

#### Состав схемы:

- [Arduino Nano ATmega328P 16 MHz 5V;](#)
- [SG92R 2,5 кг Micro 9g сервопривод](#) (на схеме SG90 1,25 кг);
- [Светочувствительный резистор LDR](#) (5528 GL5528 5537 5506 5516 5539 для Arduino);
- [Полимерная литиевая перезаряжаемая батарея 3,7 102050 1000 мАч Li-Po;](#)
- [Модуль зарядного устройства литиевой батареи TP4056 Type-c Micro USB 5 в 1A 18650;](#)
- [Регулируемый модуль питания повышающий \(между ардуинкой и платой заряда\) типа DC-DC SX1308](#) либо вместо платы заряда+повышайки использовать готовый модуль [Повышающий DC-DC преобразователь J5019 с 3У;](#)
- Резистор 100-120Ком;
- [Движковый переключатель](#) (они самые дешевые, можно любой переключатель) либо тумблер.

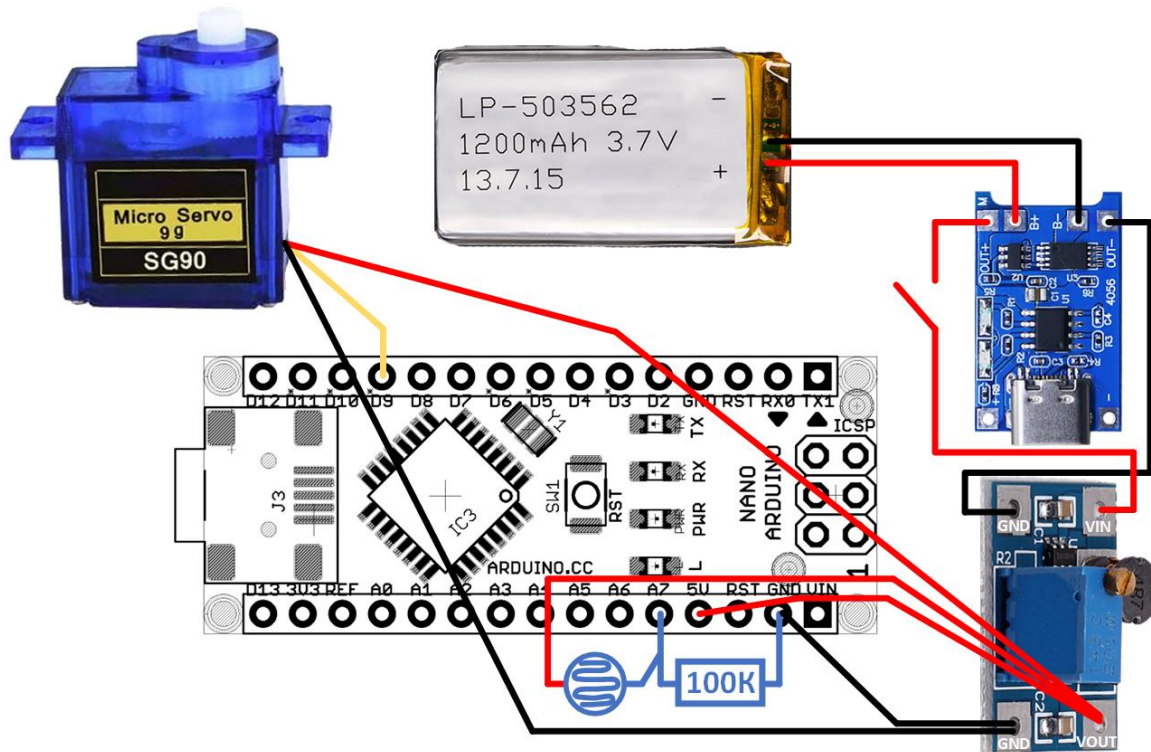


Рис.1 Схема подключения к плате Arduino Nano

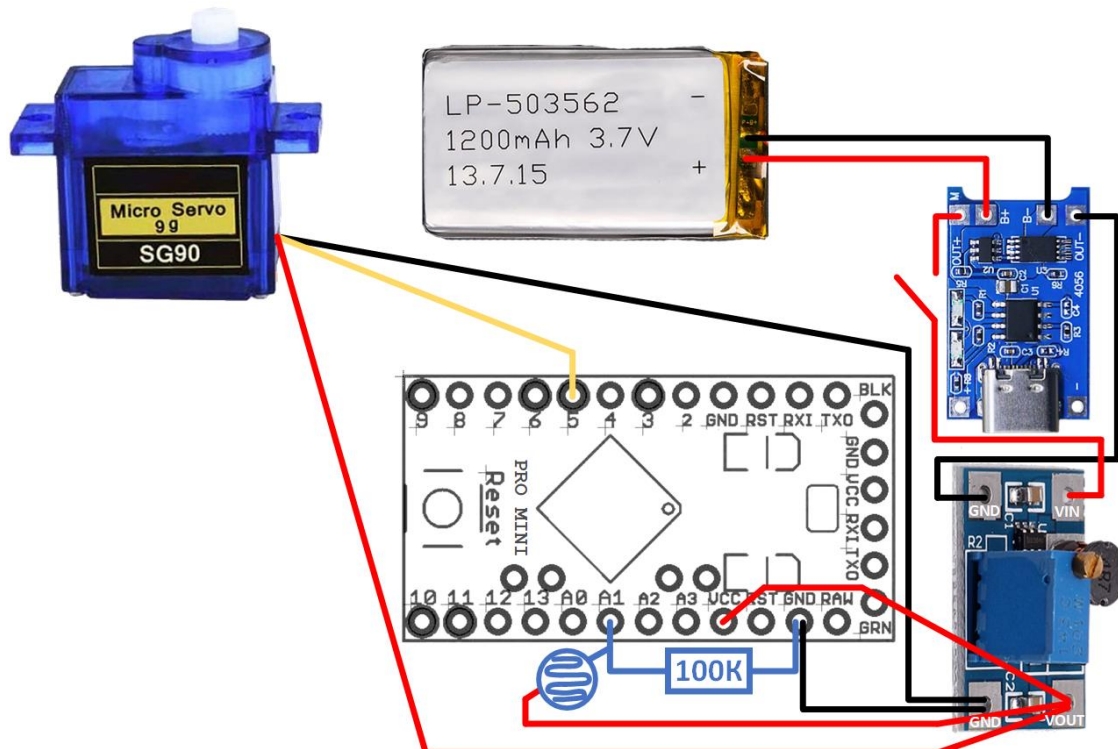


Рис.2 Схема подключения к Arduino Pro Mini

## Комментарии:

- При пайке использовать паяльную кислоту для очистки контактов, но ее затем необходимо обязательно отмывать, иначе припаянные провода очень быстро отвалятся. Проще использовать обычную канифоль, либо безотмывочный флюс;

- Перед пайкой ардуино к повышающему преобразователю нужно установить на преобразователе напряжение 4.5-5.0В (*подпаяли аккумулятор, мультиметром ткнули на Vout и GND повышайки, покрутили отверткой винтик до достижения нужного вольтажа, отпаяли батарею*). Больше 5.1В может повредить ардуино. Ввиду того, что у подстроечного резистора есть люфт, вследствие вибрации напряжение может уплыть, поэтому можно на него капнуть каплю суперклея для фиксации настройки;

- Также можно после настройки нужного напряжения выпаять преобразователь напряжения и померять сопротивление сначала между средним и левым контактом, а затем – средним и правым. По итогам этих значений сопротивлений подбираются максимально близкие резисторы и впаяются на плату. После пайки обязательно перемерять;

- Можно использовать провод [МГТФ 0.12-0.2](#) или [AWG26](#) (силиконовая оболочка, не плавится при пайке);

- Переключатель движковый, любой аналогичный [ПД9-1](#) или [SS-5](#);

- Были рекомендации по выпайке светодиодов и кнопки сброса ардуины для обеспечения дополнительной светомаскировки;

- Платы можно использовать как Arduino Nano/Pro Mini, так и Attiny 13/85/88;

### Возможные проблемы и решения:

1. Собранная схема с залитым скетчем ни под каким предлогом не хотела запускаться. А при питании от разъёма загрузки Ардуино срабатывала. В итоге она совсем перестала реагировать.

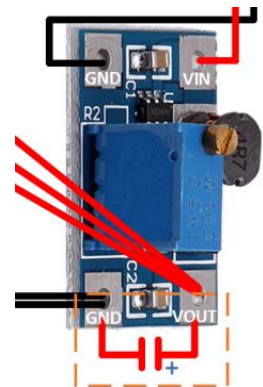
**Решение 1:** последовательно проверил работоспособность всех компонентов. Всё работает отдельно, а вместе ни-ни. Пересобрал схему внося небольшое дополнение. ([Совет увидел на Ютубе](#)) Есть такая проблемка, как невозможность Ардуино запуска сервопривода. Он вроде пытается заработать, а силы ему не хватает. Чтобы это обойти на плюс и минус питания сервопривода вешается электролитический конденсатор большой емкости (подойдет все, что больше или равно 6.3В и емкостью 100 мкФ. [Например, такой](#)). Питание более-менее выравнивается и серва начинает работать. ВАЖНО: больше 100мкф не стоит ставить. От 1000 и выше могут испортить USB-порт броском тока при включении.

**Решение 2:** если нет конденсатора, то можно попробовать поменять сервомотор. Мне помогла замена на серву из другой партии (закупки другого магазина).

2. Невозможно залить скетч в плату

Решение:

- в китайские платы на Атмеге часто надо заливать скетч через Old bootloader в Arduino IDE (Tools/Processor/ATmega 328P (Old Bootloader));
- может помочь перезагрузка компьютера/ноутбука;
- может помочь более короткий кабель;
- в зависимости от качества кабеля Type-C может помочь переворачивание на 180 градусов.



### По Ардуино:

1. Качаем Arduino IDE (<https://www.arduino.cc/en/software> – ссылки на скачивание справа, переходите на другую страницу, левая ссылка JUST DOWNLOAD - без донатов, правая – если хотите помочь проекту);
2. Устанавливаете программу из скачанного файла, запускаете;
3. Подключаете плату ардуины к компу, в программе нужно будет выбрать модель ардуинки и ее COM-порт в поле «SELECT BOARD» сразу под основным меню программы;
4. После этого копируете скетч либо сразу открываете готовый INO-файл;
5. Далее загружаете код в ардуину через меню Sketch/Upload (внизу появится окошко со статусом загрузки, ждем сообщения о завершении).

Так как плата находится под питанием от USB и если на ней уже распаяны и подключены все комплектующие, то очень удобно сразу же проверять работоспособность, не отключая от компа. Например, выставлять нужные углы поворота сервы. Что-то не устраивает – поменяли значение в скетче, повторно залили и сразу же пробуете (например, засвечивая фоторезистор).

### Пример простого скетча для однозарядного изделия:

```
#include <Servo.h>
Servo myservo;
int val;
void setup()
{
  Serial.begin(9600);
  myservo.attach(9); //Пин подключения сервы (тут D9)
}
void loop()
{
  val = analogRead(7); //Пин подключения фоторезистора (тут A7)
  Serial.println(val);
  if (val > 1010) //if above it will move to 45
  {
    myservo.write(270); //угол при свете
  }
  else if (val < 1015)
  {
    myservo.write(90); //угол без света
  }
  delay(250);
}
```

**Пример скетча для двухзарядки** (одинаково работает как с сервой 180 градусов, так и 360 градусов) на плате **Arduino Nano**:

```
#include <Servo.h>
Servo _SM1;
bool _gen1I = 0;
bool _gen1O = 0;
unsigned long _gen1P = 0UL;
bool _gen2I = 0;
bool _gen2O = 0;
unsigned long _gen2P = 0UL;
bool _count1I = 0;
int _count1_Value = 0;
int _swi1;
int _swi2;
int _swi3;
bool _tim1I = 0;
bool _tim1O = 0;
unsigned long _tim1P = 0UL;

void setup()
{
    _SM1.attach(9);                //Пин подключения сервы
}
void loop()
{
    if ((analogRead(7)) >= (800)) //Пин подключения фоторезистора (тут А7). Значение -
    800
    {
        if (_tim1I)
        {
            if (_isTimer(_tim1P, 100))
            {
                _tim1O = 1;
            }
        }
        else
        {
            _tim1I = 1;
            _tim1P = millis();
        }
    }
    else
    {
        _tim1O = 0;
        _tim1I = 0;
    }
    if (_tim1O)
    {
        if (!_count1I)
        {
            _count1I = 1;
            if(_count1_Value < 2)
            {
                _count1_Value = _count1_Value + 1;
            }
        }
    }
    else
    {
        _count1I = 0;
    }
    if(_count1_Value >= 2)
    {
        _count1_Value = 0;
    }
    if ((_count1_Value) == (0))
    {
        if (!_gen1I)
        {
            _gen1I = 1;
            _gen1O = 1;
            _gen1P = millis();
        }
    }
}
```

```

    }
}
else
{
    _gen1I = 0;
    _gen1O = 0;
}
if (_gen1I && _gen1O) _gen1O = !(_isTimer(_gen1P , 1000));
if ((_count1_Value) == (1))
{
    if (! _gen2I)
    {
        _gen2I = 1;
        _gen2O = 1;
        _gen2P = millis();
    }
}
else
{
    _gen2I = 0;
    _gen2O = 0;
}
if (_gen2I && _gen2O) _gen2O = !(_isTimer(_gen2P , 1000));
if(_gen2O)
{
    _swi2=180;                //Левый угол
}
else
{
    _swi2=0;
}
if (_gen2O)
{
    _SM1.write (_swi2);
}
if(((!( _gen1O)) && (!( _gen2O))))
{
    _swi3=92;                //Центральное положение. 2 градуса из-за зубьев штока
}
else
{
    _swi3=0;
}
if (((!( _gen1O)) && (!( _gen2O))))
{
    _SM1.write (_swi3);
}
if(_gen1O)
{
    _swi1=0;                //Правый угол
}
else
{
    _swi1=0;
}
if (_gen1O)
{
    _SM1.write (_swi1);
}
}
}
bool _isTimer(unsigned long startTime, unsigned long period)
{
    unsigned long currentTime;
    currentTime = millis();
    if (currentTime>= startTime)
    {
        return (currentTime>=(startTime + period));
    }
    else
    {
        return (currentTime >=(4294967295-startTime+period));
    }
}
}

```

**Пример скетча для двухзарядки** (одинаково работает как с сервой 180 градусов, так и 360 градусов) на плате

**Arduino Pro Mini:**

```
#include <Servo.h>
Servo _SM1;
bool _gen1I = 0;
bool _gen1O = 0;
unsigned long _gen1P = 0UL;
bool _gen2I = 0;
bool _gen2O = 0;
unsigned long _gen2P = 0UL;
bool _count1I = 0;
int _count1_Value = 0;
int _swi3;
int _swi1;
bool _tim1I = 0;
bool _tim1O = 0;
unsigned long _tim1P = 0UL;
int _swi2;
void setup()
{
    _SM1.attach(5); // Пин подключения сервы (тут 5)
}
void loop()
{
    //Плата:1
    if ((analogRead(1)) >= (800)) // Пин подключения фоторезистора (тут A1). Значение -
800
    {
        if (_tim1I)
        {
            if (_isTimer(_tim1P, 100))
            {
                _tim1O = 1;
            }
        }
        else
        {
            _tim1I = 1;
            _tim1P = millis();
        }
    }
    else
    {
        _tim1O = 0;
        _tim1I = 0;
    }
    if (_tim1O)
    {
        if (!_count1I)
        {
            _count1I = 1;
            if(_count1_Value < 2)
            {
                _count1_Value = _count1_Value + 1;
            }
        }
    }
    else
    {
        _count1I = 0;
    }
    if(_count1_Value >= 2)
    {
        _count1_Value = 0;
    }
    if ((_count1_Value) == (0))
    {
        if (!_gen1I)
        {
            _gen1I = 1;
            _gen1O = 1;
            _gen1P = millis();
        }
    }
}
```

```

    }
    else
    {
        _gen1I = 0;
        _gen1O = 0;
    }
    if (_gen1I && _gen1O) _gen1O = !(_isTimer(_gen1P , 1000));
    if ((_count1_Value) == (1))
    {
        if (! _gen2I)
        {
            _gen2I = 1;
            _gen2O = 1;
            _gen2P = millis();
        }
    }
    else
    {
        _gen2I = 0;
        _gen2O = 0;
    }
    if (_gen2I && _gen2O) _gen2O = !(_isTimer(_gen2P , 1000));
    if(_gen2O)
    {
        _swi2=160; // <--- Левый угол
    }
    else
    {
        _swi2=0;
    }
    if (_gen2O)
    {
        _SM1.write (_swi2);
    }
    if((((!(_gen1O)) && (!(_gen2O))))
    {
        _swi3=80; // <--- Центральное положение
    }
    else
    {
        _swi3=0;
    }
    if (((!(_gen1O)) && (!(_gen2O))))
    {
        _SM1.write (_swi3);
    }
    if(_gen1O)
    {
        _swi1=5; // <--- Правый угол
    }
    else
    {
        _swi1=0;
    }
    if (_gen1O)
    {
        _SM1.write (_swi1);
    }
}

bool _isTimer(unsigned long startTime, unsigned long period)
{
    unsigned long currentTime;
    currentTime = millis();
    if (currentTime>= startTime)
    {
        return (currentTime>=(startTime + period));
    }
    else
    {
        return (currentTime >=(4294967295-startTime+period));
    }
}

```