

# СОДЕРЖАНИЕ

---

Что такое Arduino .....	5
Установка Arduino IDE .....	8
1 Светодиод. Мигаем светодиодом .....	12
2 Кнопка. Обрабатываем нажатие кнопки на примере зажигания светодиода. Боремся сдребезгом .....	15
3 Потенциометр. Показываем закон Ома на примере яркости светодиода .....	20
4 Светодиодная шкала 10 сегментов. Крутим потенциометр, меняем количество светящихся светодиодов .....	23
5 RGB-светодиод. Широтно-импульсная модуляция. Переливаемся цветами радуги .....	28
6 Семисегментный индикатор одноразрядный. Выводим цифры .....	33
7 Матрица 4-разрядная из 7-сегментных индикаторов. Делаем динамическую индикацию .....	36
8 Микросхема сдвигового регистра 74НС595. Управляем матрицей из 4 разрядов, экономим выводы Ардуино .....	43
9 Матрица светодиодная 8x8 .....	47
10 Пьезоизлучатель. Управляем пьезоизлучателем: меняем тон, длительность, играем Имперский марш .....	50
11 Транзистор MOSFET. Показываем усилительные качества транзистора. На примере электродвигателя изменяем обороты .....	55
12 Реле. Управляем реле через транзистор .....	58
13 Фоторезистор. Обрабатываем освещённость, зажига я или гася светодиоды .....	62
14 Датчик температуры аналоговый LM335. Принцип работы, пример работы .....	66
15 Индикатор LCD1602. Принцип подключения, вывод информации на него .....	69
16 Графический индикатор на примере Nokia 5110 .....	73
17 Сервопривод. Крутим потенциометр, меняем положение .....	77

18	Джойстик. Обработка данных от джойстика. Управление Pan/Tilt Bracket с помощью джойстика .....	81
19	Шаговый двигатель 4-фазный, с управлением на ULN2003 (L293) .....	85
20	Датчик температуры DS18B20 .....	89
21	Датчик влажности и температуры DHT11 .....	93
22	Датчики газов. Принцип работы, пример работы .....	97
23	Ультразвуковой датчик расстояния HC-SR04. Принцип работы, подключение, пример.....	100
24	3-осевой гироскоп + акселерометр на примере GY-521.....	104
25	ИК-фотоприемник и ИК-пульт. Обработка команд от пульта.....	107
26	Часы реального времени. Принцип работы, подключение, примеры .....	111
27	SD-карта. Чтение и запись данных .....	117
28	Считыватель RFID на примере RC522. Принцип работы, подключение, примеры .....	120
29	Работа с Интернетом на примере Arduino Ethernet shield W5100 .....	126
30	Беспроводная связь. Модуль Wi-Fi ESP8266.....	131
31	Беспроводная связь. Модуль Bluetooth HC-05 .....	137
32	Беспроводная связь. Модуль GSM/GPRS SIM900 .....	142
33	Модуль GPS. Принцип работы, подключение, примеры .....	147

# Что такое Arduino

Arduino – это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Плата Arduino состоит из микроконтроллера Atmel AVR и элементов обвязки для программирования и интеграции с другими схемами. На многих платах присутствует линейный стабилизатор напряжения +5 В или +3,3 В. Тактирование осуществляется на частоте 16 или 8 МГц кварцевым резонатором (в некоторых версиях – керамическим резонатором). В микроконтроллер предварительно прошивается загрузчик Boot-Loader, поэтому внешний программатор не нужен. Устройство программируется через USB без использования программаторов.

Существует несколько версий платформ Arduino. Версия Leonardo базируется на микроконтроллере ATmega32u4. Uno, Nano, Duemilanove построены на микроконтроллере Atmel ATmega328. Старые версии платформы Diecimila и первая рабочая Duemilanoves были разработаны на основе Atmel ATmega168. Arduino Mega2560, в свою очередь, построена на микроконтроллере ATmega2560. А самые последние версии Arduino Due – на базе микропроцессора Cortex.

Версия UNO (рис. 1) является одной из самых популярных и широко используемой для небольших проектов.

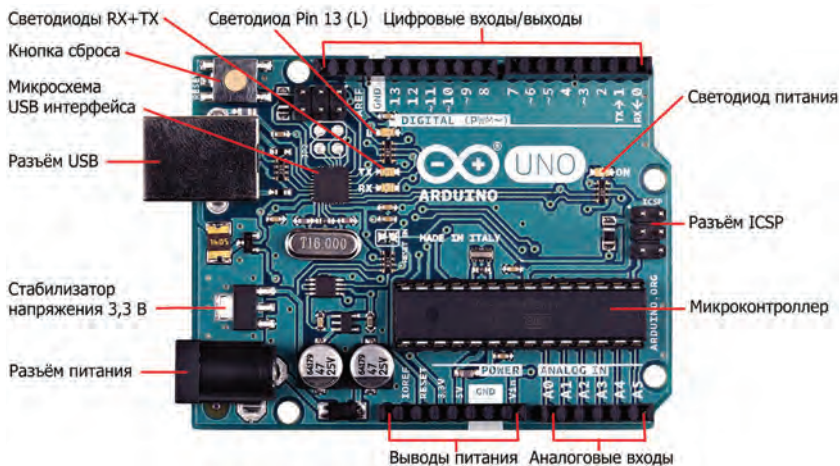


Рис. 1. Плата Arduino UNO

Характеристики платы Arduino UNO показаны в табл. 1.1.

**Таблица 1.1**

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Напряжение питания (рекомендуемое)	7–12 В
Напряжение питания (предельное)	6–20 В
Цифровые входы/выходы	14 (из них 6 могут использоваться в качестве ШИМ-выходов)
Аналоговые входы	6
Максимальный ток одного вывода	40 мА
Максимальный выходной ток вывода 3.3 В	50 мА
Flash-память	32 КБ (ATmega328), из которых 0,5 КБ используются загрузчиком
SRAM	2 КБ (ATmega328)
EEPROM	1 КБ (ATmega328)
Тактовая частота	16 МГц

Каждый из 14 цифровых выводов может работать в качестве входа или выхода. Уровень напряжения на выводах ограничен 5 В. Максимальный ток, который может отдавать или потреблять один вывод, составляет 40 мА. Все выводы сопряжены с внутренними подтягивающими резисторами (по умолчанию отключенными) номиналом 20–50 кОм. Помимо этого, некоторые выводы Ардуино могут выполнять дополнительные функции:

- последовательный интерфейс: выводы 0 (RX) и 1 (TX);
- внешние прерывания: выводы 2 и 3;
- ШИМ: выводы 3, 5, 6, 9, 10 и 11 могут выводить 8-битные аналоговые значения в виде ШИМ-сигнала;
- интерфейс SPI: выводы 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK);
- светодиод: 13. Встроенный светодиод, подсоединенный к выводу 13.

В Arduino Uno есть 6 аналоговых входов (A0–A5), каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 различных значения). По умолчанию измерение напряжения осуществляется относительно диапазона от 0 до 5 В. Тем не менее верхнюю границу этого диапазона можно изменить, используя вывод AREF и функцию `analogReference()`. Некоторые из аналоговых входов имеют дополнительные функции:

- TWI: вывод A4 или SDA и вывод A5 или SCL.

В Arduino Uno есть восстанавливаемые предохранители, защищающие USB-порт компьютера от коротких замыканий и перегрузок. Несмотря на то что большинство компьютеров имеют собственную защиту, такие предохранители обеспечивают дополнительный уровень защиты. Если от USB-порта потребляется ток более 500 мА, предохранитель автоматически разорвет соединение до устранения причин короткого замыкания или перегрузки.

# Установка Arduino IDE

Разработка собственных приложений на базе плат, совместимых с архитектурой Arduino, осуществляется в официальной бесплатной среде программирования Arduino IDE. Среда предназначена для написания, компиляции и загрузки собственных программ в память микроконтроллера, установленного на плате Arduino-совместимого устройства. Основой среды разработки является язык Processing/Wiring – это фактически обычный C++, дополненный простыми и понятными функциями для управления вводом/выводом на контактах. Существуют версии среды для операционных систем Windows, Mac OS и Linux.

Последнюю версию среды Arduino можно скачать со страницы загрузки официального сайта <http://arduino.cc/en/Main/Software>.

Рассмотрим установку Arduino IDE на компьютере с операционной системой Windows. Отправляемся на страницу <http://arduino.cc/en/Main/Software>, выбираем версию для операционной системы Windows и скачиваем архивный файл. Он содержит все необходимое, в том числе и драйверы. По окончании загрузки распаковываем скачанный файл в удобное для себя место.

Теперь необходимо установить драйверы. Подключаем Arduino к компьютеру. На контроллере должен загореться индикатор питания – зеленый светодиод. Windows начинает попытку установки драйвера, которая заканчивается сообщением «Программное обеспечение драйвера не было установлено». Открываем Диспетчер устройств. В составе устройств находим значок Arduino Uno – устройство отмечено восклицательным знаком. Щелкаем правой кнопкой мыши на значке Arduino Uno и в открывшемся окне выбираем пункт **Обновить драйверы** и далее пункт **Выполнить поиск драйверов на этом компьютере**. Указываем путь к драйверу – ту папку на компьютере, куда распаковывали скачанный архив. Пусть это будет папка drivers каталога установки Arduino – например, C:\arduino-1.0\drivers. Игнорируем все предупреждения Windows и получаем в результате сообщение **Обновление программного обеспечения для данного устройства завершено успешно**. В заголовке окна будет указан и COM-порт, на который установлено устройство. Теперь можно запустить Arduino IDE.

Среда разработки Arduino (см. рис. 2) состоит из:

- редактора программного кода;
- области сообщений;

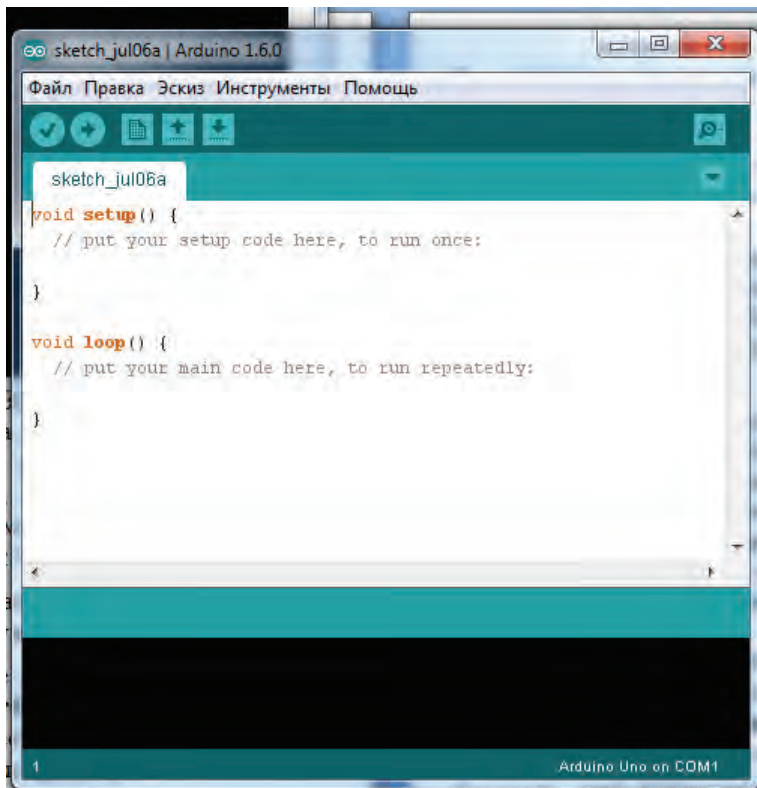


Рис. 2. Среда Arduino IDE

- окна вывода текста;
- панели инструментов с кнопками часто используемых команд;
- нескольких меню.

Программа, написанная в среде Arduino, носит название *скетч*. Скетч пишется в текстовом редакторе, который имеет цветовую подсветку создаваемого программного кода. Во время сохранения и экспорта проекта в области сообщений появляются пояснения и информация об ошибках. Окно вывода текста показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины.

Разрабатываемым скетчам дополнительная функциональность может быть добавлена с помощью библиотек, представляющих собой специальным образом оформленный программный код, реализующий некоторый функционал, который можно подключить к создаваемому проекту. Специализированных библиотек существует множество. Обычно библиотеки пишутся так, чтобы упростить решение той или иной задачи и скрыть от разработчика детали программно-аппаратной реализации. Среда Arduino IDE поставляется с набором стандартных библиотек. Они находятся в подкаталоге `libraries` каталога установки Arduino. Необходимые библиотеки могут быть также загружены с различных ресурсов. Если библиотека установлена правильно, то она появляется в меню **Эскиз | Импорт библиотек**. Выбор библиотеки в меню приведет к добавлению в исходный код строки

```
#include <имя библиотеки.h>
```

Эта директива подключает заголовочный файл с описанием объектов, функций и констант библиотеки, которые теперь могут быть использованы в проекте. Среда Arduino будет компилировать создаваемый проект вместе с указанной библиотекой.

Перед загрузкой скетча требуется задать необходимые параметры в меню **Инструменты | Плата** (Tools | Board) (рис. 3) и **Инструменты | Последовательный порт** (рис. 4).

Современные платформы Arduino перезагружаются автоматически перед загрузкой. На старых платформах необходимо нажать кнопку перезагрузки. На большинстве плат во время процесса загрузки будут мигать светодиоды RX и TX.

При загрузке скетча используется загрузчик (bootloader) Arduino – небольшая программа, загружаемая в микроконтроллер на плате. Она позволяет загружать программный код без использования дополнительных аппаратных средств. Работа загрузчика распознается по миганию светодиода на цифровом выводе D13.

Монитор последовательного порта (Serial Monitor) отображает данные, посылаемые в платформу Arduino (плату USB или плату последовательной шины).

Теперь, когда мы немного узнали об Arduino и среде программирования Arduino IDE, перейдем к практическим занятиям – экспериментам.



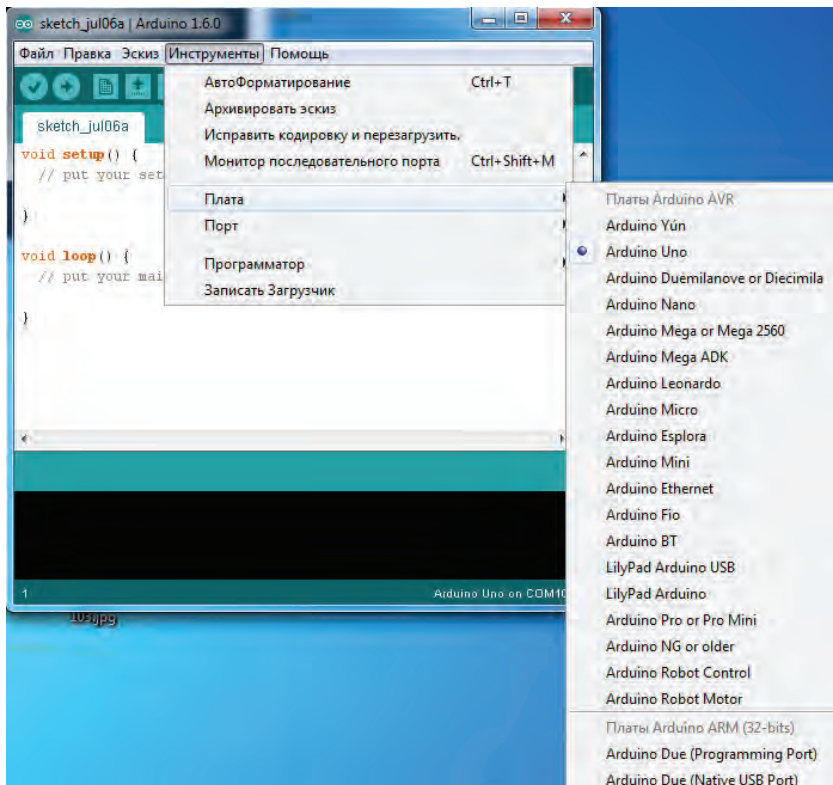


Рис. 3. Выбор Arduino платы

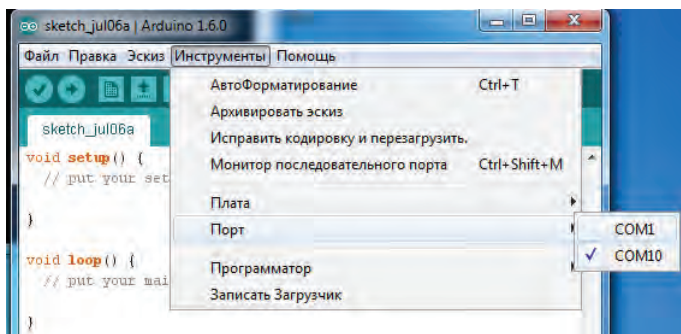


Рис. 4. Выбор порта подключения платы Arduino

# 1 Светодиод. Мигаем светодиодом

В этом эксперименте мы научимся управлять светодиодом. Заставим его мигать.

## **Необходимые компоненты:**

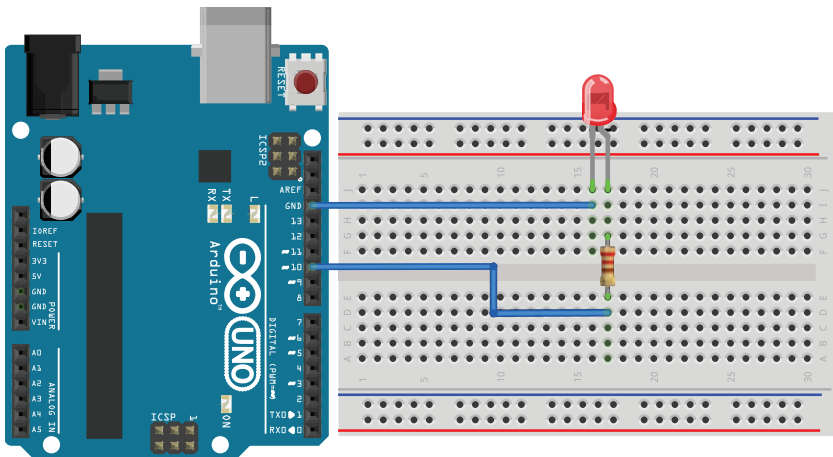
- контроллер Arduino UNO;
- плата для прототипирования;
- светодиод;
- резистор 220 Ом;
- провод папа-папа.

Светодиод – это полупроводниковый прибор, преобразующий электрический ток непосредственно в световое излучение. По-английски светодиод называется light emitting diode, или LED. Цветовые характеристики светодиодов зависят от химического состава использованного в нем полупроводника. Светодиод излучает в узкой части спектра, его цвет чист, что особенно ценят дизайнеры. Светодиод механически прочен и исключительно надежен, его срок службы может достигать 100 тысяч часов, что почти в 100 раз больше, чем у лампочки накаливания, и в 5–10 раз больше, чем у люминесцентной лампы. Наконец, светодиод – низковольтный электроприбор, а стало быть, безопасный.

Светодиоды поляризованы, имеют значение, в каком направлении подключать их. Положительный вывод светодиода (более длинный) называется анодом, отрицательный – катодом. Как и все диоды, светодиоды позволяют току течь только в одном направлении – от анода к катоду. Поскольку ток протекает от положительного к отрицательному, анод светодиода должен быть подключен к цифровому сигналу 5 В, а катод должен быть подключен к земле.

Мы будем подключать светодиод к цифровому контакту D10 Arduino последовательно с резистором. Светодиоды должны быть всегда соединены последовательно с резистором, который выступает в качестве ограничителя тока. Чем больше значение резистора, тем больше он ограничивает ток. В этом эксперименте мы используем резистор номиналом 220 Ом. Схема подключения приведена на рис. 1.1.

Как подобрать ограничительный резистор и как будет влиять номинал резистора на яркость светодиода, мы рассмотрим в эксперименте 3.



fritzing

Рис. 1.1. Схема подключения светодиода

Светодиод последовательно с резистором подключаем к цифровому выводу Arduino D10. По умолчанию все выводы Arduino сконфигурированы как входы. Мы собираемся использовать вывод Arduino как выход, поэтому необходимо его переконфигурировать, выдав контроллеру соответствующую команду.

```
pinMode(10, OUTPUT);
```

Для мигания светодиода необходимо попеременно с определенным интервалом подавать на вывод Arduino сигналы HIGH (высокий уровень или 1) и LOW (низкий уровень или 0). Интервал изменения сигнала на выходе D10 Arduino будем устанавливать с помощью команды `delay()`, задерживающей выполнение скетча на заданное время в миллисекундах (мс).

Скетч эксперимента приведен в листинге 1.1.

### Листинг 1.1

```
const int LED=10;    // вывод для подключения светодиода 10 (D10)
void setup()
{
    // Конфигурируем вывод подключения светодиода как выход (OUTPUT)
    pinMode(LED, OUTPUT);
}
void loop()
```

```
{  
  // включаем светодиод, подавая на вывод 1 (HIGH)  
  digitalWrite(LED,HIGH);  
  // пауза 1 сек (1000 мс)  
  delay(1000);  
  // выключаем светодиод, подавая на вывод 0 (LOW)  
  digitalWrite(LED,LOW);  
  // пауза 1 сек (1000 мс)  
  delay(1000);  
}
```

Порядок подключения:

1. Длинную ножку светодиода (анод) подключаем к цифровому выводу D10 Arduino, другую (катод) – через резистор 220 Ом к выводу GND (см. рис. 1.1).
2. Загружаем в плату Arduino скетч из листинга 1.1.
3. Наблюдаем процесс мигания светодиода.

Теперь мы можем поэкспериментировать с периодом мигания светодиода, меняя в скетче значения задержки в функции `delay()`. На странице <http://arduino-kit.ru/0001> вы можете посмотреть видеоурок данного эксперимента и скачать скетч, представленный в листинге 1.1.