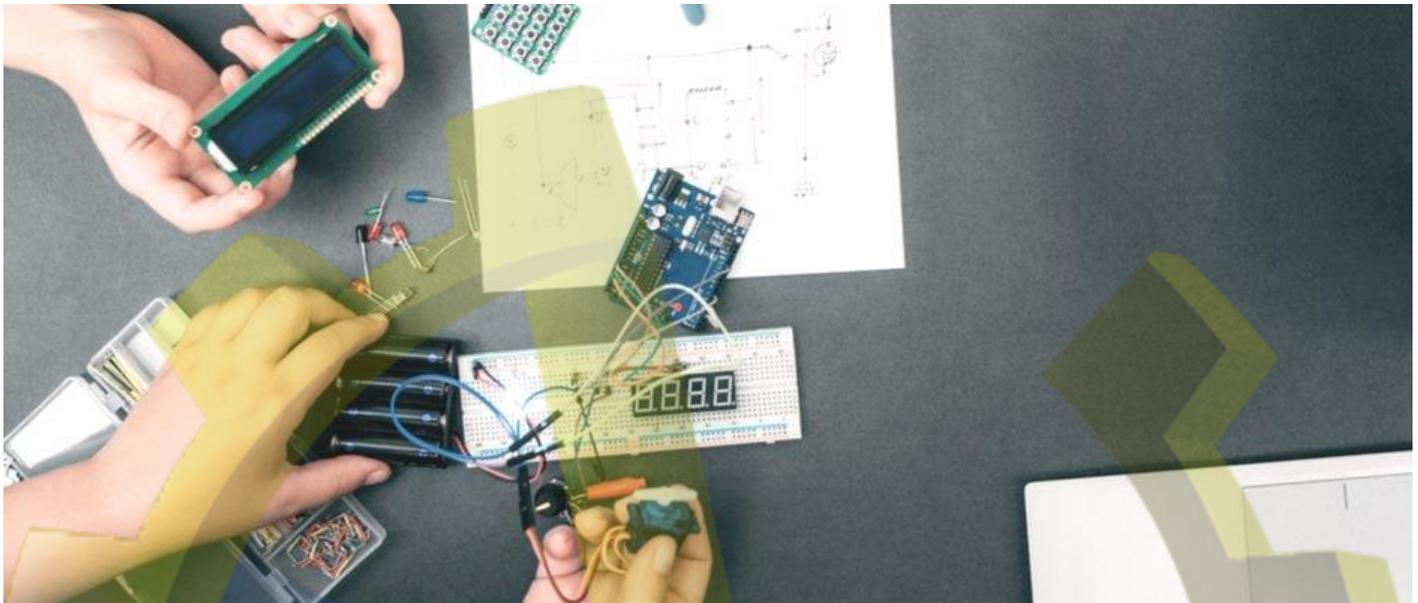


An Introduction to Arduino



[Arduino](#) is a user-friendly, open-source platform for working with microcontrollers to build electronic projects. The platform includes a suite of programmable [circuit boards](#) (microcontrollers), the IDE (Integrated Development Environment), and related accessories. From DIY espresso machines to indoor aquaponics to art installations and [IoT devices](#), your imagination's the limit when it comes to the kinds of things you can build with Arduino. In this article we'll introduce some of the key concepts behind this popular electronics prototyping platform.

How does Arduino Work?

Ever wonder how your coffee machine knows what temperature to heat your coffee to? The tiny brain behind your typical commercial coffee machine is a microcontroller that does basic things like receive the analog input from a thermistor and output the temperature onto a digital display. The same can be said of a remote control car which receives commands from a wireless controller over RF waves, allowing you to control the motors that spin the wheels. The

Arduino platform gives you a programmable board (microcontroller) with inputs and outputs that can be used to bring your physical projects to life.

THE ARDUINO IDE

The Arduino IDE is the program you use to write the code that tells your Arduino board what to do. The software is based on [Processing](#), a popular IDE and software sketchbook geared toward the visual and electronic arts industry, and teaching people the basics of how to program. Like Processing, Arduino's clear syntax, simple API, and low barrier to entry make it an excellent platform for [prototyping](#) ideas or teaching people how to code. Arduino's user-friendly IDE makes the platform a favorite among students, engineers, and artists alike.

The Arduino language is actually a set of [C/C++](#) functions that can be called from within your code, which means standard C and C++ constructs will work in Arduino, provided they are [avr-g++ supported](#). Since the Arduino board is just another AVR development board, you can also use other build tools like Makefiles and/or AVR Studio in place of the Arduino IDE. You can also download a plugin for your C++ IDE of choice like [Eclipse](#).

ARDUINO PRODUCT LINE

Arduino caters to a diverse clientele of teachers, students, hackers, hobbyists, engineers, artists, and pretty much anyone interested in adding a little interactivity to their projects or environments. Fortunately, Arduino has an [impressive catalog of boards, modules, shields, and kits](#) to match. Let's take a look at some of the broader categories that make up the Arduino Product Line.

Boards

The cornerstone of the Arduino platform is its line of AVR microcontroller boards. A typical Arduino [board](#) will have a USB jack connecting to a computer, a series of analog and digital I/O pins, [reset](#) button, power LED indicator, TX/RX (short for transmission/receiving) LEDs, a main IC (integrated circuit), and a voltage regulator. All of these features are present on the [Arduino Uno](#) which is the flagship entry-level model that most people are familiar with. However the exact configuration will vary from one board to another depending on the intended application, so this list is by no means exhaustive.

Shields

Need to add some additional functionality to your Arduino board? Try equipping a shield—a pre-built circuit board that attaches to the top of an Arduino board for added capability—wireless communication, LCD screen, cellular support, controlling motors, etc. If there's something you're trying to accomplish chances are there's a shield that can help you out.

Modules

Modules are simply a smaller-form factor of classic Arduino boards. They are great for projects where space is a premium and you don't need all the extra pins you might get on a typical board. For example, the [Arduino Micro](#) is basically just an [Arduino Leonardo](#) in a smaller form factor.

Wearables

Arduino's line of “wearable” boards for soft textile projects are instantly recognizable by their small size and circular form factor. Boards like the Arduino GEMMA or LILYPAD were made to be woven into clothing or accessories to build everything from Alzheimer's assistants to [flame throwing gauntlets](#).

Truth be told, the last one mentioned was actually done with an [Arduino Pro Mini](#), but it drives home the point that these categories are really just guidelines; it's ultimately up to you how you use a particular board.

The Sensors

An Arduino board can't do anything without its sensors, the devices that can measure a range of things from temperature to radioactivity. Let's take a look at some of the basic categories of sensors you'll encounter in beginner projects:

- **Thermistors**, which are resistors that change resistance depending on the temperature can be used to measure temperature.

- **Photocells**, which are resistors that change resistance depending on the amount of light they are exposed to can be used to detect light.
- **Flex sensors** increase in resistance as the sensor is flexed, and can be used to detect the degree to which something is bent, like a door.
- **Pressure sensors**, are typically piezoresistors that change in resistance depending on how hard they are pressed.
- **Accelerometers** detect acceleration, and are often used to measure the static acceleration due to gravity for tilt sensing applications or to measure the dynamic acceleration of a moving object.

This list is by no means exhaustive. There are sensors for measuring things like barometric pressure, carbon monoxide, humidity, and much more. If you can name it, you can probably sense it. The key here is that the Arduino possesses the analog pins you need to take that raw analog input and do something with it.

USING ARDUINO TO MAKE IOT DEVICES

What do you get when you connect your Arduino project to a network? An [Internet of Things \(IoT\)](#) device of course. Arduino has a suite of products dedicated to connecting your projects to the web, but really you can use any board provided you have a means of hooking them up to a network.

Some boards come equipped with a fully networked computer like the [Arduino YÚN](#), which comes with WIFI and Ethernet support, and an Atheros AR9331 processor that runs Linino OS, a Linux distribution based on OpenWrt. If you're looking to save money, you can equip one of the more basic Arduino boards with a [Wireless SD Shield](#) for wireless capability.

Of course nothing says you have to remain exclusively within the confines of the Arduino ecosystem to create your IoT device. The [Johnny-Five](#) framework is a JavaScript Robotics and IoT platform that allows sensors connected through Arduino devices to be controlled from a Raspberry Pi. Following the hub and spoke model, you can imagine a single server hosted on a Raspberry Pi, managing all your sensors and Arduino devices to turn your home into a smart house.