

Муниципальное бюджетное учреждение дополнительного образования
«Городской центр детского технического творчества
им. В.П.Чкалова» г. Казани



16 городской Конкурс-выставка технического творчества школьников «Дети. Техника. Творчество»

Начало работы с Arduino.

Первые шаги

*Рекомендации для педагогов дополнительного образования
(для начинающих)*

Васянин Евгений Александрович,
педагог дополнительного образования

Начало работы с Arduino

На этом уроке, мы разберем, что нужно для начала работы с Arduino?

Результатом работы первого занятия будет сборка электрической схемы «Маячок»

Ардуино – это мостик из виртуального компьютерного мира в мир реальных вещей и устройств. Написав программу на обычном компьютере, мы управляем с ее помощью не виртуальными объектами, а вполне себе реальными датчиками, двигателями, экранами. Мы меняем мир вокруг себя – просто программируя на компьютере, используя бесплатный софт и множество уже готовых примеров библиотек

С чего начать?

Подготовить:

- Плата **Ардуино** с USB кабелем для подключения к компьютеру.
- Макетная монтажная плата и провода.
- Комплект базовых электронных компонентов и переходник для батарейки типа крона.
- Установленная на компьютер среда программирования **Arduino IDE**.

В дальнейшем, если занятия действительно увлекут и будет желание продолжить эксперименты, то список оборудования будет расширяться:

[Датчики](#)

Экраны и индикаторы.

Двигатели и [сервоприводы](#), реле и [драйверы двигателей](#).

Модули связи.

Разнообразные дополнительные модули и [платы расширения](#) (шилды)

Плата Ардуино

В плату **можно** втыкать провода и подключать множество разных элементов. Чаще всего, для соединения **используется** макетная плата для монтажа без пайки. **Можно добавлять** светодиоды, датчики, кнопки, двигатели, модули связи, реле и создавать сотни вариантов интересных проектов умных устройств. **Плата Ардуино** – **это** умная розетка, которая будет включать и выключать все присоединенное в зависимости от того, как ее запрограммировали

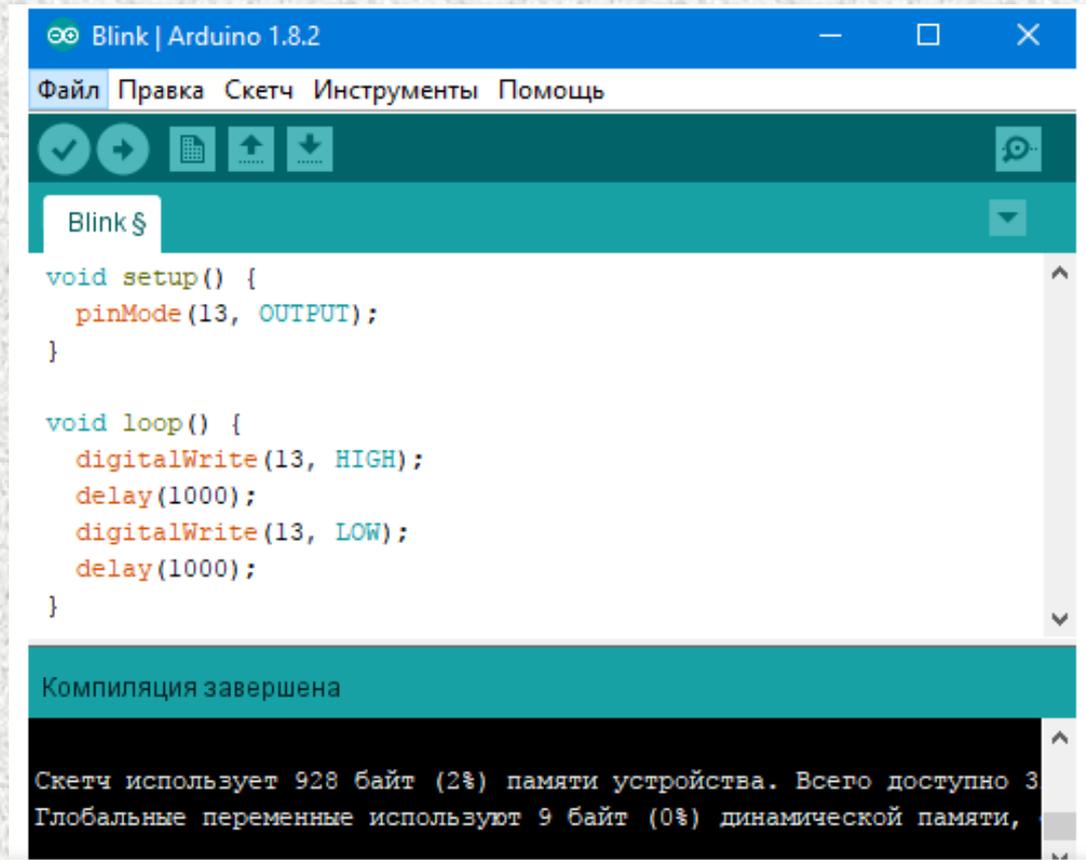


Работа над проектом

Вся работа над проектом **разбивается на следующие этапы:**

- Придумываем идею и проектируем.
- Собираем электрическую схему. Тут нам пригодится макетная плата, упрощающая монтаж элементов. Безусловно, понадобятся навыки работы с электронными приборами и умение [пользоваться мультиметром](#).
- Подключаем [плату Arduino](#) к компьютеру через USB.
- [Пишем программу](#) и записываем ее в плату буквально нажатием одной кнопки на экране в [специальной среде программирования Arduino](#).
- Отсоединяем от компьютера. Теперь устройство будет работать автономно – при включении питания оно будет управляться той программой, которую мы в него записали.

Программа и среда программирования



```
Arduino IDE: Blink | Arduino 1.8.2
Файл  Правка  Скетч  Инструменты  Помощь

Blink $

void setup() {
  pinMode(13, OUTPUT);
}

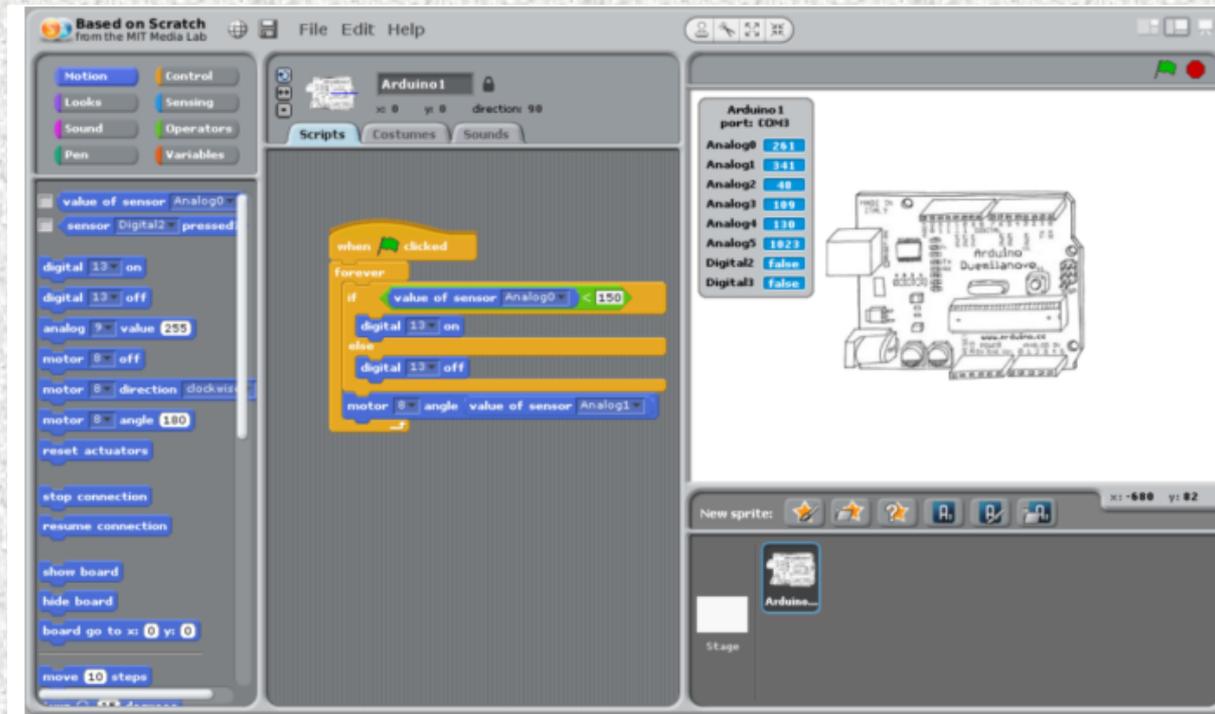
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}

Компиляция завершена

Скетч использует 928 байт (2%) памяти устройства. Всего доступно 3
Глобальные переменные используют 9 байт (0%) динамической памяти,
```

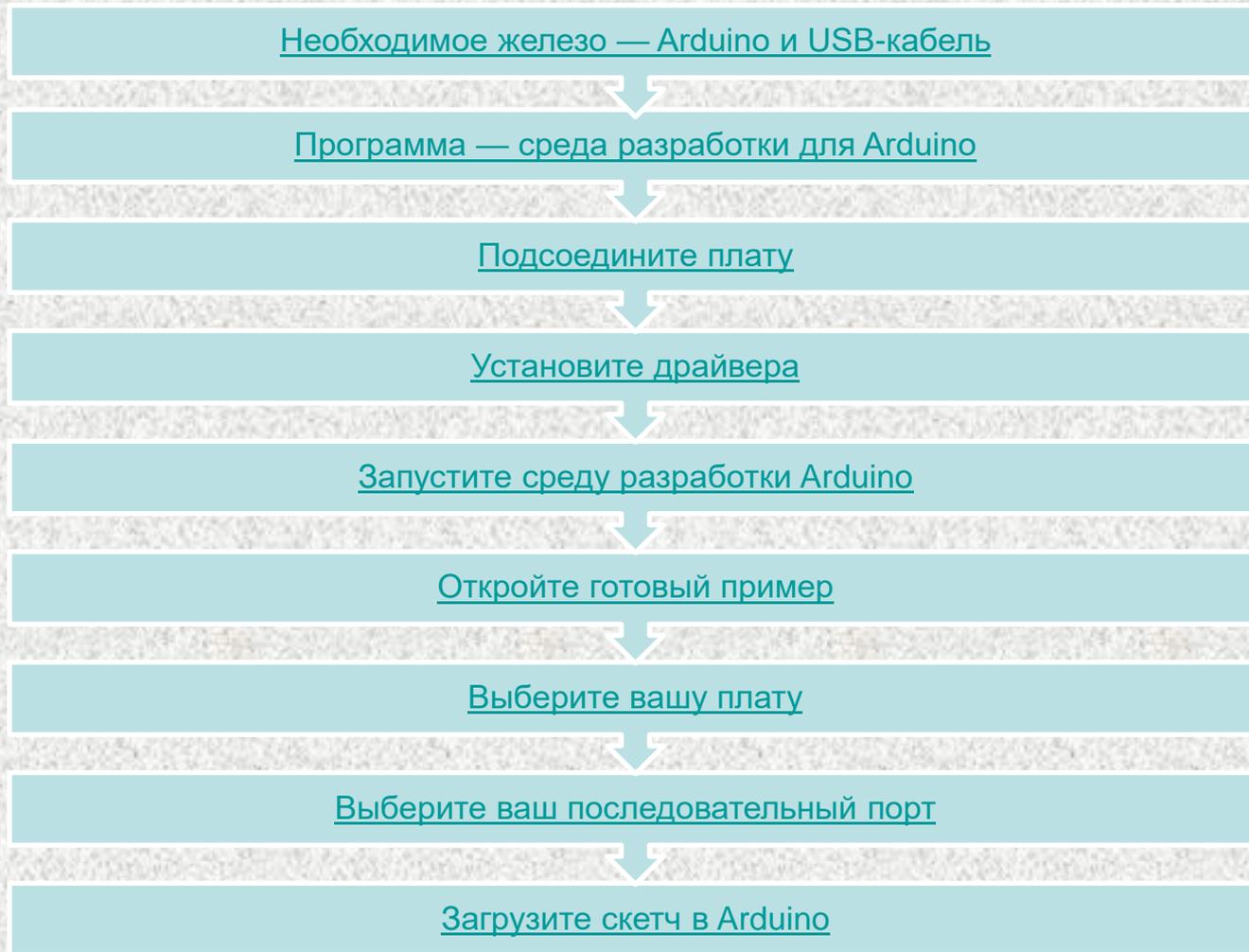
На экране показана программа (на сленге ардуинчиков текст программы называется “скетч”), которая будет мигать лампочкой, подсоединенной к 13 входу на плате Ардуино UNO. Как видим, программа вполне проста и состоит из понятных для знающих английский язык инструкций. В языке программирования Arduino используется свой диалект языка C++, но все возможности C++ поддерживаются

Есть и другой вариант написания кода – визуальный редактор. Тут не нужно ничего писать – можно просто перемещать блоки и складывать из них нужный алгоритм. Программа загрузится в подключенную плату одним нажатием кнопки мыши!



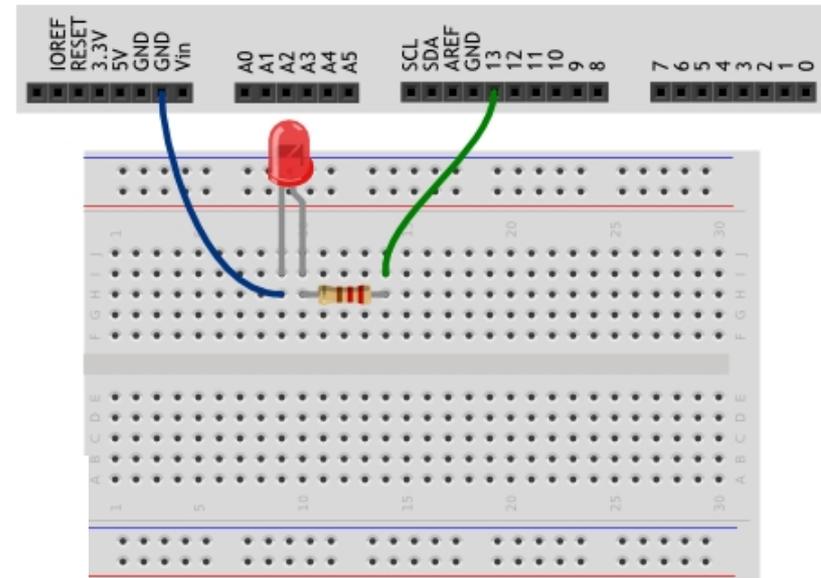
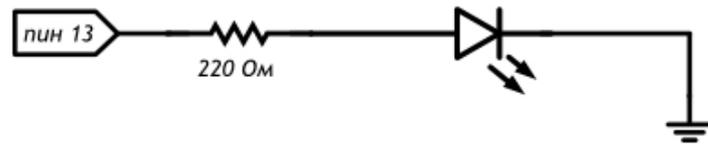
Визуальную среду рекомендуется использовать школьникам младших классов, более старшим инженерам лучше сразу изучать “настоящий” Ардуино – это довольно просто, к тому же знания C++ никому не повредят

Как подключить плату Arduino к компьютеру и загрузить ваш первый скетч



Алгоритм действий при сборке электрической схемы «Маячок»

СХЕМА



НЕОБХОДИМОЕ ОБОРУДОВАНИЕ

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

ОБРАТИТЬ ВНИМАНИЕ!

Не забудьте, как соединены рельсы в беспаячной 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

p010 blink.ino

```
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
}
```

КОММЕНТАРИИ 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

Видеоинструкция

<https://cloud.mail.ru/public/iZys/zBxdmk94o>