

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

Институт информационных систем и технологий

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Программирование специализированных вычислительных устройств

Отчет по лабораторной работе

«Основы программирования контроллеров. Применение циклов задержки и прерываний.»

Выполнил студент гр. ИДБ-22-06 Мустафаева П.М.

Проверил преподаватель Лаверычев М.А.

Москва 2022г.

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 1.1**

Используя механизм прерывания, описанный в теоретической части, модифицировать скетч управления источником света, реализованный в рамках выполнения задания №3 на прошлой лабораторной работе. Результаты моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ЗАДАНИЯ**

volatile boolean x = false;

void setup(){

pinMode(13, OUTPUT);

pinMode(2, INPUT);

attachInterrupt(0, pr, CHANGE);

}

void loop(){

}

void pr(){

x=!x;

digitalWrite(13, x);

}

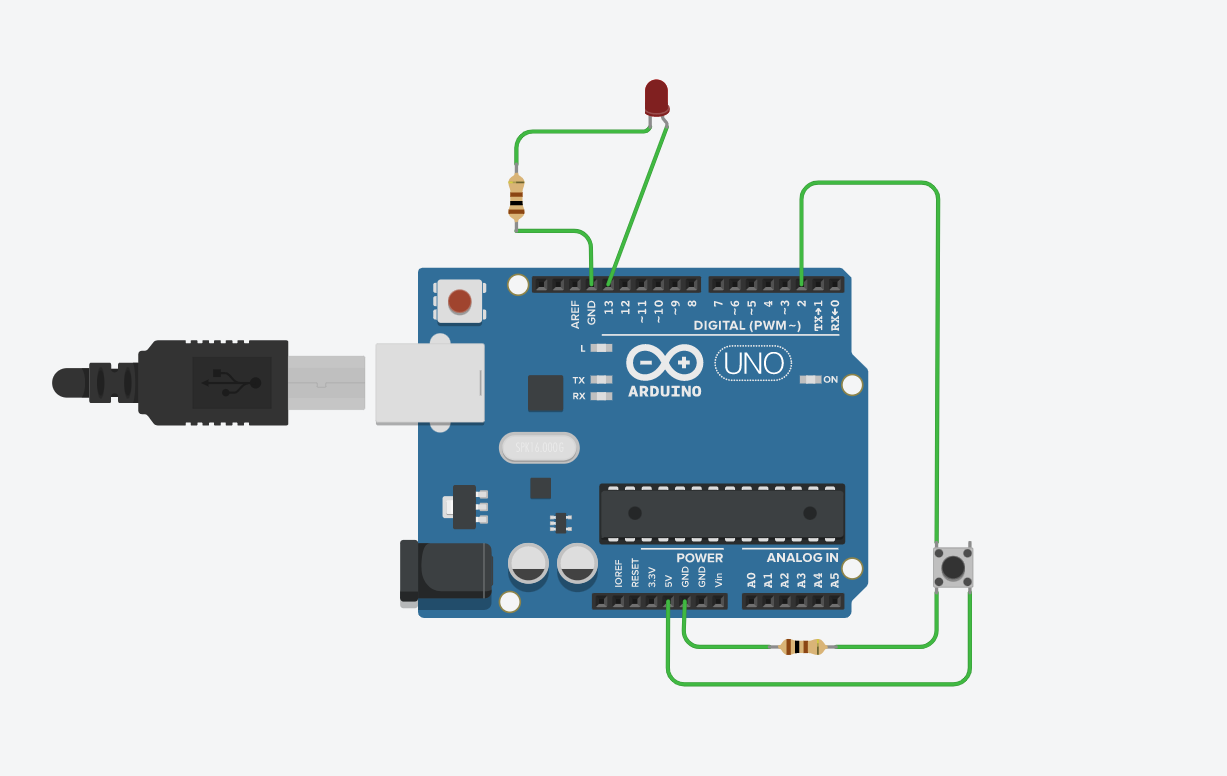


Рис. 1.1 Результат выполнения кода

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 1.2**

Написать скетч, реализующий процесс управления яркостью источника света с помощью делителя напряжения (потенциометра). Результат моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ КОДА**

void setup()

{

pinMode(5, OUTPUT);

pinMode(A0, INPUT);

}

void loop()

{

int x;

x = analogRead(A0) / 4;

analogWrite(5, x);

}

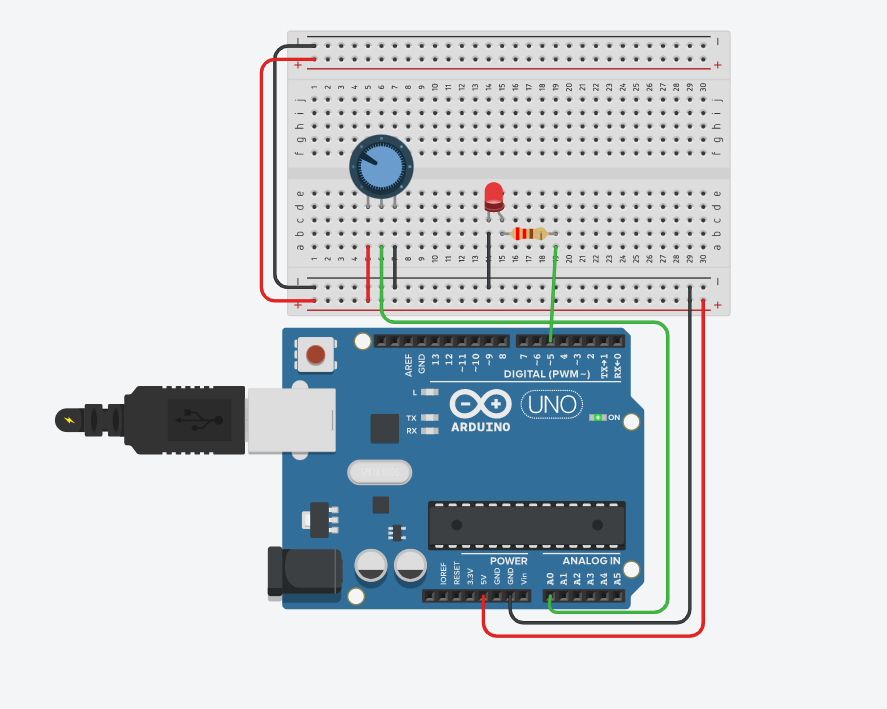


Рис. 1.2 Результат выполнения кода

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 2**

Для схемы, представленной на рисунке, напишите скетч, реализующий процесс плавного изменения яркости источников света при изменении положения потенциометра. При этом в крайних положениях потенциометра горел только один источник света. Результат моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ КОДА**

void setup()

{

pinMode(11, OUTPUT);

pinMode(3, OUTPUT);

pinMode(A0, INPUT);

Serial.begin(9600);

}

void loop(){

int x;

x = analogRead(A0) / 4;

analogWrite(11, 255 - x);

analogWrite(3, x);

Serial.println(x);

}

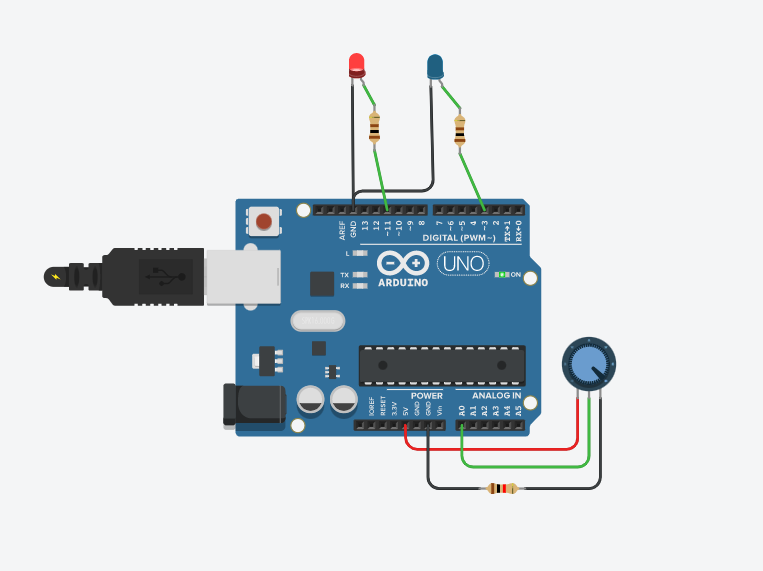


Рис.2.1 Результат выполнения кода

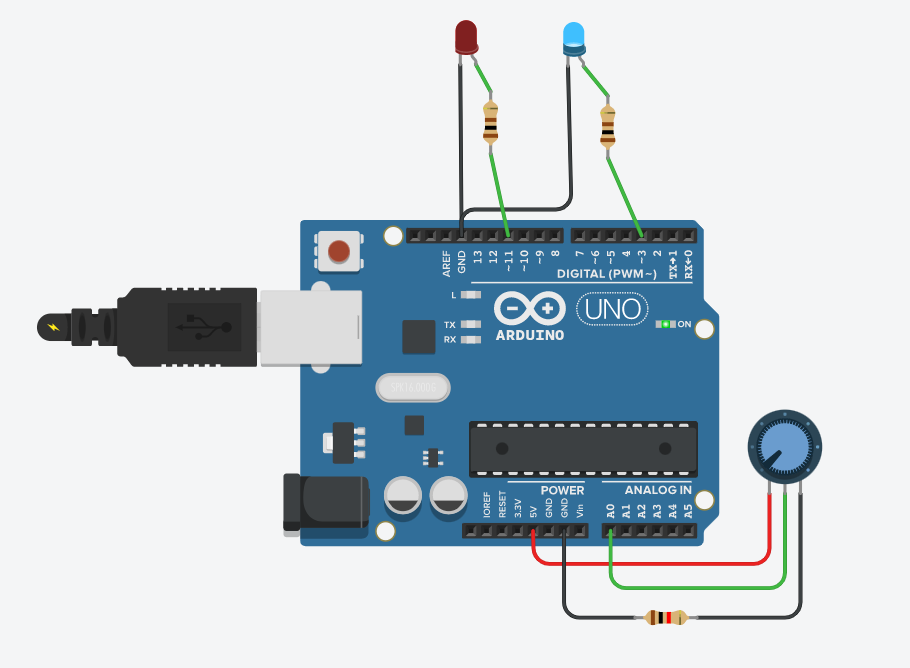


Рис. 2.2 Результат выполнения кода

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 3.1**

Используя возможности виртуальной среды моделирования, соберите схему управления сервоприводом с ограниченным углом вращения с помощью двух кнопок. Напишите скетч, реализующий управление сервоприводом в зависимости от однократного нажатия кнопки. При нажатии одной из кнопок привод должен приходить в движение и менять положение с текущего на конечное, соответствующего значению кнопки: первая – 0 градусов, вторая – 180 градусов. Результат моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ КОДА**

#include<Servo.h>

Servo servo1;

void setup(){

servo1.attach(10);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

}

int f1, f2;

void loop(){

f1 = digitalRead(7);

if(f1 == HIGH){

servo1.write(90);

}

f2 = digitalRead(8);

if(f2 == HIGH){

servo1.write(0);

}

}

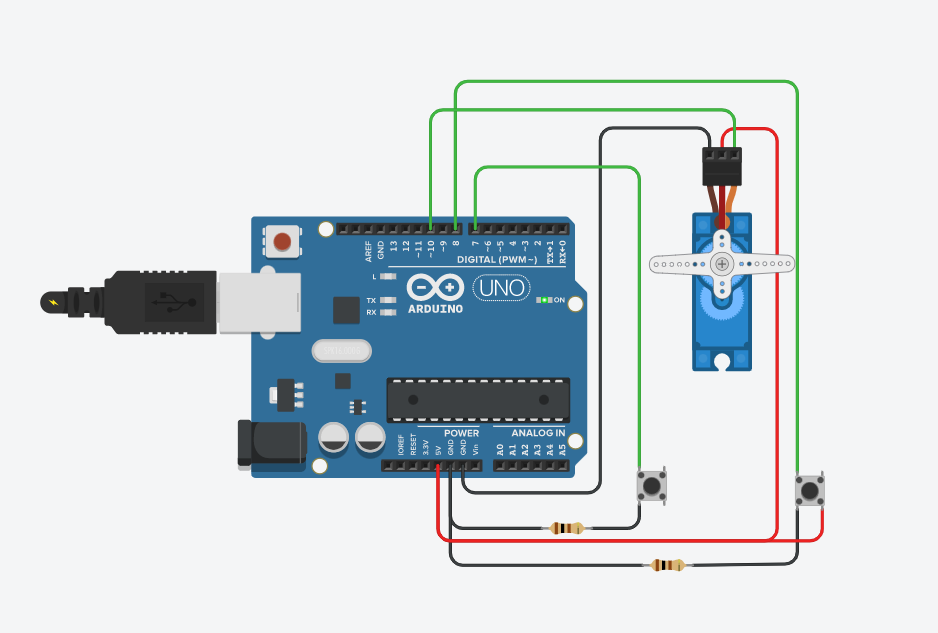


Рис. 3.1 Результат выполнения кода

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 3.2**

Модифицировать схему управления сервоприводом ограниченным углом вращения, заменив кнопки на потенциометр. Написать скетч, реализующий следующий закон управления: в зависимости от угла поворота потенциометра необходимо изменять положение вала сервопривода. Результат моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ КОДА**

#include<Servo.h>

Servo servo1;

int x;

void setup(){

servo1.attach(10);

pinMode(A0, INPUT);

}

void loop(){

x = analogRead(A0);

x = map(x, 0, 1023, 0, 180);

servo1.write(x);

}

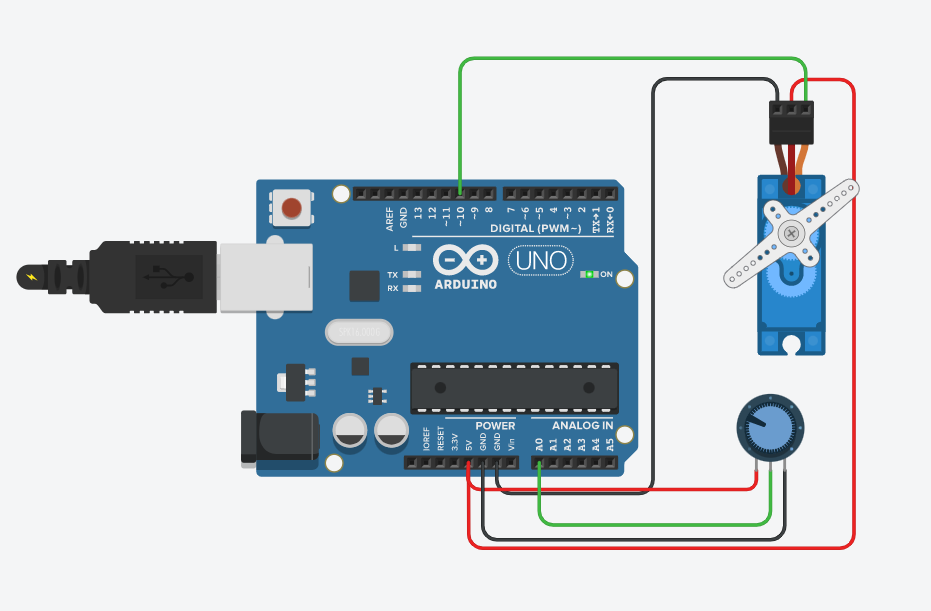


Рис. 3.2 Результат выполнения кода

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 3.3**

Внесите изменения в скетч управления сервоприводом, изменив закон управления: в зависимости от поворота потенциометра необходимо изменять скорость вращения вала сервопривода. Результат моделирования занесите в отчёт.

**РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ КОДА**

#include<Servo.h>

Servo servo1;

int x;

void setup(){

servo1.attach(10);

pinMode(A0, INPUT);

}

void loop(){

x = analogRead(A0);

x = map(x, 0, 1023, 0, 180);

servo1.write(x);

delay(20);

}

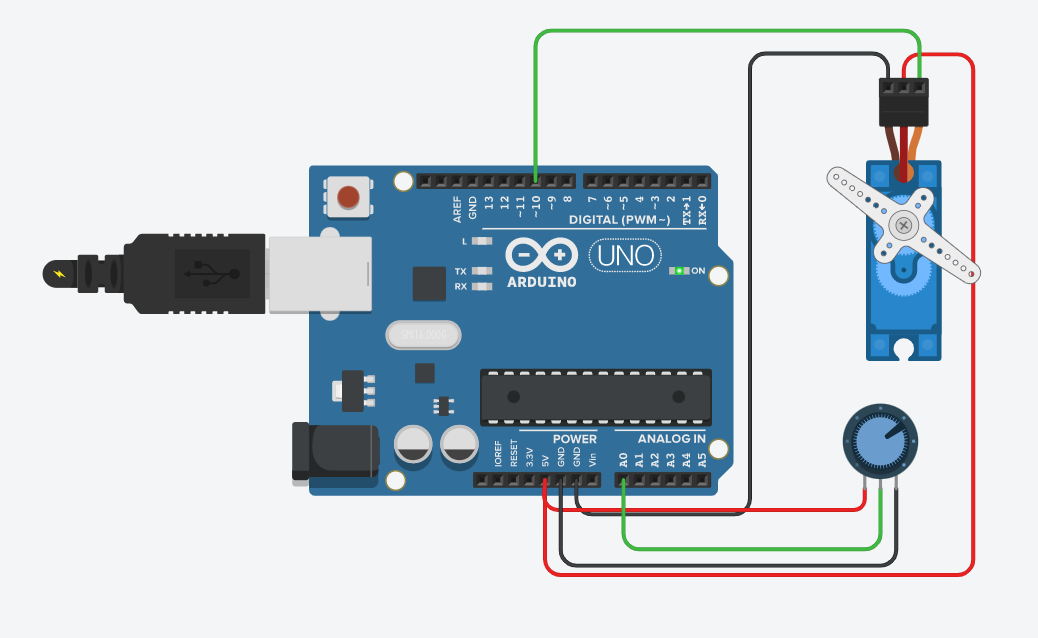


Рис. 3.3 Результат выполнения кода