

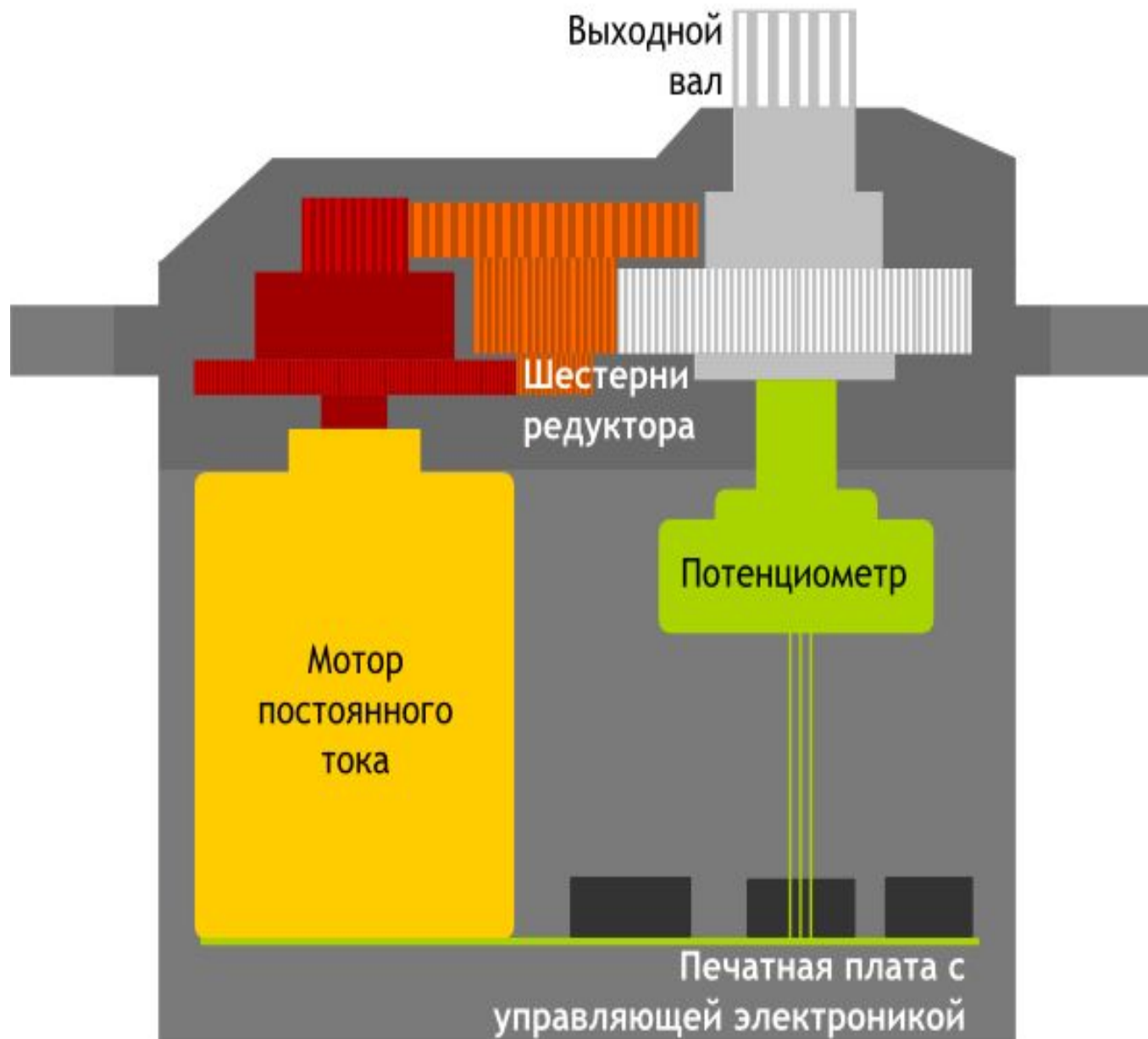
# Сервопривод

# Сервопривод



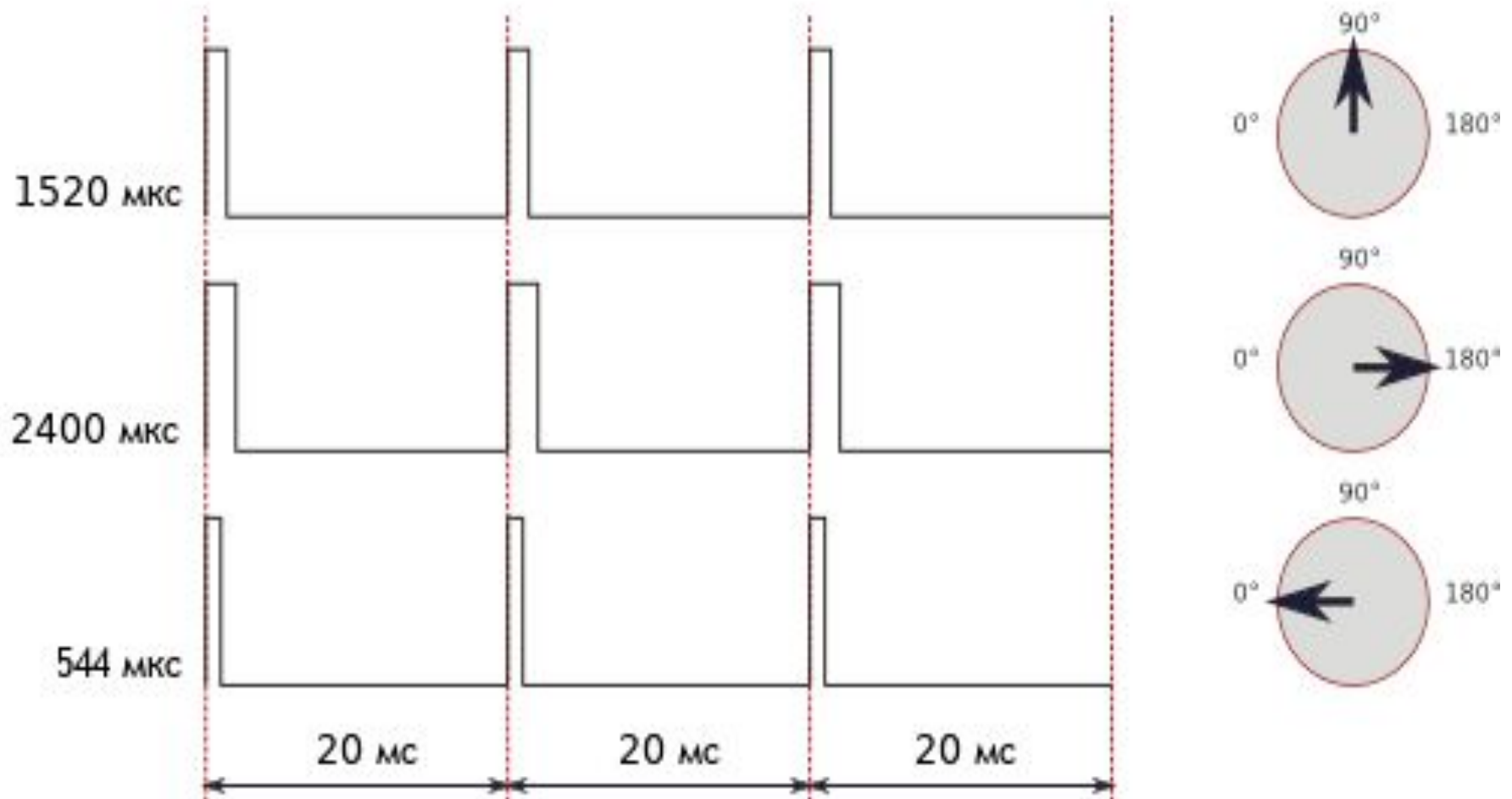
- 1) Сервопривод получает на вход значение управляющего параметра. Например, угол поворота
- 2) Блок управления сравнивает это значение со значением на своём датчике
- 3) На основе результата сравнения привод производит некоторое действие, например: поворот, ускорение или замедление так, чтобы

# Устройство сервопривод



# Управление сервоприводом. Интерфейс управляющих сигналов

Чтобы указать сервоприводу желаемое положение, по предназначенному для этого проводу необходимо посылать управляющий сигнал. Управляющий сигнал — импульс с длительностью 1 мс.



# Сервопривод FS5106B

## Характеристики:

Диапазон вращения:  $180^\circ$

Напряжение питания: 4,8–6 В

Крутящий момент: 5 кг·см при 4,8 В; 6 кг·см при 6 В

Скорость вращения:  $60^\circ$  за 0,18 сек при 4,8 В;  $60^\circ$  за 0,16 сек при 6 В

Внутренний интерфейс:  
аналоговый

Материал шестерней: нейлон

Материал корпуса: пластик

Габариты: 40,8 × 20,1 × 38 мм



# Multiservo Shield

## Характеристики:

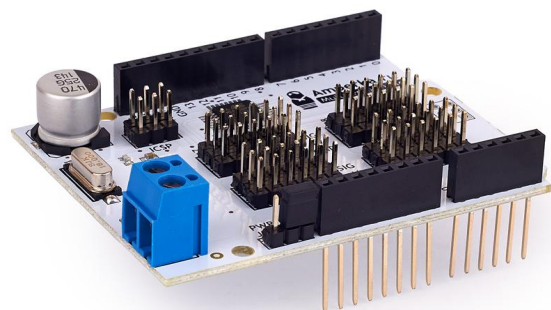
Максимальный постоянный ток  
на сервы: 10 А

Интерфейс управления: I<sup>2</sup>C

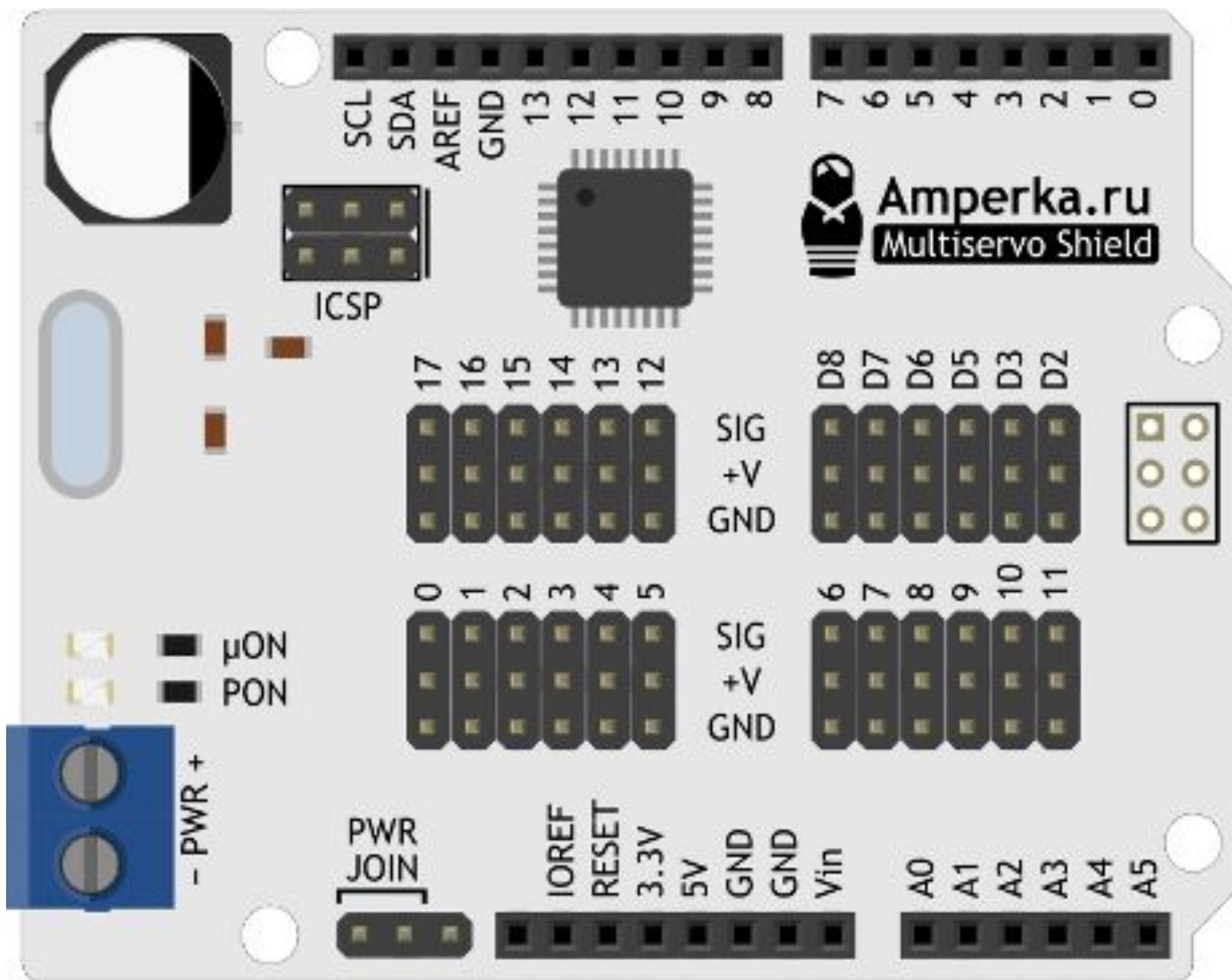
Количество сервоприводов: 18  
по I<sup>2</sup>C, 6 через Arduino

Потребляемый  
микроконтроллером ток: 15 мА

Диапазон рабочих температур:  
−40...+85 °C



# Устройство сервопривод



# Пример работы Multiservo Shield

Необходимо подключить библиотеки:

```
#include <Wire.h> // библиотека для работы с I2C
```

```
#include <Multiservo.h> // библиотека для работы с Multiservo Shield
```

Определяем пины — для каждого сервопривода нужно определить свой пин:

```
#define SCAN_SERVO_PIN 9
```

Создаем объект для работы с сервоприводом. Для каждого сервопривода нужно создать свой объект. Тип объекта — Multiservo, имя объекта — scanner\_servo:

```
Multiservo scanner_servo;
```



# Пример работы Multiservo Shield

## setup()

Подключаем Multiservo Shield по протоколу I2C к Arduino:

```
Wire.begin();
```

Определяем на каком пине будет работать сервопривод под именем scanner\_servo:

```
scanner_servo.attach(SCAN_SERVO_PIN);
```

## loop()

Даем команду сервоприводу scanner\_servo встать в положение 30°:

```
scanner_servo.write(30);
```

**ОБЯЗАТЕЛЬНО!!!** После команды scanner\_servo.write, делаем небольшую задержку для того, чтобы сервопривод успел выполнить команду:

```
delay(20);
```