



# Comparing the Arduino Nano vs Uno: How to Choose the Right Board for Your Project

By Linux Code / November 3, 2023

For makers and hobbyists, few platforms have had as big an impact as the open-source Arduino microcontroller boards. Arduino has enabled everything from basic DIY electronics projects to advanced robotics applications.

Of the many Arduino boards available, two of the most popular options are the Arduino Nano and Arduino Uno. At first glance, the Nano and Uno look quite similar – after all, they both have the standard Arduino board layout and components. However, under the hood there are some important distinctions between the two boards that are worth considering when selecting which one to use for your next Arduino project.

In this comprehensive guide, we'll examine how the Arduino Nano and Uno boards stack up across a variety of categories including size, specifications, performance, accessories, and use cases. Read on to learn how to choose the right Arduino board for your needs.

## A Brief History of Arduino

To understand the similarities and differences between Arduino boards like the Nano and Uno, it helps to know the origins of Arduino.

The initial concept for Arduino was developed in 2005 in Ivrea, Italy by a team led by Massimo Banzi as an affordable and accessible open-source microcontroller platform for students and hobbyists. The goal was to make it easier for non-engineers to create electronics prototypes and everyday objects that could sense and interact with the physical world.

The team drew inspiration from open-source programming languages like Processing and wiring platform created for artist projects. In 2005, the team produced the very first Arduino board based on an 8-bit Atmel AVR microcontroller and a development environment to program it.

Since then, Arduino has exploded in popularity around the world as an easy-to-learn platform for DIY electronics and rapid prototyping. The open-source ecosystem enables both software and hardware extensibility. Today there are many official and third-party Arduino boards available. Two of the most widely used models are the Arduino Uno and Arduino Nano boards.

## Introducing the Arduino Nano

The Arduino Nano is a small, breadboard-friendly board that was introduced in 2008. Some key features of the Arduino Nano include:

- Based on the ATmega328P microcontroller (as of version 3.x)
- Small form factor – only 18mm x 45mm
- Weighs just 7 grams
- 16MHz clock speed
- 32KB flash memory and 2KB SRAM
- Mini USB connector for power and communication
- 22 digital I/O pins – 6 can be used for PWM output
- 8 analog input pins

- Reset button
- Onboard power regulation from USB or external source

The compact size of the Nano along with the robust feature set has made it a popular choice for all kinds of Arduino projects, from simple homemade gadgets to advanced robotics applications. The Nano's light weight and variety of power options also make it ideal for portable and wearable electronics builds.

Over the years, the Nano board has been revised from the original design. The current version 3.x introduced in 2018 uses the Atmega328P, adding more memory over the Atmega168 chip used in earlier versions.

## Introducing the Arduino Uno

The Arduino Uno is perhaps the most well-known Arduino board, serving as the flagship model. The Uno was first released in 2010 as an evolution of earlier boards like the Arduino Duemilanove. It quickly became the standard Arduino board used by beginners and experts alike.

Some key capabilities of the Arduino Uno include:

- Based on ATmega328P microcontroller
- Clock speed of 16 MHz
- 32KB flash memory and 2KB SRAM
- 14 digital I/O pins and 6 analog input pins
- USB connector for power and communication
- Power via USB, DC barrel jack or Vin pin
- ICSP header for in-circuit programming and debugging
- Reset button to reboot the board

With abundant memory, I/O capabilities and onboard power regulation, the Arduino Uno can serve as the controller for all kinds of electronics projects from simple to complex. The larger form factor also leaves plenty of room for connecting wires, sensors and other components.

Since its release, the Uno design has been revised and updated as well. The current R3 version was released in 2011 and represented an evolutionary upgrade rather than dramatic changes, further cementing the Uno as a foundational Arduino board.

## Size Comparison

One of the most obvious differences between the Arduino Nano and Uno is their size. Let's take a closer look at the dimensions:

- **Arduino Nano:** 18mm x 45mm
- **Arduino Uno:** 68mm x 53mm

As you can see, the aptly named Nano board has a much smaller footprint – less than half the size of the Uno!

This size can be significant when integrating the boards into projects where compactness is key. For example, here is a simple Arduino project that toggles an LED on and off:

```
int ledPin = 5; // LED connected to digital pin 5

void setup() {

    pinMode(ledPin, OUTPUT); // Set pin 5 as output

}

void loop() {

    digitalWrite(ledPin, HIGH); // Turn LED on

    delay(1000); // Wait 1 second

    digitalWrite(ledPin, LOW); // Turn LED off
```

```
    delay(1000); // Wait 1 second
}
```

This LED blink code will run on both the Nano and Uno, but the Nano's tiny form factor lends itself better to applications like small wearable devices where minimal footprint is desired.

Some other examples where the Nano's compact size gives it an advantage:

- Small robots and RC vehicles
- IoT and smart home nodes
- Miniature drones and toys

Of course, the Uno's larger dimensions provide benefits as well – the extra space allows for integrating more robust circuits and shields. It also makes prototyping simpler since connections aren't as cramped.

Ultimately, consider the available space when selecting between the Nano and Uno and opt for the petit Nano when size and form factor are important.

## Technical Specifications

Beyond just size, let's explore some of the technical specifications and capabilities of each board:

specification	Arduino Nano	Arduino Uno
Microcontroller	ATmega328P	ATmega328P
Operating Voltage	5V	5V
Input Voltage Limit	7-12V	6-20V
Digital I/O Pins	22 (6 PWM)	14 (6 PWM)
Analog Input Pins	8	6
DC Current per I/O Pin	40 mA	20 mA
Flash Memory	32 KB (16 KB – older versions)	32 KB
SRAM	2 KB	2 KB
Clock Speed	16 MHz	16 MHz
Length	18mm	68mm
Width	45mm	53mm
Weight	7g	25g

As the table illustrates, the Nano and Uno share many common specs being based around variants of the ATmega328 microcontroller. However, we can draw some key comparisons:

- The Nano has more available I/O pins overall, providing 8 analog inputs over 6 on the Uno. This makes it more flexible for connecting sensors and peripherals.
- The Nano can deliver 40mA per I/O pin, double the current of the Uno. This makes it better suited for driving motors, LEDs, and other higher power devices directly.
- The Nano operates on a wider input voltage range (7-12V), while the Uno can accept up to 20V. The extended input range lets you power the Nano from a wider variety of sources.
- Older versions of the Nano had 16KB of flash storage compared to the Uno's 32KB. However, current Nanos now have 32KB as well to provide storage space for complex programs.

So in terms of capabilities, both boards are quite similar. But the Nano does provide advantages like more I/O, higher power delivery for components, and a wider operational voltage range – ideal features for electronics projects with space constraints.

# Power Sources

Another area where the Nano and Uno differ is the available power sources for each board. Let's compare the power options:

## Arduino Nano

- USB port
- 5V pin from an external power supply
- 7-12V pin from an external power source
- Onboard voltage regulator accepts up to 12V input

## Arduino Uno

- USB port
- DC power jack for 7-12V wall adapter
- Vin pin for regulated 5V power supply
- No onboard voltage regulator

The key item to note is that the Nano has an on-board voltage regulator that allows it to be powered from an unregulated 7-12V power source, such as a 9V battery. This provides more flexibility compared to the Uno.

The Nano's regulator means it can be powered portably for use in projects away from a computer. And you can drive the Nano at higher voltages more suitable for motors and sensors.

In contrast, the Uno relies on regulated 5V power sources only. This does limit its use in portable applications or scenarios where you need to deliver higher operational voltages.

So for battery-powered or remote projects, the Nano's onboard voltage regulation gives it the advantage. But the Uno simplifies wiring when using a standard 5V power brick.

# Communication Interfaces

Arduino boards like the Nano and Uno provide built-in hardware support for various wired communication protocols to communicate with peripherals and other devices. Let's look at the communication interfaces available on each:

## Arduino Nano

- I2C
- SPI
- UART (serial)

## Arduino Uno

- I2C
- SPI
- UART (serial)

In terms of communication capabilities, the boards are identical – both support I2C, SPI and serial interfaces. These allow you to connect the Arduino to all kinds of external components like sensors, displays, motors and more.

For example, here is Arduino code to initialize communication with an I2C temperature sensor:

```
#include <Wire.h>

void setup() {

    Wire.begin(); // Initialize I2C communication

}

void loop() {
```

```
// Code to get temperature readings from sensor over I2C

}
```

This will work seamlessly on both the Nano and Uno since both have the requisite I2C interface. Similar code can leverage the SPI and Serial interfaces as well.

So you'll have equal built-in communication abilities with the Nano and Uno for projects that need to talk to peripherals and other hardware.

## Shields and Accessories

Another consideration is compatibility with Arduino add-on boards called shields that provide additional capabilities and components. For example, Arduino Ethernet and motor driver shields.

Both the Arduino Nano and Uno are designed to be compatible with shields conforming to the original Arduino Uno footprint. However, the way that the boards can interface with shields differs.

The Arduino Uno has extended header pins that plug directly into shields for the most secure physical connection. Many shields are designed specifically for the Uno footprint.

The Nano lacks extended header pins, so is not as seamlessly integrated with shields. But it can still be used with shields by soldering jumper wires between the Nano pins and shield headers. Or using a special adapter board.

So the Uno has the edge for directly plugging into shields. But the Nano is still shield compatible with a bit of wiring work.

There are also many Arduino Nano-sized boards and shields becoming available as the Nano has grown in popularity. So you can find Nano-specific accessories.

Ultimately both boards can interface with the vast array of Arduino shields and components to expand their capabilities through add-on boards. But the Uno's design provides the simplest out-of-the-box shield integration.

## Sample Projects

To better illustrate how the Nano and Uno can be applied, let's look at some sample project ideas that play to the strengths of each board:

### Arduino Nano Projects

- Wearable pedometer with accelerometer
- Tiny motorized robot or drone
- Wireless home sensors powered by battery
- Car dashboard data display
- Kickstarter notifier pulling online data

### Arduino Uno Projects

- Home automation controller
- Robotic arm with servos
- Smart weather station
- Aquarium controller
- Laser tripwire alarm
- 3D printer controller

For projects where small footprint is critical – like wearables and mini robotics – the Nano is a natural fit. For more complex projects with many wired components, the Uno makes it easier to integrate shields and accessories.

Of course, you can use either board for most any Arduino project with proper wiring. But the factors outlined above make each board better suited for certain applications.

## Arduino Software Considerations

Another consideration between the Nano and Uno is compatibility with the Arduino Software (IDE). This Java-based IDE allows you to write and upload code to any Arduino board.

The good news is that both boards are fully supported by the latest Arduino IDE and work with the vast majority of existing Arduino libraries and code. You can use the IDE to code a project, then upload it seamlessly to either a Nano or Uno.

However, there are some small hardware differences that mean not all Nano boards work with older IDE versions. For example, the Nano's 16U2 USB interface chip requires IDE version 1.6.12 or higher.

So as long as you keep your IDE up to date, you'll have no issues with cross-compatibility. But it's something to check if uploading code to an older Nano. The Uno has wider IDE support.

## Debugging and Troubleshooting

When working with any microcontroller board, you'll inevitably need to debug issues in malfunctioning circuits or buggy code. Let's discuss some debugging considerations for each board.

For **code debugging**, both the Nano and Uno can leverage the same techniques:

- Liberal use of `Serial.print()` statements to output diagnostic data
- Using the IDE's built-in Serial Monitor to watch debug info
- LEDs to visually indicate program flow and pin states
- Tools like `printf()` and assertions to add runtime checks

For **circuit debugging**, options like:

- Multimeter to check voltages and continuity
- Oscilloscope to monitor signal waveforms
- Current meter to check power consumption
- Jumper wires and breadboard for structured troubleshooting
- Swapping in known working components

Overall, standard debugging practices apply equally to both boards. The Uno's larger size does provide some advantages for troubleshooting circuits since connections are less cramped.

## Community and Learning Resources

Given their widespread adoption, both the Arduino Nano and Uno have huge communities with abundant tutorials and projects to leverage. A quick search will turn up documentation on most any use case or project idea.

Some key resources include:

- Official Arduino project hub with thousands of projects
- Detailed function documentation on [arduino.cc](http://arduino.cc)
- Massive discussion forums to find solutions
- YouTube tutorials covering both basic and advanced topics
- Instructables and Hackaday for project ideas

There are also countless books, blogs, and courses teaching Arduino using these boards. So beginners and experts alike will have no shortage of learning content.

While examples often default to the Uno, resources for the Nano are plentiful as well. Just search for "Arduino Nano" instead of "Arduino Uno" when looking for Nano-specific guidance.

## Conclusion

While the Arduino Nano and Uno share the same underlying processor and capabilities, their differences make each model better suited for particular types of DIY electronics projects.

The petite Nano is ideal where a tiny footprint is needed – like in wearable devices. Its extra I/O pins and onboard voltage regulation also make it great for battery-powered mobile projects.

The full-sized Uno shines for its abundance of space for connections. Its seamless integration with shields also simplifies building prototypes and more advanced circuits.

Hopefully this guide has illuminated factors like size, specs, power, and use cases to consider when choosing a board for your next Arduino project. The good news is you can't go wrong with either of these extremely versatile boards that form the backbone of the Arduino ecosystem.

So go forth and build something awesome with your newfound knowledge! Whether it's a tiny robot or expansive home automation system, the Nano and Uno have you covered.

[← Previous Post](#)

[Next Post →](#)

## Related Posts

### 2D Array – An In-Depth Guide for Arduino

[Arduino](#)

Arrays are an essential component of programming in Arduino. They allow you to store multiple values of the same data type in a single variable....

### 3 Different Ways to Power Up the Arduino Nano

[Arduino](#)

Hey there! Welcome to this beginner-friendly guide on powering up the popular Arduino Nano board. The compact Nano makes it easy to build all kinds...

### 3 Ways to Solar Power an Arduino with Linux

[Arduino](#)

Hello fellow Linux enthusiast! Are you looking for the best way to power your next Arduino project using solar energy? If so, you've come to...

### A Complete Guide to Arduino Boards and Models

[Arduino](#)

Arduino is an incredibly popular open-source electronics platform used by millions of hobbyists and professionals alike for DIY electronics projects. At the heart of every...

### A Comprehensive Guide to Exporting and Using Arduino BIN Files

[Arduino](#)

Boards like the Arduino Uno, Nano and Mega 2560 have exploded in popularity in recent years. Powering everything from DIY robots to interactive art installations,...

### A Detailed Guide to Reading Analog Signals with Arduino

[Arduino](#)

Introduction Welcome! This guide will provide you with a deep understanding of how to read and interpret analog signals from sensors using Arduino's handy `analogRead()`...

Search...



## Top 10 Linux Code Tips (for the topic),

1. [Arduino Nano vs Arduino Uno – A Comprehensive Technical Comparison](#)
2. [Voltage Regulation in the Arduino Uno Board](#)
3. [Arduino Nano vs Arduino Nano 33 IoT: A Detailed Comparison](#)
4. [Which Arduino Board Should I Choose? A Beginner's Guide to Picking the Right Model](#)
5. [Deciphering Arduino: How to Select the Right Microcontroller Board for Your Project](#)
6. [Raspberry Pi 3 vs Pi 4: How the Single-Board Computer Has Evolved and Which Should You Choose](#)
7. [Banana Pi vs Raspberry Pi: How to Choose the Best Single Board Computer for Your Application](#)
8. [Arduino LilyPad vs Arduino UNO – A Detailed Comparison](#)
9. [From Basic to Smart: Arduino UNO vs Arduino Yún](#)
10. [Installing the Powerful ESP32 Board in Arduino IDE: A Friendly, Step-by-Step Guide](#)

## Topics





[About](#)  
[Contact](#)  
[Privacy Policy](#)  
[Terms & Conds](#)  
[Disclaimer](#)  
[Cookies](#)