

Эксперимент № 3. RGB Светофор с увеличением яркости

Вам понадобится

1. Proteus 8.1 Professional
2. <http://tinkercad.com>

Список деталей для эксперимента

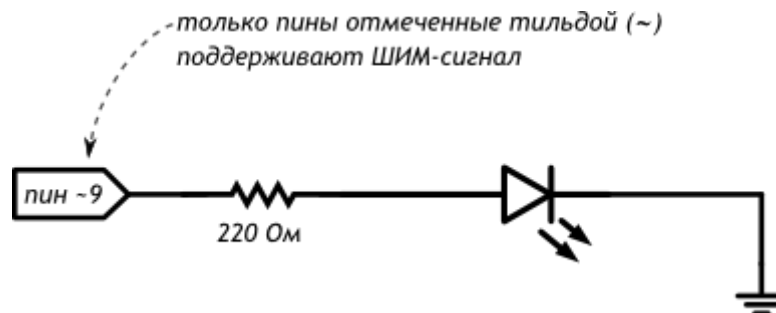
- 1 плата Arduino Uno
- 1 безопасная макетная плата
- 3 светодиода (LED RED, LED GREEN, LED BLUE) или 1 RGB светодиод
- 3 резистора номиналом 220 Ом
- провода «папа-папа»

Повторение

- 1) Понятие резистора, светодиода
- 2) Широко-импульсная модуляция
- 3) Функции digitalWrite, digitalRead, delay. Параметры функций.

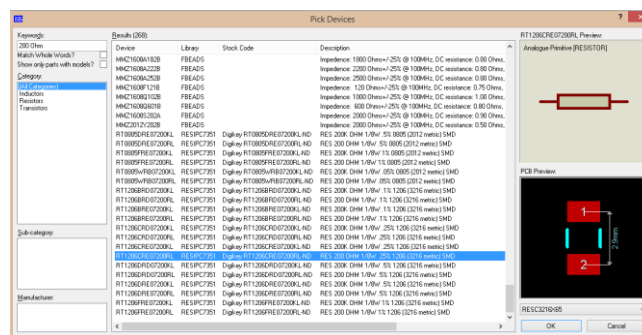
Пошаговое руководство для проведения эксперимента

Принципиальная схема

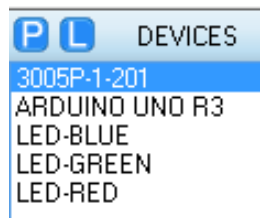


Разработка эксперимента в среде Proteus


Создайте новый эксперимент в среде Proteus. Подробное описание этапов создания проекта описано в прошлом занятии. Подтвердите создание нового проекта нажатием кнопки **Finish**. Сохраните проект, дайте имя файлу – **RGB**. Добавление и размещение нужных компонентов выполняется аналогично примерам прошлого занятия. Выберите все необходимые детали. В списке устройств добавьте по ключевому слову 7 устройств: **Arduino UNO R3, Animated LED-Red, Animated LED-Green, Animated LED-Blue** и резистор на **200 Ohm**. Для выбора устройства произведите двойной щелчок на выбранном устройстве в списке устройств.



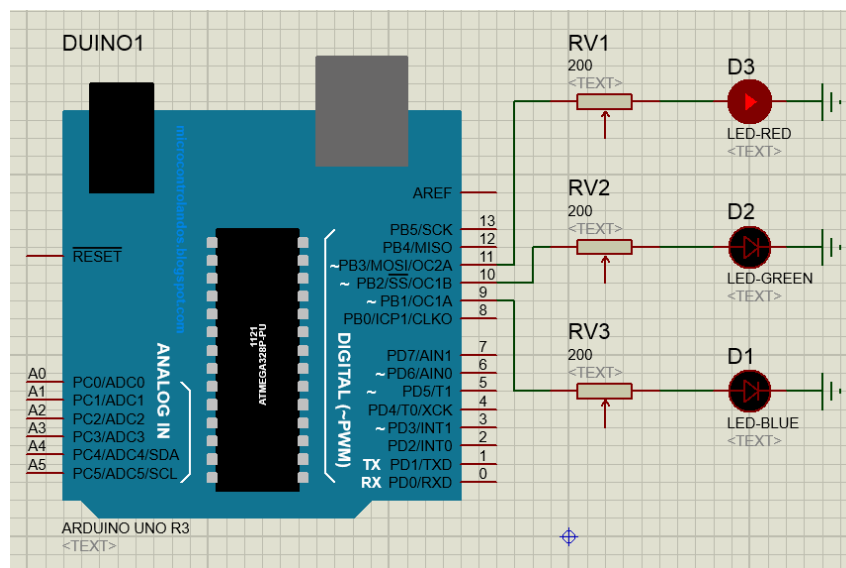
В результате правильно выполненных операций в списке устройств Devices должны отобразиться все выбранные вами детали для эксперимента.



Для размещения деталей на холсте выберите курсор **Selection Mode** на панели инструментов. Выберите курсором устройство **Arduino UNO R3**. Для размещения на холсте нажмите на холсте два раза левой кнопкой мыши. Для того чтобы соединить устройства проводами выберите точку соединения и проведите линию до точки соединения с другим устройством на холсте. **Внимание.** Соблюдайте полярность.

Осталось немного. Добавьте **Ground(-)** из списка устройств **Terminals Mode** . Добавить устройство так же легко как и все остальные. В результате правильно выполненных операций схема должна будет выглядеть следующим образом.

Обратите внимание. В этом эксперименте мы будем подключать светодиоды через резисторы номиналом на 200 Ом (200 Ohm) к цифровым портам с **широко-импульсной модуляцией**. Не любой порт Arduino поддерживает широтно-импульсную модуляцию, если вы хотите регулировать напряжение, вам подойдут пины, помеченные символом тильда «~». Для Arduino Uno это пины 3, 5, 6, 9, 10, 11



Программирование. Создание схемы завершено. Откроем **Arduino IDE**.

Приложение. Код программы «RGB – светофор» с нарастающей яркостью.

```
// даём разумное имя для пина №9 со светодиодом
// (англ. Light Emitting Diode или просто «LED»)
// Так нам не нужно постоянно вспоминать куда он подключён
#define LED_PIN11 11
#define LED_PIN10 10
#define LED_PIN9 9

void setup()
```

```

{
  // настраиваем пин со светодиодом в режим выхода,
  // как и раньше

  pinMode(LED_PIN11, OUTPUT);
  pinMode(LED_PIN10, OUTPUT);
  pinMode(LED_PIN9, 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_PIN11, 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_PIN11, 170);
  delay(250);

  // все 5 В — полный накал!
  analogWrite(LED_PIN11, 255);
  // ждём ещё немного перед тем, как начать всё заново
  delay(250);
  analogWrite(LED_PIN11, 0); // "выключаем"
  delay(1000);

  //повторяем тоже самое с остальными светодиодами подключенным к ШИМ
  // соответв. Это LED_PIN10 и LED_PIN9

  analogWrite(LED_PIN10, 85);
  delay(250);
  analogWrite(LED_PIN10, 170);
  delay(250);
  analogWrite(LED_PIN10, 255);
  delay(250);
  analogWrite(LED_PIN10, 0);
  delay(1000);
  analogWrite(LED_PIN9, 85);
  delay(250);
  analogWrite(LED_PIN9, 170);

```

```
delay(250);
analogWrite(LED_PIN9, 255);
delay(250);
analogWrite(LED_PIN9, 0);
delay(1000);
}
```

- Идентификаторы переменных (Прим.: LED_PIN) являются одним словом, и нельзя создать идентификатор в котором есть пробелы между словами, например такие: LED PIN. Лучше всего создавать осмысленные идентификаторы, так как их проще запомнить и использовать в коде.
- Идентификаторы могут состоять из латинских букв, цифр и символов подчеркивания `_`. При этом идентификатор не может начинаться с цифры.

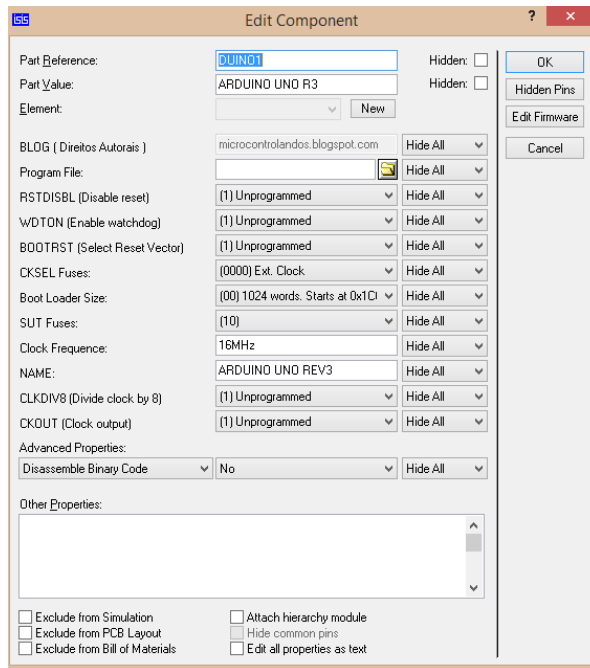
```
MTR_PIN9 // верно
_MY_PRJ1 // верно
3D_PRINT // ошибка
ПРОЕКТ_3Д // ошибка
```



- Регистр букв в идентификаторе имеет значение. Т.е. LED_PIN, LED_pin и led_pin с точки зрения компилятора — различные идентификаторы
- Директива `#define` просто говорит компилятору заменить все вхождения заданного идентификатора на значение, заданное после пробела (здесь 9, 10 и 11), эти директивы помещают в начало кода. В конце данной директивы точка с запятой ; не допустима
- В программах для пояснения кода лучше всего использовать комментарии (комментарии игнорируются компилятором, зато полезны людям для понимания чужого кода)

```
// однострочный комментарий следует после двойного слеша до конца строки
/* многострочный комментарий
помещается между парой слеш-звездочка и звездочка-слеш */
```

- Функция `analogWrite(pin, value)` не возвращает никакого значения и принимает два параметра:
 1. `pin` — номер порта, на который мы отправляем сигнал
 2. `value` — значение скважности ШИМ, которое мы отправляем на порт. Он может принимать целочисленное значение от 0 до 255, где 0 — это 0%, а 255 — это 100%

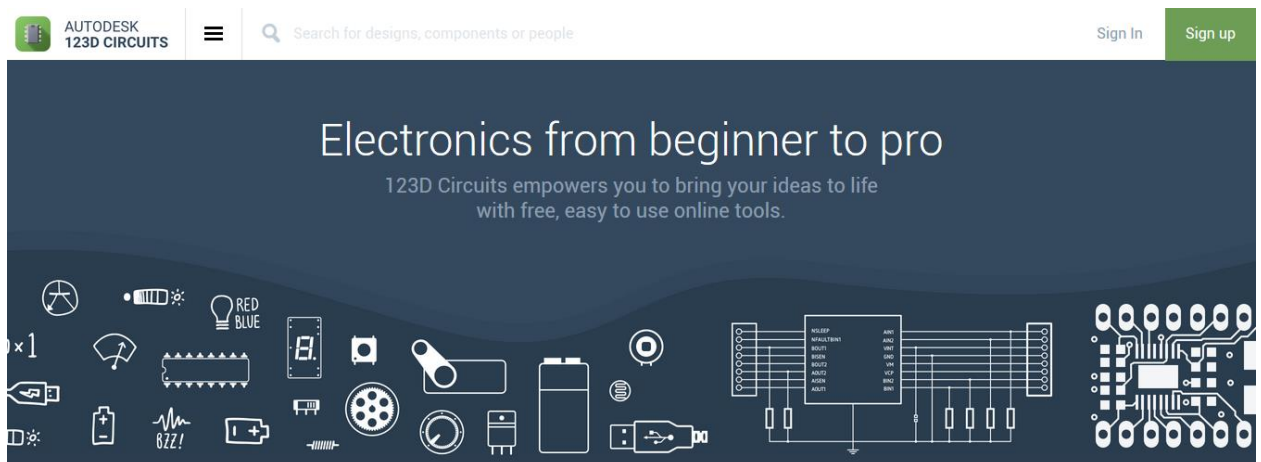
Proteus понимает только 16-ричный код, и для этого нам надо использовать **HEX-файлы!** Скопируйте этот файл из временной папки в свою папку с проектами. Вернитесь в среду Proteus. Щелкните дважды на Arduino UNO R3 в среде Proteus для открытия окна следующего вида.

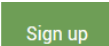


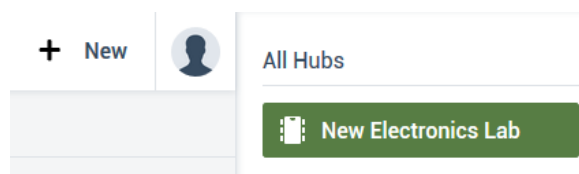
В окне нажмите на кнопке со значком  и выберите **HEX** файл из папки с проектом. Нажмите кнопку **OK**. Внизу экрана, в левой нижней части нажмите на кнопке **Run the simulation** .

Эксперимент с **Proteus** завершен!

Приступим к эксперименту с использованием **HTTP://123D.CIRCUIT.IO**

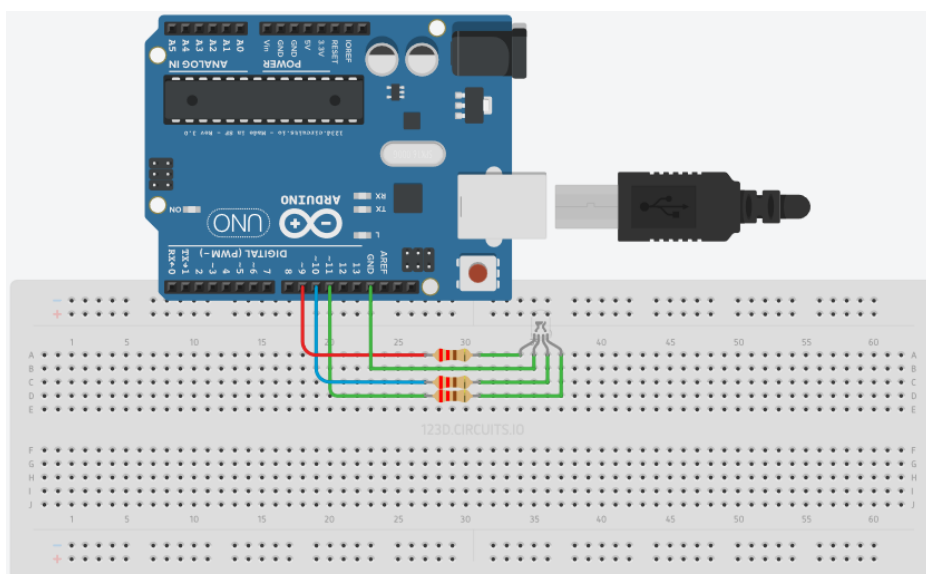


Для продолжения работы нажмите кнопку  и пройдите авторизацию используя учетную запись **Facebook**. После успешного прохождения авторизации, нажмите в верхней правой части сайта кнопку **NEW** и выберите **New Electronics Lab** для создания нового эксперимента.



Задание. Разместите на макетной плате компоненты как показано на рисунке ниже. Для соединения деталей между собой воспользуйтесь нажатиями левой кнопкой мыши. Нажмите на кнопке **Code Editor**. Скопируйте код проекта «RGB – светофор» с нарастающей яркостью» в Code

Editor. Нажмите на кнопке **Run Simulation**. Продолжите эксперименты с «RGB» проверив себя и выполнив **задания для самостоятельного решения**.



Проверь себя

- Что произойдет, если создать директиву `#define HIGH LOW`?
- Почему мы не сможем регулировать яркость светодиода, подключенного к порту 7?
- Какое усреднённое напряжение мы получим на пине 6, если вызовем функцию `analogWrite(6, 153)`?
- Какое значение параметра `value` нужно передать функции `analogWrite`, чтобы получить усреднённое напряжение 2 В?

Задания для самостоятельного решения

1. Измените код программы так, чтобы в течение секунды на светодиод последовательно подавалось усреднённое напряжение 0, 1, 2, 3, 4, 5 В.
2. Возьмите еще один светодиод, резистор на 220 Ом и соберите аналогичную схему на этой же макетке, подключив светодиод к пину номер 3 и другому входу GND, измените программу так, чтобы светодиоды мигали в противофазу: первый выключен, второй горит максимально ярко и до противоположного состояния.