NDSU | NORTH DAKOTA STATE UNIVERSITY

Tutorial

Arduino Uno Simulator using Wokwi

Contents

1 Introduction

1.1 What is Arduino Uno?

The Arduino Uno is one of the most widely used microcontroller development boards. It is built around the ATmega328P microcontroller, a low-power 8-bit chip that makes it easy for beginners and professionals to create electronic projects. Key features of the Arduino Uno include:

- **14 Digital Input/Output Pins** → Can be used to read digital signals (e.g., button press) or control devices (e.g., turning an LED ON/OFF).
- **6 Analog Input Pins** → Allow the board to read continuous signals, such as voltage from sensors (e.g., temperature or light intensity).
- **PWM (Pulse Width Modulation)** → Some digital pins can generate analog-like signals, useful for dimming LEDs or controlling motor speed.
- **USB Connectivity** → Used for programming the board and providing power.

The Arduino Uno is popular because it is open-source, affordable, and beginner-friendly, making it an excellent choice for learning embedded systems and hardware programming.

1.2 Why Simulate?

In traditional labs, you would need physical boards, sensors, wires, and other hardware. While this is important for real-world applications, it can be challenging for beginners due to wiring mistakes, hardware costs, or limited availability. By using a simulator, we can:

- **Work Safely** → No risk of burning components or damaging hardware.
- **Save Time and Cost** → No need to purchase parts; everything is virtual.
- **Debug Easily** → Visual tools help track signals, values, and errors in real time.
- **Experiment Freely** → Try ideas without worrying about breaking the board.

Simulation allows students to focus on learning logic, coding, and system design before handling real-world electronics.

1.3 Overview of Wokwi

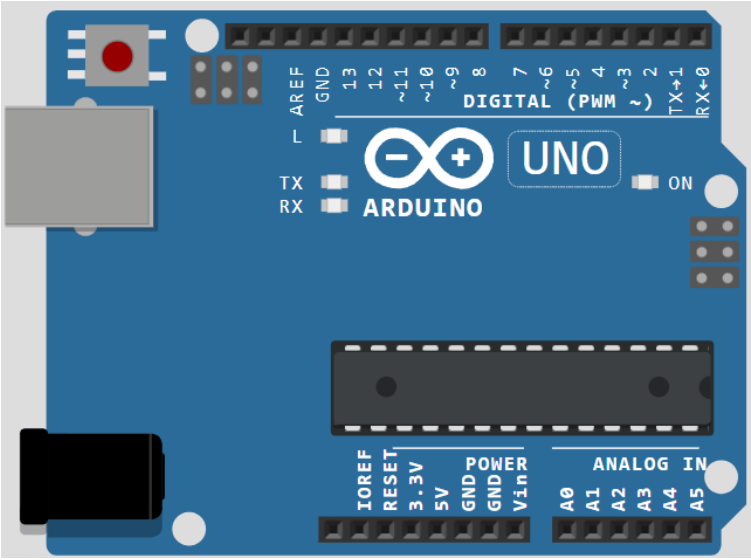
Wokwi is an online Arduino simulator that works directly in your web browser. No installation or setup is required. Key advantages of Wokwi:

- **Free and Online** → Runs on any computer with internet access.
- **Arduino Support** → Fully supports Arduino Uno, Nano, Mega, and other boards.
- **Wide Range of Components** → LEDs, buttons, sensors, motors, displays, and more.
- **Real-Time Simulation** → Code runs instantly, and you can interact with components (e.g., pressing buttons, adjusting potentiometers).
- **Serial Monitor Support** → Just like the real Arduino IDE, you can see program output in real time.

With Wokwi, you can practice coding in C for Arduino, build circuits, and simulate projects without any physical hardware, making it ideal for classroom teaching and learning.

2 Arduino Uno Board

In the Wokwi simulator, the Arduino Uno board looks exactly like the physical version, and all its components work the same way during simulation. Features available on the board are detailed below.



2.1 Power

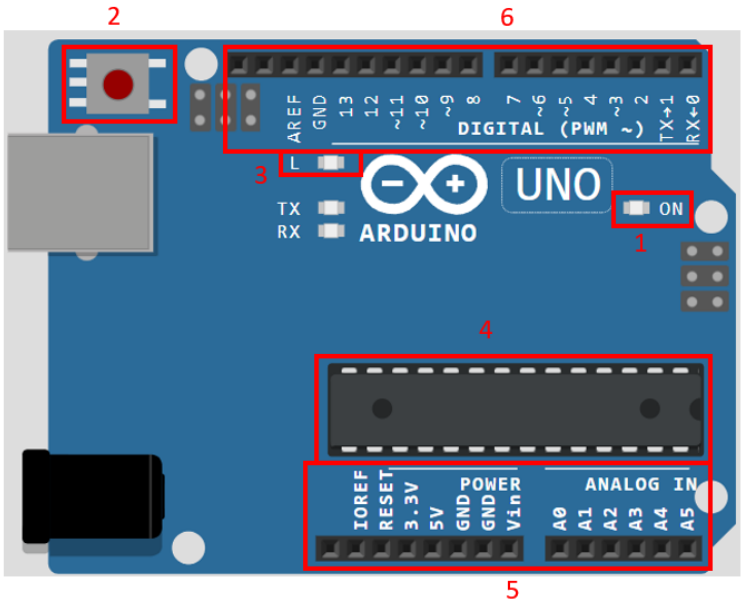
In Wokwi simulation, the board is automatically powered ON as soon as you click the green “Start Simulation” button. You don’t need to connect power manually. The power indicator is shown in red box 1 in the figure below. If the LED is glowing, it means the board is receiving power and your code is running.

2.2 Reset

The Reset button on the Arduino Uno is located near the USB connector at the top left of the board as shown in red box 2 in the below figure. Its purposes are 1) Restarts the microcontroller (ATmega328P), 2) Stops the running program and begins executing it again from the start, and 3) Useful for debugging, testing, or restarting your circuit without stopping the whole simulation. You can click the Reset button directly with your mouse. The program restarts instantly, just as it would on the real board. The ON power LED stays lit (showing the board still has power), but your code runs from the beginning.

2.3 On-board LED

The Arduino Uno board has a built-in LED as shown in red box 3 in the below figure. It’s a single LED that is labeled with the letter ”L” on the board. This LED is connected to digital pin 13 in most Arduino boards.



2.4 Microcontroller

The microcontroller on the Arduino Uno is the ATmega328P, which serves as the "brain" or "heart" of the board. Its primary purpose is to execute the program that you upload to it. In the Wokwi simulator, it functions exactly as it would on the physical board. Its purpose is to serve as the central processing unit, executing the code you write to control the board's behavior. The Wokwi simulator accurately models this behavior, allowing you to see how your program affects the microcontroller and any connected components. The microcontroller is shown in red box 4.

2.5 Analog and Power Pins

The Analog and Power pins are two crucial groups of pins on the Arduino Uno that manage how the board interacts with real-world signals and how it receives or provides electrical energy. Both are modeled accurately in the Wokwi simulator and are shown in red box 5.

The Arduino Uno board has six Analog Input pins, labeled A0 through A5. These pins are found near the Power header on the bottom right of the board. Their primary role is to measure continuous signals from analog sensors, such as reading the varying voltage output from a temperature sensor or a potentiometer. They are connected to an internal Analog-to-Digital Converter (ADC) inside the ATmega328P microcontroller. The ADC takes the analog voltage (which can be any value between 0V and 5V) and converts it into a digital number ranging from 0 to 1023 (a 10-bit resolution). Although labeled "Analog In," these pins can also be configured and used as General Purpose Digital Input/Output (GPIO) pins if you need extra digital pins in your project.

2.6 Digital Pins

The Digital Input/Output (I/O) Pins on the Arduino Uno board (Pins 0 to 13) are the primary way the microcontroller interacts with external components using a binary, or two-state, logic system. The Arduino Uno has 14 digital I/O pins in total, numbered D0 to D13. These pins function as General-Purpose Input/Output (GPIO), meaning they can be configured to:

- Input Mode: Used to read digital signals (a value of either HIGH or LOW) from a device, such as reading whether a button has been pressed or a sensor has crossed a threshold. HIGH represents the board's operating voltage, which is 5V on the Uno. LOW represents 0V (Ground).

- Output Mode: Used to control devices by setting the pin voltage to either HIGH (5V) or LOW (0V), such as turning an LED ON/OFF or activating a relay.

3 Language and Sketches

3.1 The Language

The programming language used for Arduino, and thus within the Wokwi simulator, is based on C++ but is simplified by the Wiring framework. This language is specifically designed for ease of use in electronics and embedded programming. It abstracts complex hardware register manipulation into simple, clear functions like `pinMode()`, `digitalWrite()`, and `analogRead()`.

3.2 The Sketch

An Arduino program is referred to as a sketch. Every sketch is structured around two mandatory functions:

- `void setup()`: This function is the initialization block. it runs only once immediately after the program starts or the board is reset, and its primary purpose is to set up the necessary initial conditions for the Arduino. Within this function, you configure hardware settings such as defining which digital pins will act as inputs or outputs using the `pinMode()` function, starting any necessary serial communication with `Serial.begin()`, and initializing external libraries or variables.

- `void loop()`: This function is executed directly after the completion of the `setup()`. This is the main execution block of the sketch. This function runs the program's primary logic continuously and repeatedly until simulation is stopped, allowing the Arduino to perform its ongoing tasks, such as constantly checking the state of sensors with `digitalRead()`, controlling actuators with `digitalWrite()` or `analogWrite()`, and responding to changing conditions in the environment.