

## ПРАКТИЧЕСКАЯ МИКРОЭЛЕКТРОНИКА. ЗАНЯТИЕ 2. ВВЕДЕНИЕ В АРДУИНО

Это вторая статья из цикла, в которой мы познакомимся с платформой Arduino (Ардуино), узнаем основные характеристики, настроим драйвера и напишем первую программу Hello, World!

Прежде чем узнать, что такое Ардуино, давайте поговорим о микропроцессорах и их месте в мире компьютеров. Практически в каждом доме, в каждой школе есть персональный компьютер. Но ведь так было далеко не всегда! Наверняка каждого из вас упрекали родители: «в наше время не было компьютеров... вы даже не представляете что это такое – жизнь без компьютеров». И действительно, персональный компьютер появился относительно недавно, в 1970–1980 годах за рубежом, а в России термин ПК обрел массовость в середине 90-х. А что же было до них? Приведу интересную цитату из книги нобелевского лауреата по физике Ричарда Фейнмана «Вы, конечно, шутите, мистер Фейнман», в которой он рассказывает, как создавалась первая в мире атомная бомба: «Мы разработали программу, но у нас пока не было машин для реальной проверки. Поэтому мы посадили в комнату девушек: одна была «умножителем», другая – «слагателем». Еще одна возводила в куб: все, что она делала, – возводила в третью степень число на карточке и отправляла ее следующей девушке». А так как расчетов было много, то и девушек потребовалось изрядное количество. И вот что Фейнман рассказывает об опыте внедрения первых вычислительных ма-

шин IBM: «Оказалось, что скорость, с которой мы теперь были в состоянии вычислять, стала чертовски большой – намного больше, чем при другом способе, когда каждый человек все шаги продельвал сам. Единственная разница состояла в том, что машины IBM не уставали и могли работать в три смены. А вот девушки через некоторое время уставали».

Итак, первыми пользователями электронных вычислительных машин, или компьютеров, были ученые и военные. Благодаря ядерным исследованиям в Лос-Аламосе, США, возникла потребность в суперкомпьютерах, и фирмы смогли эту потребность удовлетворить. Такие компьютеры занимали целые комнаты и этажи, всегда использовалось самое новейшее оборудование. Они были предназначены для решения специфических задач, тех задач, которые можно было хорошо распараллелить. И тогда, и сейчас по этим критериям определяют суперкомпьютеры.

Конечно, для многих фирм суперкомпьютеры были (и остаются) дороги, им нужны высокопроизводительные компьютеры общего назначения (то есть для решения широкого круга задач), которые бы помещались в одну стойку. Такие компьютеры называются мейнфрейм (*mainframe* – основная стойка). Они отличаются от большинства других компьютеров высокой производительностью, надежностью и отказоустойчивостью.

Для более простых вычислений, а также систем клиент-сервер существуют бо-

лее дешевые промышленные компьютеры, называемые серверами. У классического сервера в стойке нет монитора, доступ к нему осуществлялся через различные удаленные терминалы (терминальный доступ).

В 1970–1980 годах появились компьютеры для личного использования, так называемые персональные компьютеры. Они дешевле и компактнее по сравнению с вышеперечисленными компьютерами и обязательно имеют дисплей для вывода информации и средства ввода: клавиатуру, джойстик или сенсорный экран. В наши дни их можно разделить по назначению: настольные компьютеры, игровые приставки к телевизору, ноутбуки, планшетные и карманные компьютеры.

*Микроконтроллер* (рис. 1) – это маленький компьютер на одном кристалле, который совмещает в себе процессор, оперативную память и флеш-память (постоянное запоминающее устройство), генератор тактов (резонатор), а также интерфейс для периферийных устройств. По объему производства в 2006 году 55 % всех выпущенных процессоров – это микроконтроллеры. Примеры области использования МК: промышленность, медицина, транспорт, робототехника, бытовая техника, умный дом, игрушки.

Основные характеристики микроконтроллеров:

- Наличие/отсутствие
  - ОЗУ, ПЗУ,
  - возможности перепрошивки программы,
  - встроенного генератора тактовой частоты,
  - сторожевого таймера,
  - периферии.
- Архитектура:
  - 8, 16 или 32 бит,
  - набор команд.
- Частота процессора.

## ПЛАТФОРМА АРДУИНО

*Ардуино* – это программно-аппаратная платформа для быстрого прототипи-

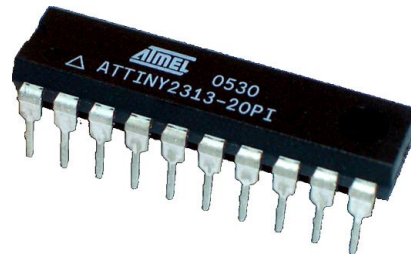


Рис. 1. Микроконтроллер

рования и создания электронных устройств, поддерживаемая сообществом разработчиков по всему миру и пользующаяся огромной популярностью в России и за рубежом. Платформа позволяет общаться и взаимодействовать с окружающим миром с помощью всевозможных датчиков, моторов, актуаторов, других плат. Она была разработана энтузиастами из Италии в 2005 году. Платформа состоит из двух частей: программной и аппаратной.

В качестве программной части выступает кроссплатформенная среда разработки Arduino IDE, которая может запускаться на операционных системах Windows, Linux, Mac OS. С помощью данной среды можно разрабатывать код, компилировать его и прошивать платы. В комплекте поставляются многочисленные примеры, библиотеки и монитор COM-порта.

В качестве аппаратной части выступают различные платы Arduino, эталонной на данный момент является Arduino UNO.

Рассмотрим основные преимущества данной платформы:

**1. Низкий порог входа для новичков.** Могу привести пример из собственного опыта. Для того чтобы запрограммировать микроконтроллер фирмы Microchip, мне потребовалось заказать микроконтроллер в одном городе, программатор в другом, прочитать Data Sheet (техническое описание) объемом 250 страниц, а затем Errata (список ошибок для данного МК). В итоге, чтобы написать простейшую программу, мне потребовался месяц. На Ардуино для аналогичной программы мне потребовалось 10 минут.

2. **Кроссплатформенная среда разработки.** В отличие от многих сред программирования, Arduino не ограничивает свободу выбора ОС.

3. **Отсутствие необходимости в программаторе.** Почти все платы имеют USB разъем.

4. **Наличие большого числа плат.** Существует несколько видов оригинальных плат для разных задач, а также их многочисленные клоны

5. **Переносимость кода.** Написав один раз код для платы Arduino UNO, вы можете перенести его на более мощную плату Arduino MEGA или более слабую Arduino NANO.

6. **Отсутствие необходимости в пайке.** Схемы собираются на беспаячной макетной плате.

7. **Открытый исходный код + открытые чертежи (Open Source + Open Hardware).** Сообщество разработчиков делится своими достижениями: кодом и чертежами.

8. **Наличие САПР (систем автоматизированного проектирования), эмуляторов.** Также с открытым исходным кодом, кроссплатформенные.

9. **Язык программирования C/C++,** один из самых популярных языков программирования.

10. **Наличие большого числа плат-расширений.** С ними Arduino Uno превращается в конструктор. Можно добавить сетевую плату Ethernet, плату Bluetooth, GPS, GSM и даже видеоплату VGA.

И, конечно, огромное сообщество людей во всем мире, которые используют Ардуино в своих проектах, благодаря чему платформа постоянно развивается, происходит обновления среды разработки, совершенствование старых плат и появление новых. Вместе с каждой библиотекой поставляется пример ее использования. Вам не придется начинать что-то делать с нуля, например, писать протокол обмена данными с GPS приемником или со сканером отпечатков пальцев. Все, что вам нужно, – проверить за пару минут работоспособность готовой библиотеки/ус-

тройства и двигаться дальше к вашей цели, не отвлекаясь по мелочам.

## ПЛАТА ARDUINO UNO

Рассмотрим внимательно базовую плату Arduino UNO (рис. 2), построенную на базе МК ATmega 328. Основные элементы подписаны на рисунке. Это 14 цифровых портов ввода/вывода, 6 аналоговых входов, кварцевый резонатор (генератор тактов) частотой 16 МГц, разъем USB, круглый разъем питания, несколько выводов питания, кнопка перезагрузки и сам МК.

Микроконтроллер – это «мозг» Ардуино, он выполняет программу, хранит промежуточные результаты в своей памяти, совершает все логические и арифметические операции, дает команды портам ввода/вывода... Кварцевый резонатор заставляет работать МК на определенной частоте, без него микроконтроллер просто не сможет работать. По сути, это «сердце» Ардуино, бьющееся 16 миллионов раз в секунду. Порты ввода/вывода – это «руки», «ноги», органы зрения, осязания, обоняния. Через них Arduino общается с окружающим миром: с предметами, людьми, другими платами, выходит в сеть и т. д. С помощью USB порта плата общается с компьютером.

5 разъемов питания слева направо:

**Reset** (сброс программы)

**3V3** – напряжение на данном выводе +3.3 В, генерируемое встроенным регулятором на плате. Максимальное потребление тока 50мА,

**5V** – напряжение на данном выводе +5 В, генерируемое встроенным регулятором на плате,

**GND** – заземление,

**Vin** – используется для подачи питания от внешнего источника в отсутствии 5 В от разъема USB или другого источника питания.

Внешнее питание (не USB) может подаваться через разъем Vin или круглый разъем с центральным плюсом (рис. 3). Платформа может работать при внешнем

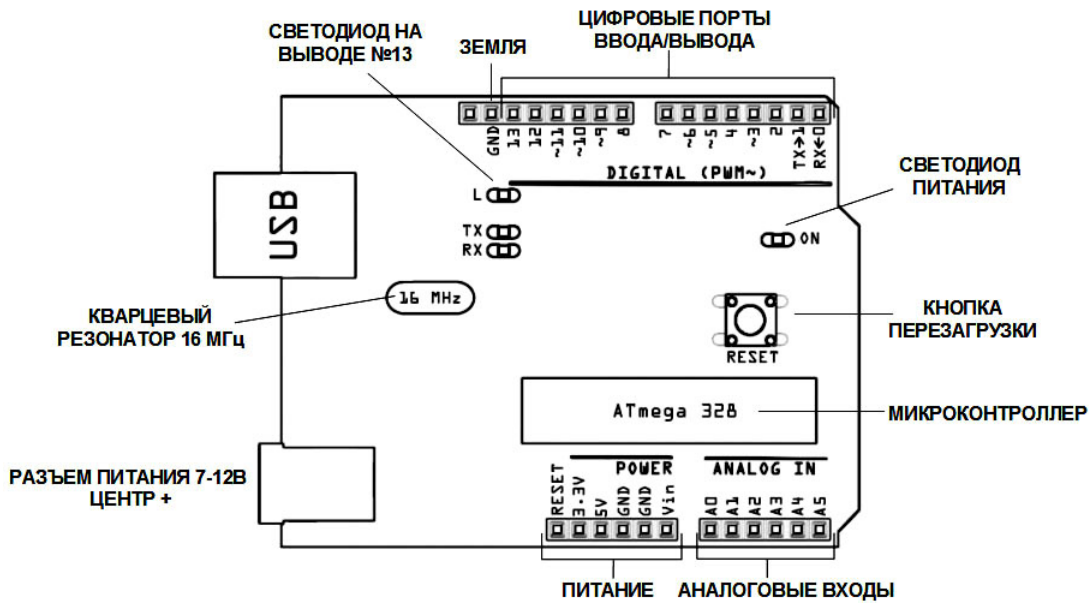


Рис. 2. Плата Arduino UNO

напряжении от 6 до 20 В, но рекомендуется использовать напряжение в диапазоне 7–12 В для предотвращения перегрева или нестабильной работы.

### ЦИФРОВЫЕ ВХОДЫ И ВЫХОДЫ

Каждый из 14 цифровых портов ввода/вывода (рис. 2) может быть как порт ввода или порт вывода с помощью функции `pinMode( )` с параметром `INPUT` или `OUTPUT`.

Чтобы сконфигурировать порт номер 13 как порт вывода, необходимо вызвать функцию `pinMode(13, OUTPUT)`. Чтобы подать на вывод номер 13 напряжение 5 В, нужно вызвать функцию `digitalWrite(13, HIGH)`, или, что то же самое, `digitalWrite(13, 1)`. Чтобы подать на вывод номер 13 напряжение 0 В, нужно вызвать функцию `digitalWrite(13, LOW)`



Рис. 3. Разъем питания (центр плюс)

или `digitalWrite(13, 0)`. Согласно спецификации, ток не должен превышать 40 мА. Этого достаточно для питания светодиода или большинства датчиков.

По умолчанию все порты Arduino сконфигурированы как порты ввода. Это значит, что, если вы хотите считать данные с порта номер 13, нет нужды вызывать функцию `pinMode(13, INPUT)`. Чтобы считать напряжение с порта вывода номер 13, нужно вызвать функцию `digitalRead(13)`, которая возвращает число 0 или 1, в зависимости от входного напряжения, поданного в порт 13. Соответствие напряжения и логических уровней происходит следующим образом (рис. 4).

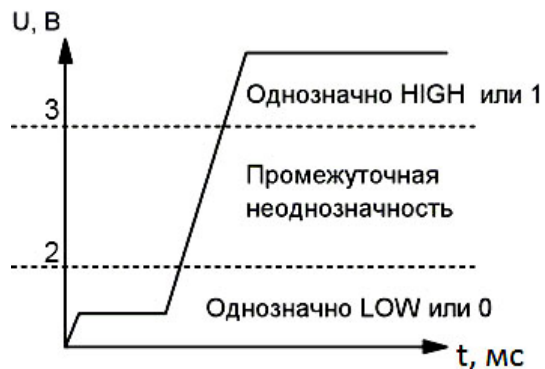


Рис. 4. Логические уровни напряжений



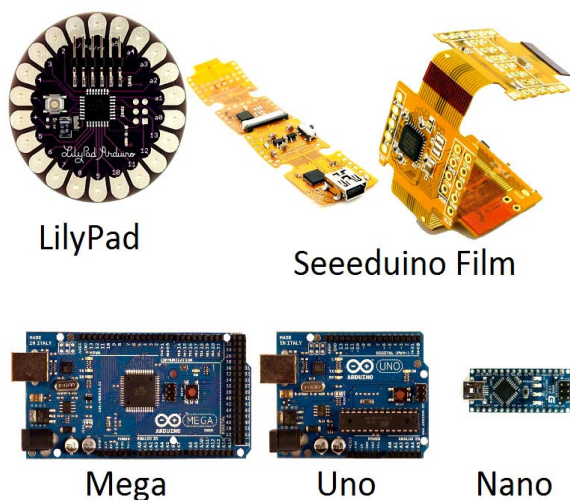


Рис. 5. Платы Arduino

Поэтому для цифровых портов ввода следует избегать напряжений между 2 и 3 В.

### АНАЛОГОВЫЕ ВХОДЫ И ВЫХОДЫ

Итак, мы уже научились подавать и считывать напряжение, равное 0 или 5 В. Но иногда требуется получить промежуточные значения напряжения, например, чтобы плавно менять обороты мотора или считывать значение с аналогового датчика освещенности, расстояния и т. д. Подробнее об этих портах читайте в следующих выпусках.

### РАЗЛИЧНЫЕ ПЛАТЫ ARDUINO

Существует множество разных Arduino-совместимых плат. Совместимость означает переносимость кода. Написав код один раз для платы Arduino

UNO, вы можете потом использовать его на других платах, учитывая ограничения конкретной платы. Платы различаются количеством встроенной памяти, частотой процессора, количеством цифровых портов ввода/вывода, аналоговых портов, размерами. Есть даже гибкие носимые платы, так называемая носимая электроника (рис. 5).

### ЧТО ВАМ ПОНАДОБИТСЯ

Доступ в Интернет, чтобы скачать среду разработки, примеры программ и библиотеки.

Также вам понадобится стол или любая другая плоская поверхность. Помните, что вы имеете дело с электрическими схемами, поэтому, если поверхность стола металлическая, накройте его материалом, не проводящим ток, например листами бумаги или тканью. Желательно, но не обязательно иметь плоскогубцы и ножницы.

Обязательно понадобится компьютер с одной из следующих ОС: Windows/Linux/Mac OS; сама плата Arduino Uno (или ее аналоги), USB-кабель. И самое главное – ваше желание и энтузиазм.

Кроме того, в начале каждого проекта будут перечислены дополнительные детали, необходимые для данного проекта. Желательно сразу приобрести набор для проектирования на Arduino.

### НАБОР ДЛЯ ПРОЕКТИРОВАНИЯ НА ARDUINO

Одно из преимуществ Ардуино – возможность создания работающего прото-

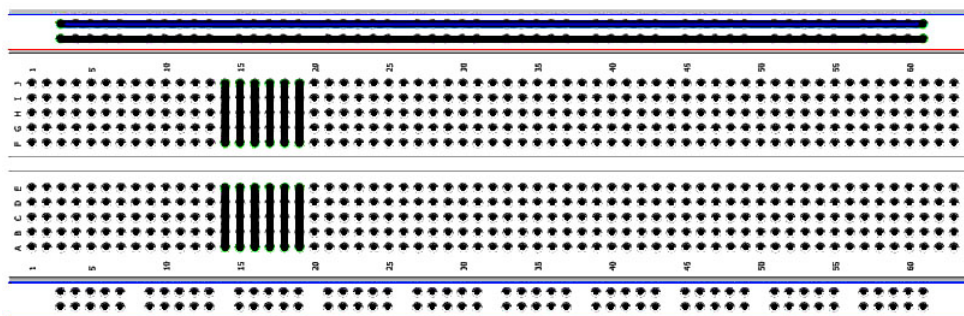


Рис. 6. Макетная плата и способ соединения отверстий внутри нее

типа устройства без пайки. Для этого требуется беспаячная макетная плата и комплект проводов-перемычек. Очень часто продаются готовые наборы, состоящие из платы, проводов, резисторов, диодов, конденсаторов, транзисторов, пьезодинамика и т. д. Многие из этих деталей можно найти в любом магазине радиоэлектроники, однако плату Arduino найти сложнее. Обзор магазинов, торгующих данными компонентами, находится по адресу: <http://техблог.рф/ардуино/магазины>.

Макетная плата состоит из рядов отверстий, которые соединены между собой горизонтально и вертикально, как на рис. 6.

### ПРИСТУПАЯ К РАБОТЕ

В данном разделе подразумевается, что у вас есть компьютер с ОС Windows XP или Vista или 7.

1. Скачайте последнюю версию Arduino IDE с сайта <http://arduino.cc>, или <http://arduino.ru>, или по ссылке <http://arduino.googlecode.com/files/arduino-0022.zip>. Распакуйте архив в папку, например C:\arduino.

2. Подключите Arduino к компьютеру при помощи USB-шнура.

3. Установите драйвер виртуального COM порта. Драйвера находятся в папке «C:\arduino\drivers» и «C:\arduino\drivers\FTDI USB Drivers» (см. рис. 7, 8).

4. Определите номер COM-порта в диспетчере устройств (рис. 9). При желании можно сменить номер COM-порта (рис. 10).

5. Запустите среду Arduino IDE по адресу C:\arduino\arduino.exe (рис. 11).

6. Выберите нужный COM-порт (рис. 12).

7. Выберите нужную плату (рис. 13).

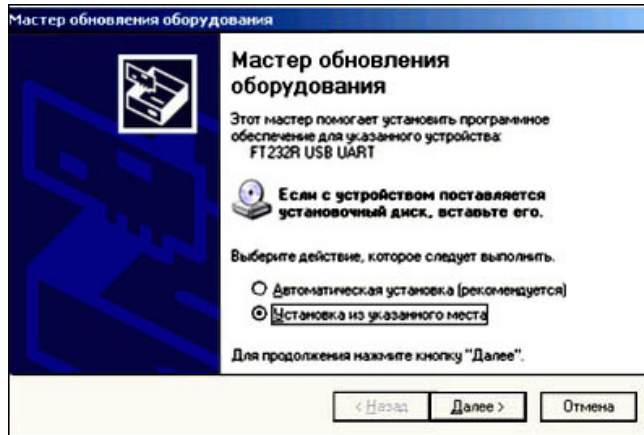


Рис. 7. Установка драйвера из указанного места

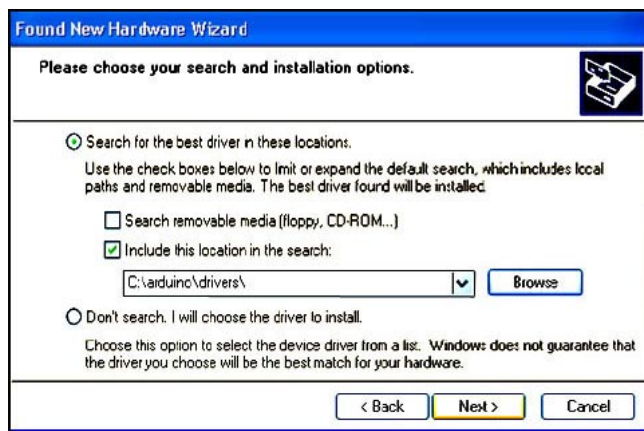


Рис. 8. Путь к драйверам

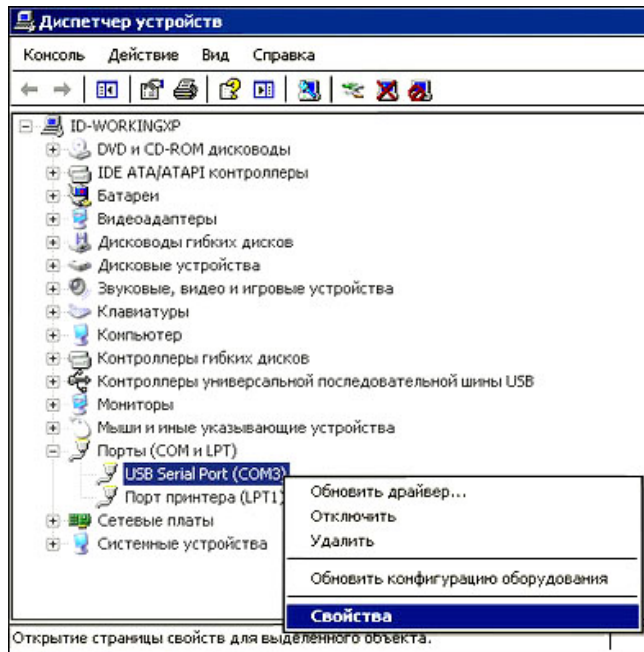


Рис. 9. Определение номера COM-порта

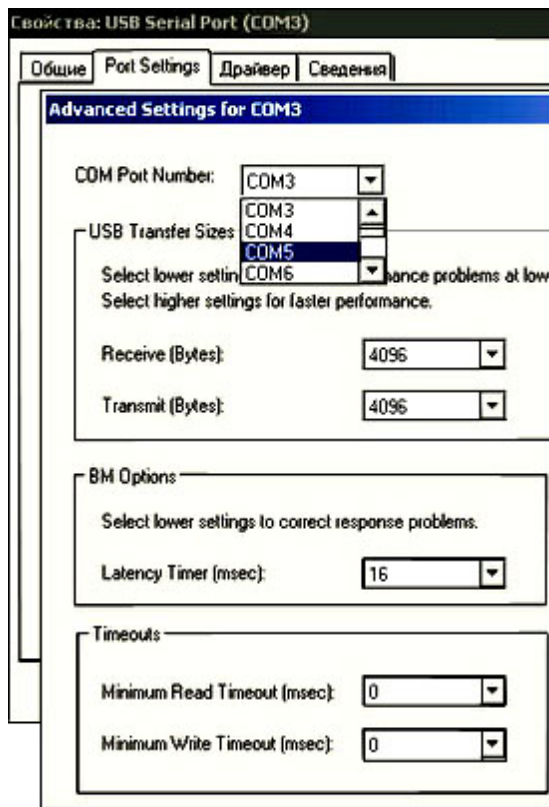


Рис. 10. Смена номера COM-порта

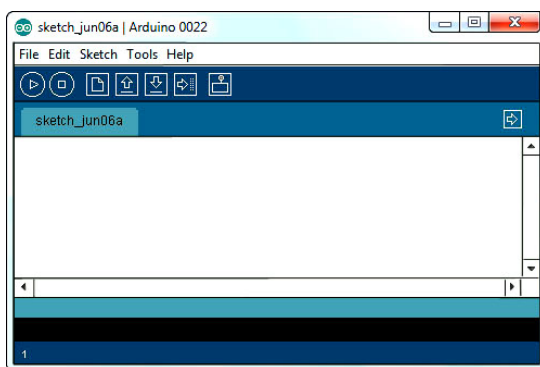


Рис. 11. Среда разработки Arduino IDE

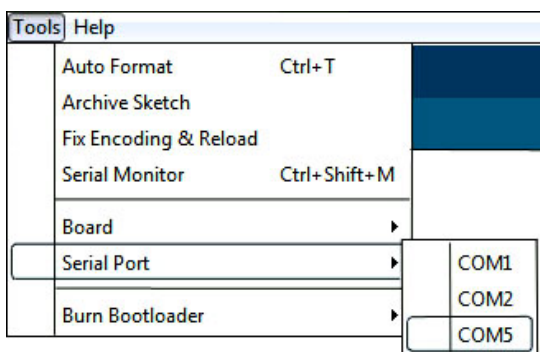


Рис. 12. Выбор COM-порта в среде Ардуино IDE

8. Напишите скетч. Скетчем называется программа (прошивка) для платы Arduino.

9. Скомпилируйте скетч и прошейте им плату.

#### ПАНЕЛЬ ИНСТРУМЕНТОВ

Панель инструментов (рис. 14) содержит следующие кнопки:

**Verify/Compile** – компилирует/проверяет код на ошибки,

**Stop** – останавливает Serial Monitor и компиляцию,

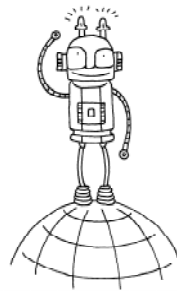
**New** – создает новый пустой скетч (прошивку),

**Open** – показывает список скетчей,

**Save** – сохраняет текущий скетч,

**Upload** – прошивает текущий скетч в Arduino,

**Serial Monitor** – отображает данные с последовательной шины Arduino (COM-порт).



#### ПРОЕКТ 1 «HELLO, WORLD!»

По традиции в начале знакомства нужно поздравиться, и, чтобы никого не обидеть, давайте поздравимся сразу со всем миром. Программы вида «Hello, World!» позволяют новичку убедиться что все установлено и настроено правильно, оценить сложность процесса написания программ на конкретном языке программирования с использованием конкретных библиотек. Кроме того, простой пример всегда можно усложнить.

Нельзя недооценивать данный тип программ. В некоторых фирмах соискателю в качестве тестового задания предлагают написать «Hello, World» – это хороший способ показать себя, добавив в проект инсталлятор, паттерны, сделав каркас для консольного или оконного приложения, скомпилировать под разные платформы и т. д.

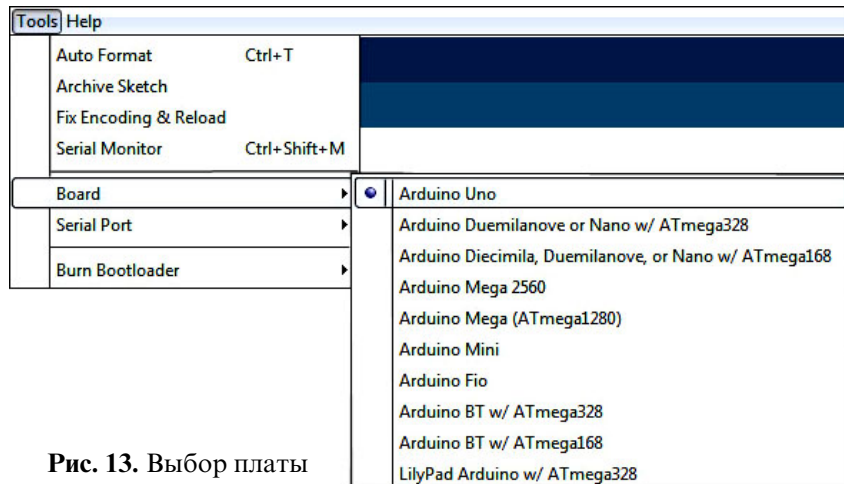
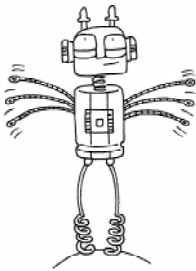


Рис. 13. Выбор платы

В мире микроконтроллеров программа «Hello, World!» означает помигать светодиодом.

**«ПРЕПОЛЕТНАЯ»  
ПОДГОТОВКА**

1. Проверить наличие платы Arduino UNO или ее аналога, USB-кабеля.
2. (Опционально) Проверить наличие макетной платы, соединительных проводов, светодиода и резистора на 150–300 Ом.

**«ПОЛЕТ»**

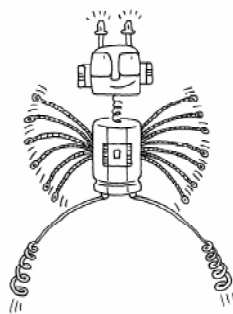
Как вы помните из прошлой статьи, у светодиода бывают несколько характеристик, основные из них – типовой (рабочий) ток и типовое падение напряжения. Если ток ничем не ограничить, он превысит допустимое значение и светодиод сгорит. Зная разность потенциалов у источника тока и падение напряжения на



Рис. 14. Панель инструментов

светодиоде, можно рассчитать требуемое падение напряжения (разность потенциалов) у резистора. Так как ток во всей последовательной цепи постоянен, то рассчитываем сопротивление по закону Ома (рис. 15).

То есть мы можем выбрать сопротивление 150 Ом или больше (но не меньше). Если у вас нет макетной платы, светодиода и резистора, то вы все равно сможете поприветствовать мир, помигав светодиодом на 13 выводе Arduino UNO (рис. 2). Ведь как раз для такого случая разработчики предусмотрительно добавили светодиод и резистор для 13 вывода.



Пример:

$$U_{\text{светодиода}} = 2\text{В}, \quad I_{\text{светодиода}} = 20\text{мА},$$

$$U_{\text{резистора}} = U_{\text{питания}} - U_{\text{светодиода}} = 5\text{В} - 2\text{В} = 3\text{В},$$

$$R_{\text{резистора}} = \frac{U_{\text{резистора}}}{I_{\text{светодиода}}} = \frac{3\text{В}}{0.02\text{А}} = 150\text{ Ом}.$$

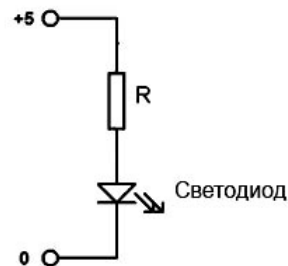


Рис. 15. Резистор как токоограничитель для светодиода



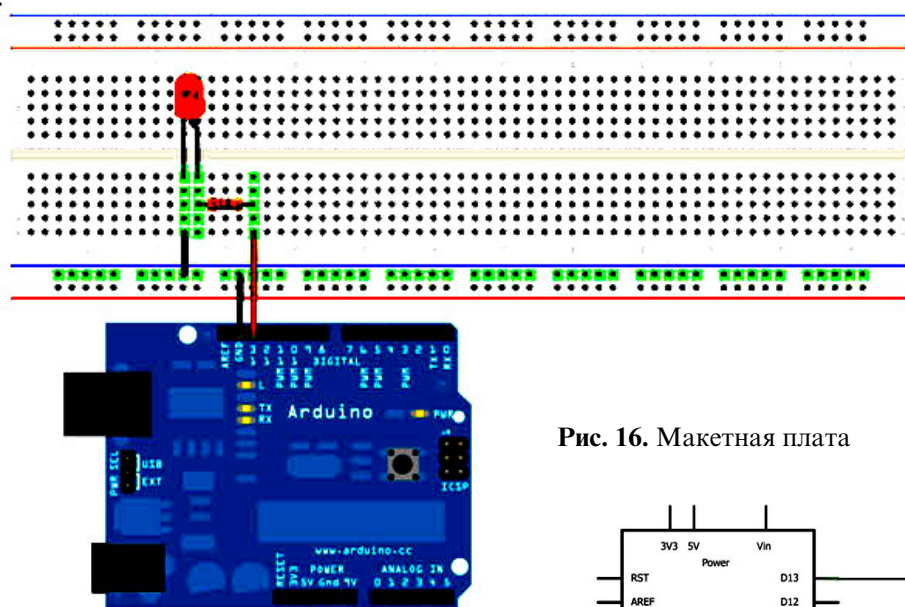


Рис. 16. Макетная плата

Приветствовать мир мы будем так: сначала включим светодиод на 1 секунду, потом выключим на 1 секунду, снова включим, выключим и так далее. Принципиальная схема представлена на рис. 16, 17.

Программа будет очень простой. В функции `setup` назначим вывод номер 13 как выход, затем включаем и выключаем светодиод, подавая на 13 вывод 0 или 5 В в бесконечном цикле (листинг 1).

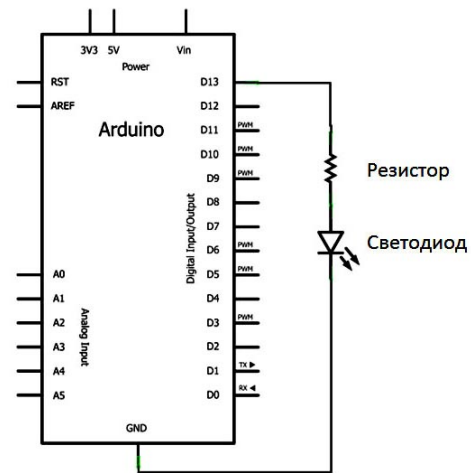


Рис. 17. Принципиальная схема

### Листинг 1

```

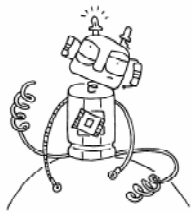
/*
  Blink.
  Включает светодиод на секунду, затем выключает на секунду в цикле.
  */

// Инициализация. Метод вызывается только 1 раз, когда стартует скетч, после подачи питания
// или после сброса платы. Используется для инициализации переменных, определения режимов
// работы выводов, запуска используемых библиотек
void setup() {
  pinMode(13, OUTPUT); // назначить 13-й вывод как выход
}

// Бесконечный цикл. После выполнения setup(), данный метод вызывается каждый раз после
// завершения последнего оператора в цикле
void loop() {
  digitalWrite(13, HIGH); // включить светодиод на 13 выводе
  delay(1000);           // подождать 1 секунду = 1000 миллисекунд
  digitalWrite(13, LOW); // выключить светодиод на 13 выводе
  delay(1000);          // подождать 1 секунду = 1000 миллисекунд
}

```

После того как вы напишете код, нажмите кнопку «Upload» (рис. 14) и дождитесь завершения прошивки. Если возникла ошибка компиляции, исправьте ошибку. Если ошибка произошла во время прошивки (статус Uploading) – проверьте USB кабель, номер COM-порта и тип платы.



### РАЗБОР «ПОЛЕТА»

Язык программирования основан на C/C++. Это один из самых популярных языков программирования, он прост в освоении, и если вы уже сталкивались с ним, то вам не составит труда написать программу на Arduino (рис. 18).

Любая, даже самая простая программа на Arduino состоит как минимум из двух обязательных функций: `setup()` и `loop()`.

Функция `setup` вызывается один раз, когда стартует скетч (при подаче напряжения или нажатии на кнопку «Reset» или

### Листинг 2

```

/*
  Самая простая программа на Arduino
*/
void setup() {
}
void loop() {
}
    
```

после успешной прошивки). Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т. д.

После завершения функции `setup()` вызывается функция `loop()`, которая делает то, что означает её название: крутится в бесконечном цикле, позволяя вашей программе совершать вычисления и взаимодействовать с окружающей средой с помощью датчиков, моторов, актуаторов, сетей и т. д. После выполнения последнего оператора внутри `loop()` начинает выполняться первый оператор. Иначе говоря, программа никогда не завершается (см. листинг 2).

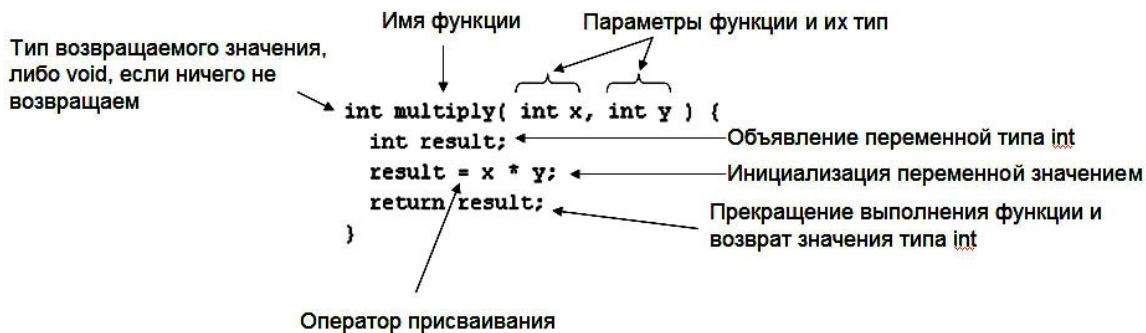


Рис. 18. Сигнатура и тело функции

Ярков Константин Евгеньевич,  
разработчик фирмы SoftDev SPb.

© Наши авторы, 2011.  
Our authors, 2011.