Arduino Reading

Getting Started with Arduino

Learn Arduino

- Read an introduction on what is Arduino and why you'd want to use it.
- What is the Arduino Software (IDE) and how do I change the default language?
- Libraries: Using and installing Arduino Libraries.
- Cores: Need to add a new board to your Arduino Software? Install the relate core and manage it.
- Troubleshooting: Advice on what to do if things don't work.

For a complete list of Guides visit the Foundations section, where you will find in-depth knowledge about the principles and techniques behind the Arduino platform.
Making the Arduino StarterKit projects and reading the book 'Getting Started with Arduino' are great ways to start learning and tinkering with coding and electronics.

Install the Arduino Software
Step-by-step instructions for setting up the Arduino Software (IDE) on your computer and connecting it to an Arduino Uno, Mega2560, Duemilanove, Mega, or Diecimila.
Click on the list on the side if you want to use another Arduino board.

- Windows
- Mac OS X
- Linux (on the Playground wiki)

What is Arduino?

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based onWiring), and the Arduino Software (IDE), based on Processing.
Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. Thesoftware, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published

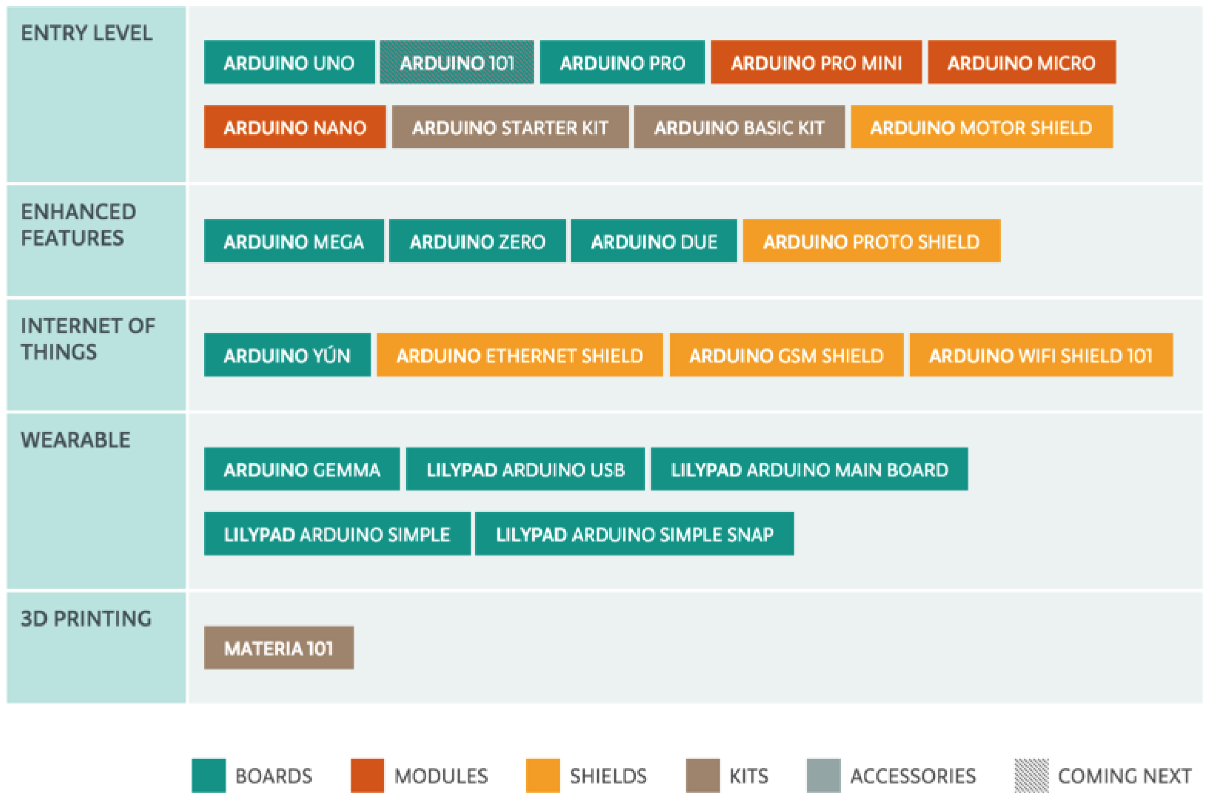
under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

How do I use Arduino?

See the getting started guide.

Arduino Products

Browse the full range of official Arduino products, including Boards, Modules (a smaller form-factor of classic boards), Shields (elements that can be plugged onto a board to give it extra features), and Kits. If you need more info you can compare the specs of each board here.
If you are wondering if your Arduino board is authentic you can learn how to spot a counterfeit board here.

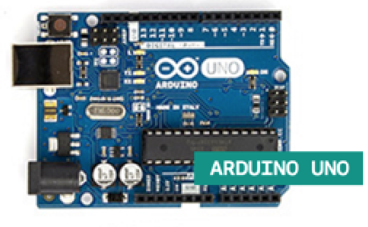


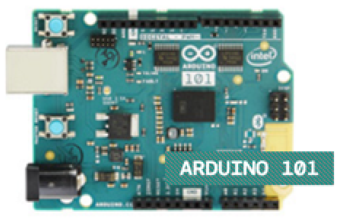
ENTRY LEVEL	ARDUINO UNO, ARDUINO 101, ARDUINO PRO, ARDUINO PRO MINI, ARDUINO MICRO, ARDUINO NANO, ARDUINO STARTER KIT, ARDUINO BASIC KIT, ARDUINO MOTOR SHIELD
ENHANCED FEATURES	ARDUINO MEGA, ARDUINO ZERO, ARDUINO DUE, ARDUINO PROTO SHIELD
INTERNET OF THINGS	ARDUINO YÚN, ARDUINO ETHERNET SHIELD, ARDUINO GSM SHIELD, ARDUINO WIFI SHIELD 101
WEARABLE	ARDUINO GEMMA, LILYPAD ARDUINO USB, LILYPAD ARDUINO MAIN BOARD, LILYPAD ARDUINO SIMPLE, LILYPAD ARDUINO SIMPLE SNAP
3D PRINTING	MATERIA 101

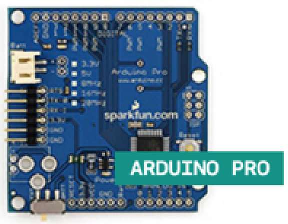
🟩 BOARDS 🟥 MODULES 🟧 SHIELDS 🟫 KITS ⬜ ACCESSORIES ▨ COMING NEXT

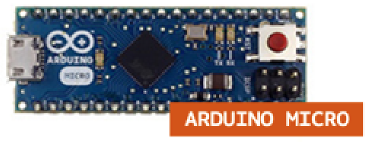
Entry Level

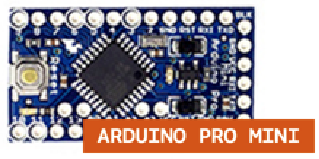
Get started with Arduino using Entry Level products: easy to use and ready to power your first creative projects. These boards and modules are the best to start learning and tinkering with electronics and coding. The StarterKit includes a book with 15 tutorials that will walk your thought the basics up to complex projects.

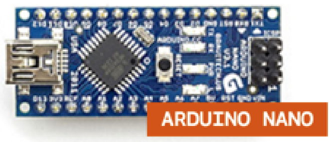


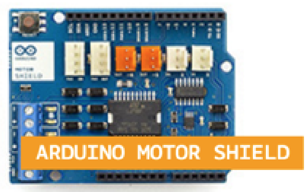






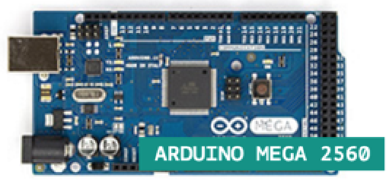


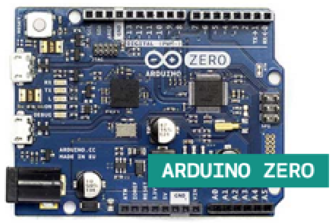


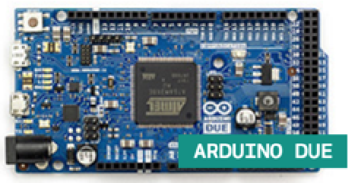


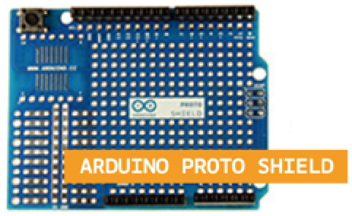
Enhanced Features

Experience the excitement of more complex projects choosing one of the boards with advanced functionalities, or faster performances.







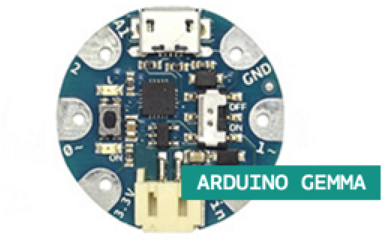


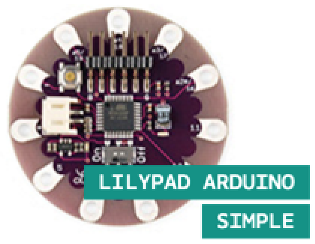
Internet of Things

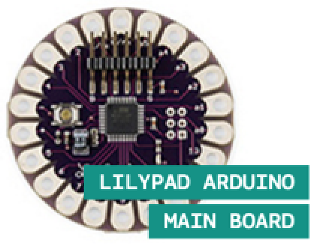
Make connected devices easily with one of these IoT products and open your creativity with the opportunities of the world wide web.

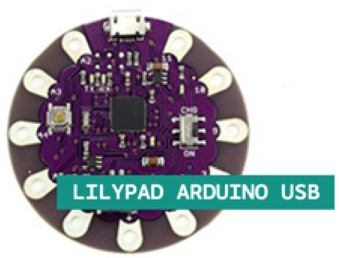
Wearable

Add smartness to your soft projects and discover the magic of sewing the power of electronics directly to textiles.



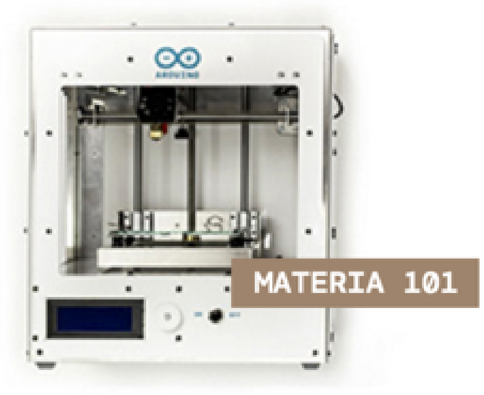






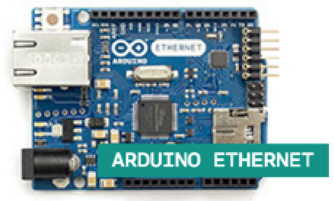
3D Printing

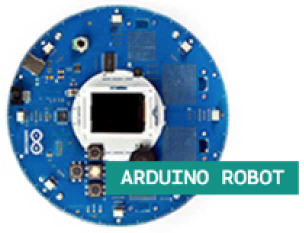
The Arduino approach to 3D printing is represented by Materia 101, a printer that allows you to start experimenting with this amazing technology in the easiest way.

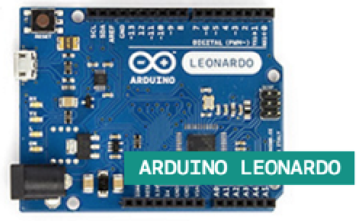


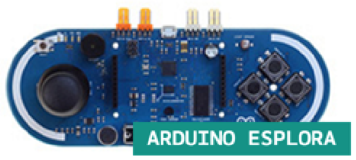
Retired

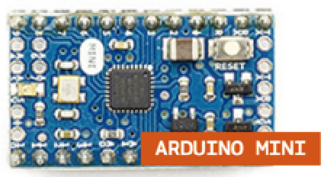
Explore the history of Arduino with a journey through all the boards, accessories, shield, kits and documentation released since 2006.

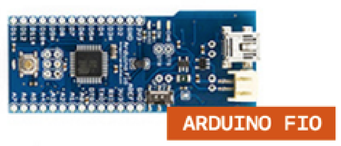


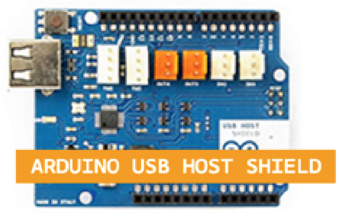


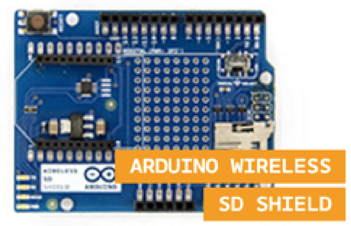


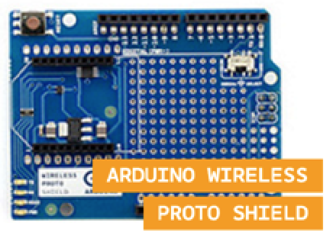


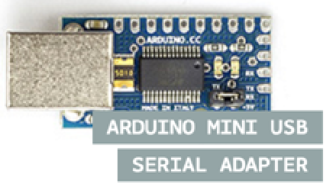


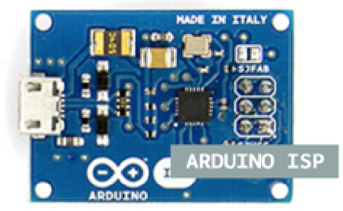


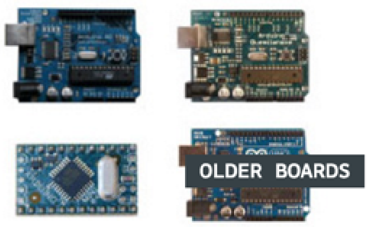












Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and*functions*.

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP
- LED_BUILTIN
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Due & Zero only

- analogReadResolution()
- analogWriteResolution()

Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()

- // (single line comment)
- /* */ (multi-line comment)
- #define
- #include

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)
- ! (not)

Pointer Access Operators

- * dereference operator
- & reference operator

Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound

- word
- long
- unsigned long
- short
- float
- double
- string - char array
- String - object
- array

Conversion

- char()
- byte()
- int()
- word()
- long()
- float()

Variable Scope & Qualifiers

- variable scope
- static
- volatile
- const

Utilities

- sizeof()
- PROGMEM

- pulseIn()

Time

- millis()
- micros()
- delay()
- delayMicroseconds()

Math

- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry

- sin()
- cos()
- tan()

Characters

- isAlphaNumeric()
- isAlpha()
- isAscii()
- isWhitespace()
- isControl()
- isDigit()
- isGraph()
- isLowerCase()
- isPrintable()
- isPunct()
- isSpace()
- isUpperCase()
- isHexadecimalDigit()

Random Numbers

- randomSeed()
- random()

Bits and Bytes

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

addition)

- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- %= (compound modulo)
- &= (compound bitwise and)
- |= (compound bitwise or)

External Interrupts

- attachInterrupt()
- detachInterrupt()

Interrupts

- interrupts()
- noInterrupts()

Communication

- Serial
- Stream

USB (32u4 based boards and Due/Zero only)

- Keyboard
- Mouse