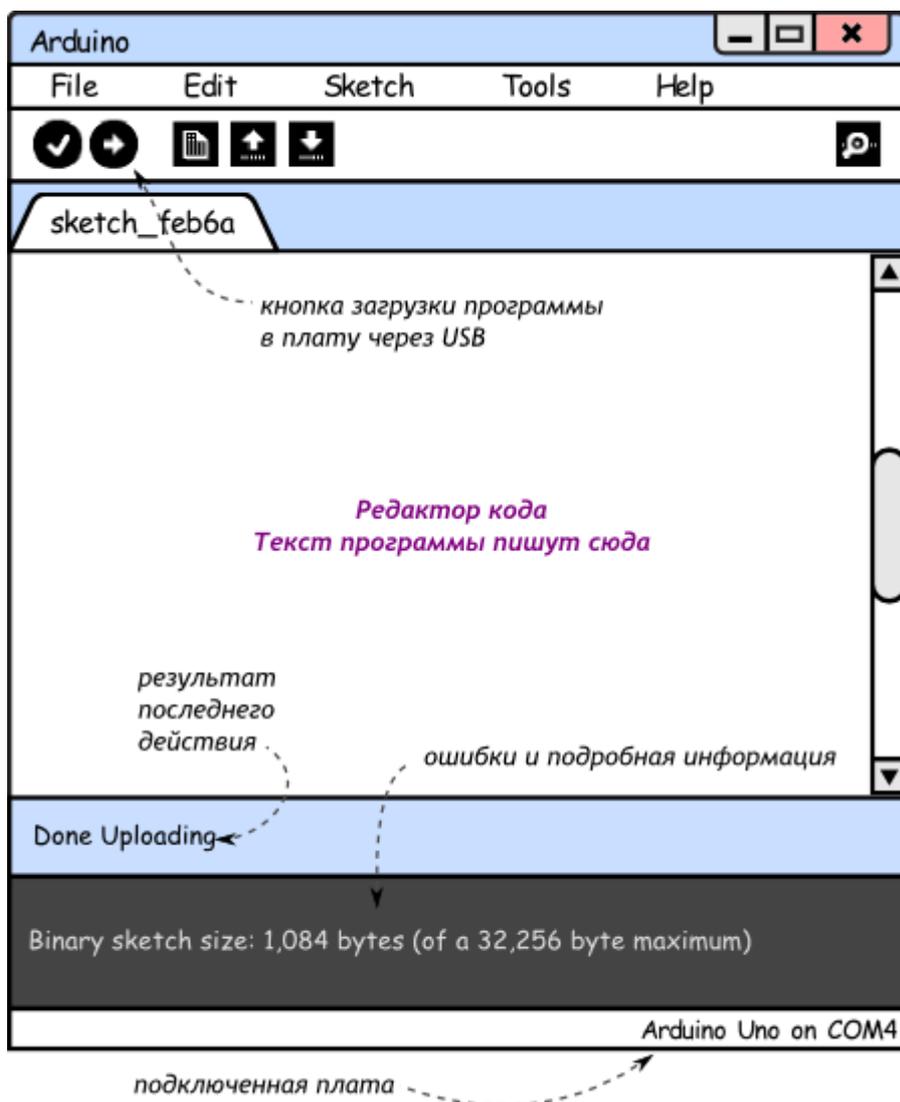


# Начало работы с Arduino

1. Скачайте Arduino IDE с [официального сайта](http://arduino.cc/en/Main/Software) (<http://arduino.cc/en/Main/Software>)
2. Если у вас Windows и Arduino IDE из zip-файла, установите драйверы из папки `drivers`
3. Подключите Arduino к компьютеру через USB
4. Запустите Arduino IDE
5. В «Tools → Board» выберите модель вашей платы
6. В «Tools → Serial Port» выберите порт, куда она подключена
7. Пишите программу или загружайте готовый пример из «File → Examples»
8. Жмите «Upload» на панели инструментов для прошивки платы!

Перепрошивать плату можно сколько угодно раз. Программа сохраняется после обесточивания платы.

## Внешний вид Arduino IDE



## Если прошивка не удаётся

Проверьте, что:

- Плата получает питание, горит светодиод «ON»
- Драйверы под Windows установились корректно, и в диспетчере устройств вы видите устройство «Arduino Uno»
- Вы выбрали правильную модель платы и правильный порт (пункты 5 и 6)
- USB-кабель исправен

# Эксперимент 1. Маячок

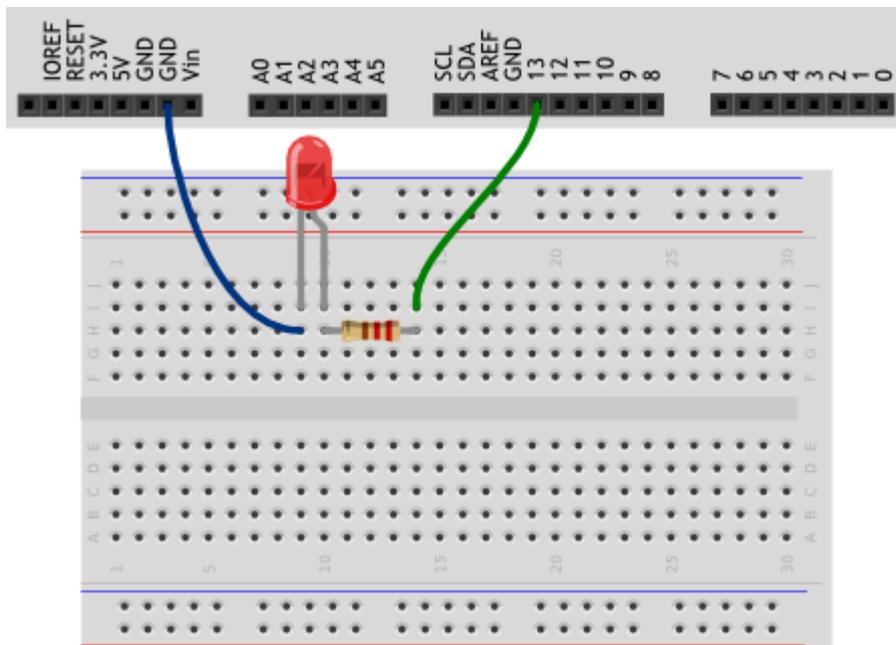
## Список деталей для эксперимента

- 1 плата [Arduino Uno](#)
- 1 бесплаечная [макетная плата](#)
- 1 [светодиод](#)
- 1 [резистор](#) номиналом 220 Ом
- 2 провода «папа-папа»

## Принципиальная схема



## Схема на макетке



## Обратите внимание

- Не забудьте, как соединены рельсы в бесплаечной [макетной плате](#). Если на вашей макетке красная и синяя линии вдоль длинных рельс прерываются в середине, значит проводник внутри макетки тоже прерывается!
- Катод («минус») светодиода – короткая ножка, именно её нужно соединять с землёй (GND)
- Не пренебрегайте резистором, иначе светодиод выйдет из строя

- Выбрать резистор нужного номинала можно с помощью [таблицы маркировки](#) или с помощью мультиметра в режиме измерения сопротивления
- Плата Arduino имеет три пина GND, используйте любой из них

## Скетч

```
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
}
```

## Пояснения 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 завершенного 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:
  - `pin` – номер цифрового порта, на который мы отправляем сигнал
  - `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

## Вопросы для проверки 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