analogWrite()

On the NANO do not use pins 5 & 6 if you want to run PWM at higher frequencies, even though it is tempting as they start at 980Hz. Instead use any of the other PWM capable pins, marked in green, and then modify TCCR2B and OCR2A as shown in example sketch called Modify\_PWM\_frequency.ino

[Analog I/O]

Description

Writes an analog value ([PWM wave](http://arduino.cc/en/Tutorial/PWM)) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

| BOARD | PWM PINS | PWM FREQUENCY |
| --- | --- | --- |
| Uno, Nano, Mini | 3, 5, 6, 9, 10, 11 | 490 Hz (pins 5 and 6: 980 Hz) |
| Mega | 2 - 13, 44 - 46 | 490 Hz (pins 4 and 13: 980 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 and 6: 980 Hz |

\* In addition to PWM capabilities on the pins noted above, the MKR, Nano 33 IoT, and Zero boards have true analog output when using analogWrite() on the DAC0 (A0) pin.  
\*\* In addition to PWM capabilities on the pins noted above, the Due has true analog output when using analogWrite() on pins DAC0 and DAC1.

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

analogWrite(pin, value)

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 to 255

}

Notes and Warnings

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