

This page is also available in **2 other languages** [Change language](#)

English

## LANGUAGE

### FUNCTIONS

#### VARIABLES

#### STRUCTURE

#### LIBRARIES

#### IOT CLOUD API

#### GLOSSARY

The Arduino Reference text is licensed under a [Creative Commons Attribution-Share Alike 3.0 License](#).

Find anything that can be improved? [Suggest corrections and new documentation via GitHub](#).

Doubts on how to use Github? Learn everything you need to know in [this tutorial](#).



Last Revision: 2019/06/21

Last Build: 2021/12/04

[EDIT THIS PAGE](#)

Reference > Language > Functions > Time > Delay

# delay()

[Time]

## Description

Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

## Syntax

delay(ms)

## Parameters

ms: the number of milliseconds to pause. Allowed data types: unsigned long.

## Returns

Nothing

## Example Code

The code pauses the program for one second before toggling the output pin.

```
int ledPin = 13; // LED connected to digital pin 13

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() {
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
}
```

Help

While it is easy to create a blinking LED with the `delay()` function a many sketches use short delays for such tasks as switch debouncing the use of `delay()` in a sketch has significant drawbacks. No other reading of sensors, mathematical calculations, or pin manipulation can go on during the delay function, so in effect, it brings most other activity to a halt. For alternative approaches to controlling timing see the [Blink Without Delay](#) sketch, which loops, polling the `millis()` function until enough time has elapsed. More knowledgeable programmers usually avoid the use of `delay()` for timing of events longer than 10's of milliseconds unless the Arduino sketch is very simple.

Certain things do go on while the `delay()` function is controlling the Atmega chip, however, because the delay function does not disable interrupts. Serial communication that appears at the RX pin is recorded, PWM (`analogWrite`) values and pin states are maintained and `interrupts` will work as they should.

### See also

EXAMPLE [Blink Without Delay](#)

[Back to top](#)[Trademark](#)[Help Center](#) **NEWSLETTER**[Contact Us](#)[Distributors](#)[Careers](#)

Enter your email to sign up

**FOLLOW US**

;C

© 2021 Arduino

[Terms Of Service](#)[Privacy Policy](#)[Security](#)[Cookie Settings](#)

Help