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

delay(1000); // waits for a second

}

Notes and Warnings

While it is easy to create a blinking LED with the delay() function and 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](http://arduino.cc/en/Tutorial/BlinkWithoutDelay) sketch, which loops, polling the [millis()](https://www.arduino.cc/reference/en/language/functions/time/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](https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite)) values and pin states are maintained, and [interrupts](https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt) will work as they should.