

Arduino. Установка программного обеспечения.

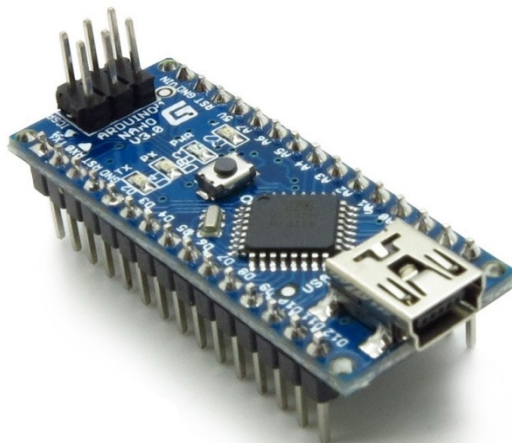
Arduino - это контроллер (управляющий модуль). На плате размещены процессор, микросхема конвертера USB и выводы для подключения к контроллеру внешних устройств. В качестве внешних устройств могут различные исполнительные устройства и элементы: светодиоды, реле, электромоторы.

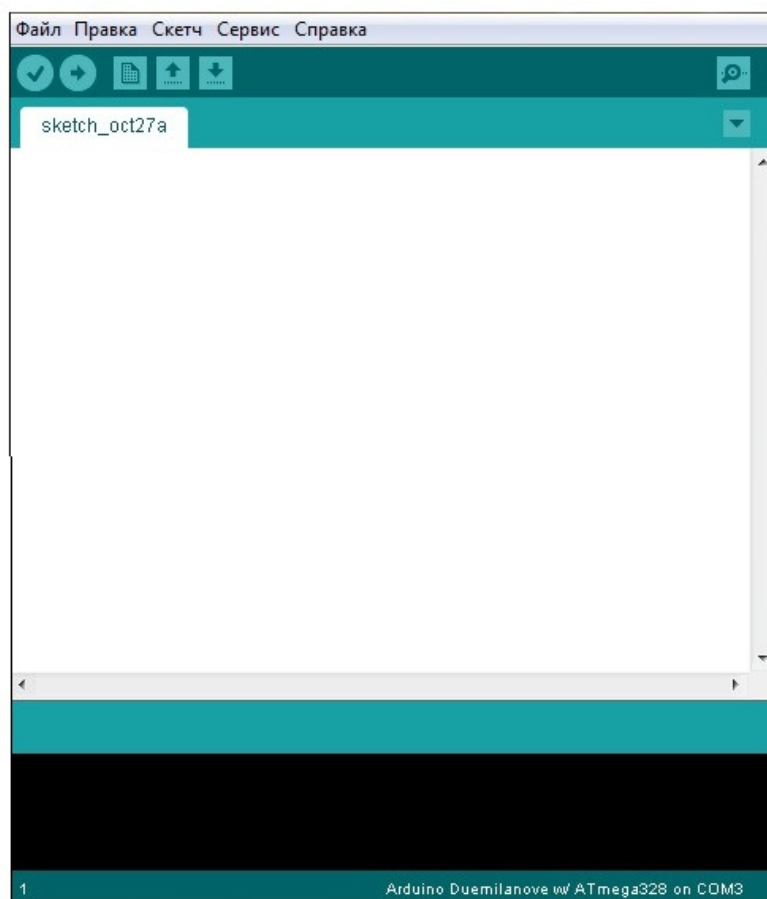
Порядок управления (алгоритм) этими устройствами определяется программой, написанной программистом для контроллера Arduino и загруженной в память процессора.

Для написания, редактирования, отладки программ создана специализированная среда – IDE. Программа пишется на языке C++. Для загрузки написанной программы в процессор не требуется каких-либо дополнительных устройств. Надо просто подключить плату Arduino к компьютеру USB кабелем.

Платформа Arduino.

Arduino состоит из двух основных частей – непосредственно платы Arduino , которая является частью аппаратного обеспечения, над которым вы работаете при создании собственных объектов. И среды разработки IDE - программного обеспечения, которое вы запускаете на своём компьютере. IDE необходимо для создания скетчей (маленьких компьютерных программ), которые затем выгружаются на плату Arduino. Скетч «говорит» плате, что делать.





Аппаратное обеспечение Arduino.

Плата Arduino - небольшая плата микроконтроллера, состоящая из небольшой схемы, содержащей целый компьютер в маленьком чипе (микроконтроллер).

Существует много вариантов этой платы. Но наши проекты мы будем реализовывать на плате Arduino Nano. Так как на наш взгляд это самый удобный вариант для наших занятий.

На рисунке представлена плата Arduino Nano, с которой мы будем работать.

Поясним назначение контактов платы.

Контакты 0-13 - контактов цифрового ввода-вывода.

Они могут быть как входами, так и выходами, что определяется вашей программой (скетчем).

Контакты 3, 5, 6, 9, 10 и 11 могут использоваться, как ШИМ – широтно-импульсная модуляция.)

Контакты 0–5 - контакты аналогового входа.

Эти контакты для аналогового входа получают аналоговые значения (например, величину напряжение в датчике) и преобразовывают их в цифровые значения от 0 до 1023.

Электрическое питание может быть подано на плату через USB-порт от компьютера, или от AC-адаптера 9 вольт через разъём питания. Плата может сама выбирать источник питания. Т.е., если плата подключена к компьютеру через USB кабель и к разъёму питания подключен адаптер, то плата выберет адаптер.

Интегрированная среда разработки (IDE).

IDE (интегрированная среда разработки) - это специальная программа, работающая на вашем компьютере, которая позволяет писать управляющую программу (скетч) для платы Arduino на простом языке. После нажатия



кнопки «загрузить»

код, который вы написали, будет транслирован в команды, понятные микроконтроллеру и загружен в память.

Процесс программирования Arduino упрощённо выглядит следующим образом:

- подключаем плату Ардуино к USB-порту компьютера.
- пишем скетч, для работы платы.
- выгружаем этот скетч на плату через USB-соединение и ждём несколько секунд для перезапуска платы.

Плата приступит к выполнению написанного вами скетча.

Установка Arduino на ваш компьютер.

Чтобы начать работать с платой Arduino, сначала необходимо установить среду разработки (IDE).

Установить IDE можно удобнее всего с сайта разработчика:

www.arduino.cc/en/Main/Software.

Последняя рабочая версия - **arduino-1.6.7**,

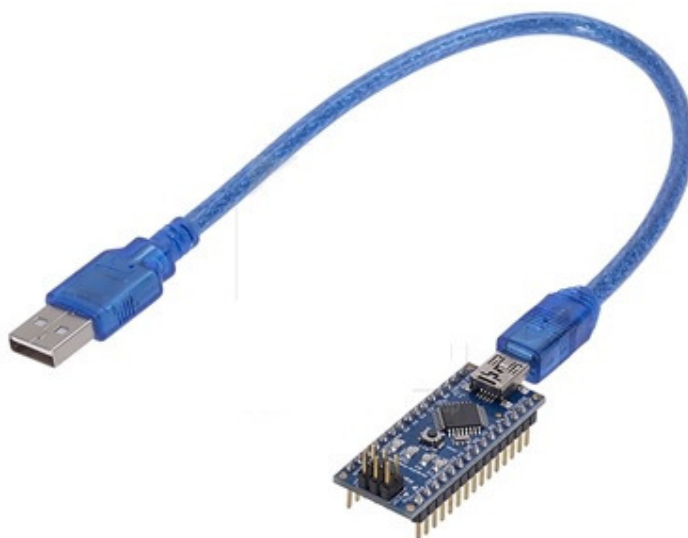
При загрузке программы с сайта www.arduino.cc вы можете выбрать подходящую для вашей операционной системы версию ПО.

После установки ПО запустите IDE.



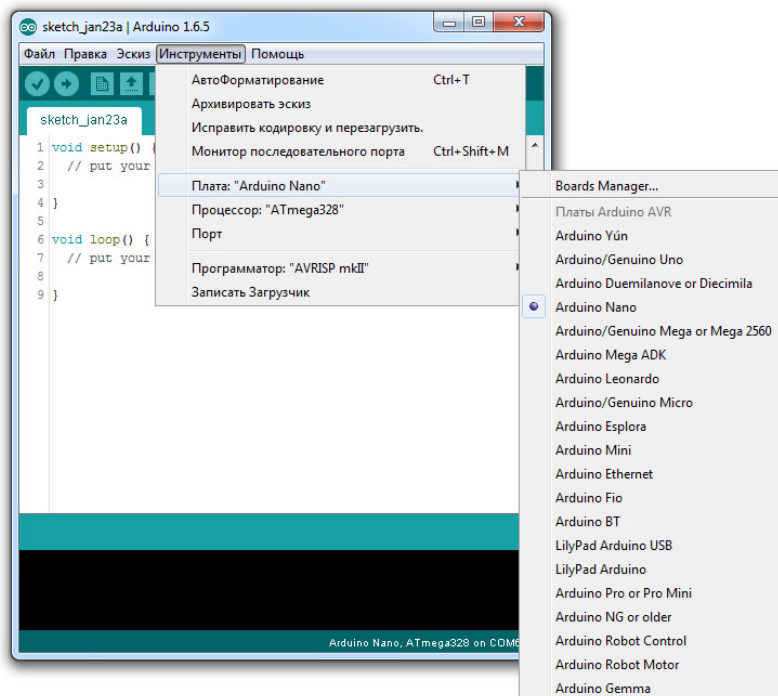
На экране монитора у вас появится окно среды проектирования IDE.

Подключите USB кабелем контроллер Arduino Nano к компьютеру.

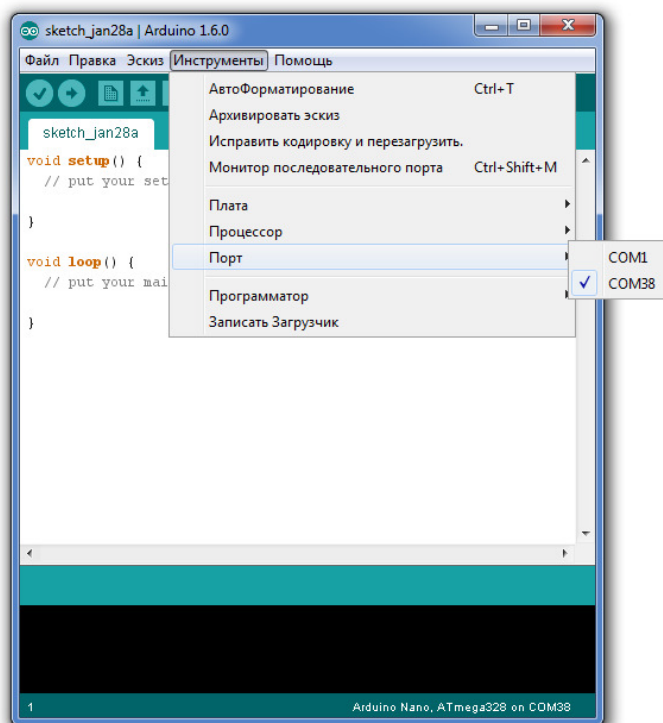


Последовательно нажмите кнопки ИСТРУМЕНТЫ, ПЛАТА, ARDUINO Nano. Далее необходимо выбрать процессор ATmega328 порт, который предложит система. Теперь плата готова к работе.

Выбор платы ARDUINO NANO



Выбор порта



Система готова к работе.

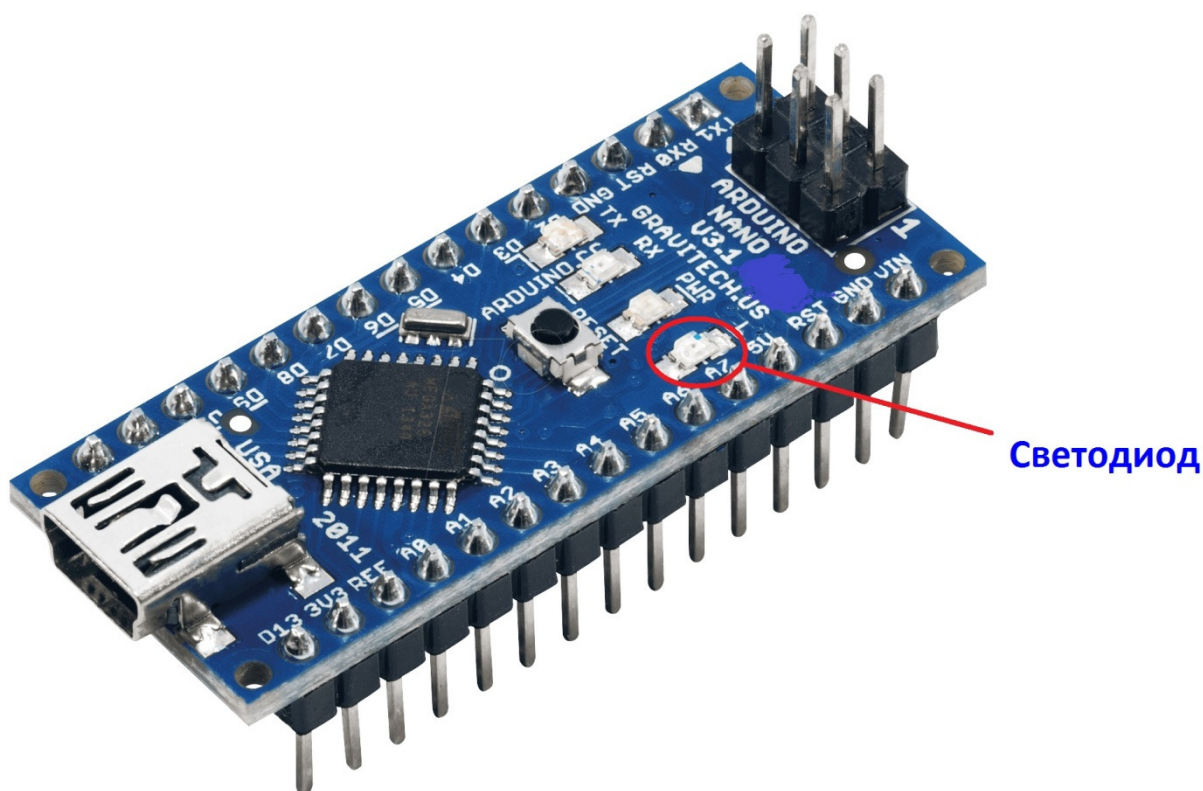
Теперь проведем небольшой ознакомительный эксперимент.

Мигание светодиодом.

Задачи эксперимента. Знакомство со средой разработки, проверкой работоспособности программы и контроллера Arduino Nano.

Скетч, который управляет миганием светодиода - первая программа, которую мы загрузим в плату Arduino.

Светоизлучающий диод (светодиод) - это маленький электрический компонент, который похож на маленькую лампочку. И ему для работы хватит напряжения питания 5 вольт. Плата Arduino уже имеет установленный светодиод. На рисунке он выделен красным кружком

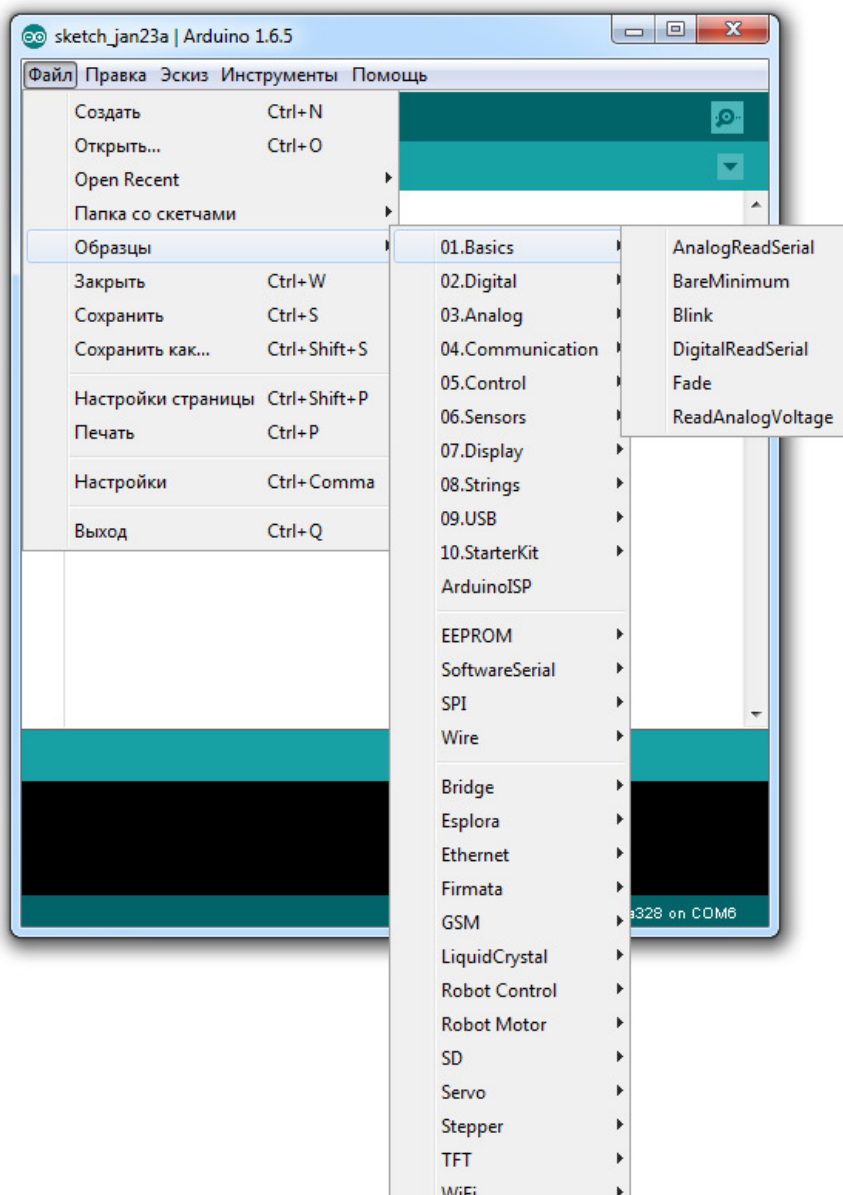


При подключении платы Arduino Nano к компьютеру светодиод мигает с частотой 0,5 Герц. Т.е. 1 секунду светится, затем на 1 секунду гаснет.

«Научим» светодиод мигать по-другому.

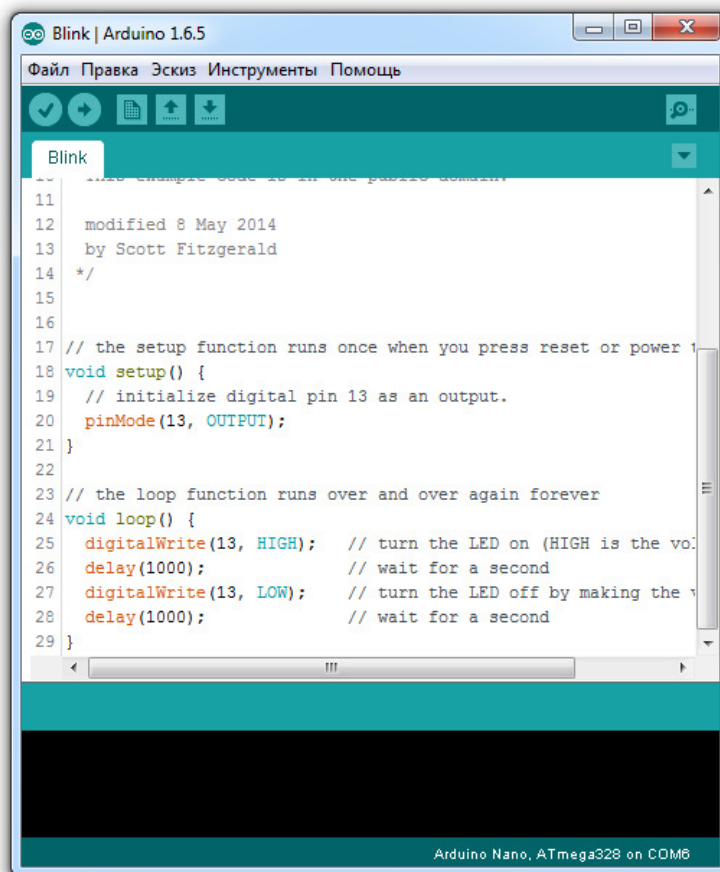
В среде разработки программ Arduino IDE есть уже большое количество примеров.

Откроем пример для управления мигания светодиодом. Для этого в Arduino IDE выбираем путь до кнопки «Blink» (Файл – образцы – 01.Basic – Blink)

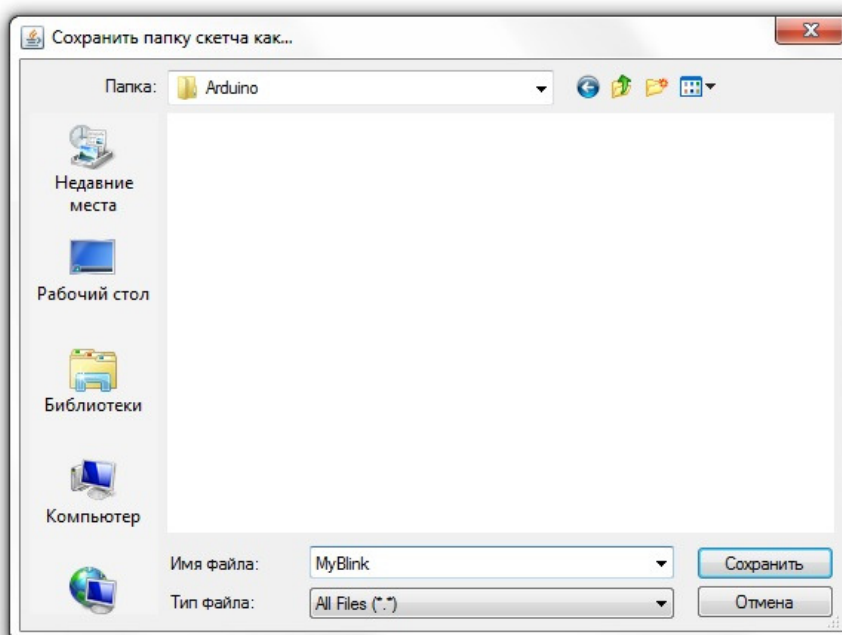


После загрузки в окне IDE появится текст программы для мигания светодиода.

Но прежде, чем начать работы по изменению скорости мигания светодиода необходимо сохранить копию этого скетча. Для этого в меню Файл в IDE выберите опцию «Сохранить как...». А затем сохраните скетч, присвоив ему имя MyBlink.



Но прежде, чем начать работы по изменению скорости мигания светодиода необходимо сохранить копию этого скетча. Для этого в меню Файл в IDE выберите опцию «Сохранить как...». А затем сохраните скетч, присвоив ему имя MyBlink.



Вы сохранили копию 'Blink' в своей папке. Это означает, что если вы хотите найти его снова, вы можете просто открыть его с помощью опции **Файл – Папка со скетчами**. Подключите плату Arduino к компьютеру с помощью USB-кабеля и убедитесь, что тип платы и тип последовательного порта установлены правильно. Дождитесь сообщения, что драйвера найдены и установлены.

Arduino IDE покажет вам текущие настройки платы в нижней части окна.



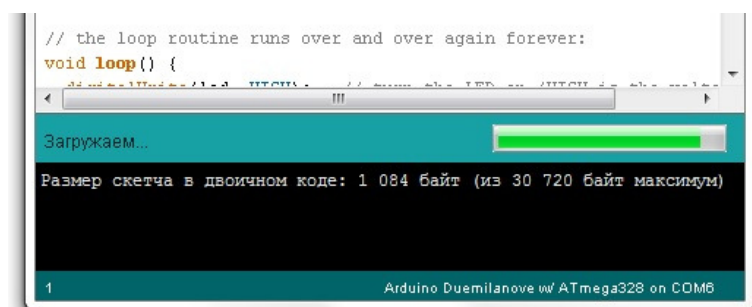
Нажмите на кнопку «Загрузить». Это вторая кнопка слева на панели инструментов.



В нижней части IDE выводит ряд сообщений. Сначала выводится сообщение о компиляции скетча. Т.е. о преобразовании программы в формат, понятный для процессора.



Далее статус процесса изменится на «Загружаем». На данный момент, светодиоды на Arduino должны начать мерцать, указывая, что скетч загружается в плату.



Наконец в строке состояния появится текст «Загрузка выполнена».

Текст в информационном окне говорит нам, что скетч занимает 1084 байта из 30720 байт максимум.


```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Загрузка выполнена.

Размер скетча в двоичном коде: 1 084 байт (из 30 720 байт максимум)
Размер скетча в двоичном коде: 1 084 байт (из 30 720 байт максимум)

1 Arduino Duemilanove w/ ATmega328 on COM6

Но, если вы забыли подключить плату, или драйверы установлены неправильно, или порт выбран неправильно, то в окне может появиться следующая надпись:

```
Последовательный порт 'COM3' не найден. Вы выбрали нужный из меню Сервис > I
at processing.app.Editor$DefaultExportHandler.run
(Editor.java:2380)
at java.lang.Thread.run(Thread.java:619)
```

1 Arduino Duemilanove w/ ATmega328 on COM3

Проверьте подключение платы к компьютеру и еще раз загрузите скетч.

Код программы.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Обратите внимание, что текст программы состоит из непосредственно инструкций и комментариев этих инструкций. Комментарии не влияют на выполнение программы процессором. С помощью комментариев можно более подробно описывать работу программы.

Многострочные комментарии выделяются символами `/*` и `*/`.

Однострочные комментарии выделяются символами `//` только в начале строки.

Так же обратите внимание, что инструкции пишутся на английском языке. Комментарии в примерах, которые входят в IDE, так же написаны на английском языке.

Но при самостоятельном написании скетча комментарии можно писать на русском языке.

Рассмотрим код программы построчно:

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

Функция **Setup** выполняется один раз после старта скетча или при включении питания Arduino. И эта функция должна присутствовать в каждом скетче.

В данном примере она определяет следующее:

Мы работаем с контактом 13 платы Arduino Nano. И то, что этот контакт мы определяем как **ВЫХОД**. Это означает то, что на выводе 13 платы будет появляться сигнал, который создает сам контроллер.

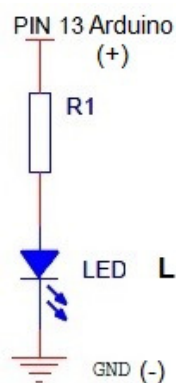
Светодиод на плате Ардуино и на самом деле подсоединен к выводу 13.

Для любого скетча необходима функция **Loop**. Эта функция вызывается сразу после завершения **Setup**, и повторяется снова и снова. Т.е. **Loop** после того, как закончит работу своей команды, сразу же начнет выполнять её снова.

```
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

Внутри цикла функции, размещена команда, которая переключает состояние пина в одно из двух возможных. **HIGH** или **LOW**. **HIGH** соответствует логической **единице** (она же в свою очередь соответствует напряжению питания микроконтроллера +5 Вольт. Состояние вывода **LOW** соответствует логическому **нулю**, это примерно 0 Вольт.

Светодиод **L** (пин 13) подключен в Arduino следующим образом:



Если пин 13 на плате Ардуино установить в состояние **HIGH** (5 Вольт), то светодиод начнет светиться так, как через него потечет электрический ток. Если пин 13 установить в состояние **LOW** (0 Вольт), то светодиод светиться не будет.

Следовательно, оператор **digitalWrite(13, HIGH)** формирует +5В на выходе 13 и зажигает таким образом светодиод **L**.

delay(1000);

Эта команда заставляет микроконтроллер остановиться и ничего не делать целую 1000 миллисекунд. (1000 мс = 1 с ;). Т.е. на выходе 13 в течении 1000 мс будет высокий уровень и светодиод будет светиться.

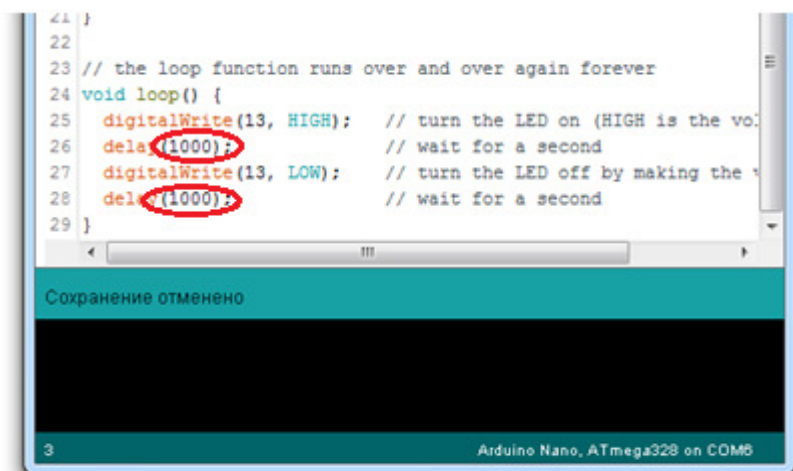
digitalWrite(13, LOW);

Эта команда устанавливает пин 13 в состояние **LOW**. Напряжение на выходе становится близким к нулю. И светодиод «гаснет».

Затем снова выполняется команда паузы на 1000 миллисекунд.

На этом выполнение цикла заканчивается, но функция **loop** снова автоматически вызывается и запускает команду **digitalWrite(13, HIGH)**. И так далее пока подается питание на контроллер.

Теперь мы попробуем изменить скорость мигания светодиода. Для этого надо изменить параметры команды **delay**.



```
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25 digitalWrite(13, HIGH); // turn the LED on (HIGH is the vo
26 delay(1000); // wait for a second
27 digitalWrite(13, LOW); // turn the LED off by making the
28 delay(1000); // wait for a second
29 }
```

Для того, что бы заставить мигать светодиод быстрее изменим значение паузы с 1000 миллисекунд на 500. Загрузим скетч и увидим, что светодиод действительно мигает в два раза быстрее.

Затем загрузите измененную вами программу в плату Arduino, нажав кнопку “Загрузить”.



Если изменения в скетч внесли без ошибок, то после загрузки светодиод на плате Arduino Nano должен мигать с длительностью, установленной в скетче