

Эксперимент № 1. Маячок. Мигание светодиодом.

Вам понадобится

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

Список деталей для эксперимента

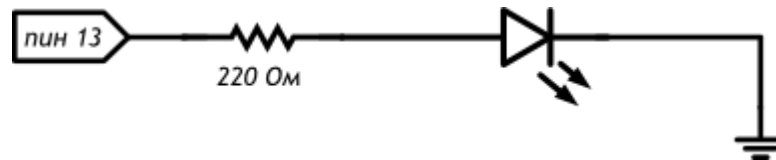
- 1 плата Arduino Uno
- 1 безопасная макетная плата
- 1 светодиод
- 1 резистор номиналом 220 Ом
- провода «папа-мама»

Повторение

- 1) Понятие электричества
- 2) Основные законы электричества
- 3) Понятие резистора, светодиода
- 4) Начало работы с Arduino
- 5) Основные разделы языка программирования C++.
- 6) Порты ввода/вывода на плате Arduino
- 7) Какие параметры может принимать функция DigitalWrite?
- 8) Что такое функция delay? Какие параметры принимает данная функция?

Пошаговое руководство для проведения эксперимента

Принципиальная схема



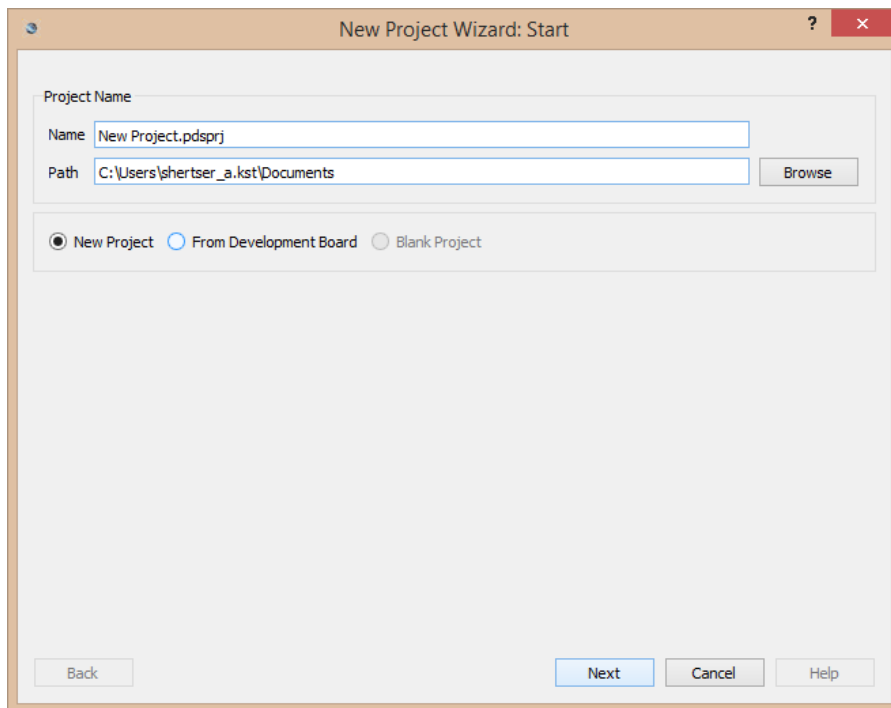
Разработка эксперимента в среде Proteus

Создадим новый эксперимент в среде Proteus

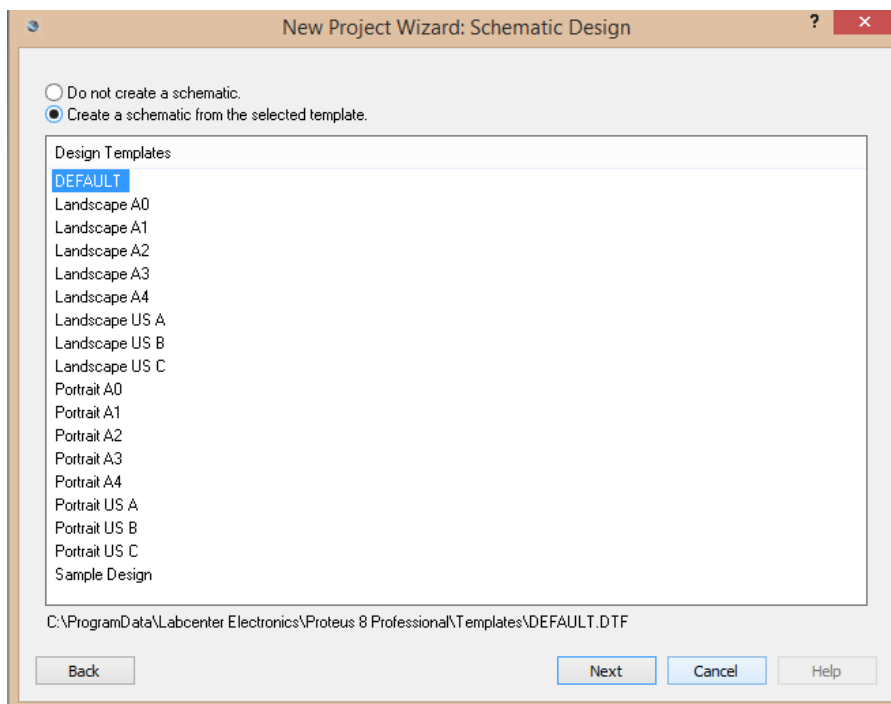


В разделе **Start** выберите **New project**

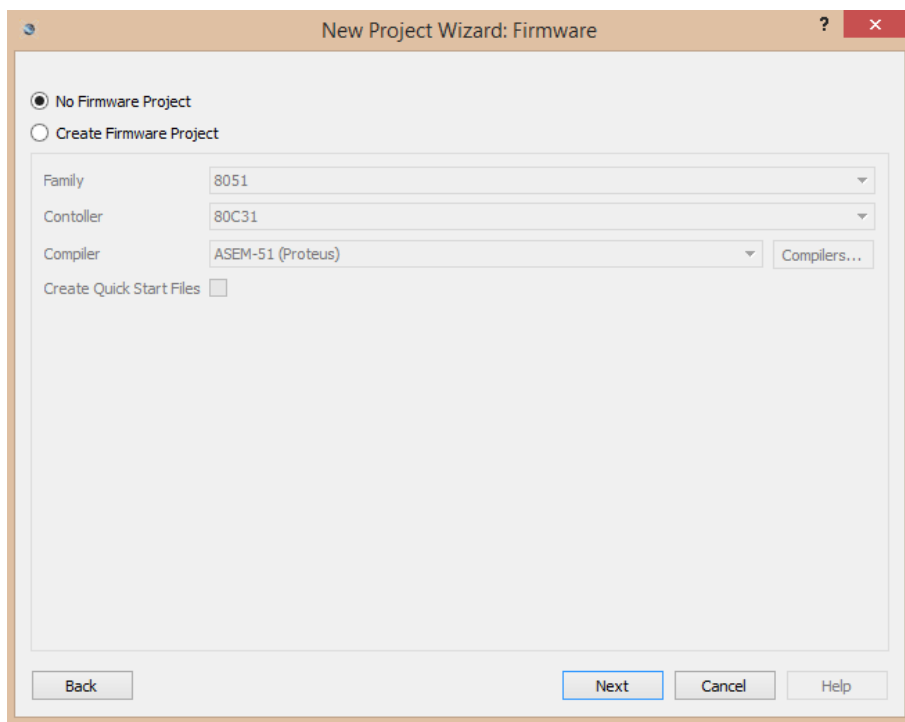
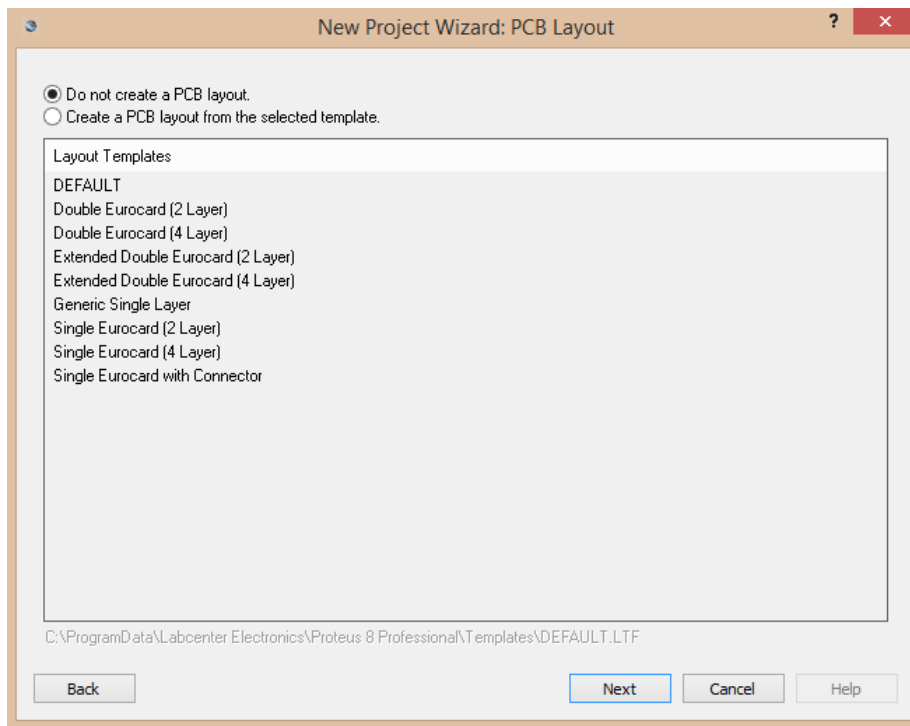
В открывшемся окне нажмите **Next**



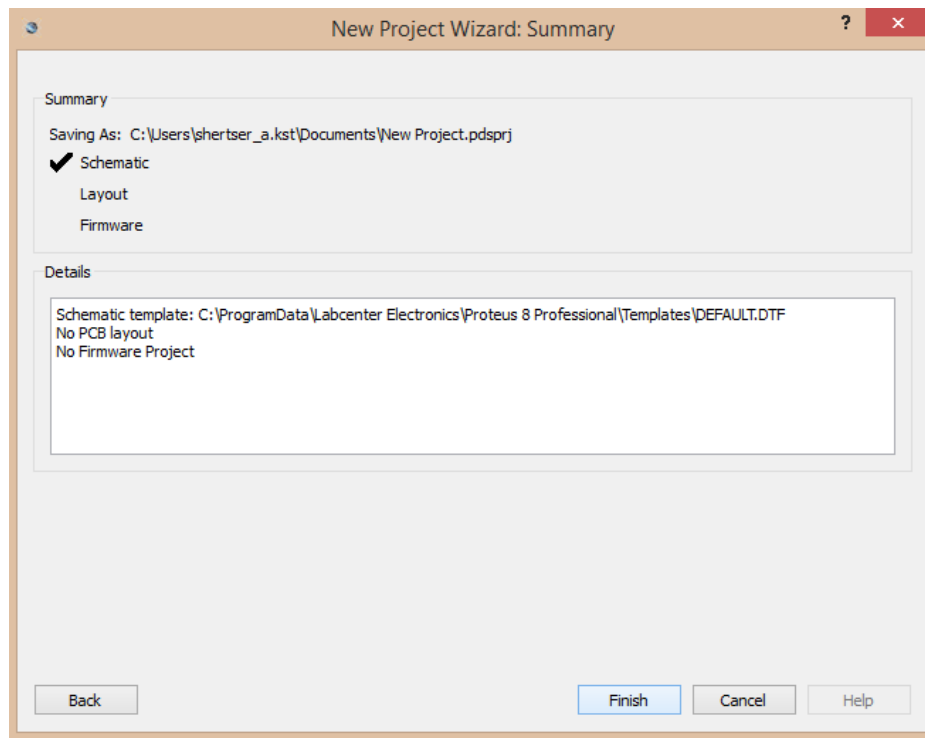
В следующем окне выберите **Create a schematic from the selected template** и выберите настройки окна по умолчанию нажав на строке **Default** в списке доступных шаблонов и нажмите **Next**



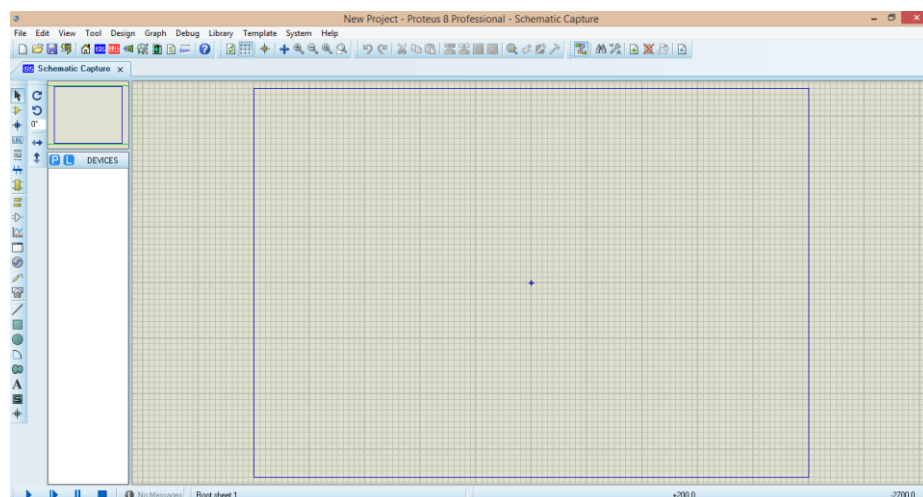
Нажмите кнопку **Next** в следующих окнах не меняя настройки



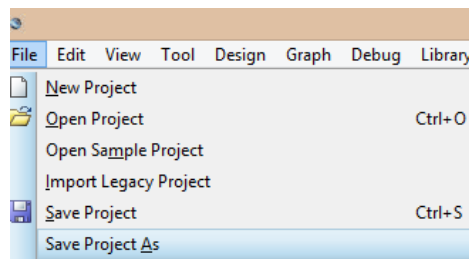
Подтвердите создание нового проекта нажатием кнопки **Finish**



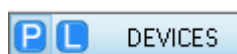
Будет создан проект в котором нам скоро предстоит смоделировать наш эксперимент.



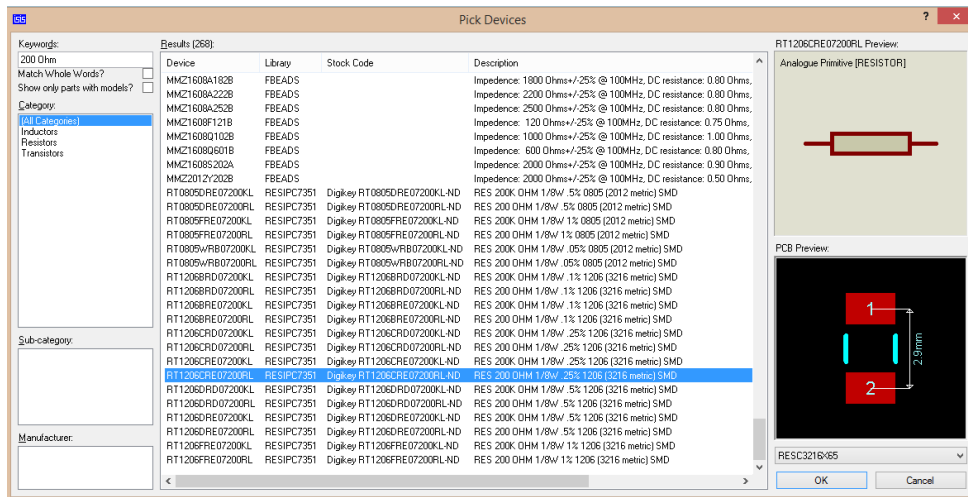
Сохраните ваш проект **в свою папку**



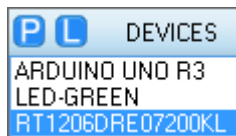
Начнем размещать нужные компоненты. Найдите на экране кнопку Pick Devices



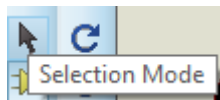
Нажмите на ней. В списке устройств найдите по ключевому слову 3 устройства: **Arduino UNO R3**, **Animated LED-Green** и резистор на **200 Ohm**. Для выбора устройства произведите двойной щелчок на выбранном устройстве в списке устройств.



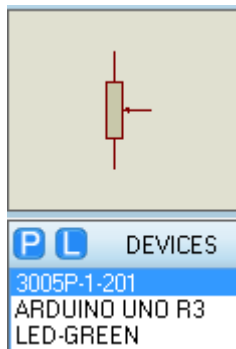
В результате правильно выполненных операций в списке устройств Devices должны отобразиться все выбранные вами детали для эксперимента.



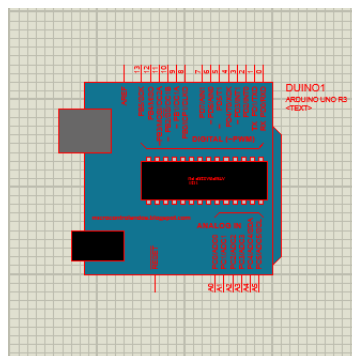
Для размещения деталей на холсте выберите курсор **Selection Mode** на панели инструментов




Выберите курсором устройство **Arduino UNO R3**.

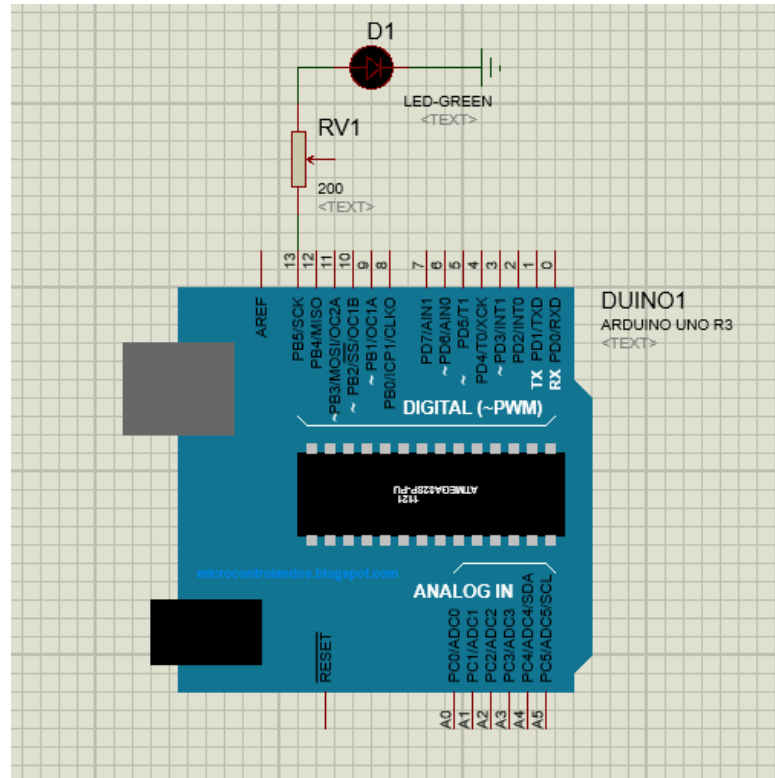


Для размещения на холсте нажмите на холсте два раза левой кнопкой мыши. Для того чтобы соединить устройства проводами выберите точку соединения и проведите линию до точки соединения с другим устройством на холсте. **Внимание.** Соблюдайте полярность

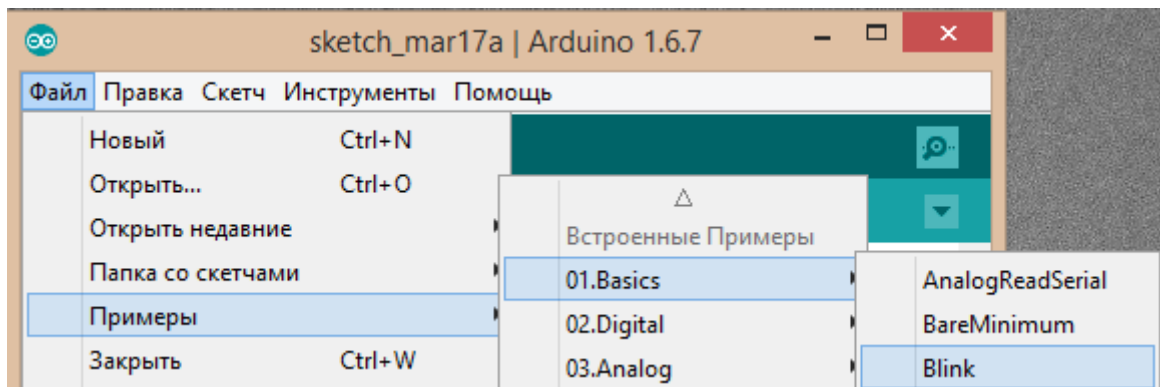


*Для размещения остальных устройств воспользуйтесь аналогичными действиями



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.




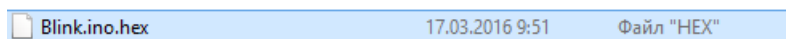
Программирование. Создание схемы завершено. Откроем **Arduino IDE** и в списке примеров откройте пример **Blink**.



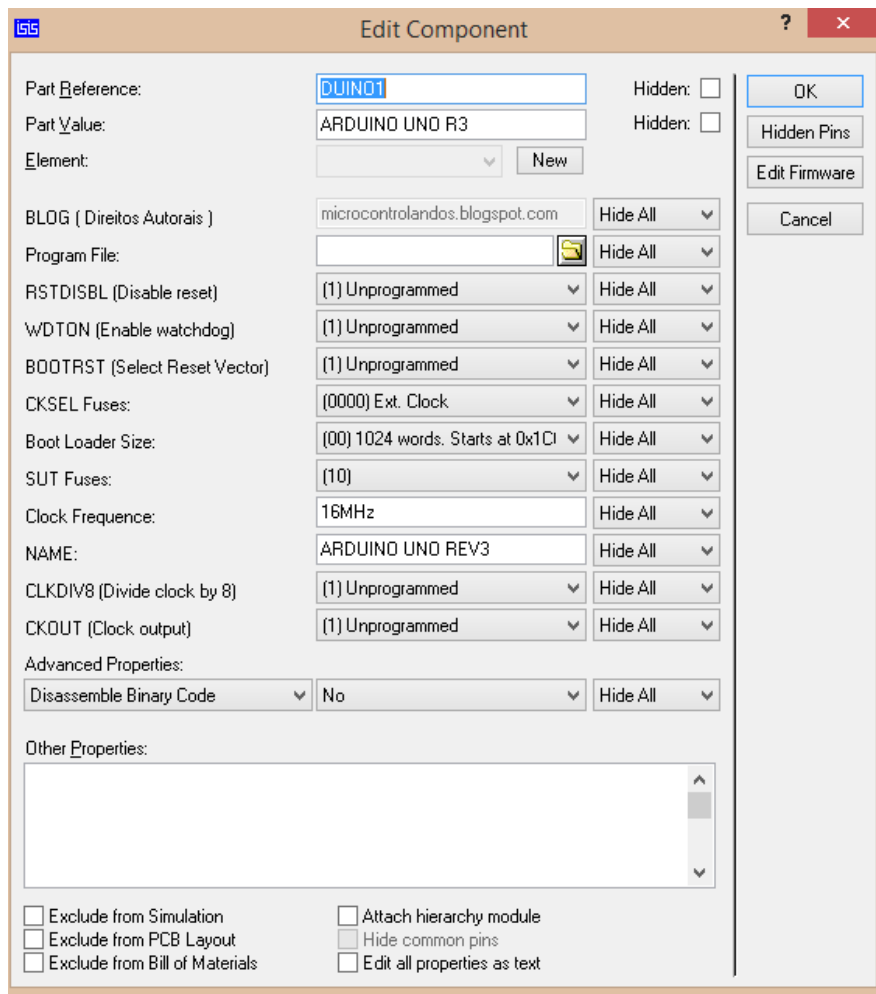
Proteus понимает только 16-ричный код, и для этого нам надо использовать **HEX-файлы!**



Как это сделать? Откройте **Файл -> Настройки** и поставьте галочку **Компиляция** . Затем  для компиляции готового скетча. Обратите внимание на путь вывода в строке отладки **«Используем предварительно скомпилированный файл»** **Используем предварительно скомпилированный файл: C:\Users\Si**.

Не закрывая Arduino IDE пройдите в эту папку (*у меня эта папка называется C:\Users\shertser_a.kst\AppData\Local\Temp) и откройте в этой папке имя папки начинающееся со слова **Build**  build9c9ef3bdfef2fccb480bc6e4bac749e4... 17.03.. **В папке найдите файл с расширением HEX**



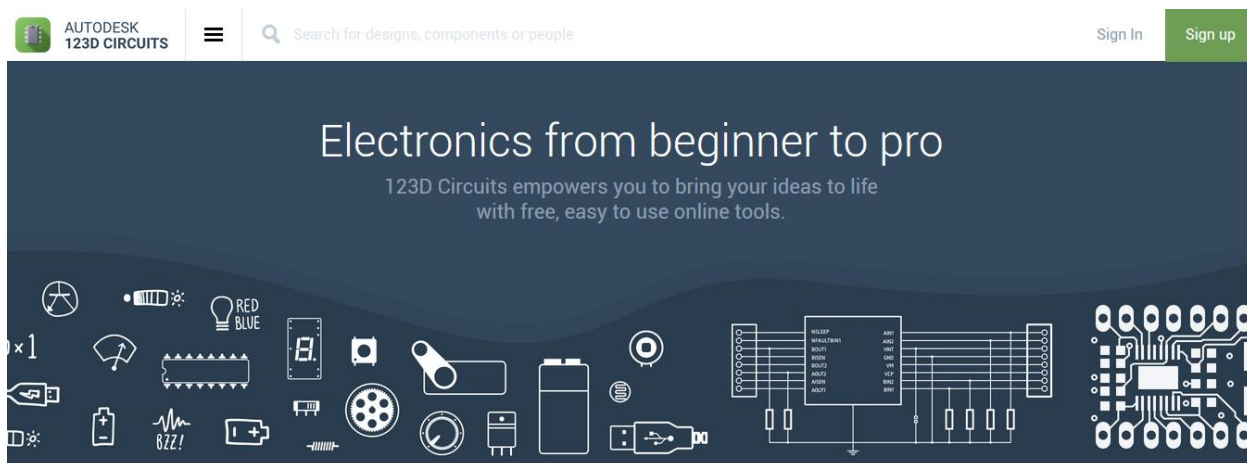
. Скопируйте этот файл в свою папку с проектами. Вернитесь в среду Proteus. Щелкните дважды на Arduino UNO R3 для открытия окна следующего вида.



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!

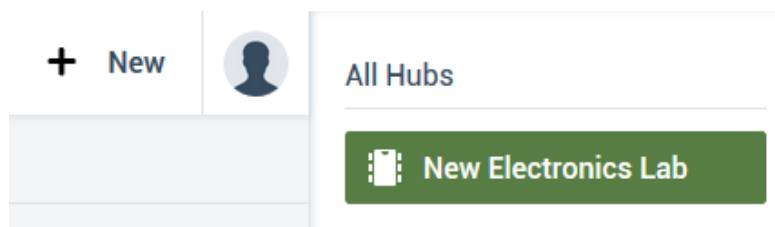
Приступим к эксперименту с использованием [HTTP://123D.CIRCUIT.IO](http://123D.CIRCUIT.IO)








Для продолжения работы нажмите кнопку  и пройдите авторизацию используя

Вход с помощью Facebook

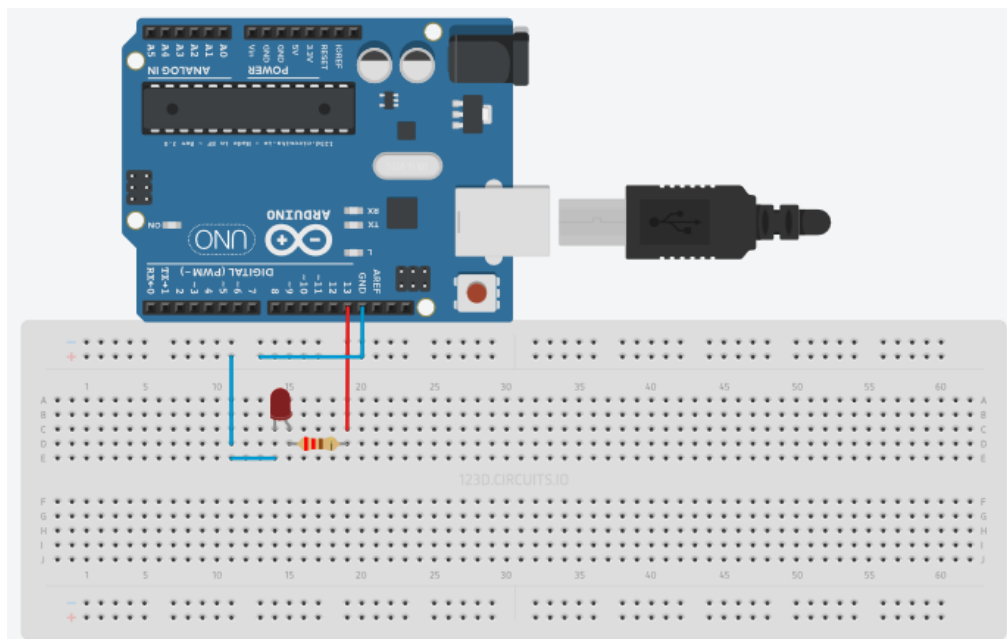
После успешного прохождения авторизации, нажмите в верхней правой части сайта кнопку **NEW** и выберите **New Electronics Lab** для создания нового эксперимента.



Панель инструментов Виртуальной лаборатории электроники состоит из

-  Кнопки вращения деталью для эксперимента
-  Кнопки удаления выбранной детали для эксперимента
-  **Code Editor** Редактора кода на языке Arduino который выполняется в фреймворке
-  **Components** Набора деталей и компонентов для эксперимента
-  **Start Simulation** Кнопки старта симуляции

Задание. Разместите на макетной плате компоненты как показано на рисунке ниже. Для соединения деталей между собой воспользуйтесь нажатиями левой кнопки мыши.



Проверьте код. Нажмите на кнопке **Code Editor**. По умолчанию в редакторе кода написан скетч для эксперимента «Маячок. Мигание светодиодом». Нажмите на кнопке **Run Simulation**. Продолжите эксперименты с «Маячком» проверив себя и выполнив **задания для самостоятельного решения**.

Приложение. Код программы «Маячок»

```
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
}
```

Пояснение к коду

- Процедура `setup` выполняется один раз при запуске микроконтроллера. Обычно она используется для конфигурации портов микроконтроллера и других настроек
- После выполнения `setup` запускается процедура `loop`, которая выполняется в бесконечном цикле. Именно этим мы пользуемся в данном примере, чтобы маячок мигал постоянно
- Процедуры `setup` и `loop` должны присутствовать в любой программе (скетче), даже если вам не нужно ничего выполнять в них — пусть они будут пустые, просто не пишите ничего между фигурными скобками. Например:

```
void setup()
{
}
```

- Запомните, что каждой открывающей фигурной скобке `{` всегда соответствует закрывающая `}`. Они обозначают границы некоего логически завершённого фрагмента кода. Следите за вложенностью фигурных скобок. Для этого удобно после каждой открывающей скобки увеличивать отступ на каждой новой строке на один символ табуляции (клавиша `Tab`)

- Обращайте внимание на ; в концах строк. Не стирайте их там, где они есть, и не добавляйте лишних. Вскоре вы будете понимать, где они нужны, а где нет.
- Функция `digitalWrite(pin, value)` не возвращает никакого значения и принимает два параметра:
 1. `pin` — номер цифрового порта, на который мы отправляем сигнал
 2. `value` — значение, которое мы отправляем на порт. Для цифровых портов значением может быть `HIGH` (высокое, единица) или `LOW` (низкое, ноль)
- Если в качестве второго параметра вы передадите функции `digitalWrite` значение, отличное от `HIGH`, `LOW`, `1` или `0`, компилятор может не выдать ошибку, но считать, что передано `HIGH`. Будьте внимательны
- Обратите внимание, что использованные нами константы: `INPUT`, `OUTPUT`, `LOW`, `HIGH`, пишутся заглавными буквами, иначе компилятор их не распознает и выдаст ошибку. Когда ключевое слово распознано, оно подсвечивается синим цветом в Arduino IDE

Проверь себя

1. Что будет, если подключить к земле анод светодиода вместо катода?
2. Что будет, если подключить светодиод с резистором большого номинала (например, 10 кОм)?
3. Что будет, если подключить светодиод без резистора?
4. Зачем нужна встроенная функция `pinMode`? Какие параметры она принимает?
5. Зачем нужна встроенная функция `digitalWrite`? Какие параметры она принимает?
6. С помощью какой встроенной функции можно заставить микроконтроллер ничего не делать?
7. В каких единицах задается длительность паузы для этой функции?

Задания для самостоятельного решения

1. Сделайте так, чтобы маячок светился полсекунды, а пауза между вспышками была равна одной секунде
2. Измените код примера так, чтобы маячок включался на три секунды после запуска устройства, а затем мигал в стандартном режиме