analogWrite()

[Analog I/O]

Description

Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive motor at various speeds. After a call to `analogWrite()`, the pin will generate a steady rectangular wave o specified duty cycle until the next call to `analogWrite()` (or a call to `digitalRead()` or `digitalWrite()`) on t same pin.

BOARD	PWM PINS *	PWM FREQUENCY
UNO (R3 and earlier), Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)
UNO R4 (Minima, WiFi) *	3, 5, 6, 9, 10, 11	490 Hz
Mega	2 - 13, 44 - 46	490 Hz (pins 4 and 13: 980 Hz)
GIGA R1 **	2 - 13	500 Hz
Leonardo, Micro, Yún	3, 5, 6, 9, 10, 11, 13	490 Hz (pins 3 and 11: 980 Hz)
UNO WiFi Rev2, Nano Every	3, 5, 6, 9, 10	976 Hz
MKR boards *	0 - 8, 10, A3, A4	732 Hz
MKR1000 WiFi **	0 - 8, 10, 11, A3, A4	732 Hz
Zero **	3 - 13, A0, A1	732 Hz
Nano 33 IoT **	2, 3, 5, 6, 9 - 12, A2, A3, A5	732 Hz
Nano 33 BLE/BLE Sense ****	1 - 13, A0 - A7	500 Hz
Due ***	2-13	1000 Hz
101	3, 5, 6, 9	pins 3 and 9: 490 Hz, pins 5 an 980 Hz

* These pins are officially supported PWM pins. While some boards have additional pins capable of PWM using them is recommended only for advanced users that can account for timer availability and potenti conflicts with other uses of those pins.
** In addition to PWM capabilities on the pins noted above, the MKR, Nano 33 IoT, Zero and UNO R4 boa have true analog output when using `analogWrite()` on the `DAC0` (`A0`) pin.
*** In addition to PWM capabilities on the pins noted above, the Due and GIGA R1 boards have true ana output when using `analogWrite()` on pins `DAC0` and `DAC1`.
**** Only 4 different pins can be used at the same time. Enabling PWM on more than 4 pins will abort th running sketch and require resetting the board to upload a new sketch again.

You do not need to call `pinMode()` to set the pin as an output before calling `analogWrite()`.
The `analogWrite` function has nothing to do with the analog pins or the `analogRead` function.

Syntax

LANGUAGE
FUNCTIONS
VARIABLES
STRUCTURE
LIBRARIES
IOT CLOUD API
GLOSSARY

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: 2024/02/18

Last Build: 2024/04/22

EDIT THIS PAGE

Parameters

`pin`: the Arduino pin to write to. Allowed data types: `int`.

`value`: the duty cycle: between 0 (always off) and 255 (always on). Allowed data types: `int`.

Returns

Nothing

Example Code

Sets the output to the LED proportional to the value read from the potentiometer.

```
int ledPin = 9;      // LED connected to digital pin 9
int analogPin = 3;   // potentiometer connected to analog pin 3
int val = 0;         // variable to store the read value

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

void loop() {
  val = analogRead(analogPin);  // read the input pin
  analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023, analogWrite values from 0 t
}
```

Notes and Warnings

The PWM outputs generated on pins 5 and 6 will have higher-than-expected duty cycles. This is because interactions with the `millis()` and `delay()` functions, which share the same internal timer used to gener those PWM outputs. This will be noticed mostly on low duty-cycle settings (e.g. 0 - 10) and may result in value of 0 not fully turning off the output on pins 5 and 6.

See also

LANGUAGE analogWriteResolution()

DEFINITION PWM

EXAMPLE Blink

Back to top