МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №1

«Программирование микроконтроллера Ардуино»

по дисциплине

«Информационно-управляющие вычислительные системы»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Мартынов Д. С. \_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТЫ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухоруков В. А\_\_\_

(подпись) (фамилия, и., о)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мосташов В. С\_\_\_

(подпись) (фамилия, и., о)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Рябов Д. А\_\_\_\_

(подпись) (фамилия, и., о)

\_\_\_\_\_\_\_\_\_\_\_\_\_ Расторопов Д. С\_\_\_

(подпись) (фамилия, и., о)

\_\_\_\_\_\_19-В-2\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

Оглавление

[Ход работы 3](#_Toc103859290)

[Эксперимент 1. Маячок 3](#_Toc103859291)

[Эксперимент 2. Маячок с нарастающей яркостью 4](#_Toc103859292)

[Эксперимент 3. Светильник с управляемой яркостью 5](#_Toc103859293)

[Эксперимент 4. Терменвокс 7](#_Toc103859294)

[Эксперимент 5. Ночной светильник 8](#_Toc103859295)

[Эксперимент 6. Пульсар 10](#_Toc103859296)

[Эксперимент 7. Бегущий огонёк 11](#_Toc103859297)

[Эксперимент 8. Мерзкое пианино 13](#_Toc103859298)

[Модель светофора тремя светодиодами 15](#_Toc103859299)

[Модель светофора с RGB светодиодом 16](#_Toc103859300)

[«Радуга» с RGB светодиодом 18](#_Toc103859301)

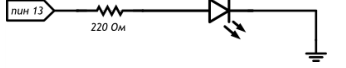
# Ход работы

## Эксперимент 1. Маячок

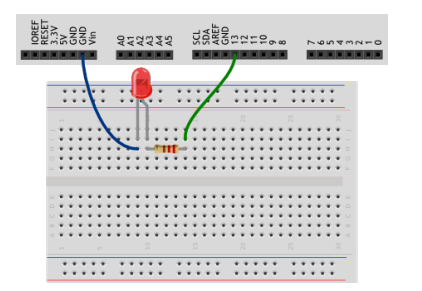
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 светодиод
* 1 резистор номиналом 220 Ом
* 2 провода «папа-папа»

### Принципиальная схема



### Схема на макете



### Код программы

void setup(){

// настраиваем пин №13 в режим выхода,

// т.е. в режим источника напряжения

pinMode(13, OUTPUT);

}

void loop(){

// подаём на пин 13 «высокий сигнал» (англ. «high»), т.е.

// выдаём 5 вольт. Через светодиод побежит ток.

// Это заставит его светиться

digitalWrite(13, HIGH);

// задерживаем (англ. «delay») микроконтроллер в этом

// состоянии на 100 миллисекунд

delay(100);

// подаём на пин 13 «низкий сигнал» (англ. «low»), т.е.

// выдаём 0 вольт или, точнее, приравниваем пин 13 к земле.

// В результате светодиод погаснет

digitalWrite(13, LOW);

// замираем в этом состоянии на 900 миллисекунд

delay(900);

// после «размораживания» loop сразу же начнёт исполняться

// вновь, и со стороны это будет выглядеть так, будто

// светодиод мигает раз в 100 мс + 900 мс = 1000 мс = 1 сек

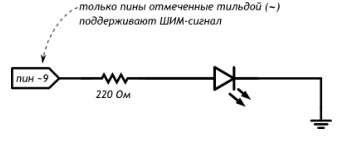
}

## Эксперимент 2. Маячок с нарастающей яркостью

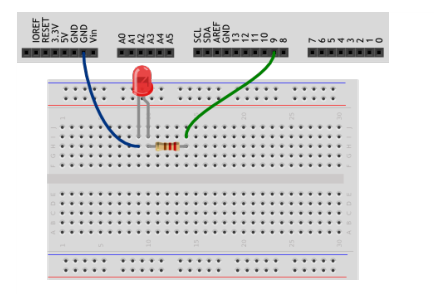
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 светодиод
* 1 резистор номиналом 220 Ом
* 2 провода «папа-папа»

### Принципиальная схема



### Схема на макете



### Код программы

#define LED\_PIN 9

void setup() {

// настраиваем пин со светодиодом в режим выхода

pinMode(LED\_PIN, OUTPUT);

}

void loop(){

// выдаём неполное напряжение на светодиод

// (он же ШИМ-сигнал, он же PWM-сигнал).

// Микроконтроллер переводит число от 0 до 255 к напряжению

// от 0 до 5 В. Например, 85 — это 1/3 от 255,

// т.е. 1/3 от 5 В, т.е. 1,66 В.

analogWrite(LED\_PIN, 85);

// держим такую яркость 250 миллисекунд

delay(250);

// выдаём 170, т.е. 2/3 от 255, или иными словами — 3,33 В.

// Больше напряжение — выше яркость!

analogWrite(LED\_PIN, 170);

delay(250);

// все 5 В — полный накал!

analogWrite(LED\_PIN, 255);

// ждём ещё немного перед тем, как начать всё заново

delay(250);

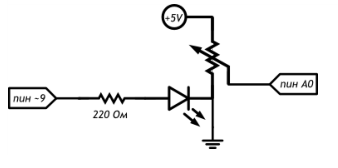
}

## Эксперимент 3. Светильник с управляемой яркостью

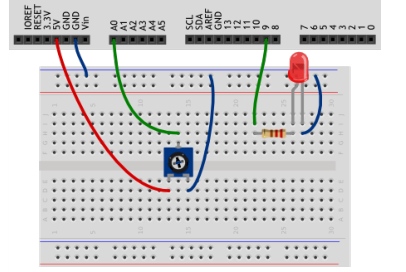
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 светодиод
* 1 резистор номиналом 220 Ом
* 6 проводов «папа-папа»
* 1 потенциометр

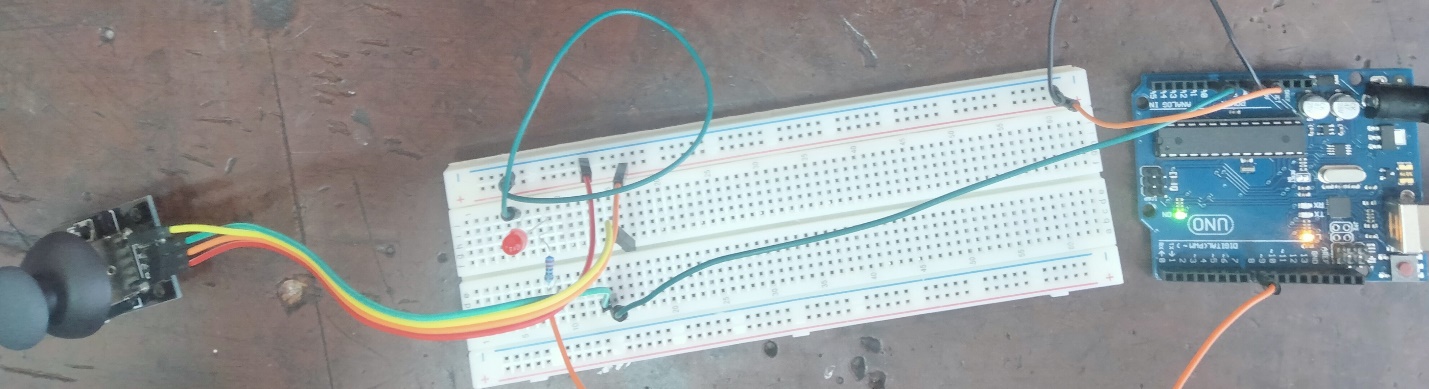
### Принципиальная схема

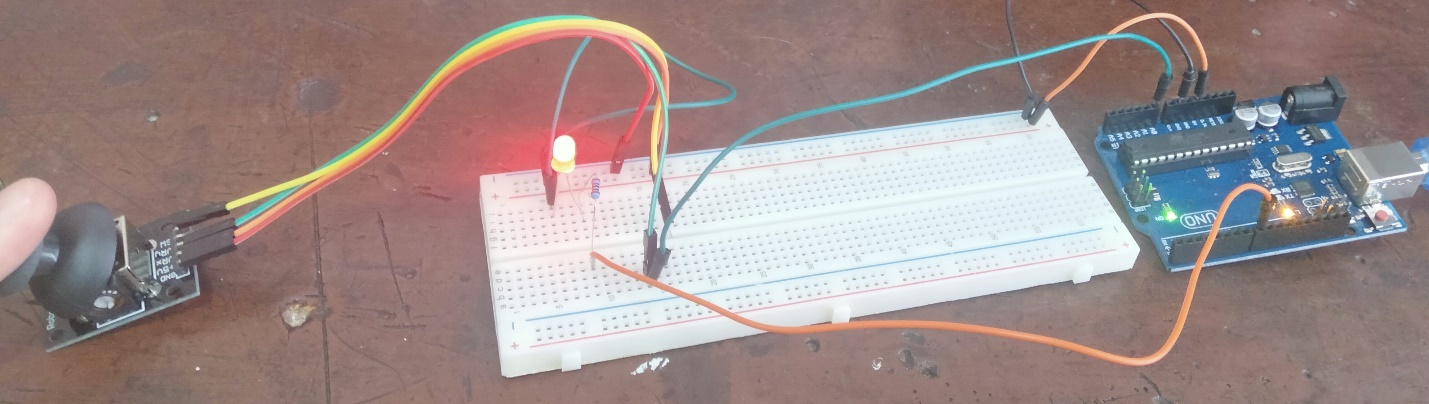


### Схема на макете



### Фото собранной схемы





### Код программы

#define LED\_PIN 9

#define POT\_PIN A0

void setup (){

pinMode (LED\_PIN, OUTPUT);

pinMode (POT\_PIN, INPUT);

}

Void loop(){

int rotation, brightness;

// считываем в rotation напряжение с потенциометра:

// микроконтроллер выдаст число от 0 до 1023

// пропорциональное углу поворота ручки

rotation = analogRead(POT\_PIN);

// в brightness записываем полученное ранее значение //rotation делённое на 4.

// В итоге мы получим целое число от 0 до 255

brightness = rotation / 4;

// выдаём результат на светодиод

analogWrite (LED\_PIN, brightness);

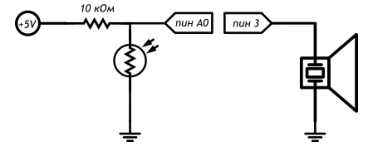
}

## Эксперимент 4. Терменвокс

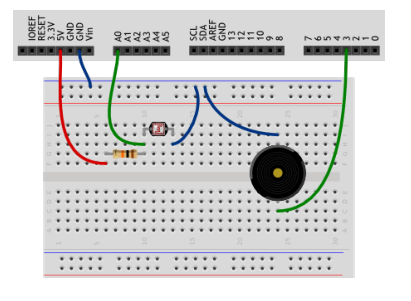
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 пьезопищалка
* 6 проводов «папа-папа»
* 1 резистор номиналом 10 кОм
* 1 фоторезистор

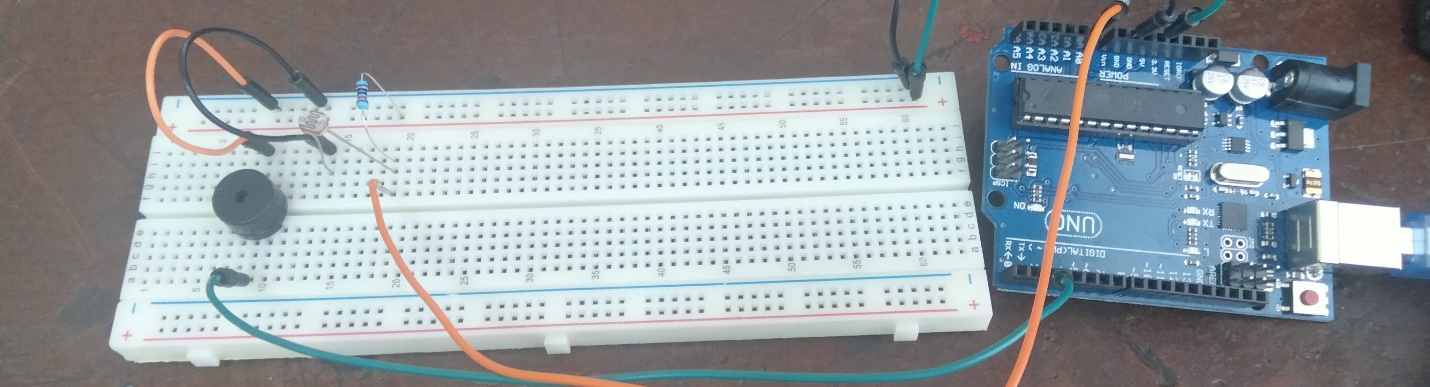
### Принципиальная схема



### Схема на макете



### Фото собранной схемы



### Код программы

#define BUZZER\_PIN 3

#define LDR\_PIN A0

void setup(){

// пин с пьезопищалкой — выход...

pinMode(BUZZER\_PIN, OUTPUT);

// ...а все остальные пины являются входами изначально,

// всякий раз при подаче питания или сбросе микроконтроллера.

// Поэтому, на самом деле, нам совершенно необязательно

// настраивать LDR\_PIN в режим входа: он и так им является

}

void loop(){

int val, frequency;

// считываем уровень освещённости так же, как для

// потенциометра: в виде значения от 0 до 1023.

val = analogRead(LDR\_PIN);

// рассчитываем частоту звучания пищалки в герцах (ноту),

// используя функцию проекции (англ. map). Она отображает

// значение из одного диапазона на другой, строя пропорцию.

// В нашем случае [0; 1023] -> [3500; 4500]. Так мы получим

// частоту от 3,5 до 4,5 кГц.

frequency = map(val, 0, 1023, 3500, 4500);

// заставляем пин с пищалкой «вибрировать», т.е. звучать

// (англ. tone) на заданной частоте 20 миллисекунд. При

// cледующих проходах loop, tone будет вызван снова и снова,

// и на деле мы услышим непрерывный звук тональностью, которая

// зависит от количества света, попадающего на фоторезистор

tone(BUZZER\_PIN, frequency, 20);

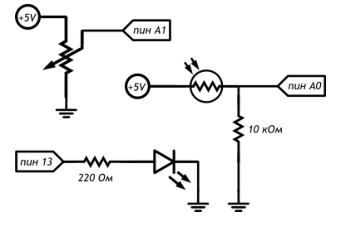
}

## Эксперимент 5. Ночной светильник

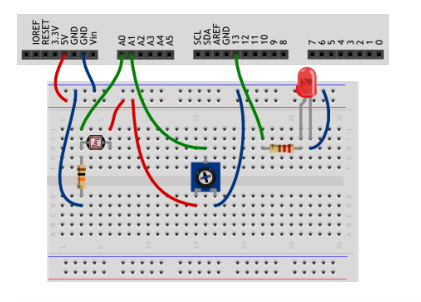
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 светодиод
* 1 фоторезистор
* 1 резистор номиналом 220 Ом
* 1 резистор номиналом 10 кОм
* 1 переменный резистор (потенциометр)
* 10 проводов «папа-папа»

### Принципиальная схема



### Схема на макете



### Код программы

#define LED\_PIN 13

#define LDR\_PIN A0

#define POT\_PIN A1

void setup(){

pinMode(LED\_PIN, OUTPUT);

}

void loop(){

// считываем уровень освещённости.

int lightness = analogRead(LDR\_PIN);

// считываем значение с потенциометра, которым мы регулируем

// пороговое значение между условными темнотой и светом

int threshold = analogRead(POT\_PIN);

// объявляем логическую переменную и назначаем ей значение

// «темно ли сейчас». Логические переменные, в отличие от

// целочисленных, могут содержать лишь одно из двух значений:

// истину (англ. true) или ложь (англ. false). Такие значения

// ещё называют булевыми (англ. boolean).

boolean tooDark = (lightness < threshold);

// используем ветвление программы: процессор исполнит один из

// двух блоков кода в зависимости от исполнения условия.

// Если (англ. «if») слишком темно...

if (tooDark) {

// ...включаем освещение

digitalWrite(LED\_PIN, HIGH);

} else {

// ...иначе свет не нужен — выключаем его

digitalWrite(LED\_PIN, LOW);

}

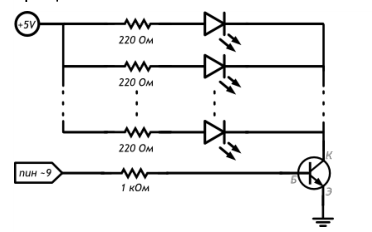
}

## Эксперимент 6. Пульсар

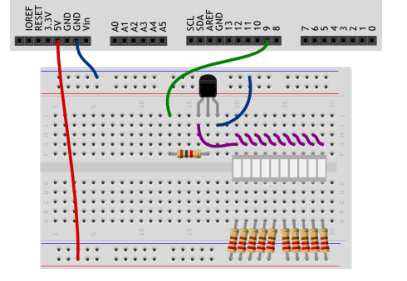
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 биполярный транзистор
* 1 светодиодная шкала
* 1 резистор номиналом 1 кОм
* 10 резисторов номиналом 220 Ом
* 13 проводов «папа-папа»

### Принципиальная схема



### Схема на макете



### Код программы

#define CONTROL\_PIN 9

int brightness = 0;

void setup(){

pinMode(CONTROL\_PIN, OUTPUT);

}

void loop(){

// увеличиваем значение яркости на единицу, чтобы нарастить

// яркость. Однако яркость не должна быть более 255, поэтому

// используем операцию остатка от деления, чтобы при

// достижении значения 255, следующим значением снова стал 0

// Y % X — это остаток от деления Y на X;

// плюс, минус, делить, умножить, скобки — как в алгебре.

brightness = (brightness + 1) % 256;

// подаём вычисленный ШИМ-сигнал яркости на пин с базой

// управляющего транзистора

analogWrite(CONTROL\_PIN, brightness);

// ждём 10 мс перед следующим наращиванием яркости. Таким

// образом, полный накал будет происходить в течение

// 256×10 = 2560 мс

delay(10);

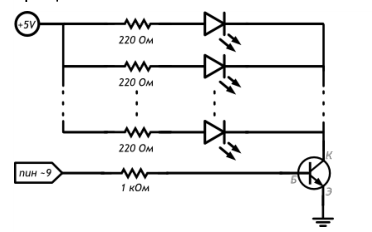
}

## Эксперимент 7. Бегущий огонёк

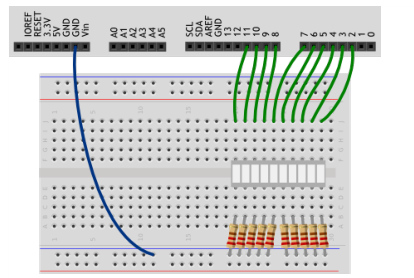
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 светодиодная шкала
* 10 резисторов номиналом 220 Ом
* 11 проводов «папа-папа»

### Принципиальная схема



### Схема на макете



### Код программы

// светодиодная шкала подключена к группе пинов расположенных

// подряд. Даём понятные имена первому и последнему пинам

#define FIRST\_LED\_PIN 2

#define LAST\_LED\_PIN 11

void setup(){

// в шкале 10 светодиодов. Мы бы могли написать pinMode 10

// раз: для каждого из пинов, но это бы раздуло код и

// сделало его изменение более проблематичным.

// Поэтому лучше воспользоваться циклом. Мы выполняем

// pinMode для (англ. for) каждого пина (переменная pin)

// от первого (= FIRST\_LED\_PIN) до последнего включительно

// (<= LAST\_LED\_PIN), всякий раз продвигаясь к следующему

// (++pin увеличивает значение pin на единицу)

// Так все пины от 2-го по 11-й друг за другом станут выходами

for (int pin = FIRST\_LED\_PIN; pin <= LAST\_LED\_PIN; ++pin)

pinMode(pin, OUTPUT);

}

void loop(){

// получаем время в миллисекундах, прошедшее с момента

// включения микроконтроллера

unsigned int ms = millis();

// нехитрой арифметикой вычисляем, какой светодиод

// должен гореть именно сейчас. Смена будет происходить

// каждые 120 миллисекунд. Y % X — это остаток от

// деления Y на X; плюс, минус, скобки — как в алгебре.

int pin = FIRST\_LED\_PIN + (ms / 120) % 10;

// включаем нужный светодиод на 10 миллисекунд, затем —

// выключаем. На следующем проходе цикла он снова включится,

// если гореть его черёд, и мы вообще не заметим отключения

digitalWrite(pin, HIGH);

delay(10);

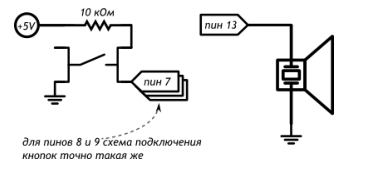
digitalWrite(pin, LOW);}

## Эксперимент 8. Мерзкое пианино

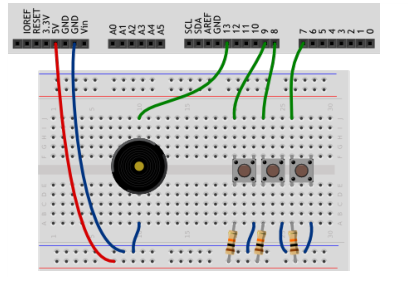
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 1 пьезопищалка
* 3 тактовых кнопки
* 3 резистора номиналом 10 кОм
* 10 проводов «папа-папа»

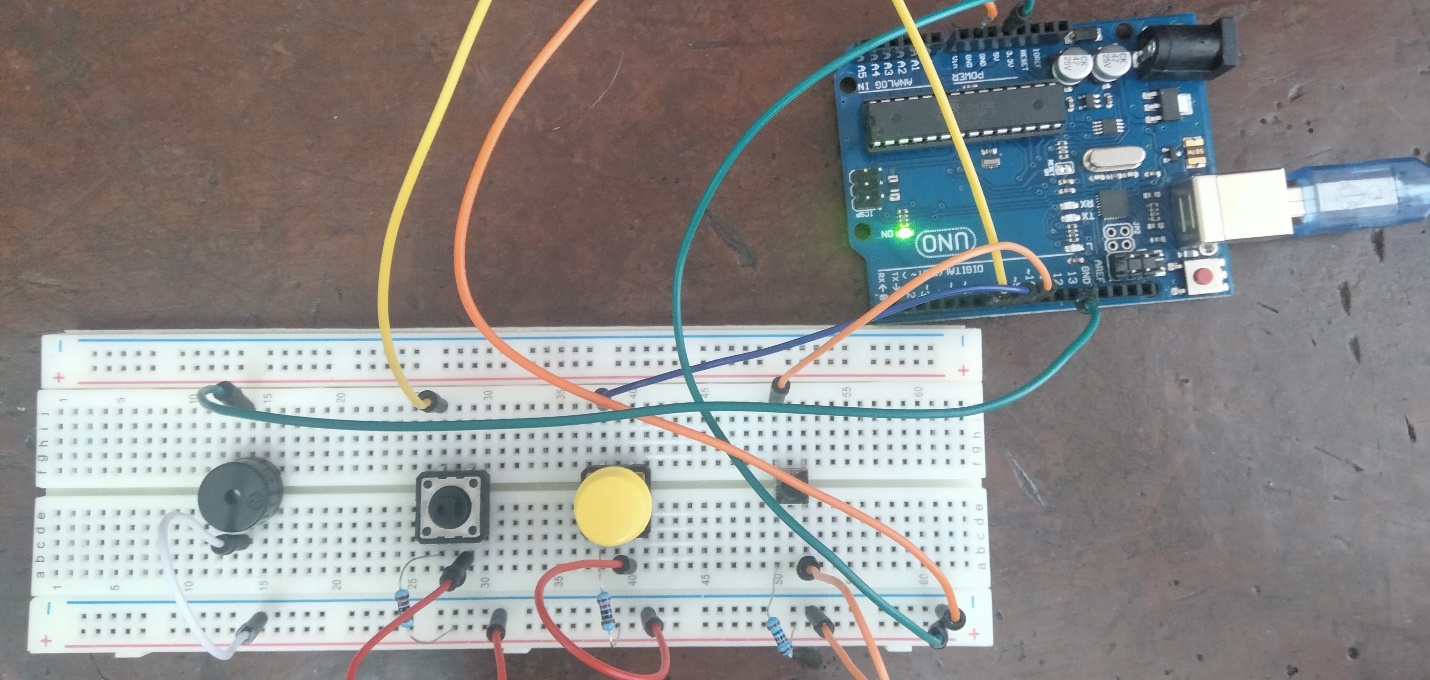
### Принципиальная схема



### Схема на макете



### Фото собранной схемы



### Код программы

#define BUZZER\_PIN 13 // пин с пищалкой (англ. «buzzer»)

#define FIRST\_KEY\_PIN 7 // первый пин с клавишей (англ. «key»)

#define KEY\_COUNT 3 // общее количество клавиш

void setup(){

pinMode(BUZZER\_PIN, OUTPUT);

}

void loop(){

// в цикле бежим по всем номерам кнопок от 0-го по 2-й

for (int i = 0; i < KEY\_COUNT; ++i) {

// на основе номера кнопки вычисляем номер её пина

int keyPin = i + FIRST\_KEY\_PIN;

// считываем значение с кнопки. Возможны всего 2 //варианта:

// \* высокий сигнал, 5 вольт, истина — кнопка отпущена

// \* низкий сигнал, земля, ложь — кнопка зажата

boolean keyUp = digitalRead(keyPin);

// проверяем условие «если не кнопка отпущена».

if (!keyUp) {

// рассчитываем высоту ноты в герцах в зависимости //от клавиши, которую рассматриваем на данном этапе //цикла. Мы получим значение 3500, 4000 или 4500

int frequency = 3500 + i \* 500;

// Заставляем пищалку пищать с нужной частотой в //течение 20 миллисекунд. Если клавиша останется зажатой, //пищалка вновь зазвучит при следующем проходе //loop, а мы услышим непрерывный звук

tone(BUZZER\_PIN, frequency, 20);

}

}

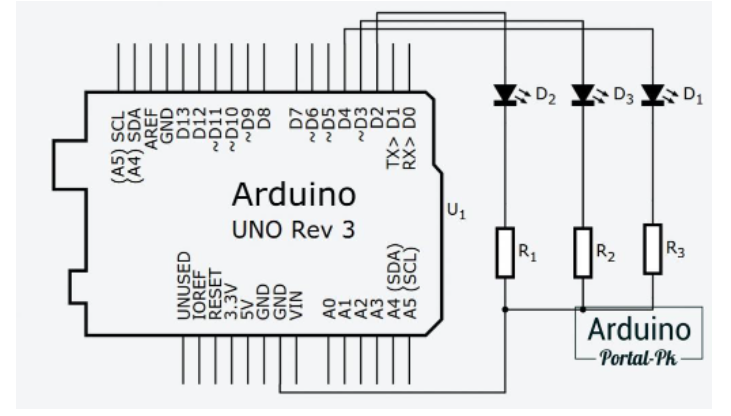
}

## Модель светофора тремя светодиодами

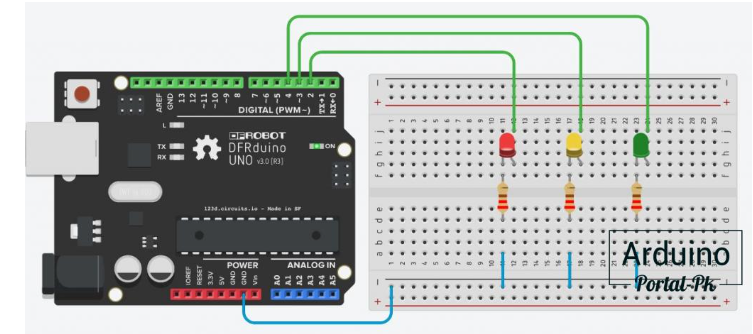
### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 3 резистора номиналом 220 Ом
* 7 проводов «папа-папа»
* 3 светодиода

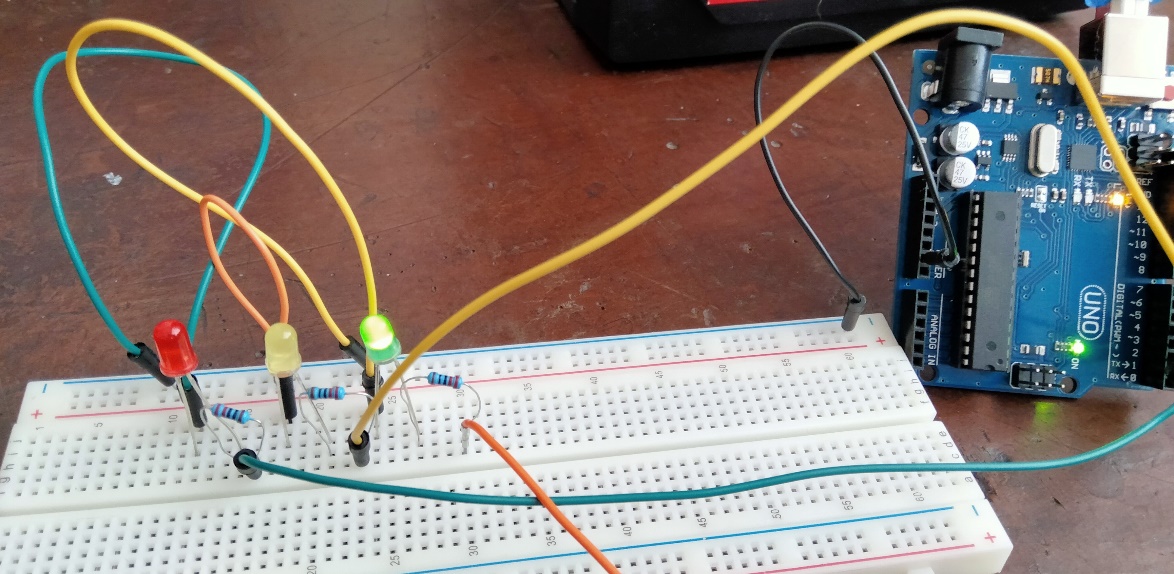
### Принципиальная схема



### Схема на макете



### Фото собранной схемы



### Код программы

int led\_red = 2;

int led\_yellow = 3;

int led\_green = 4;

void setup() {

pinMode(led\_red, OUTPUT);

pinMode(led\_yellow, OUTPUT);

pinMode(led\_green, OUTPUT);

}

void loop() {

digitalWrite(led\_red, HIGH);

delay(10000);

digitalWrite(led\_yellow, HIGH);

delay(2000);

digitalWrite (led\_green, HIGH);

digitalWrite (led\_red, LOW);

digitalWrite(led\_yellow, LOW);

delay(10000);

digitalWrite(led\_yellow, HIGH);

digitalWrite(led\_green, LOW);

delay(2000);

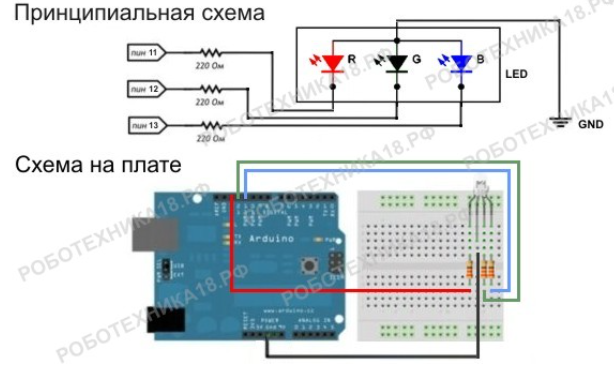
digitalWrite(led\_yellow, LOW);

}

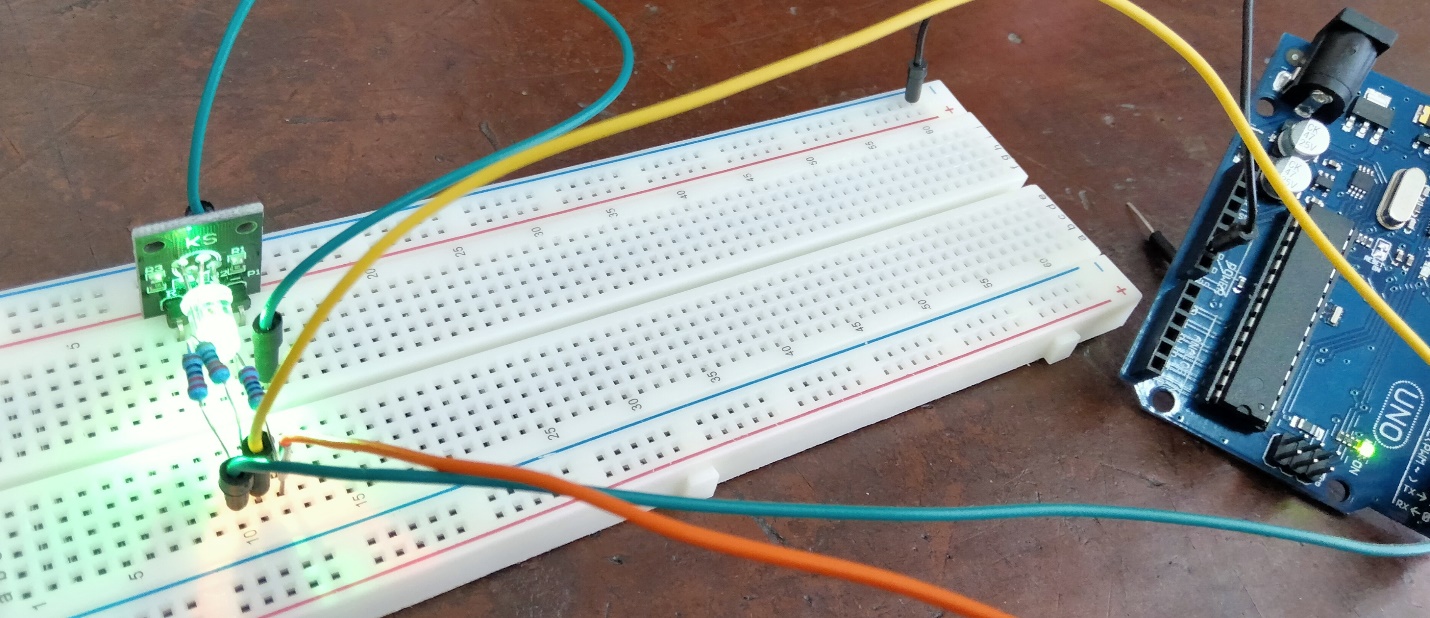
## Модель светофора с RGB светодиодом

### Список деталей для эксперимента

* 1 плата Arduino Uno
* 1 беспаечная макетная плата
* 3 резистора номиналом 220 Ом
* 5 проводов «папа-папа»
* 3 светодиода



### Фото собранной схемы



### Код программы

#define RED 11

#define GRN 12

#define BLU 13

void setup() {

pinMode(RED, OUTPUT);

pinMode(GRN, OUTPUT);

pinMode(BLU, OUTPUT);

}

void loop() {

// включение красного цвета

analogWrite(RED, 255);

delay(10000);

// включение желтого цвета

analogWrite (GRN, 255);

delay(10000);

// включение зеленого цвета

delay(10000);

analogWrite (RED, 0);

//Выключение зеленого цвета

analogWrite (GRN, 0);

}

## «Радуга» с RGB светодиодом

### Код программы

#define RED 11

#define GRN 12

#define BLU 13

void setup() {

pinMode(RED, OUTPUT);

pinMode(GRN, OUTPUT);

pinMode(BLU, OUTPUT);

}

void loop() {

// включение красного цвета

analogWrite(RED, 255);

delay(10000);

// включение оранжевого цвета

analogWrite(GRN, 165);

delay(10000);

// включение желтого цвета

analogWrite(GRN, 255);

delay(10000);

// включение зеленого цвета

analogWrite(RED, 0);

delay(10000);

// включение голубого цвета

analogWrite(BLU, 255);

delay(10000);

// включение синего цвета

analogWrite(GRN, 0);

delay(10000);

// включение фиолетового цвета

analogWrite(RED, 128);

analogWrite(BLU, 128);

delay(10000);

//Выключение фиолетового цвета

analogWrite (BLU, 0);

}