Components Presentation

Agenda

Presentation of Arduino

Digital and analog pins

Resistors

Breadboard

Arduino IDE

Arduino simulation platforms

Presentation of Arduino

A microcontroller

Microcontrollers typically contain a processor, memory (RAM and ROM), and input/output peripherals. They are designed for specific control task.

Arduino meets the features of a microcontroller  duino is a microcontroller

Arduino - Brief history

The Arduino project was launched in 2005 at the Interaction Design Institute Ivrea in Ivrea, Italy,

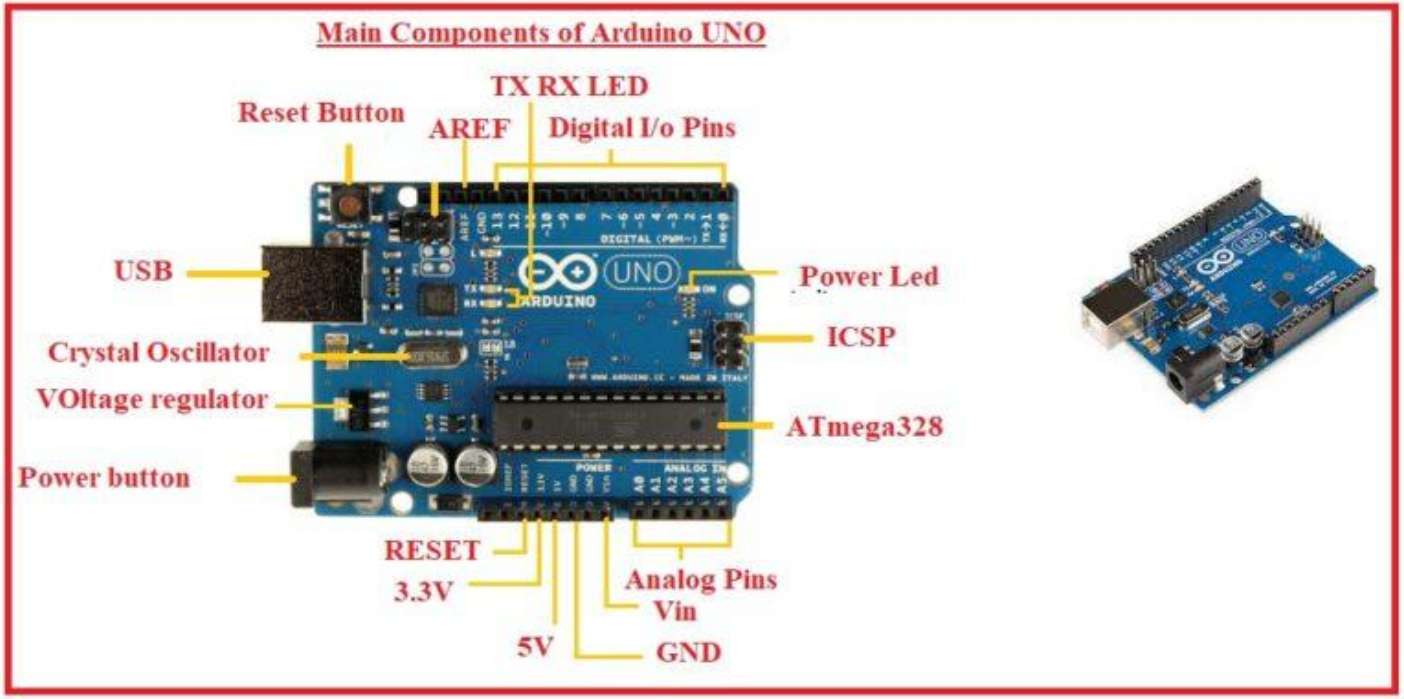
A cost-effective and user-friendly platform designed to help both beginners and experts create interactive devices using sensors and actuators.

Arduino - Microcontroller

Is an open-source electronics platform featuring user-friendly hardware and software.

It can read inputs — such as light on a sensor or a press of a button — and convert them into outputs, like activating a motor or illuminating an LED.

Arduino Uno

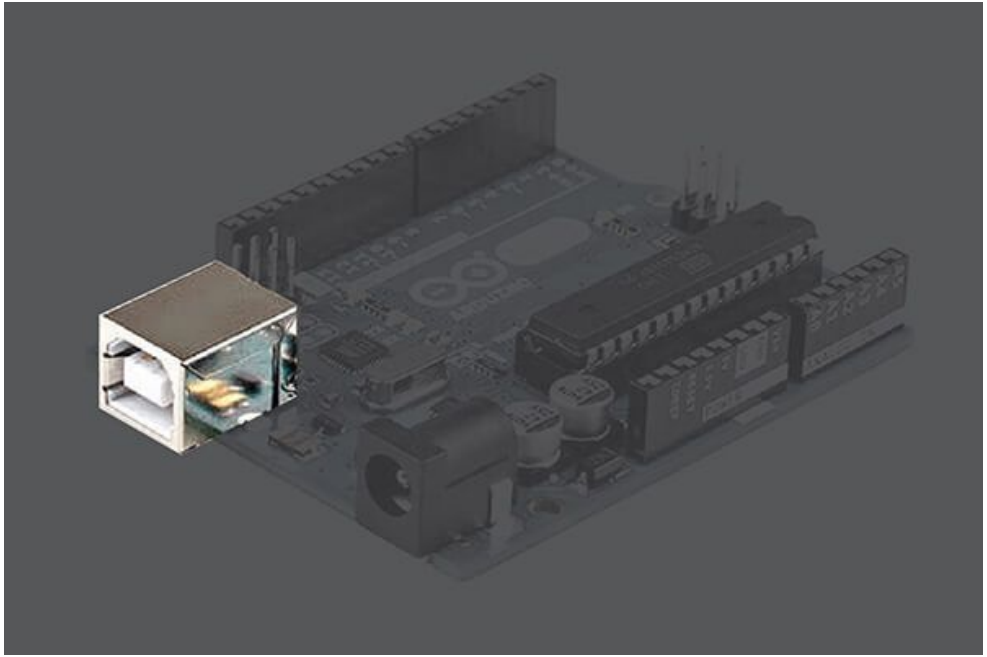

Main Components of Arduino UNO
TX RX LED
Reset Button
AREF
Digital I/o Pins
USB
Power Led
Crystal Oscillator
ICSP
VOltage regulator
ATmega328
Power button
RESET
3.3V
Analog Pins
Vin
5V
GND

Arduino Uno

USB Connector

Used to load a program from the Arduino IDE onto the Arduino board

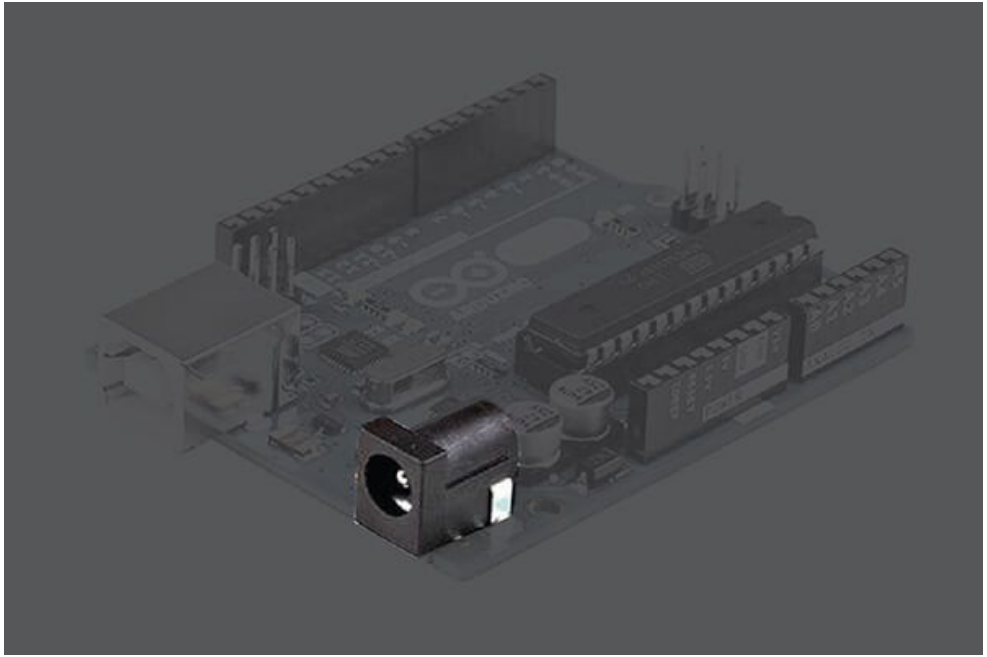
It can also be powered through this port.



Arduino Uno

Power port

The Arduino board can be powered through an AC-to-DC adapter or a battery.



Arduino Uno

Atmega328P microcontroller (The brain)

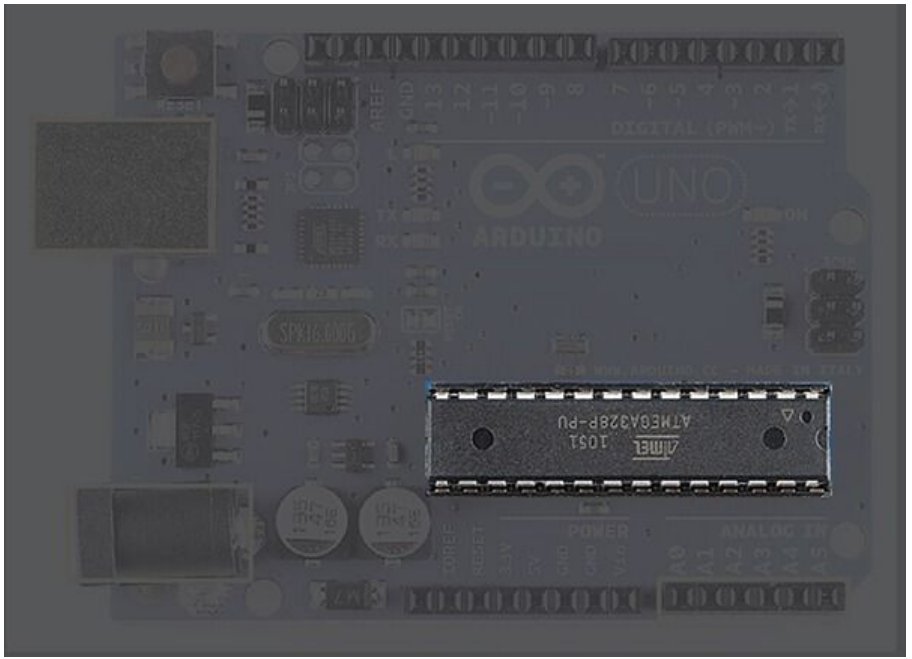
Atmega328P has the following components in it:

Flash memory of 32KB.

RAM of 2KB

CPU

Electrically Erasable Programmable Read Only Memory (EEPROM) of 1KB



Arduino Uno

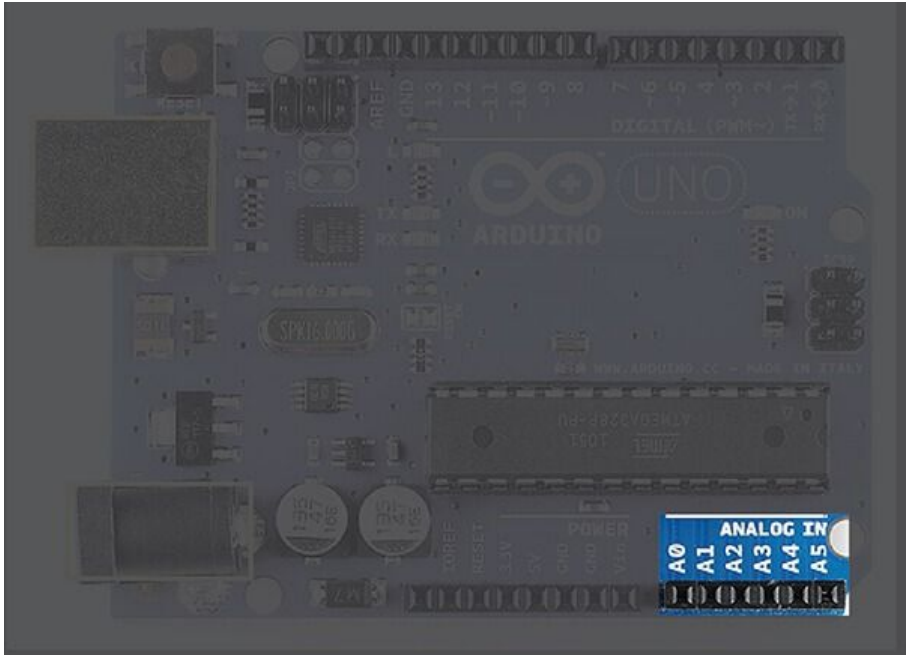
Analog input pins

The Arduino UNO board has 6 analog input pins, labeled **Analog 0 to 5**.

These pins can read the signal from an analog sensor and convert it into a digital value.

Although these pins are labeled analog and are analog input by default,

These pins can also be used for digital input or output.



Arduino Uno

Digital pins

These pins are labeled Digital 0 to 13.

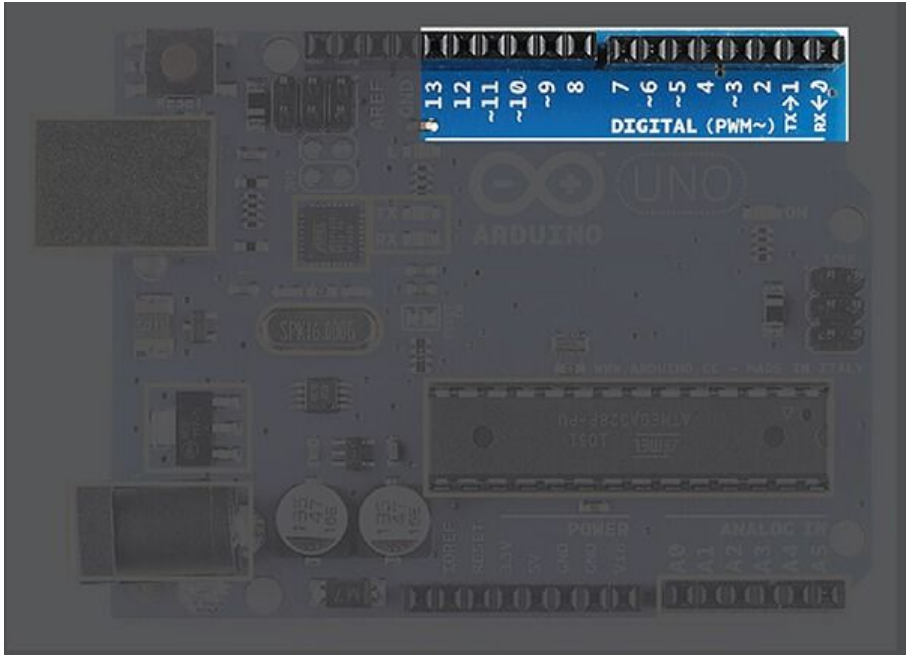
These pins can be used as either input or output pins.

When used as output, they act as a power supply source.

When used as input pins, they read the signals.

Some of the digital pins are labeled with tilde (~)

These pins act as normal digital pins but can also be used for Pulse-Width Modulation (PWM), which simulates analog output like fading a LED in and out.

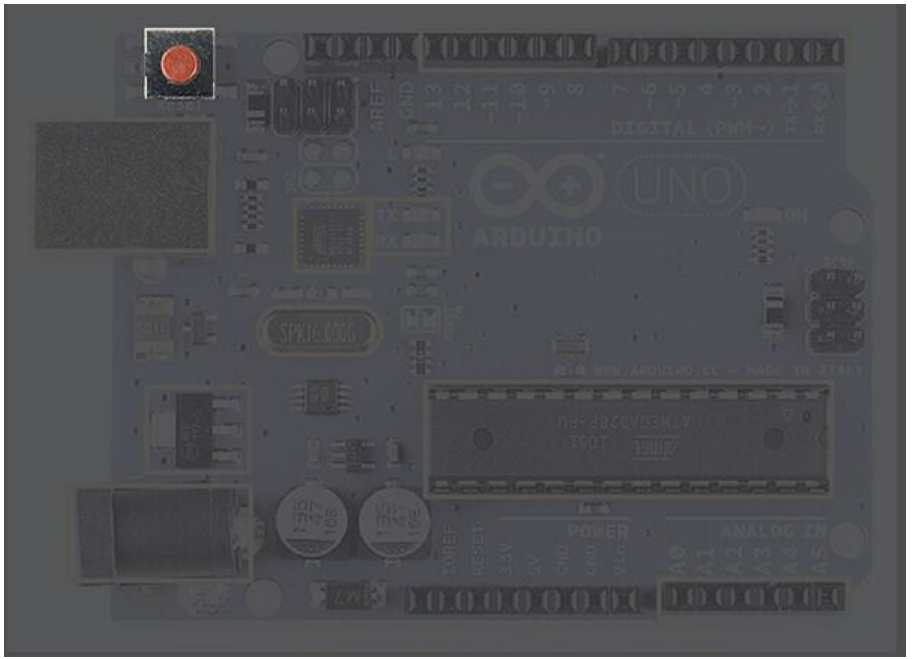


Arduino Uno

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.

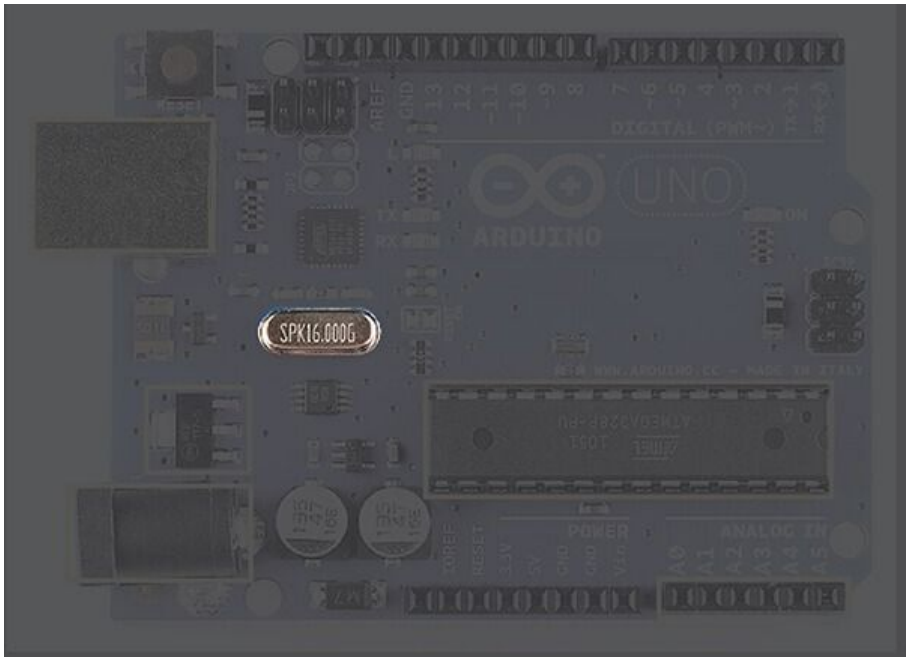


Arduino Uno

Crystal oscillator

It 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.

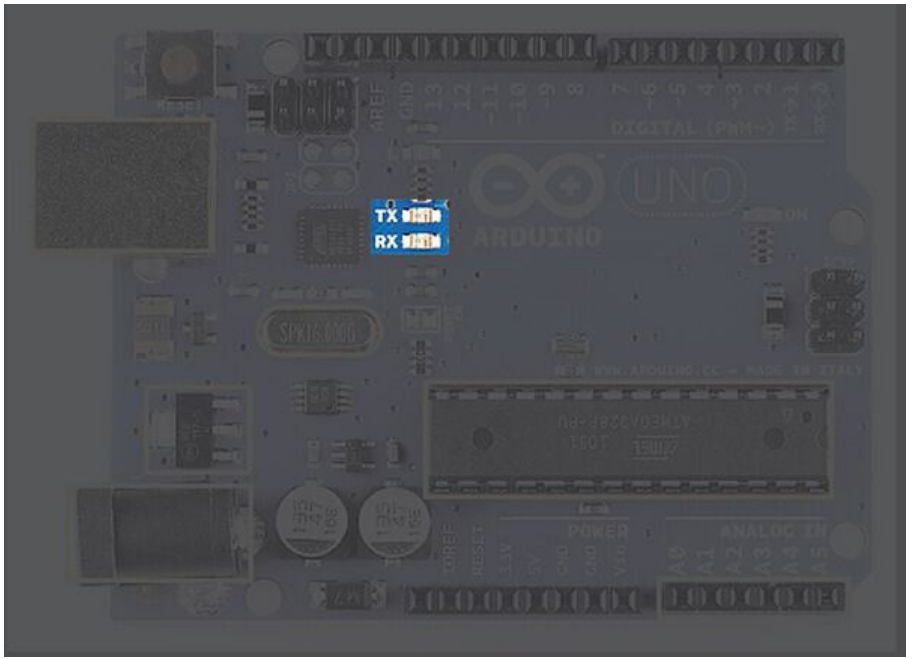


Arduino Uno

TX - RX indicator

TX stands for transmit, and RX for receive.

These are indicator LEDs which blink whenever the UNO board is transmitting or receiving data.



Digital and Analog pins

Types of signals used in Arduino:

Digital signal: is a signal that represents two states (two values) High (5 Volts) and Low (0 Volts). In Arduino, you can use the digital pins to work with digital signals, where those pins can be set to either HIGH (5 volts) or LOW (0 volts), which allows you to control external devices (led for example) or read digital inputs from sensors, buttons, etc.

Functions to be used:

```
digitalRead(digital-Pin-Number)

digitalWrite(digital-Pin-Number, HIGH/LOW)
```

Types of signals used in Arduino:

Analog signal: in contrast to digital signal with only two states, the analog signal can represent continuous values within the interval [0 V - 5 V].

The volt levels in [0 - 5] should be, of course, converted to numerical values to be used. These values will in [0-1023] in case of arduino Uno and Mega since they use precision of 10 bits.

The function to be used to read an analog signal:

```
analogRead(analog-pin)
```

Please note that Analog pins allow only to read an analog signal, not to write an analog signal.

Types of signals used in Arduino:

In case you need to output (write) an analog signal you should use the PWM pins with the function analogWrite:

`analogWrite(PWM-pin, value)`, where value should be in the interval [0-255]

The PWM pins can be used to control the speed of motors, the brightness of LEDs, etc.

Resistors

How to measure the values

To calculate the appropriate value of the resistor to be used in case of using components with arduino, you should consider the following key parameters:

`Vs= source voltage` (ex. Arduino pins)

`Vl: Load voltage` (ex. LED voltage, motor voltage, …etc.)

`I: current` (LED current, motor current)

Load voltage and electric current must be found in the datasheet of the component (led, motor,...)

How to measure the values

Ohm law:

```
R = v/l
```

Hence:

```
R = (Vs - Vl)/l Ohm
```

You will not always find a resistor that meets exactly the calculated value, so you can use the closest one to the calculated value

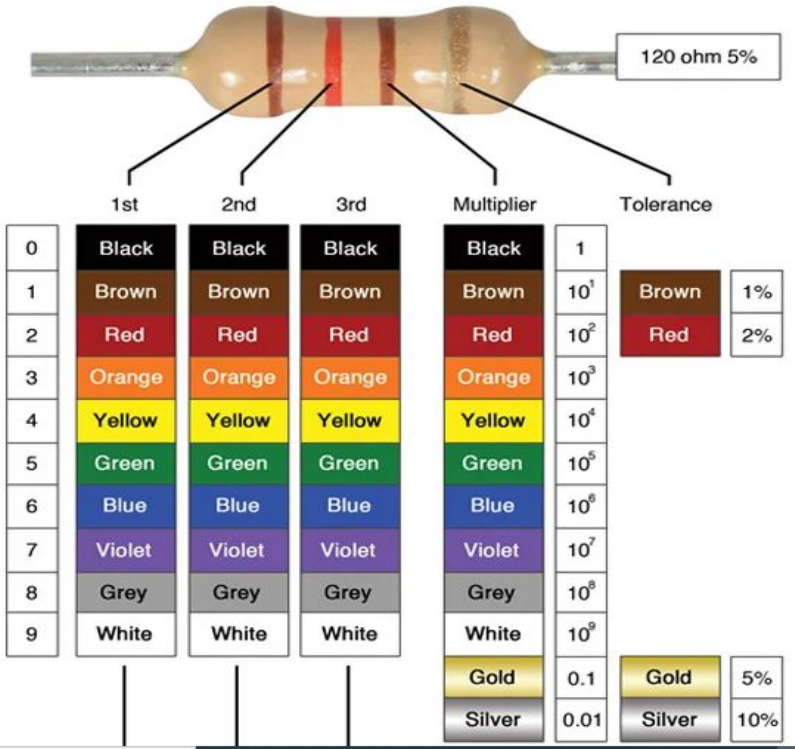
How to measure the values

The value of the resistor depends on the number of bands on it (3 bands, 4 bands, 5 bands,....)

	3 Band Resistor	4 Band Resistor	5 Band Resistor	6 Band Resistor
1st band	First Digit	First Digit	First Digit	First Digit
2nd band	Second Digit	Second Digit	Second Digit	Second Digit
3rd band	Multiplier Value	Multiplier Value	Third Digit	Third Digit
4th band		Tolerance Value	Multiplier Value	Multiplier Value
5th band			Tolerance Value	Tolerance Value
6th band				Temperature Coefficient

Understanding colors

Tolerance value represents the resistor's nominal value, which indicates how much the actual resistance can vary from the marked value



Understanding colors

For example, if you have a resistor with a nominal value of 100 ohms and a tolerance of ±10%, the calculation for the tolerance would look like this:

1. Lower value: 100 ohms - (10% of 100 ohms) = 100 - 10 = 90 ohms

2. Upper value: 100 ohms + (10% of 100 ohms) = 100 + 10 = 110 ohms

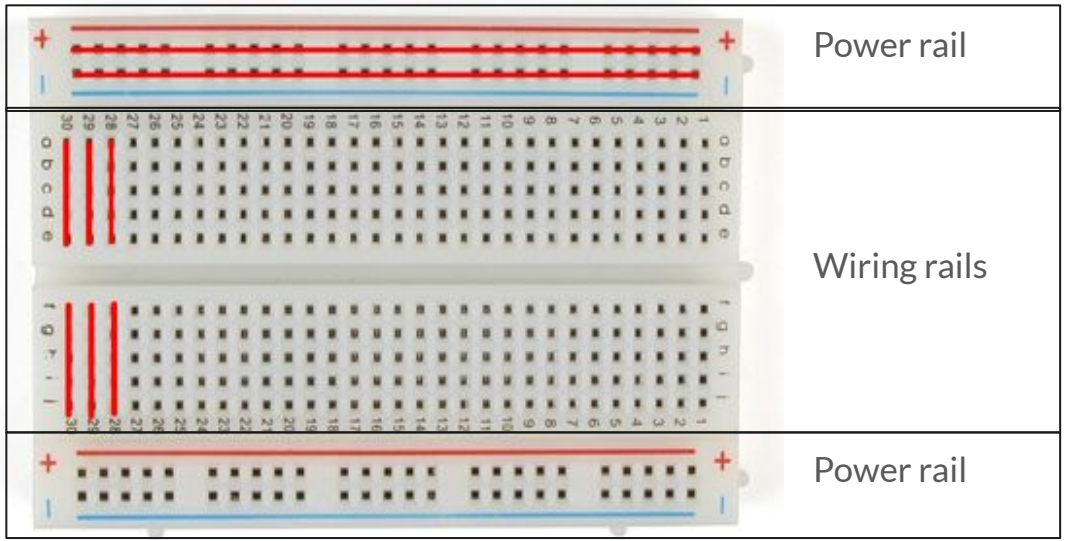
So, for a 100-ohm resistor with a ±10% tolerance, the actual resistance could range from 90 ohms to 110 ohms.

You can use this web site to calculate the value of the resistors:

https://www.calculator.net/resistor-calculator.html

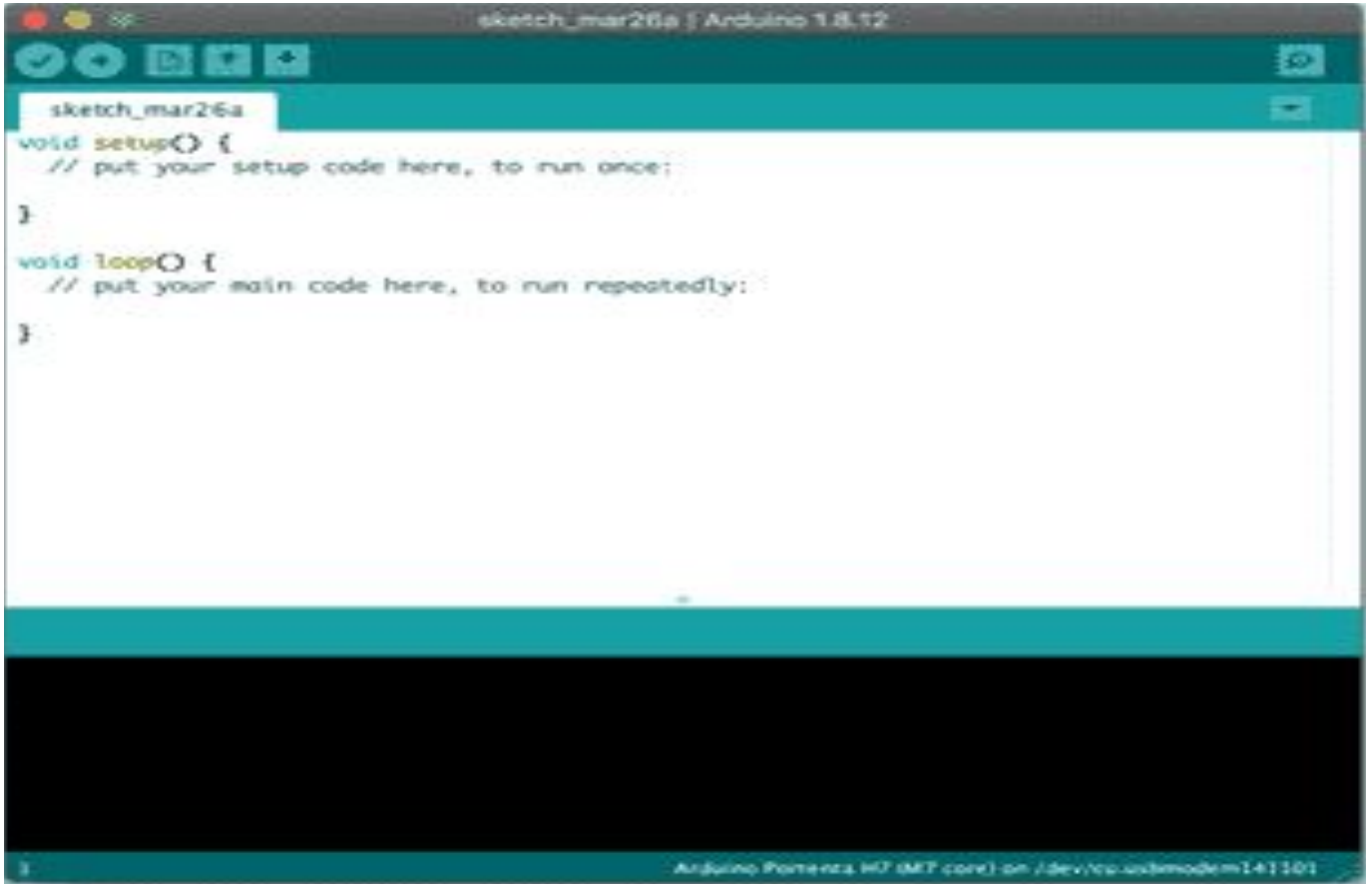
Breadboard

Breadboard



Arduino IDE

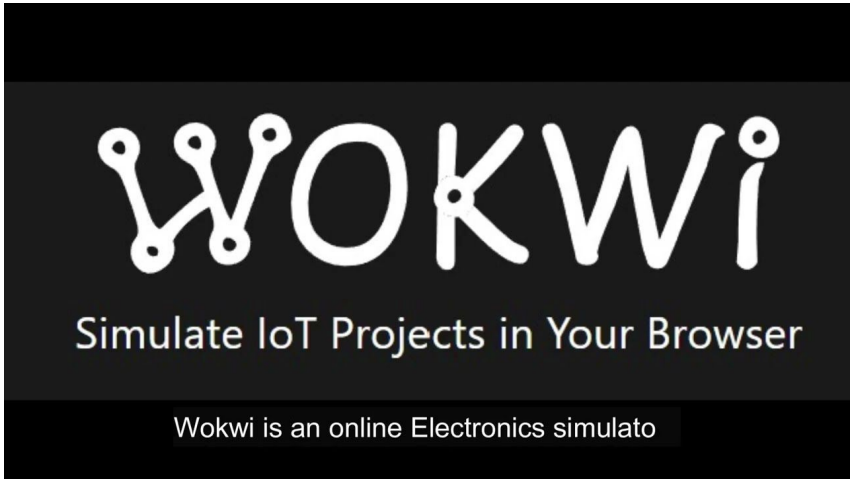
Arduino IDE


sketch_mar26a | Arduino 1.8.12
sketch_mar26a
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}

Arduino simulation plateforms

Tinkercad

Wokwi


WOKWi
Simulate IoT Projects in Your Browser
Wokwi is an online Electronics simulato