

Arduino

Introduction:

An Arduino is an open-source microcontroller development board. In plain English, you can use the Arduino to read sensors and control things like motors and lights. This allows you to upload programs to this board which can then interact with things in the real world. With this, you can make devices which respond and react to the world at large.

For instance, you can read a humidity sensor connected to a potted plant and turn on an automatic watering system if it gets too dry. Or, you can make a stand-alone chat server which is plugged into your internet router. Or, you can have it tweet every time your cat passes through a pet door. Or, you can have it start a pot of coffee when your alarm goes off in the morning.

Basically, if there is something that is in any way controlled by electricity, the Arduino can interface with it in some manner. And even if it is not controlled by electricity, you can probably still use things which are (like motors and electromagnets), to interface with it.

Why Arduino?

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.

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

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

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.

Before you can start doing anything with the Arduino, you need to download and install the Arduino IDE (integrated development environment)

Programming Language:

Arduino doesn't run either **C** or **C++**. It runs machine code compiled from either **C**, **C++** or any other language that has a compiler for the **Arduino** instruction set using special rules of code structuring. However, it's possible to **use Arduino** with **Python** or another high-level programming language. ... If **you** already know the basics of **Python**,

then **you'll** be able to get started with **Arduino** by using **Python** to control it.

Types of Arduino:

There are a number of different types of Arduinos to choose from.

1. Arduino Uno - Best Arduino for Beginners. ...
2. Arduino Nano - Best Arduino for Breadboards and Portable Projects. ...
3. Arduino Mega - Best Arduino for Advanced Users. ...
4. Arduino Due - Best Arduino for Running Large Programs. ...
5. Arduino Ethernet REV 3 - Best Arduino for IoT Projects. ...
6. Arduino MKR Zero - Best Arduino for Audio.

Arduino Uno

The most common version of Arduino is the Arduino Uno. This board is what most people are talking about when they refer to an Arduino. In the next step, there is a more complete rundown of its features.

Arduino Uno is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.

It allows the designers to control and sense the external electronic devices in the real world.

UNO is not the only board in the Arduino family. There are other Arduino boards However, the Arduino UNO board became more popular than other boards in the family because it has documentation that is much more detailed. This led to its increased adoption for electronic prototyping, creating a vast community of electronic geeks and hobbyists.

In recent times, the UNO board has become synonymous with Arduino.

The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.



components of Arduino UNO board

The major components of Arduino UNO board are as follows:

- USB connector
- Power port
- Microcontroller
- Analog input pins
- Digital pins
- Reset switch
- Crystal oscillator
- USB interface chip
- TX RX LEDs

Now let's take a closer look at each component.

USB connector:

This is a printer USB port used to load a program from the Arduino IDE onto the Arduino board. The board can also be powered through this port.

Power port:

The Arduino board can be powered through an AC-to-DC adapter or a battery. The power source can be connected by plugging in a 2.1mm center-positive plug into the power jack of the board.

2.1mm center-positive plug

The Arduino UNO board operates at a voltage of 5 volts, but it can withstand a maximum voltage of 20 volts. If the board is supplied with a higher voltage, there is a voltage regulator (it sits between the power port and USB connector) that protects the board from burning out.

Microcontroller (Atmega328P microcontroller):

It is the most prominent black rectangular chip with 28 pins. Think of it as the brains of your Arduino. The microcontroller used on the UNO board is Atmega328P by Atmel (a major microcontroller manufacturer).

Atmega328P has the following components in it:

- **Flash memory** of 32KB. The program loaded from Arduino IDE is stored here.
- **RAM** of 2KB. This is a runtime memory.
- **CPU**: It controls everything that goes on within the device. It fetches the program instructions from flash memory and runs them with the help of RAM.
- **Electrically Erasable Programmable Read Only Memory (EEPROM)** of 1KB. This is a type of nonvolatile memory, and it keeps the data even after device restart and reset.

Atmega328P is pre-programmed with bootloader. This allows you to directly upload a new Arduino program into the device, without using any

external hardware programmer, making the Arduino UNO board easy to use.

Analog input pins:

The Arduino UNO board has 6 analog input pins, labeled “Analog 0 to 5.” These pins can read the signal from an analog sensor like a temperature sensor and convert it into a digital value so that the system understands. These pins just measure voltage and not the current because they have very high internal resistance. Hence, only a small amount of current flows through these pins.

Although these pins are labeled analog and are analog input by default, these pins can also be used for digital input or output.

Digital pins:

You can find these pins labeled “Digital 0 to 13.” These pins can be used as either input or output pins. When used as output, these pins act as a power supply source for the components connected to them. When used as input pins, they read the signals from the component connected to them. When digital pins are used as output pins, they supply 40 milliamps of current at 5 volts, which is more than enough to light an LED.

Some of the digital pins are labeled with tilde (~) symbol next to the pin numbers (pin numbers 3, 5, 6, 9, 10, and 11). These pins act as normal digital pins but can also be used for Pulse-Width Modulation (PWM), which simulates analog output like fading an LED in and out.

Reset switch:

When this switch is clicked, it sends a logical pulse to the reset pin of the Microcontroller, and now runs the program again from the start. This can

be very useful if your code doesn't repeat, but you want to test it multiple times.

Crystal oscillator:

This is a quartz crystal oscillator which ticks 16 million times a second. On each tick, the microcontroller performs one operation, for example, addition, subtraction, etc.

USB interface chip:

Think of this as a signal translator. It converts signals in the USB level to a level that an Arduino UNO board understands.

TX – RX LEDs:

TX stands for transmit, and RX for receive. These are indicator LEDs which blink whenever the UNO board is transmitting or receiving data.

Applications:

- Embedded System
- Security and Defense System
- Digital Electronics and Robotics
- Parking Lot Counter
- Weighing Machines
- Traffic Light Count Down Timer
- Medical Instrument
- Emergency Light for Railways
- Home Automation
- Industrial Automation