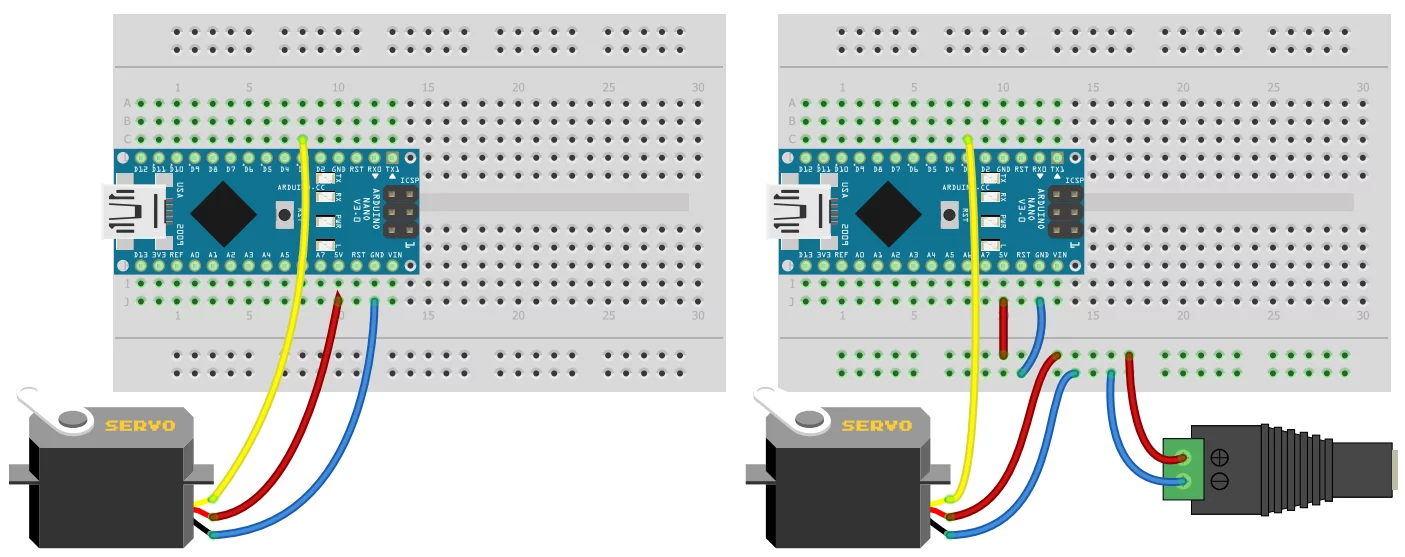
* Напряжение питания: 3-7.2V
* Крутящий момент:
  + 1.2кг/см при 4.8V
  + 1.6кг/см при 6.0V
* Рабочий угол: 160-180 градусов
* Скорость (без нагрузки): 180 градусов за 0.3с
* Интерфейс: PWM

Сервопривод управляется ШИМ сигналом, точнее длиной импульса: минимальная (0 градусов) и максимальная (~180 градусов) длина импульса колеблется в зависимости от модели и производителя сервопривода. 

## Подключение

Провода:

* Коричневый: GND
* Красный: VCC
* Жёлтый: цифровой пин



**Примечание: сервопривод потребляет довольно приличный ток (стартовый до 1А), поэтому рекомендуется питать его от внешнего источника. При питании от компьютера напряжение может просесть и МК перезагрузится, при высокой нагрузке, может выгореть защитный диод на плате Arduino!**

**Библиотеки**

Для управления сервоприводом можно использовать стандартную библиотеку **Servo**. Стандартная библиотека управляет сервоприводом очень резко, на максимальной скорости, поэтому для реальных применений лучше использовать библиотеку плавного движения серво – [ServoSmooth](https://github.com/GyverLibs/ServoSmooth).

**Servo.h**

Библиотека стандартная и идёт в комплекте с Arduino IDE. Подключаем и создаём объект сервопривода. При создании никакие аргументы не передаются:

#include <Servo.h>

Servo myservo;

**Библиотека имеет следующие методы:**

uint8\_t attach(int pin); // "подключить" с указанием пина

uint8\_t attach(int pin, int min, int max); // "подключить" с указанием пина и мин. макс. сигнала

void detach(); // отключить

void write(int value); // повернуть на угол в градусах

void writeMicroseconds(int value); // повернуть на длину импульса

**При “подключении” серво через**

**attach(pin)**

**диапазон длины импульса устанавливается стандартный: 544-2400 мкс (задан в библиотеке). Чтобы серво работал на весь диапазон – нужно попробовать покрутить его через**

**writeMicroseconds()**

**в крайних значениях диапазона и найти минимум и максимум, при которых серво “упирается”.**