

Создание фейерверков на базе Ардуино с использованием светодиодов

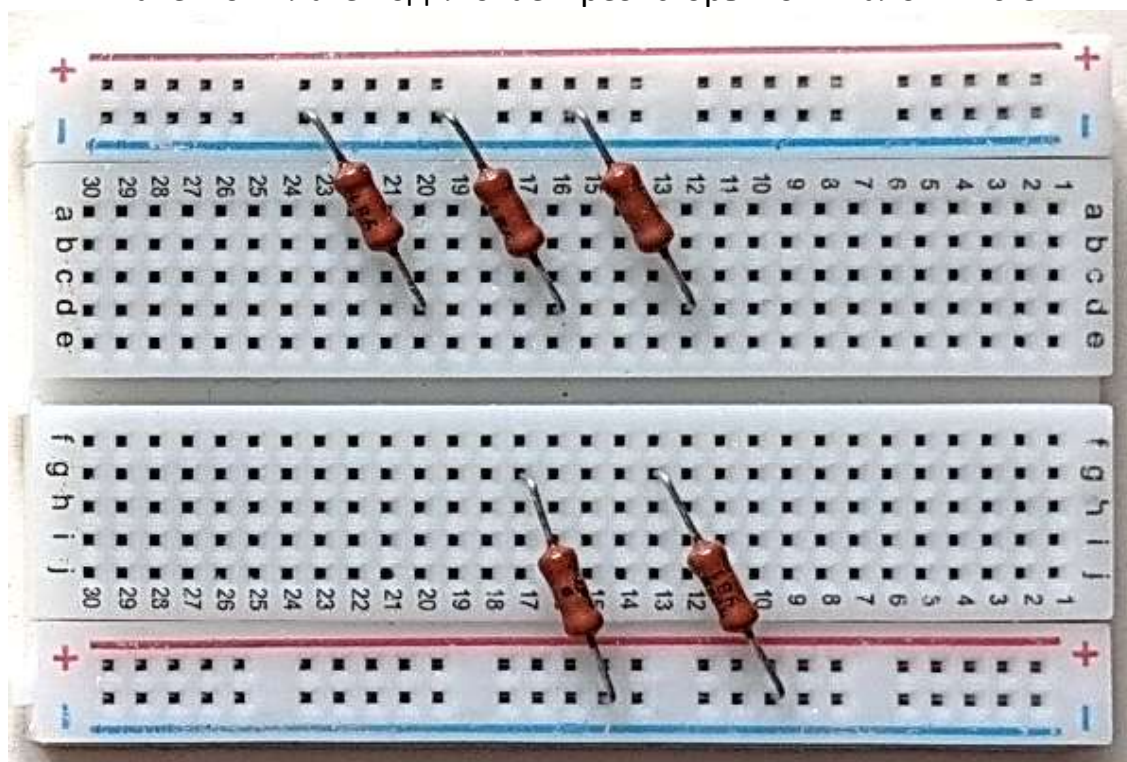
Продолжаем работу над проектами с использованием светодиодов.

Для создания более интересных визуальных эффектов мы соберём и запрограммируем другие варианты схем, в которых светодиоды на макетной плате расположены несколько иначе. Принципиальная электрическая схема и элементная база та же, что и в предыдущем проекте.

Приступим к сборке на макетной плате первого проекта мини-салюта, в котором светодиоды расположены в два ряда. Количество светодиодов прежнее – пять.

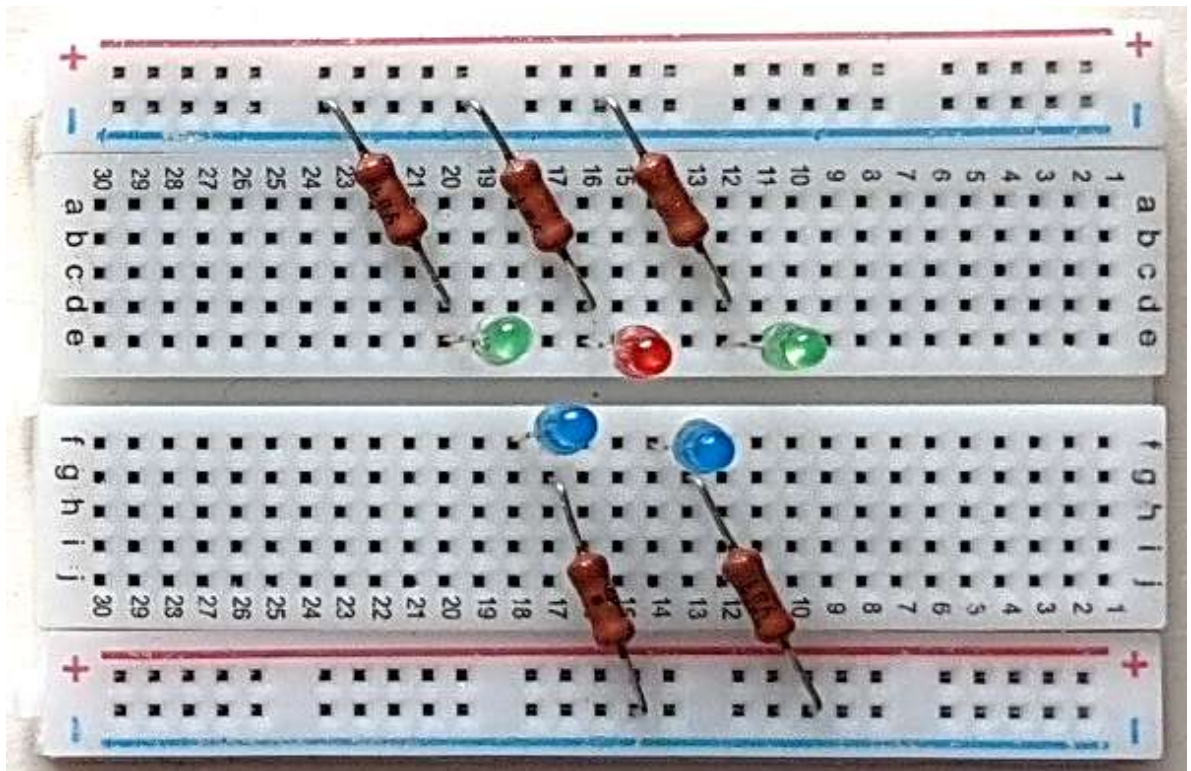
Вы можете использовать свой вариант сборки, отличающийся от предложенного нами.

1. К макетной плате подключаем резисторы номиналом 220 Ом.



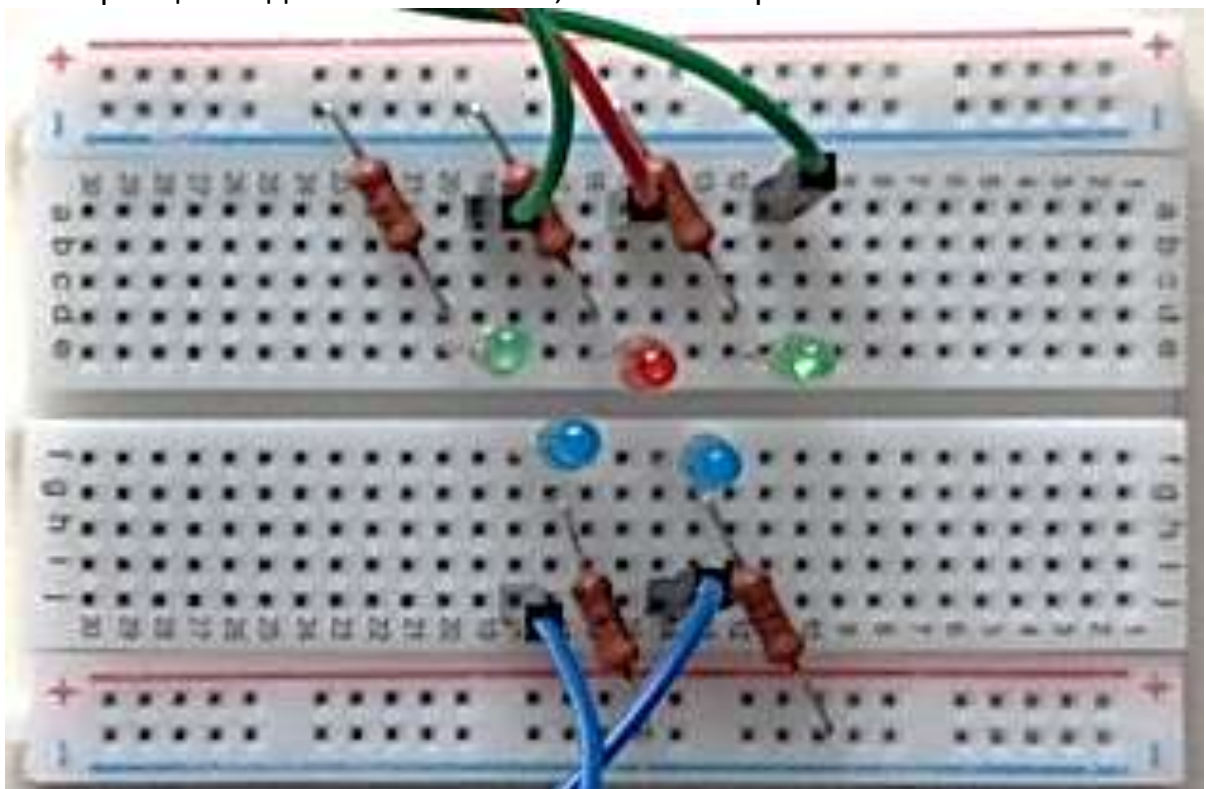
2. Затем подключаем к макетной плате разноцветные светодиоды, располагая их в два ряда. При подключении соблюдаем полярность: Анод (+) (длинный вывод) светодиода к соединительному проводу, а катод (-) (короткий вывод) – к резистору.

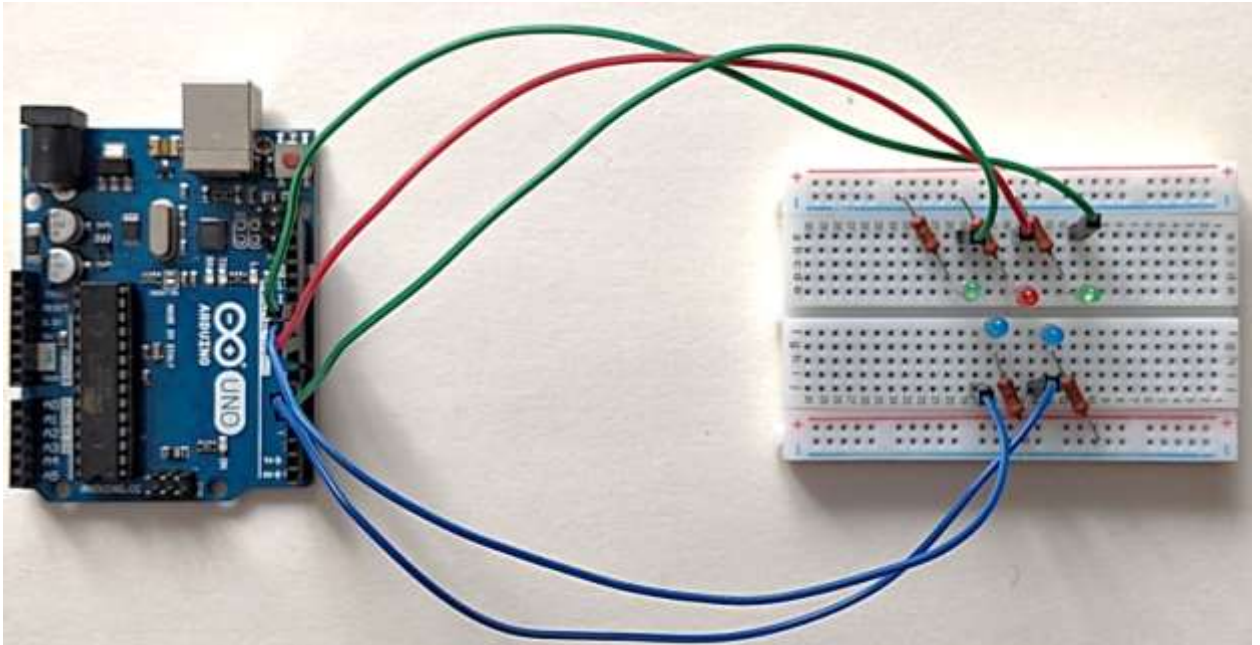
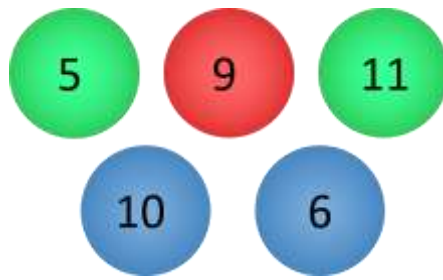
Вы можете подключить резисторы и светодиоды к макетной плате иначе, чем на фотографии. Главное, придерживаться правила: отрицательный вывод светодиода и резистор должны быть подключены по одной линии, а количество отверстий на макетной плате между ними может быть различным. Второй вывод резистора обязательно должен быть подключён к любому отверстию «-» шины питания макетной платы (обозначен синей линией).



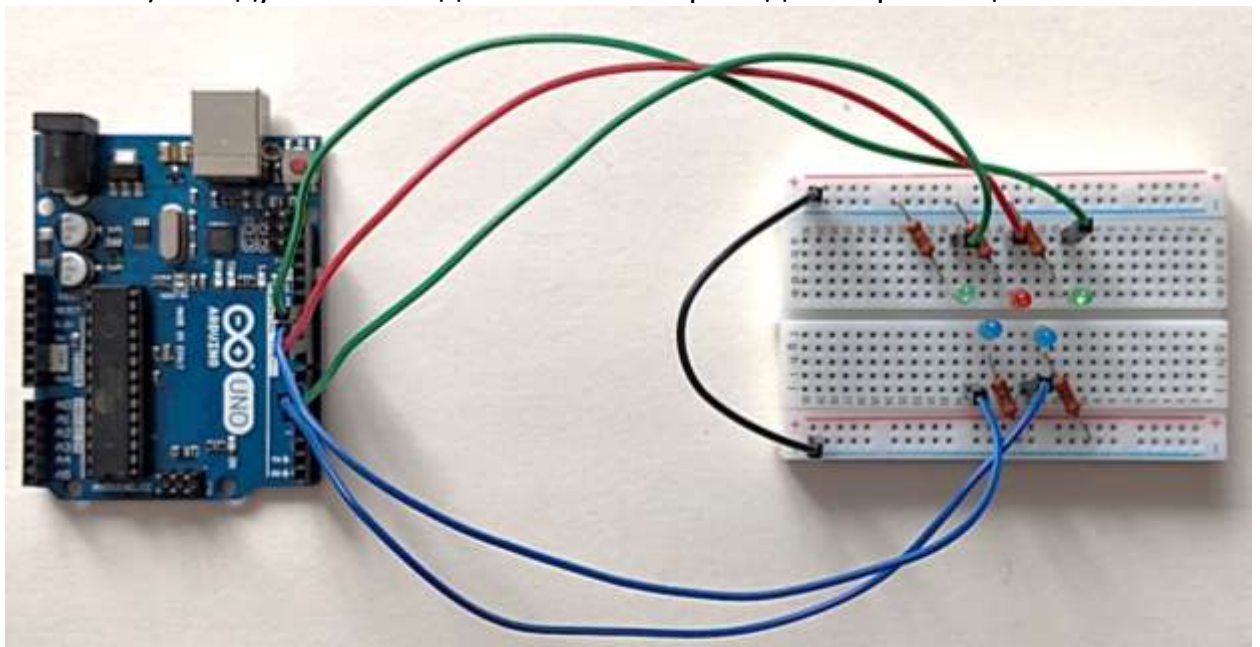
3. Подключаем плату Arduino UNO к макетной плате с помощью соединительных проводов к положительным выводам светодиодов (цвет соединительного провода соответствует цвету светодиода). Нумерация выводов представлена на рисунке: зелёные светодиоды – пин 11 и пин 5, синие светодиоды – пин 10 и пин 6, красный светодиод – пин 9.

Принцип подключения тот же, что описан ранее.

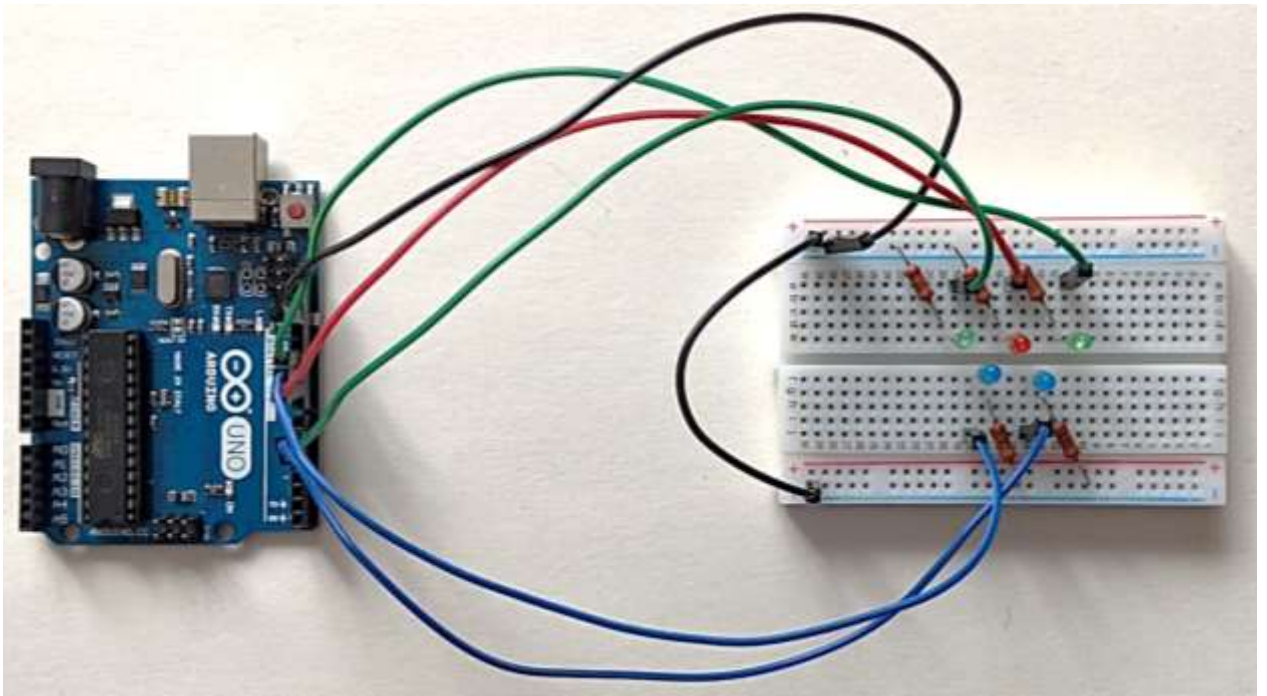




4. Соединяем «-» шины питания макетной платы, обозначенные синими линиями) между собой соединительным проводом черного цвета.



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.

Теперь рассмотрим работу заранее подготовленного нами скетча (файл [2_1_1.ino](#)), для этого скомпилируем и загрузим его на плату Ардуино.

Наблюдаем мигание светодиодов в заданной последовательности.

(Видео [2_1_1](#)) <https://youtu.be/Sk2Ps7WU-xs>

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, поскольку в данном случае микроконтроллер будет управлять светодиодом.

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

Мы помним, что функция **digitalWrite()** определяет, какой сигнал подается на указанный вывод. В данном фрагменте скетча на pin1, pin2, pin4, pin5, подаётся низкий сигнал (**LOW**) – логический 0 (напряжение 0 В) и светодиоды, подключенные к этим выводам платы, погаснут. А на pin3 подаётся высокий сигнал 1 (**HIGH**) (напряжение 5 В) – подключенный к нему светодиод загорится.

Функция **delay()** останавливает выполнение программы на 500 мс (0,5 с), т.е. в течение этого времени четыре светодиода выключены, а 1 включен и никаких действий больше не происходит.

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

После паузы 0,5 с выполняется следующий фрагмент программы, в результате которого светодиоды, подключенные к выводам pin1 и pin5 по-прежнему выключены, светодиод, подключенный к выводу pin3 продолжает гореть, а также загорятся светодиоды, подключенные к выводам pin2 и pin4.

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

После очередной паузы 0,5 с светодиоды, подключенные к выводам pin2 и pin4 продолжают светиться, также зажигаются светодиоды, подключенные к выводам pin1 и pin5, а светодиод, подключенный к выводу pin3 гаснет.

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

Через 0,5 с выполняется следующий (последний в этом цикле) фрагмент программы, в результате которого вслед за светодиодом, подключенным к выводу pin3 гаснут светодиоды, подключенные к выводам pin2 и pin4, а светодиоды, подключенные к выводам pin1 и pin5 продолжают светиться. 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 примере мы увеличили скорость мигания светодиодов и отрегулировали их яркость (эффект затухания).

Откроем **файл 2_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 2_1_2**) <https://youtu.be/QujDsPEqAXI>

Рассмотрим изменения в скетче. На рисунке они подчёркнуты 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()**.

Рассмотрим изменения в скетче, внесённые для реализации эффекта затухания. Значения неполного напряжения подбирались нами экспериментальным путём.

Мы видим, что 1-й фрагмент скетча в цикле **loop()** остался без изменений.

Во 2-м фрагменте скетча изменилась одна функция. Вместо **digitalWrite(pin3, HIGH)** здесь присутствует функция **analogWrite(pin3, 10)**, т.е. в данном случае выдаём на светодиод, подключенный к выводу pin3 вместо 5 В неполное напряжение (ШИМ-сигнал) 10 (0,2 В), за счёт этого уменьшается яркость светодиода.

```

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

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

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

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

```

```

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

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

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

    analogWrite (pin1, 1);
    digitalWrite (pin2, LOW);
    digitalWrite (pin3, LOW);
    digitalWrite (pin4, LOW);
    analogWrite (pin5, 1);
    delay (250);
}

```

В 3-м фрагменте программы мы также заменили функцию digitalWrite() на analogWrite(), но для pin2 и pin4, т.е. в данном случае мы выдаём на светодиоды, подключенные к этим выводам вместо 5 В неполное напряжение (ШИМ-сигнал) 10 (0,2 В), за счёт этого уменьшается яркость светодиодов, подключённых к портам pin2 и pin4.

В 4-м фрагменте программы мы также заменили функцию digitalWrite() на analogWrite(), но для pin1 и pin5. Но здесь мы выдаём на светодиоды, подключенные к этим выводам, вместо 5 В неполное напряжение (ШИМ-

сигнал) 1 (0,02 В), за счёт чего значительно уменьшается яркость светодиодов, подключённых к портам pin1 и pin5.

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:

2_2_1 – https://youtu.be/ZDmq4CpyH_E

2_2_2 – <https://youtu.be/f4An6bv5m14>

2_3_1 – https://youtu.be/NgERO9yC6_A