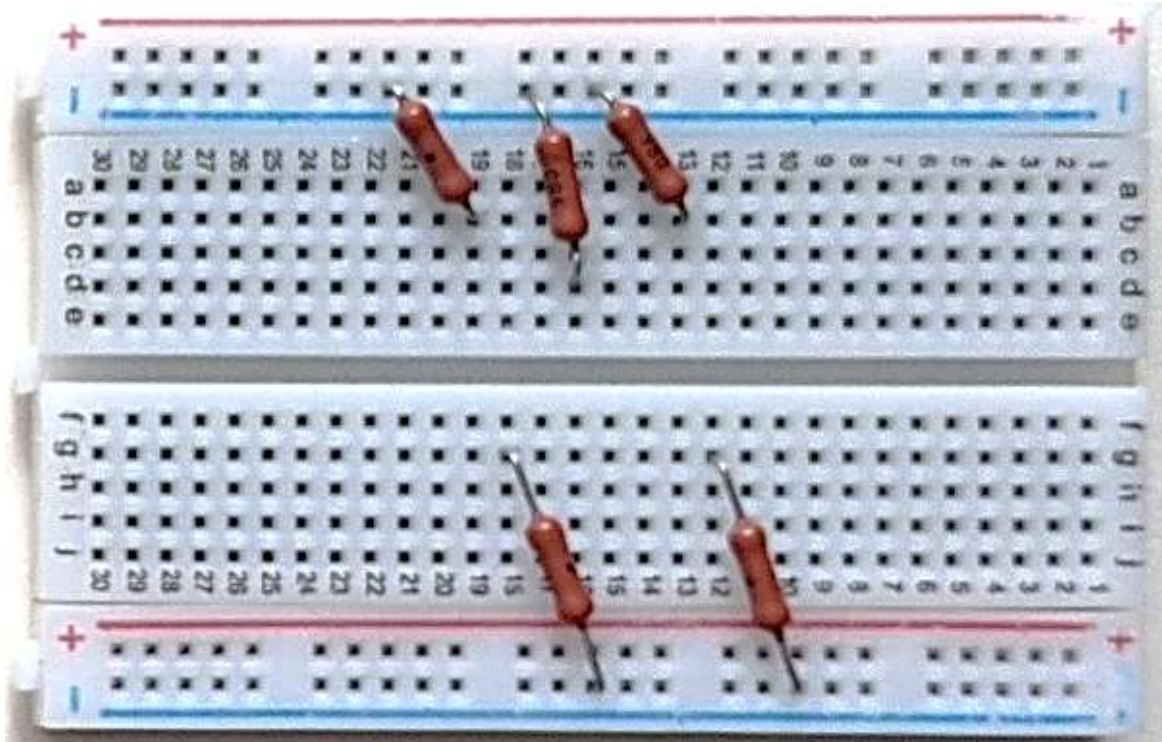


Создание фейерверков на базе Ардуино с использованием светодиодов

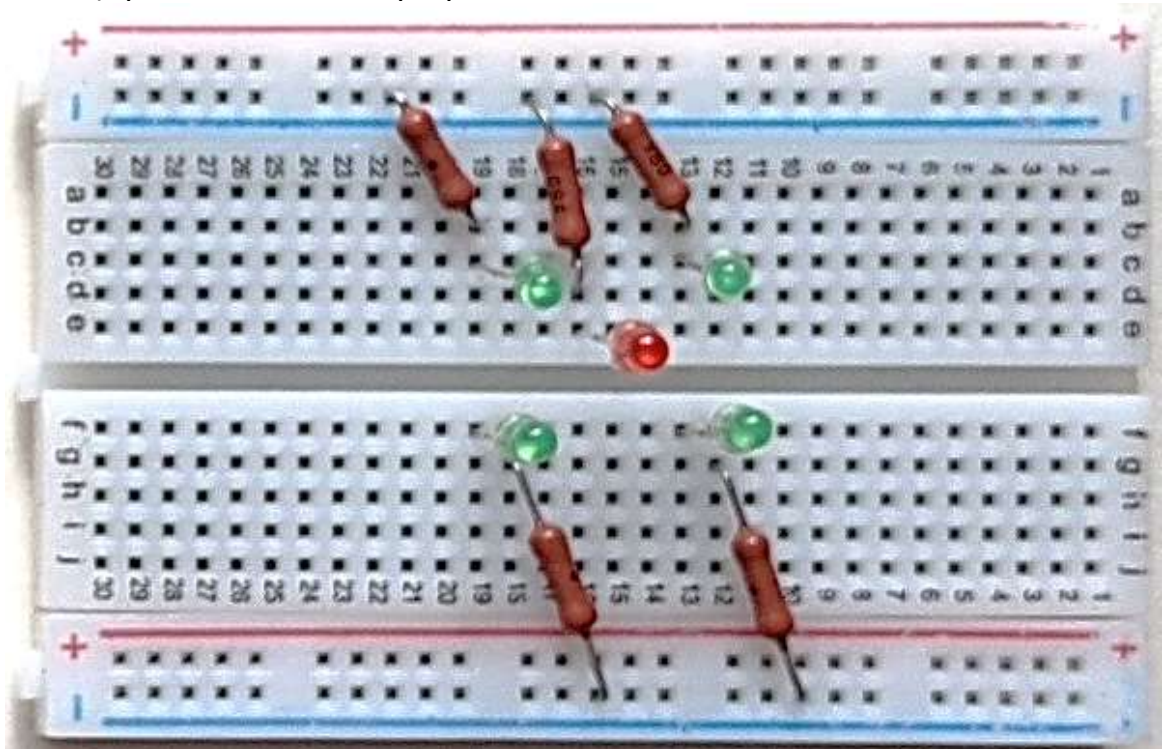
Приступим к сборке на макетной плате второго проекта мини-салюта, в котором светодиоды расположены в три ряда. Количество светодиодов прежние – пять.

Вы можете использовать свой вариант сборки, отличающийся от предложенного нами.

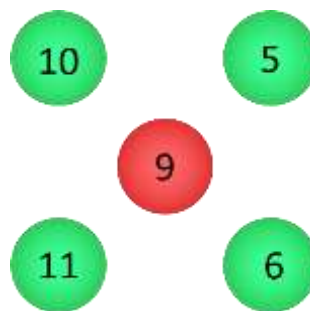
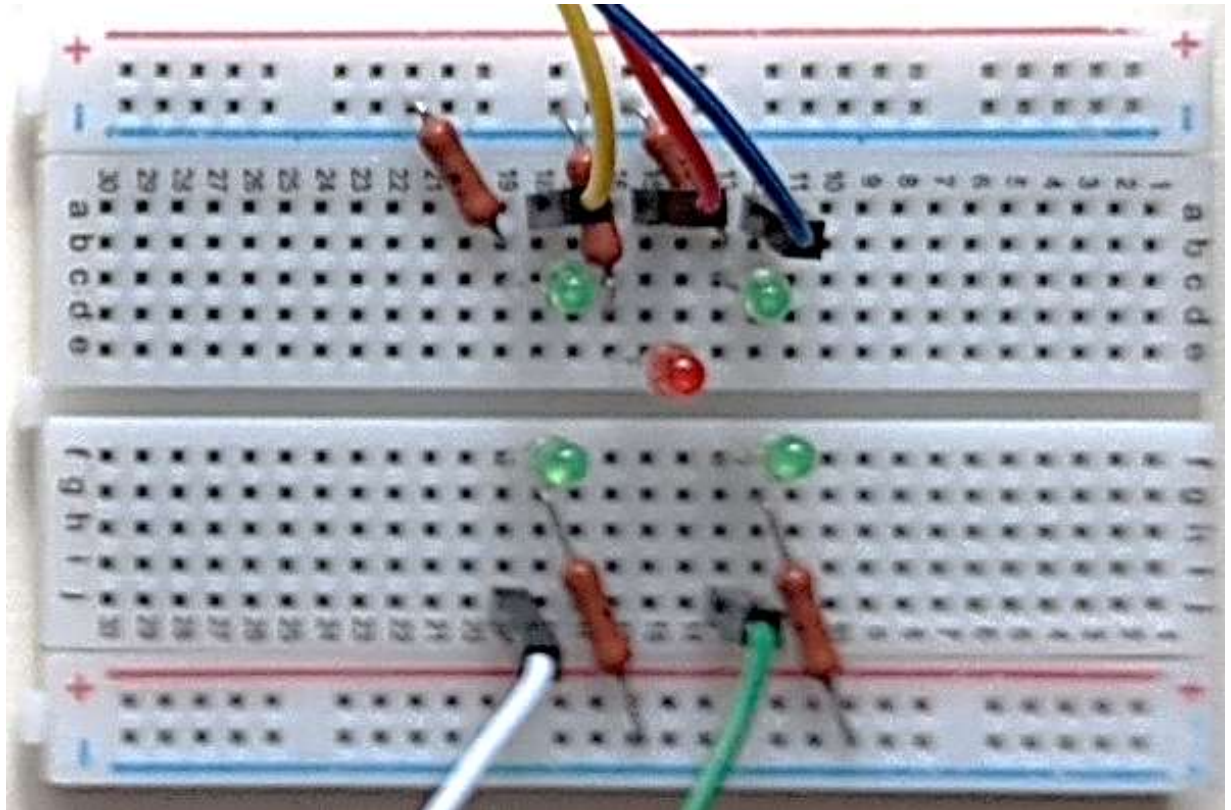
1. К макетной плате подключаем резисторы номиналом 220 Ом.

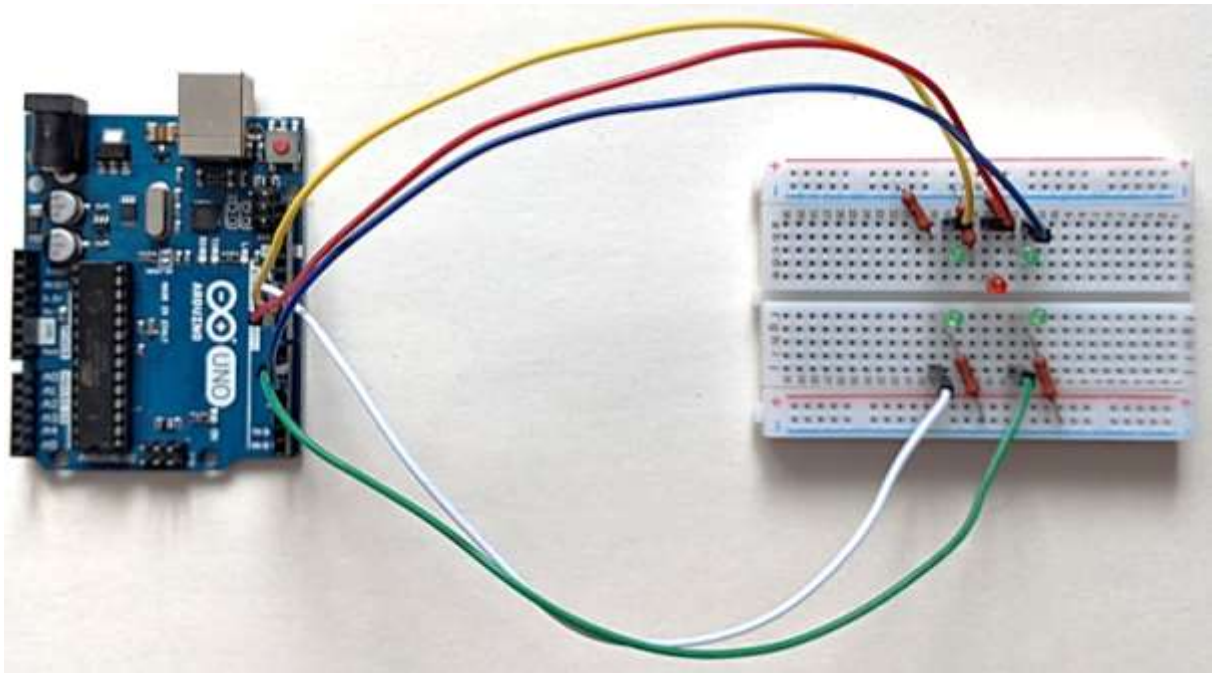


2. Затем подключаем к макетной плате светодиоды (4 зелёных, 1 красный), располагая их в три ряда.

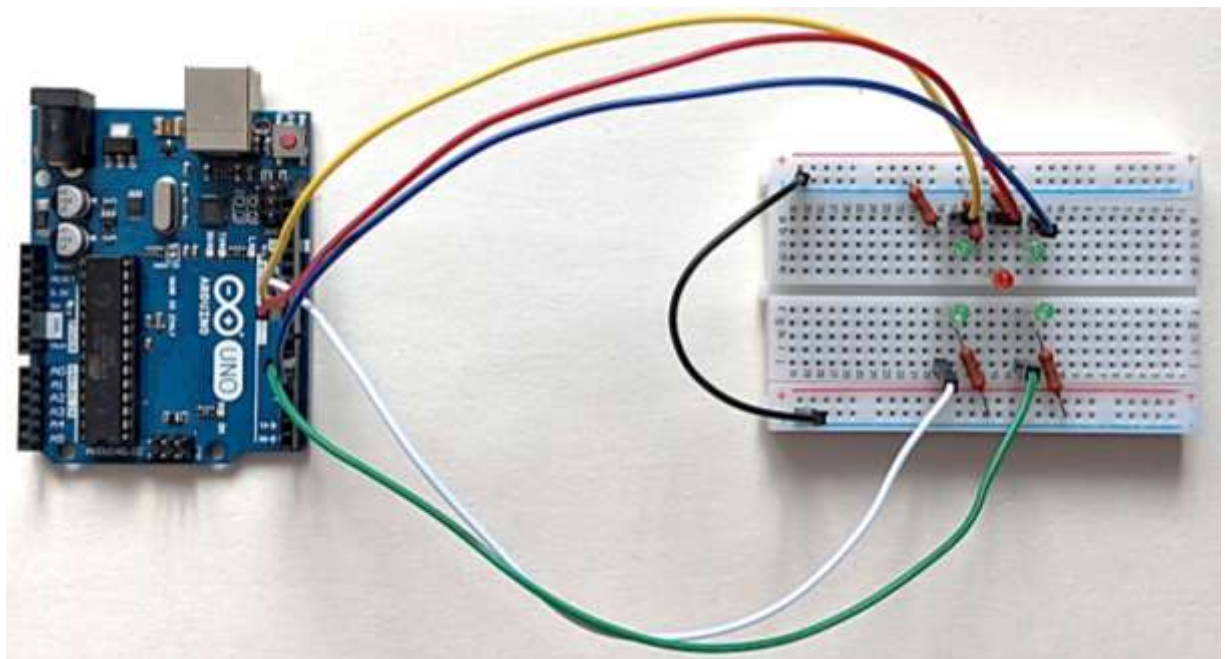


3. Подключаем плату Arduino UNO к макетной плате с помощью соединительных проводов к положительным выводам светодиодов. Нумерация выводов представлена на рисунке: зелёные светодиоды– пин 5 (синий провод), пин 6 (зелёный провод), пин 10 (жёлтый провод) и пин 11 (белый провод), красный светодиод– пин 9 (красный провод). Принцип подключения тот же, что описан ранее.

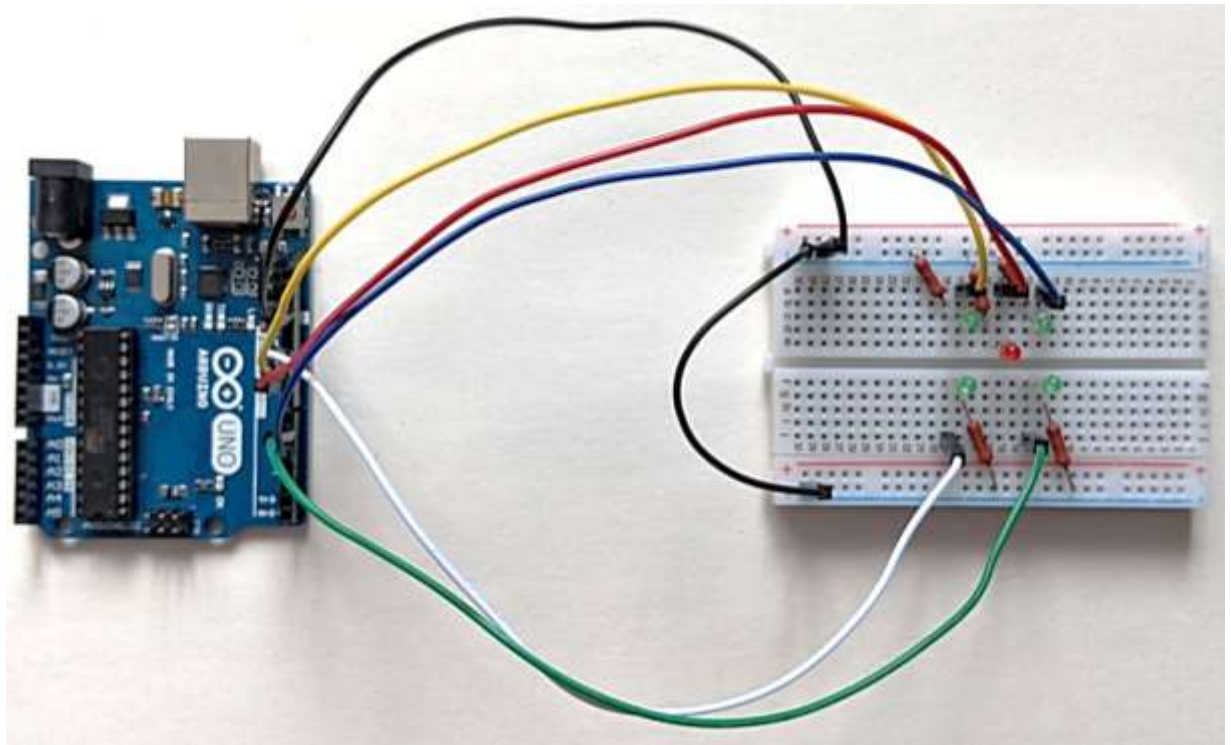




4. Соединяем «-» шины питания макетной платы, обозначенные синими линиями) между собой соединительным проводом черного цвета.



5. Вывод GND «-» подключаем соединительным проводом чёрного цвета ко второму выводу резисторов (синяя линия шины питания макетной платы).



6. С помощью кабеля USB (A — B) подключаем Arduino UNO к компьютеру.

Далее рассмотрим работу заранее подготовленного нами скетча (файл [3_1_1.ino](#)), для этого скомпилируем и загрузим его на плату Ардуино.

Наблюдаем мигание светодиодов в заданной последовательности.

(Видео [3_1_1](#)) <https://youtu.be/QlcXZxaSQHA>

Рассмотрим код программы.

```
// вводим 5 переменных и присваиваем им значения (номера выводов для подключения светодиодов)
int pin1 = 5;
int pin2 = 6;
int pin3 = 9;
int pin4 = 10;
int pin5 = 11;
```

Как и в скетчах предыдущих проектов вводим 5 переменных pin1 – pin5 и присваиваем им значения (номера выводов для подключения светодиодов: 5, 6, 9, 10, 11)

```
void setup()
{
    // пины устанавливаем в режим выхода
    pinMode (pin1, OUTPUT);
    pinMode (pin2, OUTPUT);
    pinMode (pin3, OUTPUT);
    pinMode (pin4, OUTPUT);
    pinMode (pin5, OUTPUT);
}
```

Устанавливаем выводы в режим выхода(OUTPUT), т.е. в режим источника напряжения.

```
void loop()
{
  digitalWrite(pin1, HIGH); // на pin1 подаётся высокий сигнал - логическая 1 (5В)
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, LOW); // на pin4 подаётся низкий сигнал - логический 0 (0В)
  digitalWrite(pin5, LOW);
  delay(500);
}
```

Мы помним, что функция **digitalWrite()** определяет, какой сигнал подается на указанный вывод. В данном фрагменте скетча на pin1, pin2, pin3 подаётся высокий сигнал (**HIGH**) – логическая 1 (напряжение 5 В) – подключенные к ним светодиоды включены. А на pin4 и pin5, подаётся низкий сигнал (**LOW**) – логический 0 (напряжение 0 В), т.е. светодиоды, подключенные к этим выводам выключены.

Функция **delay()** останавливает выполнение программы на 500 мс (0,5 с), т.е. в течение этого времени три светодиода светятся, а два выключены и никаких действий больше не происходит.

```
digitalWrite(pin1, LOW);
digitalWrite(pin2, LOW);
digitalWrite(pin3, HIGH);
digitalWrite(pin4, HIGH);
digitalWrite(pin5, HIGH);
delay(500);
}
```

После паузы 0,5 с выполняется следующий (последний в этом цикле) фрагмент программы, в результате которого светодиод, подключенный к выводу pin3 продолжает светиться, а также включаются светодиоды, подключенные к выводам pin4 и pin5, а светодиоды, подключенные к выводам pin1 и pin2 гаснут.

Через 0,5 с процедура loop выполняется снова (с первой строки цикла).

Таким образом, цикл **void loop()** повторяется бесконечно, пока плата Ардуино подключена к источнику питания.

С следующим примером мы увеличили скорость мигания светодиодов и отрегулировали их яркость (эффект затухания).

Откроем **файл 3_1_2.ino**, загрузим его на плату Ардуино. Мы видим, что здесь светодиоды мигают в той же последовательности, но в 2 раза быстрее и с эффектом затухания (**видео 3_1_2**) <https://youtu.be/guAZpYDxsu4>.

Рассмотрим изменения в скетче. На рисунке они отмечены линией зелёного цвета.

Мы помним, что за скорость мигания светодиодов отвечает функция **delay()**. В предыдущем скетче пауза была 0,5 с **delay(500)**, в данном скетче мы

изменили её на 0,25 с **delay(250)**, т.е. выполнение программы будет останавливаться на 0,25 с, а не на 0,5 с, как в предыдущем скетче.

За управление яркостью подключенного светодиода в скетче отвечает функция **analogWrite()**. Значения неполного напряжения подбирались нами экспериментальным путём.

```
void loop()
{
  digitalWrite(pin1, HIGH);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, LOW);
  delay(500);
}
```

```
digitalWrite(pin1, LOW);
digitalWrite(pin2, LOW);
digitalWrite(pin3, HIGH);
digitalWrite(pin4, HIGH);
digitalWrite(pin5, HIGH);
delay(500);
}
```

```
void loop()
{
  analogWrite(pin1, 40);
  analogWrite(pin2, 40);
  analogWrite(pin3, 40);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, LOW);
  delay(250);

  digitalWrite(pin1, HIGH);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  analogWrite(pin4, 10);
  analogWrite(pin5, 10);
  delay(250);
}
```

```
analogWrite(pin1, 10);
analogWrite(pin2, 10);
analogWrite(pin3, 40);
analogWrite(pin4, 40);
analogWrite(pin5, 40);
delay(250);
```

```
digitalWrite(pin1, LOW);
digitalWrite(pin2, LOW);
digitalWrite(pin3, HIGH);
digitalWrite(pin4, HIGH);
digitalWrite(pin5, HIGH);
delay(250);
}
```

Рассмотрим изменения в скетче, внесённые для реализации эффекта затухания.

Мы видим, что количество фрагментов в цикле **loop()** увеличилось с 2-х до 4-х.

Мы дублировали первый фрагмент цикла **loop()** и заменили функцию **digitalWrite()** на **analogWrite()** в 1-м фрагменте программы для pin1, pin2 pin3, т.е. выдаём на светодиоды, подключенные к этим выводам вместо 5 В

неполное напряжение (ШИМ-сигнал) 40 (0,78 В), тем самым уменьшая яркость светодиодов, подключённых к выводам pin1, pin2, pin3 почти в 6,5 раз. А светодиоды, подключенные к выводам pin4, pin5 выключены, как и в первом варианте скетча.

Для второй копии фрагмента программы мы оставили без изменения функцию digitalWrite() для pin1, pin2, pin3, т.е. подаём на них напряжение 5 В, но заменили функцию digitalWrite() на analogWrite () для pin4, pin5, т.е. выдаём на светодиоды, подключенные к этим выводам вместо 0 В неполное напряжение (ШИМ-сигнал) 10 (0,2 В). Т.е. после паузы 0,25 с яркость светодиодов, подключённых к выводам pin1, pin2, pin3 возрастает до максимума, а светодиоды, подключённые к выводам pin4, pin5 включаются, но яркость их снижена в 50 раз.

В 3-м добавленном нами фрагменте программы, мы использовали для всех выводов функцию analogWrite () для pin1, pin2 значение аргумента равно 10 (0,2 В), а для pin3, pin4 и pin5 – 40 (0,78 В). Т.е. после паузы 0,25 с яркость светодиодов, подключённых к выводам pin1, pin2 снижается в 4 раза, яркость светодиода, подключенного к выводу pin3 снижается в 6, 5 раз, а у светодиодов, подключённых к выводам pin4, pin5 яркость в 4 раза возрастает.

4-й (последний в этом цикле) фрагмент программы такой же, как и в первом варианте скетча, здесь для всех выводов используется функция digitalWrite(). Т.е. после паузы 0,25 с светодиоды, подключенные к выводу pin1 и pin2 гаснут, а светодиоды, подключенные к выводам pin3, pin4 и pin5 продолжают светиться, но их яркость возрастает до максимума.

Через 0,25 с процедура loop выполняется снова (с первой строки цикла).

Таким образом, при выполнении данного скетча создаётся эффект затухания, в отличие от предыдущей программы, в результате работы которой светодиоды просто включались и выключались в заданной последовательности.

Мы подготовили несколько вариантов светодиодных мини-салютов, которые представлены ниже. Вы можете посмотреть видео, скачать скетчи к каждому из них и самостоятельно их проанализировать.

Поэкспериментируйте, внося в скетчи изменения значений аргументов функций delay, digitalWrite, analogWrite.

Ссылки на видео:

3_2_1 – <https://youtu.be/bfFnRXRMn-I>

3_2_2 – <https://youtu.be/TFzhRBKy44Q>

3_3_1 – <https://youtu.be/jicgPicLlal>

3_4_1 – https://youtu.be/h_1EKD1ieBM