About the Tutorial

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Audience

This tutorial is intended for enthusiastic students or hobbyists. With Arduino, one can get to know the basics of micro-controllers and sensors very quickly and can start building prototype with very little investment.

Prerequisites

Before you start proceeding with this tutorial, we assume that you are already familiar with the basics of C and C++. If you are not well aware of these concepts, then we will suggest you go through our short tutorials on C and C++. A basic understanding of microcontrollers and electronics is also expected.

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com
# Table of Contents

About the Tutorial ........................................................................................................... i
Audience ............................................................................................................................ i
Prerequisites ..................................................................................................................... i
Copyright & Disclaimer .................................................................................................. i
Table of Contents ........................................................................................................... ii

**ARDUINO — BASICS** ................................................................................................ 1

1. Arduino — Overview ................................................................................................... 2
   Board Types .................................................................................................................. 3

2. Arduino — Board Description .................................................................................... 6

3. Arduino — Installation ............................................................................................... 9

4. Arduino — Program Structure .................................................................................... 17

5. Arduino — Data Types ............................................................................................... 19
   void............................................................................................................................. 19
   Boolean....................................................................................................................... 19
   Char ............................................................................................................................ 20
   unsigned char........................................................................................................... 21
   byte............................................................................................................................. 21
   int 21
   Unsigned int ............................................................................................................. 21
   Word............................................................................................................................ 21
   Long ............................................................................................................................. 21
   unsigned long .......................................................................................................... 22
   short ............................................................................................................................. 22
   float ............................................................................................................................ 22
   double......................................................................................................................... 23

6. Arduino — Variables & Constants ............................................................................. 24
   What is Variable Scope? ............................................................................................ 24

7. Arduino — Operators .................................................................................................. 26
   Arithmetic Operators................................................................................................. 26
   Comparison Operators .............................................................................................. 27
   Boolean Operators ..................................................................................................... 29
   Bitwise Operators ...................................................................................................... 30
   Compound Operators ............................................................................................... 31

8. Arduino — Control Statements .................................................................................. 33
   if statement ............................................................................................................... 34
   If ...else statement .................................................................................................. 35
   if...else if ...else statement ..................................................................................... 37
   Switch Case Statement ............................................................................................ 39
   Conditional Operator ? : ......................................................................................... 41
   Rules of Conditional Operator ............................................................................... 41

9. Arduino — Loops ......................................................................................................... 42
while loop ................................................................................................................................. 42
do...while loop .......................................................................................................................... 43
for loop .................................................................................................................................. 44
Nested Loop ............................................................................................................................ 45
Infinite loop ............................................................................................................................... 45

10. Arduino - Functions ................................................................................................................. 47
    Function Declaration ........................................................................................................... 48

11. Arduino – Strings .................................................................................................................. 51
    String Character Arrays ..................................................................................................... 51
    Manipulating String Arrays ............................................................................................... 52
    Functions to Manipulate String Arrays .............................................................................. 54
    Array Bounds ....................................................................................................................... 58

12. Arduino – String Object ........................................................................................................ 59
    What is an Object? ................................................................................................................. 59
    When to Use a String Object ............................................................................................... 61

13. Arduino – Time ..................................................................................................................... 62
    delay() function .................................................................................................................... 62
    delayMicroseconds() function .......................................................................................... 63
    millis() function .................................................................................................................. 63
    micros() function ................................................................................................................ 64

14. Arduino – Arrays .................................................................................................................. 66
    Declaring Arrays .................................................................................................................. 67
    Examples Using Arrays ........................................................................................................ 67
    Arduino – Passing Arrays to Functions .............................................................................. 70
    Multidimensional Arrays ...................................................................................................... 73

ARDUINO – FUNCTION LIBRARIES ......................................................................................... 78

15. Arduino – I/O Functions ....................................................................................................... 79
    Pins Configured as INPUT ................................................................................................... 79
    Pull-up Resistors .................................................................................................................. 79
    Pins Configured as OUTPUT ............................................................................................... 80
    pinMode() Function .......................................................................................................... 80
    digitalWrite() Function .................................................................................................... 81
    analogRead( ) function ....................................................................................................... 82

16. Arduino – Advanced I/O Function ..................................................................................... 84
    analogReference() Function .............................................................................................. 84

17. Arduino – Character Functions ........................................................................................... 86
    Examples ............................................................................................................................... 87

18. Arduino – Math Library ....................................................................................................... 93
    Library Macros ..................................................................................................................... 93
    Library Functions ................................................................................................................ 95
    Example ............................................................................................................................... 99

19. Arduino – Trigonometric Functions .................................................................................... 101
ARDUINO ─ SENSORS ........................................................................................................... 161

35. Arduino – Humidity Sensor .............................................................................................. 162

36. Arduino – Temperature Sensor ...................................................................................... 168

37. Arduino – Water Detector / Sensor .................................................................................. 171

38. Arduino – PIR Sensor ...................................................................................................... 174

39. Arduino – Ultrasonic Sensor ............................................................................................ 179

40. Arduino – Connecting Switch ......................................................................................... 183

ARDUINO ─ MOTOR CONTROL ............................................................................................ 187

41. Arduino – DC Motor ......................................................................................................... 188
   Motor Speed Control .......................................................................................................... 190
   Spin Direction Control ...................................................................................................... 192

42. Arduino – Servo Motor .................................................................................................... 196

43. Arduino – Stepper Motor ................................................................................................ 201

ARDUINO AND SOUND ....................................................................................................... 205

44. Arduino – Tone Library .................................................................................................... 206

45. Arduino – Wireless Communication .................................................................................. 212

46. Arduino – Network Communication .................................................................................. 217
Arduino – Basics
1. ARDUINO – OVERVIEW

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are:

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.

- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

- Finally, Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package.
Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.
Here is a list of different Arduino boards available.

**Arduino boards based on ATMEGA328 microcontroller**

<table>
<thead>
<tr>
<th>Board Name</th>
<th>Operating Volt</th>
<th>Clock Speed</th>
<th>Digital i/o</th>
<th>Analog Inputs</th>
<th>PWM</th>
<th>UART</th>
<th>Programming Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno R3</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>USB via ATMega16U2</td>
</tr>
<tr>
<td>Arduino Uno R3 SMD</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>USB via ATMega16U2</td>
</tr>
<tr>
<td>Red Board</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>USB via FTDI</td>
</tr>
<tr>
<td>Arduino Pro 3.3v/8 MHz</td>
<td>3.3V</td>
<td>8 MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino Pro 5V/16MHz</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino mini 05</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino Pro mini 3.3v/8mhz</td>
<td>3.3V</td>
<td>8 MHz</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino Pro mini 5v/16mhz</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino Ethernet</td>
<td>5V</td>
<td>16MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>Arduino Fio</td>
<td>3.3V</td>
<td>8MHz</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>LilyPad Arduino 328 main board</td>
<td>3.3V</td>
<td>8MHz</td>
<td>14</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>FTDI- Compatible Header</td>
</tr>
<tr>
<td>LilyPad Arduino simply board</td>
<td>3.3V</td>
<td>8MHz</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>FTDI- Compatible Header</td>
</tr>
</tbody>
</table>

**Arduino boards based on ATMEGA32u4 microcontroller**

<table>
<thead>
<tr>
<th>Board Name</th>
<th>Operating Volt</th>
<th>Clock Speed</th>
<th>Digital i/o</th>
<th>Analog Inputs</th>
<th>PWM</th>
<th>UART</th>
<th>Programming Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>TutorialsPoint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9
### Arduino boards based on ATMEGA2560 microcontroller

<table>
<thead>
<tr>
<th>Board Name</th>
<th>Operating Volt</th>
<th>Clock Speed</th>
<th>Digital i/o</th>
<th>Analog Inputs</th>
<th>PWM</th>
<th>UART</th>
<th>Programming Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Mega 2560 R3</td>
<td>5V</td>
<td>16MHz</td>
<td>54</td>
<td>16</td>
<td>14</td>
<td>4</td>
<td>USB via ATMega16U2</td>
</tr>
<tr>
<td>Mega Pro 3.3V</td>
<td>3.3V</td>
<td>8MHz</td>
<td>54</td>
<td>16</td>
<td>14</td>
<td>4</td>
<td>FTDI-Compatible Header</td>
</tr>
<tr>
<td>Mega Pro 5V</td>
<td>5V</td>
<td>16MHz</td>
<td>54</td>
<td>16</td>
<td>14</td>
<td>4</td>
<td>FTDI-Compatible Header</td>
</tr>
<tr>
<td>Mega Pro Mini 3.3V</td>
<td>3.3V</td>
<td>8MHz</td>
<td>54</td>
<td>16</td>
<td>14</td>
<td>4</td>
<td>FTDI-Compatible Header</td>
</tr>
</tbody>
</table>

### Arduino boards based on AT91SAM3X8E microcontroller

<table>
<thead>
<tr>
<th>Board Name</th>
<th>Operating Volt</th>
<th>Clock Speed</th>
<th>Digital i/o</th>
<th>Analog Inputs</th>
<th>PWM</th>
<th>UART</th>
<th>Programming Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Due</td>
<td>3.3V</td>
<td>84MHz</td>
<td>54</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>USB native</td>
</tr>
</tbody>
</table>
In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.
Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).

2 Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3 Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4 Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5,17 Arduino Reset

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7,8,9 Pins (3.3, 5, GND, Vin)

- 3.3V (6): Supply 3.3 output volt
- 5V (7): Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND (8)(Ground): There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9): This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
**Analog pins**

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

**Main microcontroller**

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEG Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

**ICSP pin**

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

**Power LED indicator**

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

**TX and RX LEDs**

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

**Digital I / O**

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic
values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.

**AREF**

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.
After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

**Step 1:** First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

![Standard USB Cable](image1.jpg)

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

![A to Mini-B Cable](image2.jpg)

**Step 2:** Download Arduino IDE Software.
You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

![Opening arduino-nightly-windows.zip]

You have chosen to open:

arduino-nightly-windows.zip
which is: WinRAR ZIP archive (148 MB)
from: https://downloads.arduino.cc

What should Firefox do with this file?

- [ ] Open with WinRAR archiver (default)
- [ ] Save File
- [ ] Do this automatically for files like this from now on.

**Step 3: Power up your board.**

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

**Step 4: Launch Arduino IDE.**

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.
Step 5: Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.
To create a new project, select File --> New.

To open an existing project example, select File -> Example -> Basics -> Blink.
Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

**Step 6: Select your Arduino board.**

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board.