**Лабораторная работа № 1**

**Управление яркостью трехцветного светодиода с клавиатуры**

В этом примере мы будем управлять микроконтроллером нажатием клавиш клавиатуры

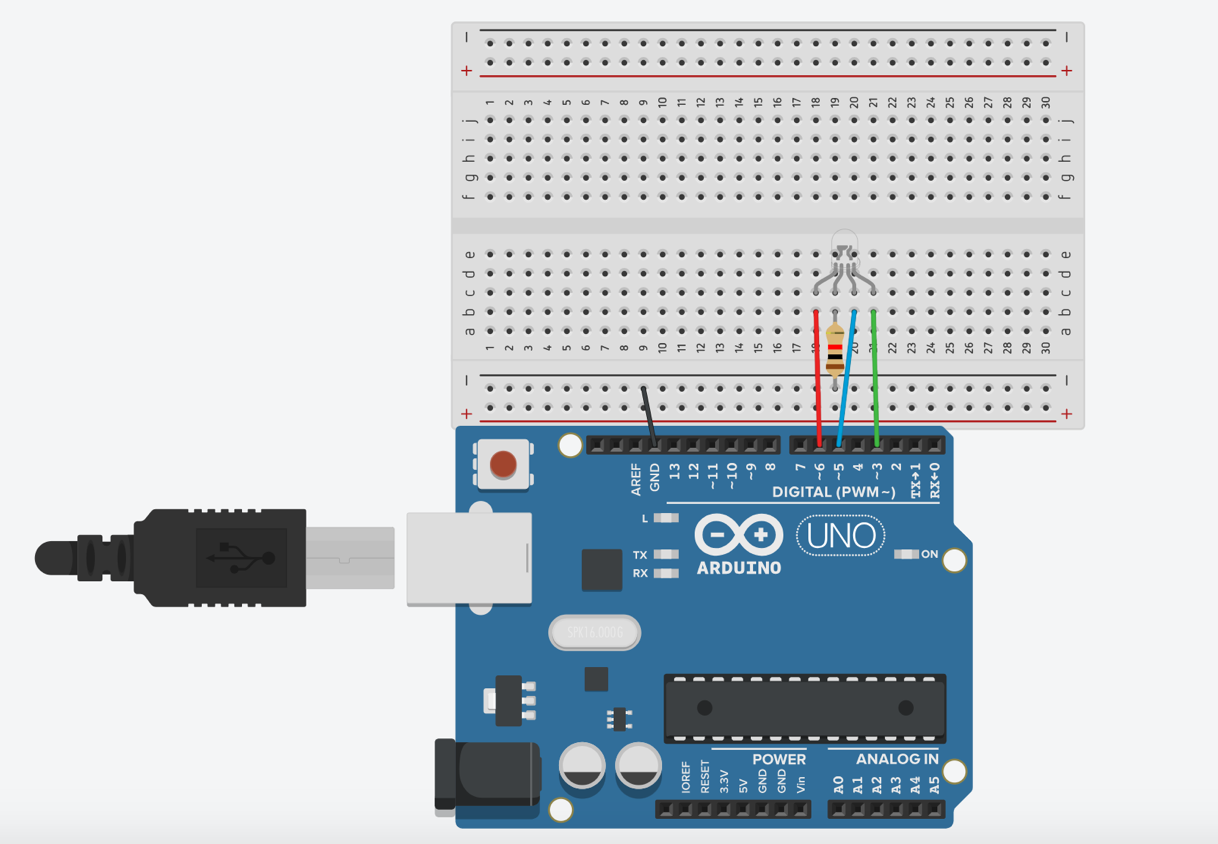
компьютера. Этот проект будет работать на любой совместимой с Arduino платой — при условии, что она оснащена тремя контактами цифрового вывода, с которыми можно использовать команду analogWrite().

**Требуемые компоненты**

* Трехцветный (RGB) светодиод с общим катодом, 1 шт.
* Резистор на 220 Ом, 1 шт.
* Совместимая с Arduino плата (на рис. 2.8, вверху,
* показана плата MKR1000, а внизу — плата Arduino 101), 1 шт. Используемые возможности платы: УАПП (UART), вывод ШИМ (PWM).
* Беспаечная макетная плата, 1 шт.
* Персональный компьютер.

1. **Подключение платы**

Подключение анодов трехцветного светодиода к выводам ШИМ микроконтроллерной платы: для платы Arduino 101 (внизу) — 3, 5 и 6.



1. **Создаем коммуникационный протокол**

Итак, схему мы собрали, теперь нужно решить, как нам взаимодействовать с микроконтроллером, чтобы управлять светодиодами. Для этого нам требуется создать коммуникационный протокол, который в нашем случае будет очень простым:

* чтобы выбрать цвет светодиода, посылаем первую букву требуемого цвета в нижнем регистре: r, g или b;
* чтобы задать уровень яркости выбранного светодиода, посылаем цифру: от 0 до 9.

1. **Код**

1.Прежде всего, надо задать константы, в которых будут храниться номера контактов, к которым подключены аноды светодиода (эти значения устанавливаются в зависимости от типа платы). Также нужно задать переменные: для хранения номера контакта текущего светодиода и для хранения значения яркости:

// Управление трехцветным светодиодом

// Программа управляет трехцветным RGB светодиодом, чьи выходы присоединены к коннекторам:

// R (красный) - 6

// G (зеленый) - 3

// В (синий) - 5

// Вначале обозначаем список констант

const int greenPin = 3; // инициализируем пин под красный цвет

const int redPin = 6; // инициализируем пин под зеленый цвет

const int bluePin = 5; // инициализируем пин под синий цвет

int currentPin = 0; // текущий контакт для задания яркости

int brightness = 10; // контакт показывающий текущий уровень яркости

2.Далее метод setup()запускает процедуру последовательной передачи данных и задает контакты вывода на выводы светодиода:

void setup() {

// Инициализируем последовательную передачу данных

Serial.begin(9600);

// Задаем кнтакты вывода

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

3.B основном цикле все зависит от наличия байтов для чтения:

void loop() {

// если будфер содержит последовательные данные, считываем байт

if (Serial.available() > 0) {

int inByte = Serial.read();

4. Если какие-либо входящие данные поступили, мы обрабатываем из них только несколько предопределенных значений. Tак, получив одно из требуемых нам значений, мы используем операторы if, чтобы задать номер контакта и значение яркости:

И, наконец, задаем на текущем выбранном контакте (светодиоде)требуемый уровень яркости с помощью метода analogWrite():

// реагируем только на значения r, g, b и цифры от 0 до 9

// игнорируем все другие значения

if (inByte == 'r') {

currentPin = redPin;

}

if (inByte == 'g') {

currentPin = greenPin;

}

if (inByte == 'b') {

currentPin = bluePin;

}

// сопоставляем значения входящего байта диапазону команды Analog Read()

if (inByte >= '0' && inByte <= '9') {

brightness = map(inByte, '0', '9', 0, 255);

// устанавливаем заданную яркость на выбранном пине

analogWrite (currentPin, brightness);

}

}

}

Ссылка на итоговый вариант:

<https://www.tinkercad.com/things/eky3HvVyOcZ-bodacious-esboo-migelo/editel?sharecode=QY3XKov6dqbsDQbyRWNax_G22jxgxDVNlomEUw-CotI>

**Усложняем задачу**

В предыдущем проекте мы управляли микроконтроллером с компьютера, используя для этого очень простой протокол. Теперь давайте добавим возможность «новогодней гирлянды».

1) Реализуйте функцию автоматического переключения светодиода в режиме:

1 секунда – горение красного цвета

1 секунда – горение синего цвета

1 секунда – горение зеленого цвета

2) Реализуйте функцию включения и остановки гирлянды по заданной команде

3) Добавьте в схему еще 2 rgb светодиода

4) Запустите гирлянду на полученной схеме из 3 диодов

\*\*\*

5) Создайте режим для гирлянды, в котором диоды будут переключаться не одновременно в одинаковом режиме, а последовательно переливаясь разными цветами.

Решение:

<https://www.tinkercad.com/things/09h8xmMbYla-spectacular-jarv/editel?sharecode=4lK05Fj02c-Yb4vXC5SimQhSW_iuKshoexShdUIgBmc>

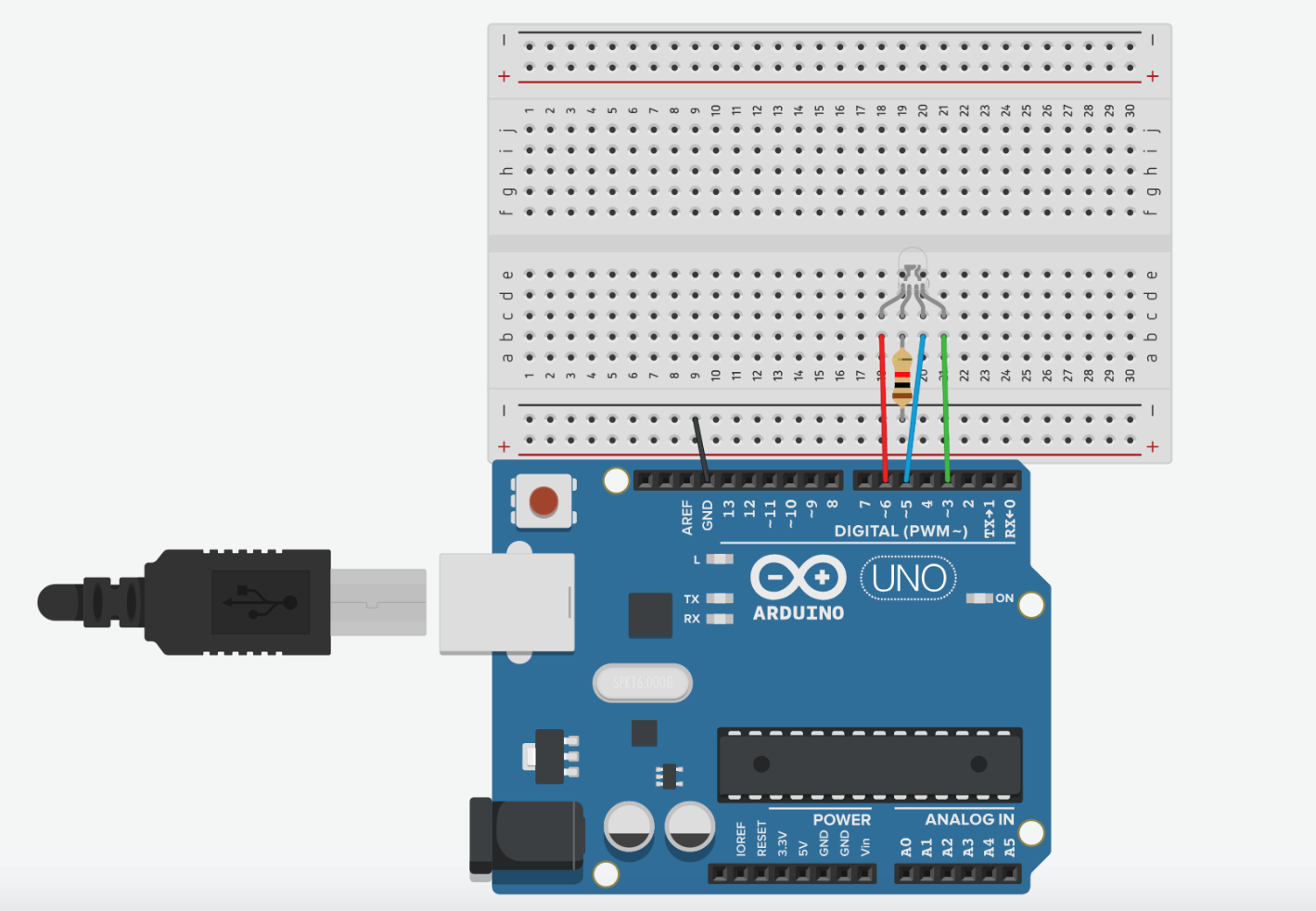
<https://www.tinkercad.com/things/4M4wA07ogwF-frantic-uusam-kup/editel?sharecode=qfPYaZkF3fUWCUkCMzgI_vAXuznH7N9bOoT7oPji7b0>

<https://www.tinkercad.com/things/g3k23alkqj5-epic-kasi/editel?sharecode=A0dhyahyIs6huyHS5gOThWktI56Go7lSEPwwvmnc1_0>

**Усложняем задачу Пункт 1:**

1. **Подключение платы**

Подключение анодов трехцветного светодиода к выводам ШИМ микроконтроллерной платы: для платы Arduino 101 (внизу) — 3, 5 и 6.



**2. Код**

1.Прежде всего, надо задать константы, в которых будут храниться номера контактов, к которым подключены аноды светодиода (эти значения устанавливаются в зависимости от типа платы). Также нужно задать переменные: для хранения номера контакта текущего светодиода и для хранения значения яркости:

// Управление трехцветным светодиодом

// Программа управляет трехцветным RGB светодиодом, чьи выходы присоединены к коннекторам:

// R (красный) - 6

// G (зеленый) - 3

// В (синий) - 5

// Вначале обозначаем список констант

const int greenPin = 3; // инициализируем пин под красный цвет

const int redPin = 6; // инициализируем пин под зеленый цвет

const int bluePin = 5; // инициализируем пин под синий цвет

int currentPin = 0; // текущий контакт для задания яркости

int brightness = 10; // контакт показывающий текущий уровень яркости

2.Далее метод setup() запускает процедуру последовательной передачи данных и задает контакты вывода на выводы светодиода:

void setup()

{

// Задаем контакты вывода

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

3. Основной цикл

Подаем HIGH или LOW значение на цифровой вход/выход (pin).

На вход/выход (pin) был установлен режим выход (OUTPUT) функцией pinMode(), поэтому для значение HIGH напряжение на соответствующем вход/выходе (pin) будет 5В (3.3В для 3.3V плат), и 0В(земля) для LOW.

void loop()

{

digitalWrite(redPin, HIGH); // включает светодиод

delay(1000); // ждет секунду

digitalWrite(redPin, LOW); // выключает светодиод

delay(1000); // ждет секунду

digitalWrite(greenPin, HIGH); // включает светодиод

delay(1000); // ждет секунду

digitalWrite(greenPin, LOW); // выключает светодиод

delay(1000); // ждет секунду

digitalWrite(bluePin, HIGH); // включает светодиод

delay(1000); // ждет секунду

digitalWrite(bluePin, LOW); // выключает светодиод

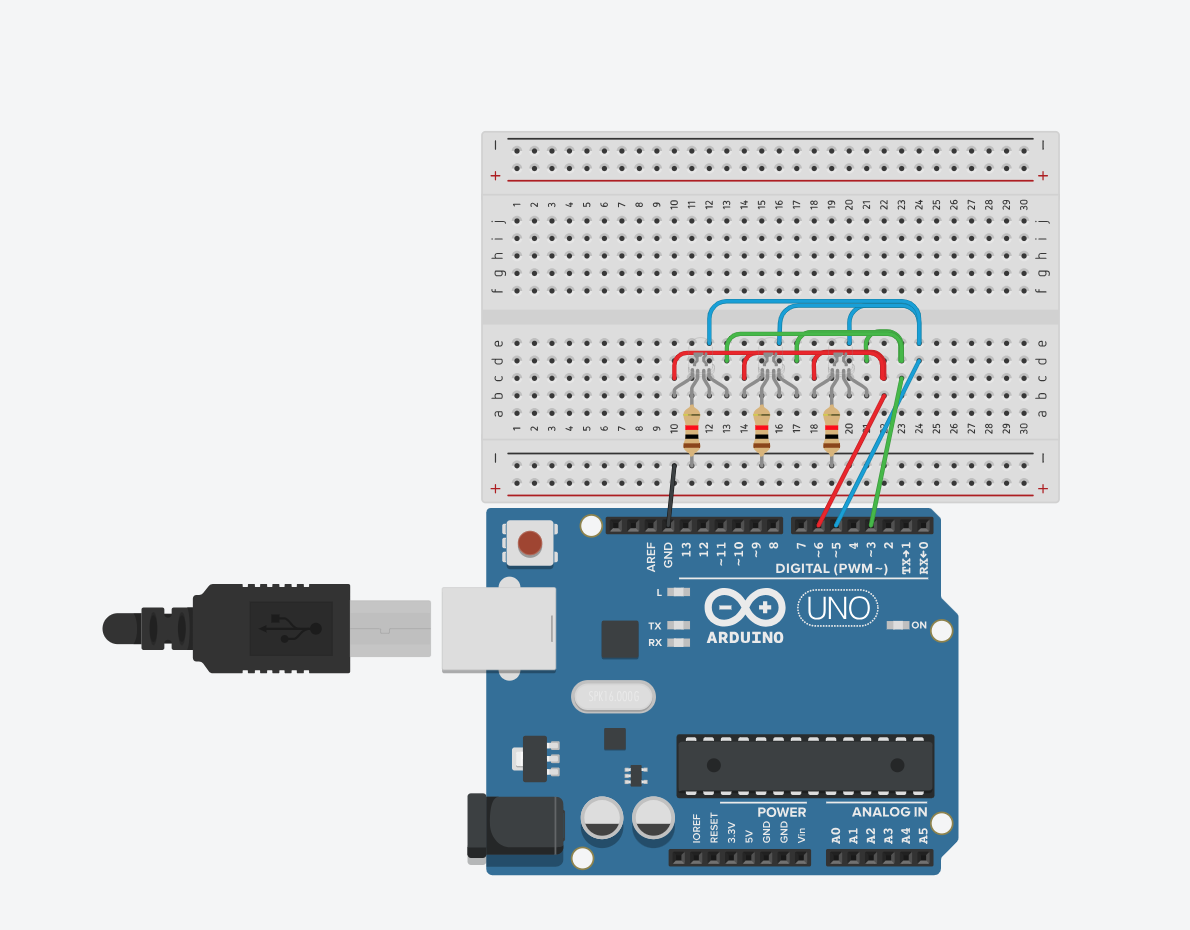
delay(1000); // ждет секунду

}

**Усложняем задачу Пункт 2-4:**

1. **Подключение платы**

Подключение анодов трех трехцветных светодиодов к выводам ШИМ микроконтроллерной платы: для платы Arduino 101 (внизу) — 3, 5 и 6.

****

**2. Код**

1. Задаем номера контактов, к которым подключены аноды светодиода

//Красный пин 6

#define R\_PIN 6

//ЗЕЛЕНЫЙ ПИН 3

#define G\_PIN 3

//синий пин 5

#define B\_PIN 5

2. Далее метод setup() запускает процедуру последовательной передачи данных и задает контакты вывода на выводы светодиода:

void setup( ) {

pinMode(R\_PIN, OUTPUT);

pinMode(G\_PIN, OUTPUT);

pinMode(B\_PIN, OUTPUT);

}

3. Основной цикл

void loop() {

digitalWrite(R\_PIN,1); // включает светодиод красным

delay(1000); // ждет секунду

digitalWrite(R\_PIN,0); // выключает светодиод

digitalWrite(G\_PIN,1); // включает светодиод зеленым

delay(1000); // ждет секунду

digitalWrite(G\_PIN,0); // выключает светодиод

digitalWrite(B\_PIN,1); // включает светодиод синим

delay(1000); // ждет секунду

digitalWrite(B\_PIN,0); // выключает светодиод

}

**Усложняем задачу Пункт 5:**

1. **Подключение платы**

Подключение анодов трех трехцветных светодиодов к выводам ШИМ микроконтроллерной платы: для платы Arduino 101 (внизу) — 3, 5 и 6.

**A circuit board with wires connected to it

Description automatically generated**

**2. Код**

//Красный пин 6

#define R\_PIN 6

//ЗЕЛЕНЫЙ ПИН 3

#define G\_PIN 3

//синий пин 5

#define B\_PIN 5

void setup( ) {

pinMode(R\_PIN, OUTPUT);

pinMode(G\_PIN, OUTPUT);

pinMode(B\_PIN, OUTPUT);

}

void loop() {

digitalWrite(R\_PIN,1);

delay(1000);

digitalWrite(R\_PIN,0);

digitalWrite(G\_PIN,1);

delay(1000);

digitalWrite(G\_PIN,0);

digitalWrite(B\_PIN,1);

delay(1000);

digitalWrite(B\_PIN,0);

}